Tanium Gateway User Guide

2026H2 Semiannual Release

Version 1.43

November 19, 2026



The information in this document is subject to change without notice. Further, the information provided in this document is provided "as is" and is believed to be accurate, but is presented without any warranty of any kind, express or implied, except as provided in Tanium's customer sales terms and conditions. Unless so otherwise provided, Tanium assumes no liability whatsoever, and in no event shall Tanium or its suppliers be liable for any indirect, special, consequential, or incidental damages, including without limitation, lost profits or loss or damage to data arising out of the use or inability to use this document, even if Tanium Inc. has been advised of the possibility of such damages.

Any IP addresses used in this document are not intended to be actual addresses. Any examples, command display output, network topology diagrams, and other figures included in this document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Please visit https://help.tanium.com for the most current Tanium product documentation.

This documentation may provide access to or information about content, products (including hardware and software), and servicesprovided by third parties ("Third Party Items"). With respect to such Third Party Items, Tanium Inc. and its affiliates (i) are not responsible for such items, and expressly disclaim all warranties and liability of any kind related to such Third Party Items and (ii) will not be responsible for any loss, costs, or damages incurred due to your access to or use of such Third Party Items unless expressly set forth otherwise in an applicable agreement between you and Tanium.

Further, this documentation does not require or contemplate the use of or combination with Tanium products with any particular Third Party Items and neither Tanium nor its affiliates shall have any responsibility for any infringement of intellectual property rights caused by any such combination. You, and not Tanium, are responsible for determining that any combination of Third Party Items with Tanium products is appropriate and will not cause infringement of any third party intellectual property rights.

Tanium is committed to the highest accessibility standards for our products. To date, Tanium has focused on compliance with U.S.Federal regulations - specifically Section 508 of the Rehabilitation Act of 1998. Tanium has conducted 3rd party accessibility assessments over the course of product development for many years. We certify against the WCAG 2.2 / VPAT 2.5 standards for all major product modules. Tanium performs ongoing reassessments as appropriate, given the scope and type of changes to the UI and product features from release to release. In the most recent testing, the Tanium Console UI achieved supports or partially supports for all applicable WCAG 2.2 criteria. Tanium can make available any VPAT reports on a module-by-module basis as part of a larger solution planning process for any customer or prospect.

As new products and features are continuously delivered, Tanium will conduct testing to identify potential gaps in compliance with accessibility guidelines. Tanium is committed to making commercially reasonable efforts to address any gaps as quickly as is feasible, given the severity of the issue and scope of the changes. Tanium factors these objectives into the ongoing delivery schedule of features and releases with our existing resources.

Tanium welcomes customer input on making solutions accessible based on your Tanium modules and assistive technology requirements. Accessibility requirements are important to the Tanium customer community and we are

©2026 Tanium Inc. All Rights Reserved

committed to prioritizing these compliance efforts as part of our overall product roadmap. Tanium maintains transparency on our progress and milestones and welcomes any further questions or discussion around this work. Contact your sales representative, Tanium Support atsupport@tanium.com, or email accessibility@tanium.com to make further inquiries.

Tanium is a trademark of Tanium, Inc. in the U.S. and other countries. Third-party trademarks mentioned are the property of the irrespective owners.

Kubernetes®, K8s®, and the Kubernetes logo are registered trademarks of The Linux Foundation®.

Android, ChromeOS, and Google are trademarks of Google LLC.

iOS, iPad, iPadOS, iPhone, Mac, and macOS are trademarks of Apple Inc., and registered in the U.S. and other countries and regions.

Patents: https://www.tanium.com/patents

Tanium Open Source Disclosure : https://tanium.github.io/opensource/

© 2026 Tanium Inc. All rights reserved.

Table of contents

Overview 15
- Gateway topology 15
- Query explorer 17
- Authentication 21
- Rate limits 21
- Root endpoint 22
- Example cURL syntax 22
- Pagination 22
- Filters 27
- Interoperability with other Tanium products 31

Getting started 33

Requirements 35
- Core platform dependencies 35
- Solution dependencies 35
  - Tanium recommended installation 35
  - Import specific solutions 36
  - Required dependencies 36
  - Feature-specific dependencies 36
- Tanium™ Module Server 37
- Endpoints 38
- Host and network security requirements 38
  - Ports 38
  - Security exclusions 38
  - API access 38
- User role requirements 39

Installing 41
- Before you begin 41

©2026 Tanium Inc. All Rights Reserved

Import Gateway 41
Upgrade Gateway 42
Verify Gateway version 42
Troubleshoot issues 42

Using Gateway 43
Create a persona limited to submitting only query requests 44
Test a query in Tanium Console 45
Use the Documentation Explorer to view the schema 46

Maintaining 48
Gateway monthly maintenance 48

Troubleshooting 49
Collect logs 49
Queries return unexpected results or errors 49
Request error handling 50
Syntax Validation 50
Structural Validation 50
Error Extensions 51
Error codes 55
Issue: error message for enumerated value displays the value in the request 58
Issue: ResourceExhausted error in response to valid query request 59
Issue: repeated 503 error code in responses 61
Uninstall Gateway 62
Contact Tanium Support 62

Reference: Gateway examples 63
System examples 63
Get server time 63
Get current user information 63
General examples 64
Get endpoints 65
Get endpoints and update sensor maximum age 66
Get endpoints IDs from Tanium Data Service 68
Get endpoints with filter excluding errors 69
Get endpoints with filter excluding [no results] 71

©2026 Tanium Inc. All Rights Reserved

Get endpoints using aliases 73
Get endpoints using multiple sensors 74
Get rich endpoint data 76
Get a set of endpoints 78
Get a set of endpoints and update sensor maximum age 80
Unregistered sensor query 83
(1.1 - Experimental (Active Development)) Get mobile devices 84
Get mobile devices from Tanium Data Service 85
Unregistered parameterized sensor query 86
Get selected columns from sensor query (query.endpoints.edges.node.sensorReadings.columns) 88
Get selected columns from a sensor query 88
Get hidden columns from a sensor query 89
Filter sensor query results (query.endpoints.edges.node.sensorReadings (sensors.filter)) 91
Filter the results from a sensor query 91
Filter sensor query results from a filtered set of endpoints 93
Apply multi-column filter to sensor query results 94
Apply multi-column filter to sensor query results for filtered endpoints 96
Filter sensor query results with restrictOwner option 98
Endpoint cursor query 101
Paginated query 102
Get refreshed collection of endpoints (query.endpoints.collectionInfo.startCursor, query.endpoints(refresh)) 103
Retrieve collection start cursor 103
Refresh collection using starting cursor 104
Software characteristics query with filter 105
Get applications installed on endpoint 107
Get Microsoft Sentinel endpoint information 108
(1.0 - Experimental (Early Development)) Get endpoint counts 110
Get the count of endpoints 110
Get the count of endpoints by entity type 112
(1.0 - Experimental (Early Development)) Get refreshed collection of endpoint counts using one sensor 117
Retrieve and refresh endpoint count collection start cursor using one sensor 117
(1.0 - Experimental (Early Development)) Get refreshed collection of endpoint counts using multiple sensors 120

©2026 Tanium Inc. All Rights Reserved

Retrieve and refresh endpoint count collection start cursor using multiple sensors 120
Refresh endpoint count collection using starting cursor 121
(1.1 - Experimental (Release Candidate)) Sort returned endpoints 123
(1.1 - Experimental (Active Development)) Get endpoints using single sensor 127
(1.2 - Experimental (Release Candidate)) Get endpoints using single sensor and converting result data type 129
(1.2 - Experimental (Release Candidate)) Get question ID 130
Action examples 131
Approve a scheduled action 131
Create action 132
Create scheduled action 135
Delete action or scheduled action 140
Get action information 140
Get multiple actions 143
Get package spec information 147
Get parameters for a specific package spec 148
Get scheduled action information 149
Get scheduled actions (query.scheduledActions) 151
Get multiple scheduled actions 151
Get scheduled actions (filter by name) 154
Perform an action operation 156
Target endpoints when deploying actions 161
Preview action 162
Target computer group to create scheduled action (scheduledActionCreate) 163
Perform action with sensor filter (actionPerform) 164
Show a scheduled action 165
Stop an action 166
DEPRECATED: Create action (subset of endpoints) 166
DEPRECATED: Get action details 167
Action group examples 168
Create action group 169
Delete action group by ID or name 169
Delete action group by ID 169
Delete action group by name 170
Get action group by ID or name 170

©2026 Tanium Inc. All Rights Reserved

Get action group by ID ... 171
Get action group by name ... 172
Get multiple action groups ... 172
API token examples ... 174
Create API token ... 174
Get API tokens for current user ... 176
Rotate API token ... 177
Delete API token ... 178
Asset examples ... 178
Get Asset products ... 179
Get endpoints based on Asset products ... 183
Update Asset endpoints based on source ... 185
Update Asset endpoints based on endpoint ID (EID) ... 189
Upsert (create or update by ID) Asset endpoint ... 191
Import assets ... 197
Automate examples ... 200
Add a playbook Add Custom Tag (Non-Windows) step ... 200
Add a playbook Add Custom Tag (Windows) step ... 203
Add a playbook Deploy Action step ... 206
Add email notifications to a playbook ... 208
Add a playbook Linux Quarantine step ... 212
Add a playbook Manual Instruction step ... 215
Add a playbook Reboot Non-Windows Machine step ... 219
Add a playbook Reboot Windows Machine step ... 222
Add a playbook Remove Linux Quarantine step ... 225
Add a playbook Remove Windows Quarantine step ... 227
Add a playbook Require User Confirmation step ... 230
Add steps that inherit endpoint targeting ... 234
Add a playbook Verify Condition step ... 243
Add a playbook Wait step ... 245
Add a playbook Windows Quarantine step ... 247
Create a playbook run from a playbook ... 250
Get a playbook by ID and version ... 253
Get a playbook run by ID ... 255
Get a playbook schedule by ID ... 258

©2026 Tanium Inc. All Rights Reserved

Get multiple playbooks 259
Get multiple playbook runs 263
Get multiple playbook schedules 270
Upsert (create or update by ID) a playbook 273
Upsert (create or update by ID) a playbook schedule 281
Delete a playbook 284
Delete a playbook schedule 290
Disable an enabled playbook schedule 292
Enable a disabled playbook schedule 294
End a playbook run 296
Pause a running playbook run 299
Resume a paused playbook run 302
Update playbook content set membership 305
Add Install Patch step to a playbook 306
Benchmark examples 310
Get endpoint with Risk Score 2.0 information 310
Comply examples 312
Get endpoints with compliance findings information 312
Get endpoint compliance findings information 312
Get filtered endpoint compliance findings information 314
Get filtered endpoint compliance findings information from filtered set of endpoints 317
Get endpoints with vulnerability findings information 319
Get endpoint vulnerability findings information 319
Get endpoint vulnerability findings information (CVSS v3 and CISA KEV) 321
Get endpoint vulnerability findings information (CVSS v4 and CISA KEV) 323
Get filtered endpoint vulnerability findings information 326
Get filtered endpoint vulnerability findings information from filtered set of endpoints 328
Computer group examples 330
Create a type of computer group 330
Create computer group 330
Create filter group 331
Create computer group that also filters 332
Create computer group based on existing computer groups 333
Create computer group based on sensor 334
Create computer group using multiple criteria 335

©2026 Tanium Inc. All Rights Reserved

Delete computer group by ID or name 336
Delete computer group by ID 336
Delete computer group by name 337
Get a computer group by ID or name 337
Get computer group by ID 337
Get computer group by name 338
Get multiple computer groups, filter groups, or management rights groups 339
Get computer groups 339
Get manual computer groups 340
(1.2 - Experimental (Release Candidate)) Get computer groups by name 341
Get standard computer groups 342
Get filter groups 343
Get management rights groups 345
(1.1 - Experimental (Active Development)) Get management rights groups current user and persona can access 346
Connect examples 347
Connect v1 REST API replacement 348
Get Connect connection by ID 349
Get Connect connection run by ID 350
Get multiple Connect connection runs 351
Get multiple Connect connections 352
Get multiple Connect connections by ID 354
Get multiple Connect connections by description 355
Get Connect connection by ID and displayName 356
Get SQL Server stored procedures 357
Get SQL Server stored procedures through a satellite 359
Start a Connect connection run 361
Stop the active run for a Connect connection 361
Stop a specific Connect connection run 362
Start multiple Connect connection runs 363
Stop multiple Connect connection runs 364
Enable multiple Connect connections for scheduled runs 365
Disable multiple Connect connections from scheduled runs 367
Delete multiple Connect connections 368
Transfer ownership of multiple Connect connections 369

©2026 Tanium Inc. All Rights Reserved

Change the persona of multiple Connect connections ... 370
Deploy examples ... 371
Deploy a package to all endpoints ... 372
Get all deploy packages ... 372
Get status of software deployment ... 373
LEGACY: Get details of all packages ... 374
Direct Connect examples ... 378
Open direct connection to endpoint ... 378
Get direct connection status ... 379
Ping direct connection to endpoint ... 380
Get data from endpoint ... 381
Get process from endpoint ... 382
Get past 24 hours of alerts from endpoint ... 383
Stop endpoint process ... 386
Close endpoint connection ... 386
LEGACY: Open a connection to an endpoint ... 387
LEGACY: Ping the connection to an endpoint ... 388
LEGACY: Get data from an endpoint ... 389
LEGACY: Get process from an endpoint ... 390
LEGACY - Get alerts from an endpoint ... 391
Get past 24 hours of alerts from endpoint ... 391
Get 48 hours of alerts from endpoint ... 393
Get alerts from prior date from endpoint ... 393
LEGACY - Stop a process on an endpoint ... 394
LEGACY - Close connection to an endpoint ... 395
Discover examples ... 396
Add Discover labels to Discover interfaces ... 396
Create a Discover label ... 397
Delete a Discover label by name ... 399
Get a Discover interface by ID ... 399
Get multiple Discover interfaces ... 402
Get a Discover label by name ... 406
Get multiple Discover labels ... 407
Remove Discover labels from Discover interfaces ... 408
Email examples ... 409

©2026 Tanium Inc. All Rights Reserved

(1.0 - Experimental (Early Development)) Get email server summary by ID 409
(1.0 - Experimental (Early Development)) Get multiple email server summaries 411
(1.0 - Experimental (Early Development)) Submit request to send email 412
Integrity Monitor examples 414
Create Integrity Monitor monitor 414
Create Integrity Monitor watchlist 417
Create Integrity Monitor watchlist Unix path 421
Create Integrity Monitor watchlist Windows file path 422
Create Integrity Monitor watchlist Windows registry path 424
Delete Integrity Monitor watchlist path 426
Deploy Integrity Monitor monitors 426
Deploy Integrity Monitor watchlists 427
Get Integrity Monitor monitor by ID 428
Get Integrity Monitor service status 429
Get Integrity Monitor watchlist by ID 430
Get multiple Integrity Monitor monitors 432
Get multiple Integrity Monitor watchlists 436
Mark Integrity Monitor monitors for deletion 439
Mark Integrity Monitor watchlists for deletion 440
Update Integrity Monitor monitor by ID 441
Update Integrity Monitor watchlist 444
Update Integrity Monitor watchlist Unix path 447
Update Integrity Monitor watchlist Windows file path 448
Update Integrity Monitor watchlist Windows registry path 450
Jump Gate examples 451
Create Jump Gate request 452
Patch examples 454
Get patch applicability 454
Create a Linux patch deployment 462
Create a Windows patch deployment 465
Get multiple patch definitions 470
Get a patch deployment by ID 472
Stop a patch deployment by ID 474
Upsert (create or update by ID) a patch list 477
Provision examples 482

©2026 Tanium Inc. All Rights Reserved

(1.0 - Experimental (Early Development)) Cancel retirement action 482
(1.0 - Experimental (Early Development)) Create retirement action 484
(1.0 - Experimental (Early Development)) Create retirement action definition 486
(1.0 - Experimental (Early Development)) Delete retirement action definitions 490
Get retirement action by ID 491
(1.0 - Experimental (Early Development)) Get retirement action definition by ID 494
(1.0 - Experimental (Early Development)) Get multiple retirement action definitions 496
(1.0 - Experimental (Early Development)) Get multiple retirement actions 499
(1.0 - Experimental (Early Development)) Resolve retirement action 505
(1.0 - Experimental (Early Development)) Update retirement action notes 507
(1.0 - Experimental (Early Development)) Update retirement action definition 510
Reporting examples 513
Delete a report by ID 514
Delete a Reporting dashboard by ID 514
Get a report by ID 515
Get an ID of a report 517
Get reports 518
Get report results by report ID 520
Export a report by ID 526
Import a report definition 526
Retrieve report results collection start cursor 527
Sensor examples 530
Get normal sensors 530
Get virtual sensors 532
Get sensors available to Tanium Data Service 535
Get hidden sensors 536
Get sensor parameters (query.getSensorParameters) 539
Get sensor parameters 539
Get sensor parameters showing hidden status 541
Register sensor for harvest (mutation.sensorHarvest) 543
Register a sensor for harvest 543
Register a sensor for harvest using a cursor 544
Threat Response example 545
Assign Resolved status to alert 545

©2026 Tanium Inc. All Rights Reserved

Reference: Deprecated fields **546**

Reference: Filter syntax **562**
- Actions query filter syntax 564
- Asset query filter syntax 567
- Automate query filter syntax 570
- Comply query filter syntax 598
- Computer group creation filter syntax 601
- Computer group query filter syntax 608
- Discover query filter syntax 611
- Endpoint query filter syntax 613
- Package spec query filter syntax 630
- Patch query filter syntax 632
- Provision query filter syntax 640
- Reporting query filter syntax 645
- Scheduled actions query filter syntax 649
- Sensor query filter syntax 652

Reference: Integration with Insomnia API Client **654**
- Prerequisites 654
- Configure Insomnia to communicate with Tanium 655
- Verify integration 656
- Generate code example from example request 657

Reference: Paginated query syntax **658**

Reference: Stability index updates **666**

©2026 Tanium Inc. All Rights Reserved

Gateway overview

Gateway provides a single and stable API integration point for various Tanium solutions. It is designed for Tanium partners and customers interested in building integrated solutions with the Tanium Core Platform.

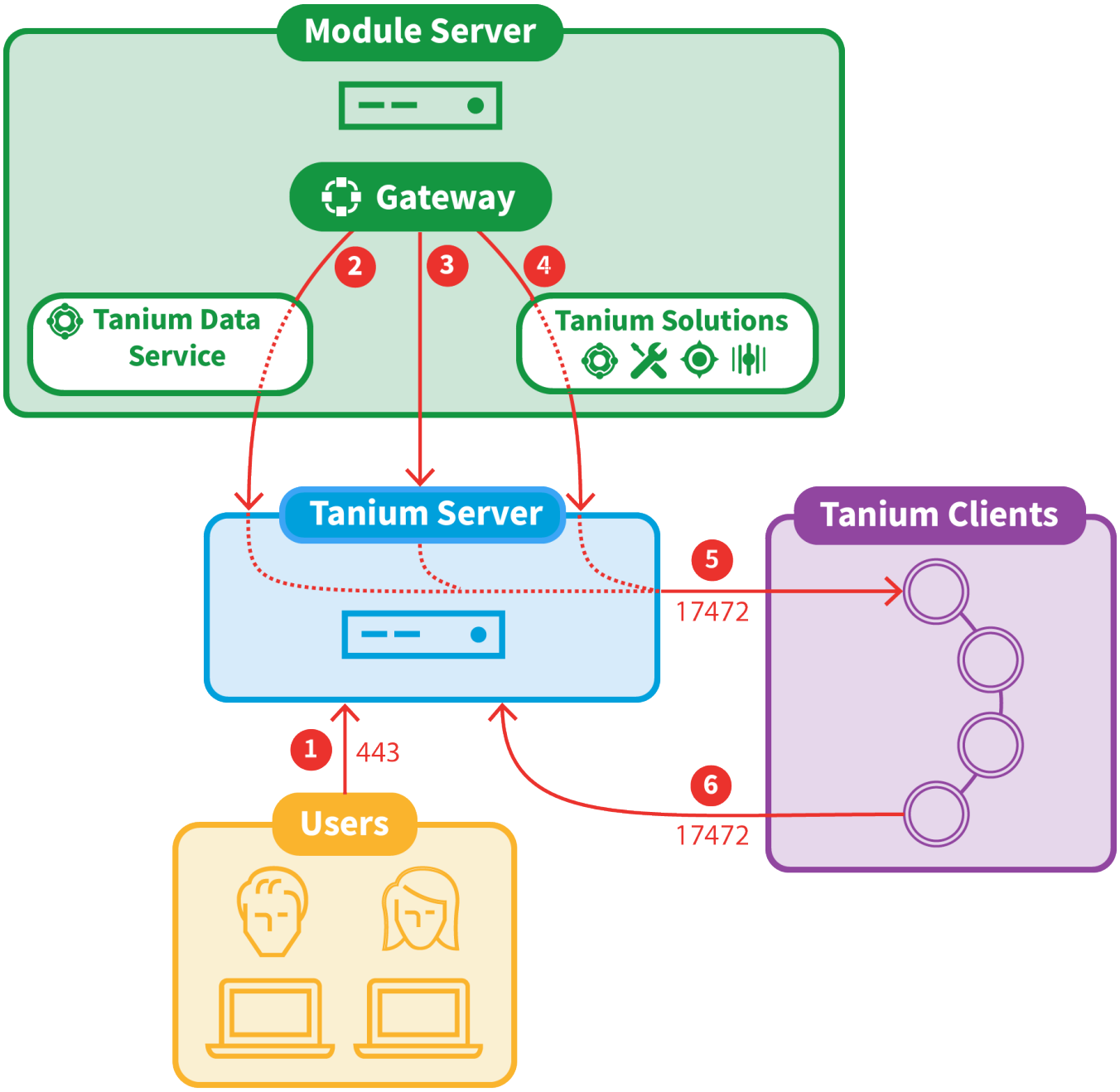
Note: To view examples of filtering syntax, see Reference: Filter syntax on page 562. To view examples of available functions, see Reference: Gateway examples on page 63. To download a file containing examples, see Reference: Integration with Insomnia API Client on page 654.

Gateway topology

The Gateway service runs on the Tanium™ Module Server. The Gateway service passes user requests to Tanium™ Data Service (TDS) or the Tanium™ Server; the Tanium Server collects data from Tanium™ Clients or updates data based on the request, and returns the results to the Gateway service. The Gateway service passes the results to the user that submitted the request.

Figure 1. Gateway topology

©2026 Tanium Inc. All Rights Reserved



With Gateway, information passes between users and the Tanium Server through the following workflow (matching the numbers in Gateway topology on page 15):

1. Users send requests to the Gateway to be performed on endpoints, such as action deployment and queries for question results.

2. By default, the Gateway relays user queries to TDS, which continuously collects results for all registered sensors through the Tanium Server. Continuous collection from endpoints ensures that results are available in the TDS cache for both online and offline endpoints.

©2026 Tanium Inc. All Rights Reserved

3. Users can optionally specify queries that go directly from the Gateway to the Tanium Server for collecting live results from endpoints that are currently online.

4. Users can specify requests that go from the Gateway to Tanium solutions to be performed on endpoints. For example, a user might request Tanium™ Direct Connect to establish a direct connection to a specific endpoint.

5. The Tanium Server issues questions and deploys actions (and associated content such as packages) to Tanium™ Clients.

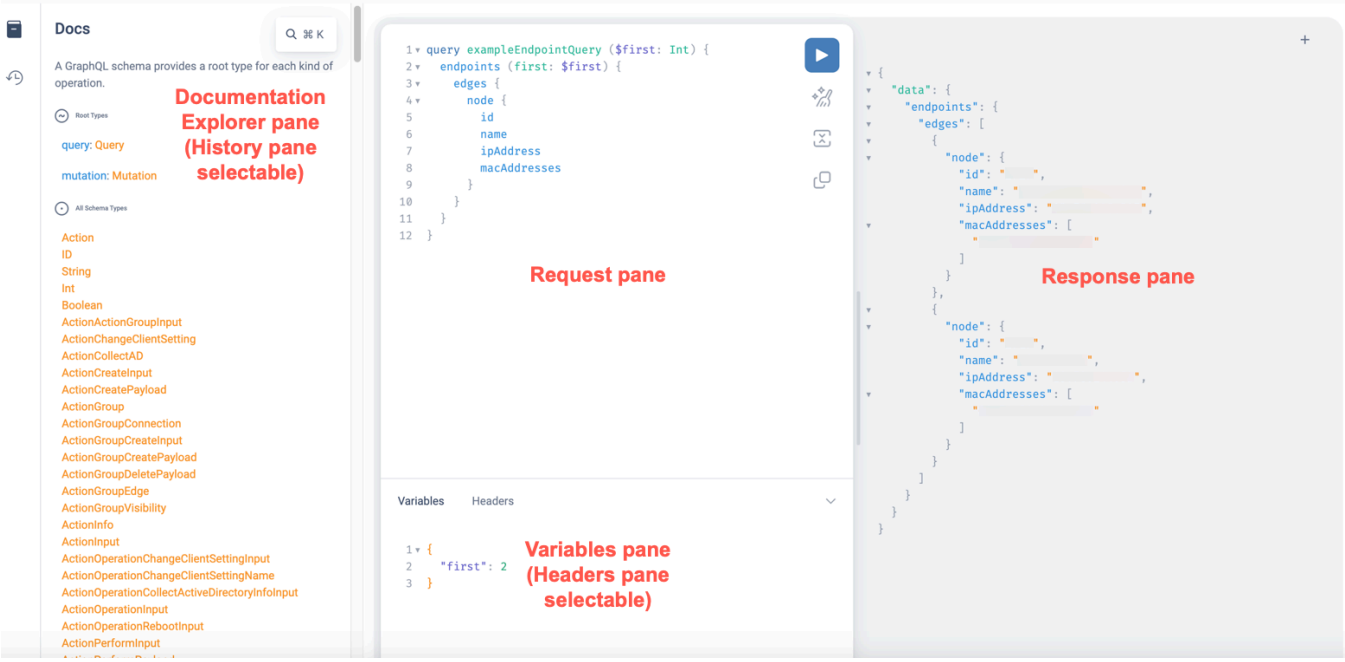
6. Tanium Clients return the question results or action statuses to the Tanium Server. The server responds to the request with the results and statuses. If the user sent the request through the Gateway query explorer, the results pane displays the response.

Query explorer

Gateway includes an interactive query explorer that you can use to write and run queries and mutations in Tanium™ Console. Use the query explorer to try new queries and discover what data is available.

You can find the query explorer on the Gateway **Overview** page:

Figure 2. Query explorer



©2026 Tanium Inc. All Rights Reserved

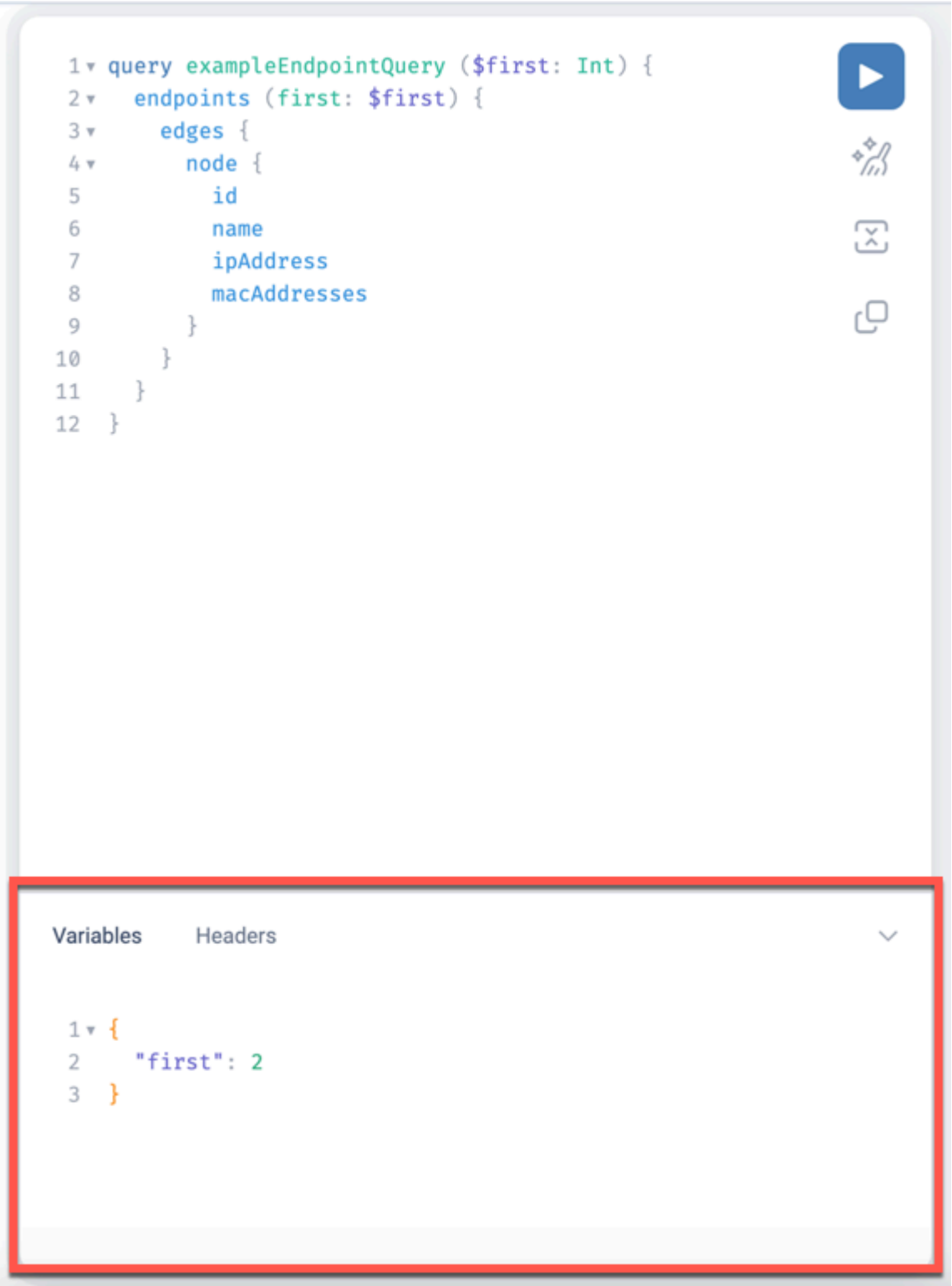
Tip: If the query explorer does not appear on the Gateway **Overview** page, click **Customize Page** and make sure the **Query Explorer** option is selected.

Query variables

If a query or mutation uses variables, expand the **Variables** pane and include the variables in the pane that expands.

Figure 3. Variables pane

©2026 Tanium Inc. All Rights Reserved



For the best results, use query variables when passing arguments, such as when you use pagination or filtering in your request. For more information, see Pagination on page 22 and Filters on page 27.

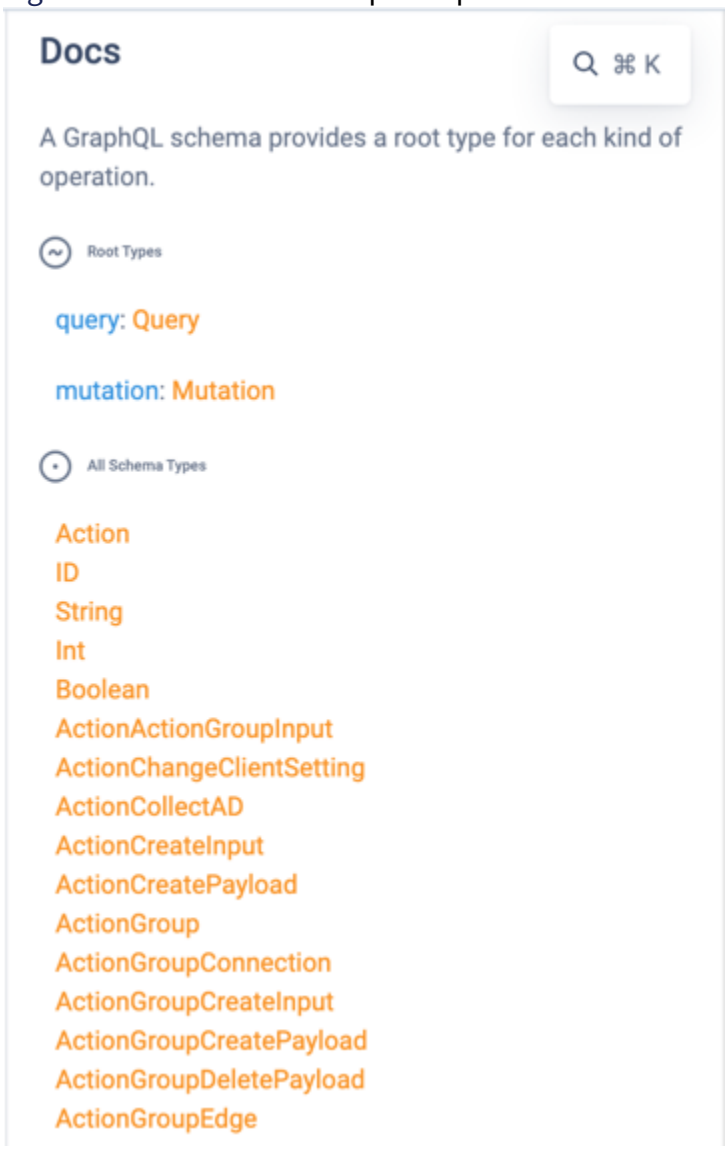
©2026 Tanium Inc. All Rights Reserved

Schema reference

Gateway contains a schema reference that documents all queries, mutations, and objects that are available in Gateway. The schema reference is generated directly from the schema; refer to the schema reference in Gateway for the most up-to-date documentation.

To view the schema reference in the query explorer, click **Docs** to expand the **Documentation Explorer** pane of the query explorer.

Figure 4. Documentation Explorer pane



©2026 Tanium Inc. All Rights Reserved

Note: The query explorer uses the GraphiQL interactive browser to send GraphQL queries and mutations to the Tanium Server. For more information about the options that are available in GraphiQL, see https://graphql.org/learn/.

Authentication

Requests that are sent from the query explorer in Tanium Console are authenticated and authorized with the session ID of the user who is signed in. The Tanium Server uses the role-based access control (RBAC) permissions of the user account to determine which content you can query and mutate.

Requests that are sent from outside Tanium Console are authenticated and authorized with either session IDs or API tokens. You must include an API token or session ID in the authorization header of all requests that are sent to Gateway. Gateway uses the RBAC permissions of the requesting user to determine content access for all queries and mutations. To view an example cURL query that shows the authorization header, see Example cURL syntax on page 22.

Best Practice:

Use API tokens to send requests through Gateway instead of session IDs. While session IDs time out after five minutes of inactivity, you can set a longer timeout for API tokens. You can create API tokens in Tanium Console, through the Tanium Core Platform REST API, or through Gateway.

For more information about additional best practices regarding API tokens, see Tanium Console User Guide: Managing API tokens. For more information about how to create tokens through Gateway, see API token examples on page 174.

Important: When requests are sent outside Tanium Console, make sure to use the correct URL to send requests. To view examples, see Root endpoint on page 22 and Example cURL syntax on page 22.

Rate limits

Gateway has no specific rate limits.

©2026 Tanium Inc. All Rights Reserved

Root endpoint

To send queries and mutations outside Tanium Console, use the following address:

https://<server>/plugin/products/gateway/graphql

Example cURL syntax

```
curl --request POST \
   --url https://localhost/plugin/products/gateway/graphql \
   --header 'Content-Type: application/json' \
   --header 'session: token-356d5f5bbb3671f28e24f65be3bdd54d9d81001ca823efaabc5fbff251' \
   --data '{"query":"{\n  now\n}\n"}'
```

Pagination

Queries that return many results are paginated to reduce resource utilization. Gateway uses the standard Relay GraphQL pagination specification to provide users the option to explicitly control pagination.

Paginated queries return a connection type that is prefixed with the name of the data type, such as `EndpointConnection`. Queries accept standard arguments to control the pagination. For more information about pagination syntax, see Reference: Paginated query syntax on page 658.

Example of a paginated query:

```
query paginationExample {
  endpoints {
    edges {
      node {
        id
        serialNumber
      }
    }
    pageInfo {
      hasNextPage
      endCursor
    }
  }
}
```

Cursors

Use cursors to control relay pagination. Cursors are opaque strings that point to records within queried collections, and can be used to request the records after the cursor. All collections support

©2026 Tanium Inc. All Rights Reserved

forward traversal, and some also support backward traversal.

Cursors are valid only for the query for which they were returned. Cursors are generally valid for five minutes after their most recent use. Any queries that deviate from this policy are documented in the query field. It is a best practice to use query variables with cursors to quickly update the pagination in your request.

Connection results are stable and consistent when traversed with cursors unless documented in the query field.

Example of an endpoint query with cursors:

```
query useCursorsForEndpoints ($cursor: Cursor, $results: Int) {
  endpoints(after: $cursor, first: $results) {
    edges {
      node {
        name
        ipAddress
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the pagination cursor and results variables in the **Variables** panel:

```
{
  "cursor": "NTc2NTM4MDow",
  "results": 5
}
```

Connection and edges

The connection includes an `edge` field that returns a list of typed edges, such as `EndpointEdge`. Each edge contains at least two fields:

- A `node` field with the actual data type, such as `Endpoint`
- A `cursor` field with a cursor for the record

The connection type also includes a `pageInfo` field that contains at least two fields:

- `hasNextPage` indicates if there are more records

©2026 Tanium Inc. All Rights Reserved

- `endCursor` is the cursor of the last record in the returned page, if any

Some connection types feature other metadata, such as `totalRecords`.

Arguments

Paginated queries support at least two arguments: `first` indicates the number of records to return, and `after` is the value of the record cursor that precedes the records in the requested page. When fully paginated, this value is the same as the `endCursor` value from the previous page. Both arguments have sensible defaults.

Paginated queries that support backward traversal allow two corresponding arguments: `last` and `before`. You must pass both arguments together to traverse backwards in a query.



Note: A single query supports either forward or backward traversal, but not both. The server returns an error response for queries with arguments for both forward and backward traversals.

When a paginated request extends beyond the collection, the query returns only the available results.

For the best results, use query variables with the pagination arguments to quickly update the response results.

Example of a request for a page of data within a collection:

```
query getTenEndpointsAfterCursor ($cursor: Cursor, $results: Int) {
  endpoints(after: $cursor, first: $results) {
    edges {
      node {
        id
        serialNumber
      }
    }
    pageInfo {
      hasNextPage
      endCursor
    }
  }
}
```

Include the pagination cursor and results variables in the **Variables** panel:

```
{
  "cursor": "NTc2NTM4MDow",
```

©2026 Tanium Inc. All Rights Reserved

```
  "results": 10
}
```

Refreshing collections

The records returned and paginated over in a query are intended to be stable, ensuring callers are able to traverse the entire collection without gaps or duplicates. With certain collections, particularly the results of querying endpoints with the Tanium Server data source, new records might become available after the first stable paginated collection has been returned by Gateway. There are two primary scenarios for needing these late-arriving data:

- When fetching live data for a specific endpoint or small set of endpoints
- When fetching live data about as many endpoints as practical

When you want to obtain the latest data available for a paginated query, you can request a repopulation of the collection with the `refresh` argument, which takes a cursor as its value. If the gateway can obtain potentially new data for the collection, it obtains the data and returns a new pagination starting at the beginning of the collection. The old cursor values are no longer valid.

Since a common scenario for this capability is when no records were initially available, queries that support the `refresh` argument provide a `collectionInfo` field on their connection type. This field includes a `startCursor` field, which is always defined as pointing to the beginning of the entire collection. The `collectionInfo` might include other fields which help the user decide if they would like to refresh the cursor or read the results.

You can always perform a new query with no cursor instead of using the `refresh` argument, but this does not take advantage of the ongoing effort to collect records from the previous query. Using the `refresh` argument is the most efficient and effective way to obtain as complete a collection as possible or required.

Example of an endpoint query against an endpoint that is slow to respond, with `collectionInfo.startCursor` included in the response:

```
query getSlowEndpoint($first: Int, $path: String, $value: String, $expectedCount: Int) {
  endpoints(
    first: $first
    filter: {path: $path, value: $value}
    source: {ts: {expectedCount: $expectedCount}}
  ) {
    edges {
      node {
        id
      }
```

©2026 Tanium Inc. All Rights Reserved

```
    }
    collectionInfo {
      startCursor
      success
      active
    }
  }
}
```

Add the pagination and filter variables in the **Variables** panel:

```
{
  "first": 1,
  "path": "name",
  "value": "slow-endpoint",
  "expectedCount": 1
}
```

Example response for an ongoing collection with a `startCursor` value:

```
{
  "data": {
    "endpoints": {
      "edges": [],
      "collectionInfo": {
        "startCursor": "4t2cB0ZGNPqy4J4P7O7opwz/24/5zgPi6OinDAA=:0",
        "success": false,
        "active": true
      }
    }
  }
}
```

Example request refreshing the collection using the returned `startCursor` value:

```
query getEndpointRefresh($refresh: Cursor, $first: Int, $path: String, $value: String,
$expectedCount: Int) {
  endpoints(
    refresh: $refresh
    first: $first
    filter: {path: $path, value: $value}
    source: {ts: {expectedCount: $expectedCount}}
  ) {
    edges {
      node {
        id
      }
    }
    collectionInfo {
      startCursor
      success
      active
    }
  }
}
```

Add the pagination and filter variables in the **Variables** panel:

```
{
```

```
  "refresh": "4t2cB0ZGNPqy4J4P7O7opwz/24/5zgPi6OinDAA=:0",
  "first": 1,
  "path": "name",
  "value": "slow-endpoint",
  "expectedCount": 1
}
```

Example response with a completed collection:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "id": 1
          }
      ],
      "collectionInfo": {
        "startCursor": "4t2cB0ZGNPqy4J4P7O7opwz/24/5zgPi6OinDAA=:0",
        "success": true,
        "active": true
      }
    }
  }
}
```

Filters

Most queries that return multiple results provide support to filter the results. Such queries provide a `filter` argument.

It is a best practice to use query variables with filter arguments to quickly update the filter.

For more information about specific filter syntax, see Reference: Filter syntax on page 562.

Simple filters

Simple filters are single filters that constrain the values of fields that participate in the query. You can specify simple filters in the `path` property with a period to separate levels in the graph starting at the record type. For example:

```
query getEndpointsByEmail ($path:String, $value:String){
  endpoints(filter: {path: $path, value: $value}) {
    edges {
      node {
        id
        primaryUser {
          email
        }
      }
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Include the path and value variables in the **Variables** panel:

```
{
  "path": "primaryUser.email",
  "value": "user@example.com"
}
```

The query does not need to return the filtered path. Not all field paths are filterable. Refer to the schema to see which paths cannot be filtered.

Simple filters must also contain a string `value` property.

You can specify an operator in the `op` property, which is an enumerated type and defaults to the equality operator. For example:

```
query getGTE4ProcessorEndpoints($path: String, $value: String, $fieldfilterop:
FieldFilterOp!) {
  endpoints(filter: {path: $path, value: $value, op: $fieldfilterop}) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the path, value, and operator variables in the **Variables** panel:

```
{
  "path": "processor.logicalProcessors",
  "value": "4",
  "fieldfilterop": "GTE"
}
```



Note: Not all operators are valid for all fields.

Compound filters

Compound filters contain multiple simple or compound filters that appear in the `filters` property. By default, all child filters must pass for a record to be included. If the `or` argument is given with a `true` value, then a record is included if any child filter matches.

Example of a simple compound filter:

©2026 Tanium Inc. All Rights Reserved

```
query getEndpointsTwoFilters($path1: String, $path2: String, $value1: String, $value2:
String) {
  endpoints(
    filter: {filters: [{path: $path1, value: $value1}, {path: $path2, value: $value2}]}
  ) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the path and value variables in the **Variables** panel:

```
{
  "path1": "serialNumber",
  "value1": "x",
  "path2": "name",
  "value2": "y"
}
```

Negated filters

You can negate both simple and compound filters with a `negated` property of `true`. For example, the following query returns endpoints whose serial number does not contain the letter `x`:

```
query getEndpointsNegate($path: String, $value: String, $negated: Boolean!) {
  endpoints(
    filter: {filters: {path: $path, value: $value, negated: $negated}}
  ) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the path, value, and negated variables in the **Variables** panel:

```
{
  "path": "serialNumber",
  "value": "x",
  "negated" : true
}
```

Field filters

Filters apply to the entire record. Some records contain fields that are collections; you can also filter these fields. When you filter a field, the filter applies to both the child collection and to the records. For example, if you search for endpoints with an installed application named `Tanium Client` with

©2026 Tanium Inc. All Rights Reserved

a filter on the field, Gateway returns only those endpoints with such an application, as well as only the matching application:

```
query getEndpointsWithTC ($path: String, $value: String){
  endpoints {
    edges {
      node {
        installedApplications(filter: {path: $path, value: $value}) {
          name
          version
        }
      }
    }
  }
}
```

Include the path, value, and negated variables in the **Variables** panel:

```
{
  "path": "name",
  "value": "Tanium Client"
}
```

Field filters can be simple or compound. Compound filters are limited to one level of children that must use the equality operator, and require all child filters. For example:

```
query getEndpointsMultipleFilters ($path1: String, $value1: String, $path2: String, $value2:
String){
  endpoints {
    edges {
      node {
        installedApplications(
          filter: {
            filters: [
              {path: $path1, value: $value1},
              {path: $path2, value: $value2}
            ]
          }
        ) {
          name
          version
        }
      }
    }
  }
}
```

Include the path and value variables in the **Variables** panel:

```
{
  "path1": "name",
  "value1": "Tanium Client",
  "path2": "version",
  "value2": "7.5.0.0"
}
```

©2026 Tanium Inc. All Rights Reserved

Interoperability with other Tanium products

Gateway works with the following solutions:

- Tanium Core Platform
    - Actions
    - Action groups
    - Computer groups
    - Tanium Data Service (TDS)
    - Tanium Direct Connect
    - Packages
    - Tanium™ Reporting
- Tanium™ Asset
- Tanium™ Automate
- Tanium™ Blob Service
- Tanium™ Comply
- Tanium™ Connect
- Tanium™ Deploy
- Tanium™ Discover
- Tanium™ Email
- Tanium™ Impact
- Tanium™ Integrity Monitor
- Tanium™ Patch

©2026 Tanium Inc. All Rights Reserved

- Tanium™ Performance
- Tanium™ Provision
- Tanium™ Reveal
- Tanium™ Benchmark
- Tanium™ Threat Response

©2026 Tanium Inc. All Rights Reserved

Getting started with Gateway

Step 1: Review the requirements

Review the system, network, security, and user role requirements. For more information, see Gateway requirements on page 35.

Step 2: Install Gateway

For more information, see Installing Gateway on page 41.

Step 3: Install any integrated solutions that use the Gateway

Import any integrated solutions that you want to use. For information about which Tanium solutions use Gateway. For more information, see Gateway overview on page 15.

Step 4: Grant Gateway permissions

Grant permissions to users to use Gateway. Uses with the **Administrator** reserved role have access by default. For more information, see User role requirements on page 39.

Note: If you create a user in Tanium Console, you must also create a matching user account in the identity provider that you use for Tanium. For more information, see Tanium Console User Guide: Create a user and Tanium Console User Guide: User authentication.

Step 5: Test queries through the Tanium™ Console

Use the interactive query explorer to test queries through Tanium Console. For more information, see

©2026 Tanium Inc. All Rights Reserved

Test a query in Tanium Console on page 45.

Step 6: (Optional) Test queries through cURL

Test queries through cURL or through your preferred API tool, such as Insomnia. For more information, see Example cURL syntax on page 22 and Reference: Integration with Insomnia API Client on page 654.

Step 7: Explore sample queries and mutations

Explore sample queries and mutations to see what you can do with Gateway. For more information, see Reference: Gateway examples on page 63.

©2026 Tanium Inc. All Rights Reserved

Gateway requirements

Review the requirements before you install and use Gateway.

Core platform dependencies

Make sure that your environment meets the following requirements:

- Tanium license that includes Gateway. For more information about how to get Gateway, see Tanium Solutions & Bundles.
- Tanium™ Core Platform servers
- Tanium Client: No client requirements.
- Tanium Console
- Tanium content: Gateway uses sensors that are included in the Core Content and Core AD Query content packs.

Solution dependencies

Other Tanium solutions are required for Gateway to function (required dependencies) or for specific Gateway features to work (feature-specific dependencies). The installation method that you select determines if the Tanium Server automatically imports dependencies or if you must manually import them.

Note: Some Gateway dependencies have their own dependencies, which you can see in the lists of required and feature-specific dependencies below. Where a dependency links to its own user guide, that link opens the guide for the latest version of the solution, not necessarily the minimum version that Gateway requires.

Tanium recommended installation

©2026 Tanium Inc. All Rights Reserved

If you select **Tanium Recommended Installation** when you import Gateway, the Tanium Server automatically imports all your licensed solutions at the same time. For more information, see Tanium Console User Guide: Import all modules and services.

Import specific solutions

If you select only Gateway to import, the Tanium Server automatically imports the latest available versions of any required dependencies that are missing. If some required dependencies are already imported but their versions are earlier than the minimum required for Gateway, the server automatically updates those dependencies to the latest available versions.

Required dependencies

Gateway has the following required dependencies:

- Tanium™ Data Service (TDS)
- Tanium™ RDB Service

Feature-specific dependencies

If you select only Gateway to import, you must manually import or update its feature-specific dependencies regardless of Tanium Console or Tanium Core Platform versions. Gateway has the following feature-specific dependencies:

- Tanium Asset is required to submit a mutation request to import assets (`assetsImport`), and to submit a request that retrieves Asset products (`query.assetProducts`) or endpoints based on installed Asset products (`query.assetProductEndpoints`).
- Tanium Automate is required to submit a request involving playbook and playbook run functionality.
- Tanium Benchmark is required to submit a request involving Risk/Benchmark functionality.
- Tanium Blob is required to submit a request involving blob storage.
- Tanium Comply is required to submit a request involving the System Vulnerability and System Compliance risk vectors.

©2026 Tanium Inc. All Rights Reserved

- Tanium Comply is required to submit a request that filters Comply-related fields (`complianceFindings` or `cveFindings`).
- Tanium Connect is required to submit a request to retrieve connections, retrieve connection runs, or start or stop connection runs.
- Tanium Direct Connect is required to submit a request involving a direct connection with an endpoint.
- Tanium Deploy is required to submit a request involving Deploy functionality.
- Tanium Discover is required to submit a request involving Discover label or interface functionality.
- Tanium Email Service is required to submit a request retrieving email summaries or to send emails.
- Tanium Impact is required to submit a request involving the Administrative Access risk vector.
- Tanium Integrity Monitor is required to submit a request involving Integrity Monitor watchlists and watchlist paths.
- Tanium Patch is required to submit a request creating, retrieving, or stopping a patch deployment.
- Tanium Performance is required to submit a request involving Performance functionality.
- Tanium Reveal is required to submit a request involving the Password Identification risk vector.
- Tanium™ Reporting is required to submit a request involving Reporting functionality and to submit a request using `query.reportResultData (summarize)` and `query.reportResultData.edges.node.count`.
- Tanium Threat Response is required to submit a request involving Threat Response functionality.

Tanium™ Module Server

©2026 Tanium Inc. All Rights Reserved

Gateway is installed and runs as a service on the Module Server host computer. The impact on the Module Server is minimal and depends on usage.

For information about Module Server sizing in a Windows deployment, see Tanium Core Platform User Guide for Windows Deployments: Reference: Host system resource guidelines.

Endpoints

Gateway does not directly deploy packages to endpoints. However, you can use Gateway to deploy packages through Tanium Deploy.

- For more information about Tanium Deploy endpoint requirements, see Tanium Deploy User Guide: Endpoints.
- For more information about Tanium Client operating system support, see Tanium™ Client Management User Guide: Client version and operating system requirements.

Host and network security requirements

Specific ports and processes are needed to run Gateway.

Ports

The following ports are required for Gateway communication.

Security exclusions

Gateway requires no specific security exclusions.

For more information about security exclusions to define across Tanium, see Reference: Endpoint security exclusions. For Windows-based Tanium Core Platform deployments, see Tanium Core Platform Server security exclusions.

API access

To access the Gateway using a method other than the query explorer in Tanium Console, you must first create an API Token. For more information, see Tanium Console User Guide: Create an API token.

©2026 Tanium Inc. All Rights Reserved

Use the following URL for Gateway access:

URL for Tanium Gateway API access

URL	Notes
<taniumModuleServerURL> /plugin/products/gateway/graphql	The maximum payload size for API requests and responses is 10 MB.

User role requirements

Predefined roles grant the following permissions, which are required to use Gateway functionality:

- solution permissions on page 40
- administration, platform content, and global permissions on page 40: The solution permissions automatically provide some of these permissions.

Some Gateway solution permissions provide permissions that are associated with other solutions. Furthermore, all the platform content permissions and some solution permissions apply to content sets that might contain content for other solutions in addition to Gateway. If you create custom roles for Gateway users, be sure to assign all the solution permissions and content set permissions that are required for Gateway functionality. If you are not sure which permissions are required for a custom role, view the permissions of the predefined roles. For more information, see the following sections of the Tanium Console User Guide:

- RBAC overview (provides an overview of roles, permissions, and content sets)
- Provided role permissions
- View effective role permissions
- Configure a custom role



Note: Do not assign the Gateway **Service Account** and Gateway **Service Account - All Content Sets** roles to users. These roles are for internal purposes only.

©2026 Tanium Inc. All Rights Reserved

Gateway solution permissions

Permission	Gateway User [1]
Gateway Access the Gateway workbench	✓ SHOW
Gateway Api Access the Gateway workbench, use API tokens, and access the Gateway API	✓ EXECUTE
Gateway Service Account Provides access for the Gateway service.	✗

[1] The permissions for this role provide solution permissions for other solutions. To view the provided permissions, see View effective role permissions .

Gateway administration, platform content, and global permissions

Permission	Permission Type	Gateway User
Token - Use	Administration	✓ SPECIAL
Plugin	Platform Content	✓ EXECUTE READ
Sensor	Platform Content	✓ READ

To view the content sets to which the platform content permissions apply, see View effective role permissions .

©2026 Tanium Inc. All Rights Reserved

Installing Gateway

Use Tanium Console **Solutions** page to install Gateway and choose either automatic or manual configuration:

- Automatic configuration: Gateway is installed with any required dependencies and other selected products. This option is the best practice for most deployments. For more information about the automatic configuration for Gateway, see Tanium Console User Guide: Import all modules and services.
- Manual configuration: Manually install Gateway and the required dependencies. For more information, see Import Gateway on page 41.

Before you begin

- Read the .
- Review the Gateway requirements on page 35.
- If you are upgrading from a previous version, see Upgrade Gateway on page 42.
- Review the User role requirements on page 39 and assign the correct roles to Gateway users. For more information, see Tanium Console User Guide: Managing roles.
    - To import the Gateway solution, you must be assigned the Administrator reserved role.

Import Gateway

About this task

Perform the following steps to install the Gateway solution on the Tanium Server.

©2026 Tanium Inc. All Rights Reserved

Note: If you have multiple Tanium Servers in an active-active configuration, you only need to perform these steps on one Tanium Server if you have Tanium Core Platform 7.4.3.1204 or later.

1. Sign in to Tanium Console with an account that has the **Administrator** reserved role.
2. From the Main menu, go to **Administration** > **Configuration** > **Solutions**.
3. In the **Content** section, select the checkbox for **Gateway** and click **Import**.

   Tip: If you need to install any prerequisite Tanium solutions or content, select the corresponding checkboxes for those solutions as well.
4. Review the content to import and click **Begin Install**.

Result

Upgrade Gateway

For the steps to upgrade Gateway, see Tanium Console User Guide: Managing Tanium solutions. To verify that the correct version is installed after the upgrade, see Verify Gateway version on page 42.

Verify Gateway version

After you import or upgrade Gateway, verify that the correct version is installed.

1. Refresh your browser.
2. From the Main menu, go to **Shared Services** > **Gateway**.
3. To display version information, click Info .

Troubleshoot issues

If you experience issues with installing Gateway, see Queries return unexpected results or errors on page 49.

©2026 Tanium Inc. All Rights Reserved

Using Gateway

Use Gateway to build API-based integrations with the Tanium Core Platform. This service consolidates information from multiple Tanium modules into a unified view of information on the endpoints in the environment. Gateway intelligently routes requests to the services and sources that provide the most recent information and the most reliable mutations.

Note: To view examples of filtering syntax, see Reference: Filter syntax on page 562. To view examples of available functions, see Reference: Gateway examples on page 63. To download a file containing examples, see Reference: Integration with Insomnia API Client on page 654.

Gateway uses GraphQL to request data (queries) and to make changes (mutations). With GraphQL, you can compose queries in Gateway to retrieve the exact data that you want as well as filter the results to a set of endpoints.

Important: Requests scheduled to run during an upgrade of Gateway, or any Tanium solution that your request relies on, might be interrupted or fail. Schedule your connections to run at a different time than your Tanium upgrades.

Use Gateway to:

- Query endpoints through the Tanium Server, or access data through Tanium Data Service (TDS)
- Create, delete, and query actions
- Query packages
- Open a connection to an endpoint through Tanium Direct Connect and retrieve data from the endpoint

To view examples of available functions, see Reference: Gateway examples on page 63.

©2026 Tanium Inc. All Rights Reserved

Create a persona limited to submitting only query requests

About this task

If you want to restrict a user's ability to modify information using Gateway, configure a role with read-only rights to certain Tanium Core Platform functionality, then create a persona and add the role to the persona. If a user with this persona active submits a mutation, the response contains an error because the persona does not have write access.

> **Note:** The following role and persona limit Tanium Core Platform permissions for questions and actions. If you want to restrict other functionality, such as that associated with other Tanium solutions, you might need to add additional read-only permissions.

To create a custom role, see Tanium Console User Guide: Configure a custom role. To create a persona, see Tanium Console User Guide: Create a persona.

1. From the Main menu, go to **Administration** > **Permissions** > **Roles** and click **New Role**.
2. Specify a **Role Name**, such as `Gateway read-only role`.
3. Set the **Permission Type** to **Allow**.
4. Select the **Gateway API** execute permission to allow the role to submit requests.
5. Select the following permissions to restrict question access to read-only:
   - Interact - **Ask Dynamic Questions special** special
   - Platform Content Permissions - **Filter Group read**
   - Platform Content Permissions - **Sensor read**

   Select the following permissions to restrict action access to read-only:
   - Platform Content Permissions - **Action read**
   - Platform Content Permissions - **Filter Group read**
   - Platform Content Permissions - **Package read**
   - Platform Content Permissions - **Sensor read**
6. Click **Save**.
7. From the Main menu, go to **Administration** > **Permissions** > **Personas** and click **New Persona**.

©2026 Tanium Inc. All Rights Reserved

8. Specify a persona **Name**, such as `Gateway read-only persona`.
9. Click **Manage Roles**.
10. Select **Gateway read-only role**.
11. Click **Apply**.
12. Configure the user group, user, and computer group assignments.
13. Click **Save**.

Test a query in Tanium Console

About this task

To access the query explorer in Tanium Console and run a query, perform the following steps:

1. From the Main menu, go to **Shared Services** > **Gateway**.
2. Enter a query in the query pane.
   For example, paste the following query to get the time from the Tanium Server:

   ```
   {
     now
   }
   ```

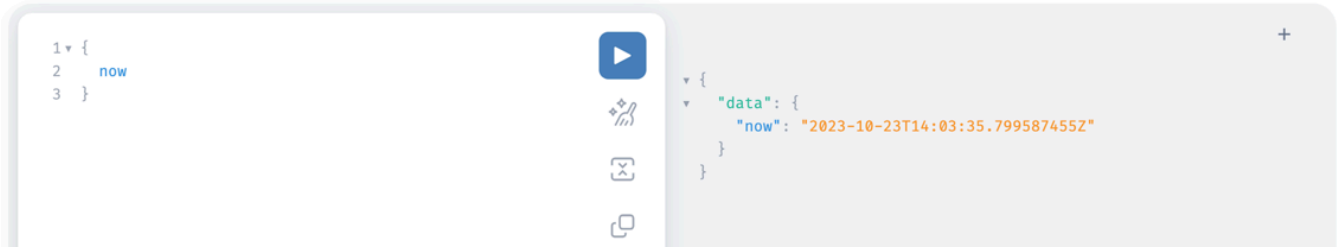
   Tip: If the query explorer does not appear on the Gateway **Overview** page, click **Customize Page** and make sure the **Query Explorer** option is selected.
3. (Optional) If a query or mutation uses variables, expand the **QUERY VARIABLES** pane and include the variables in the pane that expands.
4. Click Execute Query .

Result

Gateway sends the query to the server and returns the response in the results pane.

Figure 5. Example query and response

©2026 Tanium Inc. All Rights Reserved



For more information on the query explorer, see Query explorer on page 17.

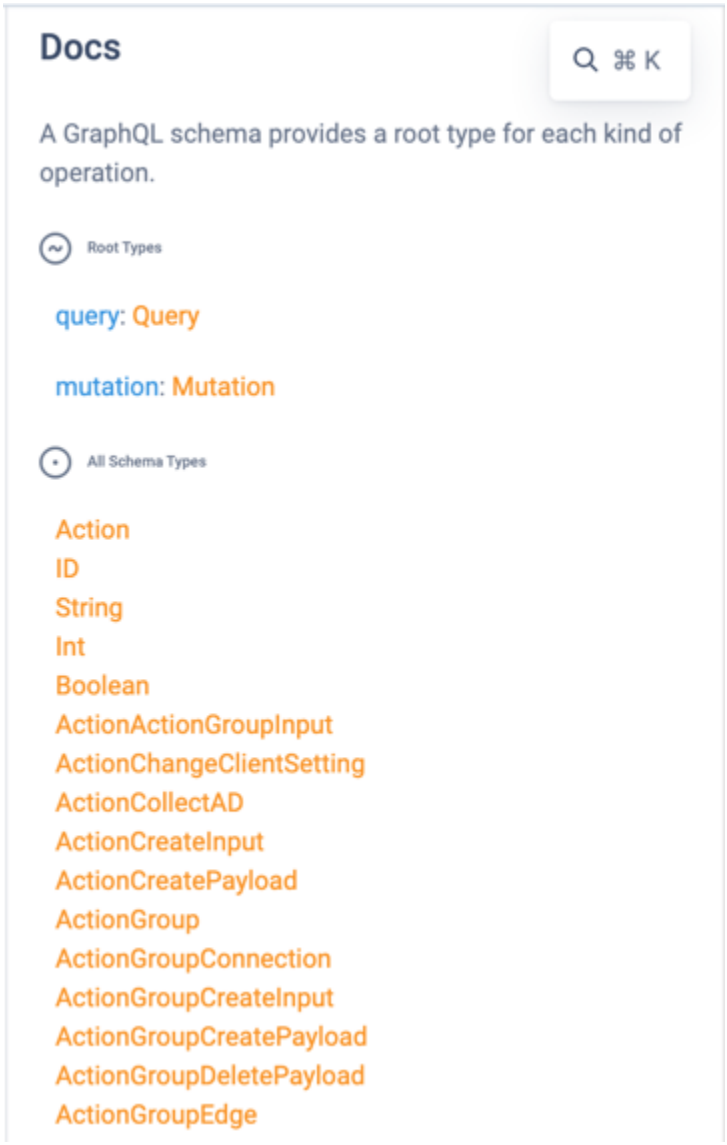
Use the Documentation Explorer to view the schema

Gateway contains a schema reference that documents all queries, mutations, and objects that are available in Gateway. The schema reference is generated directly from the schema; refer to the schema reference in Gateway for the most up-to-date documentation.

To view the schema reference in the query explorer, click **Docs** to expand the **Documentation Explorer** pane of the query explorer.

Figure 6. Documentation Explorer pane

©2026 Tanium Inc. All Rights Reserved



©2026 Tanium Inc. All Rights Reserved

Maintaining Gateway

Perform monthly maintenance tasks to ensure that Gateway successfully performs scheduled activities on all the targeted endpoints and does not overuse endpoint or network resources. If Gateway is not performing as expected, you might need to troubleshoot issues or change settings. For more information about related procedures, see Troubleshooting Gateway on page 49.

Gateway monthly maintenance

1. Verify that API tokens have not expired. For more information, see Tanium Console User Guide: View API token details.
2. Verify whether any API tokens are due for rotation based on the policy of your organization. For more information, see Tanium Console User Guide: Rotate an API token.
3. Revoke any API tokens that are no longer needed. For more information, see Tanium Console User Guide: Revoke API tokens.
4. Check for deprecated fields and, if necessary, update the integration scripts that use them. For more information, see Tanium™ Gateway User Guide: Reference: Deprecated fields.
5. To investigate and remediate Gateway issues, see Tanium Gateway User Guide: Troubleshooting Gateway.

©2026 Tanium Inc. All Rights Reserved

Troubleshooting Gateway

If Gateway is not performing as expected, you might need to troubleshoot issues.

Collect logs

About this task

The information is saved in a ZIP file that you can download with your browser. The ZIP file is also known as a support bundle.

1. From the **Gateway Overview** page, click Help ?, then the **Troubleshooting** tab.
2. Click **Download Support Package**. A tanium-api-gateway-support-[timestamp].zip file downloads to the local download directory.
3. Contact Tanium Support to determine the best option to send the ZIP file. For more information, see Contact Tanium Support on page 62.

Queries return unexpected results or errors

- The Gateway service redirects queries and mutations to other Tanium solutions. If Gateway returns unexpected results or errors, make sure that all prerequisites are installed at the minimum recommended version. For information, see Gateway requirements on page 35.
- If all queries and mutations return a **502 Gateway Timeout** error, make sure the Tanium System User Service and the Tanium Gateway service are running on the Tanium Module Server.
- If you recently installed Gateway or the System User Service, restart the Tanium Module Server.
- Queries and mutations that use the **eid** element require Interact 2.9 or later.
- If you recently added or changed parameters on an existing sensor in Console, allow a few minutes for the Gateway service to pick up the change through its periodic sensor sync. A restart is not required.

©2026 Tanium Inc. All Rights Reserved

Request error handling

When requests cannot be fully satisfied, the Gateway service gives the caller accurate, practical error information. The Gateway service honors the GraphQL error specification with some custom extensions for greater specificity.

Syntax Validation

Before processing any portion of a request, the service parses its syntax. If the request is not valid GraphQL, or if it does not conform to the schema, the service returns a response with the syntax errors, including line and column positions whenever possible.

Examples of syntax errors include:

- Unbalanced parentheses, braces, brackets, and quotation marks
- References to fields that do not exist

The GraphQL error specification requires reporting these errors to the caller by line and column in the request.



Note: When possible, the Query pane in the Query Explorer identifies syntax errors in the request.

Structural Validation

If the request passes syntax validation, the service evaluates the request for structural validity, which checks for errors not expressible with the GraphQL type system. Structural error examples include:

- Restrictions on type values, such as:
    - A string may not be empty
    - An integer must be positive
- Constraints on input objects, such as:

©2026 Tanium Inc. All Rights Reserved

        - Exactly one of two or more entries must be present
    - Existence of values in a known set, such as:
        - A named sensor must exist
    - References to fields that cannot be satisfied in the current environment, such as:
        - A sensor does not exist

The GraphQL `extensions` error field for custom fields contains these errors.

Error Extensions

Errors that pertain to a specific field in the request are identified by a `fieldPath` entry in the `extensions` object. Similar to the `path` entry on the GraphQL error type, the `fieldPath` consists of a list of strings and integers which indicate keys in objects and offsets in lists, respectively. The root of the `fieldPath` is the request field that triggered the GraphQL error.



Note: The numbering for list items starts at `0` for the first item and increments for each additional item. For example, if a `fieldPath` contains `5`, this refers to the sixth item in the list.

The `extensions` object may also have these fields:

Fields contained by the extensions object

Extensions field	Description
`code`	an error code attributable to the field itself
`context`	a map of contextual information that refines or specifies the error
`message`	a short error explanation

For example, with the following syntactically valid request:

```
{
  endpoints {
    edges {
      node {
        name
```

©2026 Tanium Inc. All Rights Reserved

```
        serialNumber
      }
    }
  }
}
```

If the serial number is not available in the underlying data store, the Gateway service returns a response including an extended error that the value does not exist:

```
{
  "errors": [
    {
      "message": "Invalid request",
      "path": ["endpoints"],
      "extensions": {
        "fieldPath": ["edges", "node", "serialNumber"],
        "code": "validation_exists",
        "message": "must refer to an existing sensor",
        "context": {
          "source": "tds"
        }
      }
    }
  ]
}
```

Invalid argument errors contain an extended `argumentErrors` entry. The value of that entry is a list of objects that may have the following entries:

Possible values for the object list in the argumentErrors entry

Extensions field	Description
`code`	an error code attributable to the field itself
`context`	a map of contextual information that refines or specifies the error
`path`	the path to the argument error, relative to the field
`message`	a short error explanation

For example, with the following request that does not define a `name` or `path` value:

```
{
  endpoints(filter: {filters: [{memberOf: {name: ""}}, {path:""}]}) {
    edges {
      node {
        name
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

The Gateway service returns a response including both errors stating a value is a required:

```
{
  "errors": [
    {
      "message": "Invalid request",
      "path": ["endpoints"],
      "extensions": {
        "argumentErrors": [
          {
            "code": "validation_required",
            "path": ["filter", "filters", 0, "memberOf", "name"]
          },
          {
            "code": "validation_required",
            "path": ["filter", "filters", 1, "path"]
          }
        ]
      }
    }
  ]
}
```

Extended errors may identify argument errors within sub-fields. For example, the following request specifies a non-existent sensor name:

```
{
  endpoints {
    edges {
      node {
        sensorReadings(sensors:[{name: "A Non-Existent Sensor"}]) {
          columns {
            values
          }
        }
      }
    }
  }
}
```

The Gateway service returns a response including the sub-field error that the value does not exist:

```
{
  "errors": [
    {
      "message": "Invalid request",
      "path": ["endpoints"],
      "extensions": {
        "fieldPath": ["edges", "node", "sensorReadings"],
        "argumentErrors": [
          {
            "code": "validation_exists",
            "path": ["sensors", 0, "name"]
          }
        ]
      }
    }
  ]
}
```

©2026 Tanium Inc. All Rights Reserved

Whenever possible, the response includes the valid portions of a query. For example, the following request references a valid sensor (Custom Tags) and a not valid sensor (A Non-existent Sensor):

```
{
  endpoints {
    edges {
      node {
        name
        ipAddress
        os {
          generation
        }
        sensorReadings(sensors: [{name: "Custom Tags"}, {name: "A Non-existent Sensor"}]) {
          columns {
            sensor {
              name
            }
            name
            values
          }
        }
      }
    }
  }
}
```

The response includes the error from A Non-Existent Sensor not being found, and all information that can otherwise be returned from the Custom Tags sensor:

```
{
  "errors": [
    {
      "message": "sensors: (1: (name: must refer to an existing sensor.).).",
      "path": [
        "endpoints"
      ],
      "extensions": {
        "argumentErrors": [
          {
            "code": "validation_exists",
            "context": {
              "source": "tds"
            },
            "message": "must refer to an existing sensor",
            "path": [
              "sensors",
              1,
              "name"
            ]
          }
        ],
        "fieldPath": [
          "edges",
          "node",
          "sensorReadings"
        ]
      }
    }
  ],
  "data": {
```

©2026 Tanium Inc. All Rights Reserved

```
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host",
            "ipAddress": "192.0.2.10",
            "os": {
              "generation": "Windows 10"
            },
            "sensorReadings": {
              "columns": [
                {
                  "sensor": {
                    "name": "Custom Tags"
                  },
                  "name": "Custom Tags",
                  "values": [
                    "custom-tag-1",
                    "custom-tag-2"
                  ]
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```


Note: Similar to lists in the `fieldPath` referring to offsets in lists, if an error occurs on an item in a list, the error `message` might refer to the list offset instead of the field name. In the above response, the `message` refers to item `1` in the list; because list numbering starts at `0`, this points to the second sensor (A Non-existent Sensor).

Error codes

General error codes include:

General error codes

Error code	Description
`401 Unauthorized`	If using an API token, you do not have permission to use the token for authentication, or the token is not valid or does not exist.
`403 Forbidden`	You do not have the permissions required for your request. See the `message` for more information.

©2026 Tanium Inc. All Rights Reserved

Error code	Description
`500 Internal Server Error`	The response field cannot be resolved due to a server error.
`502 Bad Gateway`	The Gateway service received an invalid response from the server.
`503 Service Unavailable`	The Gateway service is running, but cannot handle a request.
`COMPLEXITY_LIMIT_EXCEEDED`	The request contains too many fields, or the request would return too many results, based on field complexity weight and query maximum complexity.
`incompatible_solution`	A required Tanium solution is not installed at the correct version.
`insufficient_permissions`	The user account making the request does not have the proper permissions to access a resource.

Syntax error codes include:

Syntax error codes

Error code	Description
`GRAPHQL_PARSE_FAILED`	The request contains unbalanced parentheses, braces, brackets, or quotation marks. See the `message` for more information.
`GRAPHQL_VALIDATION_FAILED`	The request contains an unrecognized argument, a required argument is not included, or the maximum alias or depth request limit is exceeded. See the `message` for more information.
`INVALID_GRAPHQL`	The request contains an unknown operation. See the `message` for more information.

Structural error codes include:

Structural error codes

Error code	Description
`validation_allowed`	The field is not allowed in this position.
`validation_data_limit`	The request pagination settings result in a response size greater

©2026 Tanium Inc. All Rights Reserved

Error code	Description
	than the maximum response size allowed.
`validation_date`	The field must contain a valid RFC 3339 long date format (`YYYY-MM-DD`).
`validation_deadline_exceeded`	The system did not respond within an amount of time, dependent on the specific request.
`validation_exists`	The field must refer to an existing entity.
`validation_in_invalid`	The field must have a value from a specific set.
`validation_invalid_action_perform`	The action operation must specify only one operation, specify `package` or `operation` , or a tag value contains whitespace characters or a number sign (`#`).
`validation_invalid_dec_process_name`	The request contains a process name that does not match the process ID.
`validation_invalid_field_filter`	The field filter is not valid, or is not allowed for filtering on sensors with multi-character delimiters.
`validation_invalid_for_source`	The source does not allow the request.
`validation_invalid_group`	The request must provide `name` , `id` , or `filter` for the group.
`validation_invalid_ref`	The reference must include either a name or ID.
`validation_invalid_use`	The filter is not valid in its context.
`validation_integration_name`	The integration name must contain a value that does not start with `tanium-` .
`validation_ipaddress`	The field must contain an IPv4 address or IPv4 CIDR range.
`validation_key_unexpected`	The field is not allowed.
`validation_min_greater_equal_than_required`	The field must have a value greater than or equal to its

©2026 Tanium Inc. All Rights Reserved

Error code	Description
	documented minimum.
`validation_nil`	The field must be absent.
`validation_nil_or_not_empty_required`	The field must be absent or have a non-empty value.
`validation_not_nil_required`	The field must have a value.
`validation_overdetermined`	The field's object must not have inconsistent entries.
`validation_param_missing`	The request must define all required parameter values for referenced sensors.
`validation_param_unexpected`	The request must define valid parameter values for referenced sensors.
`validation_persona_name`	The persona must be a persona assigned to the account that submitted the request.
`validation_positive_integer`	The field must contain a positive integer.
`validation_required`	The field must have a non-empty value.
`validation_timestamp`	The field's value must be a valid RFC 3339 string.
`validation_unique`	The field's value must be unique in its documented container.
`validation_unregistered`	The sensor must be registered with Tanium Data Service, or the sensor parameter values must be an exact match for a parameterized sensor instance registered with Tanium Data Service.

Issue: error message for enumerated value displays the value in the request

For enumerated type fields in requests, if you pass the enumerated value in quotation marks, the response error message appears to state that you must pass that value. For example, with the

©2026 Tanium Inc. All Rights Reserved

following request, `any` (a boolean field) is set to `"false"` and `visibility` (an enumerated type) is set to `"ALL"` (in quotation marks):

```
mutation createActionGroup {
  actionGroupCreate(
    input: {any: "false", computerGroups: [{name: "Example Computer Group"}], name: "Example
Action Group", visibility: "ALL"}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

The response constrains validation error messages similar to the following because the validation engine parses `"false"` and `"ALL"` as non-enumerated values, and requires that they be passed in the request without quotation marks.

```
{
  "errors": [
    {
      "message": "Boolean cannot represent a non boolean value: \"false\"",
      "extensions": {
        "code": "GRAPHQL_VALIDATION_FAILED"
      }
    },
    {
      "message": "Enum \"ActionGroupVisibility\" cannot represent non-enum value: \"ALL\".
Did you mean the enum value \"ALL\"?",
      "extensions": {
        "code": "GRAPHQL_VALIDATION_FAILED"
      }
    }
  ]
}
```

Do one of the following:

- Pass the value as a JSON variable definition.
- Pass the value in your request without quotation marks.
- If using the Query Explorer, enter an enumerated type field and a colon, press Shift + Space when your cursor is after the colon, then select an enumerated value.

Issue: ResourceExhausted error in response to valid query request

©2026 Tanium Inc. All Rights Reserved

Query requests that retrieve a large amount of information in the response might return a `ResourceExhausted` error, due to the size of the response in bytes being too large to return. For example, the following request specifies returning the first `99999` results, and requests multiple fields with the potential to return many results per field, increasing the overall size of the response:

```
query endpointsLargePagination {
  endpoints(first: 99999) {
    edges {
      node {
        id
        eidFirstSeen
        eidLastSeen
        namespace
        computerID
        systemUUID
        name
        domainName
        serialNumber
        manufacturer
        model
        ipAddress
        macAddresses
        isVirtual
        isEncrypted
        chassisType
        os {
          name
          language
        }
        services {
          status
          startupMode
        }
        installedApplications {
          uninstallable
          name
          version
          silentUninstallString
        }
        deployedSoftwarePackages {
          id
          name
          applicability
          gallery
        }
        compliance {
          cveFindings {
            cveYear
            summary
            severity
            cveId
            cvssScore
          }
          complianceFindings {
            standard
            standardVersion
            state
            id
            ruleId
          }
```

©2026 Tanium Inc. All Rights Reserved

```
        }
      }
    }
  }
}
```

If this response size in bytes is too large, the response might include a `ResourceExhausted` error code:

```
{
  "errors": [
    {
      "message": "Error fetching endpoints: response error: rpc error: code =
ResourceExhausted desc = grpc: received message larger than max (28156311 vs. 4194304)",
      "extensions": {
        "serviceName": "graphql-platform",
        "code": "DOWNSTREAM_SERVICE_ERROR"
      }
    }
  ],
  "data": {
    "endpoints": null
  }
}
```

Reduce the response size. You have the following options:

- Reduce the number of results returned per page.
- Reduce or otherwise constrain the information you request, especially if a given field might return an array for each result with multiple result values.



Note: For the best results, due to the variability in size of Tanium deployments, retrieve 100 records per page for an initial test of your paginated query, and adjust the number of records returned per page based on performance and response time.

Issue: repeated 503 error code in responses

The Gateway service imposes maximums on allocated memory and concurrent response size to optimize overall performance and resource use. If you submit too many simultaneous requests to the Gateway service and exceed either maximum, you might receive 503 errors in response to your requests.

Wait 60 seconds for the Gateway service to process queued requests, then try again. If this message persists, Contact Tanium Support on page 62.

©2026 Tanium Inc. All Rights Reserved

Uninstall Gateway

If you need to uninstall Gateway, perform the following steps.

Important: Consult with Tanium Support before you uninstall or reinstall Gateway.

1. Sign in to Tanium Console as a user with the Administrator role.
2. From the Main menu, go to **Administration** > **Configuration** > **Solutions**.
3. In the **Content** section, select the Gateway row and click **Uninstall**.
4. Review the summary and click **Yes** to proceed with the uninstallation.
5. When prompted to confirm, enter your password.

Note: The uninstall does not remove the Gateway log from the Tanium Module Server.

- To remove the log in a TanOS environment, manually delete the /opt/Tanium/TaniumModuleServer/services/gateway-files/ directory.
- To remove the log in a Windows environment, manually delete the \Program Files\Tanium\Tanium Module Server\services\gateway-files\ directory.

Contact Tanium Support

To contact Tanium Support for help, sign in to the Tanium Customer Community.

For more information about submitting a support case, see Case Submission Form.

©2026 Tanium Inc. All Rights Reserved

Reference: Gateway examples

Use the following request examples to learn about the functionality and syntax of queries and mutations in Gateway. For information about integrating with the Insomnia API client, downloading request examples in Insomnia JSON format, and generating code examples from the request examples using Insomnia, see Reference: Integration with Insomnia API Client on page 654.

Important: Requests scheduled to run during an upgrade of Gateway, or any Tanium solution that your request relies on, might be interrupted or fail. Schedule your connections to run at a different time than your Tanium upgrades.

System examples

The following requests retrieve data about your Tanium instance.

Get server time

(query.now)

The following query retrieves the local time from the Tanium Server.

```
query exampleGetServerTime {
  now
}
```

Example response:

```
{
  "data": {
    "now": "2021-11-08T19:22:03Z"
  }
}
```

Get current user information

(query.currentUserContext)

About this task

©2026 Tanium Inc. All Rights Reserved

The following query retrieves information about the user submitting the request, including the selected persona and the personas available to that user. If the response contains a `null` value for the persona ID, the current user has the default persona selected.

```
query exampleGetCurrentUser {
  currentUserContext {
    user {
      id
      username
      displayName
    }
    persona {
      id
      name
    }
    availablePersonas {
      id
      name
    }
  }
}
```

Example response:

```
{
  "data": {
    "currentUserContext": {
      "user": {
        "id": "123",
        "username": "example_user",
        "displayName": "Example user"
      },
      "persona": {
        "id": "234",
        "name": "example_persona"
      },
      "availablePersonas": [
        {
          "id": "234",
          "name": "example_persona"
        },
        {
          "id": "345",
          "name": "example_persona_2"
        }
      ]
    }
  }
}
```

General examples

The following queries retrieve data from the endpoints in your environment.

©2026 Tanium Inc. All Rights Reserved

Get endpoints

(query.endpoints(source))

The following query retrieves known endpoints from the Tanium Server.

```
query exampleGetEndpoints($count: Int, $time: Int) {
  endpoints(source: {ts: {expectedCount: $count, stableWaitTime: $time}}) {
    edges {
      node {
        computerID
        name
        serialNumber
        ipAddress
      }
    }
  }
}
```

Include the count and time variables in the **Variables** panel or in your variables dictionary:

```
{
  "count": 1,
  "time": 10
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "937672696",
            "name": "ubuntu-test",
            "serialNumber": "Not Specified",
            "ipAddress": "10.168.20.30"
          }
        },
        {
          "node": {
            "computerID": "1867570226",
            "name": "CentOS-test-1",
            "serialNumber": "Not Specified",
            "ipAddress": "10.168.20.40"
          }
        },
        {
          "node": {
            "computerID": "2711217959",
            "name": "CentOS-test-2",
            "serialNumber": "Not Specified",
            "ipAddress": "10.168.20.50"
          }
        }
      ]
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Get endpoints and update sensor maximum age

(query.endpoints (source.ts.maxAge))

The following query retrieves endpoints from the Tanium Server and returns results from the Installed Applications and Running Applications sensors. Because `maxAge` is defined for the Tanium Server source, it applies to all sensors (Installed Applications, Running Applications) and sensor-related fields (`computerID`, `name`, `serialNumber`, and `ipAddress`) in the request. If any cached sensor data is older than 300 seconds, the relevant sensor reevaluates to retrieve fresh data.

Note: You can only use `maxAge` with the Tanium Server (`ts`) data source. The response returns a validation error if you try to use `maxAge` with the Tanium Data Service (`tds`) data source.

Note: Setting this value to very small periods of time might cause excessive resource usage on endpoints and frequent sensor evaluation.

```
query exampleGetEndpoints($count: Int, $time: Int, $maxAge: Int) {
  endpoints(source: {ts: {expectedCount: $count, stableWaitTime: $time, maxAge: $maxAge}}) {
    edges {
      node {
        computerID
        name
        serialNumber
        ipAddress
        sensorReadings(sensors: [{name: "Installed Applications"}, {name: "Running
Applications"}]) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

Include the count, time, and maximum age variables in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "count": 1,
  "time": 10,
  "maxAge": 300
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "0123456789",
            "name": "example_host",
            "serialNumber": "1234567890",
            "ipAddress": "192.168.0.10",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "example_application",
                    "example_application_2"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.2.3",
                    "2.3.4"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Name",
                  "values": [
                    "N/A on Linux"
                  ],
                  "sensor": {
                    "name": "Running Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    ""
                  ],
                  "sensor": {
                    "name": "Running Applications"
                  }
                }
              ]
            }
          }
        },
        {
```

```
          "node": {
            "computerID": "2345678901",
            "name": "example_host_2",
            "serialNumber": "5678901234",
            "ipAddress": "192.168.0.20",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "Mozilla Firefox (AArch64 en-US)",
                    "Notepad++ (64-bit x64)"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "121.0",
                    "8.5.1"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Name",
                  "values": [
                    "Microsoft Teams",
                    "Microsoft Edge"
                  ],
                  "sensor": {
                    "name": "Running Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "24004.1403.2634.2418",
                    "121.0.2277.128"
                  ],
                  "sensor": {
                    "name": "Running Applications"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```


Get endpoints IDs from Tanium Data Service

(query.endpoints)

©2026 Tanium Inc. All Rights Reserved

The following query retrieves all endpoint IDs from Tanium Data Service.

```
query exampleGetTDSEndpointIDs {
  endpoints {
    edges {
      node {
        id
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "id": "12345"
          }
        },
        {
          "node": {
            "id": "54321"
          }
        },
        {
          "node": {
            "id": "21212"
          }
        }
      ]
    }
  }
}
```

Get endpoints with filter excluding errors

(query.endpoints (source.ts.excludeErrors | source.tds.excludeErrors))

Retrieved endpoints might contain error messages as field values. You can exclude endpoints retrieved from either the `ts` or `tds` data source from the response if they contain common error messages within returned fields. The following query retrieves the first 2 endpoints from Tanium Data Service, and does not exclude endpoints with common error values in returned fields.

```
query exampleExcludeEndpointsErrors($first: Int, $excludeErrors: Boolean!) {
  endpoints(first: $first, source: {tds: {excludeErrors: $excludeErrors}}) {
    edges {
      node {
        name
        ipAddress
        manufacturer
        model
      }
```

©2026 Tanium Inc. All Rights Reserved

```
    }
  }
}
```

Include the pagination and `excludeErrors` variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "excludeErrors": false
}
```

Example response containing an endpoint with a common error value in the `manufacturer` and `model` fields:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host-1",
            "ipAddress": "192.0.2.10",
            "manufacturer": "VMware, Inc.",
            "model": "VMware Virtual Platform"
          }
        },
        {
          "node": {
            "name": "example-host-2",
            "ipAddress": "192.0.2.20",
            "manufacturer": "TSE-Error: Unknown - dmidecode unavailable",
            "model": "TSE-Error: Unknown - dmidecode unavailable"
          }
        }
      ]
    }
  }
}
```

If you update the `excludeErrors` variable definition to `true`, the query response excludes endpoints with common error values in the returned fields:

```
{
  "first": 2,
  "excludeErrors": true
}
```

Example response containing no endpoints with a common error value in the `manufacturer` and `model` fields:

```
{
  "data": {
    "endpoints": {
      "edges": [
```

©2026 Tanium Inc. All Rights Reserved

```
        {
          "node": {
            "name": "example-host-1",
            "ipAddress": "192.0.2.10",
            "manufacturer": "VMware, Inc.",
            "model": "VMware Virtual Platform"
          }
        },
        {
          "node": {
            "name": "example-host-3",
            "ipAddress": "192.0.2.30",
            "manufacturer": "Apple",
            "model": "MacPro6,1"
          }
        }
      ]
    }
  }
}
```

Get endpoints with filter excluding [no results]

(query.endpoints (source.ts.excludeNoResults | source.tds.excludeNoResults))

Retrieved endpoints might contain `[no results]` as field values. You can exclude endpoints retrieved from either the `ts` or `tds` data source from the response if they contain `[no results]` in returned fields. The following query retrieves the first 2 endpoints from Tanium Data Service, and does not exclude endpoints with `[no results]` in returned fields.

```
query exampleExcludeEndpointsNoResults($first: Int, $excludeNoResults: Boolean!) {
  endpoints(first: $first, source: {tds: {excludeNoResults: $excludeNoResults}}) {
    edges {
      node {
        name
        ipAddress
        macAddresses
      }
    }
  }
}
```

Include the pagination and `excludeErrors` variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "excludeNoResults": false
}
```

Example response containing an endpoint with `[no results]` in the `macAddresses` field:

```
{
  "data": {
```

©2026 Tanium Inc. All Rights Reserved

```
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host-1",
            "ipAddress": "192.0.2.10",
            "macAddresses": [
              "00:00:5E:00:53:10"
            ]
          }
        },
        {
          "node": {
            "name": "example-host-2",
            "ipAddress": "192.0.2.20",
            "macAddresses": [
              "[no results]"
            ]
          }
        }
      ]
    }
  }
}
```

If you update the `excludeNoResults` variable definition to `true`, the query response excludes endpoints with `[no results]` in the returned fields:

```
{
  "first": 2,
  "excludeNoResults": true
}
```

Example response containing no endpoints with `[no results]` in the `macAddresses` field:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host-1",
            "ipAddress": "192.0.2.10",
            "macAddresses": [
              "00:00:5E:00:53:10"
            ]
          }
        },
        {
          "node": {
            "name": "example-host-3",
            "ipAddress": "192.0.2.30",
            "macAddresses": [
              "00:00:5E:00:53:30"
            ]
          }
        }
      ]
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Get endpoints using aliases

(query.endpoints(source))

The following query retrieves known endpoints using aliases to rename the information returned and prevent data collision due to overlapping field names. The `tds` alias retrieves the `id` and `name` of known endpoints from Tanium Data Service. The `ts` alias retrieves the IPv6 address `value` of known endpoints from the Tanium Server and the sensor `name`.

```
query exampleGetEndpointsAliases($time: Int) {
  tds: endpoints {
    edges {
      node {
        id
        name
      }
    }
  }
  ts: endpoints(source: {ts: {stableWaitTime: $time}}) {
    edges {
      node {
        sensorReadings(sensors: [{name: "IPv6 Address"}]) {
          columns {
            name
            values
          }
        }
      }
    }
  }
}
```

Include the time and sensor name variables in the **Variables** panel or in your variables dictionary:

```
{
  "time": 10
}
```

Example response:

```
{
  "data": {
    "tds": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "example-name"
          }
        },
        {
          "node": {
            "id": "54321",
```

©2026 Tanium Inc. All Rights Reserved

```
            "name": "example-name-2"
          }
        }
      ]
    },
    "ts": {
      "edges": [
        {
          "node": {
            "sensorReadings": {
              "columns": [
                {
                  "name": "IPv6 Address",
                  "values": [
                    "2001:db8::3"
                  ]
                }
              ]
            }
          }
        },
        {
          "node": {
            "sensorReadings": {
              "columns": [
                {
                  "name": "IPv6 Address",
                  "values": [
                    "2001:db8::2"
                  ]
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Get endpoints using multiple sensors

(query.endpoints.edges.node.sensorReadings)

The following query retrieves known endpoints using multiple sensors. The request specifies the sensor name as one of the fields returned in the request, to identify the sensor that reported the information.

```
query exampleGetEndpointsApplications {
  endpoints {
    edges {
      node {
        computerID
        sensorReadings(sensors: [{name: "Installed Applications"}, {name: "Running
Applications"}]) {
          columns {
            name
            values
            sensor {
```

©2026 Tanium Inc. All Rights Reserved

```
              name
            }
          }
        }
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "987654321",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "Tanium Client 7.8.1.3126"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "7.8.1.3126"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Silent Uninstall String",
                  "values": [
                    "\"C:\\Program Files (x86)\\Tanium\\Tanium Client\\uninst.exe\""
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Uninstallable",
                  "values": [
                    "Is Uninstallable"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Name",
                  "values": [
                    "Tanium Client"
                  ],
                  "sensor": {
```

```
                    "name": "Running Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "7.8.1.3126"
                  ],
                  "sensor": {
                    "name": "Running Applications"
                  }
                },
                {
                  "name": "Process Name",
                  "values": [
                    "TaniumClient.exe"
                  ],
                  "sensor": {
                    "name": "Running Applications"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Get rich endpoint data

(query.endpoints(first))

The following query demonstrates using nested fields to retrieve categorized endpoint data.

Note: The `first: $first` argument passes the query variable value `2` and retrieves two records; set this value higher to retrieve more records at a time. For more information about pagination arguments, see Pagination on page 22.

```
query exampleGetRichEndpointData($first: Int) {
  endpoints(first: $first) {
    edges {
      node {
        name
        computerID
        ipAddress
        isVirtual
        chassisType
        systemUUID
        domainName
        os {
          name
          platform
          generation
```

©2026 Tanium Inc. All Rights Reserved

```
        }
        processor {
          architecture
          cacheSize
          consumption
          cpu
          family
          manufacturer
          speed
        }
        lastLoggedInUser
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasNextPage
    }
  }
}
```

Include the first pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "Test-01",
            "computerID": "1234567890",
            "ipAddress": "192.0.2.10",
            "isVirtual": false,
            "chassisType": "TSE-Error: Unknown - dmidecode unavailable",
            "systemUUID": "TSE-Error: Unknown - dmidecode unavailable",
            "domainName": "(none)",
            "os": {
              "name": "Red Hat Enterprise Linux Server release 5.11 (Tikanga)",
              "platform": "Linux",
              "generation": "Red Hat Enterprise Linux 5"
            },
            "processor": {
              "architecture": "x86_64",
              "cacheSize": "16384 KB",
              "consumption": "9.9 %",
              "cpu": "Intel Core Processor (Haswell, no TSX, IBRS)",
              "family": "6",
              "manufacturer": "GenuineIntel",
              "speed": "2600 Mhz"
            },
            "lastLoggedInUser": "reboot"
          }
        },
        {
          "node": {
```

©2026 Tanium Inc. All Rights Reserved

```
            "name": "Test-02",
            "computerID": "3216549870",
            "ipAddress": "192.0.2.20",
            "isVirtual": true,
            "chassisType": "Virtual",
            "systemUUID": "[no results]",
            "domainName": "(none)",
            "os": {
              "name": "CentOS Linux release 8.4.2105",
              "platform": "Linux",
              "generation": "CentOS 8"
            },
            "processor": {
              "architecture": "x86_64",
              "cacheSize": "35840 KB",
              "consumption": "18.6 %",
              "cpu": "Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz",
              "family": "6",
              "manufacturer": "GenuineIntel",
              "speed": "2400 Mhz"
            },
            "lastLoggedInUser": "tester-5"
          }
        }
      ],
      "pageInfo": {
        "startCursor": "NTc2NTM4MDow",
        "endCursor": "NTc2NTM4MDoxOQ==",
        "hasNextPage": true
      }
    }
  }
}
```

Get a set of endpoints

(query.endpoints(filter).edges.node.sensorReadings)

The following query retrieves a set of endpoints. The query demonstrates the use of the `sensorReadings` field and contains a filter argument to retrieve endpoints whose names contain the letter `t`. The results are paginated to 3 records.

```
query exampleGetEndpointNamesWithT($first: Int, $filterOp: FieldFilterOp!, $path: String,
$value: String) {
  endpoints(first: $first, filter: {op: $filterOp, path: $path, value: $value}) {
    edges {
      node {
        name
        ipAddress
        sensorReadings(sensors: [{name: "EID Last Seen"}]) {
          columns {
            name
            values
          }
        }
      }
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Include the filter-related and sensor name variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 3,
  "filterOp": "MATCHES",
  "path": "name",
  "value": "t.*"
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "test-1",
            "ipAddress": "192.0.2.10",
            "sensorReadings": {
              "columns": [
                {
                  "name": "EID Last Seen",
                  "values": [
                    "Mon, 08 Nov 2021 21:29:28 +0000"
                  ]
                }
              ]
            }
          }
        },
        {
          "node": {
            "name": "test-2",
            "ipAddress": "192.0.2.20",
            "sensorReadings": {
              "columns": [
                {
                  "name": "EID Last Seen",
                  "values": [
                    "Mon, 08 Nov 2021 21:29:28 +0000"
                  ]
                }
              ]
            }
          }
        },
        {
          "node": {
            "name": "test-3",
            "ipAddress": "192.0.2.30",
            "sensorReadings": {
              "columns": [
                {
                  "name": "EID Last Seen",
                  "values": [
                    "Mon, 08 Nov 2021 21:17:17 +0000"
                  ]
```

©2026 Tanium Inc. All Rights Reserved

```
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Get a set of endpoints and update sensor maximum age

(query.endpoints(filter).edges.node.sensorReadings)

The following query retrieves a set of endpoints. The query demonstrates the use of the `sensorReadings` field and contains a filter argument to retrieve endpoints whose names contain the letter `t`, and returns the total memory on each endpoint. The results are paginated to 3 records. Because `maxAge` is defined for the Total Memory sensor, the sensor reevaluates if the cached Total Memory sensor results are older than 300 seconds. This `maxAge` setting does not affect other response fields (`name` and `ipAddress`) that are not associated with the Total Memory sensor.

Note: You can only use `maxAge` with the Tanium Server (`ts`) data source. The response returns a validation error if you try to use `maxAge` with the Tanium Data Service (`tds`) data source.

Note: Setting this value to very small periods of time might cause excessive resource usage on endpoints and frequent sensor evaluation.

```
query exampleGetEndpointNamesWithTSMaxAge($first: Int, $filterOp: FieldFilterOp!, $path:
String, $value: String, $maxAge: Int) {
  endpoints(
    first: $first
    source: {ts: {stableWaitTime: 10}}
    filter: {op: $filterOp, path: $path, value: $value}
  ) {
    edges {
      node {
        name
        ipAddress
        sensorReadings(sensors: [{name: "Total Memory", maxAge: $maxAge}]) {
          columns {
            name
            values
          }
        }
      }
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Include the filter-related and sensor name variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 3,
  "filterOp": "MATCHES",
  "path": "name",
  "value": "t.*",
  "maxAge": 300
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "t_example_host",
            "ipAddress": "192.0.2.10",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Total Memory",
                  "values": [
                    "4000 MB"
                  ]
                }
              ]
            }
          }
        },
        {
          "node": {
            "name": "t_example_host_2",
            "ipAddress": "192.0.2.20",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Total Memory",
                  "values": [
                    "3900 MB"
                  ]
                }
              ]
            }
          }
        },
        {
          "node": {
            "name": "t_example_host_3",
            "ipAddress": "192.0.2.30",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Total Memory",
                  "values": [
```

©2026 Tanium Inc. All Rights Reserved

```
                    "3800 MB"
                  ]
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Similarly, if you define multiple sensors, the `maxAge` setting only affects sensors for which it is defined. The following query adds the Operating System sensor to the prior query, but does not define a `maxAge` setting for that sensor. The Total Memory sensor reevaluates if the cached Total Memory sensor results are older than the `maxAge` setting (300 seconds). The Operating System sensor reevaluates based on the Operating System sensor default maximum age setting (1 day), and not the `maxAge` setting for the Total Memory sensor.

```
query exampleGetEndpointNamesWithTS1MaxAge($first: Int, $filterOp: FieldFilterOp!, $path:
String, $value: String, $maxAge: Int) {
  endpoints(
    first: $first
    source: {ts: {stableWaitTime: 10}}
    filter: {op: $filterOp, path: $path, value: $value}
  ) {
    edges {
      node {
        name
        ipAddress
        sensorReadings(sensors: [{name: "Total Memory", maxAge: $maxAge}, {name: "Operating
System"}]) {
          columns {
            name
            values
          }
        }
      }
    }
  }
}
```

Include the filter-related and sensor name variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 3,
  "filterOp": "MATCHES",
  "path": "name",
  "value": "t.*",
  "maxAge": 300
}
```

©2026 Tanium Inc. All Rights Reserved

Unregistered sensor query

(query.endpoints.edges.node.sensorReadings)

The following query retrieves the operating system platform from all endpoints.

> **Note:** In Gateway, a sensor is unregistered if the sensor is not represented by a named field in the Gateway schema. This has no correlation to registering sensors in Tanium Data Service.

```
query exampleGetEndpointsUnregisteredSensor {
  endpoints {
    edges {
      node {
        id
        name
        sensorReadings(sensors: [{name: "OS Platform"}]) {
          columns {
            name
            values
          }
        }
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "Test-01",
            "sensorReadings": {
              "columns": [
                {
                  "name": "OS Platform",
                  "values": [
                    "Linux"
                  ]
                }
              ]
            }
          }
        },
        {
          "node": {
            "id": "54321",
            "name": "Test-03",
            "sensorReadings": {
              "columns": [
```

©2026 Tanium Inc. All Rights Reserved

```
                {
                  "name": "OS Platform",
                  "values": [
                    "Linux"
                  ]
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

(1.1 - Experimental (Active Development)) Get mobile devices

(query.endpoints (source.ts.scope))

About this task

The following query retrieves a mobile device.

```
query exampleGetMobileDevices($first: Int!) {
  endpoints(
    first: $first
    source: {ts: {scope: ALL_TS_SOURCED_ENTITIES}}
    filter: {path: "entityType", op: EQ, value: "Mobile Device"}
  ) {
    edges {
      node {
        id
        namespace
        entityType
        entityProviderType
        isEndpoint
        entityProviderName
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "id": "3273300",
```

©2026 Tanium Inc. All Rights Reserved

```
            "namespace": "",
            "entityType": "Mobile Device",
            "entityProviderType": "Microsoft Intune",
            "isEndpoint": true,
            "entityProviderName": "Example Entity"
          }
        }
      ]
    }
  }
}
```

Get mobile devices from Tanium Data Service

(query.endpoints (source.tds.scope))

About this task

The following query retrieves a mobile device from Tanium Data Service.

```
query exampleGetMobileDevicesTDS($first: Int!) {
  endpoints(
    first: $first
    source: {tds: {scope: ALL_TS_SOURCED_ENTITIES}}
    filter: {path: "entityType", op: EQ, value: "Mobile Device"}
  ) {
    edges {
      node {
        id
        namespace
        entityType
        entityProviderType
        isEndpoint
        entityProviderName
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "id": "3573598",
            "namespace": "Microsoft Intune-Mobile Device",
            "entityType": "Mobile Device",
            "entityProviderType": "Microsoft Intune",
```

©2026 Tanium Inc. All Rights Reserved

```
            "isEndpoint": true,
            "entityProviderName": "Example Entity"
          }
        }
      ]
    }
  }
}
```

Unregistered parameterized sensor query

(query.endpoints.edges.node.sensorReadings)

The following query checks to see if each endpoint contains the C:\Windows\py.exe file.



Note: In Gateway, a sensor is unregistered if the sensor is not represented by a named field in the Gateway schema. This has no correlation to registering sensors in Tanium Data Service.

```
query exampleGetEndpointsParams($paramName1: String!, $paramValue1: String!) {
  endpoints(source: {ts: {}}) {
    edges {
      node {
        name
        id
        sensorReadings(
          sensors: [{name: "File Exists", params: [{name: $paramName1, value:
$paramValue1}]}]
        ) {
          columns {
            sensor {
              name
              params {
                name
                value
              }
            }
            values
          }
        }
      }
    }
  }
}
```

Include the sensor name and parameter variables in the **Variables** panel or in your variables dictionary:

```
{
  "paramName1": "file",
  "paramValue1": "C:\\Windows\\py.exe"
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "Test-01",
            "id": "12345",
            "sensorReadings": {
              "columns": [
                {
                  "sensor": {
                    "name": "File Exists",
                    "params": [
                      {
                        "name": "file",
                        "value": "C:\\Windows\\py.exe"
                      }
                    ]
                  },
                  "values": [
                    "File does not exist"
                  ]
                }
              ]
            }
          }
        },
        {
          "node": {
            "name": "[no results]",
            "id": "54225",
            "sensorReadings": {
              "columns": [
                {
                  "sensor": {
                    "name": "File Exists",
                    "params": [
                      {
                        "name": "file",
                        "value": "C:\\Windows\\py.exe"
                      }
                    ]
                  },
                  "values": [
                    "[no results]"
                  ]
                }
              ]
            }
          }
        },
        {
          "node": {
            "name": "[no results]",
            "id": "65456",
            "sensorReadings": {
              "columns": [
                {
                  "sensor": {
                    "name": "File Exists",
```


©2026 Tanium Inc. All Rights Reserved

```
                    "params": [
                      {
                        "name": "file",
                        "value": "C:\\Windows\\py.exe"
                      }
                    ]
                  },
                  "values": [
                    "[no results]"
                  ]
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Get selected columns from sensor query (query.endpoints.edges.node.sensorReadings.columns)

GET SELECTED COLUMNS FROM A SENSOR QUERY

The following query retrieves the first 2 endpoint results, and returns only the `Name` and `Silent Uninstall String` columns from the Installed Applications sensor.

```
query exampleGetEndpointsApplicationsColumns ($first: Int) {
  endpoints (first: $first) {
    edges {
      node {
        computerID
        sensorReadings(sensors: [{name: "Installed Applications", columns: ["Name", "Silent
Uninstall String"]}]) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

©2026 Tanium Inc. All Rights Reserved

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "987654321",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "Tanium Client 7.8.1.3126"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Silent Uninstall String",
                  "values": [
                    "\"C:\\Program Files (x86)\\Tanium\\Tanium Client\\uninst.exe\""
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```


GET HIDDEN COLUMNS FROM A SENSOR QUERY

The following query retrieves the first 2 endpoint results, and based on `includeHiddenColumns: true`, returns all columns from the Application Crashes in Last X Days sensor, including a non-hidden column (Process) and a hidden column (Date).

To return hidden columns, you can also specify column names in the filter. By default, hidden columns are not returned.

```
query exampleGetEndpointsHiddenColumns ($first: Int) {
  endpoints (first: $first) {
    edges {
      node {
        computerID
        sensorReadings(sensors: [{name: "Application Crashes in Last X Days"}],
includeHiddenColumns: true
        ) {
          columns {
            name
            values
            sensor {
```

©2026 Tanium Inc. All Rights Reserved

```
              name
            }
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "1814669575",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Process",
                  "values": [
                    "example.exe"
                  ],
                  "sensor": {
                    "name": "Application Crashes in Last X Days"
                  }
                },
                {
                  "name": "Date",
                  "values": [
                    "12/30/2022"
                  ],
                  "sensor": {
                    "name": "Application Crashes in Last X Days"
                  }
                }
              ]
            }
          }
        },
        {
          "node": {
            "computerID": "2508280668",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Process",
                  "values": [
                    "example2.exe"
                  ],
                  "sensor": {
                    "name": "Application Crashes in Last X Days"
                  }
```

©2026 Tanium Inc. All Rights Reserved

```
                },
                {
                  "name": "Date",
                  "values": [
                    "12/29/2022"
                  ],
                  "sensor": {
                    "name": "Application Crashes in Last X Days"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Filter sensor query results (query.endpoints.edges.node.sensorReadings (sensors.filter))

FILTER THE RESULTS FROM A SENSOR QUERY

The following query retrieves the first 2 endpoint results, and returns the application `Name` and `Version` columns for only the endpoints that have the `tar` application installed, based on the results from the Installed Applications sensor.

```
query exampleGetEndpointsInstalledTar($first: Int) {
  endpoints(first: $first) {
    edges {
      node {
        computerID
        sensorReadings(
          sensors: [{name: "Installed Applications", filter: {column: "Name", op: EQ, value:
"tar", restrictOwner: false}}]
        ) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "1234567890",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.34"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        },
        {
          "node": {
            "computerID": "2345678901",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.33"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
```


©2026 Tanium Inc. All Rights Reserved

```
}
```

FILTER SENSOR QUERY RESULTS FROM A FILTERED SET OF ENDPOINTS

The following query retrieves the first 2 endpoints that are a member of the `All Linux` computer group, and returns the application `Name` and `Version` columns for only the endpoints that have the `tar` application installed, based on the results from the Installed Applications sensor.

```
query exampleGetLinuxEndpointsInstalledTar ($first: Int){
  endpoints(first: $first, filter: {memberOf: {name: "All Linux"}}) {
    edges {
      node {
        computerID
        sensorReadings(
          sensors: [{name: "Installed Applications", filter: {column: "Name", op: EQ, value:
"Tar", restrictOwner: false}}]
        ) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "3456789012",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
```

©2026 Tanium Inc. All Rights Reserved

```
                {
                  "name": "Version",
                  "values": [
                    "1.32"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        },
        {
          "node": {
            "computerID": "4567890123",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.30"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```


APPLY MULTI-COLUMN FILTER TO SENSOR QUERY RESULTS

The following query retrieves the first 2 endpoints, and returns the application `Name` and `Version` columns for only the endpoints that have the `tar` application version `1.30` installed, based on the results from the Installed Applications sensor.

```
query exampleGetEndpointsInstalledTar13($first: Int) {
  endpoints(first: $first) {
    edges {
      node {
        computerID
        sensorReadings(
          sensors: [{name: "Installed Applications", filter: {restrictOwner: false, filters:
[{column: "Name", op: EQ, value: "Tar"}, {column: "Version", op: EQ, value: "1.30"}]}}]
```

©2026 Tanium Inc. All Rights Reserved

```
        ) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "4567890123",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.30"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        },
        {
          "node": {
            "computerID": "5678901234",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
```

```
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.30"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

APPLY MULTI-COLUMN FILTER TO SENSOR QUERY RESULTS FOR FILTERED ENDPOINTS

The following query retrieves the first two endpoints that are members of the `All Linux` computer group, filtering out endpoints that return `[no results]`, and returns the application `Name` and `Version` columns for only the endpoints that have the `tar` application version `1.30` installed, based on the results from the Installed Applications sensor.

```
query exampleGetLinuxEndpointsInstalledTar130($first: Int, $excludeNoResults: Boolean!) {
  endpoints(
    first: $first
    source: {ts: {excludeNoResults: $excludeNoResults}}
    filter: {memberOf: {name: "All Linux"}}
  ) {
    edges {
      node {
        computerID
        sensorReadings(
          sensors: [{name: "Installed Applications", filter: {restrictOwner: false, filters:
[{column: "Name", op: EQ, value: "Tar"}, {column: "Version", op: EQ, value: "1.30"}]}}]
        ) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the pagination variable in the **Variables** panel or in your variables dictionary, and set `excludeNoResults` to `true` to filter out endpoints that return `[no results]`:

```
{
  "first": 2,
  "excludeNoResults": true
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "6789012345",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.30"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        },
        {
          "node": {
            "computerID": "7890123456",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.30"
                  ],
                  "sensor": {
```

©2026 Tanium Inc. All Rights Reserved

```
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

FILTER SENSOR QUERY RESULTS WITH RESTRICTOWNER OPTION

You can control which rows are returned by a sensor when you filter the sensor in the `sensorReadings` field. Use the `restrictOwner` option to control whether the filter in `sensorReadings` also decides which endpoints are returned.

- `restrictOwner: false`: Filter only the sensor rows. All endpoints that you request still appear, although some endpoints might return `[no results]`. Use this option when you want to retrieve all endpoints, but only those endpoints that match rows for that sensor.
- `restrictOwner: true`: Filter both endpoints and sensor rows. Only endpoints with at least one matching row are returned. Use this option when you want only endpoints that have a match.

The following query uses `restrictOwner: false` to retrieve all endpoints and then filters those endpoints. The query is paginated to return the first two results.

```
query exampleGetEndpointsInstalledTarRestrictOwnerFalse{
  endpoints(first: 2) {
    edges {
      node {
        computerID
        sensorReadings(
          sensors: [{
            name: "Installed Applications",
            filter: { column: "Name", op: EQ, value: "tar", restrictOwner: false }
          }]
        ) {
          columns { name values sensor { name } }
        }
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Note: To obtain similar results with a Tanium question, ask `Get Installed Applications from all entities with Is Linux equals True` and then filter the results by `Installed Applications: Name matches "tar"`. The same endpoints appear unpaginated.



Example response:


```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "2680890512",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "tar"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "1.34"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Silent Uninstall String",
                  "values": [
                    "nothing"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Uninstallable",
```


©2026 Tanium Inc. All Rights Reserved

```
                "values": [
                  "Not Uninstallable"
                ],
                "sensor": {
                  "name": "Installed Applications"
                }
              }
            ]
          }
        }
      },
      {
        "node": {
          "computerID": "3752377178",
          "sensorReadings": {
            "columns": [
              {
                "name": "Name",
                "values": [
                  "[no results]"
                ],
                "sensor": {
                  "name": "Installed Applications"
                }
              },
              {
                "name": "Version",
                "values": [
                  "[no results]"
                ],
                "sensor": {
                  "name": "Installed Applications"
                }
              },
              {
                "name": "Silent Uninstall String",
                "values": [
                  "[no results]"
                ],
                "sensor": {
                  "name": "Installed Applications"
                }
              },
              {
                "name": "Uninstallable",
                "values": [
                  "[no results]"
                ],
                "sensor": {
                  "name": "Installed Applications"
                }
              }
            ]
          }
        }
      }
    ]
  }
}
```


The following query uses `restrictOwner: true` to retrieve only endpoints that match the filter.

©2026 Tanium Inc. All Rights Reserved

The query is paginated to return the first two results.

```
query exampleGetEndpointsInstalledTarRestrictOwnerTrue {
  endpoints(first: 2) {
    edges {
      node {
        computerID
        sensorReadings(
          sensors: [{name: "Installed Applications", filter: {column: "Name", op: EQ, value:
"tar", restrictOwner: true}}]
        ) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```



Note: To obtain similar results with a Tanium question, include the condition in the targeting, such as `Get Installed Applications from all entities with Installed Applications:Name contains tar`. The same endpoints with tar installed appear unpaginated.

Endpoint cursor query

(query.endpoints.pageInfo)

The following query retrieves cursors for endpoint records. You can use these cursors to retrieve specific endpoint records on a page. For more information about cursors, see Pagination on page 22.

```
query exampleGetEndpointCursors {
  endpoints {
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
  }
}
```

Example response:

```
{
  "data": {
```

©2026 Tanium Inc. All Rights Reserved

```
    "endpoints": {
      "pageInfo": {
        "hasPreviousPage": false,
        "hasNextPage": true,
        "startCursor": "NTc2NTM4MDow",
        "endCursor": "NTc2NTM4MDoxOQ=="
      }
    }
  }
}
```

Paginated query

(query.endpoints(after, first))

The following query retrieves the first five endpoint records after the given cursor. For more information about cursors, see Pagination on page 22.

```
query exampleGetFirstFiveEndpoints($cursor: Cursor, $first: Int) {
  endpoints(after: $cursor, first: $first) {
    edges {
      node {
        name
        id
        ipAddress
      }
    }
    pageInfo {
      hasNextPage
      startCursor
      endCursor
    }
  }
}
```

Include the pagination-related variables in the **Variables** panel or in your variables dictionary:

```
{
  "cursor": "NTc2NTM4MDow",
  "first": 5
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "Test-06",
            "id": "6172",
            "ipAddress": "192.0.2.10"
          }
        },
        {
          "node": {
```

©2026 Tanium Inc. All Rights Reserved

```
            "name": "Test-07",
            "id": "87654",
            "ipAddress": "192.0.2.20"
          }
        },
        {
          "node": {
            "name": "Test-14",
            "id": "43584",
            "ipAddress": "192.0.2.30"
          }
        },
        {
          "node": {
            "name": "Test-03",
            "id": "37233",
            "ipAddress": "[no results]"
          }
        },
        {
          "node": {
            "name": "Test-55",
            "id": "12139",
            "ipAddress": "[no results]"
          }
        }
      ],
      "pageInfo": {
        "hasNextPage": true,
        "startCursor": "NTc2NTM4MDow",
        "endCursor": "NTc2NTM4MDoxOQ=="
      }
    }
  }
}
```

Get refreshed collection of endpoints (query.endpoints.collectionInfo.startCursor, query.endpoints(refresh))

RETRIEVE COLLECTION START CURSOR

The following query retrieves an endpoint that is slow to respond, with `collectionInfo.startCursor` included in the response to use for refreshing the request. For more information about how to refresh a collection, see Refreshing collections on page 25.

```
query exampleGetSlowEndpoint($first: Int, $path: String, $value: String, $expectedCount: Int)
{
  endpoints(
    first: $first
    filter: {path: $path, value: $value}
    source: {ts: {expectedCount: $expectedCount}}
  ) {
    edges {
      node {
        id
      }
```

©2026 Tanium Inc. All Rights Reserved

```
    }
    collectionInfo {
      startCursor
      success
      active
    }
  }
}
```

Add the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1,
  "path": "name",
  "value": "slow-endpoint",
  "expectedCount": 1
}
```

Example response for an ongoing collection with a `startCursor` value:

```
{
  "data": {
    "endpoints": {
      "edges": [],
      "collectionInfo": {
        "startCursor": "4t2cB0ZGNPqy4J4P7O7opwz/24/5zgPi6OinDAA=:0",
        "success": false,
        "active": true
      }
    }
  }
}
```

REFRESH COLLECTION USING STARTING CURSOR

The following query refreshes the collection using the returned `startCursor` value:

```
query exampleGetEndpointRefresh($refresh: Cursor, $first: Int, $path: String, $value: String,
$expectedCount: Int) {
  endpoints(
    refresh: $refresh
    first: $first
    filter: {path: $path, value: $value}
    source: {ts: {expectedCount: $expectedCount}}
  ) {
    edges {
      node {
        id
      }
    }
    collectionInfo {
      startCursor
      success
      active
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Add the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "refresh": "4t2cB0ZGNPqy4J4P7O7opwz/24/5zgPi6OinDAA=:0",
  "first": 1,
  "path": "name",
  "value": "slow-endpoint",
  "expectedCount": 1
}
```

Example response with a completed collection:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "id": 1
          }
        }
      ],
      "collectionInfo": {
        "startCursor": "4t2cB0ZGNPqy4J4P7O7opwz/24/5zgPi6OinDAA=:0",
        "success": true,
        "active": true
      }
    }
  }
}
```

Software characteristics query with filter

(query.endpoints.edges.node.deployedSoftwarePackages)

The following query retrieves endpoints that contain software installed by Deploy, where the package ID is `123`.

```
query exampleGetEndpointsPackage123($value1: String, $value2: String) {
  endpoints {
    edges {
      node {
        ipAddress
        isVirtual
        domainName
        os {
          generation
        }
        lastLoggedInUser
        deployedSoftwarePackages(
          filter: {filters: [{op: EQ, path: "id", value: $value1}, {op: EQ, path:
"applicability", value: $value2}]}
        ) {
          id
        }
      }
    }
```

```
  }
}
```

Include the value variables in the **Variables** panel or in your variables dictionary:

```
{
  "value1": "123",
  "value2": "Installed"
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "ipAddress": "192.0.2.10",
            "isVirtual": true,
            "domainName": "(none)",
            "os": {
              "generation": "Debian 9"
            },
            "lastLoggedInUser": "[no results]",
            "deployedSoftwarePackages": [
              {
                "id": "123"
              }
            ]
          }
        },
        {
          "node": {
            "ipAddress": "192.0.2.20",
            "isVirtual": true,
            "domainName": "(none)",
            "os": {
              "generation": "CentOS 7"
            },
            "lastLoggedInUser": "admin",
            "deployedSoftwarePackages": [
              {
                "id": "123"
              }
            ]
          }
        },
        {
          "node": {
            "ipAddress": "192.0.2.30",
            "isVirtual": true,
            "domainName": "(none)",
            "os": {
              "generation": "Ubuntu 20.04"
            },
            "lastLoggedInUser": "admin",
            "deployedSoftwarePackages": [
              {
                "id": "123"
              }
```

©2026 Tanium Inc. All Rights Reserved

```
            ]
          }
        }
      ]
    }
  }
}
```

Get applications installed on endpoint

(query.endpoints.edges.node.installedApplications)

The following query retrieves the applications installed on the first `2` endpoints, the application version, the program used to uninstall the application, and whether you can uninstall the application using Tanium.

```
query exampleGetInstalledApplications($first: Int) {
  endpoints(first: $first) {
    edges {
      node {
        id
        name
        installedApplications {
          name
          silentUninstallString
          uninstallable
          version
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "example-host",
            "installedApplications": [
              {
                "name": "Notepad++ (64-bit x64)",
                "silentUninstallString": "\"C:\\Program Files\\Notepad++\\uninstall.exe\"",
                "uninstallable": false,
                "version": "8.4.2"
              },
```

©2026 Tanium Inc. All Rights Reserved

```
              {
                "name": "Microsoft Edge",
                "silentUninstallString": "\"C:\\Program Files
(x86)\\Microsoft\\Edge\\Application\\103.0.1264.49\\Installer\\setup.exe\" --uninstall --
msedge --channel=stable --system-level --verbose-logging",
                "uninstallable": false,
                "version": "103.0.1264.62"
              },
              {
                "name": "Tanium Client 7.8.1.3126",
                "silentUninstallString": "\"C:\\Program Files (x86)\\Tanium\\Tanium
Client\\uninst.exe\"",
                "uninstallable": false,
                "version": "7.8.1.3126"
              }
            ]
          }
        },
        {
          "node": {
            "id": "23456",
            "name": "example-host-2",
            "installedApplications": [
              {
                "name": "Google Chrome",
                "silentUninstallString": "MsiExec.exe
/X{6C53E336-8514-3BA8-8852-427A9FB5DC91}",
                "uninstallable": true,
                "version": "103.0.5060.114"
              },
              {
                "name": "Tanium Client 7.8.1.3126",
                "silentUninstallString": "\"C:\\Program Files (x86)\\Tanium\\Tanium
Client\\uninst.exe\"",
                "uninstallable": false,
                "version": "7.8.1.3126"
              }
            ]
          }
        }
      ]
    }
  }
}
```

Get Microsoft Sentinel endpoint information

(query.endpoints.edges.node.sentinel)

The following query retrieves Microsoft Sentinel information for the first `2` endpoints, including Windows Defender and System Center Configuration Manager (SCCM) client details.

```
query exampleGetSentinelInfo($first: Int) {
  endpoints(first: $first) {
    edges {
      node {
        id
        name
        sentinel {
```

©2026 Tanium Inc. All Rights Reserved

```
          sccmClient {
            health
            reason
          }
          windowsDefender {
            antiSpywareEnabled
            antivirusEnabled
            antivirusSignatureUpdateAge
            clientVersion
            healthy
            installed
            processEnabled
            processStartType
            quickScanAge
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "Example_host",
            "sentinel": {
              "sccmClient": {
                "health": "Healthy",
                "reason": ""
              },
              "windowsDefender": {
                "antiSpywareEnabled": "True",
                "antivirusEnabled": "True",
                "antivirusSignatureUpdateAge": "0",
                "clientVersion": "1.1.19500.2",
                "healthy": "true",
                "installed": "True",
                "processEnabled": "True",
                "processStartType": "Normal",
                "quickScanAge": "1"
              }
            }
          }
        },
        {
          "node": {
            "id": "23456",
            "name": "Example_host_2",
            "sentinel": {
```

©2026 Tanium Inc. All Rights Reserved

```
              "sccmClient": {
                "health": "Needs Attention",
                "reason": "SCCM Not Installed"
              },
              "windowsDefender": {
                "antiSpywareEnabled": "True",
                "antivirusEnabled": "True",
                "antivirusSignatureUpdateAge": "1",
                "clientVersion": "1.1.19500.2",
                "healthy": "true",
                "installed": "True",
                "processEnabled": "True",
                "processStartType": "Normal",
                "quickScanAge": "0"
              }
            }
          }
        }
      ]
    }
  }
}
```

(1.0 - Experimental (Early Development)) Get endpoint counts

(query.endpointCounts)

GET THE COUNT OF ENDPOINTS

You can retrieve the count of endpoints that match sensor column values. Each node returned in the response can contain sensor columns and associated values for that sensor result row, and the count of endpoints matching those values. For multi-column sensors, you can return all column values.

This query only supports the Tanium Server (**ts**) source. If you define the Tanium Data Service (**tds**) source, the response returns an error.

The following query retrieves endpoint counts from the Tanium Server based on the Installed Applications multi-column sensor values.

```
query exampleCountEndpointAllSensorColumns($stableWaitTime: Int, $sensorName: String!) {
  endpointCounts(
    input: {source: {ts: {stableWaitTime: $stableWaitTime}}, sensors: [{name: $sensorName}]}
  ) {
    edges {
      node {
        count
        columns {
          columnName
          sensor {
            name
          }
          values
        }
```

©2026 Tanium Inc. All Rights Reserved

```
      }
    }
  }
}
```

Include the wait time and sensor name variables in the **Variables** panel or in your variables dictionary:

```
{
  "stableWaitTime": 60,
  "sensorName": "Installed Applications"
}
```

Example response:

```
{
  "data": {
    "endpointCounts": {
      "edges": [
        {
          "node": {
            "count": 3,
            "columns": [
              {
                "columnName": "Name",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "curl",
                  "libcurl3-gnutls:amd64",
                  "libcurl4:amd64"
                ]
              },
              {
                "columnName": "Version",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "8.6.0",
                  "16.04",
                  "8.6.0-3"
                ]
              },
              {
                "columnName": "Silent Uninstall String",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "nothing",
                  "nothing",
                  "nothing"
                ]
              },
              {
                "columnName": "Uninstallable",
                "sensor": {
                  "name": "Installed Applications"
                },
```

©2026 Tanium Inc. All Rights Reserved

```
                "values": [
                  "Not Uninstallable",
                  "Not Uninstallable",
                  "Not Uninstallable"
                ]
              }
            ]
          }
        },
        {
          "node": {
            "count": 2,
            "columns": [
              {
                "columnName": "Name",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "Notepad++ (64-bit x64)"
                ]
              },
              {
                "columnName": "Version",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "8.5.1"
                ]
              },
              {
                "columnName": "Silent Uninstall String",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "\"C:\\Program Files\\Notepad++\\uninstall.exe\""
                ]
              },
              {
                "columnName": "Uninstallable",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "Not Uninstallable"
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```


GET THE COUNT OF ENDPOINTS BY ENTITY TYPE

The following query retrieves endpoint counts from the Tanium Server based on the entity type.

©2026 Tanium Inc. All Rights Reserved

```
query getEndpointTypeCounts {
  endpointCounts(
    input: {source: {ts: {scope: ALL_TS_SOURCED_ENTITIES}}, sensors: [{name: "Entity Type"}]}
  ) {
    edges {
      node {
        columns {
          columnName
          values
        }
        count
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "endpointCounts": {
      "edges": [
        {
          "node": {
            "columns": [
              {
                "columnName": "Entity Type",
                "values": [
                  "Computer"
                ]
              }
            ],
            "count": 162
          }
        },
        {
          "node": {
            "columns": [
              {
                "columnName": "Entity Type",
                "values": [
                  "Mobile Device"
                ]
              }
            ],
            "count": 8
          }
        }
      ]
    }
  }
}
```

The following query retrieves endpoint counts from the Tanium Server based on the Installed Applications multi-column sensor, but filters to return only endpoints with libcurl installed.

```
query exampleCountEndpointSelectSensorColumn($first: Int!, $stableWaitTime: Int, $sensorName:
String!, $columnName: String, $op: SensorValueFilterOp!, $columnValue: String) {
  endpointCounts(
    input: {source: {ts: {stableWaitTime: $stableWaitTime}}, sensors: [{name: $sensorName,
filter: {column: $columnName, op: $op, value: $columnValue}}]}
```

©2026 Tanium Inc. All Rights Reserved

```
  ) {
    edges {
      node {
        count
        columns {
          columnName
          sensor {
            name
          }
          values
        }
      }
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "stableWaitTime": 60,
  "sensorName": "Installed Applications",
  "columnName": "Name",
  "op": "EQ",
  "columnValue": "libcurl"
}
```

Example response:

```
{
  "data": {
    "endpointCounts": {
      "edges": [
        {
          "node": {
            "count": 7,
            "columns": [
              {
                "columnName": "Name",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "libcurl"
                ]
              },
              {
                "columnName": "Version",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "7.61.1"
                ]
              },
              {
                "columnName": "Silent Uninstall String",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "nothing"
```

©2026 Tanium Inc. All Rights Reserved

```
              ]
            },
            {
              "columnName": "Uninstallable",
              "sensor": {
                "name": "Installed Applications"
              },
              "values": [
                "Not Uninstallable"
              ]
            }
          ]
        }
      },
      {
        "node": {
          "count": 3,
          "columns": [
            {
              "columnName": "Name",
              "sensor": {
                "name": "Installed Applications"
              },
              "values": [
                "libcurl"
              ]
            },
            {
              "columnName": "Version",
              "sensor": {
                "name": "Installed Applications"
              },
              "values": [
                "7.76.1"
              ]
            },
            {
              "columnName": "Silent Uninstall String",
              "sensor": {
                "name": "Installed Applications"
              },
              "values": [
                "nothing"
              ]
            },
            {
              "columnName": "Uninstallable",
              "sensor": {
                "name": "Installed Applications"
              },
              "values": [
                "Not Uninstallable"
              ]
            }
          ]
        }
      }
    ]
  }
}
```


The following query retrieves endpoint counts from the Tanium Server based on the Custom Tag

©2026 Tanium Inc. All Rights Reserved

Exists single column sensor, and passes the **tag** parameter value **the-tag** to check for endpoints assigned that custom tag.

```
query exampleCountEndpointParamSensor($first: Int!, $stableWaitTime: Int, $sensorName:
String!, $paramName: String!, $paramValue: String!) {
  endpointCounts(
    input: {source: {ts: {stableWaitTime: $stableWaitTime}}, sensors: [{name: $sensorName,
params: [{name: $paramName, value: $paramValue}]}]}
  ) {
    edges {
      node {
        count
        columns {
          columnName
          sensor {
            name
          }
          values
        }
      }
    }
  }
}
```

Include the time, sensor, and parameter variables in the **Variables** panel or in your variables dictionary:

```
{
  "stableWaitTime": 60,
  "sensorName": "Custom Tag Exists",
  "paramName": "tag",
  "paramValue": "the-tag"
}
```

Example response:

```
{
  "data": {
    "endpointCounts": {
      "edges": [
        {
          "node": {
            "count": 10,
            "columns": [
              {
                "columnName": "Custom Tag Exists",
                "sensor": {
                  "name": "Custom Tag Exists"
                },
                "values": [
                  "False"
                ]
              }
            ]
          }
        },
        {
          "node": {
            "count": 2,
```

©2026 Tanium Inc. All Rights Reserved

```
            "columns": [
              {
                "columnName": "Custom Tag Exists",
                "sensor": {
                  "name": "Custom Tag Exists"
                },
                "values": [
                  "True"
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

(1.0 - Experimental (Early Development)) Get refreshed collection of endpoint counts using one sensor

(query.endpointCounts.collectionInfo.startCursor, query.endpointCounts(refresh))

RETRIEVE AND REFRESH ENDPOINT COUNT COLLECTION START CURSOR USING ONE SENSOR

The following query retrieves an endpoint count that is slow to respond, with `collectionInfo.startCursor` included in the response to use for refreshing the request. For more information about how to refresh a collection, see Refreshing collections on page 25.

```
query exampleGetSlowEndpointCount($first: Int!, $stableWaitTime: Int, $sensorName: String!) {
  endpointCounts(
    input: {first: $first, source: {ts: {stableWaitTime: $stableWaitTime}}, sensors: [{name:
$sensorName}]}
  ) {
    edges {
      node {
        count
        columns {
          columnName
          sensor {
            name
          }
          values
        }
      }
    }
    collectionInfo {
      startCursor
      success
      active
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
```

©2026 Tanium Inc. All Rights Reserved

```
    }
  }
}
```

Add the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1,
  "stableWaitTime": 60,
  "sensorName": "Installed Applications"
}
```

Example response for an ongoing collection with a `startCursor` value:

```
{
  "data": {
    "endpoints": {
      "edges": [],
      "collectionInfo": {
        "startCursor": "OEtXWEZYSnVicks4K09rTTJMR0s0d3pJcUlyakRPQ29pdU1NQUE9PTow",
        "success": false,
        "active": true
      },
      "pageInfo": {
        "hasPreviousPage": false,
        "hasNextPage": false,
        "startCursor": "OEtXWEZYSnVicks4K09rTTJMR0s0d3pJcUlyakRPQ29pdU1NQUE9PTow",
        "endCursor": "OEtXWEZYSnVicks4K09rTTJMR0s0d3pJcUlyakRPQ29pdU1NQUE9PTow"
      }
    }
  }
}
```

The following query refreshes the endpoint count collection using the returned `startCursor` value:

```
query exampleRefreshSlowEndpointCount($refresh: Cursor, $first: Int!, $stableWaitTime: Int,
$sensorName: String!) {
  endpointCounts(
    input: {refresh: $refresh, first: $first, source: {ts: {stableWaitTime:
$stableWaitTime}}, sensors: [{name: $sensorName}]}
  ) {
    edges {
      node {
        count
        columns {
          columnName
          sensor {
            name
          }
          values
        }
      }
    }
    collectionInfo {
      startCursor
      success
      active
    }
```

```
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
  }
}
```

Add the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "refresh": "OEtXWEZYSnVicks4K09rTTJMR0s0d3pJcUlyakRPQ29pdU1NQUE9PTow",
  "first": 1,
  "stableWaitTime": 60,
  "sensorName": "Installed Applications"
}
```

Example response with a completed collection:

```
{
  "data": {
    "endpointCounts": {
      "edges": [
        {
          "node": {
            "count": 1,
            "columns": [
              {
                "columnName": "Name",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "Cygwin Bash Prompt Here (x86_64)",
                  "Tanium Client 7.6.1.6062"
                ]
              },
              {
                "columnName": "Version",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "N/A",
                  "7.6.1.6062"
                ]
              },
              {
                "columnName": "Silent Uninstall String",
                "sensor": {
                  "name": "Installed Applications"
                },
                "values": [
                  "C:\\cygwin64\\bin\\sh -c \"PATH=/bin /bin/chere -u -s bash\"",
                  "\"C:\\Program Files (x86)\\Tanium\\Tanium Client\\uninst.exe\""
                ]
              },
              {
                "columnName": "Uninstallable",
                "sensor": {
```

©2026 Tanium Inc. All Rights Reserved

```
                "name": "Installed Applications"
              },
              "values": [
                "Not Uninstallable",
                "Not Uninstallable"
              ]
            }
          ]
        }
      }
    ],
    "collectionInfo": {
      "startCursor": "OEtXWEZYSnljcm1nb3JjRTJMR0s0d3kwcTRyakRPeXJpdU1NQUE9PTow",
      "success": false,
      "active": true
    },
    "pageInfo": {
      "hasPreviousPage": false,
      "hasNextPage": false,
      "startCursor": "OEtXWEZYSnVicks4K09rTTJMR0s0d3pJcUlyakRPQ29pdU1NQUE9PTow",
      "endCursor": "OEtXWEZYSnVicks4K09rTTJMR0s0d3pJcUlyakRPQ29pdU1NQUE9PTow"
    }
  }
 }
}
```


(1.0 - Experimental (Early Development)) Get refreshed collection of endpoint counts using multiple sensors

(query.endpointCounts.collectionInfo.startCursor, query.endpointCounts(refresh))

RETRIEVE AND REFRESH ENDPOINT COUNT COLLECTION START CURSOR USING MULTIPLE SENSORS

The following query retrieves an endpoint count that is slow to respond, based on the `Operating System Generation` and `RAM` sensors, with `collectionInfo.startCursor` included in the response to use for refreshing the request. For more information about how to refresh a collection, see Refreshing collections on page 25.

```
query exampleGetSlowEndpointCount2($first: Int!, $stableWaitTime: Int, $sensorName1: String!,
$sensorName2: String!) {
  endpointCounts(
    input: {first: $first, source: {ts: {stableWaitTime: $stableWaitTime}}, sensors: [{name:
$sensorName1}, {name: $sensorName2}]}
  ) {
    edges {
      node {
        count
        columns {
          columnName
          sensor {
            name
          }
          values
        }
      }
```

©2026 Tanium Inc. All Rights Reserved

```
    }
    collectionInfo {
      startCursor
      success
      active
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
  }
}
```

Add the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1,
  "stableWaitTime": 60,
  "sensorName1": "Operating System Generation",
  "sensorName2": "RAM"
}
```

Example response for an ongoing collection with a `startCursor` value:

```
{
  "data": {
    "endpoints": {
      "edges": [],
      "collectionInfo": {
        "startCursor": "NkJNRUJBVFlqODZiQ1pMbStPVU01dHo0NVF5ZTNmamxEQUE9OjA=",
        "success": false,
        "active": true
      },
      "pageInfo": {
        "hasPreviousPage": false,
        "hasNextPage": false,
        "startCursor": "NkJNRUJBVFlqODZiQ1pMbStPVU01dHo0NVF5ZTNmamxEQUE9OjA=",
        "endCursor": "NkJNRUJBVFlqODZiQ1pMbStPVU01dHo0NVF5ZTNmamxEQUE9OjA="
      }
    }
  }
}
```

REFRESH ENDPOINT COUNT COLLECTION USING STARTING CURSOR

The following query refreshes the endpoint count collection using the returned `startCursor` value:

```
query exampleRefreshSlowEndpointCount2($refresh: Cursor, $first: Int!, $stableWaitTime: Int,
$sensorName1: String!, $sensorName2: String!) {
  endpointCounts(
    input: {refresh: $refresh, first: $first, source: {ts: {stableWaitTime:
$stableWaitTime}}, sensors: [{name: $sensorName1}, {name: $sensorName2}]}
  ) {
    edges {
```

©2026 Tanium Inc. All Rights Reserved

```
      node {
        count
        columns {
          columnName
          sensor {
            name
          }
          values
        }
      }
    }
    collectionInfo {
      startCursor
      success
      active
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
  }
}
```

Add the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "refresh": "NkJNRUJBVFlqODZiQ1pMbStPVU01dHo0NVF5ZTNmamxEQUE9OjA=",
  "first": 1,
  "stableWaitTime": 60,
  "sensorName1": "Operating System Generation",
  "sensorName2": "RAM"
}
```

Example response with a completed collection:

```
{
  "data": {
    "endpointCounts": {
      "edges": [
        {
          "node": {
            "count": 1,
            "columns": [
              {
                "columnName": "Operating System Generation",
                "sensor": {
                  "name": "Operating System Generation"
                },
                "values": [
                  "Windows 10"
                ]
              },
              {
                "columnName": "RAM",
                "sensor": {
                  "name": "RAM"
                },
                "values": [
                  "4096 MB"
```

©2026 Tanium Inc. All Rights Reserved

```
                ]
              }
            ]
          }
        }
      ],
      "collectionInfo": {
        "startCursor": "NkJNRUJBVFlqODZiQ1pMbStPVU01dHo0NVF5ZTNmamxEQUE9OjA=",
        "success": false,
        "active": true
      },
      "pageInfo": {
        "hasPreviousPage": false,
        "hasNextPage": false,
        "startCursor": "NkJNRUJBVFlqODZiQ1pMbStPVU01dHo0NVF5ZTNmamxEQUE9OjA=",
        "endCursor": "NkJNRUJBVFlqODZiQ1pMbStPVU01dHo0NVF5ZTNmamxEQUE9OjA="
      }
    }
  }
}
```

(1.1 - Experimental (Release Candidate)) Sort returned endpoints

(query.endpoints(sort))

You can sort endpoints in the response based on either a single field or a single sensor column. The following query retrieves the first three endpoints from the Tanium Server and sorts them in ascending order based on the `computerID` value.

```
query exampleSortEndpointsByField($count: Int, $time: Int, $sort: EndpointFieldSort!, $first:
Int) {
  endpoints(source: {ts: {expectedCount: $count, stableWaitTime: $time}}, sort:[$sort],
first: $first) {
    edges {
      node {
        computerID
        name
        serialNumber
        ipAddress
      }
    }
  }
}
```

Include the count, time, sort, and pagination variables in the **Variables** panel or in your variables dictionary:

```
{
  "count": 1,
  "time": 10,
  "sort": {"path": "computerID", "order": "asc"},
  "first": 3
}
```

Example response:

©2026 Tanium Inc. All Rights Reserved

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "1234567890",
            "name": "ubuntu-test",
            "serialNumber": "Not Specified",
            "ipAddress": "10.168.20.30"
          }
        },
        {
          "node": {
            "computerID": "2345678901",
            "name": "CentOS-test-1",
            "serialNumber": "Not Specified",
            "ipAddress": "10.168.20.40"
          }
        },
        {
          "node": {
            "computerID": "3456789012",
            "name": "CentOS-test-2",
            "serialNumber": "Not Specified",
            "ipAddress": "10.168.20.50"
          }
        }
      ]
    }
  }
}
```

The following query retrieves the first three endpoints from the Tanium Server and sorts them in descending order based on the `CPU Consumption` single column sensor.

```
query exampleSortEndpointsSingleColumnSensor($count: Int, $time: Int, $sort:
EndpointFieldSort!, $first: Int) {
  endpoints(source: {ts: {expectedCount: $count, stableWaitTime: $time}}, sort: [$sort],
first: $first) {
    edges {
      node {
        computerID
        name
        serialNumber
        ipAddress
        processor {
          cpu
          consumption
          highConsumption
        }
      }
    }
  }
}
```

Include the count, time, sort, and pagination variables in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "count": 1,
  "time": 10,
  "sort": {
    "sensor": {
      "name": "CPU Consumption"
    },
    "order": "desc"
  },
  "first": 3
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "1234567890",
            "name": "example-host-1",
            "serialNumber": "12345678-90ab-cdef-1234-567890abcdef",
            "ipAddress": "192.0.2.10",
            "processor": {
              "cpu": "AWS Graviton2",
              "consumption": "95 %",
              "highConsumption": "N/A on Linux"
            }
          }
        },
        {
          "node": {
            "computerID": "2345678901",
            "name": "example-host-3",
            "serialNumber": "fedcba09-8765-4321-fedc-ba0987654321",
            "ipAddress": "192.0.2.30",
            "processor": {
              "cpu": "Intel(R) Core(TM) i7-8700B CPU @ 3.20GHz",
              "consumption": "85 %",
              "highConsumption": "N/A on Mac"
            }
          }
        },
        {
          "node": {
            "computerID": "3456789012",
            "name": "example-host-2",
            "serialNumber": "1234567890",
            "ipAddress": "192.0.2.20",
            "processor": {
              "cpu": "Intel(R) Core(TM) i7-8700B CPU @ 3.20GHz",
              "consumption": "66 %",
              "highConsumption": "75 %"
            }
          }
        }
      ]
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

The following query retrieves the first three endpoints from the Tanium Server and sorts them in ascending order based on the multi-column `Disk Free Space Status` sensor, `Available Space (%)` column, with a `low` parameter value of `15`.

```
query exampleSortEndpointsMultiColumnSensor($count: Int, $time: Int, $sort:
EndpointFieldSort!, $first: Int) {
  endpoints(source: {ts: {expectedCount: $count, stableWaitTime: $time}}, sort: [$sort],
first: $first) {
    edges {
      node {
        computerID
        name
        serialNumber
        ipAddress
        disks {
          name
          free
          total
          usedSpace
          usedPercentage
        }
      }
    }
  }
}
```

Include the count, time, sort, and pagination variables in the **Variables** panel or in your variables dictionary:

```
{
  "count": 1,
  "time": 10,
  "sort": {
    "sensor": {
      "name": "Disk Free Space Status", "column": "Available Space (%)",
"params":{"name":"low", "value":"15"}
    },
    "order": "asc"
  },
  "first": 3
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "1234567890",
            "name": "example-host-3",
            "serialNumber": "12345678-90ab-cdef-1234-567890abcdef",
            "ipAddress": "192.0.2.30",
            "disks": [
              {
                "name": "C",
                "free": "1 GB",
```

©2026 Tanium Inc. All Rights Reserved

```
                "total": "30 GB",
                "usedSpace": "28 GB",
                "usedPercentage": "93%"
              }
            ]
          }
        },
        {
          "node": {
            "computerID": "2345678901",
            "name": "example-host-2",
            "serialNumber": "fedcba09-8765-4321-fedc-ba0987654321",
            "ipAddress": "192.0.2.20",
            "disks": [
              {
                "name": "/",
                "free": "562M",
                "total": "7.6G",
                "usedSpace": "7.1G",
                "usedPercentage": "93%"
              },
              {
                "name": "/boot/efi",
                "free": "92M",
                "total": "98M",
                "usedSpace": "6.3M",
                "usedPercentage": "7%"
              }
            ]
          }
        },
        {
          "node": {
            "computerID": "3456789012",
            "name": "example-host-1",
            "serialNumber": "1234567890",
            "ipAddress": "192.0.2.10",
            "disks": [
              {
                "name": "C",
                "free": "13 GB",
                "total": "59 GB",
                "usedSpace": "45 GB",
                "usedPercentage": "77%"
              }
            ]
          }
        }
      ]
    }
  }
}
```


(1.1 - Experimental (Active Development)) Get endpoints using single sensor

(query.endpoints.edges.node.sensorReading)

The following query retrieves known endpoints using a single sensor. The request specifies the sensor name as one of the fields returned in the request, to identify the sensor that reported the

©2026 Tanium Inc. All Rights Reserved

information.

```
query exampleGetEndpointsApplications {
  endpoints {
    edges {
      node {
        computerID
        sensorReading(sensor: {name: "Installed Applications", columns: ["Name"]}) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "987654321",
            "sensorReading": {
              "columns": [
                {
                  "name": "Name",
                  "values": [
                    "Tanium Client 7.8.1.3126"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Version",
                  "values": [
                    "7.8.1.3126"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Silent Uninstall String",
                  "values": [
                    "\"C:\\Program Files (x86)\\Tanium\\Tanium Client\\uninst.exe\""
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                },
                {
                  "name": "Uninstallable",
```

©2026 Tanium Inc. All Rights Reserved

```
                  "values": [
                    "Is Uninstallable"
                  ],
                  "sensor": {
                    "name": "Installed Applications"
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

(1.2 - Experimental (Release Candidate)) Get endpoints using single sensor and converting result data type

(query.endpoints.edges.node.sensorReading)

The following query retrieves known endpoints using a single sensor, Tanium Reboot Days Ago, then converts the numeric sensor result data type in the response to float data type values. Note that you can return the raw string values (values), numeric sensor result data types as float values (floatValues), or integer sensor result data types as integers (intValues).

```
query exampleGetEndpointsApplications2 {
  endpoints {
    edges {
      node {
        computerID
        sensorReading(sensor: {name: "Tanium Reboot Days Ago"}) {
          floatValues
        }
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "987654321",
            "sensorReading": {
              "floatValues": [5]
            }
          }
        }
      ]
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

(1.2 - Experimental (Release Candidate)) Get question ID

(query.endpoints.collectionInfo.question.id)

When you submit an `endpoints` GraphQL query request using the Tanium Server (`ts`) data source, a Tanium question is issued to gather the endpoint data. You can retrieve the corresponding question ID in a GraphQL query response. The following query retrieves endpoint information for the first two endpoints, including computer ID, name, IP address, and serial number, and also retrieves the Tanium question ID used to gather this information.

```
query exampleGetQuestionID($first: Int, $count: Int, $time: Int) {
  endpoints(
    first: $first
    source: {ts: {expectedCount: $count, stableWaitTime: $time}}
  ) {
    edges {
      node {
        computerID
        name
        serialNumber
        ipAddress
      }
    }
    collectionInfo {
      question {
        id
      }
    }
  }
}
```

Include the count, time, and pagination variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "count": 2,
  "time": 10
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "computerID": "0987654321",
            "name": "example_endpoint",
            "serialNumber": "1234567890",
            "ipAddress": "192.0.2.10"
```

©2026 Tanium Inc. All Rights Reserved

```
          }
        },
        {
          "node": {
            "computerID": "2345678901",
            "name": "example_endpoint_2",
            "serialNumber": "6789012345",
            "ipAddress": "192.0.2.20"
          }
        }
      ],
      "collectionInfo": {
        "question": {
          "id": "21018950"
        }
      }
    }
  }
}
```

Action examples

The following queries and mutations create actions, retrieve action details, stop actions, approve scheduled actions, and delete actions.

Approve a scheduled action

(mutation.scheduledActionApprove)

The following mutation approves the scheduled action that matches the action ID.

```
mutation exampleApproveScheduledAction($id: ID!) {
  scheduledActionApprove(ref: {id: $id}) {
    approved
    id
    error {
      message
    }
  }
}
```

Include the scheduled action ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345"
}
```

Example response:

```
{
  "data": {
    "scheduledActionApprove": {
      "approved": true,
```

©2026 Tanium Inc. All Rights Reserved

```
      "id": "12345",
      "error": null
    }
  }
}
```

Create action

(mutation.actionCreate)

Note: There are multiple mutations that create actions.

- Use `mutation.actionCreate` to deploy a one-time action for either a custom package or for a Tanium package that is not supported by `mutation.actionPerform`. For examples, see Create action on page 132.
- Use `mutation.scheduledActionCreate` to deploy a recurring action for either a custom package or a Tanium package that is not supported by `mutation.actionPerform`. For examples, see Create scheduled action on page 135.
- For all other uses cases, use `mutation.actionPerform`, which can deploy any type of package. This includes if you want to deploy a predefined package, especially if you deploy the package to both Windows and non-Windows endpoints. For examples, see Perform an action operation on page 156.

Note: Each request that calls `mutation.actionCreate` creates a copy of the referenced package with the action parameters, even if a similar package already exists.

The following mutation creates an action, referencing a package that does not require parameters and target computer group, and setting distribute over time and expiration details.

```
mutation exampleCreateTaniumAction($comment: String, $distributeSeconds: Int, $expireSeconds:
Int, $name: String, $packageID: ID, $startTime: Time, $actionGroupID: ID, $targetGroupID: ID)
{
  actionCreate(
    input: {comment: $comment, name: $name, package: {id: $packageID}, targets: {actionGroup:
{id: $actionGroupID}, targetGroup: {id: $targetGroupID}}, schedule: { distributeSeconds:
$distributeSeconds, expireSeconds: $expireSeconds, startTime: $startTime}}
  ) {
    action {
      id
```

©2026 Tanium Inc. All Rights Reserved

```
    }
    error {
      message
    }
  }
}
```

Include the action creation variables in the **Variables** panel or in your variables dictionary:

```
{
  "comment": "Example created action",
  "distributeSeconds": 60,
  "expireSeconds": 900,
  "name": "Example created action",
  "packageID": 12345,
  "startTime": "2026-02-12T00:00:00+00:00",
  "targetGroupID": 67890,
  "actionGroupID": 12345
}
```

Example response:

```
{
  "data": {
    "actionCreate": {
      "action": {
        "id": "123456789"
      },
      "error": null
    }
  }
}
```

If your package requires parameters, pass `params` as part of the `actionCreate (input.package)` argument.

```
mutation exampleCreateTaniumActionWithParameters($comment: String, $distributeSeconds: Int,
$expireSeconds: Int, $name: String, $packageID: ID, $param1: String!, $param2: String!,
$startTime: Time, $actionGroupID: ID, $targetGroupID: ID) {
  actionCreate(
    input: {comment: $comment, name: $name, package: {id: $packageID, params: [$param1,
$param2]}, targets: {actionGroup: {id: $actionGroupID}, targetGroup: {id: $targetGroupID}},
schedule: { distributeSeconds: $distributeSeconds, expireSeconds: $expireSeconds, startTime:
$startTime}}
  ) {
    action {
      id
    }
    error {
      message
    }
  }
}
```

Include the action creation variables in the **Variables** panel or in your variables dictionary:

```
{
  "comment": "Example created action",
```

©2026 Tanium Inc. All Rights Reserved

```
  "distributeSeconds": 60,
  "expireSeconds": 1200,
  "name": "Example created action",
  "packageID": 12345,
  "param1": "example_value_1",
  "param2": "example_value_2",
  "startTime": "2026-02-12T00:00:00+00:00",
  "targetGroupID": 67890,
  "actionGroupID": 12345
}
```

The following example shows how to tag metadata through `mutation.actionCreate`.

```
mutation exampleActionCreateWithMetadata($input: ActionCreateInput!) {
  actionCreate(input: $input) {
    action {
      id
      name
      targets {
        actionGroup {
          id
          name
        }
        targetGroup {
          id
          name
        }
      }
      package {
        id
        name
      }
      metadata {
        name
        value
      }
    }
    error {
      message
    }
  }
}
```

Include the action operation variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "One-time tag action",
    "package": {
      "name": "Custom Tagging - Add Tags (Non-Windows)",
      "params": ["tagA tagB"]
    },
    "targets": { "targetGroup": { "name": "All Linux" } },
    "metadata": [
      { "name": "Metadata A", "value": "Value A" },
      { "name": "Metadata B", "value": "Value B" }
    ]
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "actionCreate": {
      "action": {
        "id": "278",
        "name": "One-time tag action",
        "targets": {
          "actionGroup": {
            "id": "3",
            "name": "Default"
          },
          "targetGroup": {
            "id": "1382",
            "name": ""
          }
        },
        "package": {
          "id": "253",
          "name": "Custom Tagging - Add Tags (Non-Windows)"
        },
        "metadata": [
          {
            "name": "Metadata A",
            "value": "Value A"
          },
          {
            "name": "Metadata B",
            "value": "Value B"
          }
        ]
      },
      "error": null
    }
  }
}
```


Create scheduled action

(mutation.scheduledActionCreate)

©2026 Tanium Inc. All Rights Reserved

Note: There are multiple mutations that create actions.

- Use `mutation.actionCreate` to deploy a one-time action for either a custom package or for a Tanium package that is not supported by `mutation.actionPerform`. For examples, see Create action on page 132.
- Use `mutation.scheduledActionCreate` to deploy a recurring action for either a custom package or a Tanium package that is not supported by `mutation.actionPerform`. For examples, see Create scheduled action on page 135.
- For all other uses cases, use `mutation.actionPerform`, which can deploy any type of package. This includes if you want to deploy a predefined package, especially if you deploy the package to both Windows and non-Windows endpoints. For examples, see Perform an action operation on page 156.

Note: Each request that calls `mutation.scheduledActionCreate` creates a copy of the referenced package with the action parameters, even if a similar package already exists.

The following query creates a scheduled action, referencing a package without parameters and target computer group, and setting start and end time, distribute over time, and expiration details.

```
mutation exampleCreateScheduledAction($comment: String, $distributeSeconds: Int, $endTime:
Time, $expireSeconds: Int, $name: String, $packageID: ID, $reissueSeconds: Int!, $startTime:
Time, $actionGroupID: ID, $targetGroupID: ID, $metadata: [MetadataInput!]) {
  scheduledActionCreate(
    input: {comment: $comment, schedule: {distributeSeconds: $distributeSeconds, endTime:
$endTime, expireSeconds: $expireSeconds, reissueSeconds: $reissueSeconds, startTime:
$startTime}, name: $name, package: {id: $packageID}, targets: {actionGroup: {id:
$actionGroupID}, targetGroup: {id: $targetGroupID}}, metadata: $metadata}
  ) {
    scheduledAction {
      approved
      approver {
        persona {
          id
        }
        user {
          id
        }
      }
      comment
      creator {
        persona {
          id
        }
        user {
```

©2026 Tanium Inc. All Rights Reserved

```
          id
        }
      }
      distributeSeconds
      endTime
      expireSeconds
      id
      issueCount
      issueSeconds
      lastAction {
        id
        metadata {
          name
          value
        }
      }
      lastStartTime
      name
      nextStartTime
      package {
        id
      }
      startTime
      status
      targets {
        actionGroup {
          id
        }
        targetGroup {
          id
        }
      }
      metadata {
        name
        value
      }
    }
    error {
      message
    }
  }
}
```

Include the action-related variables in the **Variables** panel or in your variables dictionary:

```
{
  "comment": "Example scheduled action",
  "distributeSeconds": 60,
  "endTime": "2026-02-16T00:00:00+00:00",
  "expireSeconds": 900,
  "name": "Example scheduled action",
  "packageID": 12345,
  "reissueSeconds": 901,
  "startTime": "2026-02-12T00:00:00+00:00",
  "targetGroupID": 67890,
  "actionGroupID": 123456
}
```

Example response:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "data": {
    "scheduledActionCreate": {
      "scheduledAction": {
        "approved": true,
        "approver": {
          "persona": null,
          "user": {
            "id": "567"
          }
        },
        "comment": "Example scheduled action",
        "creator": {
          "persona": null,
          "user": {
            "id": "971"
          }
        },
        "distributeSeconds": 60,
        "endTime": "2026-02-16T00:00:00Z",
        "expireSeconds": 900,
        "id": "863509",
        "issueCount": 0,
        "issueSeconds": 901,
        "lastAction": {
          "id": "16076213",
          "metadata": []
        },
        "lastStartTime": null,
        "name": "Example scheduled action",
        "nextStartTime": "2026-02-12T20:42:08Z",
        "package": {
          "id": "1397203"
        },
        "startTime": "2026-02-12T20:42:08Z",
        "status": "ENABLED",
        "targets": {
          "actionGroup": {
            "id": "123456"
          },
          "targetGroup": {
            "id": "67890"
          }
        },
        "metadata": []
      },
      "error": null
    }
  }
}
```


If your package requires parameters, pass `params` as part of the `scheduledActionCreate (input.package)` argument.

```
mutation exampleCreateScheduledActionWithParameters($comment: String, $distributeSeconds:
Int, $endTime: Time, $expireSeconds: Int, $name: String, $packageID: ID, $param1: String!,
$param2: String!, $reissueSeconds: Int!, $startTime: Time, $actionGroupID: ID,
$targetGroupID: ID) {
  scheduledActionCreate(
    input: {comment: $comment, schedule: {distributeSeconds: $distributeSeconds, endTime:
$endTime, expireSeconds: $expireSeconds, reissueSeconds: $reissueSeconds, startTime:
$startTime}, name: $name, package: {id: $packageID, params: [$param1, $param2]}, targets:
{actionGroup: {id: $actionGroupID}, targetGroup: {id: $targetGroupID}}}
```

©2026 Tanium Inc. All Rights Reserved

```
  ) {
    scheduledAction {
      approved
      approver {
        persona {
          id
        }
        user {
          id
        }
      }
      comment
      creator {
        persona {
          id
        }
        user {
          id
        }
      }
      distributeSeconds
      endTime
      expireSeconds
      id
      issueCount
      issueSeconds
      lastAction {
        id
      }
      lastStartTime
      name
      nextStartTime
      package {
        id
      }
      startTime
      status
      targets {
        actionGroup {
          id
        }
        targetGroup {
          id
        }
      }
    }
    error {
      message
    }
  }
}
```

Include the action-related variables in the **Variables** panel or in your variables dictionary:

```
{
  "comment": "Example scheduled action",
  "distributeSeconds": 60,
  "endTime": "2026-02-14T00:00:00+00:00",
  "expireSeconds": 1200,
  "name": "Example scheduled action",
  "packageID": 12345,
  "param1": "example_value_1",
```

©2026 Tanium Inc. All Rights Reserved

```
  "param2": "example_value_2",
  "reissueSeconds": 60,
  "startTime": "2026-02-12T00:00:00+00:00",
  "targetGroupID": 67890,
  "actionGroupID": 123456
}
```

Delete action or scheduled action

(mutation.scheduledActionDelete)

The following mutation deletes an action or scheduled action based on the ID.

```
mutation exampleDeleteScheduledAction($id: ID!) {
  scheduledActionDelete(ref: { id: $id }) {
    id
    error {
      message
    }
  }
}
```

Include the action ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345"
}
```

Example response:

```
{
  "data": {
    "scheduledActionDelete": {
      "id": "12345",
      "error": null
    }
  }
}
```

Get action information

(query.action)

The following query retrieves information about an action that matches the ID.

```
query exampleGetAction($id: ID!) {
  action(ref: {id: $id}) {
    approver {
      persona {
        id
        name
      }
      user {
        id
```

©2026 Tanium Inc. All Rights Reserved

```
      name
    }
  }
  comment
  creationTime
  creator {
    persona {
      id
      name
    }
    user {
      id
      name
    }
  }
  distributeSeconds
  expirationTime
  expireSeconds
  id
  name
  package {
    id
    name
    params
    sensorSourcedParams {
      name
      value
    }
  }
  results {
    completed
    downloading
    expired
    failed
    failedVerification
    id
    pendingVerification
    running
    verified
    waiting
    waitingToRetry
  }
  scheduledAction {
    approved
    comment
    distributeSeconds
    endTime
    expireSeconds
    id
    issueCount
    issueSeconds
    lastAction {
      id
    }
    lastStartTime
    name
    nextStartTime
    package {
      id
    }
    startTime
    status
  }
  startTime
```

©2026 Tanium Inc. All Rights Reserved

```
    status
    stopped
    stoppedFlag
  }
}
```

Include the action ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345"
}
```

Example response:

```
{
  "data": {
    "action": {
      "approver": {
        "persona": null,
        "user": {
          "id": "1",
          "name": "Admin"
        }
      },
      "comment": "",
      "creationTime": "2022-07-10T09:40:20Z",
      "creator": {
        "persona": null,
        "user": {
          "id": "1",
          "name": "Admin"
        }
      },
      "distributeSeconds": 1800,
      "expirationTime": "2022-07-10T10:10:12Z",
      "expireSeconds": 1800,
      "id": "12345",
      "name": "Example action",
      "package": {
        "id": "23456",
        "name": "Example package",
        "params": null,
        "sensorSourcedParams": [
          {
            "name": "Example parameter",
            "value": "example_value"
          }
        ]
      },
      "results": {
        "completed": 4,
        "downloading": 0,
        "expired": 0,
        "failed": 0,
        "failedVerification": 0,
        "id": "",
        "pendingVerification": 0,
        "running": 0,
        "verified": 0,
        "waiting": 0,
        "waitingToRetry": 0
```

©2026 Tanium Inc. All Rights Reserved

```
      },
      "scheduledAction": {
        "approved": true,
        "comment": "",
        "distributeSeconds": 1800,
        "endTime": null,
        "expireSeconds": 1800,
        "id": "34567",
        "issueCount": 146,
        "issueSeconds": 3600,
        "lastAction": {
          "id": "45678"
        },
        "lastStartTime": "2022-07-14T17:40:12Z",
        "name": "Example scheduled action",
        "nextStartTime": "2022-07-14T18:40:12Z",
        "package": {
          "id": "56789"
        },
        "startTime": "2022-07-08T16:40:12Z",
        "status": "ENABLED"
      },
      "startTime": "2022-07-10T09:40:12Z",
      "status": null,
      "stopped": false,
      "stoppedFlag": false
    }
  }
}
```

Get multiple actions

(query.actions)



Note: Requests using `query.actions` support up to 200 records returned per page.

The following query retrieves the first `2` actions.

```
query exampleGetMultipleActions($first: Int) {
  actions(first: $first) {
    edges {
      node {
        approver {
          persona {
            id
            name
          }
          user {
            id
            name
          }
        }
        comment
        creationTime
        creator {
```

©2026 Tanium Inc. All Rights Reserved

```
        persona {
          id
          name
        }
        user {
          id
          name
        }
      }
      distributeSeconds
      expirationTime
      expireSeconds
      id
      name
      package {
        id
        name
        params
        sensorSourcedParams {
          name
          value
        }
      }
      results {
        completed
        downloading
        expired
        failed
        failedVerification
        id
        pendingVerification
        running
        verified
        waiting
        waitingToRetry
      }
      scheduledAction {
        approved
        comment
        distributeSeconds
        endTime
        expireSeconds
        id
        issueCount
        issueSeconds
        lastAction {
          id
        }
        lastStartTime
        name
        nextStartTime
        package {
          id
        }
        startTime
        status
      }
      startTime
      status
      stopped
      stoppedFlag
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:


```
{
  "data": {
    "actions": {
      "edges": [
        {
          "node": {
            "approver": {
              "persona": null,
              "user": {
                "id": "123",
                "name": "example user"
              }
            },
            "comment": "",
            "creationTime": "2023-12-22T07:10:04Z",
            "creator": {
              "persona": null,
              "user": {
                "id": "123",
                "name": "example user"
              }
            },
            "distributeSeconds": 60,
            "expirationTime": "2023-12-22T08:10:04Z",
            "expireSeconds": 3600,
            "id": "1234",
            "name": "Example action",
            "package": {
              "id": "321",
              "name": "Example package",
              "params": null,
              "sensorSourcedParams": null
            },
            "results": {
              "completed": 0,
              "downloading": 0,
              "expired": 0,
              "failed": 0,
              "failedVerification": 0,
              "id": "",
              "pendingVerification": 0,
              "running": 0,
              "verified": 0,
              "waiting": 0,
              "waitingToRetry": 0
            },
            "scheduledAction": {
              "approved": true,
              "comment": "",
              "distributeSeconds": 60,
              "endTime": null,
```


©2026 Tanium Inc. All Rights Reserved

```
            "expireSeconds": 3600,
            "id": "1",
            "issueCount": 10,
            "issueSeconds": 3600,
            "lastAction": {
              "id": "329"
            },
            "lastStartTime": "2023-12-22T16:10:04Z",
            "name": "Additional example action",
            "nextStartTime": "2023-12-22T17:10:04Z",
            "package": {
              "id": "432"
            },
            "startTime": "2023-12-22T07:10:04Z",
            "status": "ENABLED"
          },
          "startTime": "2023-12-22T07:10:04Z",
          "status": "CLOSED",
          "stopped": false,
          "stoppedFlag": false
        }
      },
      {
        "node": {
          "approver": {
            "persona": null,
            "user": {
              "id": "123",
              "name": "example user"
            }
          },
          "comment": "",
          "creationTime": "2023-12-22T07:10:06Z",
          "creator": {
            "persona": null,
            "user": {
              "id": "123",
              "name": "example user"
            }
          },
          "distributeSeconds": 60,
          "expirationTime": "2023-12-22T08:10:07Z",
          "expireSeconds": 3600,
          "id": "2345",
          "name": "Additional example action",
          "package": {
            "id": "543",
            "name": "Additional example package",
            "params": null,
            "sensorSourcedParams": null
          },
          "results": {
            "completed": 0,
            "downloading": 0,
            "expired": 0,
            "failed": 0,
            "failedVerification": 0,
            "id": "",
            "pendingVerification": 0,
            "running": 0,
            "verified": 0,
            "waiting": 0,
            "waitingToRetry": 0
          },
```


©2026 Tanium Inc. All Rights Reserved

```
            "scheduledAction": {
              "approved": true,
              "comment": "",
              "distributeSeconds": 60,
              "endTime": null,
              "expireSeconds": 3600,
              "id": "2",
              "issueCount": 10,
              "issueSeconds": 3600,
              "lastAction": {
                "id": "654"
              },
              "lastStartTime": "2023-12-22T16:10:07Z",
              "name": "Additional scheduled action",
              "nextStartTime": "2023-12-22T17:10:07Z",
              "package": {
                "id": "184"
              },
              "startTime": "2023-12-22T07:10:07Z",
              "status": "ENABLED"
            },
            "startTime": "2023-12-22T07:10:07Z",
            "status": "CLOSED",
            "stopped": false,
            "stoppedFlag": false
          }
        }
      ]
    }
  }
}
```

Get package spec information

(query.packageSpecs)

The following query retrieves the first `2` package spec objects.

```
query exampleGetPackageSpecs($first: Int) {
  packageSpecs(first: $first) {
    edges {
      node {
        id
        name
        contentSetName
        command
        commandTimeoutSeconds
        expireSeconds
        lastUpdatedBy
        params {
          key
          label
          defaultValue
        }
        sensorSourcedParams {
          name
        }
      }
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "packageSpecs": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "Example package 1",
            "contentSetName": "Base",
            "command": "cmd.exe /d /c $1",
            "commandTimeoutSeconds": 60,
            "expireSeconds": 660,
            "lastUpdatedBy": "admin",
            "params": [
              {
                "key": "$1",
                "label": "Example parameter",
                "defaultValue": null
              }
            ],
            "sensorSourcedParams": []
          }
        },
        {
          "node": {
            "id": "23456",
            "name": "Example package 2",
            "contentSetName": "Base",
            "command": "cmd.exe /d /c ||Example sensor||",
            "commandTimeoutSeconds": 60,
            "expireSeconds": 660,
            "lastUpdatedBy": "admin",
            "params": [],
            "sensorSourcedParams": [
              {
                "name": "Example sensor"
              }
            ]
          }
        }
      ]
    }
  }
}
```

Get parameters for a specific package spec

(query.packageSpecs)

©2026 Tanium Inc. All Rights Reserved

The following query retrieves the package spec object matching ID `123`, including associated parameter information.

```
query exampleGetPackageParametersByID($filter: FieldFilter) {
  packageSpecs(filter: $filter) {
    edges {
      node {
        id
        name
        params {
          key
          label
          defaultValue
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "path": "id",
    "op": "EQ",
    "value": "123"
  }
}
```

Example response:

```
{
  "data": {
    "packageSpecs": {
      "edges": [
        {
          "node": {
            "id": "123",
            "name": "Custom Tagging - Add Tags",
            "params": [
              {
                "key": "$1",
                "label": "Add tags",
                "defaultValue": null
              }
            ]
          }
        }
      ]
    }
  }
}
```

Get scheduled action information

(query.scheduledAction)

©2026 Tanium Inc. All Rights Reserved

The following query retrieves information about a scheduled action that matches the ID.

```
query exampleGetScheduledAction($id: ID!) {
  scheduledAction(ref: {id: $id}) {
    approver {
      persona {
        id
        name
      }
      user {
        id
        name
      }
    }
    comment
    creator {
      persona {
        id
        name
      }
      user {
        id
        name
      }
    }
    distributeSeconds
    expireSeconds
    id
    lastAction {
      comment
      id
      name
      results {
        completed
        downloading
        expired
        failed
      }
      startTime
    }
    name
    package {
      id
      name
      params
      sensorSourcedParams {
        name
        value
      }
    }
    startTime
    status
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345"
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "scheduledAction": {
      "approver": {
        "persona": null,
        "user": {
          "id": "1",
          "name": "Admin"
        }
      },
      "comment": "Example scheduled action",
      "creator": {
        "persona": null,
        "user": {
          "id": "1",
          "name": "Admin"
        }
      },
      "distributeSeconds": 60,
      "expireSeconds": 1200,
      "id": "12345",
      "lastAction": {
        "comment": "",
        "id": "56789",
        "name": "Example scheduled action",
        "results": {
          "completed": 18,
          "downloading": 0,
          "expired": 0,
          "failed": 0
        },
        "startTime": "2022-07-29T04:34:35Z"
      },
      "name": "Example scheduled action",
      "package": {
        "id": "23456",
        "name": "Example package",
        "params": null,
        "sensorSourcedParams": [
          {
            "name": "Example parameter",
            "value": "example_value"
          }
        ]
      },
      "startTime": "2022-07-17T14:05:35Z",
      "status": "ENABLED"
    }
  }
}
```


Get scheduled actions (query.scheduledActions)

GET MULTIPLE SCHEDULED ACTIONS

The following query retrieves the first `2` scheduled actions.

©2026 Tanium Inc. All Rights Reserved

```
query exampleGetScheduledActions($first: Int) {
  scheduledActions(first: $first) {
    edges {
      node {
        approved
        approver {
          user {
            name
          }
        }
        comment
        creator {
          user {
            name
          }
        }
        distributeSeconds
        endTime
        expireSeconds
        id
        issueCount
        issueSeconds
        lastAction {
          id
        }
        lastStartTime
        name
        nextStartTime
        package {
          id
          name
          params
          sensorSourcedParams {
            name
            value
          }
        }
        startTime
        status
        targets {
          actionGroup {
            id
            name
          }
          targetGroup {
            id
            name
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

©2026 Tanium Inc. All Rights Reserved

```
{
  "data": {
    "scheduledActions": {
      "edges": [
        {
          "node": {
            "approved": true,
            "approver": {
              "user": {
                "name": "Admin"
              }
            },
            "comment": "Example scheduled action",
            "creator": {
              "user": {
                "name": "Admin"
              }
            },
            "distributeSeconds": 1800,
            "endTime": null,
            "expireSeconds": 1800,
            "id": "12345",
            "issueCount": 358,
            "issueSeconds": 3600,
            "lastAction": {
              "id": "23456"
            },
            "lastStartTime": "2022-08-10T14:26:25Z",
            "name": "Example scheduled action",
            "nextStartTime": "2022-08-10T15:26:25Z",
            "package": {
              "id": "34567",
              "name": "Example package",
              "params": [
                "1"
              ],
              "sensorSourcedParams": null
            },
            "startTime": "2022-07-26T17:26:25Z",
            "status": "ENABLED",
            "targets": {
              "actionGroup": {
                "id": "45678",
                "name": "Example action group"
              },
              "targetGroup": {
                "id": "56789",
                "name": "Example target group"
              }
            }
          }
        },
        {
          "node": {
            "approved": true,
            "approver": {
              "user": {
                "name": "Admin"
              }
            },
            "comment": "Example scheduled action 2",
            "creator": {
              "user": {
                "name": "Admin"
```


©2026 Tanium Inc. All Rights Reserved

```
            }
          },
          "distributeSeconds": 1800,
          "endTime": null,
          "expireSeconds": 2100,
          "id": "98765",
          "issueCount": 191,
          "issueSeconds": 5400,
          "lastAction": {
            "id": "87654"
          },
          "lastStartTime": "2022-08-10T14:27:15Z",
          "name": "Example scheduled action 2",
          "nextStartTime": "2022-08-10T15:57:15Z",
          "package": {
            "id": "76543",
            "name": "Example package 2",
            "params": null,
            "sensorSourcedParams": null
          },
          "startTime": "2022-07-29T18:57:15Z",
          "status": "ENABLED",
          "targets": {
            "actionGroup": {
              "id": "65432",
              "name": "Example action group 2"
            },
            "targetGroup": {
              "id": "54321",
              "name": "Example target group 2"
            }
          }
        }
      }
    ]
  }
 }
}
```

GET SCHEDULED ACTIONS (FILTER BY NAME)

The following query retrieves the scheduled action with a name matching `Example scheduled action`.

```
query exampleGetScheduledActionsFilteredByName($path: String, $op: FieldFilterOp!, $value:
String) {
  scheduledActions(filter: {path: $path, op: $op, value: $value}) {
    edges {
      node {
        approved
        approver {
          user {
            name
          }
        }
        comment
        creator {
          user {
            name
```

```
          }
        }
        distributeSeconds
        endTime
        expireSeconds
        id
        issueCount
        issueSeconds
        lastAction {
          id
        }
        lastStartTime
        name
        nextStartTime
        package {
          id
          name
          params
          sensorSourcedParams {
            name
            value
          }
        }
        startTime
        status
        targets {
          actionGroup {
            id
            name
          }
          targetGroup {
            id
            name
          }
        }
      }
    }
  }
}
```

Include the filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "path": "name",
  "op": EQ,
  "value": "Example scheduled action"
}
```

Example response:

```
{
  "data": {
    "scheduledActions": {
      "edges": [
        {
          "node": {
            "approved": true,
            "approver": {
              "user": {
                "name": "Admin"
              }
```

```
            },
            "comment": "Example scheduled action",
            "creator": {
              "user": {
                "name": "Admin"
              }
            },
            "distributeSeconds": 1800,
            "endTime": null,
            "expireSeconds": 1800,
            "id": "12345",
            "issueCount": 358,
            "issueSeconds": 3600,
            "lastAction": {
              "id": "23456"
            },
            "lastStartTime": "2022-08-10T14:26:25Z",
            "name": "Example scheduled action",
            "nextStartTime": "2022-08-10T15:26:25Z",
            "package": {
              "id": "34567",
              "name": "Example package",
              "params": [
                "1"
              ],
              "sensorSourcedParams": null
            },
            "startTime": "2022-07-26T17:26:25Z",
            "status": "ENABLED",
            "targets": {
              "actionGroup": {
                "id": "45678",
                "name": "Example action group"
              },
              "targetGroup": {
                "id": "56789",
                "name": "Example target group"
              }
            }
          }
        }
      ]
    }
  }
}
```

Perform an action operation

(mutation.actionPerform)

©2026 Tanium Inc. All Rights Reserved

Note: There are multiple mutations that create actions.

- Use `mutation.actionCreate` to deploy a one-time action for either a custom package or for a Tanium package that is not supported by `mutation.actionPerform`. For examples, see Create action on page 132.
- Use `mutation.scheduledActionCreate` to deploy a recurring action for either a custom package or a Tanium package that is not supported by `mutation.actionPerform`. For examples, see Create scheduled action on page 135.
- For all other uses cases, use `mutation.actionPerform`, which can deploy any type of package. This includes if you want to deploy a predefined package, especially if you deploy the package to both Windows and non-Windows endpoints. For examples, see Perform an action operation on page 156.

Note: Each request that calls `mutation.actionPerform` creates a copy of the referenced package with the action parameters, even if a similar package already exists.

The `actionPerform` field allows one of the following enumerated action operations:

- `addTags` - add tags to targeted endpoints
- `changeClientSetting` - update one of the following settings on targeted endpoints:
  - `HotCachePercentage` - the hot cache to cold cache setting percentage
  - `LogVerbosityLevel` - the logging level
  - `RandomSensorDelayInSeconds` - the sensor execution randomization value
  - `StateProtectedFlag` - whether queries are encrypted and protected in certain states
- `collectActiveDirectoryInfo` - collect AD information needed for sensors
- `reboot` - reboot targeted endpoints

©2026 Tanium Inc. All Rights Reserved

- `removeTags` - remove tags from targeted endpoints
- `restartService` - restart a service on targeted endpoints
- `startService` - start a service on targeted endpoints
- `stopService` - stop a service on targeted endpoints

You can configure only one action operation per mutation request, but can target any number of endpoints per request using the `input.targets` argument.

- If you do not define an `actionGroup` argument, the action defaults to targeting the **Default** action group.
- If you do not define a `targetGroup` argument, the action defaults to targeting the **All Computers** computer group.
- If you do not define a `platforms` argument, the action defaults to targeting endpoints, regardless of operating system.
- If you do not define an `endpoints` argument, the action defaults to targeting all endpoints that match all of the `input.targets` arguments.

The following example adds tags to endpoints belonging to the `exampleActionGroup` action group and `exampleComputers` computer group, running Solaris operating systems.

```
mutation examplePerformActionOperation($comment: String, $name: String, $addTags: String!,
$endTime: Time, $expireSeconds: Int, $distributeSeconds: Int, $reissueSeconds: Int,
$actionGroupName: String, $targetGroupName: String, $platform: EndpointPlatform!) {
  actionPerform(
    input: {comment: $comment, name: $name, operation: {addTags: [$addTags]}, schedule:
{endTime: $endTime, expireSeconds: $expireSeconds, distributeSeconds: $distributeSeconds,
reissueSeconds: $reissueSeconds}, targets: {actionGroup: {name: $actionGroupName},
targetGroup: {name: $targetGroupName}, platforms: [$platform]}}
  ) {
    scheduledActions {
      platforms
      scheduledAction {
        id
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the action operation variables in the **Variables** panel or in your variables dictionary:

```
{
  "comment": "Example action perform",
  "name": "Example action perform",
  "addTags": "example-tag",
  "endTime": "2022-08-01T14:59:35+00:00",
  "expireSeconds": 660,
  "distributeSeconds": 10,
  "reissueSeconds": 700,
  "actionGroupName": "exampleActionGroup",
  "targetGroupName": "exampleComputers",
  "platform": "Solaris"
}
```

This example response includes a scheduled action ID (`data.actionPerform.scheduledActions.scheduledAction.id`). Use the `query.scheduledAction` request and pass the scheduled action ID in the `ref` argument to retrieve information about the scheduled action. For more information, see the example syntax of Get scheduled action information on page 149.

```
{
  "data": {
    "actionPerform": {
      "scheduledActions": [
        {
          "platforms": [
            "Solaris"
          ],
          "scheduledAction": {
            "id": "12345"
          }
        }
      ],
      "error": null
    }
  }
}
```

The following example shows how to tag metadata through `mutation.actionPerform`.

```
mutation exampleActionPerformWithMetadata($input: ActionPerformInput!) {
  actionPerform(input: $input) {
    scheduledActions {
      platforms
      scheduledAction {
        id
        name
        targets {
          actionGroup {
            id
            name
          }
          targetGroup {
            id
            name
          }
```

©2026 Tanium Inc. All Rights Reserved

```
        }
        package {
          id
          name
        }
        metadata {
          name
          value
        }
        lastAction {
          id
          metadata {
            name
            value
          }
        }
      }
    }
    error {
      message
    }
  }
}
```

Include the action operation variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "Recurring tag action",
    "package": {
      "name": "Custom Tagging - Add Tags (Non-Windows)",
      "params": ["tagA tagB"]
    },
    "targets": { "targetGroup": { "name": "All Linux" } },
    "schedule": { "reissueSeconds": 7200 },
    "metadata": [
      { "name": "Metadata A", "value": "Value A" },
      { "name": "Metadata B", "value": "Value B" }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "actionPerform": {
      "scheduledActions": [
        {
          "platforms": null,
          "scheduledAction": {
            "id": "39",
            "name": "Recurring tag action",
            "targets": {
              "actionGroup": {
                "id": "3",
                "name": "Default"
              },
              "targetGroup": {
                "id": "1381",
                "name": ""
```

©2026 Tanium Inc. All Rights Reserved

```
            }
          },
          "package": {
            "id": "252",
            "name": "Custom Tagging - Add Tags (Non-Windows)"
          },
          "metadata": [
            {
              "name": "Metadata A",
              "value": "Value A"
            },
            {
              "name": "Metadata B",
              "value": "Value B"
            }
          ],
          "lastAction": {
            "id": "277",
            "metadata": [
              {
                "name": "Metadata A",
                "value": "Value A"
              },
              {
                "name": "Metadata B",
                "value": "Value B"
              }
            ]
          }
        }
      }
    ],
    "error": null
  }
 }
}
```


Target endpoints when deploying actions

(query.endpoints.edges.node.name (path.filters))

When you deploy an action through Tanium Gateway, you can determine the endpoints to target with `input.targets.targetGroup`. You can either use a saved computer group or a sensor-based filter. Before you run an action, confirm which endpoints are to be affected. For more information about targeting guidance, see the following:

- Tanium Developer Site: Targeting Best Practices (requires sign-in)

©2026 Tanium Inc. All Rights Reserved

Tip: Currently, there is no dedicated request to preview GraphQL requests to determine which endpoints are targeted. To test, run a read-only endpoints query with a filter that matches the same intent. Check `totalRecords` in the response and verify a few names before you run a mutation to ensure you are targeting the intended endpoints. For an example, see Preview action on page 162.

PREVIEW ACTION

The following query retrieves the first two endpoints whose `name` contains `example`.

```
query examplePreviewEndpointsByName($first: Int, $nameContains: String) {
  endpoints(
    first: $first
    filter: {path: "name", op: CONTAINS, value: $nameContains}
  ) {
    totalRecords
    edges {
      node {
        name
      }
    }
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "nameContains": "example"
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "totalRecords": 236,
      "edges": [
        {
          "node": {
            "name": "example.endpoint.loc"
          }
        },
        {
          "node": {
            "name": "cloud.example.com"
          }
        }
      ]
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

TARGET COMPUTER GROUP TO CREATE SCHEDULED ACTION (SCHEDULEDACTIONCREATE)

The following mutation targets the `All Windows` computer group for a scheduled action.

> **Note:** This example pass two strings in `package.params`. Some packages require `package.params` to have a specific length (for example, exactly one string).

```
mutation exampleTargetComputerGroupForScheduledActionCreate($name: String!, $comment:
String!, $packageName: String!, $packageParams: String!, $targetGroup: String!,
$distributeSeconds: Int!, $reissueSeconds: Int!) {
  scheduledActionCreate(
    input: {name: $name, comment: $comment, package: {name: $packageName, params:
[$packageParams]}, targets: {targetGroup: {name: $targetGroup}}, schedule:
{distributeSeconds: $distributeSeconds, reissueSeconds: $reissueSeconds}}
  ) {
    scheduledAction {
      id
      targets {
        targetGroup {
          id
          name
        }
      }
    }
    error {
      message
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "name": "My example action",
  "comment": "My example comment",
  "packageName": "Custom Tagging - Add Tags",
  "packageParams": "tagA tagB",
  "targetGroup": "All Windows",
  "distributeSeconds": 3600,
  "reissueSeconds": 5000
}
```

Example response:

```
{
  "data": {
    "scheduledActionCreate": {
      "scheduledAction": {
        "id": "123456",
        "targets": {
          "targetGroup": {
            "id": "12345678",
            "name": ""
          }
```

©2026 Tanium Inc. All Rights Reserved

```
        }
      },
      "error": null
    }
  }
}
```



Note: The `targets.targetGroup.name` field in the response might be empty even when the action succeeds. It is normal if the computer group ID from a `computerGroup` query does not match `targets.targetGroup.id`. To confirm a group, run a `computerGroup` query at the root of the schema. For example, `computerGroup(ref: { name: "All Windows" }) { … })`, or use `endpoints(filter: { memberOf: { name: "All Windows" } }).`

PERFORM ACTION WITH SENSOR FILTER (`ACTIONPERFORM`)

The following query uses a sensor filter to create an action that targets endpoints whose name contains `example`.

```
mutation exampleActionPerform($name: String, $comment: String, $tagA: String!, $tagB:
String!, $filterString: String, $sensorName: String) {
  actionPerform(
    input: {name: $name, comment: $comment, operation: {addTags: [$tagA, $tagB]}, targets:
{targetGroup: {filter: {sensor: {name: $sensorName}, op: CONTAINS, value: $filterString,
negated: false, any: false}}}}
  ) {
    scheduledActions {
      scheduledAction {
        id
        targets {
          targetGroup {
            id
            name
          }
        }
      }
    }
    error {
      message
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "name": "My example action",
  "comment": "My example comment",
  "sensorName": "Computer Name",
  "tagA": "tagA",
  "tagB": "tagB",
  "filterString": "example"
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:

```
{
  "data": {
    "actionPerform": {
      "scheduledActions": [
        {
          "scheduledAction": {
            "id": "123456",
            "targets": {
              "targetGroup": {
                "id": "12345678",
                "name": ""
              }
            }
          }
        },
        {
          "scheduledAction": {
            "id": "123457",
            "targets": {
              "targetGroup": {
                "id": "12345680",
                "name": ""
              }
            }
          }
        }
      ],
      "error": null
    }
  }
}
```

SHOW A SCHEDULED ACTION

The following query retrieves a scheduled action.

```
query exampleScheduledAction ($id: ID!) {
  scheduledAction(ref: { id: $id }) {
    id
    name
    status
    targets {
      targetGroup {
        id
        name
      }
    }
    package {
      id
      name
    }
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "id": 12345
}
```

Stop an action

(mutation.actionStop)

The following mutation stops an action that matches the ID.

```
mutation exampleStopAction($id: ID!) {
  actionStop(ref: { id: $id }) {
    id
    error {
      message
    }
  }
}
```

Include the action ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345"
}
```

Example response:

```
{
  "data": {
    "actionStop": {
      "id": "12345",
      "error": null
    }
  }
}
```

DEPRECATED: Create action (subset of endpoints)

(mutation.createAction)

> **Note:** This is deprecated in favor of `mutation.actionPerform` and `mutation.scheduledActionCreate`. For more information, see the Perform an action operation on page 156 and Create scheduled action on page 135.

The following mutation deploys an action to increase the verbosity of log levels on Debian endpoints.

```
mutation exampleCreateActionLogging($description: String, $group: String, $platform:
EndpointPlatform!, $setting: SettingName!, $settingValue: Any!) {
```

©2026 Tanium Inc. All Rights Reserved

```
  createAction(
    action: {description: $description, target: {targetGroup: $group, platforms:
[$platform]}, changeClientSetting: {name: $setting, value: $settingValue}}
  ) {
    id
  }
}
```

Include the action-related variables in the **Variables** panel or in your variables dictionary:

```
{
  "description": "Increasing log verbosity level on all debian endpoints for
troubleshooting",
  "group": "All Debian",
  "platform": "Linux",
  "setting": "LOG_VERBOSITY_LEVEL",
  "settingValue": "41"
}
```

Example response:

```
{
  "data": {
    "createAction": {
      "id": "82"
    }
  }
}
```

DEPRECATED: Get action details

(query.lastActionDetails, query.lastActionResults)

> **Note:** This is deprecated in favor of `query.scheduledAction`. For more information, see the example syntax of Get scheduled action information on page 149.

The following parameterized query retrieves details and any results for an action.

```
query exampleGetActionDetails($id: ID!) {
  lastActionDetails(id: $id) {
    id
    name
    comment
    expireSeconds
    creationTime
    startTime
    expirationTime
    distributeSeconds
    status
    stoppedFlag
  }
  lastActionResults(id: $id) {
    id
```

©2026 Tanium Inc. All Rights Reserved

```
    waiting
    downloading
    running
    waitingToRetry
    completed
    expired
    failed
    pendingVerification
    verified
    failedVerification
  }
}
```

Include the endpoint ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12323"
}
```

Example response:

```
{
  "data": {
    "lastActionDetails": {
      "id": "219",
      "name": "Distribute Tanium Standard Utilities",
      "comment": "Distribute Tanium Standard Utilities",
      "expireSeconds": 3900,
      "creationTime": "2022-03-14T18:05:41Z",
      "startTime": "2022-03-14T18:05:35Z",
      "expirationTime": "2022-03-14T19:10:35Z",
      "distributeSeconds": 3600,
      "status": "OPEN",
      "stoppedFlag": false
    },
    "lastActionResults": {
      "id": "219",
      "waiting": 0,
      "downloading": 0,
      "running": 0,
      "waitingToRetry": 0,
      "completed": 1,
      "expired": 0,
      "failed": 0,
      "pendingVerification": 0,
      "verified": 0,
      "failedVerification": 0
    }
  }
}
```

Action group examples

The following queries and mutations create, retrieve, and delete action groups.

©2026 Tanium Inc. All Rights Reserved

Create action group

(mutation.actionGroupCreate)

The following mutation creates an action group that targets the defined computer group. All user groups can view this action group, and endpoints must be a member of the defined computer group to be targeted by the action group.

```
mutation exampleCreateActionGroup($any: Boolean!, $computerGroupName: String,
$actionGroupName: String!, $visibility: ActionGroupVisibility!) {
  actionGroupCreate(
    input: {any: $any, computerGroups: [{name: $computerGroupName}], name: $actionGroupName,
visibility: $visibility}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the targeting, targeted computer group, action group name, and visibility variables in the **Variables** panel or in your variables dictionary:

```
{
  "any": false,
  "computerGroupName": "Example computer group",
  "actionGroupName": "Example action group",
  "visibility": "ALL"
}
```

Example response:

```
{
  "data": {
    "actionCreate": {
      "action": {
        "id": "12345"
      },
      "error": null
    }
  }
}
```

Delete action group by ID or name

(mutation.actionGroupDelete)

DELETE ACTION GROUP BY ID

©2026 Tanium Inc. All Rights Reserved

The following mutation deletes the action group that matches the ID.

```
mutation exampleDeleteActionGroup($id: ID) {
  actionGroupDelete(ref: {id: $id}) {
    error {
      message
    }
    id
  }
}
```

Include the action group ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345"
}
```

Example response:

```
{
  "data": {
    "actionGroupDelete": {
      "error": null,
      "id": "12345"
    }
  }
}
```

DELETE ACTION GROUP BY NAME

You can also delete an action group that matches a name.

```
mutation exampleDeleteActionGroupByName($name: String) {
  actionGroupDelete(ref: {name: $name}) {
    error {
      message
    }
    id
  }
}
```

Include the action group name variable in the **Variables** panel or in your variables dictionary:

```
{
  "name": "Example action group"
}
```

Get action group by ID or name

(query.actionGroup)

©2026 Tanium Inc. All Rights Reserved

GET ACTION GROUP BY ID

The following query retrieves an action group that matches the ID.

```
query exampleGetActionGroupByID($id: ID!) {
  actionGroup(ref: {id: $id}) {
    any
    computerGroups {
      id
      name
      contentSet {
        id
        name
      }
      expression
      filterEnabled
      managementRightsEnabled
    }
    id
    name
    userGroups {
      id
      name
    }
    visibility
  }
}
```

Include the action group ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345"
}
```

Example response:

```
{
  "data": {
    "actionGroup": {
      "any": false,
      "computerGroups": [
        {
          "id": "23456",
          "name": "Example computer group",
          "contentSet": {
            "id": "1",
            "name": "Reserved"
          },
          "expression": " any Computer Name not matches .*",
          "filterEnabled": true,
          "managementRightsEnabled": true
        }
      ],
      "id": "12345",
      "name": "Example action group",
      "userGroups": null,
      "visibility": "ALL"
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

GET ACTION GROUP BY NAME

You can also retrieve an action group that matches a name.

```
query exampleGetActionGroupByName($name: String!) {
  actionGroup(ref: {name: $name}) {
    any
    computerGroups {
      id
      name
      contentSet {
        id
        name
      }
      expression
      filterEnabled
      managementRightsEnabled
    }
    id
    name
    userGroups {
      id
      name
    }
    visibility
  }
}
```

Include the action group name variable in the **Variables** panel or in your variables dictionary:

```
{
  "name": "Example action group"
}
```

Get multiple action groups

(query.actionGroups)

The following query retrieves the first `2` action groups. For more information about pagination, see Pagination on page 22.

```
query exampleGetActionGroups($first: Int) {
  actionGroups(first: $first) {
    edges {
      node {
        id
        name
        visibility
        userGroups {
          id
          name
        }
```

©2026 Tanium Inc. All Rights Reserved

```
        computerGroups {
          id
          name
          type
          contentSet {
            id
          }
          expression
          filterEnabled
          managementRightsEnabled
        }
        any
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "actionGroups": {
      "edges": [
        {
          "node": {
            "id": "1",
            "name": "Example action group",
            "visibility": "ALL",
            "userGroups": null,
            "computerGroups": [
              {
                "id": "1",
                "name": "Example computer group",
                "type": "STANDARD",
                "contentSet": {
                  "id": "1"
                },
                "expression": "All Computers",
                "filterEnabled": true,
                "managementRightsEnabled": true
              }
            ],
            "any": true
          }
        },
        {
          "node": {
            "id": "2",
            "name": "Example action group 2",
            "visibility": "ALL",
            "userGroups": null,
            "computerGroups": [
              {
                "id": "2",
                "name": "No Computers",
```

©2026 Tanium Inc. All Rights Reserved

```
                "type": "STANDARD",
                "contentSet": {
                  "id": "1"
                },
                "expression": "No Computers",
                "filterEnabled": true,
                "managementRightsEnabled": true
              }
            ],
            "any": true
          }
        }
      ]
    }
  }
}
```

API token examples

The following mutations create, rotate, and delete API tokens.

Create API token

(mutation.apiTokenGrant)

The following mutation creates an API token associated with the IP addresses and persona, valid for 365 days. This requires the **Token - Use** permission.

Best Practice:

Do not use a single token for multiple processes. For example, do not use a single token for a ServiceNow integration and a Splunk integration, or for multiple ServiceNow instances, or for multiple connections from one ServiceNow instance to Tanium Cloudthe Tanium Server. When a token rotates for any process, that token stops working for any other processes that use it.

For additional best practices to use API tokens, see Tanium Console User Guide: Managing API tokens.

Important: Include the `apiTokenGrant.token.tokenString` field in your request, then copy the token string in the response. You cannot view the token string in the response after you navigate to another page or send another request and receive a different response from the gateway service.

©2026 Tanium Inc. All Rights Reserved

```
mutation exampleCreateAPIToken($trustedIPAddress1: String!, $trustedIPAddress2: String!,
$personaName: String!, $expiresInDays: Int!, $notes: String!) {
  apiTokenGrant(
    input: {trustedIPAddresses: [$trustedIPAddress1, $trustedIPAddress2], personaName:
$personaName, expiresInDays: $expiresInDays, notes: $notes}
  ) {
    token {
      created
      expiration
      id
      lastUsed
      notes
      persona {
        name
      }
      tokenString
      trustedIPAddresses
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "trustedIPAddress1": "192.0.2.0/24",
  "trustedIPAddress2": "198.51.100.10",
  "expiresInDays": 365,
  "personaName": "testPersona",
  "notes": "Gateway token"
}
```

Example response:

```
{
  "data": {
    "apiTokenGrant": {
      "token": {
        "created": "2022-05-10T15:22:13Z",
        "expiration": "2023-05-10T15:22:13Z",
        "id": "4029",
        "lastUsed": "2022-05-10T15:22:13Z",
        "notes": "Gateway token",
        "persona": {
          "name": "testPersona"
        },
        "tokenString": "token-9ab934979aea20a3d56a822441a4329f470326f5e7ef2af66783147533",
        "trustedIPAddresses": [
          "192.0.2.0/24",
          "198.51.100.10"
        ]
      },
      "error": null
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Get API tokens for current user

(query.myAPITokens)

The following query retrieves API tokens associated with the active session user. This requires the **Token - Use** permission.



Important: You cannot view API token strings in the response.

```
query exampleGetMyAPITokens {
  myAPITokens {
    tokens {
      created
      expiration
      id
      lastUsed
      notes
      persona {
        name
      }
      trustedIPAddresses
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Example response:

```
{
  "data": {
    "myAPITokens": {
      "tokens": [
        {
          "created": "2022-05-10T17:10:23Z",
          "expiration": "2023-05-10T17:10:23Z",
          "id": "4036",
          "lastUsed": "2022-05-10T17:10:23Z",
          "notes": "Gateway token",
          "persona": null,
          "trustedIPAddresses": [
            "192.0.2.0/24",
            "198.51.100.10"
          ]
        }
      ],
      "error": null
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Rotate API token

(mutation.apiTokenRotate)

The following mutation rotates an existing API token, deleting it and creating a new one with the same properties. This requires the **Token - Use** permission.

Best Practice: If you integrate your Tanium deployment with another platform such as ServiceNow or Splunk, use the Tanium API to configure the integration to automatically rotate its token when token expiration is imminent. Automatic rotation reduces the maintenance workload for an integration.

Important: Include the `apiTokenRotate.token.tokenString` field in your request, then copy the token string in the response. You cannot view the token string in the response after you navigate to another page or send another request and receive a different response from the gateway service.

```
mutation exampleRotateAPIToken($tokenString: String!) {
  apiTokenRotate(input: {tokenString: $tokenString}) {
    token {
      created
      expiration
      id
      lastUsed
      notes
      persona {
        name
      }
      tokenString
      trustedIPAddresses
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the token string variable in the **Variables** panel or in your variables dictionary:

```
{
  "tokenString": "token-9668a9a11e082ea9b062f204f4e9b98e8785d006388e99e00fb6efbaaa"
}
```

Example response:

©2026 Tanium Inc. All Rights Reserved

```
{
  "data": {
    "apiTokenRotate": {
      "token": {
        "created": "2022-05-10T17:14:05Z",
        "expiration": "2023-05-10T17:14:05Z",
        "id": "4038",
        "lastUsed": "2022-05-10T17:14:05Z",
        "notes": "Gateway token",
        "persona": "testPersona",
        "tokenString": "token-9a4b38683e26bd6fa19c085b11850e3f5aec96cd89e4a0e6a0b8fdb2c6",
        "trustedIPAddresses": [
          "192.0.2.0/24",
          "198.51.100.10"
        ]
      },
      "error": null
    }
  }
}
```

Delete API token

(mutation.apiTokenRevoke)

The following mutation deletes an existing API token. This requires the **Token - Revoke** permission.

```
mutation exampleDeleteAPIToken($id: ID!) {
  apiTokenRevoke(input: {id: $id}) {
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the token ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "10"
}
```

Example response:

```
{
  "data": {
    "apiTokenRevoke": {
      "error": null
    }
  }
}
```

Asset examples

©2026 Tanium Inc. All Rights Reserved

The following requests require Asset and retrieve Asset products or endpoints based on Asset products, update Asset product tracking, or import assets to update existing assets or create new assets.

Asset requests have the following feature-specific dependencies at the specified minimum versions:

- Asset 1.23.8 or later is required to submit a mutation request to import assets (`mutation.assetsImport`).
- Asset 1.24.76 or later is required to submit a request that retrieves Asset products (`query.assetProducts`) or endpoints based on installed Asset products (`query.assetProductEndpoints`).
- Asset 1.33.181 or later is required to submit a request that upserts (creates or updates by ID) endpoints (`mutation.assetUpsertEndpoints`), update endpoints based on Asset source (`mutation.assetUpdateEndpoints`), or update endpoints based endpoint ID (`query.assetUpdateEndpointsUsingEid`).

Get Asset products

query.assetProducts

About this task

The following query requires Asset 1.24.76 or later and retrieves Asset products:

```
query exampleGetAssetProducts($first: Int) {
  assetProducts(
    first: $first
  ) {
    edges {
      node {
        vendor
        name
        installation {
          installedCount
          usedCount
          unusedCount
          pendingUsage
        }
        usage {
          usageNotDetected
          notInstalled
          baselining
          limited
          normal
          high
        }
        tracking {
```

©2026 Tanium Inc. All Rights Reserved

```
          state
          reportingPeriodDays
          normalMinutesUsedPerDay
          highMinutesUsedPerDay
          baselinePeriodDays
        }
        versions {
          version
          installs
        }
      }
    }
  }
}
```

Include the cursor variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "assetProducts": {
      "edges": [
        {
          "node": {
            "vendor": "Example vendor",
            "name": "Example product",
            "installation": {
              "installedCount": 8,
              "usedCount": 0,
              "unusedCount": 0,
              "pendingUsage": null
            },
            "usage": {
              "usageNotDetected": 0,
              "notInstalled": 0,
              "baselining": 0,
              "limited": 0,
              "normal": 0,
              "high": 0
            },
            "tracking": {
              "state": "Cataloged",
              "reportingPeriodDays": 90,
              "normalMinutesUsedPerDay": 60,
              "highMinutesUsedPerDay": 240,
              "baselinePeriodDays": 7
            },
            "versions": [
              {
                "version": "1.2.3",
                "installs": 2
              },
              {
                "version": "1.2.4",
                "installs": 2
              }
```

©2026 Tanium Inc. All Rights Reserved

```
            ]
          }
        },
        {
          "node": {
            "vendor": "Example vendor 2",
            "name": "Example product 2",
            "installation": {
              "installedCount": 1,
              "usedCount": 0,
              "unusedCount": 0,
              "pendingUsage": null
            },
            "usage": {
              "usageNotDetected": 0,
              "notInstalled": 0,
              "baselining": 0,
              "limited": 0,
              "normal": 0,
              "high": 0
            },
            "tracking": {
              "state": "Cataloged",
              "reportingPeriodDays": 90,
              "normalMinutesUsedPerDay": 60,
              "highMinutesUsedPerDay": 240,
              "baselinePeriodDays": 7
            },
            "versions": [
              {
                "version": "0.1.2",
                "installs": 1
              }
            ]
          }
        }
      ]
    }
  }
}
```

The following query request filters and retrieves only Asset products with a vendor name of `Example vendor`, an Asset state of `Cataloged`, and the search string `Product` in either the vendor name or product name:

```
query exampleGetAssetProductsFiltered($first: Int, $vendor: String!, $search: String,
$states: AssetProductState!) {
  assetProducts(
    first: $first, filter: {
      vendors: [$vendor],
      search: $search,
      states: [$states]
    }
  ) {
    edges {
      node {
        vendor
        name
        installation {
          installedCount
          usedCount
```

```
          unusedCount
          pendingUsage
        }
        usage {
          usageNotDetected
          notInstalled
          baselining
          limited
          normal
          high
        }
        tracking {
          state
          reportingPeriodDays
          normalMinutesUsedPerDay
          highMinutesUsedPerDay
          baselinePeriodDays
        }
        versions {
          version
          installs
        }
      }
    }
  }
}
```

Include the cursor and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "vendor": "Example vendor",
  "search": "Product",
  "states": "Cataloged"
}
```

Example response:

```
{
  "data": {
    "assetProducts": {
      "edges": [
        {
          "node": {
            "vendor": "Example vendor",
            "name": "Example product",
            "installation": {
              "installedCount": 8,
              "usedCount": 0,
              "unusedCount": 0,
              "pendingUsage": null
            },
            "usage": {
              "usageNotDetected": 0,
              "notInstalled": 0,
              "baselining": 0,
              "limited": 0,
              "normal": 0,
              "high": 0
            },
            "tracking": {
```

©2026 Tanium Inc. All Rights Reserved

```
              "state": "Cataloged",
              "reportingPeriodDays": 90,
              "normalMinutesUsedPerDay": 60,
              "highMinutesUsedPerDay": 240,
              "baselinePeriodDays": 7
            },
            "versions": [
              {
                "version": "0.1.2",
                "installs": 2
              },
              {
                "version": "3.4.5",
                "installs": 2
              }
            ]
          }
        },
        {
          "node": {
            "vendor": "Example vendor",
            "name": "Product example",
            "installation": {
              "installedCount": 1,
              "usedCount": 0,
              "unusedCount": 0,
              "pendingUsage": null
            },
            "usage": {
              "usageNotDetected": 0,
              "notInstalled": 0,
              "baselining": 0,
              "limited": 0,
              "normal": 0,
              "high": 0
            },
            "tracking": {
              "state": "Cataloged",
              "reportingPeriodDays": 90,
              "normalMinutesUsedPerDay": 60,
              "highMinutesUsedPerDay": 240,
              "baselinePeriodDays": 7
            },
            "versions": [
              {
                "version": "6.7.8",
                "installs": 1
              }
            ]
          }
        }
      ]
    }
  }
}
```


Get endpoints based on Asset products

query.assetProductEndpoints

©2026 Tanium Inc. All Rights Reserved

About this task

The following query requires Asset 1.24.76 or later and retrieves the first 2 endpoints with the `Example name` product installed, associated with the `Example vendor` vendor and version `1.2.3`:

```
query exampleGetEndpointsAssetProduct($first: Int, $vendor: String, $name: String, $version:
String) {
  assetProductEndpoints(
    first: $first
    filter: {vendor: $vendor, name: $name, version: $version}
  ) {
    edges {
      node {
        id
        eid
        computerName
        computerId
        serialNumber
        osPlatform
        operatingSystem
        servicePack
        manufacturer
        ipAddress
        userName
        createdAt
        updatedAt
      }
    }
  }
}
```

Include the cursor and filtering variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "vendor": "Example vendor",
  "name": "Example name",
  "version": "1.2.3"
}
```

Example response:

```
{
  "data": {
    "assetProductEndpoints": {
      "edges": [
        {
          "node": {
            "id": 123,
            "eid": 456789,
            "computerName": "example-host-1",
            "computerId": "123456789",
            "serialNumber": "1234567890abcdef",
            "osPlatform": "Windows",
            "operatingSystem": "Windows 10 Enterprise",
            "servicePack": "No Service Pack found",
            "manufacturer": "VMware, Inc.",
```

©2026 Tanium Inc. All Rights Reserved

```
            "ipAddress": "192.0.2.10",
            "userName": "example-user",
            "createdAt": "2022-09-29T07:00:16.000Z",
            "updatedAt": "2022-11-10T23:00:06.000Z"
          }
        },
        {
          "node": {
            "id": 456,
            "eid": 789012,
            "computerName": "example-host-2",
            "computerId": "987654321",
            "serialNumber": "fedcba0987654321",
            "osPlatform": "Windows",
            "operatingSystem": "Windows 10 Enterprise",
            "servicePack": "No Service Pack found",
            "manufacturer": "VMware, Inc.",
            "ipAddress": "192.0.2.20",
            "userName": "example-user",
            "createdAt": "2022-09-29T07:00:16.000Z",
            "updatedAt": "2022-10-31T14:00:05.000Z"
          }
        }
      ]
    }
  }
}
```


Update Asset endpoints based on source

mutation.assetUpdateEndpoints

About this task

Stability: 1.2 - Experimental (Release Candidate)

The following mutation requires Asset 1.33.181 or later and updates the endpoint record based on the default Tanium source. If reconciliation keys are included, asset reconciliation is used. If no endpoints match the keys, the response returns an error.

Note: To update an endpoint in the Tanium source, you must specify the `Computer Name`, `Computer ID`, and `Endpoint Fingerprint` keys. Asset finds the endpoint if at least two of the keys match.

Note: After you submit a successful request, it can take up to two import cycles, or approximately 20 minutes, for attributes to appear in Asset and in reports.

©2026 Tanium Inc. All Rights Reserved

```
mutation exampleAssetUpdateEndpointsTaniumSource($input: AssetUpdateEndpointsInput!) {
  assetUpdateEndpoints(
    input: $input
  ) {
    assets {
      eid
      source {
        namespace
        id
        name
      }
      status
      error
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "sourceName": "Tanium",
    "entityNames": [
      "Computer Name",
      "Computer ID",
      "Endpoint Fingerprint",
      "Asset Custom: custom_entity_1"
    ],
    "assets": [
      {
        "entities": [
          {
            "entityName": "Computer Name",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Computer Name",
                    "value": "computer1"
                  }
                ]
              }
            ]
          },
          {
            "entityName": "Computer ID",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Computer ID",
                    "value": "1234567890"
                  }
                ]
              }
            ]
          },
          {
            "entityName": "Endpoint Fingerprint",
            "entityRowItems": [
              {
                "attributes": [
```

©2026 Tanium Inc. All Rights Reserved

```
                  {
                    "attributeName": "Endpoint Fingerprint",
                    "value":
"1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
                  }
                ]
              }
            ]
          },
          {
            "entityName": "Asset Custom: custom_entity_1",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Asset Custom: custom_entity_1",
                    "value": "example_value"
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "assetUpdateEndpoints": {
      "assets": [
        {
          "eid": "12345",
          "source": {
            "namespace": "Asset: Tanium",
            "id": "1",
            "name": "Tanium"
          },
          "status": "Updated",
          "error": null
        }
      ]
    }
  }
}
```

The following mutation updates the endpoint record based on a Microsoft Azure source:

Note: To update an endpoint in the Tanium source, you must specify the `Computer Name`, `Computer ID`, and `Endpoint Fingerprint` keys. Asset finds the endpoint if at least two of the keys match. For an example, see Update Asset endpoint by Asset source on page 185.

```
mutation exampleAssetUpdateEndpointsMicrosoftAzureSource($input: AssetUpdateEndpointsInput!)
```

©2026 Tanium Inc. All Rights Reserved

```
{
  assetUpdateEndpoints(
    input: $input
  ) {
    assets {
      eid
      source {
        namespace
        id
        name
      }
      status
      error
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "sourceName": "example Microsoft Azure source",
    "entityNames": [
      "Cloud Instance ID",
      "Asset Custom: custom_entity_1"
    ],
    "assets": [
      {
        "entities": [
          {
            "entityName": "Cloud Instance ID",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Cloud Instance ID",
                    "value": "12345678-90ab-cdef-1234-567890abcdef"
                  }
                ]
              }
            ]
          },
          {
            "entityName": "Asset Custom: custom_entity_1",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Asset Custom: custom_entity_1",
                    "value": "Test 1"
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:

```
{
  "data": {
    "assetUpdateEndpoints": {
      "assets": [
        {
          "eid": "149157",
          "source": {
            "namespace": "Asset: example Microsoft Azure source",
            "id": "102",
            "name": "example Microsoft Azure source"
          },
          "status": "Updated",
          "error": null
        }
      ]
    }
  }
}
```

Update Asset endpoints based on endpoint ID (EID)

mutation.assetUpdateEndpointsUsingEid

About this task

Stability: 1.2 - Experimental (Release Candidate)

The following mutation requires Asset 1.33.181 or later and updates the endpoint record for an Asset source that matches the endpoint ID (EID). If reconciliation keys are included, asset reconciliation is used. If no endpoints match the keys, the response returns an error.

Note: To update an endpoint in the Tanium source, you must specify the `Computer Name`, `Computer ID`, and `Endpoint Fingerprint` keys. Asset finds the endpoint if at least two of the keys match. For an example, see Update Asset endpoint by Asset source on page 185.

Note: After you submit a successful request, it can take up to two import cycles, or approximately 20 minutes, for attributes to appear in Asset and in reports.

```
mutation exampleAssetUpdateEndpointsEID($input: AssetUpdateEndpointsUsingEidInput!) {
  assetUpdateEndpointsUsingEid(input: $input) {
    assets {
      eid
      source {
        namespace
```

©2026 Tanium Inc. All Rights Reserved

```
        id
        name
      }
      status
      error
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "sourceName": "example_source",
    "entityNames": [
      "Asset Custom: custom_entity_1"
    ],
    "assets": [
      {
        "eid": "12345",
        "entities": [
          {
            "entityName": "Asset Custom: custom_entity_1",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Asset Custom: custom_entity_1",
                    "value": "example_test"
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "assetUpdateEndpointsUsingEid": {
      "assets": [
        {
          "eid": "12345",
          "source": {
            "namespace": "Asset: example_source",
            "id": "123",
            "name": "example_source"
          },
          "status": "Updated",
          "error": null
        }
      ]
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Upsert (create or update by ID) Asset endpoint

mutation.assetUpsertEndpoints

About this task

Stability: 1.2 - Experimental (Release Candidate)

The `assetUpsertEndpoints` mutation requires Asset 1.33.181 or later and either creates a new endpoint record or updates an existing endpoint record using the defined `sourceName` for Import API sources. Asset creation or update is based on comparing key field values. If reconciliation keys are included, asset reconciliation is used. To determine whether an existing asset updates, or a new asset is created:

1. If an imported asset key matches an existing asset key imported using the `sourceName` Import API source, that existing asset updates. The status in the response equals `Updated`.

2. If an imported asset key matches an existing asset reconciliation key imported using the Tanium source only, that existing asset is reconciled, and then updates. The status in the response equals `Reconciled`.

3. Otherwise, if no existing asset matches the imported asset, the asset is imported as a new asset using the `sourceName` Import API source. The status in the response is equal to `Created`.

Note: If the request is structurally and syntactically valid, each asset object imports separately. A failed asset import does not prevent other valid asset objects in that request from importing.

Note: After you submit a successful request, it can take up to two import cycles, or approximately 20 minutes, for attributes to appear in Asset and in reports.

The following mutation request creates an asset, if no existing asset in the `example_source` Import API source matches on the `Computer Name` key field, and updates a single `attributeName` value:

```
mutation exampleAssetUpsertEndpoints($input: AssetUpsertEndpointsInput!) {
```

©2026 Tanium Inc. All Rights Reserved

```
  assetUpsertEndpoints(input: $input) {
    assets {
      eid
      source {
        namespace
        id
        name
      }
      status
      error
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "sourceName": "example_source",
    "entityNames": [
      "Computer Name",
      "Asset Custom: custom_entity_1"
    ],
    "assets": [
      {
        "entities": [
          {
            "entityName": "Computer Name",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Computer Name",
                    "value": "computer1"
                  }
                ]
              }
            ]
          },
          {
            "entityName": "Asset Custom: custom_entity_1",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Asset Custom: custom_entity_1",
                    "value": "Test 1"
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```

As the request is evaluated, the asset object defines a value for the `computerName` key field, matching the key field defined for the `example_source` source, and is imported using that source.

©2026 Tanium Inc. All Rights Reserved

The example response shows a status of `Created`:

```
{
  "data": {
    "assetUpsertEndpoints": {
      "assets": [
        {
          "eid": "12345",
          "source": {
            "namespace": "Asset: example_source",
            "id": "123",
            "name": "example_source"
          },
          "status": "Created",
          "error": null
        }
      ]
    }
  }
}
```

The following mutation request updates an asset if an asset in the `example_source` Import API source matches on the `Computer Name` key field, and updates two `attributeName` values:

```
mutation exampleAssetUpsertEndpointsMultiColumn($input: AssetUpsertEndpointsInput!) {
  assetUpsertEndpoints(input: $input) {
    assets {
      eid
      source {
        namespace
        id
        name
      }
      status
      error
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "sourceName": "example_source",
    "entityNames": [
      "Computer Name",
      "Asset Custom: custom_entity_2"
    ],
    "assets": [
      {
        "entities": [
          {
            "entityName": "Computer Name",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Computer Name",
                    "value": "computer1"
```

©2026 Tanium Inc. All Rights Reserved

```
                }
              ]
            }
          ]
        },
        {
          "entityName": "Asset Custom: custom_entity_2",
          "entityRowItems": [
            {
              "attributes": [
                {
                  "attributeName": "column_1",
                  "value": "column_1_test"
                },
                {
                  "attributeName": "column_2",
                  "value": "column_2_test"
                }
              ]
            }
          ]
        }
      ]
    }
  }
}
```

As the request is evaluated, the asset object defines a value for the `computerName` key field, matching the key field defined for the `example_source` Import API source, and is imported using that source. The example response shows a status of `Updated`:

```
{
  "data": {
    "assetUpsertEndpoints": {
      "assets": [
        {
          "eid": "12345",
          "source": {
            "namespace": "Asset: example_source",
            "id": "123",
            "name": "example_source"
          },
          "status": "Updated",
          "error": null
        }
      ]
    }
  }
}
```

The following mutation request updates an asset if an asset in the `example_source` Import API source matches on the `Computer Name` key field, and updates two rows with two `attributeName` values:

```
mutation exampleAssetUpsertEndpointsMultiRow($input: AssetUpsertEndpointsInput!) {
  assetUpsertEndpoints(input: $input) {
    assets {
```

©2026 Tanium Inc. All Rights Reserved

```
      eid
      source {
        namespace
        id
        name
      }
      status
      error
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "sourceName": "ktai_example_custom_source",
    "entityNames": [
      "Computer Name",
      "Asset Custom: ktai_custom_entity_3"
    ],
    "assets": [
      {
        "entities": [
          {
            "entityName": "Computer Name",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "Computer Name",
                    "value": "computer1"
                  }
                ]
              }
            ]
          },
          {
            "entityName": "Asset Custom: ktai_custom_entity_3",
            "entityRowItems": [
              {
                "attributes": [
                  {
                    "attributeName": "multi_row_1",
                    "value": "row_1_column_1_test"
                  },
                  {
                    "attributeName": "multi_row_2",
                    "value": "row_1_column_2_test"
                  }
                ]
              },
              {
                "attributes": [
                  {
                    "attributeName": "multi_row_1",
                    "value": "row_2_column_1_test"
                  },
                  {
                    "attributeName": "multi_row_2",
                    "value": "row_2_column_2_test"
                  }
```

©2026 Tanium Inc. All Rights Reserved

```
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```

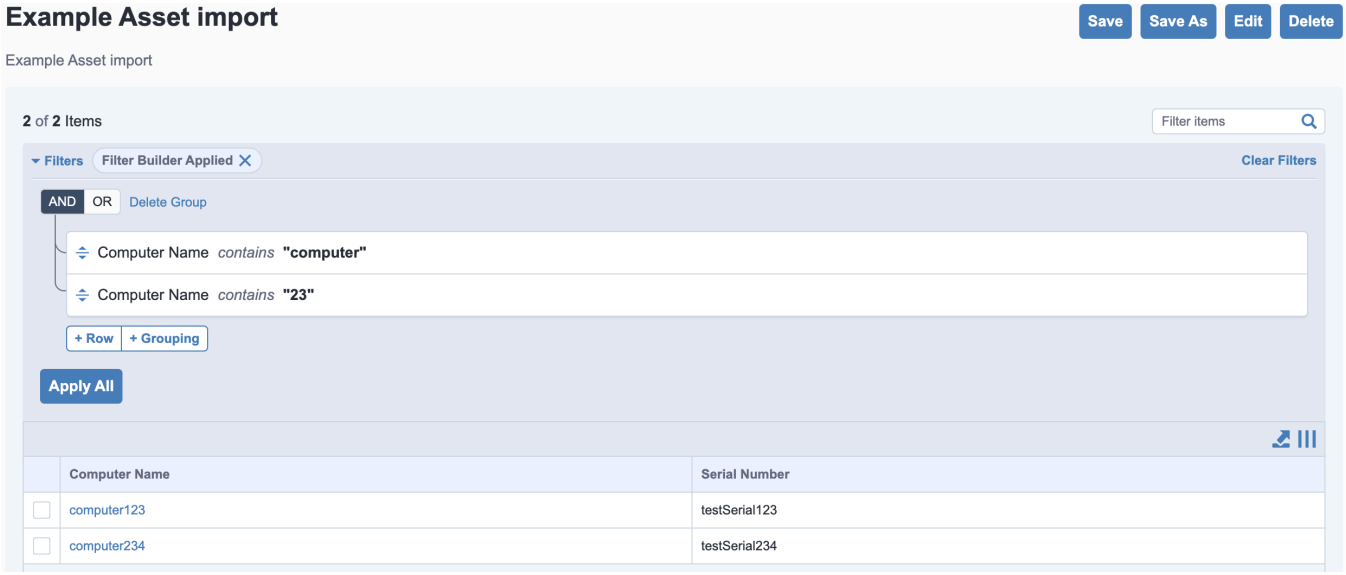
As the request is evaluated, the asset object defines a value for the `computerName` key field, matching the key field defined for the `example_source` Import API source, and is imported using that source. The example response shows a status of `Updated`:

```
{
  "data": {
    "assetUpsertEndpoints": {
      "assets": [
        {
          "eid": "12345",
          "source": {
            "namespace": "Asset: example_source",
            "id": "123",
            "name": "example_source"
          },
          "status": "Updated",
          "error": null
        }
      ]
    }
  }
}
```

To verify the import, create a report based on the custom source definition, and filter the report to search for imported assets. For example, creating a report with **Computer Name** and **Serial Number** as columns, then filtering **Computer Name** for values containing `computer` and `23` returns the imported assets.

Figure 7. Example Asset import report

©2026 Tanium Inc. All Rights Reserved



Note: You can also add the Asset source ID to view only assets from that source.

For more information, see Tanium Asset User Guide: Create a report.

Import assets

mutation.assetsImport

About this task

Stability: 0 - Deprecated

Note: This is deprecated in favor of `mutation.assetUpsertEndpoints`. For more information, see the example syntax on page 191.

Note: Before you submit this mutation request, you must configure an Asset custom source. For more information and an example workflow, see Tanium Asset User Guide: Create an Import API source.

©2026 Tanium Inc. All Rights Reserved

Note: After you submit a successful request, it can take up to two import cycles, or approximately 20 minutes, for attributes to appear in Asset and in reports.

The `assetsImport` mutation requires Asset 1.23.8 and imports assets defined in the request `json` variable value, using the defined `sourceName` custom source. To determine whether an existing asset updates, or a new asset is created:

1. If an imported asset key matches an existing asset key imported using the `sourceName` custom source, that existing asset updates.
2. If no existing asset from the configured source matches, but values in the imported asset match an existing asset key imported using a source different from the `sourceName` custom source, and the other source allows the `sourceName` custom source to update, the existing asset from the other source updates.
3. Otherwise, if no existing asset matches the imported asset, the asset is imported as a new asset using the `sourceName` custom source.

Note: If the request is structurally and syntactically valid, each asset object imports separately. A failed asset import does not prevent other valid asset objects in that request from importing.

Define the `json` attribute as a JSON object array, with each object corresponding to an asset. Ensure that the Asset custom source passes the JSON object array according to the following guidelines:

- Use the backslash escape character with each quotation mark (`\"`) in the JSON array.
- You must define key field values for each object. If you do not define the key field values, the object does not import and the response returns `ErrorMissingKey` status for that object.
- You can define source field mappings to destination fields for each object. If you do not define a source field mapping, values from those source fields are imported as `null` values for that object.
- Field values are defined per object; you cannot globally define a single field value in the request to apply to every imported asset object.
- You can define up to 5000 asset objects per array, and by extension, per request.

©2026 Tanium Inc. All Rights Reserved

The following example Asset custom source, named `test-name`, defines two field mappings for `computer_name` and `serial_number` and defines `computer_name` as a key field.

```
{
  "keys": [
    "computer_name"
  ],
  "fieldMaps": [
    {
      "destination": "computer_name",
      "source": "computerName"
    },
    {
      "destination": "serial_number",
      "source": "serialNumber"
    }
  ]
}
```

The following mutation request imports assets based on the variables definition:

```
mutation exampleImportAssets($source: String!, $json: String!) {
  assetsImport(input: {sourceName: $source, json: $json}) {
    assets {
      id
      index
      status
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary.

The following variables manually define the `test-name` source for importing and two asset objects in the JSON object array. Each object has a value for the key field (`computerName`) and for the `serialNumber` field mapping.

```
{
  "source": "test-name",
  "json": "[{\"computerName\": \"computer123\", \"serialNumber\": \"testSerial123\"},
{\"computerName\": \"computer234\", \"serialNumber\": \"testSerial234\"}]"
}
```

As the request is evaluated, each asset object defines a value for the `computerName` key field, matching the key field defined for the `test-name` custom source, and both are imported using that source. The example response shows both objects with a status of `Created`:

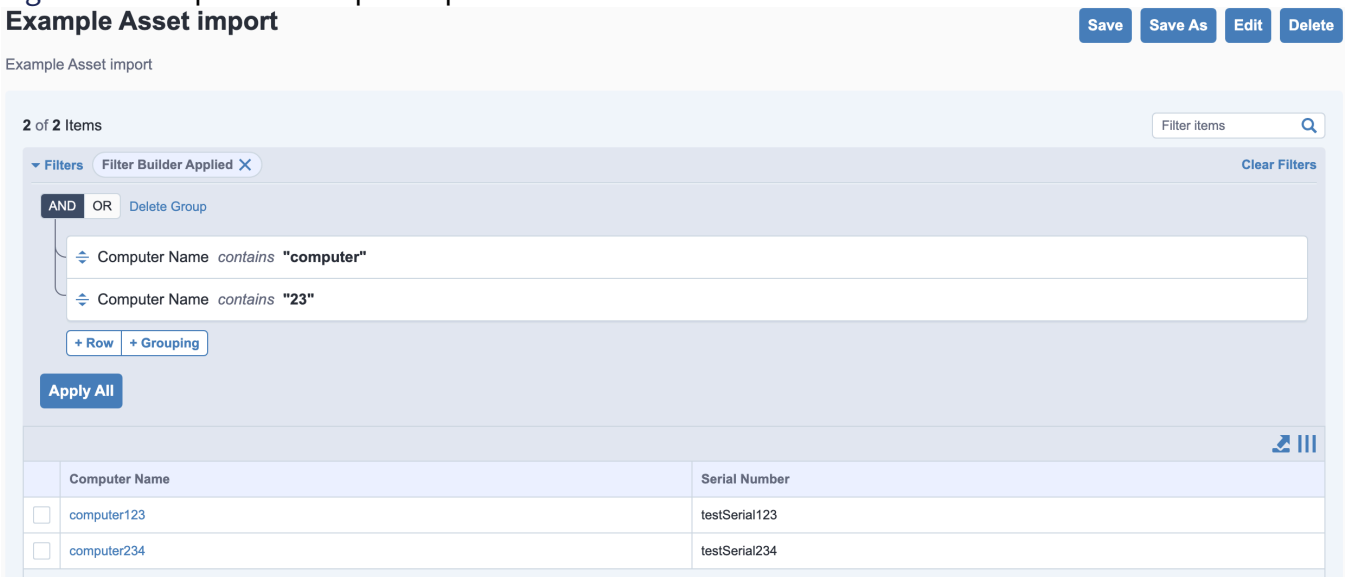
```
{
  "data": {
    "assetsImport": {
      "assets": [
        {
          "id": 6,
          "index": 0,
```

©2026 Tanium Inc. All Rights Reserved

```
          "status": "Created"
        },
        {
          "id": 7,
          "index": 1,
          "status": "Created"
        }
      ]
    }
  }
}
```


To verify the import, create a report based on the custom source definition, and filter the report to search for imported assets. For example, creating a report with **Computer Name** and **Serial Number** as columns, then filtering **Computer Name** for values containing `computer` and `23` returns the imported assets.

Figure 8. Example Asset import report



For more information, see Tanium Asset User Guide: Create a report.

Automate examples

The following queries and mutations require Automate to create, retrieve, update, and delete playbooks and playbook runs, and start a playbook run from a playbook.

Add a playbook Add Custom Tag (Non-Windows) step

(mutation.upsertPlaybook (input.steps))

About this task

©2026 Tanium Inc. All Rights Reserved

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.



Note: This step type extends the **Deploy Action** step for a specific package. If you add this step type using a GraphQL request instead of through Tanium Console, the step name appears as **Deploy Action** when you edit the playbook. Confirm the package ID and name for your environment, as it might differ from the example syntax.

The **Add Custom Tag (Non-Windows)** step requires the following permissions to deploy the **Custom Tagging - Add Tags (Non-Windows)** package:

- **Interact Execute action** permission
- **Package read** permission is required on the **Core Content** content set.
- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.



Note: The **Administrator** reserved role, **Content Administrator** reserved role, and **Interact Power User** role have all these permissions.

For more information, see Tanium Console User Guide: Action management permissions.

The **Add Custom Tag (Non-Windows)** step deploys the **Custom Tagging - Add Tags (Non-Windows)** package to targeted endpoints, which assigns one or more custom tags to targeted endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Deploy Action** step that runs the **Custom Tagging - Add Tags (Non-Windows)** package on the targeted endpoints belonging to the **All Linux** computer group:

```
mutation exampleUpsertPlaybookAddTagLinux($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
```

©2026 Tanium Inc. All Rights Reserved

```
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_add_tag_non_windows",
    "description": "Example playbook - Add Tags to Non-Windows Machine step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_add_tag_non_windows_step",
        "description": "Example step 1: Add Tags to Non-Windows Machines",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Linux"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1200
            },
            "completionPercentage": 1,
            "package": {
              "id": "36",
              "name": "Custom Tagging - Add Tags (Non-Windows)",
```

©2026 Tanium Inc. All Rights Reserved

```
              "params": [
                "example_tag"
              ]
            },
            "skipVerification": false
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_add_tag_non_windows",
        "steps": [
          {
            "description": "Example step 1: Add Tags to Non-Windows Machines",
            "name": "example_add_tag_non_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Add Custom Tag (Windows) step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

Note: This step type extends the **Deploy Action** step for a specific package. If you add this step type using a GraphQL request instead of through Tanium Console, the step name appears as **Deploy Action** when you edit the playbook. Confirm the package ID and name for your environment, as it might differ from the example syntax.

The **Add Custom Tag (Windows)** step requires the following permissions to deploy the **Custom**

©2026 Tanium Inc. All Rights Reserved

Tagging - Add Tags package:

- **Interact Execute action** permission
- **Package read** permission is required on the **Core Content** content set.
- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.



Note: The **Administrator** reserved role, **Content Administrator** reserved role, and **Interact Power User** role have all these permissions.

For more information, see Tanium Console User Guide: Action management permissions.

The **Add Custom Tag (Windows)** step deploys the **Custom Tagging - Add Tags** package to targeted endpoints, which assigns one or more custom tags to targeted endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Deploy Action** step that runs the **Custom Tagging - Add Tags** package on the targeted endpoints belonging to the **All Windows** computer group:

```
mutation exampleUpsertPlaybookAddTagWindows($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "input": {
    "name": "example_playbook_add_tag_windows",
    "description": "Example playbook - Add Tags to Windows Machine step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_add_tag_windows_step",
        "description": "Example step 1: Add Tags to Windows Machines",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1200
            },
            "completionPercentage": 1,
            "package": {
              "id": "35",
              "name": "Custom Tagging - Add Tags",
              "params": [
                "example_tag"
              ]
            },
            "skipVerification": false
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
```

```
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_add_tag_windows",
        "steps": [
          {
            "description": "Example step 1: Add Tags to Windows Machines",
            "name": "example_add_tag_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Deploy Action step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

The **Deploy Action** step requires the following permissions to deploy actions, including action locks:

- **Interact Execute action** permission
- **Package read** permission on content sets determine which packages are available for selection in actions.
- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.

> **Note:** The **Administrator** reserved role, **Content Administrator** reserved role, and **Interact Power User** role have all these permissions.

For more information, see Tanium Console User Guide: Action management permissions.

©2026 Tanium Inc. All Rights Reserved

The **Deploy Action** step deploys an action that runs a package on targeted endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Deploy Action** step that runs the **Example Package** package on the targeted endpoints belonging to the **All Windows** computer group:

```
mutation exampleUpsertPlaybookDeployAction($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_deploy_action",
    "description": "Example playbook - Deploy action step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_deploy_action_step",
        "description": "Example step 1: Deploy example package",
        "endpointTargeting": {
          "memberOf": {
            "name": "All Windows"
          },
          "negated": false,
```

©2026 Tanium Inc. All Rights Reserved

```
          "any": true
        },
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 12000
            },
            "completionPercentage": 1,
            "package": {
              "name": "Example package"
            },
            "skipVerification": false
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_deploy_action",
        "steps": [
          {
            "description": "Example step 1: Deploy example package",
            "name": "example_deploy_action_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add email notifications to a playbook

(mutation.upsertPlaybook (input.notificationSettings.email))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

You can configure a playbook to send email notifications to recipient email addresses based on

©2026 Tanium Inc. All Rights Reserved

playbook run events. For information on the available playbook run event types, see `PlaybookNotificationEventType` in the Documentation Explorer.

The following mutation creates a playbook, because no ID is specified in the input, and defines email notifications when a playbook run starts or ends:

```
mutation exampleUpsertPlaybookEmailNotification($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      contentSet {
        name
      }
      createdTime
      description
      id
      name
      notificationSettings {
        email {
          events
          recipients
        }
      }
      startingStep {
        id
        name
        description
      }
      steps {
        id
        description
        name
        nextStep {
          id
          name
          taskArgument {
            __typename
          }
        }
        previousStep {
          id
          name
          taskArgument {
            __typename
          }
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "input": {
    "name": "example_playbook_upsert_create",
    "description": "Example playbook upsert create",
    "contentSet": {
      "id": "1234"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "12345678-90ab-cdef-1234-567890abcdef",
    "steps": [
      {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_step_1",
        "description": "Example step 1: Initiation",
        "nextStepID": "22345678-90ab-cdef-1234-567890abcde2",
        "taskArgument": {
          "instruction": {
            "instruction": "# Initiate (start) this step, then confirm when complete."
          }
        }
      },
      {
        "id": "22345678-90ab-cdef-1234-567890abcde2",
        "name": "example_step_2",
        "description": "Example step 2: Processing",
        "nextStepID": "32345678-90ab-cdef-1234-567890abcde3",
        "taskArgument": {
          "instruction": {
            "instruction": "# Work (Process) this step, then confirm when complete."
          }
        }
      },
      {
        "id": "32345678-90ab-cdef-1234-567890abcde3",
        "name": "example_step_3",
        "description": "Example step 3: Resolution",
        "nextStepID": null,
        "taskArgument": {
          "instruction": {
            "instruction": "# Finish (Resolve) this step, then confirm when complete."
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
```

```
      "contentSet": {
        "name": "Automate"
      },
      "createdTime": "2024-08-06T14:26:32.079536Z",
      "description": "Example playbook upsert create",
      "id": "fedcba09-8765-4321-fedc-ba0987654321",
      "name": "example_playbook_upsert_create",
      "notificationSettings": {
        "email": {
          "events": [
            "START",
            "END"
          ],
          "recipients": [
            "host@example.com"
          ]
        }
      },
      "startingStep": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_step_1",
        "description": "Example step 1: Initiation"
      },
      "steps": [
        {
          "id": "12345678-90ab-cdef-1234-567890abcdef",
          "description": "Example step 1: Initiation",
          "name": "example_step_1",
          "nextStep": {
            "id": "22345678-90ab-cdef-1234-567890abcde2",
            "name": "example_step_2",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentInstruction"
            }
          },
          "previousStep": null
        },
        {
          "id": "22345678-90ab-cdef-1234-567890abcde2",
          "description": "Example step 2: Processing",
          "name": "example_step_2",
          "nextStep": {
            "id": "32345678-90ab-cdef-1234-567890abcde3",
            "name": "example_step_3",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentInstruction"
            }
          },
          "previousStep": {
            "id": "12345678-90ab-cdef-1234-567890abcdef",
            "name": "example_step_1",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentInstruction"
            }
          }
        },
        {
          "id": "32345678-90ab-cdef-1234-567890abcde3",
          "description": "Example step 3: Resolution",
          "name": "example_step_3",
          "nextStep": null,
          "previousStep": {
            "id": "22345678-90ab-cdef-1234-567890abcde2",
            "name": "example_step_2",
```


©2026 Tanium Inc. All Rights Reserved

```
              "taskArgument": {
                "__typename": "AutomateTaskArgumentInstruction"
              }
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Linux Quarantine step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.



Note: This step type extends the **Deploy Action** step for a specific package. If you add this step type using a GraphQL request instead of through Tanium Console, the step name appears as **Deploy Action** when you edit the playbook. Confirm the package ID and name for your environment, as it might differ from the example syntax.



Important: You must have the **Content Administrator** reserved role for Tanium Console. For more information, see Tanium Core Platform User Guide: Managing Roles.

The **Linux Quarantine** step requires the following permissions to deploy the **Threat Response - Quarantine Endpoint Template [Linux]** package:

- **Interact Execute action** permission
- **Package read** permission is required on the **Incident Response** content set.
- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.

©2026 Tanium Inc. All Rights Reserved

For more information, see Tanium Console User Guide: Action management permissions.

The **Linux Quarantine** step deploys the **Threat Response - Quarantine Endpoint Template [Linux]** package to targeted endpoints, which isolates a targeted endpoint from the network. For information about quarantining endpoints, see Tanium Threat Response User Guide: Quarantining endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Deploy Action** step that runs the **Threat Response - Quarantine Endpoint Template [Linux]** package on the targeted endpoints belonging to the **All Linux** computer group:

```
mutation exampleUpsertPlaybookQuarantineLinux($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_quarantine_linux",
    "description": "Example playbook - Quarantine Linux step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_quarantine_linux_step",
```

©2026 Tanium Inc. All Rights Reserved

```
        "description": "Example step 1: Quarantine Linux",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Linux"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1800
            },
            "completionPercentage": 1,
            "package": {
              "name": "Threat Response - Quarantine Endpoint Template [Linux]",
              "params": [
                "1",
                "1",
                "1",
                "1",
                "1",
                "1",
                "example notification",
                "IN:UDP:10.0.0.21:32:161\nOUT:UDP:10.0.0.21:32:162",
                "192.0.2.10",
                "192.0.2.20",
                "1",
                ""
              ]
            },
            "skipVerification": false
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_quarantine_linux",
        "steps": [
          {
            "description": "Example step 1: Quarantine Linux",
            "name": "example_quarantine_linux_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
```

©2026 Tanium Inc. All Rights Reserved

```
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Manual Instruction step

(mutation.upsertPlaybook (input.steps), mutation.playbookRunPerformOperation)

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

The **Manual Instruction** step displays an instruction, generally in Markdown format, to the Tanium operator. The Tanium operator must confirm the completion to allow the playbook to proceed to the next step.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Manual Instruction** step, which shows information to a Tanium operator:

```
mutation exampleUpsertPlaybookManualInstruction($input: PlaybookUpsertInput!) {
  playbookUpsert(
    input: $input
  ) {
    playbook {
      createdTime
      description
      id
      name
      revision
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "input": {
    "name": "example_playbook_manual_instruction",
    "description": "Example playbook - Manual Instruction step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": ["START", "END"],
        "recipients": ["host@example.com"]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_manual_instruction_step",
        "description": "Example step 1: External process",
        "taskArgument": {
          "instruction": {
            "instruction": "# Check the external system output. Confirm this step when
complete."
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "createdTime": "2023-08-21T13:15:58.764932Z",
        "description": "Example playbook - Manual Instruction step",
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_manual_instruction",
        "revision": 1,
        "steps": [
          {
            "description": "Example step 1: External process",
            "name": "example_manual_instruction_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentInstruction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

For a running playbook with a **Manual Instruction** step at index `0`, the following mutation issues the step confirmation.

©2026 Tanium Inc. All Rights Reserved

Note: The `PlaybookRunOperation` object enumerates available playbook run operations.

```
mutation exampleConfirmPlaybookRunStep($input: PlaybookRunPerformOperationInput!) {
  playbookRunPerformOperation(
    input: $input
  ) {
    playbookRun {
      createdTime
      endTime
      events {
        createdTime
        initiator {
          persona {
            id
          }
          user {
            id
          }
        }
        name
      }
      id
      playbook {
        id
      }
      runAs {
        persona {
          id
        }
        user {
          id
        }
      }
      scheduleStartTime
      startTime
      status
      steps {
        confirmationResult {
          comment
          completedUser {
            id
          }
          completionTime
        }
        endTime
        playbookStep {
          description
          name
        }
        startTime
        status
        error
      }
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "input": {
    "id": "fedcba09-8765-4321-fedc-ba0987654321",
    "operation": "CONFIRM_STEP",
    "parameters": {
      "confirmStep": {
        "runStepIndex": 0,
        "comment": "# Example confirmation of Manual Instruction step."
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "playbookRunPerformOperation": {
      "playbookRun": {
        "createdTime": "2023-08-21T13:48:30.173539Z",
        "endTime": null,
        "events": [
          {
            "createdTime": "2023-08-21T06:48:30.1720982-07:00",
            "initiator": {
              "persona": null,
              "user": {
                "id": "1"
              }
            },
            "name": "CREATE"
          },
          {
            "createdTime": "2023-08-21T13:48:30.1800994Z",
            "initiator": null,
            "name": "START"
          },
          {
            "createdTime": "2023-08-21T13:48:30.2041051Z",
            "initiator": null,
            "name": "WAIT"
          },
          {
            "createdTime": "2023-08-21T06:51:37.6948538-07:00",
            "initiator": {
              "persona": null,
              "user": {
                "id": "1"
              }
            },
            "name": "CONFIRM"
          }
        ],
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "playbook": {
          "id": "12345678-90ab-cdef-1234-567890abcdef"
        },
        "runAs": {
          "persona": null,
          "user": {
            "id": "1"
          }
        },
```

©2026 Tanium Inc. All Rights Reserved

```
        "scheduleStartTime": null,
        "startTime": "2023-08-21T13:48:30.180099Z",
        "status": "IN_PROGRESS",
        "steps": [
          {
            "confirmationResult": {
              "comment": "# Example confirmation of Manual Instruction step.",
              "completedUser": {
                "id": "1"
              },
              "completionTime": "2023-08-21T06:51:37.6948538-07:00"
            },
            "endTime": null,
            "playbookStep": {
              "description": "Example step 1: External process",
              "name": "example_manual_instruction_step"
            },
            "startTime": "2023-08-21T13:48:30.2041051Z",
            "status": "IN_PROGRESS",
            "error": ""
          }
        ]
      }
    }
  }
}
```


Add a playbook Reboot Non-Windows Machine step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

Note: This step type extends the **Deploy Action** step for a specific package. If you add this step type through a GraphQL request instead of through Tanium Console, the step name appears as **Deploy Action** when you edit the playbook. Confirm the package ID and name for your environment, as it might differ from the example syntax.

The **Reboot Non-Windows Machine** step requires the following permissions to deploy the **Reboot Non-Windows Machine** package:

- **Interact Execute action** permission
- **Package read** permission on the **Core Content** content set.

©2026 Tanium Inc. All Rights Reserved

- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.



Note: The **Administrator** reserved role, **Content Administrator** reserved role, and **Interact Power User** role have all these permissions.

For more information, see Tanium Console User Guide: Action management permissions.

The **Reboot Non-Windows Machine** step deploys the **Reboot Non-Windows Machine** package to targeted endpoints, which reboots the endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Deploy Action** step that runs the **Reboot Non-Windows Machine** package on the targeted endpoints belonging to the **All Linux** computer group:

```
mutation exampleUpsertPlaybookRebootNonWindows($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_reboot_non_windows",
    "description": "Example playbook - Reboot Non-Windows Machine step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
```

©2026 Tanium Inc. All Rights Reserved

```
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_reboot_non_windows_step",
        "description": "Example step 1: Reboot Non-Windows Machines",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Linux"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1200
            },
            "completionPercentage": 1,
            "package": {
              "id": "47",
              "name": "Reboot Non-Windows Machine",
              "params": [
                "REBOOT",
                "30"
              ]
            },
            "skipVerification": false
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_reboot_non_windows",
        "steps": [
          {
            "description": "Example step 1: Reboot Non-Windows Machines",
            "name": "example_reboot_non_windows_step",
```

©2026 Tanium Inc. All Rights Reserved

```
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Reboot Windows Machine step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

Note: This step type extends the **Deploy Action** step for a specific package. If you add this step type using a GraphQL request instead of through Tanium Console, the step name appears as **Deploy Action** when you edit the playbook. Confirm the package ID and name for your environment, as it might differ from the example syntax.

The **Reboot Windows Machine** step requires the following permissions to deploy the **Reboot Windows Machine** package:

- **Interact Execute action** permission
- **Package read** permission on the **Core Content** content set.
- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.

Note: The **Administrator** reserved role, **Content Administrator** reserved role, and **Interact Power User** role have all these permissions.

©2026 Tanium Inc. All Rights Reserved

For more information, see Tanium Console User Guide: Action management permissions.

The **Reboot Windows Machine** step deploys the **Reboot Windows Machine** package to targeted endpoints, which reboots the endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Deploy Action** step that runs the **Reboot Windows Machine** package on the targeted endpoints belonging to the **All Windows** computer group:

```
mutation exampleUpsertPlaybookRebootWindows($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_reboot_windows",
    "description": "Example playbook - Reboot Windows Machine step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_reboot_windows_step",
        "description": "Example step 1: Reboot Windows Machines",
        "endpointTargeting": {
```

©2026 Tanium Inc. All Rights Reserved

```
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1200
            },
            "completionPercentage": 1,
            "package": {
              "id": "47",
              "name": "Reboot Windows Machine",
              "params": [
                "REBOOT",
                "yes",
                "30"
              ]
            },
            "skipVerification": false
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_reboot_windows",
        "steps": [
          {
            "description": "Example step 1: Reboot Windows Machines",
            "name": "example_reboot_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Add a playbook Remove Linux Quarantine step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.



Note: This step type extends the **Deploy Action** step for a specific package. If you add this step type using a GraphQL request instead of through Tanium Console, the step name appears as **Deploy Action** when you edit the playbook. Confirm the package ID and name for your environment, as it might differ from the example syntax.



Important: You must have the **Content Administrator** reserved role for Tanium Console. For more information, see Tanium Core Platform User Guide: Managing Roles.

The **Remove Linux Quarantine** step requires the following permissions to deploy the **Threat Response - Unquarantine Endpoint [Linux]** package:

- **Interact Execute action** permission
- **Package read** permission is required on the **Incident Response** content set.
- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.

For more information, see Tanium Console User Guide: Action management permissions.

The **Remove Linux Quarantine** step deploys the **Threat Response - Unquarantine Endpoint [Linux]** package to targeted endpoints, which restores network connectivity to a targeted quarantined endpoint. For information about quarantining endpoints, see Tanium Threat Response User Guide: Quarantining endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a

©2026 Tanium Inc. All Rights Reserved

Deploy Action step that runs the **Threat Response - Unquarantine Endpoint [Linux]** package on the targeted endpoints that belong to the **All Linux** computer group:

```
mutation exampleUpsertPlaybookRemoveQuarantineLinux($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_remove_quarantine_linux",
    "description": "Example playbook - Remove Quarantine Linux step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_remove_quarantine_linux_step",
        "description": "Example step 1: Remove Quarantine Linux",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Linux"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
```

©2026 Tanium Inc. All Rights Reserved

```
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1800
            },
            "completionPercentage": 1,
            "package": {
              "name": "Threat Response - Unquarantine Endpoint [Linux]"
            },
            "skipVerification": false
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_remove_quarantine_linux",
        "steps": [
          {
            "description": "Example step 1: Remove Quarantine Linux",
            "name": "example_remove_quarantine_linux_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Remove Windows Quarantine step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

©2026 Tanium Inc. All Rights Reserved

Note: This step type extends the **Deploy Action** step for a specific package. If you add this step type using a GraphQL request instead of through Tanium Console, the step name appears as **Deploy Action** when you edit the playbook. Confirm the package ID and name for your environment, as it might differ from the example syntax.

Important: You must have the **Content Administrator** reserved role for Tanium Console. For more information, see Tanium Core Platform User Guide: Managing Roles.

The **Remove Windows Quarantine** step requires the following permissions to deploy the **Threat Response - Unquarantine Endpoint [Windows]** package:

- **Interact Execute action** permission
- **Package read** permission is required on the **Incident Response** content set.
- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.

For more information, see Tanium Console User Guide: Action management permissions.

The **Remove Windows Quarantine** step deploys the **Threat Response - Unquarantine Endpoint [Windows]** package to targeted endpoints, which restores network connectivity to a targeted quarantined endpoint. For information about quarantining endpoints, see Tanium Threat Response User Guide: Quarantining endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Deploy Action** step that runs the **Threat Response - Unquarantine Endpoint [Windows]** package on the targeted endpoints belonging to the **All Windows** computer group:

```
mutation exampleUpsertPlaybookRemoveQuarantineWindows($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
```

©2026 Tanium Inc. All Rights Reserved

```
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_remove_quarantine_windows",
    "description": "Example playbook - Remove Quarantine Windows step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_remove_quarantine_windows_step",
        "description": "Example step 1: Remove Quarantine Windows",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1800
            },
            "completionPercentage": 1,
            "package": {
              "name": "Threat Response - Unquarantine Endpoint [Windows]"
            },
            "skipVerification": false
          }
        }
```

©2026 Tanium Inc. All Rights Reserved

```
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_remove_quarantine_windows",
        "steps": [
          {
            "description": "Example step 1: Remove Quarantine Windows",
            "name": "example_remove_quarantine_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Require User Confirmation step

(mutation.upsertPlaybook (input.steps), mutation.playbookRunPerformOperation)

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

The **Require User Confirmation** step displays an instruction, generally in Markdown format, to the Tanium operator. This step requires a Tanium operator to confirm that the playbook can proceed to the next step.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Require User Confirmation** step, which displays information to a Tanium operator and requires user confirmation before continuing to the next playbook run step:

```
mutation exampleUpsertPlaybookUserConfirmation($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      createdTime
      description
      id
      name
```

©2026 Tanium Inc. All Rights Reserved

```
      revision
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_require_user_confirmation",
    "description": "Example playbook - Require User Confirmation step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": ["START", "END"],
        "recipients": ["host@example.com"]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_require_user_confirmation_step",
        "description": "Example step 1: Verification",
        "taskArgument": {
          "confirmation": {
            "instruction": "# Verify that the system is running, then confirm when complete."
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "createdTime": "2023-08-21T13:15:58.764932Z",
        "description": "Example playbook - Require User Confirmation step",
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_require_user_confirmation",
        "revision": 1,
        "steps": [
```

©2026 Tanium Inc. All Rights Reserved

```
          {
            "description": "Example step 1: Verification",
            "name": "example_require_user_confirmation_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentConfirmation"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

For a running playbook with a Require User Confirmation step at index `0`, the following mutation issues the step confirmation.



Note: The `PlaybookRunOperation` object enumerates available playbook run operations.

```
mutation exampleConfirmPlaybookRunStep($input: PlaybookRunPerformOperationInput!) {
  playbookRunPerformOperation(
    input: $input
  ) {
    playbookRun {
      createdTime
      endTime
      events {
        createdTime
        initiator {
          persona {
            id
          }
          user {
            id
          }
        }
        name
      }
      id
      playbook {
        id
      }
      runAs {
        persona {
          id
        }
        user {
          id
        }
      }
      scheduleStartTime
      startTime
      status
      steps {
        confirmationResult {
          comment
```

©2026 Tanium Inc. All Rights Reserved

```
          completedUser {
            id
          }
          completionTime
        }
        endTime
        playbookStep {
          description
          name
        }
        startTime
        status
        error
      }
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "dfe531d2-634c-4143-b80b-2e1e3987f6e8",
    "operation": "CONFIRM_STEP",
    "parameters": {
      "confirmStep": {
        "runStepIndex": 0,
        "comment": "# Example confirmation of Require User Confirmation step."
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "playbookRunPerformOperation": {
      "playbookRun": {
        "createdTime": "2023-08-21T13:29:18.306899Z",
        "endTime": null,
        "events": [
          {
            "createdTime": "2023-08-21T06:29:18.3059042-07:00",
            "initiator": {
              "persona": null,
              "user": {
                "id": "1"
              }
            },
            "name": "CREATE"
          },
          {
            "createdTime": "2023-08-21T13:29:18.3089013Z",
            "initiator": null,
            "name": "START"
          },
          {
            "createdTime": "2023-08-21T13:29:18.332902Z",
            "initiator": null,
            "name": "WAIT"
```

©2026 Tanium Inc. All Rights Reserved

```
          },
          {
            "createdTime": "2023-08-21T06:31:25.3657962-07:00",
            "initiator": {
              "persona": null,
              "user": {
                "id": "1"
              }
            },
            "name": "CONFIRM"
          }
        ],
        "id": "dfe531d2-634c-4143-b80b-2e1e3987f6e8",
        "playbook": {
          "id": "12345678-90ab-cdef-1234-567890abcdef"
        },
        "runAs": {
          "persona": null,
          "user": {
            "id": "1"
          }
        },
        "scheduleStartTime": null,
        "startTime": "2023-08-21T13:29:18.308901Z",
        "status": "IN_PROGRESS",
        "steps": [
          {
            "confirmationResult": {
              "comment": "# Example confirmation of Require User Confirmation step.",
              "completedUser": {
                "id": "1"
              },
              "completionTime": "2023-08-21T06:31:25.3657962-07:00"
            },
            "endTime": null,
            "playbookStep": {
              "description": "Example step 1: Verification",
              "name": "example_require_user_confirmation_step"
            },
            "startTime": "2023-08-21T13:29:18.332902Z",
            "status": "IN_PROGRESS",
            "error": ""
          }
        ]
      }
    }
  }
}
```


Add steps that inherit endpoint targeting

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

©2026 Tanium Inc. All Rights Reserved

When configuring steps that target endpoints, note the following:

- Targeting inheritance relies upon referencing a prior step ID. When creating a playbook, manually assign a step ID for reference. Ensure that the step ID is unique, to avoid step ID reference collision issues when resolving a referenced step ID. Playbook steps without a user-defined ID are assigned a GUID in the format `12345678-90ab-cdef-1234-567890abcdef`.

- Not all steps target endpoints. For example, the **Wait** step waits for the configured amount of time before the playbook run proceeds to the next step, without targeting endpoints.

- You cannot inherit targeting from a **Verify Condition** step.

- If a step targets endpoints, such as `(input.steps.endpointTargeting)`, you must define one, and only one, of `customTargeting`, `previousStepTargeting`, or `previousStepCompletedTargeting`.

    - `customTargeting` supports targeting endpoints based on computer group membership, sensor evaluation, or a combination of both. For more information on supported syntax, see `ComputerGroupFilter` in the Documentation Explorer.

    - `previousStepTargeting` targets the same endpoints as a prior targeting step index value in the playbook step array.

    - `previousStepCompletedTargeting` targets the endpoints that completed the prior targeting step in the playbook step array.

- If the first step in a playbook targets endpoints, you must define `customTargeting`. Because no steps precede the first step, you cannot inherit targeting from prior steps.

- For any step in a playbook that targets endpoints after the first step, you can define `customTargeting` for the step, or inherit endpoint targeting using either `previousStepTargeting` or `previousStepCompletedTargeting`, referencing a prior step index.

- If you inherit endpoint targeting with either `previousStepTargeting` or `previousStepCompletedTargeting`, and the referenced step index does not target endpoints, the evaluation traverses backwards from the referenced step to find a step that targets endpoints, then uses that step targeting. If no steps in this evaluation target endpoints, this results in an error.

©2026 Tanium Inc. All Rights Reserved

- Depending on the step type, additional endpoint filtering might be required to narrow the targeted endpoints, regardless of endpoint targeting method.

The following mutation creates a playbook, because no ID is specified in the input, and defines three steps:

- step ID `12345678-90ab-cdef-1234-567890abcdef`: a **Deploy Action** step that uses the **Custom Tagging - Add Tags** to apply the `example-tag` tag to endpoints in the `All Windows` computer group
- step ID `02345678-90ab-cdef-1234-567890abcde0`: a **Deploy Action** step that inherits endpoint targeting from step ID `12345678-90ab-cdef-1234-567890abcdef` and defines the `All Windows 10` computer group as an additional filter
- step ID `92345678-90ab-cdef-1234-567890abcde9`: a **Verify Condition** step that sets a `successPercentage` value of `1` (100%) of all endpoints that completed step ID `02345678-90ab-cdef-1234-567890abcde0`

```
mutation exampleUpsertPlaybookStepInheritance($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      createdTime
      description
      id
      name
      revision
      steps {
        id
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_target_inheritance",
    "description": "Example playbook - Step target inheritance",
    "contentSet": {
```

©2026 Tanium Inc. All Rights Reserved

```
    "id": "1234"
  },
  "notificationSettings": {
    "email": {
      "events": [
        "START",
        "END"
      ],
      "recipients": [
        "host@example.com"
      ]
    }
  },
  "startingStepID": "12345678-90ab-cdef-1234-567890abcdef",
  "steps": [
    {
      "id": "12345678-90ab-cdef-1234-567890abcdef",
      "name": "example_add_tag_windows_step",
      "description": "Example step 1: Add Tags to Windows Machines",
      "endpointTargeting": {
        "customTargeting": {
          "memberOf": {
            "name": "All Windows"
          },
          "negated": false,
          "any": true
        }
      },
      "nextStepID": "02345678-90ab-cdef-1234-567890abcde0",
      "taskArgument": {
        "deployAction": {
          "actionGroup": {
            "id": "3",
            "name": "Default"
          },
          "actionDuration": {
            "seconds": 12000
          },
          "completionPercentage": 1,
          "package": {
            "id": "35",
            "name": "Custom Tagging - Add Tags",
            "params": [
              "example_tag"
            ]
          }
        }
      }
    },
    {
      "id": "02345678-90ab-cdef-1234-567890abcde0",
      "name": "example_deploy_action_step",
      "description": "Example step 2: Deploy action to Windows 10",
      "endpointTargeting": {
        "previousStepTargetingID": "12345678-90ab-cdef-1234-567890abcdef"
      },
      "nextStepID": "92345678-90ab-cdef-1234-567890abcde9",
      "taskArgument": {
        "deployAction": {
          "actionGroup": {
            "name": "All Windows 10"
          },
          "completionPercentage": 1,
          "package": {
```

©2026 Tanium Inc. All Rights Reserved

```
              "name": "Custom Tagging - Add Tags"
            }
          }
        }
      },
      {
        "id": "92345678-90ab-cdef-1234-567890abcde9",
        "name": "example_verify_condition_step",
        "description": "Example step 3: Check for virtual Windows utility run",
        "endpointTargeting": {
          "previousStepCompletedTargetingID": "02345678-90ab-cdef-1234-567890abcde0"
        },
        "taskArgument": {
          "verifyCondition": {
            "endpointFilter": {
              "sensor": {
                "name": "Is Virtual"
              },
              "op": "EQ",
              "value": "true",
              "negated": false,
              "any": false
            },
            "successPercentage": 1
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "createdTime": "2023-09-28T18:29:34.408097Z",
        "description": "Example playbook - Step target inheritance",
        "id": "52345678-90ab-cdef-1234-567890abcde5",
        "name": "example_playbook_target_inheritance",
        "revision": 1,
        "steps": [
          {
            "id": "12345678-90ab-cdef-1234-567890abcdef",
            "description": "Example step 1: Add Tags to Windows Machines",
            "name": "example_add_tag_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          },
          {
            "id": "02345678-90ab-cdef-1234-567890abcde0",
            "description": "Example step 2: Deploy action to Windows 10",
            "name": "example_deploy_action_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          },
          {
            "id": "92345678-90ab-cdef-1234-567890abcde9",
            "description": "Example step 3: Check for virtual Windows utility run",
```

©2026 Tanium Inc. All Rights Reserved

```
            "name": "example_verify_condition_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentVerifyCondition"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

The following mutation creates a playbook, because no ID is specified in the input, and defines the following steps:

- step one: a **Deploy Action** step that deploys the **Custom Tagging - Add Tags** package to all endpoints in the `All Windows` computer group and sets a `completionPercentage` value of `1.0` (100%) of endpoints
- step two: a **Verify Condition** step that checks that all endpoints in the `All Windows` computer group are not virtual
- step three: a **Require User Confirmation** step that does not target endpoints
- step four: a **Wait** step that does not target endpoints
- step five: a **Require User Confirmation** step that does not target endpoints
- step six: a **Deploy Action** step that sets a `successPercentage` value of `1.0` (100%) of all endpoints that completed step one

The endpoint targeting for step six references the endpoint targeting for step one, and does not consider any endpoint targeting or lack of endpoint targeting in steps two, three, four, or five.

```
mutation exampleUpsertPlaybookStepInheritanceSkip($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      createdTime
      description
      id
      name
      revision
      steps {
        id
        description
        name
        taskArgument {
          __typename
        }
      }
```

©2026 Tanium Inc. All Rights Reserved

```
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_target_inheritance",
    "description": "Example playbook - Step target inheritance",
    "contentSet": {
      "id": "1234"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "12345678-90ab-cdef-1234-567890abcdef",
    "steps": [
      {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_add_tag_windows_step",
        "description": "Example step 1: Add Tags to Windows Machines",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "nextStepID": "22345678-90ab-cdef-1234-567890abcde2",
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 12000
            },
            "completionPercentage": 1,
            "package": {
              "id": "35",
              "name": "Custom Tagging - Add Tags",
              "params": [
                "example_tag"
              ]
```

©2026 Tanium Inc. All Rights Reserved

```
            }
          }
        }
      },
      {
        "id": "22345678-90ab-cdef-1234-567890abcde2",
        "name": "example_verify_condition_step",
        "description": "Example step 2: Check for physical Windows",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "nextStepID": "32345678-90ab-cdef-1234-567890abcde3",
        "taskArgument": {
          "verifyCondition": {
            "endpointFilter": {
              "sensor": {
                "name": "Is Virtual"
              },
              "op": "EQ",
              "value": "false",
              "negated": false,
              "any": false
            },
            "successPercentage": 1
          }
        }
      },
      {
        "id": "32345678-90ab-cdef-1234-567890abcde3",
        "name": "example_confirmation_step",
        "description": "Example step 3: Confirm Windows workstations online",
        "nextStepID": "42345678-90ab-cdef-1234-567890abcde4",
        "taskArgument": {
          "confirmation": {
            "instruction": "Verify that critical Windows workstations are online, then
confirm this step."
          }
        }
      },
      {
        "id": "42345678-90ab-cdef-1234-567890abcde4",
        "name": "example_wait_step",
        "description": "Example step 4: Wait twelve hours for background processes",
        "nextStepID": "52345678-90ab-cdef-1234-567890abcde5",
        "taskArgument": {
          "wait": {
            "duration": {
              "seconds": 43200
            }
          }
        }
      },
      {
        "id": "52345678-90ab-cdef-1234-567890abcde5",
        "name": "example_additional_confirmation_step",
        "description": "Example step 5: Confirm background processes did not error",
        "nextStepID": "62345678-90ab-cdef-1234-567890abcde6",
        "taskArgument": {
```

©2026 Tanium Inc. All Rights Reserved

```
          "confirmation": {
            "instruction": "Verify that overnight processes ran in the
C:\\logs\\overnight\\[date]-[time]-overnight-[index-value].log files. If any errors in the
log, contact network admin on call. Otherwise, confirm this step."
          }
        }
      },
      {
        "id": "62345678-90ab-cdef-1234-567890abcde6",
        "name": "example_deploy_action_step",
        "description": "Example step 6: Deploy action to copy log info on important physical
Windows workstations",
        "endpointTargeting": {
          "previousStepTargetingID": "12345678-90ab-cdef-1234-567890abcdef"
        },
        "nextStepID": null,
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "name": "Critical Hardware"
            },
            "completionPercentage": 1,
            "package": {
              "name": "Copy Log Info"
            }
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "createdTime": "2024-05-06T19:48:33.601759Z",
        "description": "Example playbook - Step target inheritance",
        "id": "f635b09c-93e9-44e5-8094-6dc7c5a420ac",
        "name": "example_playbook_target_inheritance",
        "revision": 1,
        "steps": [
          {
            "id": "12345678-90ab-cdef-1234-567890abcdef",
            "description": "Example step 1: Add Tags to Windows Machines",
            "name": "example_add_tag_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          },
          {
            "id": "22345678-90ab-cdef-1234-567890abcde2",
            "description": "Example step 2: Check for physical Windows",
            "name": "example_verify_condition_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentVerifyCondition"
            }
          },
          {
            "id": "32345678-90ab-cdef-1234-567890abcde3",
```

©2026 Tanium Inc. All Rights Reserved

```
            "description": "Example step 3: Confirm Windows workstations online",
            "name": "example_confirmation_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentConfirmation"
            }
          },
          {
            "id": "42345678-90ab-cdef-1234-567890abcde4",
            "description": "Example step 4: Wait twelve hours for background processes",
            "name": "example_wait_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentWait"
            }
          },
          {
            "id": "52345678-90ab-cdef-1234-567890abcde5",
            "description": "Example step 5: Confirm background processes did not error",
            "name": "example_additional_confirmation_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentConfirmation"
            }
          },
          {
            "id": "62345678-90ab-cdef-1234-567890abcde6",
            "description": "Example step 6: Deploy action to copy log info on important
physical Windows workstations",
            "name": "example_deploy_action_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Verify Condition step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

The **Verify Condition** step evaluates the percentage of targeted endpoints that answered a Tanium question, compared to the total number of endpoints managed using Tanium Client. The step is considered complete when 95% of endpoints have responded, or after ten minutes elapse (the question expiration interval). If the percentage meets or exceeds the defined `successPercentage` value, the step completes and the playbook run moves to the next step. Otherwise, the playbook run ends.

©2026 Tanium Inc. All Rights Reserved

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Verify Condition** step that sets a `successPercentage` value of `0.501` (50.1%) of endpoints in the `All Windows` computer group that are not virtual:

```
mutation exampleUpsertPlaybookVerifyCondition($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      createdTime
      description
      id
      name
      revision
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_verify_condition",
    "description": "Example playbook - Verify Condition step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_verify_condition_step",
        "description": "Example step 1: Check for virtual Windows",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
```

©2026 Tanium Inc. All Rights Reserved

```
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "verifyCondition": {
            "endpointFilter": {
              "sensor": {
                "name": "Is Virtual"
              },
              "op": "EQ",
              "value": "true"
            },
            "successPercentage": 0.501
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "createdTime": "2023-08-21T13:15:58.764932Z",
        "description": "Example playbook - Verify Condition step",
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_verify_condition",
        "revision": 1,
        "steps": [
          {
            "description": "Example step 1: Check for virtual Windows",
            "name": "example_verify_condition_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentVerifyCondition"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Wait step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

©2026 Tanium Inc. All Rights Reserved

The **Wait** step waits for a configured amount of time, then moves to the next step in the playbook run.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Wait** step, which waits the specified duration of 60 seconds before continuing to the next playbook run step:

```
mutation exampleUpsertPlaybookWait($input: PlaybookUpsertInput!) {
  playbookUpsert(
    input: $input
  ) {
    playbook {
      createdTime
      description
      id
      name
      revision
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_wait_step",
    "description": "Example playbook - Wait step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
```

©2026 Tanium Inc. All Rights Reserved

```
        "name": "example_wait_step",
        "description": "Example step 1: Wait 1 minute",
        "taskArgument": {
          "wait": {
            "duration": {
              "nanoseconds": 0,
              "seconds": 60
            }
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "createdTime": "2024-08-06T17:16:21.467859Z",
        "description": "Example playbook - Wait step",
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_wait_step",
        "revision": 2,
        "steps": [
          {
            "description": "Example step 1: Wait 1 minute",
            "name": "example_wait_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentWait"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Add a playbook Windows Quarantine step

(mutation.upsertPlaybook (input.steps))

About this task

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

Note: This step type extends the **Deploy Action** step for a specific package. If you add this step type using a GraphQL request instead of through Tanium Console, the step name appears as **Deploy Action** when you edit the playbook. Confirm the package ID and name for your environment, as it might differ from the example syntax.

Important: You must have the **Content Administrator** reserved role for Tanium Console. For more information, see Tanium Core Platform User Guide: Managing Roles.

The **Windows Quarantine** step requires the following permissions to deploy the **Threat Response - Quarantine Endpoint Template [Windows]** package:

- **Interact Execute action** permission
- **Package read** permission is required on the **Incident Response** content set.
- **Sensor read** and **Saved Question read** permissions are required on the **Reserved** content set. The Tanium Server uses special saved questions to track and report action status within the deploy action workflow. Therefore, these permissions are required to complete the action deployment workflow.

For more information, see Tanium Console User Guide: Action management permissions.

The **Windows Quarantine** step deploys the **Threat Response - Quarantine Endpoint Template [Windows]** package to targeted endpoints, which isolates a targeted endpoint from the network. For information about how to quarantine endpoints, see Tanium Threat Response User Guide: Quarantining endpoints.

The following mutation creates a playbook, because no ID is specified in the input, and defines a **Deploy Action** step that runs the **Threat Response - Quarantine Endpoint Template [Windows]** package on the targeted endpoints belonging to the **All Windows** computer group:

```
mutation exampleUpsertPlaybookQuarantineWindows($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
```

©2026 Tanium Inc. All Rights Reserved

```
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_quarantine_windows",
    "description": "Example playbook - Quarantine Windows step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_quarantine_windows_step",
        "description": "Example step 1: Quarantine Windows",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1800
            },
            "completionPercentage": 1,
            "package": {
              "name": "Threat Response - Quarantine Endpoint Template [Windows]",
              "params": [
                "1",
                "1",
                "1",
```

©2026 Tanium Inc. All Rights Reserved

```
                "1",
                "1",
                "1",
                "example notification",
                "IN:UDP:10.0.0.21:32:161\nOUT:UDP:10.0.0.21:32:162",
                "192.0.2.10",
                "192.0.2.20",
                "0",
                ""
              ]
            },
            "skipVerification": false
          }
        }
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_quarantine_windows",
        "steps": [
          {
            "description": "Example step 1: Quarantine Windows",
            "name": "example_quarantine_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

Create a playbook run from a playbook

(mutation.playbookStart)

About this task

The `playbookStart` mutation requires Automate 1.3 or later and the **Automate Playbook execute** permission on the content set to which a playbook belongs to start a playbook run.

The following mutation creates a playbook run from a playbook that matches the ID.

```
mutation exampleStartPlaybook($id: ID!) {
  playbookStart(input: {id: $id}) {
```

```
    playbookRun {
      createdTime
      endTime
      events {
        createdTime
        initiator {
          user {
            displayName
            username
          }
          persona {
            id
          }
        }
        name
      }
      id
      playbook {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
        }
        updatedTime
      }
      runAs {
        persona {
          id
        }
        user {
          id
        }
      }
      scheduleStartTime
      startTime
      status
      steps {
        playbookStep {
          description
          name
        }
        status
        error
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the playbook ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef"
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "playbookStart": {
      "playbookRun": {
        "createdTime": "2023-08-18T19:47:44.041634Z",
        "endTime": null,
        "events": [
          {
            "createdTime": "2023-08-18T12:47:44.0394017-07:00",
            "initiator": {
              "user": {
                "displayName": null,
                "username": "Administrator"
              },
              "persona": null
            },
            "name": "CREATE"
          }
        ],
        "id": "618903ae-a337-40cd-849c-66cb471f5768",
        "playbook": {
          "createdTime": "2023-08-18T19:23:34.008989Z",
          "description": "Example playbook update",
          "id": "12345678-90ab-cdef-1234-567890abcdef",
          "name": "example_playbook_update",
          "revision": 2,
          "steps": [
            {
              "description": "Example step 1 update: Initiation",
              "name": "example_step_1"
            },
            {
              "description": "Example step 2 update: Processing",
              "name": "example_step_2"
            },
            {
              "description": "Example step 3 update: Resolution",
              "name": "example_step_3"
            }
          ],
          "updatedTime": "2023-08-18T19:38:52.458956Z"
        },
        "runAs": {
          "persona": null,
          "user": {
            "id": "1"
          }
        },
        "scheduleStartTime": null,
        "startTime": null,
        "status": "NOT_STARTED",
        "steps": [
          {
            "playbookStep": {
              "description": "Example step 1 update: Initiation",
              "name": "example_step_1"
            },
            "status": "NOT_STARTED",
            "error": ""
          },
          {
```


©2026 Tanium Inc. All Rights Reserved

```
            "playbookStep": {
              "description": "Example step 2 update: Processing",
              "name": "example_step_2"
            },
            "status": "NOT_STARTED",
            "error": ""
          },
          {
            "playbookStep": {
              "description": "Example step 3 update: Resolution",
              "name": "example_step_3"
            },
            "status": "NOT_STARTED",
            "error": ""
          }
        ]
      },
      "error": null
    }
  }
}
```

Get a playbook by ID and version

(query.playbook)

About this task

The `playbook` query requires Automate 1.3 or later and the **Automate Playbook read** permission on the content set to which a playbook version belongs to retrieve the playbook.

The following query retrieves a playbook based on the ID and playbook version.

```
query exampleGetPlaybook($id: ID!, $revision: Int) {
  playbook(ref: {id: $id, revision: $revision}) {
    createdTime
    description
    id
    name
    notificationSettings {
      email {
        events
        recipients
      }
    }
    revision
    startingStep {
      id
      name
      description
    }
    steps {
      id
      description
      name
      taskArgument {
        __typename
```

©2026 Tanium Inc. All Rights Reserved

```
      }
    }
    updatedTime
  }
}
```

Include the playbook ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "revision": 1
}
```

Example response:

```
{
  "data": {
    "playbook": {
      "createdTime": "2023-08-18T15:53:41.091488Z",
      "description": "example description",
      "id": "12345678-90ab-cdef-1234-567890abcdef",
      "name": "Example playbook",
      "notificationSettings": {
        "email": {
          "events": [
            "START",
            "COMPLETE",
            "END",
            "ERROR",
            "WAIT"
          ],
          "recipients": [
            "username@example.com",
            "username_2@example.com"
          ]
        }
      },
      "revision": 1,
      "startingStep": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "Require User Confirmation",
        "description": "example description"
      },
      "steps": [
        {
          "id": "12345678-90ab-cdef-1234-567890abcdef",
          "description": "example description",
          "name": "Require User Confirmation",
          "taskArgument": {
            "__typename": "AutomateTaskArgumentConfirmation"
          }
        }
      ],
      "updatedTime": "2023-08-18T15:53:41.091488Z"
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Get a playbook run by ID

(query.playbookRun)

About this task

The `playbookRun` query requires Automate 1.3 or later and the **Automate Playbook read** permission on the content set to which a playbook belongs to retrieve associated playbook runs.

The following query retrieves a playbook run based on the ID.

```
query exampleGetPlaybookRun($id: ID!) {
  playbookRun(ref: {id: $id}) {
    createdTime
    endTime
    events {
      createdTime
      initiator {
        persona {
          id
        }
        user {
          id
        }
      }
      name
    }
    id
    playbook {
      activeRunsCount
      author {
        displayName
        id
        username
      }
      createdTime
      description
      id
      lastModifiedBy {
        displayName
        id
        username
      }
      name
      revision
      steps {
        description
        name
      }
      updatedTime
    }
    runAs {
      persona {
        id
      }
      user {
        id
      }
```

©2026 Tanium Inc. All Rights Reserved

```
    }
    scheduleStartTime
    startTime
    status
    steps {
      playbookStep {
        description
        name
        __typename
      }
      status
      error
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "fedcba09-8765-4321-fedc-ba0987654321"
}
```

Example response:

```
{
  "data": {
    "playbookRun": {
      "createdTime": "2023-08-18T19:49:13.113955Z",
      "endTime": null,
      "events": [
        {
          "createdTime": "2023-08-18T12:49:13.1127798-07:00",
          "initiator": {
            "persona": null,
            "user": {
              "id": "1"
            }
          },
          "name": "CREATE"
        },
        {
          "createdTime": "2023-08-18T19:49:13.1167809Z",
          "initiator": null,
          "name": "START"
        },
        {
          "createdTime": "2023-08-18T19:49:13.1397834Z",
          "initiator": null,
          "name": "WAIT"
        }
      ],
      "id": "fedcba09-8765-4321-fedc-ba0987654321",
      "playbook": {
        "activeRunsCount": 3,
        "author": {
          "displayName": null,
          "id": "1",
          "username": "Administrator"
        },
        "createdTime": "2023-08-18T19:23:34.008989Z",
        "description": "Example playbook update",
```

©2026 Tanium Inc. All Rights Reserved

```
      "id": "12345678-90ab-cdef-1234-567890abcdef",
      "lastModifiedBy": {
        "displayName": null,
        "id": "1",
        "username": "Administrator"
      },
      "name": "example_playbook_update",
      "revision": 2,
      "steps": [
        {
          "description": "Example step 1 update: Initiation",
          "name": "example_step_1"
        },
        {
          "description": "Example step 2 update: Processing",
          "name": "example_step_2"
        },
        {
          "description": "Example step 3 update: Resolution",
          "name": "example_step_3"
        }
      ],
      "updatedTime": "2023-08-18T19:38:52.458956Z"
    },
    "runAs": {
      "persona": null,
      "user": {
        "id": "1"
      }
    },
    "scheduleStartTime": null,
    "startTime": "2023-08-18T19:49:13.11678Z",
    "status": "WAITING",
    "steps": [
      {
        "playbookStep": {
          "description": "Example step 1 update: Initiation",
          "name": "example_step_1",
          "__typename": "PlaybookStepInstruction"
        },
        "status": "WAITING",
        "error": ""
      },
      {
        "playbookStep": {
          "description": "Example step 2 update: Processing",
          "name": "example_step_2",
          "__typename": "PlaybookStepInstruction"
        },
        "status": "NOT_STARTED",
        "error": ""
      },
      {
        "playbookStep": {
          "description": "Example step 3 update: Resolution",
          "name": "example_step_3",
          "__typename": "PlaybookStepInstruction"
        },
        "status": "NOT_STARTED",
        "error": ""
      }
    ]
  }
}
```


©2026 Tanium Inc. All Rights Reserved

```
}
```

Get a playbook schedule by ID

(query.playbookSchedule)

About this task

The `playbookSchedule` query requires Automate 1.3 or later and the **Automate Playbook read** permission on the content set to which a playbook belongs to retrieve associated playbook schedules.

The following query retrieves a playbook schedule based on the ID.

```
query exampleGetPlaybookScheduleByID($ref: IdRefInput!) {
  playbookSchedule(ref: $ref) {
    id
    createdTime
    cron
    enabled
    events {
      createdTime
      initiator {
        user {
          id
          displayName
          username
        }
        persona {
          id
        }
      }
      name
    }
    playbook {
      id
      contentSet {
        id
      }
      currentUserHasActiveRun
      description
      name
      revision
    }
    scheduledBy {
      user {
        id
        displayName
        username
      }
      persona {
        id
      }
    }
    updatedTime
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the playbook schedule ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "ref": {
    "id": "12345678-90ab-cdef-1234-567890abcdef"
  }
}
```

Example response:


```
{
  "data": {
    "playbookSchedule": {
      "id": "12345678-90ab-cdef-1234-567890abcdef",
      "createdTime": "2024-04-12T20:37:54.860756Z",
      "cron": "0 0 1 1 1",
      "enabled": true,
      "events": [],
      "playbook": {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "contentSet": {
          "id": "123"
        },
        "currentUserHasActiveRun": false,
        "description": "Example playbook - Scheduling",
        "name": "example_playbook_scheduling",
        "revision": 3
      },
      "scheduledBy": {
        "user": {
          "id": "1",
          "displayName": null,
          "username": "admin"
        },
        "persona": null
      },
      "updatedTime": "2024-04-12T20:37:54.860756Z"
    }
  }
}
```


Get multiple playbooks

(query.playbooks)

About this task

The `playbooks` query requires Automate 1.3 or later and the **Automate Playbook read** permission on the content set to which a playbook belongs to retrieve associated playbooks.

©2026 Tanium Inc. All Rights Reserved

Note: Queries that request too many fields or too many results might return a `COMPLEXITY_LIMIT_EXCEEDED` error code. Reduce the number of fields in the query request, or update pagination to return fewer results.

Note: Querying `playbook.canRun` might consume a greater amount of resources than querying other fields, and the response might return a `COMPLEXITY_LIMIT_EXCEEDED` error code. Only include this field in a query request if you limit the number of results returned. Consider using `playbook.steps.currentUserMissingPermissions` instead to identify whether a user requires additional permissions to run a step, and for increased query efficiency.

The following query retrieves the first two playbooks.

```
query exampleGetPlaybooks($first: Int) {
  playbooks(first: $first) {
    edges {
      node {
        createdTime
        description
        id
        name
        notificationSettings {
          email {
            events
            recipients
          }
        }
        revision
        startingStep {
          id
          description
          name
        }
        steps {
          id
          description
          name
          nextStep {
            id
            description
            name
          }
          previousStep {
            id
            description
            name
          }
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
```

©2026 Tanium Inc. All Rights Reserved

```
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    authorUserIDs
    authorUsers {
      id
      username
      displayName
    }
    contentSetIDs
    contentSets {
      id
      name
      description
    }
    totalRecords
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "playbooks": {
      "edges": [
        {
          "node": {
            "createdTime": "2024-04-09T04:19:05.445776Z",
            "description": "",
            "id": "12345678-90ab-cdef-1234-567890abcdef",
            "name": "example playbook",
            "notificationSettings": {
              "email": {
                "events": [
                  "START",
                  "COMPLETE",
                  "END",
                  "ERROR",
                  "WAIT"
                ],
                "recipients": [
                  "username@example.com",
                  "username_2@example.com"
                ]
              }
            },
            "revision": 1,
            "startingStep": {
              "id": "fedcba09-8765-4321-fedc-ba0987654321",
              "description": "initial step in playbook",
              "name": "example_start_step"
            },
```

©2026 Tanium Inc. All Rights Reserved

```
          "steps": [
            {
              "id": "fedcba09-8765-4321-fedc-ba0987654321",
              "description": "initial step in playbook",
              "name": "example_start_step",
              "nextStep": null,
              "previousStep": null,
              "taskArgument": {
                "__typename": "AutomateTaskArgumentWait"
              }
            }
          ],
          "updatedTime": "2024-04-09T04:19:05.445776Z"
        }
      },
      {
        "node": {
          "createdTime": "2024-04-11T21:12:37.234882Z",
          "description": "",
          "id": "22345678-90ab-cdef-1234-567890abcde2",
          "name": "example playbook 2",
          "notificationSettings": {
            "email": {
              "events": [],
              "recipients": []
            }
          },
          "revision": 1,
          "startingStep": {
            "id": "example-step-id-1",
            "description": "example manual instruction description",
            "name": "Manual Instruction"
          },
          "steps": [
            {
              "id": "example-step-id-1",
              "description": "example manual instruction description",
              "name": "Manual Instruction",
              "nextStep": {
                "id": "example-step-id-2",
                "description": "another example manual description",
                "name": "Manual Instruction - Cloned"
              },
              "previousStep": null,
              "taskArgument": {
                "__typename": "AutomateTaskArgumentInstruction"
              }
            },
            {
              "id": "example-step-id-2",
              "description": "another example manual description",
              "name": "Manual Instruction - Cloned",
              "nextStep": null,
              "previousStep": {
                "id": "example-step-id-1",
                "description": "example manual instruction description",
                "name": "Manual Instruction"
              },
              "taskArgument": {
                "__typename": "AutomateTaskArgumentInstruction"
              }
            }
          ],
          "updatedTime": "2024-04-11T21:12:37.234882Z"
```


©2026 Tanium Inc. All Rights Reserved

```
          }
        }
      ],
      "pageInfo": {
        "hasPreviousPage": false,
        "hasNextPage": true,
        "startCursor":
"eyJpZCI6ImU2Nzg1MmNlLTJkZTMtNDljNi1hYjFhLTM3ZGQ3NTdjZTkwMyIsImNyZWF0ZWRfdGltZSI6IjIwMjQtMDQt
MDlUMDQ6MTk6MDUuNDQ1Nzc2WiIsIm5hbWUiOiJBZGFtbyAyIn0K",
        "endCursor":
"eyJpZCI6IjEyMzA5NzQ5LWZjMzAtNDcyMi1hZTk4LTU4ZjZjNzA0YzIwYiIsImNyZWF0ZWRfdGltZSI6IjIwMjQtMDQt
MTFUMjE6MTI6MzcuMjM0ODgyWiIsIm5hbWUiOiJBZGFtbyAzIn0K"
      },
      "authorUserIDs": [
        "12",
        "23",
        "34"
      ],
      "authorUsers": [
        {
          "id": "12",
          "username": "admin",
          "displayName": null
        },
        {
          "id": "23",
          "username": "example_user_1",
          "displayName": null
        },
        {
          "id": "34",
          "username": "example_user_2",
          "displayName": null
        }
      ],
      "contentSetIDs": [
        "1",
        "1234"
      ],
      "contentSets": [
        {
          "id": "1",
          "name": "Reserved",
          "description": "Built in sensors which can not be edited or removed."
        },
        {
          "id": "1234",
          "name": "Automate",
          "description": "Default content set for Tanium Automate"
        }
      ],
      "totalRecords": 15
    }
  }
}
```


Get multiple playbook runs

(query.playbookRuns)

©2026 Tanium Inc. All Rights Reserved

About this task

The `playbookRuns` query requires Automate 1.3 or later and the **Automate Playbook read** permission on the content set to which a playbook version belongs to retrieve associated playbook runs.

> **Note:** Queries that request too many fields or too many results might return a `COMPLEXITY_LIMIT_EXCEEDED` error code. Reduce the number of fields in the query request, or update pagination to return fewer results.

> **Note:** Querying `playbook.canRun` might consume a greater amount of resources than querying other fields, and the response might return a `COMPLEXITY_LIMIT_EXCEEDED` error code. Only include this field in a query request if you limit the number of results returned. Consider using `playbook.steps.currentUserMissingPermissions` instead to identify whether a user requires additional permissions to run a step, and for increased query efficiency.

> **Note:** If you do not have access to a content set, and a playbook is reassigned to that content set, you cannot retrieve playbook runs started from this playbook version. You can retrieve playbook runs from an earlier playbook version if you have access to the content set containing the earlier playbook version.

The following query retrieves the first two playbook runs.

```
query exampleGetPlaybookRuns($first: Int) {
  playbookRuns(first: $first) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
```

©2026 Tanium Inc. All Rights Reserved

```
      playbook {
        activeRunsCount
        author {
          id
        }
        createdTime
        description
        id
        lastModifiedBy {
          displayName
          id
          username
        }
        name
        revision
        steps {
          description
          name
        }
        updatedTime
      }
      runAs {
        persona {
          id
        }
        user {
          id
        }
      }
      scheduleStartTime
      startTime
      status
      steps {
        confirmationResult {
          comment
          completedUser {
            id
          }
          completionTime
        }
        playbookStep {
          description
          name
          __typename
        }
        startTime
        status
        error
      }
    }
  }
  initiatorUserIDs
  initiatorUsers {
    id
    username
    displayName
  }
  playbooks {
    playbookID
    playbookName
  }
  pageInfo {
    hasPreviousPage
    hasNextPage
```

```
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "playbookRuns": {
      "edges": [
        {
          "node": {
            "createdTime": "2023-08-18T19:47:44.041634Z",
            "endTime": null,
            "events": [
              {
                "createdTime": "2023-08-18T12:47:44.0394017-07:00",
                "initiator": {
                  "persona": null,
                  "user": {
                    "id": "1"
                  }
                },
                "name": "CREATE"
              },
              {
                "createdTime": "2023-08-18T19:47:44.2224093Z",
                "initiator": null,
                "name": "START"
              },
              {
                "createdTime": "2023-08-18T19:47:44.3544268Z",
                "initiator": null,
                "name": "WAIT"
              }
            ],
            "id": "02345678-90ab-cdef-1234-567890abcde0",
            "playbook": {
              "activeRunsCount": 3,
              "author": {
                "id": "1"
              },
              "createdTime": "2023-08-18T19:23:34.008989Z",
              "description": "Example playbook update",
              "id": "12345678-90ab-cdef-1234-567890abcdef",
              "lastModifiedBy": {
                "displayName": null,
                "id": "1",
                "username": "Administrator"
              },
              "name": "example_playbook_update",
              "revision": 2,
```

```
            "steps": [
              {
                "description": "Example step 1 update: Initiation",
                "name": "example_step_1"
              },
              {
                "description": "Example step 2 update: Processing",
                "name": "example_step_2"
              },
              {
                "description": "Example step 3 update: Resolution",
                "name": "example_step_3"
              }
            ],
            "updatedTime": "2023-08-18T19:38:52.458956Z"
          },
          "runAs": {
            "persona": null,
            "user": {
              "id": "1"
            }
          },
          "scheduleStartTime": null,
          "startTime": "2023-08-18T19:47:44.222409Z",
          "status": "WAITING",
          "steps": [
            {
              "confirmationResult": null,
              "playbookStep": {
                "description": "Example step 1 update: Initiation",
                "name": "example_step_1",
                "__typename": "PlaybookStepInstruction"
              },
              "startTime": "2023-08-18T19:47:44.3544268Z",
              "status": "WAITING",
              "error": ""
            },
            {
              "confirmationResult": null,
              "playbookStep": {
                "description": "Example step 2 update: Processing",
                "name": "example_step_2",
                "__typename": "PlaybookStepInstruction"
              },
              "startTime": null,
              "status": "NOT_STARTED",
              "error": ""
            },
            {
              "confirmationResult": null,
              "playbookStep": {
                "description": "Example step 3 update: Resolution",
                "name": "example_step_3",
                "__typename": "PlaybookStepInstruction"
              },
              "startTime": null,
              "status": "NOT_STARTED",
              "error": ""
            }
          ]
        }
      },
      {
        "node": {
```


©2026 Tanium Inc. All Rights Reserved

```
      "createdTime": "2023-08-18T19:49:13.113955Z",
      "endTime": null,
      "events": [
        {
          "createdTime": "2023-08-18T12:49:13.1127798-07:00",
          "initiator": {
            "persona": null,
            "user": {
              "id": "1"
            }
          },
          "name": "CREATE"
        },
        {
          "createdTime": "2023-08-18T19:49:13.1167809Z",
          "initiator": null,
          "name": "START"
        },
        {
          "createdTime": "2023-08-18T19:49:13.1397834Z",
          "initiator": null,
          "name": "WAIT"
        }
      ],
      "id": "22345678-90ab-cdef-1234-567890abcde2",
      "playbook": {
        "activeRunsCount": 3,
        "author": {
          "id": "1"
        },
        "createdTime": "2023-08-18T19:23:34.008989Z",
        "description": "Example playbook update",
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "lastModifiedBy": {
          "displayName": null,
          "id": "1",
          "username": "Administrator"
        },
        "name": "example_playbook_update",
        "revision": 2,
        "steps": [
          {
            "description": "Example step 1 update: Initiation",
            "name": "example_step_1"
          },
          {
            "description": "Example step 2 update: Processing",
            "name": "example_step_2"
          },
          {
            "description": "Example step 3 update: Resolution",
            "name": "example_step_3"
          }
        ],
        "updatedTime": "2023-08-18T19:38:52.458956Z"
      },
      "runAs": {
        "persona": null,
        "user": {
          "id": "1"
        }
      },
      "scheduleStartTime": null,
      "startTime": "2023-08-18T19:49:13.11678Z",
```


©2026 Tanium Inc. All Rights Reserved

```
          "status": "WAITING",
          "steps": [
            {
              "confirmationResult": null,
              "playbookStep": {
                "description": "Example step 1 update: Initiation",
                "name": "example_step_1",
                "__typename": "PlaybookStepInstruction"
              },
              "startTime": "2023-08-18T19:49:13.1397834Z",
              "status": "WAITING",
              "error": ""
            },
            {
              "confirmationResult": null,
              "playbookStep": {
                "description": "Example step 2 update: Processing",
                "name": "example_step_2",
                "__typename": "PlaybookStepInstruction"
              },
              "startTime": null,
              "status": "NOT_STARTED",
              "error": ""
            },
            {
              "confirmationResult": null,
              "playbookStep": {
                "description": "Example step 3 update: Resolution",
                "name": "example_step_3",
                "__typename": "PlaybookStepInstruction"
              },
              "startTime": null,
              "status": "NOT_STARTED",
              "error": ""
            }
          ]
        }
      }
    ],
    "initiatorUserIDs": [
      "123",
      "234"
    ],
    "initiatorUsers": [
      {
        "id": "123",
        "username": "example_user_1",
        "displayName": null
      },
      {
        "id": "234",
        "username": "example_user_2",
        "displayName": null
      }
    ],
    "playbooks": [
      {
        "playbookID": "12345678-90ab-cdef-1234-567890abcdef",
        "playbookName": "example playbook"
      },
      {
        "playbookID": "fedcba09-8765-4321-fedc-ba0987654321",
        "playbookName": "example playbook 2"
      }
```


©2026 Tanium Inc. All Rights Reserved

```
      ],
      "pageInfo": {
        "hasPreviousPage": false,
        "hasNextPage": true,
        "startCursor":
"eyJpZCI6IjYxODkwM2FlLWEzMzctNDBjZC04NDljLTY2Y2I0NzFmNTc2OCIsImNyZWF0ZWRfdGltZSI6IjIwMjMtMDgt
MThUMTk6NDc6NDQuMDQxNjM0WiIsInN0YXR1cyI6IldBSVRJTkcifQo=",
        "endCursor":
"eyJpZCI6IjJiM2Q4NTIwLTJjZDktNGY5ZC1iNzdiLWY0ZmFjNWIyOWNlYyIsImNyZWF0ZWRfdGltZSI6IjIwMjMtMDgt
MThUMTk6NDk6MTMuMTEzOTU1WiIsInN0YXR1cyI6IldBSVRJTkcifQo="
      },
      "totalRecords": 5
    }
  }
}
```

Get multiple playbook schedules

(query.playbookSchedules)

About this task

The `playbookSchedules` query requires Automate 1.3 or later and the **Automate Playbook read** permission on the content set to which a playbook belongs to retrieve associated playbook schedules.

> **Note:** Queries that request too many fields or too many results might return a `COMPLEXITY_LIMIT_EXCEEDED` error code. Reduce the number of fields in the query request, or update pagination to return fewer results.

> **Note:** Querying `playbook.canRun` might consume a greater amount of resources than querying other fields, and the response might return a `COMPLEXITY_LIMIT_EXCEEDED` error code. Only include this field in a query request if you limit the number of results returned. Consider using `playbook.steps.currentUserMissingPermissions` instead to identify whether a user requires additional permissions to run a step, and for increased query efficiency.

The following query retrieves the first two playbook schedules.

```
query exampleGetPlaybookSchedules($first: Int) {
  playbookSchedules(first: $first) {
    edges {
      node {
        id
        createdTime
        cron
```

©2026 Tanium Inc. All Rights Reserved

```
        enabled
        events {
          createdTime
          initiator {
            user {
              id
              displayName
              username
            }
            persona {
              id
            }
          }
          name
        }
        playbook {
          id
          contentSet {
            id
          }
          currentUserHasActiveRun
          description
          name
          revision
        }
        scheduledBy {
          user {
            id
            displayName
            username
          }
          persona {
            id
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "playbookSchedules": {
      "edges": [
        {
```

```
        "node": {
          "id": "12345678-90ab-cdef-1234-567890abcdef",
          "createdTime": "2024-03-27T14:38:35.620598Z",
          "cron": "38 14 * * *",
          "enabled": false,
          "events": [],
          "playbook": {
            "id": "567890ab-cdef-1234-5678-90abcdef1234",
            "contentSet": {
              "id": "123"
            },
            "currentUserHasActiveRun": false,
            "description": "",
            "name": "Example Playbook - Scheduling",
            "revision": 1
          },
          "scheduledBy": {
            "user": {
              "id": "1",
              "displayName": null,
              "username": "admin"
            },
            "persona": null
          },
          "updatedTime": "2024-03-27T14:38:59.943286Z"
        }
      },
      {
        "node": {
          "id": "fedcba09-8765-4321-fedc-ba0987654321",
          "createdTime": "2024-03-27T14:41:32.91756Z",
          "cron": "41 14 * * *",
          "enabled": false,
          "events": [],
          "playbook": {
            "id": "ba098765-4321-fedc-ba09-87654321fedc",
            "contentSet": {
              "id": "123"
            },
            "currentUserHasActiveRun": false,
            "description": "",
            "name": "Example Playbook - Scheduling 2",
            "revision": 1
          },
          "scheduledBy": {
            "user": {
              "id": "1",
              "displayName": null,
              "username": "admin"
            },
            "persona": null
          },
          "updatedTime": "2024-03-27T14:42:01.149338Z"
        }
      }
    ],
    "pageInfo": {
      "hasNextPage": true,
      "hasPreviousPage": false,
      "startCursor":
"eyJJRCI6IjBhYmZhMzEwLTdkMDctNGFkZi05YmYxLWQzNTRlOWQ1ODQ1YiIsIkNyZWF0ZWRUaW1lIjoiMjAyNC0wMy0yN1QxNDozODozNS42MjA1OThaIn0K",
      "endCursor":
"eyJJRCI6IjM5MmJkYjIxLWExZGQtNDMxMi04MmRiLTc0NDllNmFhNGZmZiIsIkNyZWF0ZWRUaW1lIjoiMjAyNC0wMy0y
```


©2026 Tanium Inc. All Rights Reserved

```
N1QxNDo0MTozMi45MTc1NloifQo="
      },
      "totalRecords": 31
    }
  }
}
```

Upsert (create or update by ID) a playbook

(mutation.playbookUpsert)

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

Depending on the steps you add to a playbook, you might require additional Tanium solutions and permissions.

When you create a playbook, you define one or more steps as input. Each step contains a reference to the next step ID, and the playbook contains a reference to the first step ID. Depending on the step type, you configure one or more of the following:

- A step name
- A step description
- The next step ID
- The task argument that the step performs
- Additional information to display to a Tanium user, such as an instruction
- Question filtering to narrow the targeted endpoints
- Computer groups or sensors to narrow the targeted endpoints
- A success percentage of endpoints returned to determine whether to move on to the next step in the playbook run
- Other configuration specific to the step or task argument functionality

Starting with the second step in the playbook step sequence, you can configure a step to inherit endpoint targeting information from a prior step in the array based on the step ID. Each step evaluation depends on the step task argument functionality, and progression to the next step might

©2026 Tanium Inc. All Rights Reserved

depend on evaluating the percentage of endpoints returned, user input, waiting a specified amount of time, or additional success criteria.

Note: Playbooks must have unique names.

Note: Include `isNewRecord` in your request to determine whether the response created a playbook (`isNewRecord` equals `true`) or updated an existing playbook (`isNewRecord` equals `false`).

The following mutation creates a playbook, because no ID is specified in the input, and defines several steps:

```
mutation examplePlaybookUpsertCreate($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      contentSet {
        name
      }
      createdTime
      description
      id
      name
      startingStep {
        id
        name
        description
      }
      steps {
        id
        description
        name
        nextStep {
          id
          name
          taskArgument {
            __typename
          }
        }
        previousStep {
          id
          name
          taskArgument {
            __typename
          }
        }
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
```

©2026 Tanium Inc. All Rights Reserved

```
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_playbook_upsert_create",
    "description": "Example playbook upsert create",
    "contentSet": {
      "id": "1234"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "12345678-90ab-cdef-1234-567890abcdef",
    "steps": [
      {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_add_tag_windows_step",
        "description": "Example step 1: Add Tags to Windows Machines",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "nextStepID": "22345678-90ab-cdef-1234-567890abcde2",
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1200
            },
            "completionPercentage": 1,
            "package": {
              "id": "35",
              "name": "Custom Tagging - Add Tags",
              "params": [
                "example_tag"
              ]
            }
          }
```

©2026 Tanium Inc. All Rights Reserved

```
        }
      },
      {
        "id": "22345678-90ab-cdef-1234-567890abcde2",
        "name": "example_step_2",
        "description": "Example step 2: Verify Windows",
        "nextStepID": "32345678-90ab-cdef-1234-567890abcde3",
        "taskArgument": {
          "instruction": {
            "instruction": "# Process this step, then confirm when complete."
          }
        }
      },
      {
        "id": "32345678-90ab-cdef-1234-567890abcde3",
        "name": "example_remove_tag_windows_step",
        "description": "Example step 3: Remove Tags from Windows Machines",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "nextStepID": null,
        "taskArgument": {
          "deployAction": {
            "actionGroup": {
              "id": "3",
              "name": "Default"
            },
            "actionDuration": {
              "seconds": 1200
            },
            "completionPercentage": 1,
            "package": {
              "id": "37",
              "name": "Custom Tagging - Remove Tags"
            }
          }
        }
      }
    ]
  }
}
```

Example response if a playbook was created (`isNewRecord` equals `true`):

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "contentSet": {
          "name": "Automate"
        },
        "createdTime": "2024-08-06T14:26:32.079536Z",
        "description": "Example playbook upsert create",
        "id": "7261c52c-9e3f-427e-a813-0e845f7b89fd",
        "name": "example_playbook_upsert_create",
        "startingStep": {
```

©2026 Tanium Inc. All Rights Reserved

```
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_add_tag_windows_step",
        "description": "Example step 1: Add Tags to Windows Machines"
      },
      "steps": [
        {
          "id": "12345678-90ab-cdef-1234-567890abcdef",
          "description": "Example step 1: Add Tags to Windows Machines",
          "name": "example_add_tag_windows_step",
          "nextStep": {
            "id": "22345678-90ab-cdef-1234-567890abcde2",
            "name": "example_step_2",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentInstruction"
            }
          },
          "previousStep": null,
          "taskArgument": {
            "__typename": "AutomateTaskArgumentDeployAction"
          }
        },
        {
          "id": "22345678-90ab-cdef-1234-567890abcde2",
          "description": "Example step 2: Verify Windows",
          "name": "example_step_2",
          "nextStep": {
            "id": "32345678-90ab-cdef-1234-567890abcde3",
            "name": "example_remove_tag_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          },
          "previousStep": {
            "id": "12345678-90ab-cdef-1234-567890abcdef",
            "name": "example_add_tag_windows_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentDeployAction"
            }
          },
          "taskArgument": {
            "__typename": "AutomateTaskArgumentInstruction"
          }
        },
        {
          "id": "32345678-90ab-cdef-1234-567890abcde3",
          "description": "Example step 3: Remove Tags from Windows Machines",
          "name": "example_remove_tag_windows_step",
          "nextStep": null,
          "previousStep": {
            "id": "22345678-90ab-cdef-1234-567890abcde2",
            "name": "example_step_2",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentInstruction"
            }
          },
          "taskArgument": {
            "__typename": "AutomateTaskArgumentDeployAction"
          }
        }
      ]
    },
    "isNewRecord": true,
    "error": null
  }
```


©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

The following mutation updates an existing playbook with the ID `12345678-90ab-cdef-1234-567890abcdef`, and defines three Instruction steps, which display information to an end user:

```
mutation examplePlaybookUpsertUpdate($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      contentSet {
        name
      }
      createdTime
      description
      id
      name
      startingStep {
        id
        name
        description
      }
      steps {
        id
        description
        name
        nextStep {
          id
          name
          taskArgument {
            __typename
          }
        }
        previousStep {
          id
          name
          taskArgument {
            __typename
          }
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "fedcba09-8765-4321-fedc-ba0987654321",
    "name": "example_playbook_upsert_create",
    "description": "Example playbook upsert create",
    "contentSet": {
      "id": "1234"
```

©2026 Tanium Inc. All Rights Reserved

```
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "12345678-90ab-cdef-1234-567890abcdef",
    "steps": [
      {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_step_1",
        "description": "Example step 1: Initiation",
        "nextStepID": "22345678-90ab-cdef-1234-567890abcde2",
        "taskArgument": {
          "instruction": {
            "instruction": "# Initiate (start) this step, then confirm when complete."
          }
        }
      },
      {
        "id": "22345678-90ab-cdef-1234-567890abcde2",
        "name": "example_step_2",
        "description": "Example step 2: Processing",
        "nextStepID": "32345678-90ab-cdef-1234-567890abcde3",
        "taskArgument": {
          "instruction": {
            "instruction": "# Work (Process) this step, then confirm when complete."
          }
        }
      },
      {
        "id": "32345678-90ab-cdef-1234-567890abcde3",
        "name": "example_step_3",
        "description": "Example step 3: Resolution",
        "nextStepID": null,
        "taskArgument": {
          "instruction": {
            "instruction": "# Finish (Resolve) this step, then confirm when complete."
          }
        }
      }
    ]
  }
}
```

Example response if an existing playbook was updated (`isNewRecord` equals `false`):

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "contentSet": {
          "name": "Automate"
        },
        "createdTime": "2024-08-06T14:26:32.079536Z",
        "description": "Example playbook upsert create",
```

©2026 Tanium Inc. All Rights Reserved

```
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_playbook_upsert_create",
        "startingStep": {
          "id": "12345678-90ab-cdef-1234-567890abcdef",
          "name": "example_step_1",
          "description": "Example step 1: Initiation"
        },
        "steps": [
          {
            "id": "12345678-90ab-cdef-1234-567890abcdef",
            "description": "Example step 1: Initiation",
            "name": "example_step_1",
            "nextStep": {
              "id": "22345678-90ab-cdef-1234-567890abcde2",
              "name": "example_step_2",
              "taskArgument": {
                "__typename": "AutomateTaskArgumentInstruction"
              }
            },
            "previousStep": null
          },
          {
            "id": "22345678-90ab-cdef-1234-567890abcde2",
            "description": "Example step 2: Processing",
            "name": "example_step_2",
            "nextStep": {
              "id": "32345678-90ab-cdef-1234-567890abcde3",
              "name": "example_step_3",
              "taskArgument": {
                "__typename": "AutomateTaskArgumentInstruction"
              }
            },
            "previousStep": {
              "id": "12345678-90ab-cdef-1234-567890abcdef",
              "name": "example_step_1",
              "taskArgument": {
                "__typename": "AutomateTaskArgumentInstruction"
              }
            }
          },
          {
            "id": "32345678-90ab-cdef-1234-567890abcde3",
            "description": "Example step 3: Resolution",
            "name": "example_step_3",
            "nextStep": null,
            "previousStep": {
              "id": "22345678-90ab-cdef-1234-567890abcde2",
              "name": "example_step_2",
              "taskArgument": {
                "__typename": "AutomateTaskArgumentInstruction"
              }
            }
          }
        ]
      },
      "isNewRecord": false,
      "error": null
    }
  }
}
```


©2026 Tanium Inc. All Rights Reserved

Upsert (create or update by ID) a playbook schedule

(mutation.playbookScheduleUpsert)

The `playbookScheduleUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook execute** permission on the content set to which the associated playbook belongs to create or update a playbook schedule.

Create playbook schedule objects with a reference to a playbook ID to associate a given schedule with the playbook. Playbook runs for this playbook start based on the configured cron expression.

The following mutation creates a playbook schedule, because no playbook schedule ID is specified in the input. This playbook schedule is associated with playbook ID `fedcba09-8765-4321-fedc-ba0987654321` and runs using the computer group management permissions of the user and persona submitting the mutation request.



Note: Include `isNewRecord` in your request to determine whether the response created a playbook schedule (`isNewRecord` equals `true`) or updated an existing playbook schedule (`isNewRecord` equals `false`).

```
mutation examplePlaybookScheduleCreate($input: PlaybookScheduleUpsertInput!) {
  playbookScheduleUpsert(input: $input) {
    playbookSchedule {
      id
      createdTime
      cron
      enabled
      events {
        createdTime
        initiator {
          user {
            id
            displayName
            username
          }
          persona {
            id
          }
        }
        name
      }
      playbook {
        id
        contentSet {
          id
        }
        description
        name
      }
```

©2026 Tanium Inc. All Rights Reserved

```
      scheduledBy {
        user {
          id
          displayName
          username
        }
        persona {
          id
        }
      }
      updatedTime
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "playbookID": "fedcba09-8765-4321-fedc-ba0987654321",
    "cron": "0 0 1 1 1",
    "enabled": true
  }
}
```

Example response if a playbook schedule was created (`isNewRecord` equals `true`):

```
{
  "data": {
    "playbookScheduleUpsert": {
      "playbookSchedule": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "createdTime": "2024-04-12T19:41:43.118622Z",
        "cron": "0 0 1 1 1",
        "enabled": true,
        "events": [],
        "playbook": {
          "id": "fedcba09-8765-4321-fedc-ba0987654321",
          "contentSet": {
            "id": "123"
          },
          "description": "Example playbook - Scheduling",
          "name": "example_playbook_scheduling"
        },
        "scheduledBy": {
          "user": {
            "id": "1",
            "displayName": null,
            "username": "admin"
          },
          "persona": null
        },
        "updatedTime": "2024-04-12T19:41:43.118622Z"
      },
      "isNewRecord": true,
```

©2026 Tanium Inc. All Rights Reserved

```
      "error": null
    }
  }
}
```

The following mutation updates an existing playbook schedule with the ID `12345678-90ab-cdef-1234-567890abcdef`, changing the cron expression:

```
mutation examplePlaybookScheduleUpdate($input: PlaybookScheduleUpsertInput!) {
  playbookScheduleUpsert(input: $input) {
    playbookSchedule {
      id
      createdTime
      cron
      enabled
      events {
        createdTime
        initiator {
          user {
            id
            displayName
            username
          }
          persona {
            id
          }
        }
        name
      }
      playbook {
        id
        contentSet {
          id
        }
        description
        name
      }
      scheduledBy {
        user {
          id
          displayName
          username
        }
        persona {
          id
        }
      }
      updatedTime
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "input": {
    "id": "12345678-90ab-cdef-1234-567890abcdef",
    "playbookID": "fedcba09-8765-4321-fedc-ba0987654321",
    "cron": "0 0 1 1 2",
    "enabled": true
  }
}
```

Example response if an existing playbook schedule was updated (`isNewRecord` equals `false`):

```
{
  "data": {
    "playbookScheduleUpsert": {
      "playbookSchedule": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "createdTime": "2024-04-12T19:41:43.118622Z",
        "cron": "0 0 1 1 2",
        "enabled": true,
        "events": [],
        "playbook": {
          "id": "fedcba09-8765-4321-fedc-ba0987654321",
          "contentSet": {
            "id": "123"
          },
          "description": "Example playbook - Scheduling",
          "name": "example_playbook_scheduling"
        },
        "scheduledBy": {
          "user": {
            "id": "1",
            "displayName": null,
            "username": "admin"
          },
          "persona": null
        },
        "updatedTime": "2024-04-12T19:41:43.118622Z"
      },
      "isNewRecord": false,
      "error": null
    }
  }
}
```

Delete a playbook

(mutation.playbookPerformOperation)

About this task

Stability: 1.2 - Experimental (Release Candidate)

The `playbookPerformOperation` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to perform the delete operation on associated playbooks.

©2026 Tanium Inc. All Rights Reserved

The following mutation performs the `DELETE` operation on the playbook that matches the ID. This deletes the playbook and future-scheduled playbook runs associated with the playbook, but does not delete in-progress playbook runs associated with the playbook.



Note: The `PlaybookOperation` object enumerates available playbook operations.

```
mutation exampleDeletePlaybook($id: ID!, $operation: PlaybookOperation!) {
  playbookPerformOperation(input: {id: $id, operation: $operation}) {
    playbook {
      activeRunsCount
      author {
        displayName
        id
        username
      }
      createdTime
      description
      id
      lastModifiedBy {
        displayName
        id
        username
      }
      name
      revision
      runs {
        edges {
          node {
            createdTime
            endTime
            events {
              createdTime
              name
              initiator {
                user {
                  displayName
                  id
                  username
                }
                persona {
                  id
                }
              }
            }
            id
            playbook {
              id
            }
            runAs {
              user {
                displayName
                id
                username
              }
              persona {
                id
              }
```

©2026 Tanium Inc. All Rights Reserved

```
            }
            scheduleStartTime
            startTime
            status
            steps {
              error
              startTime
              endTime
            }
          }
        }
      }
      steps {
        description
        name
      }
      updatedTime
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "operation": "DELETE"
}
```

Example response:

```
{
  "data": {
    "playbookPerformOperation": {
      "playbook": {
        "activeRunsCount": 3,
        "author": {
          "displayName": null,
          "id": "1",
          "username": "Administrator"
        },
        "createdTime": "2023-08-18T19:23:34.008989Z",
        "description": "Example playbook update",
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "lastModifiedBy": {
          "displayName": null,
          "id": "1",
          "username": "Administrator"
        },
        "name": "example_playbook_update",
        "revision": 3,
        "runs": {
          "edges": [
            {
              "node": {
                "createdTime": "2023-08-18T19:47:44.041634Z",
                "endTime": null,
```

```
        "events": [
          {
            "createdTime": "2023-08-18T12:47:44.0394017-07:00",
            "name": "CREATE",
            "initiator": {
              "user": {
                "displayName": null,
                "id": "1",
                "username": "Administrator"
              },
              "persona": null
            }
          },
          {
            "createdTime": "2023-08-18T19:47:44.2224093Z",
            "name": "START",
            "initiator": null
          },
          {
            "createdTime": "2023-08-18T19:47:44.3544268Z",
            "name": "WAIT",
            "initiator": null
          }
        ],
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "playbook": {
          "id": "12345678-90ab-cdef-1234-567890abcdef"
        },
        "runAs": {
          "user": {
            "displayName": null,
            "id": "1",
            "username": "Administrator"
          },
          "persona": null
        },
        "scheduleStartTime": null,
        "startTime": "2023-08-18T19:47:44.222409Z",
        "status": "WAITING",
        "steps": [
          {
            "error": "",
            "startTime": "2023-08-18T19:47:44.3544268Z",
            "endTime": null
          },
          {
            "error": "",
            "startTime": null,
            "endTime": null
          },
          {
            "error": "",
            "startTime": null,
            "endTime": null
          },
          {
            "error": "",
            "startTime": null,
            "endTime": null
          }
        ]
      }
    },
    {
      "node": {
        "createdTime": "2023-08-18T19:49:13.113955Z",
        "endTime": null,
        "events": [
          {
```


©2026 Tanium Inc. All Rights Reserved

```
            "createdTime": "2023-08-18T12:49:13.1127798-07:00",
            "name": "CREATE",
            "initiator": {
              "user": {
                "displayName": null,
                "id": "1",
                "username": "Administrator"
              },
              "persona": null
            }
          },
          {
            "createdTime": "2023-08-18T19:49:13.1167809Z",
            "name": "START",
            "initiator": null
          },
          {
            "createdTime": "2023-08-18T19:49:13.1397834Z",
            "name": "WAIT",
            "initiator": null
          }
        ],
        "id": "eedcba09-8765-4321-fedc-ba098765432e",
        "playbook": {
          "id": "12345678-90ab-cdef-1234-567890abcdef"
        },
        "runAs": {
          "user": {
            "displayName": null,
            "id": "1",
            "username": "Administrator"
          },
          "persona": null
        },
        "scheduleStartTime": null,
        "startTime": "2023-08-18T19:49:13.11678Z",
        "status": "WAITING",
        "steps": [
          {
            "error": "",
            "startTime": "2023-08-18T19:49:13.1397834Z",
            "endTime": null
          },
          {
            "error": "",
            "startTime": null,
            "endTime": null
          },
          {
            "error": "",
            "startTime": null,
            "endTime": null
          },
          {
            "error": "",
            "startTime": null,
            "endTime": null
          }
        ]
      }
    },
    {
      "node": {
        "createdTime": "2023-08-18T19:49:17.873797Z",
        "endTime": null,
        "events": [
          {
            "createdTime": "2023-08-18T12:49:17.8731032-07:00",
            "name": "CREATE",
```


©2026 Tanium Inc. All Rights Reserved

```
                "initiator": {
                  "user": {
                    "displayName": null,
                    "id": "1",
                    "username": "Administrator"
                  },
                  "persona": null
                }
              },
              {
                "createdTime": "2023-08-18T19:49:18.1610911Z",
                "name": "START",
                "initiator": null
              },
              {
                "createdTime": "2023-08-18T19:49:18.2150748Z",
                "name": "WAIT",
                "initiator": null
              }
            ],
            "id": "dedcba09-8765-4321-fedc-ba098765432d",
            "playbook": {
              "id": "12345678-90ab-cdef-1234-567890abcdef"
            },
            "runAs": {
              "user": {
                "displayName": null,
                "id": "1",
                "username": "Administrator"
              },
              "persona": null
            },
            "scheduleStartTime": null,
            "startTime": "2023-08-18T19:49:18.161091Z",
            "status": "WAITING",
            "steps": [
              {
                "error": "",
                "startTime": "2023-08-18T19:49:18.2150748Z",
                "endTime": null
              },
              {
                "error": "",
                "startTime": null,
                "endTime": null
              },
              {
                "error": "",
                "startTime": null,
                "endTime": null
              },
              {
                "error": "",
                "startTime": null,
                "endTime": null
              }
            ]
          }
        }
      ]
    },
    "steps": [
      {
        "description": "Example step 1 update: Initiation",
        "name": "example_step_1"
      },
      {
        "description": "Example step 2 update: Processing",
        "name": "example_step_2"
```


©2026 Tanium Inc. All Rights Reserved

```
          },
          {
            "description": "Example step 3 update: Resolution",
            "name": "example_step_3"
          }
        ],
        "updatedTime": "2023-08-18T20:46:48.99018Z"
      },
      "error": null
    }
  }
}
```

Delete a playbook schedule

(mutation.playbookSchedulePerformOperation)

About this task

Stability: 1.2 - Experimental (Release Candidate)

The `playbookSchedulePerformOperation` mutation requires Automate 1.3 or later and:

- the **Automate Playbook execute** permission on the content set to which the associated playbook belongs, if you created the playbook schedule with your currently active persona, or
- the **Automate Playbook write** permission on the content set to which the associated playbook belongs, if another user created the playbook schedule, or you created the playbook schedule with a different persona than your currently active persona

The following mutation performs the `DELETE` operation on the playbook schedule that matches the ID. This deletes a playbook schedule.

Note: The `PlaybookScheduleOperation` object enumerates available playbook schedule operations.

```
mutation exampleDeletePlaybookScheduleOperation($input:
PlaybookSchedulePerformOperationInput!) {
  playbookSchedulePerformOperation(input: $input) {
    playbookSchedule {
      id
      createdTime
      cron
      enabled
      events {
        createdTime
```

```
        initiator {
          user {
            id
            displayName
            username
          }
          persona {
            id
          }
        }
        name
      }
      playbook {
        id
        description
        name
      }
      scheduledBy {
        user {
          id
          displayName
          username
        }
        persona {
          id
        }
      }
      updatedTime
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "12345678-90ab-cdef-1234-567890abcdef",
    "operation": "DELETE"
  }
}
```

Example response:

```
{
  "data": {
    "playbookSchedulePerformOperation": {
      "playbookSchedule": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "createdTime": "2024-04-12T19:41:43.118622Z",
        "cron": "0 0 1 1 1",
        "enabled": true,
        "events": [],
        "playbook": {
          "id": "fedcba09-8765-4321-fedc-ba0987654321",
          "description": "Example playbook - Scheduling",
          "name": "example_playbook_scheduling"
```

©2026 Tanium Inc. All Rights Reserved

```
        },
        "scheduledBy": {
          "user": {
            "id": "1",
            "displayName": null,
            "username": "admin"
          },
          "persona": null
        },
        "updatedTime": "2024-04-12T20:13:30.603601Z"
      },
      "error": null
    }
  }
}
```

Disable an enabled playbook schedule

(mutation.playbookSchedulePerformOperation)

About this task

Stability: 1.2 - Experimental (Release Candidate)

The `playbookSchedulePerformOperation` mutation requires Automate 1.3 or later and:

- the **Automate Playbook execute** or **Automate Playbook write** permission on the content set to which the associated playbook belongs, if you created the playbook schedule with your currently active persona, or
- the **Automate Playbook write** permission on the content set to which the associated playbook belongs, if another user created the playbook schedule, or you created the playbook schedule with a different persona than your currently active persona

The following mutation performs the `DISABLE` operation on the playbook schedule that matches the ID. This disables an enabled playbook schedule. No playbook runs are started from disabled playbook schedules. This has no effect on disabled playbook schedules.



Note: The `PlaybookScheduleOperation` object enumerates available playbook schedule operations.

```
mutation exampleDisablePlaybookScheduleOperation($input:
PlaybookSchedulePerformOperationInput!) {
  playbookSchedulePerformOperation(input: $input) {
    playbookSchedule {
      id
      createdTime
```

©2026 Tanium Inc. All Rights Reserved

```
      cron
      enabled
      events {
        createdTime
        initiator {
          user {
            id
            displayName
            username
          }
          persona {
            id
          }
        }
        name
      }
      playbook {
        id
        description
        name
      }
      scheduledBy {
        user {
          id
          displayName
          username
        }
        persona {
          id
        }
      }
      updatedTime
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "12345678-90ab-cdef-1234-567890abcdef",
    "operation": "DISABLE"
  }
}
```

Example response:

```
{
  "data": {
    "playbookSchedulePerformOperation": {
      "playbookSchedule": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "createdTime": "2024-04-12T19:41:43.118622Z",
        "cron": "0 0 1 1 1",
        "enabled": false,
        "events": [
```

©2026 Tanium Inc. All Rights Reserved

```
          {
            "createdTime": "2024-12-03T19:17:14.056452452Z",
            "initiator": {
              "user": {
                "id": "1",
                "displayName": null,
                "username": "admin"
              },
              "persona": null
            },
            "name": "DISABLED_BY_USER"
          }
        ],
        "playbook": {
          "id": "fedcba09-8765-4321-fedc-ba0987654321",
          "description": "Example playbook - Scheduling",
          "name": "example_playbook_scheduling"
        },
        "scheduledBy": {
          "user": {
            "id": "1",
            "displayName": null,
            "username": "admin"
          },
          "persona": null
        },
        "updatedTime": "2024-04-12T20:13:30.603601Z"
      },
      "error": null
    }
  }
}
```


Enable a disabled playbook schedule

(mutation.playbookSchedulePerformOperation)

About this task

Stability: 1.2 - Experimental (Release Candidate)

The `playbookSchedulePerformOperation` mutation requires Automate 1.3 or later and:

- the **Automate Playbook execute** permission on the content set to which the associated playbook belongs, if you created the playbook schedule with your currently active persona

The following mutation performs the `ENABLE` operation on the playbook schedule that matches the ID. This enables a disabled playbook schedule. This has no effect on enabled playbook schedules.

©2026 Tanium Inc. All Rights Reserved

Note: The `PlaybookScheduleOperation` object enumerates available playbook schedule operations.

```
mutation exampleEnablePlaybookScheduleOperation($input:
PlaybookSchedulePerformOperationInput!) {
  playbookSchedulePerformOperation(input: $input) {
    playbookSchedule {
      id
      createdTime
      cron
      enabled
      events {
        createdTime
        initiator {
          user {
            id
            displayName
            username
          }
          persona {
            id
          }
        }
        name
      }
      playbook {
        id
        description
        name
      }
      scheduledBy {
        user {
          id
          displayName
          username
        }
        persona {
          id
        }
      }
      updatedTime
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "12345678-90ab-cdef-1234-567890abcdef",
    "operation": "ENABLE"
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "playbookSchedulePerformOperation": {
      "playbookSchedule": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "createdTime": "2024-04-12T19:41:43.118622Z",
        "cron": "0 0 1 1 1",
        "enabled": true,
        "events": [
          {
            "createdTime": "2024-12-03T19:17:14.056452452Z",
            "initiator": {
              "user": {
                "id": "1",
                "displayName": null,
                "username": "admin"
              },
              "persona": null
            },
            "name": "DISABLED_BY_USER"
          }
        ],
        "playbook": {
          "id": "fedcba09-8765-4321-fedc-ba0987654321",
          "description": "Example playbook - Scheduling",
          "name": "example_playbook_scheduling"
        },
        "scheduledBy": {
          "user": {
            "id": "1",
            "displayName": null,
            "username": "admin"
          },
          "persona": null
        },
        "updatedTime": "2024-04-12T20:13:30.603601Z"
      },
      "error": null
    }
  }
}
```


End a playbook run

(mutation.playbookRunPerformOperation)

Stability: 1.2 - Experimental (Release Candidate)

The `playbookRunPerformOperation` mutation requires Automate 1.3 or later and the **Automate Playbook execute** permission on the playbook content set.

©2026 Tanium Inc. All Rights Reserved

Note: To end a playbook run, you must have the same permissions as required to run the playbook.

The following mutation performs the `END` operation on the playbook run that matches the ID. This ends a playbook run.

Note: The `PlaybookRunOperation` object enumerates available playbook run operations.

```
mutation exampleEndPlaybookRunOperation($input: PlaybookRunPerformOperationInput!) {
  playbookRunPerformOperation(
    input: $input
  ) {
    playbookRun {
      createdTime
      endTime
      events {
        createdTime
        initiator {
          persona {
            id
          }
          user {
            id
          }
        }
        name
      }
      id
      playbook {
        id
      }
      runAs {
        persona {
          id
        }
        user {
          id
        }
      }
      scheduleStartTime
      startTime
      status
      steps {
        confirmationResult {
          comment
          completedUser {
            id
          }
          completionTime
        }
        endTime
        playbookStep {
          description
          name
```

©2026 Tanium Inc. All Rights Reserved

```
        }
        startTime
        status
        error
      }
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "fedcba09-8765-4321-fedc-ba0987654321",
    "operation": "END"
  }
}
```

Example response:

```
{
  "data": {
    "playbookRunPerformOperation": {
      "playbookRun": {
        "createdTime": "2023-08-18T19:49:13.113955Z",
        "endTime": null,
        "events": [
          {
            "createdTime": "2023-08-18T12:49:13.1127798-07:00",
            "initiator": {
              "persona": null,
              "user": {
                "id": "1"
              }
            },
            "name": "CREATE"
          },
          {
            "createdTime": "2023-08-18T19:49:13.1167809Z",
            "initiator": null,
            "name": "START"
          },
          {
            "createdTime": "2023-08-18T19:49:13.1397834Z",
            "initiator": null,
            "name": "WAIT"
          }
        ],
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "playbook": {
          "id": "12345678-90ab-cdef-1234-567890abcdef"
        },
        "runAs": {
          "persona": null,
          "user": {
            "id": "1"
          }
        },
        "scheduleStartTime": null,
        "startTime": "2023-08-18T19:49:13.11678Z",
        "status": "ENDING",
```

©2026 Tanium Inc. All Rights Reserved

```
        "steps": [
          {
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 1 update: Initiation",
              "name": "example_step_1"
            },
            "startTime": "2023-08-18T19:49:13.1397834Z",
            "status": "WAITING",
            "error": ""
          },
          {
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 2 update: Processing",
              "name": "example_step_2"
            },
            "startTime": null,
            "status": "NOT_STARTED",
            "error": ""
          },
          {
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 3 update: Resolution",
              "name": "example_step_3"
            },
            "startTime": null,
            "status": "NOT_STARTED",
            "error": ""
          }
        ]
      }
    }
  }
}
```


Pause a running playbook run

(mutation.playbookRunPerformOperation)

About this task

Stability: 1.2 - Experimental (Release Candidate)

The `playbookRunPerformOperation` mutation requires Automate 1.3 or later and the **Automate Playbook execute** permission on the playbook content set.

Note: To pause a playbook run, you must have the same permissions as required to run the playbook.

©2026 Tanium Inc. All Rights Reserved

The following mutation performs the `PAUSE` operation on the playbook run that matches the ID. This pauses a running playbook run.

> **Note:** The `PlaybookRunOperation` object enumerates available playbook run operations.

```
mutation examplePausePlaybookRunOperation($input: PlaybookRunPerformOperationInput!) {
  playbookRunPerformOperation(
    input: $input
  ) {
    playbookRun {
      createdTime
      endTime
      events {
        createdTime
        initiator {
          persona {
            id
          }
          user {
            id
          }
        }
        name
      }
      id
      playbook {
        id
      }
      runAs {
        persona {
          id
        }
        user {
          id
        }
      }
      scheduleStartTime
      startTime
      status
      steps {
        confirmationResult {
          comment
          completedUser {
            id
          }
          completionTime
        }
        endTime
        playbookStep {
          description
          name
        }
        startTime
        status
        error
      }
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "fedcba09-8765-4321-fedc-ba0987654321",
    "operation": "PAUSE"
  }
}
```

Example response:

```
{
  "data": {
    "playbookRunPerformOperation": {
      "playbookRun": {
        "createdTime": "2023-08-18T19:49:13.113955Z",
        "endTime": null,
        "events": [
          {
            "createdTime": "2023-08-18T12:49:13.1127798-07:00",
            "initiator": {
              "persona": null,
              "user": {
                "id": "1"
              }
            },
            "name": "CREATE"
          },
          {
            "createdTime": "2023-08-18T19:49:13.1167809Z",
            "initiator": null,
            "name": "START"
          },
          {
            "createdTime": "2023-08-18T19:49:13.1397834Z",
            "initiator": null,
            "name": "WAIT"
          }
        ],
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "playbook": {
          "id": "12345678-90ab-cdef-1234-567890abcdef"
        },
        "runAs": {
          "persona": null,
          "user": {
            "id": "1"
          }
        },
        "scheduleStartTime": null,
        "startTime": "2023-08-18T19:49:13.11678Z",
        "status": "PAUSING",
        "steps": [
          {
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 1 update: Initiation",
              "name": "example_step_1"
```

©2026 Tanium Inc. All Rights Reserved

```
            },
            "startTime": "2023-08-18T19:49:13.1397834Z",
            "status": "WAITING",
            "error": ""
          },
          {
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 2 update: Processing",
              "name": "example_step_2"
            },
            "startTime": null,
            "status": "NOT_STARTED",
            "error": ""
          },
          {
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 3 update: Resolution",
              "name": "example_step_3"
            },
            "startTime": null,
            "status": "NOT_STARTED",
            "error": ""
          }
        ]
      }
    }
  }
}
```


Resume a paused playbook run

(mutation.playbookRunPerformOperation)

About this task

Stability: 1.2 - Experimental (Release Candidate)

The `playbookRunPerformOperation` mutation requires Automate 1.3 or later and the **Automate Playbook execute** permission on the playbook content set.



Note: To resume a playbook run, you must have the same permissions as required to run the playbook.

The following mutation performs the `RESUME` operation on the playbook run that matches the ID. This resumes a paused playbook run.

©2026 Tanium Inc. All Rights Reserved

Note: The `PlaybookRunOperation` object enumerates available playbook run operations.

```
mutation exampleResumePlaybookRunOperation($input: PlaybookRunPerformOperationInput!) {
  playbookRunPerformOperation(
    input: $input
  ) {
    playbookRun {
      createdTime
      endTime
      events {
        createdTime
        initiator {
          persona {
            id
          }
          user {
            id
          }
        }
        name
      }
      id
      playbook {
        id
      }
      runAs {
        persona {
          id
        }
        user {
          id
        }
      }
      scheduleStartTime
      startTime
      status
      steps {
        confirmationResult {
          comment
          completedUser {
            id
          }
          completionTime
        }
        endTime
        playbookStep {
          description
          name
        }
        startTime
        status
        error
      }
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
```

```
  "input": {
    "id": "fedcba09-8765-4321-fedc-ba0987654321",
    "operation": "RESUME"
  }
}
```

Example response:

```
{
  "data": {
    "playbookRunPerformOperation": {
      "playbookRun": {
        "createdTime": "2023-08-18T19:49:13.113955Z",
        "endTime": null,
        "events": [
          {
            "createdTime": "2023-08-18T12:49:13.1127798-07:00",
            "initiator": {
              "persona": null,
              "user": {
                "id": "1"
              }
            },
            "name": "CREATE"
          },
          {
            "createdTime": "2023-08-18T19:49:13.1167809Z",
            "initiator": null,
            "name": "START"
          },
          {
            "createdTime": "2023-08-18T19:49:13.1397834Z",
            "initiator": null,
            "name": "WAIT"
          }
        ],
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "playbook": {
          "id": "12345678-90ab-cdef-1234-567890abcdef"
        },
        "runAs": {
          "persona": null,
          "user": {
            "id": "1"
          }
        },
        "scheduleStartTime": null,
        "startTime": "2023-08-18T19:49:13.11678Z",
        "status": "RESUMING",
        "steps": [
          {
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 1 update: Initiation",
              "name": "example_step_1"
            },
            "startTime": "2023-08-18T19:49:13.1397834Z",
            "status": "WAITING",
            "error": ""
          },
          {
```

©2026 Tanium Inc. All Rights Reserved

```
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 2 update: Processing",
              "name": "example_step_2"
            },
            "startTime": null,
            "status": "NOT_STARTED",
            "error": ""
          },
          {
            "confirmationResult": null,
            "endTime": null,
            "playbookStep": {
              "description": "Example step 3 update: Resolution",
              "name": "example_step_3"
            },
            "startTime": null,
            "status": "NOT_STARTED",
            "error": ""
          }
        ]
      }
    }
  }
}
```

Update playbook content set membership

(mutation.playbookContentSetUpdate)

About this task

Stability: 1.2 - Experimental (Release Candidate)

The `playbookContentSetUpdate` mutation requires Automate 1.5 or later and the **Automate Playbook write** permission on the playbook content set to update a playbook content set membership.

The following mutation updates the playbook with the ID `12345678-90ab-cdef-1234-567890abcdef` to belong to the content set with the ID `2`:

```
mutation exampleChangePlaybookContentSet($input: PlaybookContentSetUpdateInput!) {
  playbookContentSetUpdate(input: $input) {
    playbook {
      id
      name
      contentSet {
        id
        name
        description
      }
      description
      updatedTime
    }
```

©2026 Tanium Inc. All Rights Reserved

```
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "12345678-90ab-cdef-1234-567890abcdef",
    "contentSet": {
      "id": "2"
    }
  }
}
```

Example response:

```
{
  "data": {
    "playbookContentSetUpdate": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example playbook content set update",
        "contentSet": {
          "id": "2",
          "name": "Default",
          "description": null
        },
        "description": "example description",
        "updatedTime": "2024-11-15T20:07:29.166156Z"
      },
      "error": null
    }
  }
}
```

Add Install Patch step to a playbook

(mutation.upsertPlaybook (input.steps))

About this task

Stability: 1.1 - Experimental (Active Development)

The `playbookUpsert` mutation requires Automate 1.3 or later and the **Automate Playbook write** permission on the playbook content set to create or update a playbook.

The **Install Patch** step requires Patch 3.16.191 or later and the following permissions:

©2026 Tanium Inc. All Rights Reserved

- **Computer Group read** permission
- **Patch Deployment execute** permission
- **Patch Deployment write** permission
- **Sensor read** permission

Depending on the deployment, the **Install Patch** step might require additional Patch permissions.

The **Install Patch** step deploys a patch to targeted endpoints. The **Install Patch** step always creates a one-time deployment. The patch content download and patch deployment both start when the **Install Patch** step starts. You cannot configure an **Install Patch** step to create an ongoing patch deployment, configure a start time or end time, or configure patch content downloaded to endpoints prior to the patch deployment start.



Note: The Install Patch step does not support uploading title icons or body images for end user notifications.

The following mutation creates a playbook, because no ID is specified in the input, and defines an **Install Patch** step that deploys a Windows patch on the targeted endpoints belonging to the **All Windows** computer group:

```
mutation exampleUpsertPlaybookInstallPatch($input: PlaybookUpsertInput!) {
  playbookUpsert(input: $input) {
    playbook {
      id
      name
      steps {
        description
        name
        taskArgument {
          __typename
        }
      }
    }
    isNewRecord
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "input": {
    "name": "example_playbook_install_patch",
    "description": "Example playbook - Install Patch step",
    "contentSet": {
      "id": "123"
    },
    "notificationSettings": {
      "email": {
        "events": [
          "START",
          "END"
        ],
        "recipients": [
          "host@example.com"
        ]
      }
    },
    "startingStepID": "fedcba09-8765-4321-fedc-ba0987654321",
    "steps": [
      {
        "id": "fedcba09-8765-4321-fedc-ba0987654321",
        "name": "example_install_patch_step",
        "description": "Example step 1: Install Patch on Windows endpoints",
        "endpointTargeting": {
          "customTargeting": {
            "memberOf": {
              "name": "All Windows"
            },
            "negated": false,
            "any": true
          }
        },
        "taskArgument": {
          "patchInstallDeployment": {
            "completionPercentage": 1,
            "patchDeploymentConfig": {
              "contentDeploymentType": "MANUAL_SELECTION",
              "contentSet": {
                "id": "1"
              },
              "description": "example Patch deployment for Windows",
              "distributeOver": {
                "seconds": 900
              },
              "eussHideFromActivity": false,
              "name": "example_deployment_windows",
              "overrideBlocklists": false,
              "overrideMaintenanceWindows": false,
              "patches": [],
              "patchLists": [
                {
                  "id": "12",
                  "version": null
                }
              ],
              "platform": "WINDOWS",
              "postDeploymentNotification": {
                "body": "Example body",
                "countdownToDeadline": {
                  "seconds": 600
                },
                "includeFrame": false,
                "notificationPeriodDuration": {
```

©2026 Tanium Inc. All Rights Reserved

```
                "seconds": 7200
              },
              "postponeOptions": {
                "durationOne": {
                  "seconds": 1800
                },
                "durationTwo": {
                  "seconds": 1200
                },
                "durationThree": {
                  "seconds": 600
                }
              },
              "title": "Example title",
              "translations": [
                {
                  "body": "ejemplo",
                  "languageCode": "es",
                  "title": "ejemplo"
                }
              ]
            },
            "restart": true
          },
          "deploymentDuration": {
            "nanoseconds": 0,
            "seconds": 86400
          },
          "skipVerification": false
        }
      }
    }
  ]
 }
}
```

Example response:

```
{
  "data": {
    "playbookUpsert": {
      "playbook": {
        "id": "12345678-90ab-cdef-1234-567890abcdef",
        "name": "example_playbook_install_patch",
        "steps": [
          {
            "description": "Example step 1: Install Patch on Windows endpoints",
            "name": "example_install_patch_step",
            "taskArgument": {
              "__typename": "AutomateTaskArgumentPatchInstallDeployment"
            }
          }
        ]
      },
      "isNewRecord": true,
      "error": null
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Benchmark examples

The following queries retrieve endpoints and use Benchmark to also retrieve related risk overview or risk vector information.

Certain risk vector queries require additional Tanium solutions. For more information, see Tanium Benchmark User Guide: Solution dependencies.

Get endpoint with Risk Score 2.0 information

(query.endpoints.edges.node.sensorReadings (sensors.filters))

About this task

The following query retrieves the first endpoint with associated Benchmark overview information.

```
query exampleGetEndpoints($first: Int) {
  endpoints(first: $first) {
    edges {
      node {
        name
        sensorReadings(sensors: [{ name: "Tanium Risk Score" }]) {
          columns {
            sensor {
              name
            }
            name
            values
          }
        }
      }
    }
  }
}
```

Include a variable for the number of endpoints to return in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-endpoint",
```

©2026 Tanium Inc. All Rights Reserved

```
          "sensorReadings": {
            "columns": [
              {
                "sensor": {
                  "name": "Tanium Risk Score"
                },
                "name": "Risk Score",
                "values": [
                  "438"
                ]
              },
              {
                "sensor": {
                  "name": "Tanium Risk Score"
                },
                "name": "Critical Severity Vulnerabilities Score",
                "values": [
                  "665"
                ]
              },
              {
                "sensor": {
                  "name": "Tanium Risk Score"
                },
                "name": "High Severity Vulnerabilities Score",
                "values": [
                  "665"
                ]
              },
              {
                "sensor": {
                  "name": "Tanium Risk Score"
                },
                "name": "Compliance Failures Score",
                "values": [
                  "486"
                ]
              },
              {
                "sensor": {
                  "name": "Tanium Risk Score"
                },
                "name": "Lateral Movement Risk Score",
                "values": [
                  ""
                ]
              },
              {
                "sensor": {
                  "name": "Tanium Risk Score"
                },
                "name": "Insecure TLS/SSL Score",
                "values": [
                  "0"
                ]
              },
              {
                "sensor": {
                  "name": "Tanium Risk Score"
                },
                "name": "Expired Certificates Score",
                "values": [
                  "0"
                ]
```


©2026 Tanium Inc. All Rights Reserved

```
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Comply examples

The following queries retrieve endpoints and use Comply to also retrieve associated compliance findings or vulnerability findings.

Comply requests have the following feature-specific dependencies at the specified minimum versions:

- Comply 2.10.940 or later is required to submit a request involving the System Vulnerability and System Compliance risk vectors.
- Comply 2.16.97 or later is required to submit a request that filters specific System Vulnerability and System Compliance risk vector fields.
- Gateway 1.10 or later is required to submit a request involving CISA KEV catalog information.

Get endpoints with compliance findings information

(query.endpoints.edges.node.compliance.complianceFindings)

GET ENDPOINT COMPLIANCE FINDINGS INFORMATION

(query.endpoints.edges.node.compliance.complianceFindings)

The following query retrieves the first endpoint and associated compliance findings.

```
query exampleEndpointComplianceFindings($first: Int!) {
  endpoints(first: $first) {
    edges {
      node {
        name
        ipAddress
        compliance {
          complianceFindings {
            category
            excepted
            firstFoundDate
```

©2026 Tanium Inc. All Rights Reserved

```
            id
            lastScanDate
            profile
            profileVersion
            rule
            ruleId
            standard
            standardVersion
            state
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the cursor variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "comply-findings",
            "ipAddress": "192.0.2.10",
            "compliance": {
              "complianceFindings": [
                {
                  "category": "Fail",
                  "excepted": false,
                  "firstFoundDate": "2023-01-01",
                  "id": "CIS Microsoft Windows 10 Enterprise Release 20H2 Benchmark;1.10.1;MB
Custom
Win10;xccdf_com.tanium.comply_tailoring_1639024882628;xccdf_org.cisecurity.benchmarks_rule_18
.9.4.1_L2_Ensure_Allow_a_Windows_app_to_share_application_data_between_users_is_set_to_Disabl
ed",
                  "lastScanDate": "2023-02-02",
                  "profile": "MB Custom Win10",
                  "profileVersion": "xccdf_com.tanium.comply_tailoring_1639024882628",
                  "rule": "(L2) Ensure 'Allow a Windows app to share application data between
users' is set to 'Disabled'",
                  "ruleId":
"xccdf_org.cisecurity.benchmarks_rule_18.9.4.1_L2_Ensure_Allow_a_Windows_app_to_share_applica
tion_data_between_users_is_set_to_Disabled",
                  "standard": "CIS Microsoft Windows 10 Enterprise Release 20H2 Benchmark",
                  "standardVersion": "1.10.1",
                  "state": "fail"
                }
```

©2026 Tanium Inc. All Rights Reserved

```
              ]
            }
          }
        }
      ],
      "pageInfo": {
        "startCursor": "NTc2NTM4MDow",
        "endCursor": "NTc2NTM4MDoxOQ==",
        "hasPreviousPage": false,
        "hasNextPage": true
      }
    }
  }
}
```

GET FILTERED ENDPOINT COMPLIANCE FINDINGS INFORMATION

(query.endpoints.edges.node.compliance.complianceFindings)

The following query requires Comply 2.16.97 or later and retrieves the first endpoint with a compliance finding `category` of `Error` and `FirstFoundDate` greater than or equal to `2025-07-10` and then returns the associated compliance findings for that endpoint with a `category` of `Error`.

```
query exampleFilterComplianceFindings($first: Int) {
  endpoints(first: $first) {
    edges {
      node {
        name
        ipAddress
        compliance {
          complianceFindings(
            filter: {any: false, filters: [{path: "category", value: "Error"}, {path:
"firstFoundDate", op: GTE, value: "2025-07-10"}]}
          ) {
            category
            firstFoundDate
            id
            lastScanDate
            profile
            profileVersion
            rule
            ruleId
            standard
            standardVersion
            state
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "IsolatedVM-Pdejesus",
            "ipAddress": "192.10.1.102",
            "compliance": {
              "complianceFindings": [
                {
                  "category": "Error",
                  "firstFoundDate": "2025-07-12",
                  "id": "Microsoft Windows 11 STIG SCAP Benchmark;002.001;I - Mission
Critical Public;1;xccdf_mil.disa.stig_rule_SV-253275r958478_rule",
                  "lastScanDate": "2025-07-12",
                  "profile": "I - Mission Critical Public",
                  "profileVersion": "1",
                  "rule": "Internet Information System (IIS) or its subcomponents must not be
installed on a workstation.",
                  "ruleId": "xccdf_mil.disa.stig_rule_SV-253275r958478_rule",
                  "standard": "Microsoft Windows 11 STIG SCAP Benchmark",
                  "standardVersion": "002.001",
                  "state": "error"
                },
                {
                  "category": "Error",
                  "firstFoundDate": "2025-07-12",
                  "id": "Microsoft Windows 11 STIG SCAP Benchmark;002.001;I - Mission
Critical Public;1;xccdf_mil.disa.stig_rule_SV-253278r958480_rule",
                  "lastScanDate": "2025-07-12",
                  "profile": "I - Mission Critical Public",
                  "profileVersion": "1",
                  "rule": "The Telnet Client must not be installed on the system.",
                  "ruleId": "xccdf_mil.disa.stig_rule_SV-253278r958480_rule",
                  "standard": "Microsoft Windows 11 STIG SCAP Benchmark",
                  "standardVersion": "002.001",
                  "state": "error"
                },
                {
                  "category": "Error",
                  "firstFoundDate": "2025-07-12",
                  "id": "Microsoft Windows 11 STIG SCAP Benchmark;002.001;I - Mission
Critical Public;1;xccdf_mil.disa.stig_rule_SV-253287r958478_rule",
                  "lastScanDate": "2025-07-12",
                  "profile": "I - Mission Critical Public",
                  "profileVersion": "1",
                  "rule": "The Server Message Block (SMB) v1 protocol must be disabled on the
SMB server.",
                  "ruleId": "xccdf_mil.disa.stig_rule_SV-253287r958478_rule",
                  "standard": "Microsoft Windows 11 STIG SCAP Benchmark",
```

```
                  "standardVersion": "002.001",
                  "state": "error"
                },
                {
                  "category": "Error",
                  "firstFoundDate": "2025-07-12",
                  "id": "Microsoft Windows 11 STIG SCAP Benchmark;002.001;I - Mission
Critical Public;1;xccdf_mil.disa.stig_rule_SV-253279r958480_rule",
                  "lastScanDate": "2025-07-12",
                  "profile": "I - Mission Critical Public",
                  "profileVersion": "1",
                  "rule": "The TFTP Client must not be installed on the system.",
                  "ruleId": "xccdf_mil.disa.stig_rule_SV-253279r958480_rule",
                  "standard": "Microsoft Windows 11 STIG SCAP Benchmark",
                  "standardVersion": "002.001",
                  "state": "error"
                },
                {
                  "category": "Error",
                  "firstFoundDate": "2025-07-12",
                  "id": "Microsoft Windows 11 STIG SCAP Benchmark;002.001;I - Mission
Critical Public;1;xccdf_mil.disa.stig_rule_SV-253285r958478_rule",
                  "lastScanDate": "2025-07-12",
                  "profile": "I - Mission Critical Public",
                  "profileVersion": "1",
                  "rule": "The Windows PowerShell 2.0 feature must be disabled on the
system.",
                  "ruleId": "xccdf_mil.disa.stig_rule_SV-253285r958478_rule",
                  "standard": "Microsoft Windows 11 STIG SCAP Benchmark",
                  "standardVersion": "002.001",
                  "state": "error"
                },
                {
                  "category": "Error",
                  "firstFoundDate": "2025-07-12",
                  "id": "Microsoft Windows 11 STIG SCAP Benchmark;002.001;I - Mission
Critical Public;1;xccdf_mil.disa.stig_rule_SV-253286r958478_rule",
                  "lastScanDate": "2025-07-12",
                  "profile": "I - Mission Critical Public",
                  "profileVersion": "1",
                  "rule": "The Server Message Block (SMB) v1 protocol must be disabled on the
system.",
                  "ruleId": "xccdf_mil.disa.stig_rule_SV-253286r958478_rule",
                  "standard": "Microsoft Windows 11 STIG SCAP Benchmark",
                  "standardVersion": "002.001",
                  "state": "error"
                },
                {
                  "category": "Error",
                  "firstFoundDate": "2025-07-12",
                  "id": "Microsoft Windows 11 STIG SCAP Benchmark;002.001;I - Mission
Critical Public;1;xccdf_mil.disa.stig_rule_SV-253288r958478_rule",
                  "lastScanDate": "2025-07-12",
                  "profile": "I - Mission Critical Public",
                  "profileVersion": "1",
                  "rule": "The Server Message Block (SMB) v1 protocol must be disabled on the
SMB client.",
                  "ruleId": "xccdf_mil.disa.stig_rule_SV-253288r958478_rule",
                  "standard": "Microsoft Windows 11 STIG SCAP Benchmark",
                  "standardVersion": "002.001",
                  "state": "error"
                },
                {
                  "category": "Error",
```


©2026 Tanium Inc. All Rights Reserved

```
                  "firstFoundDate": "2025-07-12",
                  "id": "Microsoft Windows 11 STIG SCAP Benchmark;002.001;I - Mission
Critical Public;1;xccdf_mil.disa.stig_rule_SV-256893r958552_rule",
                  "lastScanDate": "2025-07-12",
                  "profile": "I - Mission Critical Public",
                  "profileVersion": "1",
                  "rule": "Internet Explorer must be disabled for Windows 11.",
                  "ruleId": "xccdf_mil.disa.stig_rule_SV-256893r958552_rule",
                  "standard": "Microsoft Windows 11 STIG SCAP Benchmark",
                  "standardVersion": "002.001",
                  "state": "error"
                }
              ]
            }
          }
        }
      ],
      "pageInfo": {
        "startCursor": "MTQ2MDIwMzQ3OjA=",
        "endCursor": "MTQ2MDIwMzQ3OjA=",
        "hasPreviousPage": false,
        "hasNextPage": true
      }
    }
  }
}
```

GET FILTERED ENDPOINT COMPLIANCE FINDINGS INFORMATION FROM FILTERED SET OF ENDPOINTS

(query.endpoints.edges.node.compliance.complianceFindings)

The following query requires Comply 2.16.97 or later and retrieves the first endpoint that is a member of the All Windows computer group, with a compliance findings `category` of `Error` and `standardVersion` value of `1.1.1`, and returns that endpoint's associated compliance findings with a `category` of `Error` and `standardVersion` value of `1.1.1`.

```
query exampleFilterWindowsComplianceFindings($first: Int!) {
  endpoints(first: $first, filter: {memberOf: {name: "All Windows"}}) {
    edges {
      node {
        name
        ipAddress
        compliance {
          complianceFindings(
            filter: {any: false, filters: [{path: "category", value: "Error"}, {path:
"standardVersion", value: "1.1.1"}]}
          ) {
            category
            firstFoundDate
            id
            lastScanDate
            profile
            profileVersion
            rule
            ruleId
            standard
            standardVersion
```

```
            state
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the cursor variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host-2",
            "ipAddress": "192.0.2.20",
            "compliance": {
              "complianceFindings": [
                {
                  "category": "Error",
                  "firstFoundDate": "2023-01-01",
                  "id": "CIS Microsoft IIS 10 Benchmark;1.1.1;Level 1 - IIS
10;1;xccdf_org.cisecurity.benchmarks_rule_4.6_L1_Ensure_HTTP_Trace_Method_is_disabled",
                  "lastScanDate": "2023-02-02",
                  "profile": "Level 1 - IIS 10",
                  "profileVersion": "1",
                  "rule": "(L1) Ensure 'HTTP Trace Method' is disabled",
                  "ruleId":
"xccdf_org.cisecurity.benchmarks_rule_4.6_L1_Ensure_HTTP_Trace_Method_is_disabled",
                  "standard": "CIS Microsoft IIS 10 Benchmark",
                  "standardVersion": "1.1.1",
                  "state": "error"
                },
                {
                  "category": "Error",
                  "firstFoundDate": "2023-01-01",
                  "id": "CIS Microsoft IIS 10 Benchmark;1.1.1;Level 1 - IIS
10;1;xccdf_org.cisecurity.benchmarks_rule_3.4_L1_Ensure_IIS_HTTP_detailed_errors_are_hidden_f
rom_displaying_remotely",
                  "lastScanDate": "2023-02-02",
                  "profile": "Level 1 - IIS 10",
                  "profileVersion": "1",
                  "rule": "(L1) Ensure IIS HTTP detailed errors are hidden from displaying
remotely",
                  "ruleId":
"xccdf_org.cisecurity.benchmarks_rule_3.4_L1_Ensure_IIS_HTTP_detailed_errors_are_hidden_from_
displaying_remotely",
```

©2026 Tanium Inc. All Rights Reserved

```
                  "standard": "CIS Microsoft IIS 10 Benchmark",
                  "standardVersion": "1.1.1",
                  "state": "error"
                }
              ]
            }
          }
        }
      ],
      "pageInfo": {
        "startCursor": "NjA5ODg0MDow",
        "endCursor": "NjA5ODg0MDow",
        "hasPreviousPage": false,
        "hasNextPage": true
      }
    }
  }
}
```


Get endpoints with vulnerability findings information

(query.endpoints.edges.node.compliance.cveFindings)

GET ENDPOINT VULNERABILITY FINDINGS INFORMATION

(query.endpoints.edges.node.compliance.cveFindings)

The following query retrieves the first endpoint and associated vulnerability findings, including CVSS v2 score and severity, and related CISA KEV information.

```
query exampleEndpointVulnerabilityCVSSv2($first: Int!) {
  endpoints(first: $first) {
    edges {
      node {
        name
        ipAddress
        compliance {
          cveFindings {
            absoluteFirstFoundDate
            affectedProducts
            cisaDateAdded
            cisaDueDate
            cisaNotes
            cisaProduct
            cisaRequiredAction
            cisaShortDescription
            cisaVendor
            cisaVulnerabilityName
            cpes
            cveId
            cveYear
            cvssScore
            detectedProducts
            detectedCPEs
            excepted
            firstFound
```

©2026 Tanium Inc. All Rights Reserved

```
            isCisaKev
            lastFound
            scanType
            severity
            summary
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host-one",
            "ipAddress": "192.10.80.10",
            "compliance": {
              "cveFindings": [
                {
                  "absoluteFirstFoundDate": "2024-07-03",
                  "affectedProducts": [
                    ""
                  ],
                  "cisaDateAdded": null,
                  "cisaDueDate": null,
                  "cisaNotes": "",
                  "cisaProduct": "",
                  "cisaRequiredAction": "",
                  "cisaShortDescription": "",
                  "cisaVendor": "",
                  "cisaVulnerabilityName": "",
                  "cpes": [
                    ""
                  ],
                  "cveId": "CVE-2022-1966",
                  "cveYear": "2022",
                  "cvssScore": null,
                  "detectedProducts": [
                    "kernel",
                    "kernel-core",
                    "kernel-modules",
                    "oraclelinux-release"
                  ],
```

©2026 Tanium Inc. All Rights Reserved

```
                  "detectedCPEs": [
                    ""
                  ],
                  "excepted": false,
                  "firstFound": "2024-07-03",
                  "isCisaKev": false,
                  "lastFound": "2025-07-11",
                  "scanType": "oval",
                  "severity": "Unscored",
                  "summary": "A description for this vulnerability has not yet been
provided."
                }
              ]
            }
          }
        }
      ],
      "pageInfo": {
        "startCursor": "MTQ2NTkwMjQzOjA=",
        "endCursor": "MTQ2NTkwMjQzOjA=",
        "hasPreviousPage": false,
        "hasNextPage": true
      }
    }
  }
}
```

GET ENDPOINT VULNERABILITY FINDINGS INFORMATION (CVSS V3 AND CISA KEV)

(query.endpoints.edges.node.compliance.cveFindings)

The following query retrieves the first endpoint and associated vulnerability findings, including CVSS v3 score and severity, and related CISA KEV information.

```
query exampleEndpointVulnerabilityCVSSv3($first: Int!) {
  endpoints(first: $first) {
    edges {
      node {
        name
        ipAddress
        compliance {
          cveFindings {
            absoluteFirstFoundDate
            affectedProducts
            cisaDateAdded
            cisaDueDate
            cisaNotes
            cisaProduct
            cisaRequiredAction
            cisaShortDescription
            cisaVendor
            cisaVulnerabilityName
            cpes
            cveId
            cveYear
            cvssScoreV3
            cvssTemporalScoreV3
            detectedProducts
            detectedCPEs
```

©2026 Tanium Inc. All Rights Reserved

```
            excepted
            firstFound
            isCisaKev
            lastFound
            scanType
            severityV3
            summary
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example.host.two",
            "ipAddress": "1.2.3.4",
            "compliance": {
              "cveFindings": [
                {
                  "absoluteFirstFoundDate": "2025-10-30",
                  "affectedProducts": [
                    "Microsoft .NET Framework",
                    "Microsoft .NET Framework 4.7.2"
                  ],
                  "cisaDateAdded": "2025-10-14",
                  "cisaDueDate": "2025-11-04",
                  "cisaNotes": "https://msrc.microsoft.com/update-guide/en-US/vulnerability/
CVE-2025-24990 ; https://nvd.nist.gov/vuln/detail/CVE-2025-24990",
                  "cisaProduct": "Windows",
                  "cisaRequiredAction": "Apply mitigations per vendor instructions, follow
applicable BOD 22-01 guidance for cloud services, or discontinue use of the product if
mitigations are unavailable.",
                  "cisaShortDescription": "Microsoft Windows Agere Modem Driver contains an
untrusted pointer dereference vulnerability that allows for privilege escalation. An attacker
who successfully exploited this vulnerability could gain administrator privileges.",
                  "cisaVendor": "Microsoft",
                  "cisaVulnerabilityName": "Microsoft Windows Untrusted Pointer Dereference
Vulnerability",
                  "cpes": [
                    "cpe:/o:microsoft:windows_10_1507:::~~~~x64~",
                    "cpe:/o:microsoft:windows_10_1607:::~~~~x64~",
```

©2026 Tanium Inc. All Rights Reserved

```
                    "cpe:/o:microsoft:windows_10_1809:::~~~~x64~",
                    "cpe:/o:microsoft:windows_10_21h2:::~~~~x64~",
                    "cpe:/o:microsoft:windows_10_22h2:::~~~~x64~",
                    "cpe:/o:microsoft:windows_11_22h2:::~~~~x64~",
                    "cpe:/o:microsoft:windows_11_23h2:::~~~~x64~",
                    "cpe:/o:microsoft:windows_11_24h2:::~~~~x64~",
                    "cpe:/o:microsoft:windows_11_25h2:::~~~~x64~",
                    "cpe:/o:microsoft:windows_server_2008:-:sp2:~~~~x64~",
                    "cpe:/o:microsoft:windows_server_2008:r2:sp1:~~~~x64~",
                    "cpe:/o:microsoft:windows_server_2012:-",
                    "cpe:/o:microsoft:windows_server_2012:r2",
                    "cpe:/o:microsoft:windows_server_2016",
                    "cpe:/o:microsoft:windows_server_2019",
                    "cpe:/o:microsoft:windows_server_2022",
                    "cpe:/o:microsoft:windows_server_2022_23h2",
                    "cpe:/o:microsoft:windows_server_2025"
                  ],
                  "cveId": "CVE-2025-24990",
                  "cveYear": "2025",
                  "cvssScoreV3": 7.8,
                  "cvssTemporalScoreV3": 7.8,
                  "detectedProducts": [
                    "Operating System"
                  ],
                  "detectedCPEs": [
                    "cpe:/o:microsoft:windows_11:23h2::x64"
                  ],
                  "excepted": false,
                  "firstFound": "2025-10-30",
                  "isCisaKev": true,
                  "lastFound": "2025-12-03",
                  "scanType": "oval",
                  "severityV3": "High",
                  "summary": "Microsoft is aware of vulnerabilities in the third party Agere
Modem driver that ships natively with supported Windows operating systems.  This is an
announcement of the upcoming removal of ltmdm64.sys driver.  The driver has been removed in
the October cumulative update.\nFax modem hardware dependent on this specific driver will no
longer work on Windows.\nMicrosoft recommends removing any existing dependencies on this
hardware."
                }
              ]
            }
          }
        }
      ],
      "pageInfo": {
        "startCursor": "ODU5Nzk2MDow",
        "endCursor": "ODU5Nzk2MDow",
        "hasPreviousPage": false,
        "hasNextPage": true
      }
    }
  }
}
```


GET ENDPOINT VULNERABILITY FINDINGS INFORMATION (CVSS V4 AND CISA KEV)

The following query retrieves the first endpoint and associated vulnerability findings, including CVSS v4 score and severity, and related CISA KEV information.

©2026 Tanium Inc. All Rights Reserved

```
query exampleEndpointVulnerabilityCVSSv4($first: Int!) {
  endpoints(first: $first) {
    edges {
      node {
        name
        ipAddress
        compliance {
          cveFindings {
            absoluteFirstFoundDate
            affectedProducts
            cpes
            cveId
            cveYear
            cvssScoreV4
            firstFound
            lastFound
            scanType
            cvssSeverityV4
            cvssThreatScoreV4
            cvssVectorV4
            cwes {
              id
              name
            }
            mitreAttacks {
              id
              name
            }
            alias
            epssScore
            epssPercentile
            epssPercentileRange
            maxMaturity
            createdDate
            modifiedDate
            mitreLink
            nistLink
            secpodLink
            solutionLinks
            remediation
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:

©2026 Tanium Inc. All Rights Reserved

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host-one",
            "ipAddress": "192.168.4.100",
            "compliance": {
              "cveFindings": [
                {
                  "absoluteFirstFoundDate": "2025-10-31",
                  "affectedProducts": [
                    "Oracle Coherence",
                    "netty"
                  ],
                  "cpes": [
                    "cpe:/a:netty:netty"
                  ],
                  "cveId": "CVE-2025-55163",
                  "cveYear": "2025",
                  "cvssScoreV4": 8.2,
                  "firstFound": "2025-10-31",
                  "lastFound": "2025-11-25",
                  "scanType": "sbom",
                  "cvssSeverityV4": "High",
                  "cvssThreatScoreV4": 6.9,
                  "cvssVectorV4": "CVSS:4.0/AV:N/AC:L/AT:P/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/
SI:N/SA:N/E:X/CR:X/IR:X/AR:X/MAV:X/MAC:X/MAT:X/MPR:X/MUI:X/MVC:X/MVI:X/MVA:X/MSC:X/MSI:X/
MSA:X/S:X/AU:X/R:X/V:X/RE:X/U:X",
                  "cwes": [
                    {
                      "id": "CWE-770",
                      "name": "Allocation of Resources Without Limits or Throttling"
                    }
                  ],
                  "mitreAttacks": [
                    {
                      "id": "T1210",
                      "name": "T1210"
                    }
                  ],
                  "alias": "MadeYouReset",
                  "epssScore": 0.00076,
                  "epssPercentile": 0.19072,
                  "epssPercentileRange": "0-25th percentile",
                  "maxMaturity": "Proof of Concept",
                  "createdDate": "2025-08-13",
                  "modifiedDate": "2025-11-04",
                  "mitreLink": "https://www.cve.org/CVERecord?id=CVE-2025-55163",
                  "nistLink": "https://nvd.nist.gov/vuln/detail/CVE-2025-55163",
                  "secpodLink": "https://scaprepo.com/view.jsp?id=CVE-2025-55163",
                  "solutionLinks": [
                    "https://github.com/netty/netty/security/advisories/GHSA-prj3-ccx8-p6x4",
                    "http://www.openwall.com/lists/oss-security/2025/08/16/1",
                    "https://www.kb.cert.org/vuls/id/767506"
                  ],
                  "remediation": "The vendor identifies the following advisory that may
contain remediation information. Apply the patches as necessary: https://github.com/netty/
netty/security/advisories/GHSA-prj3-ccx8-p6x4,http://www.openwall.com/lists/oss-security/
2025/08/16/1,https://www.kb.cert.org/vuls/id/767506"
                }
              ]
            }
```


©2026 Tanium Inc. All Rights Reserved

```
          }
        }
      ],
      "pageInfo": {
        "startCursor": "MjE1ODU0NjUzOjA=",
        "endCursor": "MjE1ODU0NjUzOjA=",
        "hasPreviousPage": false,
        "hasNextPage": true
      }
    }
  }
}
```

GET FILTERED ENDPOINT VULNERABILITY FINDINGS INFORMATION

(query.endpoints.edges.node.compliance.cveFindings)

The following query requires Comply 2.16.97 or later and retrieves the first endpoint with a vulnerability findings `cveYear` of `2020` and `firstFound` greater than or equal to `2025-07-10` and then returns the associated vulnerability findings for that endpoint with a finding `cveYear` of `2020`.

```
query exampleFilterEndpointVulnerability($first: Int!) {
  endpoints(first: $first) {
    edges {
      node {
        name
        ipAddress
        compliance {
          cveFindings(
            filter: {any: false, filters: [{path: "cveYear", value: "2020"}, {path:
"firstFound", op: GTE, value: "2025-07-10"}]}
          ) {
            absoluteFirstFoundDate
            cveId
            cveYear
            cvssScore
            detectedProducts
            detectedCPEs
            firstFound
            lastFound
            severity
            summary
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the cursor variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Example response:


```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host-1",
            "ipAddress": "192.30.80.203",
            "compliance": {
              "cveFindings": [
                {
                  "absoluteFirstFoundDate": "2025-07-14",
                  "cveId": "CVE-2020-0817",
                  "cveYear": "2020",
                  "cvssScore": null,
                  "detectedProducts": [
                    "Operating System"
                  ],
                  "detectedCPEs": [
                    "cpe:/o:microsoft:windows_10:1507::x64"
                  ],
                  "firstFound": "2025-07-14",
                  "lastFound": "2025-07-14",
                  "severity": "Unscored",
                  "summary": "A remote code execution vulnerability exists in the Windows
Remote Desktop Client when a user connects to a malicious server. An attacker who
successfully exploited this vulnerability could execute arbitrary code on the computer of the
connecting client. An attacker could then install programs; view, change, or delete data; or
create new accounts with full user rights. NOTE: This vulnerability is currently awaiting
analysis."
                },
                {
                  "absoluteFirstFoundDate": "2025-07-14",
                  "cveId": "CVE-2020-0818",
                  "cveYear": "2020",
                  "cvssScore": null,
                  "detectedProducts": [
                    "Operating System"
                  ],
                  "detectedCPEs": [
                    "cpe:/o:microsoft:windows_10:1507::x64"
                  ],
                  "firstFound": "2025-07-14",
                  "lastFound": "2025-07-14",
                  "severity": "Unscored",
                  "summary": "An elevation of privilege vulnerability exists in the way that
the sysmain.dll handles objects in memory. An attacker who successfully exploited the
vulnerability could execute code with elevated permissions. NOTE: This vulnerability is
currently awaiting analysis."
                }
              ]
            }
          }
```


©2026 Tanium Inc. All Rights Reserved

```
        }
      ],
      "pageInfo": {
        "startCursor": "MTQ2NTkxODE5OjA=",
        "endCursor": "MTQ2NTkxODE5OjA=",
        "hasPreviousPage": false,
        "hasNextPage": true
      }
    }
  }
}
```

GET FILTERED ENDPOINT VULNERABILITY FINDINGS INFORMATION FROM FILTERED SET OF ENDPOINTS

(query.endpoints.edges.node.compliance.cveFindings)

The following query requires Comply 2.17.167 or later and retrieves the first endpoint that is a member of the All Windows computer group, with a vulnerability findings `cveYear` of `2020` and `cvssScore` value of `5`, and returns that endpoint's associated vulnerability findings with a `cveYear` of `2020` and `cvssScore` value of `5`.

```
query exampleFilterWindowsEndpointVulnerability($first: Int!) {
  endpoints(first: $first, filter: {memberOf: {name: "All Windows"}}) {
    edges {
      node {
        name
        ipAddress
        compliance {
          cveFindings(
            filter: {any: false, filters: [{path: "cveYear", value: "2020"}, {path:
"cvssScore", value: "5"}]}
          ) {
            absoluteFirstFoundDate
            cveId
            cveYear
            cvssScoreV3
            detectedProducts
            detectedCPEs
            firstFound
            lastFound
            severityV3
            summary
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the cursor variable in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "first": 1
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "edges": [
        {
          "node": {
            "name": "example-host-3",
            "ipAddress": "10.30.0.100",
            "compliance": {
              "cveFindings": [
                {
                  "absoluteFirstFoundDate": "2024-02-08",
                  "cveId": "CVE-2020-0645",
                  "cveYear": "2020",
                  "cvssScoreV3": 7.5,
                  "detectedProducts": [
                    "Operating System"
                  ],
                  "detectedCPEs": [
                    "cpe:/o:microsoft:windows_server_2016:::x64"
                  ],
                  "firstFound": "2024-02-08",
                  "lastFound": "2025-07-10",
                  "severityV3": "High",
                  "summary": "A tampering vulnerability exists when Microsoft IIS Server
improperly handles malformed request headers, aka 'Microsoft IIS Server Tampering
Vulnerability'."
                },
                {
                  "absoluteFirstFoundDate": "2024-02-08",
                  "cveId": "CVE-2020-0660",
                  "cveYear": "2020",
                  "cvssScoreV3": 7.5,
                  "detectedProducts": [
                    "Operating System"
                  ],
                  "detectedCPEs": [
                    "cpe:/o:microsoft:windows_server_2016:::x64"
                  ],
                  "firstFound": "2024-02-08",
                  "lastFound": "2025-07-10",
                  "severityV3": "High",
                  "summary": "A denial of service vulnerability exists in Remote Desktop
Protocol (RDP) when an attacker connects to the target system using RDP and sends specially
crafted requests, aka 'Windows Remote Desktop Protocol (RDP) Denial of Service
Vulnerability'."
                }
              ]
            }
          }
        }
      ],
      "pageInfo": {
        "startCursor": "MTQ2NTkyOTI5OjA=",
        "endCursor": "MTQ2NTkyOTI5OjA=",
        "hasPreviousPage": false,
```

©2026 Tanium Inc. All Rights Reserved

```
        "hasNextPage": true
      }
    }
  }
}
```

Computer group examples

The following queries and mutations create, retrieve, and delete computer groups.

Create a type of computer group

(mutation.computerGroupCreate)

You can create the following computer group types:

- computer management group (computer group) - A Tanium user can view question results from, and deploy actions to, only those endpoints that belong to a computer group that is assigned to the user persona selected for the current session. You can view computer groups in Tanium Console at **Administration** > **Permissions** > **Computer Groups**. To create a computer group, in your mutation request, configure `managementRightsEnabled` as `true`. For more information, see Create computer group on page 330.
- computer filter group (filter group) - Tanium users use filter groups as filters in questions and question results. You can view filter groups in Tanium Console at **Administration** > **Permissions** > **Filter Groups**. To create a filter group, in your mutation request, configure a content set reference at `contentSetRef.name`. For more information, see Create filter group on page 331.

You can configure a computer group to also function as a filter group. In your mutation request, configure `managementRightsEnabled` as `true` and a content set reference at `contentSetRef.name`. For more information, see Create computer group that also filters on page 332.

For more information about computer groups, see Tanium Console User Guide: Computer groups overview.

CREATE COMPUTER GROUP

The following mutation creates a computer group, configuring `managementRightsEnabled` as `true`.

```
mutation exampleCreateComputerGroup($name: String!, $groupName: String, $mrEnabled: Boolean!)
```

©2026 Tanium Inc. All Rights Reserved

```
{
  computerGroupCreate(
    input: {name: $name, filter: {memberOf: {name: $groupName}}, managementRightsEnabled:
$mrEnabled}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the name, filter condition, and management rights flag variables in the **Variables** panel or in your variables dictionary:

```
{
  "name": "Example computer group",
  "groupName": "No Computers",
  "mrEnabled": true
}
```

Example response:

```
{
  "data": {
    "computerGroupCreate": {
      "group": {
        "id": "12345"
      },
      "error": null
    }
  }
}
```

CREATE FILTER GROUP

The following mutation creates a filter group by defining a content set (`Default`) to which the group belongs.

```
mutation exampleCreateComputerGroup2($name: String!, $contentSetName: String, $groupName:
String, $mrEnabled: Boolean!) {
  computerGroupCreate(
    input: {name: $name, contentSetRef: {name: $contentSetName}, filter: {memberOf: {name:
$groupName}}, managementRightsEnabled: $mrEnabled}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the name, content set, filter condition, and management rights flag variables in the **Variables** panel or in your variables dictionary:

```
{
  "name": "Example computer group",
  "contentSetName": "Default",
  "groupName": "No Computers",
  "mrEnabled": false
}
```

Example response:

```
{
  "data": {
    "computerGroupCreate": {
      "group": {
        "id": "12345"
      },
      "error": null
    }
  }
}
```

CREATE COMPUTER GROUP THAT ALSO FILTERS

The following mutation creates a computer group that can also be used as a filter group. This request configures `managementRightsEnabled` as `true` and references the `Default` content set.

```
mutation exampleCreateComputerGroupFilter($name: String!, $contentSetName: String,
$groupName: String, $mrEnabled: Boolean!) {
  computerGroupCreate(
    input: {name: $name, contentSetRef: {name: $contentSetName}, filter: {memberOf: {name:
$groupName}}, managementRightsEnabled: $mrEnabled}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the name, filter condition, and management rights flag variables in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "name": "Example computer group that filters",
  "contentSetName": "Default",
  "groupName": "No Computers",
  "mrEnabled": true
}
```

Example response:

```
{
  "data": {
    "computerGroupCreate": {
      "group": {
        "id": "12345"
      },
      "error": null
    }
  }
}
```

CREATE COMPUTER GROUP BASED ON EXISTING COMPUTER GROUPS

The following mutation creates a computer group, defining membership based on existing computer groups. Endpoints in `Windows or Mac example group` belong to either the `All Windows` or the `All Mac` computer group.

> **Note:** If you set `any` in the filter to `false`, and the computer groups you define have mutually exclusive criteria, no endpoints are members of your created computer group because no endpoints can meet all of the mutually exclusive criteria.

```
mutation exampleCreateComputerGroup3($cgName: String!, $mrEnabled: Boolean!, $filterAny:
Boolean!, $groupName1: String, $groupName2: String) {
  computerGroupCreate(
    input: {name: $cgName, managementRightsEnabled: $mrEnabled, filter: {any: $filterAny,
filters: [{memberOf: {name: $groupName1}}, {memberOf: {name: $groupName2}}]}}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "cgName": "Windows or Mac example group",
  "mrEnabled": true,
  "filterAny": true,
  "groupName1": "All Windows",
```

©2026 Tanium Inc. All Rights Reserved

```
  "groupName2": "All Mac"
}
```

Example response:

```
{
  "data": {
    "computerGroupCreate": {
      "group": {
        "id": "12345"
      },
      "error": null
    }
  }
}
```

CREATE COMPUTER GROUP BASED ON SENSOR

The following mutation creates a computer group, defining membership based on a sensor. `All Mac example group` contains endpoints that match when the `Is Mac` sensor returns `true` for that endpoint.

```
mutation exampleCreateComputerGroup4($cgName: String!, $mrEnabled: Boolean!, $sensorName:
String, $sensorOp: FieldFilterOp!, $sensorValue: String) {
  computerGroupCreate(
    input: {name: $cgName, managementRightsEnabled: $mrEnabled, filter: {sensor: {name:
$sensorName}, op: $sensorOp, value: $sensorValue}}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "cgName": "All Mac example group",
  "mrEnabled": true,
  "sensorName": "Is Mac",
  "sensorOp": "EQ",
  "sensorValue": "true"
}
```

Example response:

```
{
  "data": {
    "computerGroupCreate": {
      "group": {
        "id": "12345"
      },
```

©2026 Tanium Inc. All Rights Reserved

```
      "error": null
    }
  }
}
```

CREATE COMPUTER GROUP USING MULTIPLE CRITERIA

The following mutation creates a computer group using multiple filters to define membership based on existing computer groups and sensors. Based on the filter logic, the following endpoints are members of `Example multiple filters computer group`:

- All virtual endpoints running Linux
- All virtual endpoints running macOS
- All virtual endpoints running Windows with a high memory consumption

```
mutation exampleCreateComputerGroup5($cgName: String!, $mrEnabled: Boolean!, $filterAny1:
Boolean!, $sensorName1: String, $sensorColumn1: String, $sensorOp1: FieldFilterOp!,
$sensorValue1: String, $filterAny2: Boolean!, $cGroup1: String, $cGroup2: String,
$filterAny3: Boolean!, $cGroup3: String, $sensorName2: String, $sensorColumn2: String,
$sensorOp2: FieldFilterOp!, $sensorValue2: String, $sensorNegated2: Boolean!) {
  computerGroupCreate(
    input: {name: $cgName, managementRightsEnabled: $mrEnabled, filter: {any: $filterAny1,
filters: [{sensor: {name: $sensorName1, column: $sensorColumn1}, op: $sensorOp1, value:
$sensorValue1}, {any: $filterAny2, filters: [{memberOf: {name: $cGroup1}}, {memberOf: {name:
$cGroup2}}, {any: $filterAny3, filters: [{memberOf: {name: $cGroup3}}, {sensor: {name:
$sensorName2, column: $sensorColumn2}, op: $sensorOp2, negated: $sensorNegated2, value:
$sensorValue2}]}]}]}}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "cgName": "Example multiple filters computer group",
  "mrEnabled": true,
  "filterAny1": false,
  "sensorName1": "Is Virtual",
  "sensorColumn1": "Is Virtual",
  "sensorOp1": "EQ",
  "sensorValue1": "true",
  "filterAny2": true,
  "cGroup1": "All Linux",
  "cGroup2": "All Mac",
  "filterAny3": false,
  "cGroup3": "All Windows",
  "sensorName2": "High Memory Consumption",
```

©2026 Tanium Inc. All Rights Reserved

```
  "sensorColumn2": "High Memory Consumption",
  "sensorOp2": "EQ",
  "sensorValue2": "Under Threshold",
  "sensorNegated2": true
}
```

Example response:

```
{
  "data": {
    "computerGroupCreate": {
      "group": {
        "id": "12345"
      },
      "error": null
    }
  }
}
```

Delete computer group by ID or name

(mutation.computerGroupDelete)

DELETE COMPUTER GROUP BY ID

The following mutation deletes the computer group that matches the ID.

```
mutation exampleDeleteComputerGroup($id: ID) {
  computerGroupDelete(ref: {id: $id}) {
    error {
      message
    }
    id
  }
}
```

Include the computer group ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345"
}
```

Example response:

```
{
  "data": {
    "computerGroupDelete": {
      "error": null,
      "id": "12345"
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

DELETE COMPUTER GROUP BY NAME

You can also delete a computer group that matches a name.

```
mutation exampleDeleteComputerGroup2($name: String) {
  computerGroupDelete(ref: {name: $name}) {
    error {
      message
    }
    id
  }
}
```

Include the computer group name variable in the **Variables** panel or in your variables dictionary:

```
{
  "name": "Example computer group"
}
```

Get a computer group by ID or name

(query.computerGroup)

GET COMPUTER GROUP BY ID

The following query retrieves a computer group that matches an ID.



Note: The `modifiedTime` field requires Gateway 1.17 or later.

```
query exampleGetComputerGroupByID($id: ID) {
  computerGroup(ref: {id: $id}) {
    id
    name
    contentSet {
      id
      name
    }
    expression
    modifiedTime
    type
    filterEnabled
    managementRightsEnabled
  }
}
```

Include the computer group ID variable in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "id": "12345"
}
```

Example response:

```
{
  "data": {
    "computerGroup": {
      "id": "12345",
      "name": "Example computer group",
      "contentSet": {
        "id": "5",
        "name": "Client Management"
      },
      "expression": "No Computers",
      "modifiedTime": "2023-12-31T23:59:59Z",
      "type": "STANDARD",
      "filterEnabled": true,
      "managementRightsEnabled": false
    }
  }
}
```

GET COMPUTER GROUP BY NAME

You can also retrieve a computer group that matches a name.



Note: The `modifiedTime` field requires Gateway 1.17 or later.

```
query exampleGetComputerGroupByName($name: String!) {
  computerGroup(ref: {name: $name}) {
    id
    name
    contentSet {
      id
      name
    }
    expression
    modifiedTime
    type
    filterEnabled
    managementRightsEnabled
  }
}
```

Include the computer group name variable in the **Variables** panel or in your variables dictionary:

```
{
  "name": "Example computer group"
}
```

©2026 Tanium Inc. All Rights Reserved

Get multiple computer groups, filter groups, or management rights groups

(query.computerGroups)

GET COMPUTER GROUPS

The following query retrieves the first `2` computer groups. For more information about pagination, see Pagination on page 22.

> **Note:** The `modifiedTime` field requires Gateway 1.17 or later.

```
query exampleGetTwoComputerGroups($first: Int) {
  computerGroups(first: $first) {
    edges {
      node {
        id
        name
        contentSet {
          id
          name
        }
        expression
        modifiedTime
        type
        filterEnabled
        managementRightsEnabled
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "computerGroups": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "Example computer group",
            "contentSet": {
              "id": "5",
              "name": "Client Management"
            },
```

©2026 Tanium Inc. All Rights Reserved

```
            "expression": "No Computers",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "STANDARD",
            "filterEnabled": true,
            "managementRightsEnabled": false
          }
        },
        {
          "node": {
            "id": "23456",
            "name": "Example computer group 2",
            "contentSet": {
              "id": "2",
              "name": "Default"
            },
            "expression": "Computer Name contains example",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "STANDARD",
            "filterEnabled": true,
            "managementRightsEnabled": false
          }
        }
      ]
    }
  }
}
```

GET MANUAL COMPUTER GROUPS

The following query retrieves the first `2` manual computer groups, whose membership is based on a manually entered list of computer names or fully qualified domain names (FQDNs).



Note: The `modifiedTime` field requires Gateway 1.17 or later.

```
query exampleGetManualGroups($first: Int, $type: ComputerGroupType) {
  computerGroups(first: $first, filter: {type: $type}) {
    edges {
      node {
        id
        name
        contentSet {
          id
          name
        }
        expression
        modifiedTime
        type
        filterEnabled
        managementRightsEnabled
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the pagination and type variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "type": "MANUAL"
}
```

Example response:

```
{
  "data": {
    "computerGroups": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "Manual Group 1",
            "contentSet": {
              "id": "23456",
              "name": ""
            },
            "expression": " Manual Group Membership equals 12345",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "MANUAL",
            "filterEnabled": true,
            "managementRightsEnabled": true
          }
        },
        {
          "node": {
            "id": "34567",
            "name": "Manual Group 2",
            "contentSet": null,
            "expression": " Manual Group Membership equals 34567",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "MANUAL",
            "filterEnabled": false,
            "managementRightsEnabled": true
          }
        }
      ]
    }
  }
}
```

(1.2 - EXPERIMENTAL (RELEASE CANDIDATE)) GET COMPUTER GROUPS BY NAME

The following query retrieves all computer groups whose name starts with `test`.

```
query exampleGetComputerGroupsWhoseNameStartsWithTest {
  computerGroups(filter:{name:"test", nameOp:STARTS_WITH},) {
    edges {
      node {
        id,
        name,
        type
      }
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Example response:

```
{
  "data": {
    "computerGroups": {
      "edges": [
        {
          "node": {
            "id": "1131",
            "name": "test ras targeting",
            "type": "STANDARD"
          }
        },
        {
          "node": {
            "id": "1132",
            "name": "test rus targeting",
            "type": "STANDARD"
          }
        }
      ]
    }
  }
}
```

GET STANDARD COMPUTER GROUPS

The following query retrieves the first `2` standard computer groups, whose membership is based on the results of a sensor filter expression.



Note: The `modifiedTime` field requires Gateway 1.17 or later.

```
query exampleGetStandardGroups($first: Int, $type: ComputerGroupType) {
  computerGroups(first: $first, filter: {type: $type}) {
    edges {
      node {
        id
        name
        contentSet {
          id
          name
        }
        expression
        modifiedTime
        type
        filterEnabled
        managementRightsEnabled
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the pagination and type variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "type": "STANDARD"
}
```

Example response:

```
{
  "data": {
    "computerGroups": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "Standard group test",
            "contentSet": {
              "id": "2",
              "name": "Default"
            },
            "expression": " Computer Name contains test ",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "STANDARD",
            "filterEnabled": true,
            "managementRightsEnabled": true
          }
        },
        {
          "node": {
            "id": "23456",
            "name": "Domain test",
            "contentSet": {
              "id": "3",
              "name": "Default Filter Groups"
            },
            "expression": " ( Domain Name contains test.example )",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "STANDARD",
            "filterEnabled": true,
            "managementRightsEnabled": true
          }
        }
      ]
    }
  }
}
```

GET FILTER GROUPS

The following query retrieves the first `2` filter groups.



Note: The `modifiedTime` field requires Gateway 1.17 or later.

©2026 Tanium Inc. All Rights Reserved

```
query exampleGetFilterGroups($first: Int, $filterEnabled: Boolean) {
  computerGroups(first: $first, filter: {filterEnabled: $filterEnabled}) {
    edges {
      node {
        id
        name
        contentSet {
          id
          name
        }
        expression
        modifiedTime
        type
        filterEnabled
        managementRightsEnabled
      }
    }
  }
}
```

Include the pagination and filter enabled variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "filterEnabled": true
}
```

Example response:

```
{
  "data": {
    "computerGroups": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "All Windows Workstations - x64",
            "contentSet": {
              "id": "3",
              "name": "Default Filter Groups"
            },
            "expression": " ( Windows OS Type contains windows workstation and \"x64/x86?\"
contains x64 )",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "STANDARD",
            "filterEnabled": true,
            "managementRightsEnabled": true
          }
        },
        {
          "node": {
            "id": "23456",
            "name": "All Windows Server 2008 R2",
            "contentSet": {
              "id": "3",
              "name": "Default Filter Groups"
            },
            "expression": " Operating System Generation equals Windows Server 2008 R2",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "STANDARD",
```

©2026 Tanium Inc. All Rights Reserved

```
            "filterEnabled": true,
            "managementRightsEnabled": true
          }
        }
      ]
    }
  }
}
```

GET MANAGEMENT RIGHTS GROUPS

The following query retrieves the first `2` management rights groups.



Note: The `modifiedTime` field requires Gateway 1.17 or later.

```
query exampleGetMRGroups($first: Int, $mrEnabled: Boolean) {
  computerGroups(first: $first, filter: {managementRightsEnabled: $mrEnabled}) {
    edges {
      node {
        id
        name
        contentSet {
          id
          name
        }
        expression
        modifiedTime
        type
        filterEnabled
        managementRightsEnabled
      }
    }
  }
}
```

Include the pagination and management rights enabled variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "mrEnabled": true
}
```

Example response:

```
{
  "data": {
    "computerGroups": {
      "edges": [
        {
          "node": {
            "id": "12345",
            "name": "Test Filter",
```

©2026 Tanium Inc. All Rights Reserved

```
            "contentSet": {
              "id": "3",
              "name": "Default Filter Groups"
            },
            "expression": " Computer Name starts with test",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "STANDARD",
            "filterEnabled": true,
            "managementRightsEnabled": true
          }
        },
        {
          "node": {
            "id": "23456",
            "name": "All Windows Workstations",
            "contentSet": {
              "id": "3",
              "name": "Default Filter Groups"
            },
            "expression": " ( Windows OS Type contains windows workstation )",
            "modifiedTime": "2023-12-31T23:59:59Z",
            "type": "STANDARD",
            "filterEnabled": true,
            "managementRightsEnabled": true
          }
        }
      ]
    }
  }
}
```

(1.1 - EXPERIMENTAL (ACTIVE DEVELOPMENT)) GET MANAGEMENT RIGHTS GROUPS CURRENT USER AND PERSONA CAN ACCESS

The following query retrieves the list of management rights groups assigned to the requesting user and active persona. The assigned management rights groups determine the endpoints that the user can access.

Note: If the response returns an empty list, the requesting user and active persona has unrestricted access.

Note: The `managementRightsGroups` field requires Gateway 1.19 or later.

```
query exampleGetUserPersonaManagementRights {
  managementRightsGroups {
    id
    name
    contentSetV2 {
      id
```

```
      name
      description
    }
    expression
    filterEnabled
    managementRightsEnabled
    modifiedTime
    type
  }
}
```

Example response if user and active persona have unrestricted management rights:

```
{
  "data": {
    "managementRightsGroups": []
  }
}
```

Example response if user and active persona have restricted management rights:

```
{
  "data": {
    "managementRightsGroups": [
      {
        "id": "123",
        "name": "All Linux",
        "contentSetV2": {
          "id": "3",
          "name": "Default Filter Groups",
          "description": null
        },
        "expression": " Is Linux equals True",
        "filterEnabled": true,
        "managementRightsEnabled": true,
        "modifiedTime": null,
        "type": "STANDARD"
      },
      {
        "id": "234",
        "name": "All Mac",
        "contentSetV2": {
          "id": "3",
          "name": "Default Filter Groups",
          "description": null
        },
        "expression": " Is Mac equals True",
        "filterEnabled": true,
        "managementRightsEnabled": true,
        "modifiedTime": null,
        "type": "STANDARD"
      }
    ]
  }
}
```

Connect examples

©2026 Tanium Inc. All Rights Reserved

The following queries and mutations require Connect, retrieve information about connections and connection runs, and allow you to start or stop connection runs.

Connect v1 REST API replacement

About this task

The Connect GraphQL APIs are at stability index `1.2 - Experimental (Release Candidate)`. For the best results, if a GraphQL API provides functionality, use that instead of the corresponding legacy Connect REST API. Connect REST APIs not listed in the table do not have comparable functionality in Tanium Gateway, or are otherwise not used in favor of other Connect REST APIs.

Gateway GraphQL API replacements for Connect v1 REST API endpoints

Connect v1 REST API endpoint	Gateway GraphQL API replacement	Notes
`DELETE v1/connections/:id`	n/a	Delete a connection by ID
`GET v1/connectionruns` (legacy)	`query.connectRuns`	Get connection runs
`GET v1/connections` (legacy)	`query.connectConnections`	Get connections
`GET v1/connections/:id` (legacy)	`query.connectConnection`	Get a connection by ID
`GET v1/connections/:id/runs/:connectionrunid` (legacy)	`query.connectRun`	Get a connection run by ID
`GET v1/connections/:id/runs/:connectionrunid/download`	n/a	Get the log data for a connection run
`POST v1/connections/:id/runs` (legacy)	`mutation.connectConnectionStart`	Start a connection run
`POST v1/connections/:id/stop` (legacy)	`mutation.connectConnectionStop`	Stop an in-progress connection run
`POST v1/import`	n/a	Create or modify a connection

©2026 Tanium Inc. All Rights Reserved

Get Connect connection by ID

(query.connectConnection)

About this task

The following query requires Connect 5.20 or later and retrieves a connection that matches the ID.

```
query exampleGetSingleConnection($id: ID!) {
  connectConnection(ref: {id: $id}) {
    id
    currentRun {
      id
      connectionID
      status
      createdTime
      updatedTime
    }
    description
    displayName
    isScheduled
    lastRunStatus
    lastRunTime
    pastRuns {
      edges {
        node {
          id
          connectionID
          status
          createdTime
          updatedTime
        }
      }
    }
  }
}
```

Include an ID variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "id": "123"
}
```

Example response:

```
{
  "data": {
    "connectConnection": {
      "id": "123",
      "currentRun": null,
      "description": "Example connection description",
      "displayName": "Example connection name",
      "isScheduled": true,
      "lastRunStatus": "Finished",
      "lastRunTime": "2026-05-08T17:00:15.737Z",
      "pastRuns": {
        "edges": [
```

©2026 Tanium Inc. All Rights Reserved

```
          {
            "node": {
              "id": "234",
              "connectionID": "23",
              "status": "Finished",
              "createdTime": "2026-05-08T17:00:15.737Z",
              "updatedTime": "2026-05-08T17:00:21.276Z"
            }
          },
          {
            "node": {
              "id": "345",
              "connectionID": "23",
              "status": "Finished",
              "createdTime": "2026-05-08T15:45:00.575Z",
              "updatedTime": "2026-05-08T15:45:03.567Z"
            }
          }
        ]
      }
    }
  }
}
```

Get Connect connection run by ID

(query.connectRun)

About this task

The following query requires Connect 5.20 or later and retrieves a connection run that matches the ID.

```
query exampleGetSingleConnectionRun($id: ID!) {
  connectRun(ref: {id: $id}) {
    id
    connectionID
    status
    createdTime
    updatedTime
  }
}
```

Include an ID variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "id": "123"
}
```

Example response:

```
{
  "data": {
    "connectRun": {
      "id": "123",
      "connectionID": "234",
```

```
      "status": "Finished",
      "createdTime": "2023-08-15T07:01:21.824Z",
      "updatedTime": "2023-08-15T07:01:30.912Z"
    }
  }
}
```

Get multiple Connect connection runs

(query.connectRuns)

About this task

The following query requires Connect 5.20 or later and retrieves the first 2 connection runs.

```
query exampleGetConnectionRuns($first: Int) {
  connectRuns(first: $first) {
    edges {
      node {
        id
        connectionID
        status
        createdTime
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include a pagination variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "connectRuns": {
      "edges": [
        {
          "node": {
            "id": "1",
            "connectionID": "123",
            "status": "Finished",
            "createdTime": "2023-08-15T07:01:21.824Z",
            "updatedTime": "2023-08-15T07:01:30.912Z"
          }
        },
```

©2026 Tanium Inc. All Rights Reserved

```
        {
          "node": {
            "id": "2",
            "connectionID": "234",
            "status": "Finished",
            "createdTime": "2023-08-15T07:01:19.678Z",
            "updatedTime": "2023-08-15T07:01:23.757Z"
          }
        }
      ],
      "pageInfo": {
        "hasPreviousPage": false,
        "hasNextPage": true,
        "startCursor": "MA==",
        "endCursor": "MQ=="
      },
      "totalRecords": 504
    }
  }
}
```

Get multiple Connect connections

(query.connectConnections)

About this task

The following query requires Connect 5.20 or later and retrieves the first 2 connections.

```
query exampleGetConnections($first: Int) {
  connectConnections(first: $first) {
    edges {
      node {
        id
        currentRun {
          id
          connectionID
          status
          createdTime
          updatedTime
        }
        description
        displayName
        isScheduled
        lastRunStatus
        lastRunTime
        pastRuns {
          edges {
            node {
              id
              connectionID
              status
              createdTime
              updatedTime
            }
          }
        }
      }
    }
```

©2026 Tanium Inc. All Rights Reserved

```
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include a pagination variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "connectConnections": {
      "edges": [
        {
          "node": {
            "id": "23",
            "currentRun": null,
            "description": "Example connection description",
            "displayName": "Example connection name",
            "isScheduled": true,
            "lastRunStatus": "Finished",
            "lastRunTime": "2026-05-08T15:36:00.010Z",
            "pastRuns": {
              "edges": [
                {
                  "node": {
                    "id": "234",
                    "connectionID": "23",
                    "status": "Finished",
                    "createdTime": "2026-05-08T15:36:00.010Z",
                    "updatedTime": "2026-05-08T15:46:09.098Z"
                  }
                },
                {
                  "node": {
                    "id": "345",
                    "connectionID": "23",
                    "status": "Finished",
                    "createdTime": "2026-05-08T12:36:00.012Z",
                    "updatedTime": "2026-05-08T12:46:06.430Z"
                  }
                }
              ]
            }
          }
        },
        {
          "node": {
            "id": "13",
            "currentRun": null,
            "description": "Example connection description 2",
            "displayName": "Example connection name 2",
```

©2026 Tanium Inc. All Rights Reserved

```
            "isScheduled": false,
            "lastRunStatus": "Finished",
            "lastRunTime": "2026-02-12T21:27:46.686Z",
            "pastRuns": {
              "edges": []
            }
          }
        }
      ],
      "pageInfo": {
        "hasNextPage": true,
        "hasPreviousPage": false,
        "startCursor": "MA==",
        "endCursor": "MQ=="
      },
      "totalRecords": 23
    }
  }
}
```

Get multiple Connect connections by ID

(query.connectConnections)

About this task

The following query requires Connect 5.45 or later and retrieves the connections that match either ID `1234` or `5678`.

```
query exampleGetMultipleConnectionsByIDs($first_id: ID!, $second_id: ID!) {
  connectConnections(filter: {ids: [$first_id, $second_id]}) {
    totalRecords
    edges {
      node {
        id
        displayName
        isScheduled
        lastRunTime
        lastRunStatus
      }
    }
  }
}
```

Include the ID variables in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "first_id": "1234",
  "second_id": "5678"
}
```

Example response:

```
{
  "data": {
    "connectConnections": {
```

```
      "totalRecords": 2,
      "edges": [
        {
          "node": {
            "id": "1234",
            "displayName": "Example connection 1",
            "isScheduled": false,
            "lastRunTime": "2026-02-12T21:27:46.686Z",
            "lastRunStatus": "Finished"
          }
        },
        {
          "node": {
            "id": "5678",
            "displayName": "Example connection 2",
            "isScheduled": false,
            "lastRunTime": null,
            "lastRunStatus": null
          }
        }
      ]
    }
  }
}
```

Get multiple Connect connections by description

(query.connectConnections)

About this task

The following query requires Connect 5.45 or later and retrieves connections that contain `Example` in the description.

```
query exampleGetConnectionsByDescription($target_description: String) {
  connectConnections(filter: {description: $target_description}) {
    totalRecords
    edges {
      node {
        id
        displayName
        description
        isScheduled
        lastRunStatus
        lastRunTime
      }
    }
  }
}
```

Include the description variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "target_description": "Example"
}
```

Example response:


```
{
  "data": {
    "connectConnections": {
      "totalRecords": 3,
      "edges": [
        {
          "node": {
            "id": "1234",
            "displayName": "Example connection 1",
            "description": "An example description",
            "isScheduled": true,
            "lastRunStatus": "Finished",
            "lastRunTime": "2026-05-08T15:36:00.010Z"
          }
        },
        {
          "node": {
            "id": "567",
            "displayName": "Example connection 2",
            "description": "Example description",
            "isScheduled": false,
            "lastRunStatus": null,
            "lastRunTime": null
          }
        },
        {
          "node": {
            "id": "8901",
            "displayName": "Example connection 3",
            "description": "A description that contains example",
            "isScheduled": false,
            "lastRunStatus": null,
            "lastRunTime": null
          }
        }
      ]
    }
  }
}
```


Get Connect connection by ID and displayName

(query.connectConnections)

About this task

The following query requires Connect 5.45 or later and retrieves the connection that matches ID `1234` and contains `Example Connection` in the displayName field.

```
query exampleGetConnectionByIDAndDisplayName($target_id: ID!, $target_display_name: String) {
  connectConnections(filter: { ids: [$target_id], displayName: $target_display_name }) {
    totalRecords
    edges {
      node {
        id
```

©2026 Tanium Inc. All Rights Reserved

```
        displayName
      }
    }
  }
}
```

Include the ID and display name variables in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "target_id": "1234",
  "target_display_name": "Example Connection"
}
```

Example response:

```
{
  "data": {
    "connectConnections": {
      "totalRecords": 1,
      "edges": [
        {
          "node": {
            "id": "1234",
            "displayName": "Example Connection"
          }
        }
      ]
    }
  }
}
```

Get SQL Server stored procedures

(query.connectSQLServerStoredProcedures)

About this task

The following query requires Connect 5.52 or later and retrieves the stored procedures defined in one or more schemas within a SQL Server database. For each procedure, the query returns the procedure name, whether the configured user has `EXECUTE` permission, and the declared parameters with their SQL Server data types.

```
query exampleGetSQLServerStoredProcedures(
  $server: String!
  $username: String!
  $password: String!
  $database: String!
  $schemaNames: [String!]!
) {
  connectSQLServerStoredProcedures(
    input: {
      connectionConfiguration: {
        server: $server
```

©2026 Tanium Inc. All Rights Reserved

```
        username: $username
        password: $password
        encrypt: false
        validateServerCertificate: false
        connectTimeoutMillis: 5000
        requestTimeoutMillis: 30000
        cancelTimeoutMillis: 5000
        packetSizeBytes: 4096
        multiSubnetFailover: false
      }
      database: $database
      schemaNames: $schemaNames
    }
  ) {
    schemaName
    procedures {
      name
      hasExecute
      parameters {
        name
        dataType
      }
    }
  }
}
```

Include variables in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "server": "sql.example.com",
  "username": "reportuser",
  "password": "examplepassword",
  "database": "Sales",
  "schemaNames": [
    "dbo",
    "reporting"
  ]
}
```

Example response:

```
{
  "data": {
    "connectSQLServerStoredProcedures": [
      {
        "schemaName": "dbo",
        "procedures": [
          {
            "name": "sp_onConnectionStart",
            "hasExecute": true,
            "parameters": [
              {
                "name": "@connectionName",
                "dataType": "nvarchar"
              },
              {
                "name": "@connectionId",
                "dataType": "bigint"
              },
              {
                "name": "@connectionRunId",
```

©2026 Tanium Inc. All Rights Reserved

```
                "dataType": "bigint"
              },
              {
                "name": "@source",
                "dataType": "nvarchar"
              },
              {
                "name": "@databaseTable",
                "dataType": "nvarchar"
              },
              {
                "name": "@startTime",
                "dataType": "datetime"
              },
              {
                "name": "@expectedRowCount",
                "dataType": "bigint"
              }
            ]
          },
          {
            "name": "sp_onConnectionFinish",
            "hasExecute": true,
            "parameters": [
              {
                "name": "@connectionRunId",
                "dataType": "bigint"
              },
              {
                "name": "@runStatus",
                "dataType": "nvarchar"
              },
              {
                "name": "@sourceRowCount",
                "dataType": "bigint"
              },
              {
                "name": "@destinationRowCount",
                "dataType": "bigint"
              },
              {
                "name": "@finishTime",
                "dataType": "datetime"
              }
            ]
          }
        ]
      },
      {
        "schemaName": "reporting",
        "procedures": []
      }
    ]
  }
}
```

Get SQL Server stored procedures through a satellite

(query.connectSQLServerStoredProcedures)

©2026 Tanium Inc. All Rights Reserved

About this task

The following query requires Connect 5.52 or later and retrieves the stored procedures defined in one or more schemas within a SQL Server database through a satellite. For each procedure, the query returns the procedure name, whether the configured user has `EXECUTE` permission, and the declared parameters with their SQL Server data types.

```
query exampleGetSQLServerStoredProceduresThroughSatellite(
  $server: String!,
  $username: String!,
  $password: String!,
  $database: String!,
  $satelliteID: ID!,
  $schemaNames: [String!]!
) {
  connectSQLServerStoredProcedures(
    input: {
      connectionConfiguration: {
        server: $server,
        username: $username,
        password: $password,
        satelliteID: $satelliteID,
        encrypt: false,
        validateServerCertificate: false,
        connectTimeoutMillis: 5000,
        requestTimeoutMillis: 30000,
        cancelTimeoutMillis: 5000,
        packetSizeBytes: 4096,
        multiSubnetFailover: false
      },
      database: $database,
      schemaNames: $schemaNames
    }
  ) {
    schemaName
    procedures {
      name
      hasExecute
      parameters {
        name
        dataType
      }
    }
  }
}
```

Include variables in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "server": "sql.example.com",
  "username": "reportuser",
  "password": "examplepassword",
  "database": "Sales",
  "satelliteID": "e66081b7-5900-46bb-97eb-fcbf16d2c24f",
  "schemaNames": [
    "dbo"
  ]
}
```

©2026 Tanium Inc. All Rights Reserved

Start a Connect connection run

(mutation.connectConnectionStart)

About this task

The following mutation requires Connect 5.20 or later and starts a connection run from a connection that matches the connection ID.

```
mutation exampleStartConnection($connectionID: ID!) {
  connectConnectionStart(input: {connectionID: $connectionID}) {
    run {
      id
      connectionID
      status
      createdTime
      updatedTime
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include an ID variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionID": "123"
}
```

Example response:

```
{
  "data": {
    "connectConnectionStart": {
      "run": {
        "id": "12345",
        "connectionID": "123",
        "status": "Starting",
        "createdTime": "2023-08-15T18:16:54.509Z",
        "updatedTime": "2023-08-15T18:16:54.512Z"
      },
      "error": null
    }
  }
}
```

Stop the active run for a Connect connection

(mutation.connectConnectionStop)

©2026 Tanium Inc. All Rights Reserved

About this task

The following mutation stops the active run, if any, for the specified Connect connection. If the connection has no in-progress run, this mutation succeeds with `runID` set to `null`. No run is started.

```
mutation exampleStopConnection($connectionID: ID!) {
  connectConnectionStop(input: {connectionID: $connectionID}) {
    runID
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include an ID variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionID": "234"
}
```

Example response:

```
{
  "data": {
    "connectConnectionStop": {
      "runID": "123",
      "error": null
    }
  }
}
```

Stop a specific Connect connection run

(mutation.connectConnectionRunStop)

About this task

The following mutation stops a connection run that is currently in progress. The connection run is identified by ID. This request returns an error if the connection run is not in progress.

```
mutation exampleStopConnectionRun($runID: ID!) {
  connectConnectionRunStop(input: {runID: $runID}) {
    run {
      id
      connectionID
      status
      createdTime
      updatedTime
    }
    error {
      message
      retryable
```

©2026 Tanium Inc. All Rights Reserved

```
      timedOut
    }
  }
}
```

Include an ID variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "runID": "123"
}
```

Example response:

```
{
  "data": {
    "connectConnectionRunStop": {
      "run": {
        "id": "123",
        "connectionID": "234",
        "status": "Running",
        "createdTime": "2023-08-15T18:36:29.739Z",
        "updatedTime": "2023-08-15T18:36:32.235Z"
      },
      "error": null
    }
  }
}
```

Start multiple Connect connection runs

(mutation.connectConnectionsBulkStart)

About this task

The following mutation starts a run for each connection ID in `connectionIDs`. Each connection is processed independently, and each failure is recorded in `errors` with the connection ID, its position in the request list, and a structured error. `connectionIDs` is limited to 100 IDs per request by default.

```
mutation exampleBulkStartConnections($connectionIDs: [ID!]!) {
  connectConnectionsBulkStart(input: {connectionIDs: $connectionIDs}) {
    runs {
      id
      connectionID
      status
      createdTime
      updatedTime
    }
    successCount
    failureCount
    errors {
      connectionID
      inputIndex
      error {
        code
        message
```

©2026 Tanium Inc. All Rights Reserved

```
        retryable
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include an ID list variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionIDs": [
    "123",
    "456"
  ]
}
```

Example response with one success and one failure:

```
{
  "data": {
    "connectConnectionsBulkStart": {
      "runs": [
        {
          "id": "12345",
          "connectionID": "123",
          "status": "Starting",
          "createdTime": "2026-07-17T18:16:54.509Z",
          "updatedTime": "2026-07-17T18:16:54.512Z"
        }
      ],
      "successCount": 1,
      "failureCount": 1,
      "errors": [
        {
          "connectionID": "456",
          "inputIndex": 1,
          "error": {
            "code": "validation_exists",
            "message": "Connection 456 does not exist or is not visible to the caller.",
            "retryable": null
          }
        }
      ],
      "error": null
    }
  }
}
```

Stop multiple Connect connection runs

(mutation.connectConnectionsBulkStop)

About this task

The following mutation stops the active run, if any, for each connection ID in `connectionIDs`. Connections with no run in progress are counted as successes. `connectionIDs` is limited to 100 IDs per request by default.

```
mutation exampleBulkStopConnections($connectionIDs: [ID!]!) {
  connectConnectionsBulkStop(input: {connectionIDs: $connectionIDs}) {
    stoppedRunIDs
    successCount
    failureCount
    errors {
      connectionID
      inputIndex
      error {
        code
        message
        retryable
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include an ID list variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionIDs": [
    "123",
    "456"
  ]
}
```

Example response:

```
{
  "data": {
    "connectConnectionsBulkStop": {
      "stoppedRunIDs": [
        "78901"
      ],
      "successCount": 2,
      "failureCount": 0,
      "errors": [],
      "error": null
    }
  }
}
```

Enable multiple Connect connections for scheduled runs

(mutation.connectConnectionsBulkEnable)

©2026 Tanium Inc. All Rights Reserved

About this task

The following mutation enables each connection ID in `connectionIDs` for scheduled runs. Connections are processed independently. `connectionIDs` is limited to 100 IDs per request by default.

```
mutation exampleBulkEnableConnections($connectionIDs: [ID!]!) {
  connectConnectionsBulkEnable(input: {connectionIDs: $connectionIDs}) {
    connections {
      id
      displayName
    }
    successCount
    failureCount
    errors {
      connectionID
      inputIndex
      error {
        code
        message
        retryable
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include an ID list variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionIDs": [
    "123",
    "456"
  ]
}
```

Example response:

```
{
  "data": {
    "connectConnectionsBulkEnable": {
      "connections": [
        {
          "id": "123",
          "displayName": "Trends — Endpoint Inventory"
        },
        {
          "id": "456",
          "displayName": "Comply Findings Export"
        }
      ],
      "successCount": 2,
      "failureCount": 0,
      "errors": [],
      "error": null
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Disable multiple Connect connections from scheduled runs

(mutation.connectConnectionsBulkDisable)

About this task

The following mutation disables each connection ID in `connectionIDs` from scheduled runs. Connections are processed independently. `connectionIDs` is limited to 100 IDs per request by default.

```
mutation exampleBulkDisableConnections($connectionIDs: [ID!]!) {
  connectConnectionsBulkDisable(input: {connectionIDs: $connectionIDs}) {
    connections {
      id
      displayName
    }
    successCount
    failureCount
    errors {
      connectionID
      inputIndex
      error {
        code
        message
        retryable
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include an ID list variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionIDs": [
    "123",
    "456"
  ]
}
```

Example response:

```
{
  "data": {
    "connectConnectionsBulkDisable": {
      "connections": [
        {
          "id": "123",
          "displayName": "Trends — Endpoint Inventory"
        },
```

©2026 Tanium Inc. All Rights Reserved

```
        {
          "id": "456",
          "displayName": "Comply Findings Export"
        }
      ],
      "successCount": 2,
      "failureCount": 0,
      "errors": [],
      "error": null
    }
  }
}
```

Delete multiple Connect connections

(mutation.connectConnectionsBulkDelete)

About this task

The following mutation deletes each connection ID in `connectionIDs`. Connections are processed independently. Set `deleteComponents` to `false` to preserve the source, destination, filter, and format components for reuse by other connections. `connectionIDs` is limited to 100 IDs per request by default.

```
mutation exampleBulkDeleteConnections($connectionIDs: [ID!]!, $deleteComponents: Boolean) {
  connectConnectionsBulkDelete(input: {connectionIDs: $connectionIDs, deleteComponents:
$deleteComponents}) {
    deletedIDs
    successCount
    failureCount
    errors {
      connectionID
      inputIndex
      error {
        code
        message
        retryable
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the ID list and `deleteComponents` variables in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionIDs": [
    "123",
    "456"
  ],
```

©2026 Tanium Inc. All Rights Reserved

```
  "deleteComponents": true
}
```

Example response:

```
{
  "data": {
    "connectConnectionsBulkDelete": {
      "deletedIDs": [
        "123",
        "456"
      ],
      "successCount": 2,
      "failureCount": 0,
      "errors": [],
      "error": null
    }
  }
}
```

Transfer ownership of multiple Connect connections

(mutation.connectConnectionsBulkTransferOwnership)

About this task

The following mutation transfers ownership of each connection ID in `connectionIDs` to the calling user. This mutation cannot transfer ownership to a third party. The calling user always takes ownership. The persona of each connection is reset to the active persona of the calling user. Connections are processed independently. `connectionIDs` is limited to 100 IDs per request by default.

```
mutation exampleBulkTransferOwnershipConnections($connectionIDs: [ID!]!) {
  connectConnectionsBulkTransferOwnership(input: {connectionIDs: $connectionIDs}) {
    connections {
      id
      displayName
    }
    successCount
    failureCount
    errors {
      connectionID
      inputIndex
      error {
        code
        message
        retryable
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include an ID list variable in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionIDs": [
    "123",
    "456"
  ]
}
```

Example response:

```
{
  "data": {
    "connectConnectionsBulkTransferOwnership": {
      "connections": [
        {
          "id": "123",
          "displayName": "Trends — Endpoint Inventory"
        },
        {
          "id": "456",
          "displayName": "Comply Findings Export"
        }
      ],
      "successCount": 2,
      "failureCount": 0,
      "errors": [],
      "error": null
    }
  }
}
```

Change the persona of multiple Connect connections

(mutation.connectConnectionsBulkChangePersona)

About this task

The following mutation changes the persona that each connection ID in `connectionIDs` runs as. The calling user must own every connection in the request. Set `targetPersonaID` to `0` to use the default persona of the owner at run time, or specify a persona ID available to the calling user. Connections are processed independently. `connectionIDs` is limited to 100 IDs per request by default.

```
mutation exampleBulkChangePersonaConnections($connectionIDs: [ID!]!, $targetPersonaID: ID!) {
  connectConnectionsBulkChangePersona(input: {connectionIDs: $connectionIDs, targetPersonaID:
$targetPersonaID}) {
    connections {
      id
      displayName
    }
    successCount
    failureCount
    errors {
      connectionID
      inputIndex
```

©2026 Tanium Inc. All Rights Reserved

```
      error {
        code
        message
        retryable
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the ID list and `targetPersonaID` variables in the **QUERY VARIABLES** panel or in your variables dictionary:

```
{
  "connectionIDs": [
    "123",
    "456"
  ],
  "targetPersonaID": "0"
}
```

Example response:

```
{
  "data": {
    "connectConnectionsBulkChangePersona": {
      "connections": [
        {
          "id": "123",
          "displayName": "Trends — Endpoint Inventory"
        },
        {
          "id": "456",
          "displayName": "Comply Findings Export"
        }
      ],
      "successCount": 2,
      "failureCount": 0,
      "errors": [],
      "error": null
    }
  }
}
```

Deploy examples

The following queries and mutation require Deploy, retrieve information about software packages deployed in your environment, and allow you to deploy a software package to endpoints.

©2026 Tanium Inc. All Rights Reserved

Deploy a package to all endpoints

(mutation.manageSoftware)

About this task

The following mutation deploys a package to `All Computers`.

```
mutation exampleDeployPackage($group:String) {
  manageSoftware(
    operation: INSTALL
    softwarePackageID: 2
    start: "2021-10-27T00:00:00Z"
    end: "2021-11-03T00:00:00Z"
    target: {targetGroup: $group}
  ) {
    ID
    name
  }
}
```

Include the computer group variable in the **Variables** panel or in your variables dictionary:

```
{
  "group": "All Computers"
}
```

Example response:

```
{
  "data": {
    "manageSoftware": {
      "ID": "2",
      "name": "Install Tanium Standard Utilities (Linux)"
    }
  }
}
```

Get all deploy packages

(query.softwarePackages)

About this task

The following query retrieves all Deploy packages.

```
query exampleGetDeployPackages {
  softwarePackages {
    edges {
      node {
        id
        productName
        productVendor
```

©2026 Tanium Inc. All Rights Reserved

```
        productVersion
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "softwarePackages": {
      "edges": [
        {
          "node": {
            "id": "19",
            "productName": "Firefox (x64 en-US)",
            "productVendor": "Mozilla",
            "productVersion": "98.0"
          }
        },
        {
          "node": {
            "id": "30",
            "productName": "Power BI Desktop (x64)",
            "productVendor": "Microsoft",
            "productVersion": "2.102.845.0"
          }
        },
        {
          "node": {
            "id": "43",
            "productName": "VLC media player (64-bit)",
            "productVendor": "VideoLAN",
            "productVersion": "3.0.16.0"
          }
        },
        {
          "node": {
            "id": "46",
            "productName": "Visual Studio Code (x64 en-us)",
            "productVendor": "Microsoft",
            "productVersion": "1.65.2"
          }
        }
      ]
    }
  }
}
```

Get status of software deployment

(query.softwareDeployment)

About this task

The following query retrieves the deployment status of all Deploy packages.

```
query exampleGetSoftwareDeploymentStatus {
```

```
  softwareDeployment {
    ID
    name
    status {
      completeCount
      downloadCompleteWaitingCount
      downloadingCount
      failedCount
      notApplicableCount
      runningCount
      waitingCount
    }
    errors {
      error
      count
    }
  }
}
```

Example response:

```
{
  "data": {
    "softwareDeployment": [
      {
        "ID": "5",
        "name": "Install Tanium Standard Utilities",
        "status": {
          "completeCount": 1,
          "downloadCompleteWaitingCount": 0,
          "downloadingCount": 0,
          "failedCount": 0,
          "notApplicableCount": 0,
          "runningCount": 0,
          "waitingCount": 0
        },
        "errors": null
      },
      {
        "ID": "6",
        "name": "Install Tanium Standard Utilities (Linux)",
        "status": {
          "completeCount": 0,
          "downloadCompleteWaitingCount": 0,
          "downloadingCount": 1,
          "failedCount": 0,
          "notApplicableCount": 0,
          "runningCount": 0,
          "waitingCount": 0
        },
        "errors": null
      }
    ]
  }
}
```

LEGACY: Get details of all packages

(query.packages)

©2026 Tanium Inc. All Rights Reserved

About this task

> **Note:** This is legacy in favor of `query.packageSpec`. For more information, see the `query.packageSpecs` example syntax.

The following query retrieves multiple fields for all packages.

```
query examplePackagesQuery {
  packages {
    items {
      id
      name
      displayName
      command
      commandTimeout
      expireSeconds
      contentSet {
        id
        name
      }
      processGroupFlag
      skipLockFlag
      metadata {
        adminFlag
        name
        value
      }
      sourceHash
      sourceHashChangedFlag
      sourceID
      sourceName
      parameters {
        key
        value
      }
      rawParameterDefinition
      parameterDefinition {
        parameterType
        model
        parameters {
          model
          parameterType
          key
          label
          helpString
          defaultValue
          validationExpressions {
            model
            parameterType
            expression
            helpString
          }
          promptText
          heightInLines
          maxChars
          values
          restrict
```

```
        allowEmptyList
        minimum
        maximum
        stepSize
        snapInterval
        dropdownOptions {
          model
          parameterType
          name
          value
        }
        componentType
        startDateRestriction {
          model
          parameterType
          type
          interval
          intervalCount
          unixTimeStamp
        }
        endDateRestriction {
          model
          parameterType
          type
          interval
          intervalCount
          unixTimeStamp
        }
        startTimeRestriction {
          model
          parameterType
          type
          interval
          intervalCount
          unixTimeStamp
        }
        endTimeRestriction {
          model
          parameterType
          type
          interval
          intervalCount
          unixTimeStamp
        }
        allowDisableEnd
        defaultRangeStart {
          model
          parameterType
          type
          interval
          intervalCount
          unixTimeStamp
        }
        defaultRangeEnd {
          model
          parameterType
          type
          interval
          intervalCount
          unixTimeStamp
        }
        separatorText
      }
    }
```

©2026 Tanium Inc. All Rights Reserved

```
      verifyExpireSeconds
    }
  }
}
```

Example response:

```
{
  "data": {
    "packages": {
      "items": [
        {
          "id": "1",
          "name": "Distribute Tanium Standard Utilities",
          "displayName": "Distribute Tanium Standard Utilities",
          "command": "cmd.exe /c cscript.exe //E:VBScript install-standard-utils.vbs
\"Tools\\StdUtils\"",
          "commandTimeout": 2700,
          "expireSeconds": 3300,
          "contentSet": {
            "id": "5",
            "name": "Client Management"
          },
          "processGroupFlag": true,
          "skipLockFlag": false,
          "metadata": [],
          "sourceHash": "60b3e906f92929da67341792db9675d5cd91686546f01b57857686c8c6d84fa8",
          "sourceHashChangedFlag": false,
          "sourceID": 0,
          "sourceName": "",
          "parameters": [],
          "rawParameterDefinition": null,
          "parameterDefinition": null,
          "verifyExpireSeconds": 600
        },
        {
          "id": "2",
          "name": "Distribute Tanium Standard Utilities (Linux)",
          "displayName": "Distribute Tanium Standard Utilities (Linux)",
          "command": "/bin/bash distribute-tools.sh STRICT",
          "commandTimeout": 120,
          "expireSeconds": 720,
          "contentSet": {
            "id": "5",
            "name": "Client Management"
          },
          "processGroupFlag": true,
          "skipLockFlag": false,
          "metadata": [],
          "sourceHash": "4ed7a30a1ca6c81be5a71b892dfadcdb489122cc641fa4b644f53255134215c9",
          "sourceHashChangedFlag": false,
          "sourceID": 0,
          "sourceName": "",
          "parameters": [],
          "rawParameterDefinition": null,
          "parameterDefinition": null,
          "verifyExpireSeconds": 600
        }
      ]
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Direct Connect examples

The following queries and mutations use Direct Connect to connect to a single endpoint, retrieve data, stop a process, and then close the connection. Queries that retrieve information from endpoints or terminate processes on endpoints require Performance.

Direct Connect requests have the following feature-specific dependencies at the specified minimum versions:

- Direct Connect 1.10.39 or later is required to submit a request opening, closing, or getting status of Direct Connect connections (`mutation.directConnectClose`, `mutation.directConnectOpen`, `mutation.directConnectPing`, `mutation.directConnectConnectionStatus`).
- Performance 1.10.57 or later is required to submit a request retrieving Performance data from endpoints using Direct Connect or terminating processes using Direct Connect (`mutation.directConnectProcessTerminate`, `query.directConnectEndpoint`).



Note: Note: For the best results, install Direct Connect 2.7.43 or later.

Open direct connection to endpoint

(mutation.directConnectOpen)

About this task

The following mutation uses Direct Connect to establish a connection to the endpoint with an ID of `12323`. You can retrieve IDs through the Get endpoints IDs from Tanium Data Service query. If a direct connection status is not `READY` (such as `CONNECTING` or `UNKNOWN`) after you use this mutation, you can use the Get direct connection status query to check the direct connection status.

Connections opened with `directConnectOpen` are compatible with stable Direct Connect APIs, including `directConnectEndpoint`, `directConnectPing`, `directConnectProcessTerminate`, and `directConnectClose`. Connections opened with `directConnectOpen` are not compatible with legacy Direct Connect APIs, including

©2026 Tanium Inc. All Rights Reserved

`directEndpoint`, `pingDirectConnection`, `killProcess`, and `closeDirectConnection`.



Note: Direct Connect connections close after five minutes of inactivity.

```
mutation exampleOpenEndpointConnection($id: ID!) {
  directConnectOpen(input: {endpointID: $id}) {
    connectionID
    status
  }
}
```

Include the endpoint ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12323"
}
```

Example response:

```
{
  "data": {
    "directConnectOpen": {
      "connectionID": "86d9a9ac-0229-481b-9d88-5f1bcb1b177b",
      "status": "READY"
    }
  }
}
```

Get direct connection status

(query.directConnectConnectionStatus)

About this task

The following query uses Direct Connect to retrieve the status of a direct connection. If a direct connection status is not `READY` (such as `CONNECTING` or `UNKNOWN`) after you use the Open direct connection to endpoint mutation to establish a direct connection to an endpoint, you can use the connection ID from that response in this query to determine whether the direct connection is established.

The input requires a direct endpoint connection opened using `directConnectOpen`.

©2026 Tanium Inc. All Rights Reserved

Note: Direct Connect connections close after five minutes of inactivity.

```
query exampleGetDCStatus($id: ID!) {
  directConnectConnectionStatus(input: {connectionID: $id}) {
    status
  }
}
```

Include the connection ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "86d9a9ac-0229-481b-9d88-5f1bcb1b177b"
}
```

Example response:

```
{
  "data": {
    "directConnectConnectionStatus": {
      "status": "READY"
    }
  }
}
```

Ping direct connection to endpoint

(mutation.directConnectPing)

About this task

The following mutation retrieves whether a Direct Connect connection is active. Use this mutation to check connection details or to keep the connection active.

The input requires a direct endpoint connection opened using `directConnectOpen`.

Note: Direct Connect connections close after five minutes of inactivity.

```
mutation examplePingDirectConnection($connectionID: ID!) {
  directConnectPing(input: {connectionID: $connectionID}) {
    result
  }
}
```

Include the connection ID in the **Variables** panel or in your variables dictionary:

```
{
  "connectionID": "5fc564d6-5767-47fc-abb6-25cba65409d8"
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Example response:

```
{
  "data": {
    "directConnectPing": {
      "result": true
    }
  }
}
```

Get data from endpoint

(query.directConnectEndpoint.performance)

About this task

After you establish a connection to an endpoint through Direct Connect, you can query the endpoint for specific information. The input requires a direct endpoint connection opened using `directConnectOpen`.

The following query retrieves the CPU usage on the endpoint:

```
query exampleGetEndpointData($id: ID!) {
  directConnectEndpoint(input: {connectionID: $id}) {
    performance {
      cpuUsagePercent
    }
  }
}
```

Include the connection ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef"
}
```

Example response:

```
{
  "data": {
    "directConnectEndpoint": {
      "performance": {
        "cpuUsagePercent": 28.751501243887798
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Get process from endpoint

(query.directConnectEndpoint.processes)

About this task

After you establish a connection to an endpoint through Direct Connect, you can query the endpoint for process information. The input requires a direct endpoint connection opened using `directConnectOpen`.

The following query retrieves the state of all processes running on the endpoint.

```
query exampleGetEndpointProcess($id: ID!) {
  directConnectEndpoint(input: {connectionID: $id}) {
    processes {
      all {
        pid
        ppid
        name
        commandLine
        userName
        groupName
        memoryResidentBytes
      }
    }
  }
}
```

Include the endpoint ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12323"
}
```

Example response:

```
{
  "data": {
    "directConnectEndpoint": {
      "processes": {
        "all": [
          {
            "pid": 2092,
            "ppid": 496,
            "name": "TaniumReceiver.exe",
            "commandLine": "\"C:\\Program Files\\Tanium\\Tanium Server\\TaniumReceiver.exe\"
--service",
            "userName": "admin",
            "groupName": "test-group",
            "memoryResidentBytes": 59842560
          },
          {
            "pid": 5760,
            "ppid": 1112,
            "name": "TaniumClient.exe",
```

©2026 Tanium Inc. All Rights Reserved

```
            "commandLine": "\"C:\\Program Files (x86)\\Tanium\\Tanium
Client\\TaniumClient.exe\" -c",
            "userName": "SYSTEM",
            "groupName": "NT AUTHORITY",
            "memoryResidentBytes": 17965056
          },
          {
            "pid": 1036,
            "ppid": 496,
            "name": "TaniumBlobService.exe",
            "commandLine": "\"C:\\Program Files\\Tanium\\Tanium Module
Server\\services\\blob-service\\TaniumBlobService.exe\"",
            "userName": "SYSTEM",
            "groupName": "NT AUTHORITY",
            "memoryResidentBytes": 7426048
          }
        ]
      }
    }
  }
}
```

Get past 24 hours of alerts from endpoint

(query.directConnectEndpoint.alerts)

About this task

After you establish a connection to an endpoint through Direct Connect, you can query the endpoint for alert information. You can also define a time period for retrieving events based on the following rules:

- The end time must be before the current time.
- The start time must be before the end time.
- If you define a start time without an end time, the request time period retrieves alerts from the start time to the current time.
- If you do not define a start time or end time, the request defaults to the past 24 hours.

The input requires a direct endpoint connection opened using `directConnectOpen`.

The following query retrieves alerts from an endpoint that occurred during the past 24 hours.

```
query exampleGetEndpointAlerts($id: ID!) {
  directConnectEndpoint(input: {connectionID: $id}) {
    alerts {
      all {
        schema
        key
        type
        ref
```

©2026 Tanium Inc. All Rights Reserved

```
        topProcessesExpr
        labels
        pendingAt
        start
        resolvedAt
        leadup
        value
      }
    }
  }
}
```

Include the connection ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef"
}
```

Example response:

```
{
  "data": {
    "directConnectEndpoint": {
      "alerts": {
        "all": [
          {
            "schema": 1,
            "key": "available-mem{heuristic=\"available-mem\"}",
            "type": "available-mem",
            "ref": null,
            "topProcessesExpr": null,
            "labels": {
              "heuristic": "available-mem"
            },
            "pendingAt": "2022-03-15T15:54:38.574990164Z",
            "start": "2022-03-15T15:54:38.574990164Z",
            "resolvedAt": null,
            "leadup": 300000000000,
            "value": 168.48828125
          }
        ]
      }
    }
  }
}
```

Get 48 hours of alerts from endpoint

The following query sets a time period of 48 hours from January 1, 2022 to January 3, 2022 and retrieves alerts for an endpoint from that period.

```
query exampleGetEndpointAlerts48Hours($id: ID!, $startTime: Time!, $endTime: Time) {
  directConnectEndpoint(input: {connectionID: $id}) {
    alerts {
      all(scope: {startTime: $startTime, endTime: $endTime}) {
        schema
        key
        type
```

©2026 Tanium Inc. All Rights Reserved

```
        ref
        topProcessesExpr
        labels
        pendingAt
        start
        resolvedAt
        leadup
        value
      }
    }
  }
}
```

Include the connection ID and time period variables in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "startTime": "2022-01-01T00:00:00Z",
  "endTime": "2022-01-03T00:00:00Z"
}
```

Get alerts from prior date from endpoint

The following query sets a time period start date of January 1, 2022 and retrieves alerts for an endpoint from that date to the current time.

```
query exampleGetEndpointAlertsPriorDate($id: ID!, $startTime: Time!) {
  directConnectEndpoint(input: {connectionID: $id}) {
    alerts {
      all(scope: {startTime: $startTime}) {
        schema
        key
        type
        ref
        topProcessesExpr
        labels
        pendingAt
        start
        resolvedAt
        leadup
        value
      }
    }
  }
}
```

Include the connection ID and start time variables in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "startTime": "2022-01-01T00:00:00Z"
}
```

©2026 Tanium Inc. All Rights Reserved

Stop endpoint process

(mutation.directConnectProcessTerminate)

About this task

After you establish a connection to an endpoint through Direct Connect, you can stop running processes on the endpoint. The following mutation stops a process named `notepad.exe` on an endpoint.

The input requires a direct endpoint connection opened using `directConnectOpen`.

```
mutation exampleStopEndpointProcess($connectionID: ID!, $processName: String!, $pid: Int!,
$signal: Signal!) {
  directConnectProcessTerminate(
    input: {connectionID: $connectionID, name: $processName, pid: $pid, signal: $signal}
  ) {
    result
  }
}
```

Include the connection ID and process name variables in the **Variables** panel or in your variables dictionary:

```
{
  "connectionID": "5fc564d6-5767-47fc-abb6-25cba65409d8",
  "processName": "notepad.exe",
  "pid": 7056,
  "signal": "SIGKILL"
}
```

Example response:

```
{
  "data": {
    "directConnectProcessTerminate": {
      "result": true
    }
  }
}
```

Close endpoint connection

(mutation.directConnectClose)

About this task

The following mutation closes a Direct Connect connection to an endpoint.

©2026 Tanium Inc. All Rights Reserved

Note: Direct Connect connections close after five minutes of inactivity.

```
mutation exampleCloseEndpointConnection($connectionID: ID!) {
  directConnectClose(input: {connectionID: $connectionID}) {
    result
  }
}
```

Include the connection ID in the **Variables** panel or in your variables dictionary:

```
{
  "connectionID": "5fc564d6-5767-47fc-abb6-25cba65409d8"
}
```

Example response:

```
{
  "data": {
    "directConnectClose": {
      "result": true
    }
  }
}
```

LEGACY: Open a connection to an endpoint

(mutation.openDirectConnection)

Note: This is legacy functionality. Use `mutation.directConnectOpen` instead.

The following mutation uses Direct Connect to establish a connection to the endpoint with an ID of `12323`. You can retrieve IDs through the Get endpoints IDs from Tanium Data Service query.

Connections opened with `openDirectConnection` are compatible with legacy Direct Connect APIs, including `directEndpoint`, `pingDirectConnection`, `killProcess`, and `closeDirectConnection`. Connections opened with `openDirectConnection` are not compatible with stable Direct Connect APIs, including `directConnectEndpoint`, `directConnectPing`, `directConnectProcessTerminate`, and `directConnectClose`.

Note: Direct Connect connections close after two minutes of inactivity.

©2026 Tanium Inc. All Rights Reserved

```
mutation exampleOpenEndpointConnection($id: ID) {
  openDirectConnection(input: {endpointID: $id}) {
    connectionID
  }
}
```

Include the endpoint ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12323"
}
```

Example response:

```
{
  "data": {
    "openDirectConnection": {
      "connectionID": "86d9a9ac-0229-481b-9d88-5f1bcb1b177b"
    }
  }
}
```

LEGACY: Ping the connection to an endpoint

(mutation.pingDirectConnection)



Note: This is legacy functionality. Use `mutation.directConnectPing` instead.

The following mutation retrieves the status for a Direct Connect connection. Use this mutation to check connection details or to keep the connection active.

The input requires a direct endpoint connection opened using `openDirectConnection`.



Note: Direct Connect connections close after two minutes of inactivity.

```
mutation ($connectionID: ID!) {
  pingDirectConnection(input: {connectionID: $connectionID}) {
    result
  }
}
```

Include the connection ID in the **Variables** panel or in your variables dictionary:

```
{
  "connectionID": "5fc564d6-5767-47fc-abb6-25cba65409d8"
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:

```
{
  "data": {
    "pingDirectConnection": {
      "result": true
    }
  }
}
```

LEGACY: Get data from an endpoint

(query.directEndpoint.performance)

> **Note:** This is legacy functionality. Use `query.directConnectEndpoint.performance` instead.

After you establish a connection to an endpoint through Direct Connect, you can query the endpoint for specific information.

The input requires a direct endpoint connection opened using `openDirectConnection`.

The following query retrieves the CPU usage on the endpoint:

```
query exampleGetEndpointData($id: ID) {
  directEndpoint(input : {endpointID: $id}) {
    performance {
      cpuUsagePercent
    }
  }
}
```

Include the endpoint ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12323"
}
```

Example response:

```
{
  "data": {
    "directEndpoint": {
      "performance": {
        "cpuUsagePercent": 28.751501243887798
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

LEGACY: Get process from an endpoint

(query.directEndpoint.processes)

> **Note:** This is legacy functionality. Use `query.directConnectEndpoint.processes` instead.

After you establish a connection to an endpoint through Direct Connect, you can query the endpoint for process information.

The input requires a direct endpoint connection opened using `openDirectConnection`.

The following query retrieves the state of all processes running on the endpoint.

```
query exampleGetEndpointProcess($id: ID) {
  directEndpoint(input: {endpointID: $id}) {
    processes {
      all {
        pid
        ppid
        name
        commandLine
        userName
        groupName
        memoryResidentBytes
      }
    }
  }
}
```

Include the endpoint ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12323"
}
```

Example response:

```
{
  "data": {
    "directEndpoint": {
      "processes": {
        "all": [
          {
            "pid": 2092,
            "ppid": 496,
            "name": "TaniumReceiver.exe",
            "commandLine": "\"C:\\Program Files\\Tanium\\Tanium Server\\TaniumReceiver.exe\"
--service",
            "userName": "admin",
```

©2026 Tanium Inc. All Rights Reserved

```
            "groupName": "test-group",
            "memoryResidentBytes": 59842560
          },
          {
            "pid": 5760,
            "ppid": 1112,
            "name": "TaniumClient.exe",
            "commandLine": "\"C:\\Program Files (x86)\\Tanium\\Tanium
Client\\TaniumClient.exe\" -c",
            "userName": "SYSTEM",
            "groupName": "NT AUTHORITY",
            "memoryResidentBytes": 17965056
          },
          {
            "pid": 1036,
            "ppid": 496,
            "name": "TaniumBlobService.exe",
            "commandLine": "\"C:\\Program Files\\Tanium\\Tanium Module
Server\\services\\blob-service\\TaniumBlobService.exe\"",
            "userName": "SYSTEM",
            "groupName": "NT AUTHORITY",
            "memoryResidentBytes": 7426048
          }
        ]
      }
    }
  }
}
```


LEGACY - Get alerts from an endpoint

(query.directEndpoint.alerts)

GET PAST 24 HOURS OF ALERTS FROM ENDPOINT



Note: This is legacy functionality. Use `query.directConnectEndpoint.alerts` instead.

After you establish a connection to an endpoint through Direct Connect, you can query the endpoint for alert information. You can also define a time period for retrieving events based on the following rules:

- The end time must be before the current time.
- The start time must be before the end time.
- If you define a start time without an end time, the request time period retrieves alerts from the start time to the current time.

©2026 Tanium Inc. All Rights Reserved

- If you do not define a start time or end time, the request defaults to the past 24 hours.

The input requires a direct endpoint connection opened using `openDirectConnection`.

The following query retrieves alerts from an endpoint that occurred during the past 24 hours.

```
query exampleGetEndpointAlerts($id: ID) {
  directEndpoint(input: {endpointID: $id}) {
    alerts {
      all {
        schema
        key
        type
        ref
        topProcessesExpr
        labels
        pendingAt
        start
        resolvedAt
        leadup
        value
      }
    }
  }
}
```

Include the endpoint ID in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12323"
}
```

Example response:

```
{
  "data": {
    "directEndpoint": {
      "alerts": {
        "all": [
          {
            "schema": 1,
            "key": "available-mem{heuristic=\"available-mem\"}",
            "type": "available-mem",
            "ref": null,
            "topProcessesExpr": null,
            "labels": {
              "heuristic": "available-mem"
            },
            "pendingAt": "2022-03-15T15:54:38.574990164Z",
            "start": "2022-03-15T15:54:38.574990164Z",
            "resolvedAt": null,
            "leadup": 300000000000,
            "value": 168.48828125
          }
        ]
      }
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

GET 48 HOURS OF ALERTS FROM ENDPOINT

> **Note:** This is legacy functionality. Use `query.directConnectEndpoint.alerts` instead.

The following query sets a time period of 48 hours from January 1, 2022 to January 3, 2022 and retrieves alerts for an endpoint from that period.

The input requires a direct endpoint connection opened using `openDirectConnection`.

```
query exampleGetEndpointAlerts48Hours($id: ID, $startTime: Time!, $endTime: Time) {
  directEndpoint(input: {endpointID: $id}) {
    alerts {
      all(scope: {startTime: $startTime, endTime: $endTime}) {
        schema
        key
        type
        ref
        topProcessesExpr
        labels
        pendingAt
        start
        resolvedAt
        leadup
        value
      }
    }
  }
}
```

Include the endpoint ID and time period variables in the **Variables** panel or in your variables dictionary:

```
{
  "id": "123",
  "startTime": "2022-01-01T00:00:00Z",
  "endTime": "2022-01-03T00:00:00Z"
}
```

GET ALERTS FROM PRIOR DATE FROM ENDPOINT

> **Note:** This is legacy functionality. Use `query.directConnectEndpoint.alerts` instead.

©2026 Tanium Inc. All Rights Reserved

The following query sets a time period start date of January 1, 2022 and retrieves alerts for an endpoint from that date to the current time.

The input requires a direct endpoint connection opened using `openDirectConnection`.

```
query exampleGetEndpointAlertsPriorDate($id: ID, $startTime: Time!) {
  directEndpoint(input: {endpointID: $id}) {
    alerts {
      all(scope: {startTime: $startTime}) {
        schema
        key
        type
        ref
        topProcessesExpr
        labels
        pendingAt
        start
        resolvedAt
        leadup
        value
      }
    }
  }
}
```

Include the endpoint ID and start time variables in the **Variables** panel or in your variables dictionary:

```
{
  "id": "123",
  "startTime": "2022-01-01T00:00:00Z"
}
```

LEGACY - Stop a process on an endpoint

(mutation.killProcess)

> **Note:** This is legacy functionality. Use `mutation.directConnectProcessTerminate` instead.

After you establish a connection to an endpoint through Direct Connect, you can stop running processes on the endpoint. The following mutation stops a process named `notepad.exe` on an endpoint.

The input requires a direct endpoint connection opened using `openDirectConnection`.

```
mutation exampleStopEndpointProcess($connectionID: ID, $processName: String!, $pid: Int!,
$signal: Signal!) {
```

©2026 Tanium Inc. All Rights Reserved

```
  killProcess(
    input: {connectionID: $connectionID, name: $processName, pid: $pid, signal: $signal}
  ) {
    result
  }
}
```

Include the connection ID and process name variables in the **Variables** panel or in your variables dictionary:

```
{
  "connectionID": "5fc564d6-5767-47fc-abb6-25cba65409d8",
  "processName": "notepad.exe",
  "pid": 7056,
  "signal": "SIGKILL"
}
```

Example response:

```
{
  "data": {
    "killProcess": {
      "result": true
    }
  }
}
```

LEGACY - Close connection to an endpoint

(mutation.closeDirectConnection)



Note: This is legacy functionality. Use `mutation.directConnectClose` instead.

The following mutation closes a Direct Connect connection to an endpoint.

The input requires a direct endpoint connection opened using `openDirectConnection`.



Note: Direct Connect connections close after two minutes of inactivity.

```
mutation exampleCloseEndpointConnection($connectionID: ID!) {
  closeDirectConnection(input: {connectionID: $connectionID}) {
    result
  }
}
```

Include the connection ID in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "connectionID": "5fc564d6-5767-47fc-abb6-25cba65409d8"
}
```

Example response:

```
{
  "data": {
    "closeDirectConnection": {
      "result": true
    }
  }
}
```

Discover examples

The following queries and mutation require Discover, retrieve information about Discover labels and interfaces, and allow you to create or delete labels, or update labels applied to interfaces.

Discover requests have the following feature-specific dependencies at the specified minimum versions:

- Discover 4.8.298 or later is required to submit a request involving Discover interfaces or labels (`mutation.discoverInterfaceAddLabel`, `mutation.discoverInterfaceRemoveLabel`, `mutation.discoverLabelCreate`, `mutation.discoverLabelDelete`, `query.discoverInterface`, `query.discoverInterfaces`, `query.discoverLabel`, `query.discoverLabels`).

Add Discover labels to Discover interfaces

`(mutation.discoverInterfaceAddLabel)`

About this task

The `discoverInterfaceAddLabel` mutation requires Discover 4.8.298 or later and the **Discover Tag write** permission.

The following mutation manually adds one or more standard labels to one or more specified interfaces. This mutation does not support automatic evaluation to apply labels.

```
mutation exampleAddInterfaceLabel($labelNames: [String!]!, $interfaceIDs: [ID!]!) {
  discoverInterfaceAddLabel(input: {names: $labelNames, interfaceIDs: $interfaceIDs}) {
    success
    error {
```

©2026 Tanium Inc. All Rights Reserved

```
      message
    }
    cursor
  }
}
```

Include the interface variables in the **Variables** panel or in your variables dictionary:

```
{
  "labelNames": [
    "example_label",
    "example_label_2"
  ],
  "interfaceIDs": [
    12345,
    67890
  ]
}
```

Example response:

```
{
  "data": {
    "discoverInterfaceAddLabel": {
      "success": true,
      "error": null,
      "cursor": null
    }
  }
}
```

If your mutation request returns a cursor, the label is not yet added to the interfaces that you specify. You can pass the cursor as input to a subsequent `discoverInterfaceAddLabel` mutation to determine the status of the initial request.

```
mutation exampleAddInterfaceLabel($labelNames: [String!]!, $interfaceIDs: [ID!]!) {
  discoverInterfaceAddLabel(
    input: {
      labelNames: $labelNames,
      interfaceIDs: $interfaceIDs
      cursor: "Mg=="
    }
  ) {
    success
    error {
      message
    }
    cursor
  }
}
```

Create a Discover label

`(mutation.discoverLabelCreate)`

©2026 Tanium Inc. All Rights Reserved

About this task

The `discoverLabelCreate` mutation requires Discover 4.8.298 or later and the **Discover Tag Write** permission. If you define at least one Discover location, this also requires the **Discover Location Permissions Write** permission.

> **Note:** Using Gateway, you can create a standard-type Discover label that you use to manually mark interfaces. You cannot create an automatic-type label that applies to interfaces based on configured criteria. To configure an automatic label to apply to interfaces, see Tanium Discover User Guide - Automatically label interfaces.

The following mutation creates a standard-type label.

```
mutation exampleCreateInterfaceLabel($labelName: String!) {
  discoverLabelCreate(input: {name: $labelName}) {
    label {
      name
      type
      createdTime
      modifiedTime
    }
    error {
      message
    }
  }
}
```

Include the name variable in the **Variables** panel or in your variables dictionary:

```
{
  "labelName": "Example_label"
}
```

Example response:

```
{
  "data": {
    "discoverLabelCreate": {
      "label": {
        "name": "Example_label",
        "type": "STANDARD",
        "createdTime": "2023-03-29T13:42:22.760Z",
        "modifiedTime": "2023-03-29T13:42:22.760Z"
      },
      "error": null
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Delete a Discover label by name

`(mutation.discoverLabelDelete)`

About this task

The `discoverLabelDelete` mutation requires Discover 4.8.298 or later and the **Discover Tag write** permission. If you define at least one Discover location, this also requires the **Discover Location Permissions write** permission.

The following mutation deletes a label. If you delete a label, the label is removed from all the related interfaces.

```
mutation exampleDeleteInterfaceLabel($labelName: String!) {
  discoverLabelDelete(ref: {name: $labelName}) {
    name
    error {
      message
    }
  }
}
```

Include the name variable in the **Variables** panel or in your variables dictionary:

```
{
  "labelName": "Example_label"
}
```

Example response:

```
{
  "data": {
    "discoverLabelDelete": {
      "name": "Example_label",
      "error": null
    }
  }
}
```

Get a Discover interface by ID

`(query.discoverInterface)`

About this task

The `discoverInterface` query requires Discover 4.8.298 or later and the **Discover Asset read** permission. If you define at least one Discover location, this also requires either the **User Group read** permission (returns only matching interfaces to which the user group membership allows access) or

©2026 Tanium Inc. All Rights Reserved

the **Discover Location Permissions write** permission (returns all matching interfaces).

Discover versions earlier than 4.11 do not support the `locallyAdministeredMacAddress` field to determine whether a MAC address is locally administered.

The following query retrieves the Discover interface with the ID of `123`.

```
query exampleGetSpecificInterface($id: ID!) {
  discoverInterface(ref: {id: $id}) {
    id
    macAddress
    manufacturer
    locallyAdministeredMacAddress
    hostnames
    ipAddresses
    labels {
      name
      type
      createdTime
      modifiedTime
    }
    osPlatform
    openPorts
    discoveryMethods
    lastSeenTime
    computerId
    natIPAddresses
    osGeneration
    profiles {
      name
    }
    firstSeenTime
    firstManagedTime
    lastManagedTime
    lastDiscoveredTime
    isManaged
    isUnmanageable
    isIgnored
    satellites {
      id
      name
    }
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "123"
}
```

Example response:

```
{
  "data": {
    "discoverInterface": {
      "id": "123",
```

©2026 Tanium Inc. All Rights Reserved

```
      "macAddress": "01-23-45-67-89-AB",
      "manufacturer": "Microsoft Corporation",
      "locallyAdministeredMacAddress": false,
      "hostnames": [
        "USB-81.lan.oxfordcs.com"
      ],
      "ipAddresses": [
        "192.2.2.0"
      ],
      "labels": [
        {
          "name": "M11_Unmanaged_not_Unmanageable",
          "type": "AUTOMATIC",
          "createdTime": "2022-03-15T13:08:37.177Z",
          "modifiedTime": "2022-03-15T13:08:37.177Z"
        },
        {
          "name": "Client Install Pending Fixed",
          "type": "AUTOMATIC",
          "createdTime": "2024-06-25T17:39:34.843Z",
          "modifiedTime": "2024-06-25T17:39:34.843Z"
        },
        {
          "name": "cM_4351_repro",
          "type": "AUTOMATIC",
          "createdTime": "2024-09-30T17:33:10.861Z",
          "modifiedTime": "2024-09-30T17:33:10.861Z"
        },
        {
          "name": "dain_test",
          "type": "STANDARD",
          "createdTime": "2024-11-21T18:44:53.944Z",
          "modifiedTime": "2024-11-21T18:44:53.944Z"
        }
      ],
      "osPlatform": "Windows",
      "openPorts": [
        135,
        139,
        445,
        3389,
        5357,
        49152,
        49153,
        49154,
        49155,
        49156,
        49167
      ],
      "discoveryMethods": [
        "ARP",
        "CONNECTED",
        "PING",
        "NMAP"
      ],
      "lastSeenTime": "2026-03-25T18:39:06.401Z",
      "computerId": null,
      "natIPAddresses": [
        "198.50.20.200"
      ],
      "osGeneration": null,
      "profiles": [
        {
          "name": "Bad CIDR profile"
```


©2026 Tanium Inc. All Rights Reserved

```
        },
        {
          "name": "FXF - Discover Test"
        },
        {
          "name": "SS-NMAP-HomeLab"
        }
      ],
      "firstSeenTime": "2022-09-28T14:16:10.000Z",
      "firstManagedTime": null,
      "lastManagedTime": null,
      "lastDiscoveredTime": "2026-03-25T18:39:06.397Z",
      "isManaged": false,
      "isUnmanageable": false,
      "isIgnored": false,
      "satellites": []
    }
  }
}
```

Get multiple Discover interfaces

(query.discoverInterfaces)

About this task

The `discoverInterfaces` query requires Discover 4.8.298 or later and the **Discover Asset read** permission. If you define at least one Discover location, this also requires either the **User Group read** permission (returns only matching interfaces to which the user group membership allows access) or the **Discover Location Permissions write** permission (returns all matching interfaces).

Discover versions earlier than 4.11 do not support the `locallyAdministeredMacAddress` field to determine whether a MAC address is locally administered.

The following query retrieves the first `2` Discover interfaces.

```
query exampleGetInterfaces($first: Int) {
  discoverInterfaces(first: $first) {
    edges {
      node {
        id
        macAddress
        manufacturer
        locallyAdministeredMacAddress
        hostnames
        ipAddresses
        labels {
          name
          type
          createdTime
          modifiedTime
        }
        osPlatform
        openPorts
```

```
        discoveryMethods
        lastSeenTime
        computerId
        natIPAddresses
        osGeneration
        profiles {
          name
        }
        firstSeenTime
        firstManagedTime
        lastManagedTime
        lastDiscoveredTime
        isManaged
        isUnmanageable
        isIgnored
        satellites {
          id
          name
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "discoverInterfaces": {
      "edges": [
        {
          "node": {
            "id": "1234",
            "macAddress": "01-23-45-67-89-AB",
            "manufacturer": "VMware, Inc.",
            "locallyAdministeredMacAddress": false,
            "hostnames": [],
            "ipAddresses": [
              "192.0.20.10"
            ],
            "labels": [
              {
                "name": "Example label 1",
                "type": "AUTOMATIC",
                "createdTime": "2020-05-20T14:48:30.259Z",
                "modifiedTime": "2025-07-07T19:58:02.976Z"
              },
              {
                "name": "Example label 2",
                "type": "AUTOMATIC",
                "createdTime": "2022-03-15T13:08:37.177Z",
                "modifiedTime": "2022-03-15T13:08:37.177Z"
              },
              {
                "name": "Example label 3",
```

©2026 Tanium Inc. All Rights Reserved

```
            "type": "STANDARD",
            "createdTime": "2026-03-17T13:10:32.462Z",
            "modifiedTime": "2026-03-17T13:10:32.462Z"
          }
        ],
        "osPlatform": "Windows",
        "openPorts": null,
        "discoveryMethods": [
          "ARP",
          "CONNECTED",
          "PING",
          "NMAP",
          "SATELLITE_NMAP",
          "NDP"
        ],
        "lastSeenTime": "2026-03-25T18:39:06.401Z",
        "computerId": null,
        "natIPAddresses": [
          "198.50.100.10"
        ],
        "osGeneration": "10",
        "profiles": [
          {
            "name": "Example profile 1"
          },
          {
            "name": "Example profile 2"
          },
          {
            "name": "Example profile 3"
          }
        ],
        "firstSeenTime": "2021-07-20T01:28:49.000Z",
        "firstManagedTime": "2023-08-14T12:09:41.000Z",
        "lastManagedTime": "2024-01-12T16:47:20.796Z",
        "lastDiscoveredTime": "2026-03-25T16:39:05.325Z",
        "isManaged": false,
        "isUnmanageable": false,
        "isIgnored": false,
        "satellites": [
          {
            "id": "b58a6c1a-79c6-46e5-903d-7975974ef349",
            "name": "satellite-ubuntu24-intel-test1"
          }
        ]
      }
    },
    {
      "node": {
        "id": "4567",
        "macAddress": "CD-EF-01-23-45-67",
        "manufacturer": "VMware, Inc.",
        "locallyAdministeredMacAddress": false,
        "hostnames": [],
        "ipAddresses": [
          "192.10.20.30"
        ],
        "labels": [
          {
            "name": "Example label 2",
            "type": "AUTOMATIC",
            "createdTime": "2022-03-15T13:08:37.177Z",
            "modifiedTime": "2022-03-15T13:08:37.177Z"
          },
```


©2026 Tanium Inc. All Rights Reserved

```
              {
                "name": "Example label 4",
                "type": "AUTOMATIC",
                "createdTime": "2024-03-28T19:05:55.605Z",
                "modifiedTime": "2024-03-28T19:05:55.605Z"
              },
              {
                "name": "Example label 5",
                "type": "AUTOMATIC",
                "createdTime": "2024-06-25T17:39:34.843Z",
                "modifiedTime": "2024-06-25T17:39:34.843Z"
              }
            ],
            "osPlatform": "Windows",
            "openPorts": [
              22,
              135,
              139,
              445,
              3389,
              5357
            ],
            "discoveryMethods": [
              "ARP",
              "CONNECTED",
              "PING",
              "NMAP",
              "NDP"
            ],
            "lastSeenTime": "2026-03-25T18:39:06.401Z",
            "computerId": null,
            "natIPAddresses": [
              "20.70.20.190"
            ],
            "osGeneration": "10",
            "profiles": [
              {
                "name": "Example profile 2"
              },
              {
                "name": "Example profile 4"
              },
              {
                "name": "Example profile 5"
              }
            ],
            "firstSeenTime": "2022-08-05T16:48:40.000Z",
            "firstManagedTime": null,
            "lastManagedTime": null,
            "lastDiscoveredTime": "2026-03-25T18:39:06.397Z",
            "isManaged": false,
            "isUnmanageable": false,
            "isIgnored": false,
            "satellites": []
          }
        }
      ]
    }
  }
}
```


©2026 Tanium Inc. All Rights Reserved

Get a Discover label by name

`(query.discoverLabel)`

About this task

The `discoverLabel` query requires Discover 4.8.298 or later and the **Discover Asset read** permission.

The following query retrieves the Discover label with the name of `Example_label`.

```
query exampleGetSpecificLabel($name: String!) {
  discoverLabel(ref: {name: $name}) {
    name
    type
    createdTime
    modifiedTime
    interfaceCounts {
      managed
      unmanaged
      total
      unmanageable
    }
  }
}
```

Include the name variable in the **Variables** panel or in your variables dictionary:

```
{
  "name": "Example_label"
}
```

Example response:

```
{
  "data": {
    "discoverLabel": {
      "name": "Example label 1",
      "type": "AUTOMATIC",
      "createdTime": "2024-07-23T18:51:58.230Z",
      "modifiedTime": "2024-07-23T18:51:58.230Z",
      "interfaceCounts": {
        "managed": 0,
        "unmanaged": 0,
        "total": 0,
        "unmanageable": 0
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Get multiple Discover labels

`(query.discoverLabels)`

About this task

The `discoverLabels` query requires Discover 4.8.298 or later and the **Discover Asset read** permission.

The following query retrieves the first `2` Discover labels.

```
query exampleGetLabels($first: Int) {
  discoverLabels(first: $first) {
    edges {
      node {
        name
        type
        createdTime
        interfaceCounts {
          managed
          unmanaged
          total
          unmanageable
        }
        modifiedTime
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "discoverLabels": {
      "edges": [
        {
          "node": {
            "name": "Example label 1",
            "type": "AUTOMATIC",
            "createdTime": "2024-07-23T18:51:58.230Z",
            "interfaceCounts": {
              "managed": 0,
              "unmanaged": 0,
              "total": 0,
              "unmanageable": 0
            },
            "modifiedTime": "2024-07-23T18:51:58.230Z"
          }
```

©2026 Tanium Inc. All Rights Reserved

```
        },
        {
          "node": {
            "name": "Example label 2",
            "type": "AUTOMATIC",
            "createdTime": "2020-05-20T14:48:30.268Z",
            "interfaceCounts": {
              "managed": 0,
              "unmanaged": 0,
              "total": 0,
              "unmanageable": 0
            },
            "modifiedTime": "2020-05-20T14:48:30.268Z"
          }
        }
      ]
    }
  }
}
```

Remove Discover labels from Discover interfaces

`(mutation.discoverInterfaceRemoveLabel)`

About this task

The `discoverInterfaceRemoveLabel` mutation requires Discover 4.8.298 or later and the **Discover Tag write** permission.

The following mutation manually removes one or more standard labels from one or more specified interfaces.

```
mutation exampleRemoveInterfaceLabel($labelNames: [String!]!, $interfaceIDs: [ID!]!) {
  discoverInterfaceRemoveLabel(input: {names: $labelNames, interfaceIDs: $interfaceIDs}) {
    success
    error {
      message
    }
    cursor
  }
}
```

Include the interface variables in the **Variables** panel or in your variables dictionary:

```
{
  "labelNames": [
    "example_label",
    "example_label_2"
  ],
  "interfaceIDs": [
    12345,
    67890
  ]
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:

```
{
  "data": {
    "discoverInterfaceRemoveLabel": {
      "success": true,
      "error": null,
      "cursor": null
    }
  }
}
```

If your mutation request returns a cursor, the label is not yet removed from the interfaces that you specify. You can pass the cursor as input to a subsequent `discoverInterfaceRemoveLabel` mutation to determine the status of the initial request.

```
mutation exampleRemoveInterfaceLabel($labelNames: [String!]!, $interfaceIDs: [ID!]!) {
  discoverInterfaceRemoveLabel(
    input: {
      labelNames: $labelNames,
      interfaceIDs: $interfaceIDs
      cursor: "Mg=="
    }
  ) {
    success
    error {
      message
    }
    cursor
  }
}
```

Email examples

The following requests require Email 1.2.91 or later to retrieve email server profiles, or submit requests to send email.



Note: You must configure at least one email server profile in the Tanium Connect settings. Configuring an email server profile requires Connect 5.16.178 or later.

(1.0 - Experimental (Early Development)) Get email server summary by ID

(query.emailServerSummary)

About this task

The `emailServerSummary` query requires Email 1.2.91 or later and the **Email** send permission.

©2026 Tanium Inc. All Rights Reserved

Note: You must configure at least one email server profile in the Tanium Connect settings. Configuring an email server profile requires Connect 5.16.178 or later and the **Email Config** write permission.

Use the `emailServerSummary` query to retrieve the summary information for an email server that matches the ID. Note the following:

- Email server summary information includes information needed to submit a request to send an email using `mutation.emailSend`, including the email server ID.
- You can retrieve an email server only if the email server is enabled to send emails, and you have the **Email** send permission for the content set containing the email server. If either is not true, the response does not return the email server summary information for a given email server.

The following query retrieves the email server that matches the ID, returning summary information only if the email server is enabled to send emails, and the requesting user has the **Email** send permission for the content set containing the email server.

```
query exampleGetEmailServerByID($input: IdRefInput!) {
  emailServerSummary(ref: $input) {
    id
    description
    fromAddress
    name
  }
}
```

Include an input variable in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "12345678-90ab-cdef-1234-567890abcdef"
  }
}
```

Example response:

```
{
  "data": {
    "id": "12345678-90ab-cdef-1234-567890abcdef",
    "description": "Example email server description",
    "fromAddress": "from_user@example.com",
    "name": "example_email_server"
  }
}
```

©2026 Tanium Inc. All Rights Reserved

(1.0 - Experimental (Early Development)) Get multiple email server summaries

(query.emailServerSummaries)

About this task

The `emailServerSummaries` query requires Email 1.2.91 or later and the **Email** send permission.

> **Note:** You must configure at least one email server profile in the Tanium Connect settings. Configuring an email server profile requires Connect 5.16.178 or later and the **Email Config** write permission.

Use the `emailServerSummaries` query to retrieve the summary information for multiple email servers. Note the following:

- Email server summary information includes information needed to submit a request to send an email using `mutation.emailSend`, including the email server ID.
- You can retrieve an email server only if the email server is enabled to send emails, and you have the **Email** send permission for the content set containing the email server. If either is not true, the response does not return the email server summary information for a given email server.

The following query retrieves the first 2 email servers that are enabled to send emails for which the requesting user also has the **Email** send permission for the content set containing the email server.

```
query exampleGetEmailServers($first: Int) {
  emailServerSummaries(first: $first) {
    edges {
      node {
        id
        description
        fromAddress
        name
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:

```
{
  "data": {
    "emailServerSummaries": {
      "edges": [
        {
          "node": {
            "id": "12345678-90ab-cdef-1234-567890abcdef",
            "description": "Example email server description",
            "fromAddress": "from_user@example.com",
            "name": "example_email_server"
          }
        },
        {
          "node": {
            "id": "fedcba09-8765-4321-fedc-ba0987654321",
            "description": "Example email server description 2",
            "fromAddress": "from_user_2@example.com",
            "name": "example_email_server_2"
          }
        }
      ]
    }
  }
}
```

(1.0 - Experimental (Early Development)) Submit request to send email

(mutation.emailSend)

About this task

The `emailSend` mutation requires Email 1.2.91 or later and the **Email** send permission. The ability to add attachments requires Email 1.16.17 or later.

Note: You must configure at least one email server profile in the Tanium Connect settings to submit a request to send email. Configuring an email server profile requires Connect 5.16.178 or later and the **Email Config** write permission.

Use the `emailSend` mutation to submit a request to Tanium Email to send an email. Note the following:

- Configure an email server in Connect. For more information, see Tanium Connect User Guide: Configure email server profile settings for Microsoft 365.
- Use an email server that is enabled to send emails, and to which you have the **Email** send

©2026 Tanium Inc. All Rights Reserved

permission for the content set containing the email server. Use the `emailServerSummaries` query to retrieve all email servers that meet these requirements, or the `emailServerSummary` query to verify whether the email server matching the ID meets these requirements.

- You must define a 36 character hexadecimal UUID to pass as an input ID, such as `12345678-90ab-cdef-1234-567890abcdef`. This mutation is idempotent; if you submit several requests with the same UUID, only the first successful request results in a sent email, and other requests have no effect.
- If the response is successful, the `queuedTime` value is the time that Tanium Email receives the request to send the email. The actual send time might be up to several minutes after the `queuedTime` value.
- (Optional) To include file attachments, pass an `attachments` array in the input. Each attachment object requires `name` (the file name), `contentType` (the MIME type), `data` (the file content), and `encoding`. Set `encoding` to `RAW` for plain-text content or `BASE64` for Base64-encoded binary content.
- When you include attachments, note the following validation requirements:
  - The `name`, `contentType`, and `data` fields must not be empty.
  - The `data` field must be valid Base64 when `encoding` is BASE64.
  - The combined size of all attachments must not exceed 15 MB.

The following mutation submits a request to Tanium Email to send an email.

```
mutation exampleSubmitEmailSendRequest($input: EmailSendInput!) {
  emailSend(input: $input) {
    queuedTime
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variable in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "attachments": [
```

©2026 Tanium Inc. All Rights Reserved

```
      {
        "contentType": "text/plain",
        "data": "Example plain-text attachment content.",
        "encoding": "RAW",
        "name": "notes.txt"
      },
      {
        "contentType": "application/pdf",
        "data":
"JVBERi0xLjQKJcOkw7zDtsOfCjIgMCBvYmoKPDwvTGVuZ3RoIDMgMCBSL0ZpbHRlci9GbGF0ZURlY29kZT4+",
        "encoding": "BASE64",
        "name": "report.pdf"
      }
    ],
    "bccAddresses": "bcc_user@example.com",
    "body": "Example email body: Email sent.",
    "bodyType": "TEXT",
    "ccAddresses": "cc_user@example.com",
    "emailServer": {
      "id": "12345678-90ab-cdef-1234-567890abcdef"
    },
    "id": "fedcba09-8765-4321-fedc-ba0987654321",
    "subject": "Example email subject",
    "toAddresses": "to_user@example.com"
  }
}
```

Example response:

```
{
  "data": {
    "queuedTime": "2023-01-01T00:00:00.00000000Z",
    "error": null
  }
}
```

Integrity Monitor examples

The following requests require Integrity Monitor 3.6 or later to create, retrieve, update, deploy, and delete Integrity Monitor monitors and watchlists.

Create Integrity Monitor monitor

mutation.integrityMonitorMonitorCreate

About this task

The `integrityMonitorMonitorCreate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Monitors write** permission.

The following mutation creates a monitor.

```
mutation exampleCreateIMMonitor($input: IntegrityMonitorMonitorCreateInput!) {
  integrityMonitorMonitorCreate(input: $input) {
    monitor {
      id
      collectProcessInformation
      computerGroups {
        edges {
          node {
            id
            contentSetV2 {
              id
              name
              description
            }
            expression
            filterEnabled
            managementRightsEnabled
            modifiedTime
            name
            type
          }
        }
      }
      databaseMaximumSizeMegabytes
      deploymentStatus
      description
      indexFirstScanDistributeOverTime {
        nanoseconds
        seconds
      }
      indexScanFrequency {
        nanoseconds
        seconds
      }
      monitorPruningAge {
        nanoseconds
        seconds
      }
      name
      priority
      revision
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "collectProcessInformation": true,
    "computerGroups": {
      "id": "123"
    },
    "databaseMaximumSizeMegabytes": 512,
    "description": "example monitor",
    "indexFirstScanDistributeOverTime": {
      "nanoseconds": 0,
```

©2026 Tanium Inc. All Rights Reserved

```
      "seconds": 14400
    },
    "indexScanFrequency": {
      "nanoseconds": 0,
      "seconds": 43200
    },
    "monitorPruningAge": {
      "nanoseconds": 0,
      "seconds": 86400
    },
    "name": "Example monitor",
    "priority": 1
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorMonitorCreate": {
      "monitor": {
        "id": "123",
        "collectProcessInformation": true,
        "computerGroups": {
          "edges": [
            {
              "node": {
                "id": "123",
                "contentSetV2": {
                  "id": "3",
                  "name": "Default Filter Groups",
                  "description": null
                },
                "expression": " Operating System Generation equals Windows 10",
                "filterEnabled": true,
                "managementRightsEnabled": true,
                "modifiedTime": null,
                "name": "All Windows 10",
                "type": "STANDARD"
              }
            }
          ]
        },
        "databaseMaximumSizeMegabytes": 512,
        "deploymentStatus": "PendingDeployment",
        "description": "example monitor",
        "indexFirstScanDistributeOverTime": {
          "nanoseconds": 0,
          "seconds": 14400
        },
        "indexScanFrequency": {
          "nanoseconds": 0,
          "seconds": 43200
        },
        "monitorPruningAge": {
          "nanoseconds": 0,
          "seconds": 86400
        },
        "name": "Example monitor",
        "priority": 1,
        "revision": 1
      },
```

©2026 Tanium Inc. All Rights Reserved

```
      "error": null
    }
  }
}
```

Create Integrity Monitor watchlist

mutation.integrityMonitorWatchlistCreate

About this task

The `integrityMonitorWatchlistCreate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation creates a Windows-style watchlist. Associated watchlist paths use backslashes (`\`).

```
mutation exampleCreateIMWatchlist($input: IntegrityMonitorWatchlistCreateInput!) {
  integrityMonitorWatchlistCreate(input: $input) {
    watchlist {
      id
      name
      computerGroups {
        edges {
          node {
            id
            name
            managementRightsEnabled
            filterEnabled
            contentSetV2 {
              id
              name
              description
            }
            type
            expression
          }
        }
      }
      deploymentStatus
      description
      paths {
        edges {
          node {
            path
            watchlistPathType
            includeFilePermissionChanges
            inclusions {
              edges {
                cursor
                node
              }
            }
            exclusions {
              edges {
                cursor
                node
```

```
            }
          }
        }
      }
    }
    pathStyle
    revision
  }
  error {
    message
    retryable
    timedOut
  }
 }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "computerGroups": [
      {
        "name": "All Computers"
      },
      {
        "id": "123"
      }
    ],
    "description": "example watchlist",
    "name": "Example watchlist",
    "pathStyle": "Windows"
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistCreate": {
      "watchlist": {
        "id": "123",
        "name": "Example watchlist",
        "computerGroups": {
          "edges": [
            {
              "node": {
                "id": "345",
                "name": "All Computers",
                "managementRightsEnabled": true,
                "filterEnabled": true,
                "contentSetV2": {
                  "id": "1",
                  "name": "Reserved",
                  "description": "Built in sensors which can not be edited or removed."
                },
                "type": "STANDARD",
                "expression": " any Computer Name not matches .*"
              }
            }
          ]
        },
```

©2026 Tanium Inc. All Rights Reserved

```
        "deploymentStatus": "PendingDeployment",
        "description": "example watchlist",
        "paths": {
          "edges": []
        },
        "pathStyle": "Windows",
        "revision": 1
      },
      "error": null
    }
  }
}
```

The following mutation creates a Unix-style watchlist. Associated watchlist paths use forward slashes (`/`).

```
mutation exampleCreateIMWatchlist($input: IntegrityMonitorWatchlistCreateInput!) {
  integrityMonitorWatchlistCreate(input: $input) {
    watchlist {
      id
      name
      computerGroups {
        edges {
          node {
            id
            name
            managementRightsEnabled
            filterEnabled
            contentSetV2 {
              id
              name
              description
            }
            type
            expression
          }
        }
      }
      deploymentStatus
      description
      paths {
        edges {
          node {
            path
            watchlistPathType
            includeFilePermissionChanges
            inclusions {
              edges {
                node
              }
            }
            exclusions {
              edges {
                node
              }
            }
          }
        }
      }
      pathStyle
      revision
    }
```

```
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "computerGroups": [
      {
        "name": "All Computers"
      },
      {
        "id": "123"
      }
    ],
    "description": "example watchlist",
    "name": "Example watchlist Unix",
    "pathStyle": "Unix"
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistCreate": {
      "watchlist": {
        "id": "234",
        "name": "Example watchlist Unix",
        "computerGroups": {
          "edges": [
            {
              "node": {
                "id": "345",
                "name": "All Computers",
                "managementRightsEnabled": true,
                "filterEnabled": true,
                "contentSetV2": {
                  "id": "1",
                  "name": "Reserved",
                  "description": "Built in sensors which can not be edited or removed."
                },
                "type": "STANDARD",
                "expression": " any Computer Name not matches .*"
              }
            }
          ]
        },
        "deploymentStatus": "PendingDeployment",
        "description": "example watchlist",
        "paths": {
          "edges": []
        },
        "pathStyle": "Unix",
        "revision": 1
      },
```

©2026 Tanium Inc. All Rights Reserved

```
      "error": null
    }
  }
}
```

Create Integrity Monitor watchlist Unix path

mutation.integrityMonitorWatchlistUnixPathCreate

About this task

The `integrityMonitorWatchlistUnixPathCreate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation creates a Unix-style watchlist path that uses forward slashes (`/`). This path is associated with the watchlist that matches the ID `123`.

> **Note:** The response returns an error if no watchlist matches the input ID, if the watchlist does not use Unix-style paths, or if the path is not unique in the watchlist.

```
mutation exampleCreateIMUnixPath($input: IntegrityMonitorWatchlistUnixPathCreateInput!) {
  integrityMonitorWatchlistUnixPathCreate(input: $input) {
    watchlistPath {
      id
      exclusions {
        edges {
          node
        }
      }
      inclusions {
        edges {
          node
        }
      }
      path
      watchlistPathType
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "watchlistID": "123",
    "path": "/a/b*/c",
```

©2026 Tanium Inc. All Rights Reserved

```
    "inclusions": [
      ".ssh/\\*\\*"
    ],
    "exclusions": [
      "\\*\\*.log",
      "\\*\\*.db"
    ]
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistUnixPathCreate": {
      "watchlistPath": {
        "id": "123",
        "exclusions": {
          "edges": [
            {
              "node": "\\*\\*.db"
            },
            {
              "node": "\\*\\*.log"
            }
          ]
        },
        "inclusions": {
          "edges": [
            {
              "node": ".ssh/\\*\\*"
            }
          ]
        },
        "path": "/a/b*/c",
        "watchlistPathType": "File"
      },
      "error": null
    }
  }
}
```

Create Integrity Monitor watchlist Windows file path

mutation.integrityMonitorWatchlistWindowsFilePathCreate

About this task

The `integrityMonitorWatchlistWindowsFilePathCreate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation creates a Windows-style watchlist file path that uses backslashes (`\`). This path is associated with the watchlist that matches the ID `123`.

©2026 Tanium Inc. All Rights Reserved

Note: The response returns an error if no watchlist matches the input ID, if the watchlist does not use Windows-style paths, or if the path is not unique in the watchlist.

```
mutation exampleCreateIMWinFilePath($input:
IntegrityMonitorWatchlistWindowsFilePathCreateInput!) {
  integrityMonitorWatchlistWindowsFilePathCreate(input: $input) {
    watchlistPath {
      id
      exclusions {
        edges {
          node
        }
      }
      includeFilePermissionChanges
      inclusions {
        edges {
          node
        }
      }
      path
      watchlistPathType
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "watchlistID": "123",
    "path": "C:\\Program Files",
    "inclusions": [
      "**.pem"
    ],
    "exclusions": [
      "**.log",
      "**.db"
    ],
    "includeFilePermissionChanges": true
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistWindowsFilePathCreate": {
      "watchlistPath": {
        "id": "2345",
        "exclusions": {
          "edges": [
            {
              "node": "**.db"
```

©2026 Tanium Inc. All Rights Reserved

```
            },
            {
              "node": "**.log"
            }
          ]
        },
        "includeFilePermissionChanges": true,
        "inclusions": {
          "edges": [
            {
              "node": "**.pem"
            }
          ]
        },
        "path": "C:\\Program Files",
        "watchlistPathType": "File"
      },
      "error": null
    }
  }
}
```

Create Integrity Monitor watchlist Windows registry path

mutation.integrityMonitorWatchlistWindowsRegistryPathCreate

About this task

The `integrityMonitorWatchlistWindowsRegistryPathCreate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation creates a Windows-style watchlist registry path that uses backslashes (`\`). This path is associated with the watchlist that matches the ID `123`.

Note: The response returns an error if no watchlist matches the input ID, if the watchlist does not use Windows-style paths, or if the path is not unique in the watchlist.

```
mutation exampleCreateIMWinRegPath($input:
IntegrityMonitorWatchlistWindowsRegistryPathCreateInput!) {
  integrityMonitorWatchlistWindowsRegistryPathCreate(input: $input) {
    watchlistPath {
      id
      exclusions {
        edges {
          node
        }
      }
      inclusions {
        edges {
          node
        }
      }
```

©2026 Tanium Inc. All Rights Reserved

```
      path
      watchlistPathType
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "watchlistID": "123",
    "path": "HKEY_LOCAL_MACHINE\\a\\b*\\c",
    "inclusions": [
      "Classes\\*\\ShellEx\\*\\*"
    ],
    "exclusions": [
      ".NETFramework\\*\\NGenService\\**",
      "Cryptography\\CertificateTemplateCache\\*"
    ]
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistWindowsRegistryPathCreate": {
      "watchlistPath": {
        "id": "12345",
        "exclusions": {
          "edges": [
            {
              "node": "Cryptography\\CertificateTemplateCache\\*"
            },
            {
              "node": ".NETFramework\\*\\NGenService\\**"
            }
          ]
        },
        "inclusions": {
          "edges": [
            {
              "node": "Classes\\*\\ShellEx\\*\\*"
            }
          ]
        },
        "path": "HKEY_LOCAL_MACHINE\\a\\b*\\c",
        "watchlistPathType": "Registry"
      },
      "error": null
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Delete Integrity Monitor watchlist path

mutation.integrityMonitorWatchlistPathDelete

About this task

The `integrityMonitorWatchlistPathDelete` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation deletes a watchlist path that matches the ID `123`.

```
mutation exampleDeleteIMPath($ref: IdRefInput!) {
  integrityMonitorWatchlistPathDelete(input: $ref) {
    pathID
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "ref": {
    "id": "123"
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistPathDelete": {
      "pathID": "123",
      "error": null
    }
  }
}
```

Deploy Integrity Monitor monitors

mutation.integrityMonitorStartMonitorDeployment

About this task

The `integrityMonitorStartMonitorDeployment` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Deploy Monitors** permission.

©2026 Tanium Inc. All Rights Reserved

The following mutation deploys all monitors. Monitors marked for deletion are deleted during this deployment.

```
mutation exampleStartMonitorDeployment {
  integrityMonitorStartMonitorDeployment {
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorStartMonitorDeployment": {
      "error": null
    }
  }
}
```

Deploy Integrity Monitor watchlists

mutation.integrityMonitorStartWatchlistDeployment

About this task

The `integrityMonitorStartWatchlistDeployment` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Deploy Watchlists** permission.

The following mutation deploys all watchlists. Watchlists marked for deletion are deleted during this deployment.

```
mutation exampleStartWatchlistDeployment {
  integrityMonitorStartWatchlistDeployment {
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorStartWatchlistDeployment": {
      "error": null
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Get Integrity Monitor monitor by ID

query.integrityMonitorMonitor

About this task

The `integrityMonitorMonitor` query requires Integrity Monitor 3.6 or later and the **Integrity Monitor Monitor read** permission.

The following query retrieves the monitor that matches the ID.

```
query exampleGetIMMonitorByID($id: ID!) {
  integrityMonitorMonitor(ref: {id: $id}) {
    id
    collectProcessInformation
    computerGroups {
      edges {
        node {
          id
          contentSetV2 {
            id
            name
            description
          }
          expression
          filterEnabled
          managementRightsEnabled
          modifiedTime
          name
          type
        }
      }
    }
    databaseMaximumSizeMegabytes
    deploymentStatus
    description
    indexFirstScanDistributeOverTime {
      seconds
      nanoseconds
    }
    indexScanFrequency {
      seconds
      nanoseconds
    }
    monitorPruningAge {
      seconds
      nanoseconds
    }
    name
    priority
    revision
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "id": "123"
}
```

Example response:

```
{
  "data": {
    "integrityMonitorMonitor": {
      "id": "123",
      "collectProcessInformation": true,
      "computerGroups": {
        "edges": [
          {
            "node": {
              "id": "75",
              "contentSetV2": {
                "id": "3",
                "name": "Default Filter Groups",
                "description": null
              },
              "expression": " Is Linux equals True",
              "filterEnabled": true,
              "managementRightsEnabled": true,
              "modifiedTime": null,
              "name": "All Linux",
              "type": "STANDARD"
            }
          }
        ]
      },
      "databaseMaximumSizeMegabytes": null,
      "deploymentStatus": "PendingDeployment",
      "description": "",
      "indexFirstScanDistributeOverTime": {
        "seconds": 14400,
        "nanoseconds": 0
      },
      "indexScanFrequency": {
        "seconds": 43200,
        "nanoseconds": 0
      },
      "monitorPruningAge": {
        "seconds": 2592000,
        "nanoseconds": 0
      },
      "name": "Linux Monitor",
      "priority": 3,
      "revision": 5
    }
  }
}
```

Get Integrity Monitor service status

query.integrityMonitorStatus

About this task

The `integrityMonitorStatus` query requires Integrity Monitor 3.6 or later and the following permissions:

- **Integrity Monitor Monitors read** and **Integrity Monitor Watchlists read** permissions to submit the query request
- **Integrity Monitor Deploy Monitors** permission to get monitor deployment status (`isDeployingMonitors`)
- **Integrity Monitor Deploy Watchlists** permission to get watchlist deployment status(`isDeployingWatchlists`)

The following query retrieves the Integrity Monitor service status.

```
query exampleGetIMServiceStatus {
  integrityMonitorStatus {
    isDeployingMonitors
    isDeployingWatchlists
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorStatus": {
      "isDeployingMonitors": true,
      "isDeployingWatchlists": true
    }
  }
}
```

Get Integrity Monitor watchlist by ID

query.integrityMonitorWatchlist

About this task

The `integrityMonitorWatchlist` query requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists read** permission.

The following query retrieves the watchlist that matches the ID.

```
query exampleGetIMWatchlistByID($id: ID!) {
  integrityMonitorWatchlist(ref: {id: $id}) {
    id
    name
    computerGroups {
```

©2026 Tanium Inc. All Rights Reserved

```
      edges {
        node {
          id
          name
          managementRightsEnabled
          filterEnabled
          contentSetV2 {
            id
            name
            description
          }
          type
          expression
        }
      }
    }
    deploymentStatus
    description
    paths {
      edges {
        node {
          path
          watchlistPathType
          includeFilePermissionChanges
          inclusions {
            edges {
              node
            }
          }
          exclusions {
            edges {
              node
            }
          }
        }
      }
    }
    pathStyle
    revision
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "123"
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlist": {
      "id": "123",
      "name": "Example watchlist update",
      "computerGroups": {
        "edges": [
          {
            "node": {
              "id": "1",
              "name": "All Computers",
```

©2026 Tanium Inc. All Rights Reserved

```
              "managementRightsEnabled": true,
              "filterEnabled": true,
              "contentSetV2": {
                "id": "1",
                "name": "Reserved",
                "description": "Built in sensors which can not be edited or removed."
              },
              "type": "STANDARD",
              "expression": " All Computers"
            }
          },
          {
            "node": {
              "id": "123",
              "name": "All Debian 9",
              "managementRightsEnabled": true,
              "filterEnabled": true,
              "contentSetV2": {
                "id": "3",
                "name": "Default Filter Groups",
                "description": null
              },
              "type": "STANDARD",
              "expression": " Operating System Generation equals Debian 9"
            }
          }
        ]
      },
      "deploymentStatus": "PendingDeployment",
      "description": "example watchlist update",
      "paths": {
        "edges": [
          {
            "node": {
              "path": "/a/b*/c",
              "watchlistPathType": "File",
              "includeFilePermissionChanges": true,
              "inclusions": {
                "edges": [
                  {
                    "node": ".ssh/\\*\\*"
                  }
                ]
              },
              "exclusions": {
                "edges": []
              }
            }
          }
        ]
      },
      "pathStyle": "Unix",
      "revision": 3
    }
  }
}
```


Get multiple Integrity Monitor monitors

query.integrityMonitorMonitors

©2026 Tanium Inc. All Rights Reserved

About this task

The `integrityMonitorMonitors` query requires Integrity Monitor 3.6 or later and the **Integrity Monitor Monitor read** permission.

The following query retrieves the first `2` monitors.

```
query exampleGetMultipleIMMonitors($first: Int) {
  integrityMonitorMonitors(first: $first) {
    edges {
      node {
        id
        collectProcessInformation
        computerGroups {
          edges {
            node {
              id
              contentSetV2 {
                id
                name
                description
              }
              expression
              filterEnabled
              managementRightsEnabled
              modifiedTime
              name
              type
            }
          }
        }
        databaseMaximumSizeMegabytes
        deploymentStatus
        description
        indexFirstScanDistributeOverTime {
          seconds
          nanoseconds
        }
        indexScanFrequency {
          seconds
          nanoseconds
        }
        monitorPruningAge {
          seconds
          nanoseconds
        }
        name
        priority
        revision
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "integrityMonitorMonitors": {
      "edges": [
        {
          "node": {
            "id": "123",
            "collectProcessInformation": true,
            "computerGroups": {
              "edges": []
            },
            "databaseMaximumSizeMegabytes": null,
            "deploymentStatus": "PendingDeployment",
            "description": "",
            "indexFirstScanDistributeOverTime": {
              "seconds": 0,
              "nanoseconds": 0
            },
            "indexScanFrequency": {
              "seconds": 600,
              "nanoseconds": 0
            },
            "monitorPruningAge": {
              "seconds": 2592000,
              "nanoseconds": 0
            },
            "name": "example monitor",
            "priority": 1,
            "revision": 6
          }
        },
        {
          "node": {
            "id": "234",
            "collectProcessInformation": true,
            "computerGroups": {
              "edges": [
                {
                  "node": {
                    "id": "234",
                    "contentSetV2": {
                      "id": "3",
                      "name": "Default Filter Groups",
                      "description": null
                    },
                    "expression": " Operating System Generation equals Windows Server 2008
R2",
                    "filterEnabled": true,
                    "managementRightsEnabled": true,
                    "modifiedTime": null,
                    "name": "All Windows Server 2008 R2",
                    "type": "STANDARD"
                  }
                },
                {
                  "node": {
                    "id": "345",
                    "contentSetV2": {
                      "id": "3",
                      "name": "Default Filter Groups",
                      "description": null
```


©2026 Tanium Inc. All Rights Reserved

```
            },
            "expression": " Operating System Generation equals Windows Server 2016",
            "filterEnabled": true,
            "managementRightsEnabled": true,
            "modifiedTime": null,
            "name": "All Windows Server 2016",
            "type": "STANDARD"
          }
        },
        {
          "node": {
            "id": "456",
            "contentSetV2": {
              "id": "3",
              "name": "Default Filter Groups",
              "description": null
            },
            "expression": " Operating System Generation equals Windows Server 2019",
            "filterEnabled": true,
            "managementRightsEnabled": true,
            "modifiedTime": null,
            "name": "All Windows Server 2019",
            "type": "STANDARD"
          }
        },
        {
          "node": {
            "id": "567",
            "contentSetV2": {
              "id": "3",
              "name": "Default Filter Groups",
              "description": null
            },
            "expression": " Operating System Generation equals Windows Server 2022",
            "filterEnabled": true,
            "managementRightsEnabled": true,
            "modifiedTime": null,
            "name": "All Windows Server 2022",
            "type": "STANDARD"
          }
        },
        {
          "node": null
        }
      ]
    },
    "databaseMaximumSizeMegabytes": null,
    "deploymentStatus": "PendingDeployment",
    "description": "",
    "indexFirstScanDistributeOverTime": {
      "seconds": 14400,
      "nanoseconds": 0
    },
    "indexScanFrequency": {
      "seconds": 43200,
      "nanoseconds": 0
    },
    "monitorPruningAge": {
      "seconds": 2592000,
      "nanoseconds": 0
    },
    "name": "Windows Monitor",
    "priority": 2,
    "revision": 6
```


©2026 Tanium Inc. All Rights Reserved

```
          }
        }
      ]
    }
  }
}
```

Get multiple Integrity Monitor watchlists

query.integrityMonitorWatchlists

About this task

The `integrityMonitorWatchlists` query requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists read** permission.

The following query retrieves the first `2` watchlists.

```
query exampleGetIMWatchlists($first: Int) {
  integrityMonitorWatchlists(first: $first) {
    edges {
      node {
        id
        name
        computerGroups {
          edges {
            node {
              id
              name
              managementRightsEnabled
              filterEnabled
              contentSetV2 {
                id
                name
                description
              }
              type
              expression
            }
          }
        }
        deploymentStatus
        description
        paths {
          edges {
            node {
              path
              watchlistPathType
              includeFilePermissionChanges
              inclusions {
                edges {
                  node
                }
              }
              exclusions {
                edges {
                  node
```

```
                }
              }
            }
          }
        }
        pathStyle
        revision
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlists": {
      "edges": [
        {
          "node": {
            "id": "12",
            "name": "Example AIX Watchlist",
            "computerGroups": {
              "edges": [
                {
                  "node": {
                    "id": "33",
                    "name": "All AIX",
                    "managementRightsEnabled": true,
                    "filterEnabled": true,
                    "contentSetV2": {
                      "id": "3",
                      "name": "Default Filter Groups",
                      "description": null
                    },
                    "type": "STANDARD",
                    "expression": " Is AIX equals True"
                  }
                }
              ]
            },
            "deploymentStatus": "Deployed",
            "description": "",
            "paths": {
              "edges": [
                {
                  "node": {
                    "path": "/bin",
                    "watchlistPathType": "File",
                    "includeFilePermissionChanges": true,
                    "inclusions": {
                      "edges": []
                    },
                    "exclusions": {
                      "edges": [
```

```
                    {
                      "node": "**.db"
                    },
                    {
                      "node": "**.log"
                    },
                    {
                      "node": "**.state"
                    },
                    {
                      "node": "**.tmp"
                    }
                  ]
                }
              }
            },
            {
              "node": {
                "path": "/boot",
                "watchlistPathType": "File",
                "includeFilePermissionChanges": true,
                "inclusions": {
                  "edges": [
                    {
                      "node": "efi/**"
                    },
                    {
                      "node": "grub/**"
                    },
                    {
                      "node": "grub2/**"
                    }
                  ]
                },
                "exclusions": {
                  "edges": []
                }
              }
            },
            {
              "node": {
                "path": "/etc/shells",
                "watchlistPathType": "File",
                "includeFilePermissionChanges": true,
                "inclusions": {
                  "edges": []
                },
                "exclusions": {
                  "edges": []
                }
              }
            }
          ]
        },
        "pathStyle": "Unix",
        "revision": 2
      }
    },
    {
      "node": {
        "id": "23",
        "name": "Example watchlist",
        "computerGroups": {
          "edges": [
```


©2026 Tanium Inc. All Rights Reserved

```
                {
                  "node": {
                    "id": "1",
                    "name": "All Computers",
                    "managementRightsEnabled": true,
                    "filterEnabled": true,
                    "contentSetV2": {
                      "id": "1",
                      "name": "Reserved",
                      "description": "Built in sensors which can not be edited or removed."
                    },
                    "type": "STANDARD",
                    "expression": " All Computers"
                  }
                }
              ]
            },
            "deploymentStatus": "Deployed",
            "description": "",
            "paths": {
              "edges": [
                {
                  "node": {
                    "path": "C:\\example",
                    "watchlistPathType": "File",
                    "includeFilePermissionChanges": true,
                    "inclusions": {
                      "edges": []
                    },
                    "exclusions": {
                      "edges": []
                    }
                  }
                },
                {
                  "node": {
                    "path": "HKEY_LOCAL_MACHINE\\SYSTEM\\EXAMPLE",
                    "watchlistPathType": "Registry",
                    "includeFilePermissionChanges": false,
                    "inclusions": {
                      "edges": []
                    },
                    "exclusions": {
                      "edges": []
                    }
                  }
                }
              ]
            },
            "pathStyle": "Windows",
            "revision": 3
          }
        }
      ]
    }
  }
}
```


Mark Integrity Monitor monitors for deletion

mutation.integrityMonitorMonitorMarkForDeletion

©2026 Tanium Inc. All Rights Reserved

About this task

The `integrityMonitorMonitorMarkForDeletion` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Monitors write** permission.

The following mutation marks the monitor that matches the ID `123` for deletion. Monitors marked for deletion are deleted during the next monitor deployment.

```
mutation exampleMarkIMMonitorForDeletion($ref: IdRefInput!) {
  integrityMonitorMonitorMarkForDeletion(input: $ref) {
    monitor {
      id
      name
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "ref": {
    "id": "123"
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorMonitorMarkForDeletion": {
      "monitor": {
        "id": "123",
        "name": "example monitor"
      },
      "error": null
    }
  }
}
```

Mark Integrity Monitor watchlists for deletion

mutation.integrityMonitorWatchlistMarkForDeletion

About this task

The `integrityMonitorWatchlistMarkForDeletion` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

©2026 Tanium Inc. All Rights Reserved

The following mutation marks the watchlist that matches the ID `123` for deletion. Watchlists marked for deletion are deleted during the next watchlist deployment.

```
mutation exampleDeleteWatchlist($ref: IdRefInput!) {
  integrityMonitorWatchlistMarkForDeletion(input: $ref) {
    watchlist {
      id
      name
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "ref": {
    "id": "123"
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistMarkForDeletion": {
      "watchlist": {
        "id": "123",
        "name": "example watchlist"
      },
      "error": null
    }
  }
}
```

Update Integrity Monitor monitor by ID

mutation.integrityMonitorMonitorUpdate

About this task

The `integrityMonitorMonitorUpdate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Monitors write** permission.

The following mutation updates a monitor that matches the ID. Note the following:

- The response returns an error if no monitor matches the ID, the `name` value is an empty string or `null`, or the `name` value is updated to a name that matches another monitor.

©2026 Tanium Inc. All Rights Reserved

- Fields defined with a value overwrite the existing field value with the defined value. Defining an array with at least one value for the `computerGroups` field overwrites all existing array values with the defined array values.
- Fields defined with a `null` value are not updated, if `null` is allowed.
- Fields defined with an empty string or empty array replace any existing value with the empty string or empty array, if an empty string or empty array are allowed.

```
mutation exampleUpdateIMMonitor($input: IntegrityMonitorMonitorUpdateInput!) {
  integrityMonitorMonitorUpdate(input: $input) {
    monitor {
      id
      collectProcessInformation
      computerGroups {
        edges {
          node {
            id
            contentSetV2 {
              id
              name
              description
            }
            expression
            filterEnabled
            managementRightsEnabled
            modifiedTime
            name
            type
          }
        }
      }
      databaseMaximumSizeMegabytes
      deploymentStatus
      description
      indexFirstScanDistributeOverTime {
        nanoseconds
        seconds
      }
      indexScanFrequency {
        nanoseconds
        seconds
      }
      monitorPruningAge {
        nanoseconds
        seconds
      }
      name
      priority
      revision
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "12",
    "collectProcessInformation": true,
    "computerGroups": {
      "id": "123"
    },
    "databaseMaximumSizeMegabytes": 1024,
    "description": "example monitor",
    "indexFirstScanDistributeOverTime": {
      "seconds": 14400,
      "nanoseconds": 0
    },
    "indexScanFrequency": {
      "seconds": 43200,
      "nanoseconds": 0
    },
    "monitorPruningAge": {
      "seconds": 86400,
      "nanoseconds": 0
    },
    "name": "Example monitor",
    "priority": 1
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorMonitorUpdate": {
      "monitor": {
        "id": "12",
        "collectProcessInformation": true,
        "computerGroups": {
          "edges": [
            {
              "node": {
                "id": "123",
                "contentSetV2": {
                  "id": "3",
                  "name": "Default Filter Groups",
                  "description": null
                },
                "expression": " Operating System Generation equals Windows 10",
                "filterEnabled": true,
                "managementRightsEnabled": true,
                "modifiedTime": null,
                "name": "All Windows 10",
                "type": "STANDARD"
              }
            }
          ]
        },
        "databaseMaximumSizeMegabytes": 1024,
        "deploymentStatus": "PendingDeployment",
        "description": "example monitor",
        "indexFirstScanDistributeOverTime": {
          "nanoseconds": 0,
```

©2026 Tanium Inc. All Rights Reserved

```
          "seconds": 14400
        },
        "indexScanFrequency": {
          "nanoseconds": 0,
          "seconds": 43200
        },
        "monitorPruningAge": {
          "nanoseconds": 0,
          "seconds": 86400
        },
        "name": "Example monitor",
        "priority": 1,
        "revision": 2
      },
      "error": null
    }
  }
}
```

Update Integrity Monitor watchlist

mutation.integrityMonitorWatchlistUpdate

About this task

The `integrityMonitorWatchlistUpdate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation updates a watchlist that matches the ID. Note the following:

- The response returns an error if no watchlist matches the ID, the `name` value is an empty string or `null`, or the `name` value is updated to a name that matches another watchlist with the same `pathStyle` value.
- Fields defined with a value overwrite the existing field value with the defined value. Defining an array with at least one value for the `computerGroups` field overwrites all existing array values with the defined array values.
- Fields defined with a `null` value are not updated, if `null` is allowed.
- Fields defined with an empty string or empty array replace any existing value with the empty string or empty array, if an empty string or empty array are allowed.
- If you change the `pathStyle` of a watchlist, all of the watchlist paths are deleted because paths must match the watchlist path style.

```
mutation exampleUpdateIMWatchlist($input: IntegrityMonitorWatchlistUpdateInput!) {
  integrityMonitorWatchlistUpdate(input: $input) {
```

©2026 Tanium Inc. All Rights Reserved

```
    watchlist {
      id
      name
      computerGroups {
        edges {
          node {
            id
            name
            managementRightsEnabled
            filterEnabled
            contentSetV2 {
              id
              name
              description
            }
            type
            expression
          }
        }
      }
      deploymentStatus
      description
      paths {
        edges {
          node {
            path
            watchlistPathType
            includeFilePermissionChanges
            inclusions {
              edges {
                node
              }
            }
            exclusions {
              edges {
                node
              }
            }
          }
        }
      }
      pathStyle
      revision
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "123",
    "computerGroups": [
      {
        "name": "All Computers"
      },
      {
```

©2026 Tanium Inc. All Rights Reserved

```
        "id": "123"
      }
    ],
    "description": "example watchlist update",
    "name": "Example watchlist update",
    "pathStyle": "Windows"
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistUpdate": {
      "watchlist": {
        "id": "123",
        "name": "Example watchlist update",
        "computerGroups": {
          "edges": [
            {
              "node": {
                "id": "1",
                "name": "All Computers",
                "managementRightsEnabled": true,
                "filterEnabled": true,
                "contentSetV2": {
                  "id": "1",
                  "name": "Reserved",
                  "description": "Built in sensors which can not be edited or removed."
                },
                "type": "STANDARD",
                "expression": " All Computers"
              }
            },
            {
              "node": {
                "id": "123",
                "name": "All Windows 8.1",
                "managementRightsEnabled": true,
                "filterEnabled": true,
                "contentSetV2": {
                  "id": "3",
                  "name": "Default Filter Groups",
                  "description": null
                },
                "type": "STANDARD",
                "expression": " Operating System Generation equals Windows 8.1"
              }
            }
          ]
        },
        "deploymentStatus": "PendingDeployment",
        "description": "example watchlist update",
        "paths": {
          "edges": []
        },
        "pathStyle": "Windows",
        "revision": 5
      },
      "error": null
    }
  }
```

```
}
```

Update Integrity Monitor watchlist Unix path

mutation.integrityMonitorWatchlistUnixPathUpdate

About this task

The `integrityMonitorWatchlistUnixPathUpdate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation updates a Unix-style watchlist path that uses forward slashes (`/`). This path matches the ID `1234`.



Note: The response returns an error if no Unix-style watchlist path matches the input ID.

```
mutation exampleUpdateIMUnixPath($input: IntegrityMonitorWatchlistUnixPathUpdateInput!) {
  integrityMonitorWatchlistUnixPathUpdate(input: $input) {
    watchlistPath {
      id
      exclusions {
        edges {
          node
        }
      }
      inclusions {
        edges {
          node
        }
      }
      path
      watchlistPathType
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "1234",
    "path": "/a/b*/c",
    "inclusions": [
      ".ssh/\\*\\*"
    ],
    "exclusions": [
```

©2026 Tanium Inc. All Rights Reserved

```
      "\\*\\*.log"
    ]
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistUnixPathUpdate": {
      "watchlistPath": {
        "id": "1234",
        "exclusions": {
          "edges": [
            {
              "node": "\\*\\*.log"
            }
          ]
        },
        "inclusions": {
          "edges": [
            {
              "node": ".ssh/\\*\\*"
            }
          ]
        },
        "path": "/a/b*/c",
        "watchlistPathType": "File"
      },
      "error": null
    }
  }
}
```

Update Integrity Monitor watchlist Windows file path

mutation.integrityMonitorWatchlistWindowsFilePathUpdate

About this task

The `integrityMonitorWatchlistWindowsFilePathUpdate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation updates a Windows-style watchlist file path that uses backslashes (`\`). This path matches the ID `2345`.



Note: The response returns an error if no Windows-style watchlist file path matches the input ID.

```
mutation exampleUpdateIMWinFilePath($input:
IntegrityMonitorWatchlistWindowsFilePathUpdateInput!) {
```

©2026 Tanium Inc. All Rights Reserved

```
  integrityMonitorWatchlistWindowsFilePathUpdate(input: $input) {
    watchlistPath {
      id
      exclusions {
        edges {
          node
        }
      }
      includeFilePermissionChanges
      inclusions {
        edges {
          node
        }
      }
      path
      watchlistPathType
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "2345",
    "path": "C:\\Program Files",
    "inclusions": [
      "**.pem",
      "**.log"
    ],
    "exclusions": [
      "**.db"
    ],
    "includeFilePermissionChanges": true
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistWindowsFilePathUpdate": {
      "watchlistPath": {
        "id": "2345",
        "exclusions": {
          "edges": [
            {
              "node": "**.db"
            }
          ]
        },
        "includeFilePermissionChanges": true,
        "inclusions": {
          "edges": [
            {
              "node": "**.log"
```

©2026 Tanium Inc. All Rights Reserved

```
            },
            {
              "node": "**.pem"
            }
          ]
        },
        "path": "C:\\Program Files",
        "watchlistPathType": "File"
      },
      "error": null
    }
  }
}
```

Update Integrity Monitor watchlist Windows registry path

mutation.integrityMonitorWatchlistWindowsRegistryPathUpdate

About this task

The `integrityMonitorWatchlistWindowsRegistryPathUpdate` mutation requires Integrity Monitor 3.6 or later and the **Integrity Monitor Watchlists write** permission.

The following mutation updates a Windows-style watchlist registry path that uses backslashes (`\`). This path matches the ID `12345`.

Note: The response returns an error if no Windows-style watchlist registry path matches the input ID.

```
mutation exampleUpdateIMWinRegPath($input:
IntegrityMonitorWatchlistWindowsRegistryPathUpdateInput!) {
  integrityMonitorWatchlistWindowsRegistryPathUpdate(input: $input) {
    watchlistPath {
      id
      exclusions {
        edges {
          node
        }
      }
      inclusions {
        edges {
          node
        }
      }
      path
      watchlistPathType
    }
    error {
      message
      retryable
      timedOut
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "12345",
    "path": "HKEY_LOCAL_MACHINE\\a\\b*\\c",
    "inclusions": [
      "Classes\\*\\ShellEx\\*\\*",
      ".NETFramework\\*\\NGenService\\**"
    ],
    "exclusions": [
      "Cryptography\\CertificateTemplateCache\\*"
    ]
  }
}
```

Example response:

```
{
  "data": {
    "integrityMonitorWatchlistWindowsRegistryPathUpdate": {
      "watchlistPath": {
        "id": "12345",
        "exclusions": {
          "edges": [
            {
              "node": "Cryptography\\CertificateTemplateCache\\*"
            }
          ]
        },
        "inclusions": {
          "edges": [
            {
              "node": "Classes\\*\\ShellEx\\*\\*"
            },
            {
              "node": ".NETFramework\\*\\NGenService\\**"
            }
          ]
        },
        "path": "HKEY_LOCAL_MACHINE\\a\\b*\\c",
        "watchlistPathType": "Registry"
      },
      "error": null
    }
  }
}
```

Jump Gate examples

The following request requires Jump Gate 1.5.12 or later to create Jump Gate requests.

©2026 Tanium Inc. All Rights Reserved

Create Jump Gate request

(mutation.jumpGateRequestCreate)

About this task

The `jumpGateCreate` mutation requires Jump Gate 1.5 or later and the **Jump Gate Requests write** permission.

The following mutation creates a Jump Gate request to access a Window endpoint.

```
mutation exampleJumpGateRequestCreate($input: JumpGateRequestInput!) {
  jumpGateRequestCreate(input: $input) {
    request {
      id
      createdBy {
        id
        username
        displayName
      }
      createdTime
      updatedBy {
        id
        username
        displayName
      }
      updatedTime
      closedBy {
        id
        username
        displayName
      }
      closedTime
      expirationTime
      policy {
        approvingUserIDs
        approvingUsers {
          id
          username
          displayName
        }
        approvingUserGroupIDs
      }
      justification
      targets {
        createdTime
        targetID
        targetDetails {
          name
          id
          serialNumber
          model
          isVirtual
          operatingSystemPlatform
          taniumClientIPAddress
          lastSeenTime
        }
      }
```

©2026 Tanium Inc. All Rights Reserved

```
      approvalStatus
      approvals {
        id
        updatedBy {
          id
          username
          displayName
        }
        updatedTime
        requestID
        action
        reason
      }
      notifications {
        id
        createdTime
        notificationType
      }
      settings {
        sessionIdleAfter {
          nanoseconds
          seconds
        }
        maxSessionDuration {
          nanoseconds
          seconds
        }
        maxCommandDuration {
          nanoseconds
          seconds
        }
      }
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "targets": [
      {
        "targetID": "123456",
        "accessType": "POWERSHELL"
      }
    ],
    "justification": "Debug the endpoint"
  }
}
```

Example response:

```
{
  "data": {
    "jumpGateRequestCreate": {
      "request": {
```

```
        "id": "019c4463-a011-7b4a-9789-f52cbf759b7f",
        "createdBy": {
          "id": "971",
          "username": "exampleusername",
          "displayName": null
        },
        "createdTime": "2026-02-09T21:51:45.68174Z",
        "updatedBy": {
          "id": "971",
          "username": "example.user@company.com",
          "displayName": null
        },
        "updatedTime": "2026-02-09T21:51:45.68174Z",
        "closedBy": null,
        "closedTime": null,
        "expirationTime": "2026-02-11T21:51:45.681726Z",
        "policy": null,
        "justification": "Debug endpoints",
        "targets": [
          {
            "createdTime": "2026-02-09T21:51:45.727457Z",
            "targetID": "123456",
            "targetDetails": {
              "name": "gateway-windows-endpoint",
              "id": "4477998899",
              "serialNumber": "VMware-42 1d f8 5e 99 b3 c6 64-ba 8a 6d 97 bc 45 42 85",
              "model": "VMware Virtual Platform",
              "isVirtual": null,
              "operatingSystemPlatform": "Windows",
              "taniumClientIPAddress": "10.10.99.99",
              "lastSeenTime": "2026-02-09T21:51:40Z"
            }
          }
        ],
        "approvalStatus": "PENDING",
        "approvals": [],
        "notifications": [],
        "settings": null
      },
      "error": null
    }
  }
}
```


Patch examples

The following requests require Patch 3.16.191 or later to create, retrieve, and stop Patch deployments.

Get patch applicability

(query.endpoints)

About this task

©2026 Tanium Inc. All Rights Reserved

The following query retrieves a list of patches that can be applied to endpoints in a specified computer group.

```
query exampleGetEndpointsPatch($cg: String!) {
  endpoints(filter: {memberOf: {name: $cg}}) {
    totalRecords
    edges {
      node {
        computerID
        name
        sensorReadings(sensors: [{name: "Patch - Patch List Applicability"}]) {
          columns {
            name
            values
          }
        }
      }
    }
  }
}
```

Include the computer group variable in the **Variables** panel or in your variables dictionary:

```
{
  "cg": "Corporate Systems"
}
```

Example response:

```
{
  "data": {
    "endpoints": {
      "totalRecords": 2,
      "edges": [
        {
          "node": {
            "computerID": "1847392056",
            "name": "ws-win10-prod-01",
            "sensorReadings": {
              "columns": [
                {
                  "name": "Patch List Ids",
                  "values": [
                    "1,2,3,37",
                    "1,2,3,31,8",
                    "1,2,3,35,37",
                    "1,2,3,31,90",
                    "1,3,31,42,",
                    "1,3,49,50,52,8"
                  ]
                },
                {
                  "name": "Title",
                  "values": [
                    "Windows Malicious Software Removal Tool x64 - v5.120 (KB890830)",
                    "2024-05 Cumulative Update for .NET Framework 3.5, 4.8 and 4.8.1 for
Windows 10 Version 22H2 for x64 (KB5038285)",
                    "2024-05 Cumulative Update for Windows 10 Version 22H2 for x64-based
Systems (KB5037768)",
                    "MSXML 6.0 RTM Security Update (KB925673)",
```

©2026 Tanium Inc. All Rights Reserved

```
                    "2024-04 Cumulative Update for Windows 10 Version 22H2 for x64-based
Systems (KB5036892)",
                    "2024-11 Security Update for Windows 10 Version 22H2 and Windows 10
Version 21H2 (KB5048239)"
                  ]
                },
                {
                  "name": "Install Status",
                  "values": [
                    "Not Installed",
                    "Not Installed",
                    "Not Installed",
                    "Not Installed",
                    "Installed",
                    "Not Installed"
                  ]
                },
                {
                  "name": "Uninstallable",
                  "values": [
                    "False",
                    "False",
                    "False",
                    "False",
                    "False",
                    "False"
                  ]
                },
                {
                  "name": "Severity",
                  "values": [
                    "None",
                    "Important",
                    "Critical",
                    "Critical",
                    "Critical",
                    "Important"
                  ]
                },
                {
                  "name": "Release Date",
                  "values": [
                    "5/14/2024",
                    "5/14/2024",
                    "5/14/2024",
                    "4/4/2012",
                    "4/9/2024",
                    "11/12/2024"
                  ]
                },
                {
                  "name": "Bulletins",
                  "values": [
                    "N/A",
                    "N/A",
                    "N/A",
                    "MS06-061",
                    "N/A",
                    "N/A"
                  ]
                },
                {
                  "name": "Size in Bytes",
                  "values": [
```


©2026 Tanium Inc. All Rights Reserved

```
                    "71843328",
                    "74789356",
                    "691901888",
                    "1853208",
                    "687076099",
                    "89234560"
                  ]
                },
                {
                  "name": "KB Articles",
                  "values": [
                    "KB890830",
                    "KB5038285",
                    "KB5037768",
                    "KB925673",
                    "KB5036892",
                    "KB5048239"
                  ]
                },
                {
                  "name": "CVE IDs",
                  "values": [
                    "None",
                    "None",
                    "CVE-2024-30006 CVE-2024-30008 CVE-2024-30009 CVE-2024-30010
CVE-2024-30013 CVE-2024-30014 CVE-2024-30015 CVE-2024-30016 CVE-2024-30017 CVE-2024-30019
CVE-2024-30020 CVE-2024-30022 CVE-2024-30040 CVE-2024-30051",
                    "CVE-2006-4685 CVE-2006-4686",
                    "CVE-2024-26169 CVE-2024-26184 CVE-2024-26234 CVE-2024-28899
CVE-2024-30040 CVE-2024-30051",
                    "CVE-2024-43625"
                  ]
                },
                {
                  "name": "Superseded",
                  "values": [
                    "False",
                    "False",
                    "False",
                    "False",
                    "True",
                    "False"
                  ]
                },
                {
                  "name": "Custom Field",
                  "values": [
                    "None",
                    "None",
                    "None",
                    "None",
                    "None",
                    "None"
                  ]
                },
                {
                  "name": "OS Type",
                  "values": [
                    "Windows",
                    "Windows",
                    "Windows",
                    "Windows",
                    "Windows",
                    "Windows"
```


©2026 Tanium Inc. All Rights Reserved

```
                ]
              },
              {
                "name": "Product",
                "values": [
                  "Windows 10---Windows 10 LTSB---Windows 10, version 1903 and later---
Windows 11",
                  "Windows 10, version 1903 and later",
                  "Windows 10, version 1903 and later",
                  "SQL Server Feature Pack",
                  "Windows 10, version 1903 and later",
                  "Windows 10, version 1903 and later"
                ]
              },
              {
                "name": "Classification",
                "values": [
                  "Update Rollups",
                  "Security Updates",
                  "Security Updates",
                  "Security Updates",
                  "Security Updates",
                  "Security Updates"
                ]
              }
            ]
          }
        }
      },
      {
        "node": {
          "computerID": "2502458661",
          "name": "ubuntu-2004",
          "sensorReadings": {
            "columns": [
              {
                "name": "Patch List Ids",
                "values": [
                  "1,22,28,3",
                  "1,22,28,3",
                  "1,22,28,3",
                  "1,22,28,3",
                  "1,22,28,3",
                  "1,22,28,3",
                  "1,22,28,3",
                  "1,22,28,3",
                  "1,22,28,3",
                  "1,22,26,28,3"
                ]
              },
              {
                "name": "Title",
                "values": [
                  "python3-debian 0.1.36ubuntu1.1 in focal",
                  "python3-apport 2.20.11-0ubuntu27.28 in focal",
                  "grub-common 2.04-1ubuntu26.17 in focal",
                  "python3-distro-info 0.23ubuntu1.1 in focal",
                  "iputils-ping 3:20190709-3ubuntu1 in focal",
                  "systemd-sysv 245.4-4ubuntu3.24 in focal",
                  "python3-update-manager 1:20.04.10.23 in focal",
                  "base-files 11ubuntu5.8 in focal",
                  "ubuntu-release-upgrader-core 1:20.04.41 in focal",
                  "linux-firmware 1.187.39 in focal"
                ]
```


©2026 Tanium Inc. All Rights Reserved

```
    },
    {
      "name": "Install Status",
      "values": [
        "Not Installed",
        "Not Installed",
        "Not Installed",
        "Not Installed",
        "Not Installed",
        "Not Installed",
        "Not Installed",
        "Not Installed",
        "Not Installed",
        "Not Installed"
      ]
    },
    {
      "name": "Uninstallable",
      "values": [
        "False",
        "False",
        "False",
        "False",
        "False",
        "False",
        "False",
        "False",
        "False",
        "False"
      ]
    },
    {
      "name": "Severity",
      "values": [
        "Moderate",
        "Moderate",
        "Moderate",
        "Moderate",
        "Moderate",
        "Moderate",
        "Moderate",
        "Moderate",
        "Moderate",
        "Moderate"
      ]
    },
    {
      "name": "Release Date",
      "values": [
        "7/13/2023",
        "5/12/2020",
        "6/5/2023",
        "8/3/2023",
        "1/9/2024",
        "9/17/2024",
        "9/25/2024",
        "1/18/2024",
        "3/14/2023",
        "5/15/2023"
      ]
    },
    {
      "name": "Bulletins",
      "values": [
```


©2026 Tanium Inc. All Rights Reserved

```
          "N/A",
          "N/A",
          "N/A",
          "N/A",
          "N/A",
          "N/A",
          "N/A",
          "N/A",
          "N/A",
          "N/A"
        ]
      },
      {
        "name": "Size in Bytes",
        "values": [
          "64932",
          "86980",
          "1871752",
          "5944",
          "40000",
          "10284",
          "38504",
          "60332",
          "24312",
          "136166424"
        ]
      },
      {
        "name": "KB Articles",
        "values": [
          "LAUNCHPAD-python3-debian",
          "LAUNCHPAD-python3-apport",
          "LAUNCHPAD-grub-common",
          "LAUNCHPAD-python3-distro-info",
          "LAUNCHPAD-iputils-ping",
          "LAUNCHPAD-systemd-sysv",
          "LAUNCHPAD-python3-update-manager",
          "LAUNCHPAD-base-files",
          "LAUNCHPAD-ubuntu-release-upgrader-core",
          "LAUNCHPAD-linux-firmware"
        ]
      },
      {
        "name": "CVE IDs",
        "values": [
          "None",
          "None",
          "None",
          "None",
          "None",
          "None",
          "None",
          "None",
          "None",
          "None"
        ]
      },
      {
        "name": "Superseded",
        "values": [
          "False",
          "False",
          "False",
          "False",
```


©2026 Tanium Inc. All Rights Reserved

```
          "False",
          "False",
          "False",
          "False",
          "False",
          "False"
        ]
      },
      {
        "name": "Custom Field",
        "values": [
          "None",
          "None",
          "None",
          "None",
          "None",
          "None",
          "None",
          "None",
          "None",
          "None"
        ]
      },
      {
        "name": "OS Type",
        "values": [
          "Linux",
          "Linux",
          "Linux",
          "Linux",
          "Linux",
          "Linux",
          "Linux",
          "Linux",
          "Linux",
          "Linux"
        ]
      },
      {
        "name": "Product",
        "values": [
          "Ubuntu 20",
          "Ubuntu 20",
          "Ubuntu 20",
          "Ubuntu 20",
          "Ubuntu 20",
          "Ubuntu 20",
          "Ubuntu 20",
          "Ubuntu 20",
          "Ubuntu 20",
          "Ubuntu 20"
        ]
      },
      {
        "name": "Classification",
        "values": [
          "updates",
          "security",
          "updates",
          "updates",
          "updates",
          "updates",
          "updates",
          "updates",
```


©2026 Tanium Inc. All Rights Reserved

```
                    "updates",
                    "updates"
                  ]
                }
              ]
            }
          }
        }
      ]
    }
  }
}
```

Create a Linux patch deployment

(mutation.patchCreateDeployment)

About this task

The `patchCreateDeployment` mutation requires Patch 3.16.191 or later and the **Patch Deployment write** and **Patch Deployment execute** permissions.

> **Note:** The `patchCreateDeployment` mutation does not support uploading images to pre-install and post-install user notifications.

The following mutation creates a new deployment for installing Linux patches on targeted endpoints.

```
mutation exampleCreateLinuxPatchDeployment($input: PatchCreateDeploymentInput!) {
  patchCreateDeployment(input: $input) {
    deployment {
      id
      author {
        id
        displayName
        username
      }
      contentDeploymentType
      contentSet {
        id
      }
      createdTime
      description
      downloadImmediately
      name
      notifications {
        postDeployment {
          body
          countdownToDeadline {
            nanoseconds
            seconds
          }
          notificationPeriodDuration {
            nanoseconds
```

©2026 Tanium Inc. All Rights Reserved

```
            seconds
          }
          postponeOptions {
            durationOne {
              nanoseconds
              seconds
            }
            durationTwo {
              nanoseconds
              seconds
            }
            durationThree {
              nanoseconds
              seconds
            }
          }
          title
          translations {
            body
            languageCode
            title
          }
        }
      }
      overrideBlocklists
      patches {
        id
      }
      patchLists {
        id
        version
      }
      platform
      restart
      schedule {
        distributeOver {
          nanoseconds
          seconds
        }
        endTime
        eussAvailableBeforeStart
        eussHideFromActivity
        overrideMaintenanceWindows
        startTime
        timeZone
        type
      }
      status
      stoppedTime
      targets {
        computerGroups {
          id
        }
      }
      type
      updatedTime
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "contentDeploymentType": "ALL_UPDATES",
    "contentSet": {
      "id": "123"
    },
    "description": "example Patch deployment for Linux",
    "downloadImmediately": true,
    "name": "example_deployment_linux",
    "overrideBlocklists": false,
    "patches": null,
    "patchLists": [
      {
        "id": "234",
        "version": 1
      }
    ],
    "platform": "LINUX",
    "restart": false,
    "schedule": {
      "distributeOver": {
        "seconds": 600
      },
      "endTime": "2025-02-01",
      "eussAvailableBeforeStart": false,
      "eussHideFromActivity": false,
      "overrideMaintenanceWindows": false,
      "startTime": "2025-01-01",
      "timeZone": "EST",
      "type": "SINGLE"
    },
    "targets": {
      "computerGroups": [
        {
          "id": "123"
        }
      ]
    },
    "type": "INSTALL"
  }
}
```

Example response:

```
{
  "data": {
    "patchCreateDeployment": {
      "deployment": {
        "id": "123",
        "author": {
          "id": "1",
          "displayName": null,
          "username": "Administrator"
        },
        "contentDeploymentType": "ALL_UPDATES",
        "contentSet": {
          "id": "123"
        },
        "createdTime": "2023-10-18T18:48:43.515Z",
```

©2026 Tanium Inc. All Rights Reserved

```
        "description": "example Patch deployment for Linux",
        "downloadImmediately": true,
        "name": "example_deployment_linux",
        "notifications": null,
        "overrideBlocklists": false,
        "patches": [],
        "patchLists": [
          {
            "id": "234",
            "version": 1
          }
        ],
        "platform": "LINUX",
        "restart": false,
        "schedule": {
          "distributeOver": {
            "nanoseconds": 0,
            "seconds": 600
          },
          "endTime": "2025-02-01T00:00:00.000Z",
          "eussAvailableBeforeStart": false,
          "eussHideFromActivity": false,
          "overrideMaintenanceWindows": false,
          "startTime": "2025-01-01T00:00:00.000Z",
          "timeZone": "EST",
          "type": "SINGLE"
        },
        "status": "INACTIVE",
        "stoppedTime": null,
        "targets": {
          "computerGroups": [
            {
              "id": "123"
            }
          ]
        },
        "type": "INSTALL",
        "updatedTime": "2023-10-18T18:48:43.515Z"
      },
      "error": null
    }
  }
}
```


Create a Windows patch deployment

(mutation.patchCreateDeployment)

About this task

The `patchCreateDeployment` mutation requires Patch 3.16.191 or later and the **Patch Deployment write** and **Patch Deployment execute** permissions.

©2026 Tanium Inc. All Rights Reserved

Note: The `patchCreateDeployment` mutation does not support uploading images to pre-install and post-install user notifications.

The following mutation creates a new deployment for installing Windows patches on targeted endpoints.

Note: Set `mutation.patchCreateDeployment (input.type)` equal to `INSTALL` for a Windows patch install deployment, or `UNINSTALL` for a Windows patch uninstall deployment.

```
mutation exampleCreateWindowsPatchDeployment($input: PatchCreateDeploymentInput!) {
  patchCreateDeployment(input: $input) {
    deployment {
      id
      author {
        id
        displayName
        username
      }
      contentDeploymentType
      contentSet {
        id
      }
      createdTime
      description
      downloadImmediately
      name
      notifications {
        postDeployment {
          body
          countdownToDeadline {
            nanoseconds
            seconds
          }
          notificationPeriodDuration {
            nanoseconds
            seconds
          }
          postponeOptions {
            durationOne {
              nanoseconds
              seconds
            }
            durationTwo {
              nanoseconds
              seconds
            }
            durationThree {
              nanoseconds
              seconds
            }
          }
          title
          translations {
            body
```

©2026 Tanium Inc. All Rights Reserved

```
            languageCode
            title
          }
        }
      }
      overrideBlocklists
      patches {
        id
      }
      patchLists {
        id
        version
      }
      platform
      restart
      schedule {
        distributeOver {
          nanoseconds
          seconds
        }
        endTime
        eussAvailableBeforeStart
        eussHideFromActivity
        overrideMaintenanceWindows
        startTime
        timeZone
        type
      }
      status
      stoppedTime
      targets {
        computerGroups {
          id
        }
      }
      type
      updatedTime
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "contentDeploymentType": "MANUAL_SELECTION",
    "contentSet": {
      "id": "123"
    },
    "description": "example Patch deployment for Windows",
    "downloadImmediately": true,
    "name": "example_deployment_windows",
    "notifications": {
      "postDeployment": {
        "body": "example text display",
        "countdownToDeadline": {
          "seconds": 15
```

©2026 Tanium Inc. All Rights Reserved

```
        },
        "notificationPeriodDuration": {
          "seconds": 1440
        },
        "postponeOptions": {
          "durationOne": {
            "seconds": 60
          },
          "durationTwo": {
            "seconds": 120
          },
          "durationThree": {
            "seconds": 240
          }
        },
        "title": "example title",
        "translations": [
          {
            "body": "ejemplo",
            "languageCode": "es",
            "title": "texto"
          }
        ]
      }
    },
    "overrideBlocklists": false,
    "patches": null,
    "patchLists": [
      {
        "id": "234",
        "version": 1
      }
    ],
    "platform": "WINDOWS",
    "restart": true,
    "schedule": {
      "distributeOver": {
        "seconds": 600
      },
      "endTime": null,
      "eussAvailableBeforeStart": false,
      "eussHideFromActivity": false,
      "overrideMaintenanceWindows": false,
      "startTime": "2025-01-01",
      "timeZone": "EST",
      "type": "ONGOING"
    },
    "targets": {
      "computerGroups": [
        {
          "id": "345"
        }
      ]
    },
    "type": "INSTALL"
  }
}
```

Example response:

```
{
  "data": {
```

```
  "patchCreateDeployment": {
    "deployment": {
      "id": "123",
      "author": {
        "id": "1",
        "displayName": null,
        "username": "admin"
      },
      "contentDeploymentType": "MANUAL_SELECTION",
      "contentSet": {
        "id": "123"
      },
      "createdTime": "2023-10-24T17:59:00.274Z",
      "description": "example Patch deployment for Windows",
      "downloadImmediately": true,
      "name": "example_deployment_windows",
      "notifications": {
        "postDeployment": {
          "body": "example text display",
          "countdownToDeadline": {
            "nanoseconds": 0,
            "seconds": 0
          },
          "notificationPeriodDuration": {
            "nanoseconds": 0,
            "seconds": 1440
          },
          "postponeOptions": {
            "durationOne": {
              "nanoseconds": 0,
              "seconds": 60
            },
            "durationTwo": {
              "nanoseconds": 0,
              "seconds": 120
            },
            "durationThree": {
              "nanoseconds": 0,
              "seconds": 240
            }
          },
          "title": "example title",
          "translations": [
            {
              "body": "ejemplo",
              "languageCode": "es",
              "title": "texto"
            }
          ]
        }
      },
      "overrideBlocklists": false,
      "patches": [],
      "patchLists": [
        {
          "id": "234",
          "version": 1
        }
      ],
      "platform": "WINDOWS",
      "restart": true,
      "schedule": {
        "distributeOver": {
          "nanoseconds": 0,
```


©2026 Tanium Inc. All Rights Reserved

```
            "seconds": 600
          },
          "endTime": null,
          "eussAvailableBeforeStart": false,
          "eussHideFromActivity": false,
          "overrideMaintenanceWindows": false,
          "startTime": "2025-01-01T00:00:00.000Z",
          "timeZone": "EST",
          "type": "ONGOING"
        },
        "status": "INACTIVE",
        "stoppedTime": null,
        "targets": {
          "computerGroups": [
            {
              "id": "345"
            }
          ]
        },
        "type": "INSTALL",
        "updatedTime": "2023-10-24T17:59:00.274Z"
      },
      "error": null
    }
  }
}
```

Get multiple patch definitions

(query.patchDefinitions)

About this task

The `patchDefinition` query requires Patch 3.17.175 or later and the **Patch show** permission.

The following query retrieves the first patch definition that matches the specified CVE ID.



Note: The filter supports one, and only one, CVE ID to filter the returned patch definitions.

```
query exampleGetPatchDefinitions($first: Int, $filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(first: $first, filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
```

©2026 Tanium Inc. All Rights Reserved

```
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1,
  "filter": {
    "cveIds": [
      "CVE-2014-123456"
    ]
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Example response:

```
{
  "data": {
    "patchDefinitions": {
      "edges": [
        {
          "node": {
            "id": "1234567890abcdef1234567890abcdef",
            "createdDate": "2021-04-20",
            "cveIds": [
              "CVE-2014-123456"
            ],
            "isSuperseded": false,
            "platform": "LINUX",
            "releaseDate": "2020-01-01",
            "severity": "Important",
            "sizeInBytes": 12345678,
            "title": "example patch definition"
          }
        }
      ],
      "pageInfo": {
        "hasNextPage": false,
        "hasPreviousPage": false,
        "startCursor":
"eyJpZCI6ImE4OWE0ZGFkZWY2YWU0YTAzZDBjYzY5NGFhZjg4MzMzIiwiY3JlYXRlZERhdGUiOiIyMDIxLTA0LTIwVDA2
OjAxOjQ2Ljg1MloiLCJyZWxlYXNlRGF0ZSI6IjIwMTItMDYtMThUMDA6MDA6MDAuMDAwWiJ9",
        "endCursor":
"eyJpZCI6ImE4OWE0ZGFkZWY2YWU0YTAzZDBjYzY5NGFhZjg4MzMzIiwiY3JlYXRlZERhdGUiOiIyMDIxLTA0LTIwVDA2
OjAxOjQ2Ljg1MloiLCJyZWxlYXNlRGF0ZSI6IjIwMTItMDYtMThUMDA6MDA6MDAuMDAwWiJ9"
      },
      "totalRecords": 1
    }
  }
```

```
}
```

Get a patch deployment by ID

(query.patchDeployment)

About this task

The `patchDeployment` query requires Patch 3.16.191 or later and the **Patch Deployment read** permission.

The following query retrieves the Patch deployment that matches the specified ID.

```
query exampleGetPatchDeployment($id: ID!) {
  patchDeployment(ref: {id: $id}) {
    id
    author {
      id
      displayName
      username
    }
    contentDeploymentType
    contentSet {
      id
    }
    createdTime
    description
    downloadImmediately
    name
    notifications {
      postDeployment {
        body
        countdownToDeadline {
          nanoseconds
          seconds
        }
        notificationPeriodDuration {
          nanoseconds
          seconds
        }
        postponeOptions {
          durationOne {
            nanoseconds
            seconds
          }
          durationTwo {
            nanoseconds
            seconds
          }
          durationThree {
            nanoseconds
            seconds
          }
        }
        title
        translations {
          body
```

```
          languageCode
          title
        }
      }
    }
    overrideBlocklists
    patches {
      id
    }
    patchLists {
      id
      version
    }
    platform
    restart
    schedule {
      distributeOver {
        nanoseconds
        seconds
      }
      endTime
      eussAvailableBeforeStart
      eussHideFromActivity
      overrideMaintenanceWindows
      startTime
      timeZone
      type
    }
    status
    stoppedTime
    targets {
      computerGroups {
        id
      }
    }
    type
    updatedTime
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "123"
}
```

Example response:

```
{
  "data": {
    "patchDeployment": {
      "id": "123",
      "author": {
        "id": "1",
        "displayName": null,
        "username": "Administrator"
      },
      "contentDeploymentType": "ALL_UPDATES",
      "contentSet": {
        "id": "123"
      },
```

©2026 Tanium Inc. All Rights Reserved

```
      "createdTime": "2023-10-18T18:48:43.515Z",
      "description": "example Patch deployment for Linux",
      "downloadImmediately": true,
      "name": "example_deployment_linux",
      "notifications": null,
      "overrideBlocklists": false,
      "patches": [],
      "patchLists": [
        {
          "id": "234",
          "version": 1
        }
      ],
      "platform": "LINUX",
      "restart": false,
      "schedule": {
        "distributeOver": {
          "nanoseconds": 0,
          "seconds": 600
        },
        "endTime": "2025-02-01T00:00:00.000Z",
        "eussAvailableBeforeStart": false,
        "eussHideFromActivity": false,
        "overrideMaintenanceWindows": false,
        "startTime": "2025-01-01T00:00:00.000Z",
        "timeZone": "EST",
        "type": "SINGLE"
      },
      "status": "INACTIVE",
      "stoppedTime": null,
      "targets": {
        "computerGroups": [
          {
            "id": "1"
          }
        ]
      },
      "type": "INSTALL",
      "updatedTime": "2023-10-18T18:48:44.693Z"
    }
  }
}
```

Stop a patch deployment by ID

(mutation.patchStopDeployment)

About this task

The `patchStopDeployment` mutation requires Patch 3.16.191 or later and the **Patch Deployment write** and **Patch Deployment execute** permissions.

The following mutation updates a deployment that matches the ID `123`.

```
mutation exampleStopPatchDeployment($id: ID!) {
  patchStopDeployment(ref: {id: $id}) {
    deployment {
```

©2026 Tanium Inc. All Rights Reserved

```
      id
      author {
        id
        displayName
        username
      }
      contentDeploymentType
      contentSet {
        id
      }
      createdTime
      description
      downloadImmediately
      name
      notifications {
        postDeployment {
          body
          countdownToDeadline {
            nanoseconds
            seconds
          }
          notificationPeriodDuration {
            nanoseconds
            seconds
          }
          postponeOptions {
            durationOne {
              nanoseconds
              seconds
            }
            durationTwo {
              nanoseconds
              seconds
            }
            durationThree {
              nanoseconds
              seconds
            }
          }
          title
          translations {
            body
            languageCode
            title
          }
        }
      }
      overrideBlocklists
      patches {
        id
      }
      patchLists {
        id
        version
      }
      platform
      restart
      schedule {
        distributeOver {
          nanoseconds
          seconds
        }
        endTime
        eussAvailableBeforeStart
```

©2026 Tanium Inc. All Rights Reserved

```
        eussHideFromActivity
        overrideMaintenanceWindows
        startTime
        timeZone
        type
      }
      status
      stoppedTime
      targets {
        computerGroups {
          id
        }
      }
      type
      updatedTime
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "id": "123"
}
```

Example response:

```
{
  "data": {
    "patchStopDeployment": {
      "deployment": {
        "id": "123",
        "author": {
          "id": "1",
          "displayName": null,
          "username": "Administrator"
        },
        "contentDeploymentType": "ALL_UPDATES",
        "contentSet": {
          "id": "123"
        },
        "createdTime": "2023-10-18T18:48:43.515Z",
        "description": "example Patch deployment for Linux",
        "downloadImmediately": true,
        "name": "example_deployment_linux",
        "notifications": null,
        "overrideBlocklists": false,
        "patches": null,
        "patchLists": [
          {
            "id": "234",
            "version": 1
          }
        ],
        "platform": "LINUX",
        "restart": false,
```

©2026 Tanium Inc. All Rights Reserved

```
        "schedule": {
          "distributeOver": {
            "nanoseconds": 0,
            "seconds": 600
          },
          "endTime": "2025-02-01T00:00:00.000Z",
          "eussAvailableBeforeStart": false,
          "eussHideFromActivity": false,
          "overrideMaintenanceWindows": false,
          "startTime": "2025-01-01T00:00:00.000Z",
          "timeZone": "EST",
          "type": "SINGLE"
        },
        "status": "STOPPED",
        "stoppedTime": "2023-10-18T19:16:28.105Z",
        "targets": {
          "computerGroups": [
            {
              "id": "1"
            }
          ]
        },
        "type": "INSTALL",
        "updatedTime": "2023-10-18T19:16:28.123Z"
      },
      "error": null
    }
  }
}
```


Upsert (create or update by ID) a patch list

(mutation.patchListUpsert)

The `patchListUpsert` mutation requires Patch 3.24.83 or later and the **Patch Patchlist execute** and **Patch Patchlist write** permissions.

The `patchListUpsert` mutation either creates a new patch list or updates the existing patch list that matches an ID.



Note: Although you can manually select patches to include in a patch list, it is more efficient to use rules to dynamically populate lists of patches.

The following mutation creates a new Linux patch list. The list defines two rules:

- include all patches with "example" in the title
- include all critical severity patches released in the past seven days

```
mutation examplePatchListCreate($input: PatchListUpsertInput!) {
```

©2026 Tanium Inc. All Rights Reserved

```
  patchListUpsert(input: $input) {
    patchList {
      id
      name
      author {
        id
        displayName
        username
      }
      contentSet {
        id
        description
        name
      }
      createdTime
      includeSupersededPatches
      isTaniumManaged
      patchDefinitionIDs
      platform
      rules {
        name
        conditions {
          comparisonOperator
          fieldName
          fieldValue
        }
      }
      updatedTime
      version
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "name": "example_patch_list",
    "platform": "LINUX",
    "contentSet": {
      "id": "234"
    },
    "patchDefinitionIDs": [],
    "includeSupersededPatches": false,
    "rules": [
      {
        "name": "example_rule_1",
        "conditions": [
          {
            "fieldName": "TITLE",
            "comparisonOperator": "CONTAINS",
            "fieldValue": {
              "value": "example"
            }
          }
        ]
      },
```

©2026 Tanium Inc. All Rights Reserved

```
      {
        "name": "example_rule_2",
        "conditions": [
          {
            "fieldName": "RELEASE_DATE",
            "comparisonOperator": "ON_OR_BEFORE_RELATIVE_DAYS",
            "fieldValue": {
              "releaseDateDaysAgo": 7
            }
          },
          {
            "fieldName": "SEVERITY",
            "comparisonOperator": "EQUALS",
            "fieldValue": {
              "severity": "CRITICAL"
            }
          }
        ]
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "patchListUpsert": {
      "patchList": {
        "id": "123",
        "name": "example_patch_list",
        "author": {
          "id": "1",
          "displayName": null,
          "username": "admin"
        },
        "contentSet": {
          "id": "53",
          "description": "Default container for objects.",
          "name": "Patch Content Set"
        },
        "createdTime": "2025-02-03T14:24:43.327Z",
        "includeSupersededPatches": false,
        "isTaniumManaged": false,
        "patchDefinitionIDs": [],
        "platform": "LINUX",
        "rules": [
          {
            "name": "example_rule_1",
            "conditions": [
              {
                "comparisonOperator": "CONTAINS",
                "fieldName": "TITLE",
                "fieldValue": "example"
              }
            ]
          },
          {
            "name": "example_rule_2",
            "conditions": [
              {
                "comparisonOperator": "ON_OR_BEFORE_RELATIVE_DAYS",
```

©2026 Tanium Inc. All Rights Reserved

```
                "fieldName": "RELEASE_DATE",
                "fieldValue": "7"
              },
              {
                "comparisonOperator": "EQUALS",
                "fieldName": "SEVERITY",
                "fieldValue": "CRITICAL"
              }
            ]
          }
        ],
        "updatedTime": "2025-02-03T14:24:43.370Z",
        "version": 1
      },
      "error": null
    }
  }
}
```

The following mutation updates the patch list with the ID `123`.

```
mutation examplePatchListUpdate($input: PatchListUpsertInput!) {
  patchListUpsert(input: $input) {
    patchList {
      id
      name
      author {
        id
        displayName
        username
      }
      contentSet {
        id
        description
        name
      }
      createdTime
      includeSupersededPatches
      isTaniumManaged
      patchDefinitionIDs
      platform
      rules {
        name
        conditions {
          comparisonOperator
          fieldName
          fieldValue
        }
      }
      updatedTime
      version
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "input": {
    "id": "123",
    "name": "example_patch_list",
    "platform": "LINUX",
    "contentSet": {
      "id": "234"
    },
    "patchDefinitionIDs": [],
    "includeSupersededPatches": false,
    "rules": [
      {
        "name": "example_rule_1",
        "conditions": [
          {
            "fieldName": "TITLE",
            "comparisonOperator": "CONTAINS",
            "fieldValue": {
              "value": "example"
            }
          }
        ]
      }
    ]
  }
}
```

Example response:

```
{
  "data": {
    "patchListUpsert": {
      "patchList": {
        "id": "123",
        "name": "example_patch_list",
        "author": {
          "id": "1",
          "displayName": null,
          "username": "admin"
        },
        "contentSet": {
          "id": "53",
          "description": "Default container for objects.",
          "name": "Patch Content Set"
        },
        "createdTime": "2025-02-03T14:24:43.327Z",
        "includeSupersededPatches": false,
        "isTaniumManaged": false,
        "patchDefinitionIDs": [],
        "platform": "LINUX",
        "rules": [
          {
            "name": "example_rule_1",
            "conditions": [
              {
                "comparisonOperator": "CONTAINS",
                "fieldName": "TITLE",
                "fieldValue": "example"
              }
            ]
          }
        ],
```

©2026 Tanium Inc. All Rights Reserved

```
        "updatedTime": "2025-02-03T14:24:43.370Z",
        "version": 1
      },
      "error": null
    }
  }
}
```

Provision examples

The following requests require Provision to create, retrieve, update, and delete retirement action definitions, and create, retrieve, and update retirement actions.

Provision requests have the following feature-specific dependencies at the specified minimum versions:

- Provision 1.19 or later is required to submit a query or mutation request related to retirement action definitions or retirement actions.

(1.0 - Experimental (Early Development)) Cancel retirement action

(mutation.provisionCancelRetirementAction)

About this task

The `provisionCancelRetirementAction` mutation requires Provision 1.19 or later and the **Provision Retirement Actions write** permission.

The following mutation cancels the retirement action that matches the ID `12`, updating the status from `pending` to `canceled`.



Note: This mutation has no effect if the retirement action status is not equal to `pending`.

```
mutation exampleProvisionCancelRetirementAction($input: ID!) {
  provisionCancelRetirementAction(input: $input) {
    retirementAction {
      id
      definitionID
      status
      eventLogs {
        edges {
          node {
            id
            stepName
            status
```

©2026 Tanium Inc. All Rights Reserved

```
            details
            actionID
            messageID
            createdTime
            endpointActionCode
          }
        }
        totalRecords
      }
      notes {
        comment
        requestedBy
        ticketNumber
      }
      manualEntry
      ipAddress
      natIPAddress
      createdTime
      updatedTime
      createdBy {
        persona {
          id
          name
        }
        user {
          id
          username
          displayName
        }
      }
      updatedBy {
        persona {
          id
          name
        }
        user {
          id
          username
          displayName
        }
      }
    }
    error {
      message
    }
  }
}
```

Include the input variable in the **Variables** panel or in your variables dictionary:

```
{
  "input": 12
}
```

Example response:

```
{
  "data": {
    "provisionCancelRetirementAction": {
      "retirementAction": {
        "id": "12",
```

```
        "definitionID": "1",
        "status": "canceled",
        "eventLogs": {
          "edges": [],
          "totalRecords": 0
        },
        "notes": {
          "comment": "this is a comment",
          "requestedBy": "requested by test",
          "ticketNumber": null
        },
        "manualEntry": true,
        "ipAddress": null,
        "natIPAddress": null,
        "createdTime": "2025-02-17T15:37:43.031212Z",
        "updatedTime": "2025-02-17T15:41:55.423116Z",
        "createdBy": {
          "persona": {
            "id": "0",
            "name": null
          },
          "user": {
            "id": "1",
            "username": "admin",
            "displayName": null
          }
        },
        "updatedBy": {
          "persona": {
            "id": "0",
            "name": null
          },
          "user": {
            "id": "1",
            "username": "admin",
            "displayName": null
          }
        }
      },
      "error": null
    }
  }
}
```


(1.0 - Experimental (Early Development)) Create retirement action

(mutation.provisionCreateRetirementAction)

About this task

The `provisionCreateRetirementAction` mutation requires Provision 1.19 or later and the **Provision Retirement Actions write** permission.

The following mutation creates a retirement action to retire a single endpoint, based on the retirement action definition that matches the ID `1`. You can target an endpoint using either an `endpointID` input value or `targetEndpointParameters` values, but not both.

©2026 Tanium Inc. All Rights Reserved

Note: Device retirement actions are supported on Windows endpoints only.

```
mutation exampleProvisionCreateRetirementAction($input:
ProvisionCreateRetirementActionInput!) {
  provisionCreateRetirementAction(input: $input) {
    retirementAction {
      id
      definitionID
      status
      notes {
        comment
        requestedBy
        ticketNumber
      }
      manualEntry
      ipAddress
      natIPAddress
      createdTime
      updatedTime
      createdBy {
        persona {
          id
          name
        }
        user {
          id
          username
          displayName
        }
      }
      updatedBy {
        persona {
          id
          name
        }
        user {
          id
          username
          displayName
        }
      }
    }
    error {
      message
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "definitionID": "1",
    "notes": {
      "comment": "this is a comment",
      "ticketNumber": null,
      "requestedBy": "requested by test"
    },
    "targetEndpointParameters": {
      "computerName": "target computerName",
```

©2026 Tanium Inc. All Rights Reserved

```
      "serialNumber": "target serialNumber",
      "manufacturer": "target manufacturer",
      "model": "target model"
    }
  }
}
```

Example response:

```
{
  "data": {
    "provisionCreateRetirementAction": {
      "retirementAction": {
        "id": "12",
        "definitionID": "1",
        "status": "pending",
        "notes": {
          "comment": "this is a comment",
          "requestedBy": "requested by test",
          "ticketNumber": null
        },
        "manualEntry": true,
        "ipAddress": null,
        "natIPAddress": null,
        "createdTime": "2025-02-17T15:37:43.031212Z",
        "updatedTime": "2025-02-17T15:37:43.031212Z",
        "createdBy": {
          "persona": {
            "id": "0",
            "name": null
          },
          "user": {
            "id": "1",
            "username": "admin",
            "displayName": null
          }
        },
        "updatedBy": {
          "persona": {
            "id": "0",
            "name": null
          },
          "user": {
            "id": "1",
            "username": "admin",
            "displayName": null
          }
        }
      },
      "error": null
    }
  }
}
```

(1.0 - Experimental (Early Development)) Create retirement action definition

(mutation.provisionCreateRetirementActionDefinition)

About this task

The `provisionCreateRetirementActionDefinition` mutation requires Provision 1.19 or later and the **Provision Retirement Action Definitions write** permission.

Create a retirement action definition as a template for retirement actions. Configure retirement action options based on the following:

- If `userLock` is set to `true`, `userUnlock` must be set to `false`.
- If `userUnlock` is set to `true`, `userLock`, `deleteFiles`, and `bitlockerFreeze` must all be set to `false`.
- If `deleteFiles` is set to `true`, `userUnlock` must be set to `false`.

  Tip: For the best results, if `deleteFiles` is set to `true`, set `userLock` and `bitlockerFreeze` to `true`, and assign a value to `bitlockerMessage`.

- If `bitlockerFreeze` is set to `true`, `userUnlock` must be set to `false`, and you must assign a value to `bitlockerMessage`.

  Tip: For the best results, if `bitlockerFreeze` is set to `true`, set `userLock` to `true`.



Note: Device retirement actions are supported on Windows endpoints only.



Note: Set `enabled` to `true` to allow users to create a retirement action using this retirement action definition.

The following mutation creates a retirement action definition with `userUnlock` set equal to `true`, and defined notes:

```
mutation exampleProvisionCreateRetirementActionDefinition($input:
ProvisionCreateRetirementActionDefinitionInput!) {
  provisionCreateRetirementActionDefinition(input: $input) {
    actionDefinition {
```

©2026 Tanium Inc. All Rights Reserved

```
      id
      name
      description
      endpointActionOptions {
        userLock
        userUnlock
        deleteFiles
        bitlockerFreeze
        bitlockerMessage
      }
      enabled
      deletedTime
      createdTime
      updatedTime
      createdBy {
        user {
          username
          id
          displayName
        }
        persona {
          name
          id
        }
      }
      updatedBy {
        user {
          username
          id
          displayName
        }
        persona {
          name
          id
        }
      }
      notesDefinition {
        requestedBy {
          notesRequired
          label
        }
        ticketNumber {
          notesRequired
          label
        }
        comment {
          notesRequired
          label
        }
      }
      retirementActionsCount
    }
    error {
      message
      retryable
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "input": {
    "name": "Example retirement action definition",
    "description": "example retirement action definition",
    "endpointActionOptions": {
      "userLock": false,
      "userUnlock": true,
      "deleteFiles": false,
      "bitlockerFreeze": false,
      "bitlockerMessage": ""
    },
    "enabled": true,
    "notesDefinition": {
      "requestedBy": {
        "label": "requestedBy label",
        "notesRequired": "OPTIONAL"
      },
      "comment": {
        "label": "comment label",
        "notesRequired": "REQUIRED"
      },
      "ticketNumber": {
        "notesRequired": "NOT_ALLOWED"
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "provisionCreateRetirementActionDefinition": {
      "actionDefinition": {
        "id": "6",
        "name": "Example retirement action definition",
        "description": "example retirement action definition",
        "endpointActionOptions": {
          "userLock": false,
          "userUnlock": true,
          "deleteFiles": false,
          "bitlockerFreeze": false,
          "bitlockerMessage": ""
        },
        "enabled": true,
        "deletedTime": null,
        "createdTime": "2025-02-17T15:07:40.792724Z",
        "updatedTime": "2025-02-17T15:07:40.792724Z",
        "createdBy": {
          "user": {
            "username": "admin",
            "id": "1",
            "displayName": null
          },
          "persona": {
            "name": null,
            "id": "0"
          }
        },
        "updatedBy": {
          "user": {
            "username": "admin",
            "id": "1",
```

©2026 Tanium Inc. All Rights Reserved

```
            "displayName": null
          },
          "persona": {
            "name": null,
            "id": "0"
          }
        },
        "notesDefinition": {
          "requestedBy": {
            "notesRequired": "OPTIONAL",
            "label": "requestedBy label"
          },
          "ticketNumber": {
            "notesRequired": "NOT_ALLOWED",
            "label": null
          },
          "comment": {
            "notesRequired": "REQUIRED",
            "label": "comment label"
          }
        },
        "retirementActionsCount": 0
      },
      "error": null
    }
  }
}
```

(1.0 - Experimental (Early Development)) Delete retirement action definitions

(mutation.provisionDeleteRetirementActionDefinitions)

About this task

The `provisionDeleteRetirementActionDefinitions` mutation requires Provision 1.19 or later and the **Provision Retirement Action Definitions write** permission.

Delete one or more retirement action definitions that match the input IDs.

Note: If you delete a retirement action definition, the `deleted_at` value is updated to the current timestamp. You can still get the retirement action definition in a query request, but cannot use it to create retirement actions.

Note: Deleting a retirement action definition does not affect any retirement actions based on the retirement action definition.

The following mutation deletes the retirement action definitions that match the IDs `1` and `2`.

©2026 Tanium Inc. All Rights Reserved

```
mutation exampleProvisionDeleteRetirementActionDefinitions($input:
ProvisionDeleteRetirementActionDefinitionsInput!) {
  provisionDeleteRetirementActionDefinitions(input: $input) {
    error {
      message
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "ids": [
      "1",
      "2"
    ]
  }
}
```

Example response:

```
{
  "data": {
    "provisionDeleteRetirementActionDefinitions": {
      "error": null
    }
  }
}
```

Get retirement action by ID

(query.provisionRetirementAction)

About this task

The `provisionRetirementAction` query requires Provision 1.19 or later and the **Provision Retirement Actions read** permission.

The following query retrieves the retirement action that matches the ID `1`.

```
query exampleGetProvisionRetirementAction($ref: IdRefInput!) {
  provisionRetirementAction(ref: $ref) {
    id
    definitionID
    status
    notes {
      requestedBy
      ticketNumber
      comment
    }
    manualEntry
    targetEndpointParameters {
      computerName
      serialNumber
```

©2026 Tanium Inc. All Rights Reserved

```
      manufacturer
      model
    }
    endpointID
    ipAddress
    natIPAddress
    createdTime
    updatedTime
    eventLogs {
      edges {
        cursor
        node {
          id
          stepName
          status
          details
          actionID
          messageID
          createdTime
          endpointActionCode
        }
      }
      pageInfo {
        hasNextPage
        hasPreviousPage
        startCursor
        endCursor
      }
      totalRecords
    }
    createdBy {
      user {
        username
        id
        displayName
      }
      persona {
        id
        name
      }
    }
    updatedBy {
      user {
        username
        id
        displayName
      }
      persona {
        id
        name
      }
    }
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "ref": {
    "id": "1"
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "provisionRetirementAction": {
      "id": "12",
      "definitionID": "1",
      "status": "pending",
      "notes": {
        "requestedBy": "updated requested by",
        "ticketNumber": null,
        "comment": "updated new comment"
      },
      "manualEntry": true,
      "targetEndpointParameters": {
        "computerName": "target computerName",
        "serialNumber": "target serialNumber",
        "manufacturer": "target manufacturer",
        "model": "target model"
      },
      "endpointID": "",
      "ipAddress": null,
      "natIPAddress": null,
      "createdTime": "2025-02-17T15:44:51.771139Z",
      "updatedTime": "2025-02-17T15:48:06.389774Z",
      "eventLogs": {
        "edges": [],
        "pageInfo": {
          "hasNextPage": false,
          "hasPreviousPage": false,
          "startCursor": null,
          "endCursor": null
        },
        "totalRecords": 0
      },
      "createdBy": {
        "user": {
          "username": "admin",
          "id": "1",
          "displayName": null
        },
        "persona": {
          "id": "0",
          "name": null
        }
      },
      "updatedBy": {
        "user": {
          "username": "admin",
          "id": "1",
          "displayName": null
        },
        "persona": {
          "id": "0",
          "name": null
        }
      }
    }
  }
}
```


©2026 Tanium Inc. All Rights Reserved

(1.0 - Experimental (Early Development)) Get retirement action definition by ID

(query.provisionRetirementActionDefinition)

About this task

The `provisionRetirementActionDefinition` query requires Provision 1.19 or later and the **Provision Retirement Action Definitions read** permission.

The following query retrieves the retirement action definition that matches the ID `1`.

```
query exampleGetProvisionRetirementActionDefinitionByID($ref: IdRefInput!) {
  provisionRetirementActionDefinition(ref: $ref) {
    id
    name
    description
    endpointActionOptions {
      userLock
      userUnlock
      deleteFiles
      bitlockerFreeze
      bitlockerMessage
    }
    enabled
    deletedTime
    createdTime
    updatedTime
    createdBy {
      user {
        id
        username
        displayName
      }
      persona {
        id
        name
      }
    }
    updatedBy {
      user {
        id
        username
        displayName
      }
      persona {
        id
        name
      }
    }
    notesDefinition {
      requestedBy {
        notesRequired
        label
      }
      ticketNumber {
        notesRequired
        label
      }
```

©2026 Tanium Inc. All Rights Reserved

```
      comment {
        notesRequired
        label
      }
    }
    retirementActionsCount
  }
}
```

Include the ID variable in the **Variables** panel or in your variables dictionary:

```
{
  "ref": {
    "id": "1"
  }
}
```

Example response:

```
{
  "data": {
    "provisionRetirementActionDefinition": {
      "id": "1",
      "name": "Example retirement action definition",
      "description": "example description",
      "endpointActionOptions": {
        "userLock": false,
        "userUnlock": true,
        "deleteFiles": false,
        "bitlockerFreeze": false,
        "bitlockerMessage": ""
      },
      "enabled": true,
      "deletedTime": null,
      "createdTime": "2025-02-17T15:07:40.792724Z",
      "updatedTime": "2025-02-17T15:07:40.792724Z",
      "createdBy": {
        "user": {
          "id": "1",
          "username": "admin",
          "displayName": null
        },
        "persona": {
          "id": "0",
          "name": null
        }
      },
      "updatedBy": {
        "user": {
          "id": "1",
          "username": "admin",
          "displayName": null
        },
        "persona": {
          "id": "0",
          "name": null
        }
      },
      "notesDefinition": {
        "requestedBy": {
          "notesRequired": "OPTIONAL",
```

©2026 Tanium Inc. All Rights Reserved

```
          "label": "requestedBy label"
        },
        "ticketNumber": {
          "notesRequired": "NOT_ALLOWED",
          "label": ""
        },
        "comment": {
          "notesRequired": "REQUIRED",
          "label": "comment label"
        }
      },
      "retirementActionsCount": 0
    }
  }
}
```

(1.0 - Experimental (Early Development)) Get multiple retirement action definitions

(query.provisionRetirementActionDefinitions)

About this task

The `provisionRetirementActionDefinitions` query requires Provision 1.19 or later and the **Provision Retirement Action Definitions read** permission.

The following query retrieves the first `2` retirement action definitions.

```
query exampleGetMultipleProvisionRetirementActionDefinitions($first: Int) {
  provisionRetirementActionDefinitions(first: $first) {
    edges {
      cursor
      node {
        id
        name
        description
        endpointActionOptions {
          userLock
          userUnlock
          deleteFiles
          bitlockerFreeze
          bitlockerMessage
        }
        enabled
        deletedTime
        createdTime
        updatedTime
        createdBy {
          user {
            username
            id
            displayName
          }
          persona {
            id
            name
          }
        }
```

```
        updatedBy {
          user {
            username
            id
            displayName
          }
          persona {
            id
            name
          }
        }
        notesDefinition {
          requestedBy {
            label
            notesRequired
          }
          ticketNumber {
            label
            notesRequired
          }
          comment {
            label
            notesRequired
          }
        }
        retirementActionsCount
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "provisionRetirementActionDefinitions": {
      "edges": [
        {
          "cursor": "ZGVmaW5pdGlvbjow",
          "node": {
            "id": "2",
            "name": "example retirement action definition 2",
            "description": "example description",
            "endpointActionOptions": {
              "userLock": false,
              "userUnlock": false,
              "deleteFiles": true,
              "bitlockerFreeze": false,
```

©2026 Tanium Inc. All Rights Reserved

```
            "bitlockerMessage": ""
          },
          "enabled": true,
          "deletedTime": null,
          "createdTime": "2023-08-09T19:51:31.623351Z",
          "updatedTime": "2023-08-09T19:53:01.446715Z",
          "createdBy": {
            "user": {
              "username": "admin",
              "id": "1",
              "displayName": null
            },
            "persona": {
              "id": "0",
              "name": null
            }
          },
          "updatedBy": {
            "user": {
              "username": "admin",
              "id": "1",
              "displayName": null
            },
            "persona": {
              "id": "0",
              "name": null
            }
          },
          "notesDefinition": {
            "requestedBy": {
              "label": "",
              "notesRequired": "OPTIONAL"
            },
            "ticketNumber": {
              "label": "",
              "notesRequired": "REQUIRED"
            },
            "comment": {
              "label": "",
              "notesRequired": "OPTIONAL"
            }
          },
          "retirementActionsCount": 2
        }
      },
      {
        "cursor": "ZGVmaW5pdGlvbjox",
        "node": {
          "id": "1",
          "name": "example retirement action definition",
          "description": "",
          "endpointActionOptions": {
            "userLock": true,
            "userUnlock": false,
            "deleteFiles": false,
            "bitlockerFreeze": true,
            "bitlockerMessage": "Please contact the Helpdesk to unlock"
          },
          "enabled": true,
          "deletedTime": null,
          "createdTime": "2023-07-21T13:16:05.660767Z",
          "updatedTime": "2023-08-09T19:50:40.716427Z",
          "createdBy": {
            "user": {
```


©2026 Tanium Inc. All Rights Reserved

```
                "username": "admin",
                "id": "1",
                "displayName": null
              },
              "persona": {
                "id": "0",
                "name": null
              }
            },
            "updatedBy": {
              "user": {
                "username": "admin",
                "id": "1",
                "displayName": null
              },
              "persona": {
                "id": "0",
                "name": null
              }
            },
            "notesDefinition": {
              "requestedBy": {
                "label": "Admin User",
                "notesRequired": "REQUIRED"
              },
              "ticketNumber": {
                "label": "",
                "notesRequired": "NOT_ALLOWED"
              },
              "comment": {
                "label": "",
                "notesRequired": "NOT_ALLOWED"
              }
            },
            "retirementActionsCount": 1
          }
        }
      ],
      "pageInfo": {
        "hasNextPage": true,
        "hasPreviousPage": false,
        "startCursor": "ZGVmaW5pdGlvbjow",
        "endCursor": "ZGVmaW5pdGlvbjox"
      },
      "totalRecords": 8
    }
  }
}
```


(1.0 - Experimental (Early Development)) Get multiple retirement actions

(query.provisionRetirementActions)

About this task

The `provisionRetirementActions` query requires Provision 1.19 or later and the **Provision Retirement Actions read** permission.

©2026 Tanium Inc. All Rights Reserved

The following query retrieves the first `2` retirement actions.

```
query exampleGetMultipleProvisionRetirementActions($sort:
ProvisionRetirementActionFieldSort!, $filter: [ProvisionFieldFilter!]) {
  provisionRetirementActions(sort: $sort, filter: $filter) {
    edges {
      cursor
      node {
        id
        definitionID
        status
        notes {
          requestedBy
          ticketNumber
          comment
        }
        manualEntry
        targetEndpointParameters {
          computerName
          serialNumber
          manufacturer
          model
        }
        endpointID
        ipAddress
        natIPAddress
        createdTime
        updatedTime
        eventLogs {
          edges {
            cursor
            node {
              id
              stepName
              status
              details
              actionID
              messageID
              createdTime
              endpointActionCode
            }
          }
          pageInfo {
            hasNextPage
            hasPreviousPage
            startCursor
            endCursor
          }
          totalRecords
        }
        createdBy {
          user {
            username
            id
            displayName
          }
          persona {
            id
            name
          }
        }
        updatedBy {
          user {
```

©2026 Tanium Inc. All Rights Reserved

```
            username
            id
            displayName
          }
          persona {
            id
            name
          }
        }
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the sort and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "sort": {
    "field": "computerName",
    "order": "desc"
  },
  "filter": {
    "value": "Win",
    "filterField": "computerName"
  }
}
```

Example response:

```
{
  "data": {
    "provisionRetirementActions": {
      "edges": [
        {
          "cursor": "cmV0aXJlbWVudEFjdGlvbjow",
          "node": {
            "id": "1",
            "definitionID": "12",
            "status": "canceled",
            "notes": {
              "requestedBy": "example requestor",
              "ticketNumber": "n/a",
              "comment": "canceled"
            },
            "manualEntry": false,
            "targetEndpointParameters": {
              "computerName": "example computer",
              "serialNumber": "1234567890abcdef",
              "manufacturer": "example manufacturer",
              "model": "example model"
            },
            "endpointID": "12345",
            "ipAddress": null,
            "natIPAddress": null,
```

©2026 Tanium Inc. All Rights Reserved

```
        "createdTime": "2024-02-22T19:58:28.193981Z",
        "updatedTime": "2024-02-22T20:18:03.226979Z",
        "eventLogs": {
          "edges": [],
          "pageInfo": {
            "hasNextPage": false,
            "hasPreviousPage": false,
            "startCursor": null,
            "endCursor": null
          },
          "totalRecords": 0
        },
        "createdBy": {
          "user": {
            "username": "admin",
            "id": "1",
            "displayName": null
          },
          "persona": {
            "id": "0",
            "name": null
          }
        },
        "updatedBy": {
          "user": {
            "username": "admin",
            "id": "1",
            "displayName": null
          },
          "persona": {
            "id": "0",
            "name": null
          }
        }
      }
    },
    {
      "cursor": "cmV0aXJlbWVudEFjdGlvbjox",
      "node": {
        "id": "2",
        "definitionID": "12",
        "status": "inProgress",
        "notes": {
          "requestedBy": "example requestor",
          "ticketNumber": "12345678",
          "comment": null
        },
        "manualEntry": false,
        "targetEndpointParameters": {
          "computerName": "example computer 2",
          "serialNumber": "VMware-12 34 56 78 90 ab cd ef-12 34 56 78 90 ab cd ef",
          "manufacturer": "VMware, Inc.",
          "model": "VMware20,1"
        },
        "endpointID": "23456",
        "ipAddress": "192.0.2.10:8443",
        "natIPAddress": "192.0.2.20",
        "createdTime": "2023-08-28T20:00:52.62558Z",
        "updatedTime": "2025-02-03T19:47:07.99349Z",
        "eventLogs": {
          "edges": [
            {
              "cursor": "ZXZlbnRsb2c6MA==",
              "node": {
```


©2026 Tanium Inc. All Rights Reserved

```
          "id": "1",
          "stepName": "begin-action",
          "status": "completed",
          "details": "",
          "actionID": "2",
          "messageID": "0",
          "createdTime": "2023-09-18T17:30:06.565055Z",
          "endpointActionCode": 0
        }
      },
      {
        "cursor": "ZXZlbnRsb2c6MQ==",
        "node": {
          "id": "2",
          "stepName": "Confirmation",
          "status": "issued",
          "details": "Retirement accepted, execution has started.",
          "actionID": "2",
          "messageID": "0",
          "createdTime": "2023-09-18T17:30:06.626969Z",
          "endpointActionCode": 4608
        }
      },
      {
        "cursor": "ZXZlbnRsb2c6Mg==",
        "node": {
          "id": "3",
          "stepName": "DeviceWipe",
          "status": "issued",
          "details": "Step started.",
          "actionID": "2",
          "messageID": "0",
          "createdTime": "2023-09-18T17:30:06.64245Z",
          "endpointActionCode": 512
        }
      },
      {
        "cursor": "ZXZlbnRsb2c6Mw==",
        "node": {
          "id": "4",
          "stepName": "DeviceWipe",
          "status": "information",
          "details": "Device wipe is in progress.",
          "actionID": "2",
          "messageID": "0",
          "createdTime": "2023-09-18T17:30:06.64245Z",
          "endpointActionCode": 17232
        }
      },
      {
        "cursor": "ZXZlbnRsb2c6NA==",
        "node": {
          "id": "5",
          "stepName": "DeviceWipe",
          "status": "information",
          "details": "Modifying standby and monitor timeout power settings.",
          "actionID": "2",
          "messageID": "0",
          "createdTime": "2023-09-18T17:30:06.64245Z",
          "endpointActionCode": 17248
        }
      },
      {
        "cursor": "ZXZlbnRsb2c6NQ==",
```


©2026 Tanium Inc. All Rights Reserved

```
            "node": {
              "id": "6",
              "stepName": "DeviceWipe",
              "status": "information",
              "details": "Verifying standby and monitor timeout power settings.",
              "actionID": "2",
              "messageID": "0",
              "createdTime": "2023-09-18T17:30:06.64245Z",
              "endpointActionCode": 17249
            }
          },
          {
            "cursor": "ZXZlbnRsb2c6Ng==",
            "node": {
              "id": "7",
              "stepName": "DeviceWipe",
              "status": "completed",
              "details": "Standby and monitor timeout power settings modified.",
              "actionID": "2",
              "messageID": "0",
              "createdTime": "2023-09-18T17:30:06.64245Z",
              "endpointActionCode": 17504
            }
          },
          {
            "cursor": "ZXZlbnRsb2c6Nw==",
            "node": {
              "id": "8",
              "stepName": "DeviceWipe",
              "status": "issued",
              "details": "Step resumed.",
              "actionID": "2",
              "messageID": "0",
              "createdTime": "2023-09-18T17:34:06.349452Z",
              "endpointActionCode": 513
            }
          },
          {
            "cursor": "ZXZlbnRsb2c6OA==",
            "node": {
              "id": "9",
              "stepName": "DeviceWipe",
              "status": "information",
              "details": "Device wipe is in progress.",
              "actionID": "2",
              "messageID": "0",
              "createdTime": "2023-09-18T17:34:06.365397Z",
              "endpointActionCode": 17232
            }
          }
        ],
        "pageInfo": {
          "hasNextPage": true,
          "hasPreviousPage": false,
          "startCursor": "ZXZlbnRsb2c6MA==",
          "endCursor": "ZXZlbnRsb2c6MTk="
        },
        "totalRecords": 27
      },
      "createdBy": {
        "user": {
          "username": "admin",
          "id": "1",
          "displayName": null
```


©2026 Tanium Inc. All Rights Reserved

```
            },
            "persona": {
              "id": "0",
              "name": null
            }
          },
          "updatedBy": {
            "user": {
              "username": "admin",
              "id": "1",
              "displayName": null
            },
            "persona": {
              "id": "0",
              "name": null
            }
          }
        }
      }
    ],
    "pageInfo": {
      "hasNextPage": false,
      "hasPreviousPage": false,
      "startCursor": "cmV0aXJlbWVudEFjdGlvbjow",
      "endCursor": "cmV0aXJlbWVudEFjdGlvbjox"
    },
    "totalRecords": 2
  }
 }
}
```


(1.0 - Experimental (Early Development)) Resolve retirement action

(mutation.provisionResolveRetirementAction)

About this task

The `provisionResolveRetirementAction` mutation requires Provision 1.19 or later and the **Provision Retirement Actions write** permission.

The following mutation resolves the retirement action that matches the ID `1`, updating the status from `in progress` to `completed`.

Note: This mutation has no effect if the retirement action status is not equal to `in progress`.

```
mutation exampleProvisionResolveRetirementAction($input: ID!) {
  provisionResolveRetirementAction(input: $input) {
    retirementAction {
      id
      definitionID
      status
```

©2026 Tanium Inc. All Rights Reserved

```
      eventLogs {
        edges {
          node {
            id
            stepName
            status
            details
            actionID
            messageID
            createdTime
            endpointActionCode
          }
        }
        totalRecords
      }
      notes {
        comment
        requestedBy
        ticketNumber
      }
      manualEntry
      ipAddress
      natIPAddress
      createdTime
      updatedTime
      createdBy {
        persona {
          id
          name
        }
        user {
          id
          username
          displayName
        }
      }
      updatedBy {
        persona {
          id
          name
        }
        user {
          id
          username
          displayName
        }
      }
    }
    error {
      message
    }
  }
}
```

Include the input variable in the **Variables** panel or in your variables dictionary:

```
{
  "input": 1
}
```

Example response:

©2026 Tanium Inc. All Rights Reserved

```
{
  "data": {
    "provisionResolveRetirementAction": {
      "retirementAction": {
        "id": "12",
        "definitionID": "1",
        "status": "completed",
        "eventLogs": {
          "edges": [],
          "totalRecords": 0
        },
        "notes": {
          "comment": null,
          "requestedBy": null,
          "ticketNumber": null
        },
        "manualEntry": true,
        "ipAddress": null,
        "natIPAddress": null,
        "createdTime": "2025-02-17T16:35:25.039883Z",
        "updatedTime": "2025-02-17T16:59:27.427305Z",
        "createdBy": {
          "persona": {
            "id": "0",
            "name": null
          },
          "user": {
            "id": "1",
            "username": "admin",
            "displayName": null
          }
        },
        "updatedBy": {
          "persona": {
            "id": "0",
            "name": null
          },
          "user": {
            "id": "1",
            "username": "admin",
            "displayName": null
          }
        }
      },
      "error": null
    }
  }
}
```


(1.0 - Experimental (Early Development)) Update retirement action notes

(mutation.provisionUpdateRetirementAction)

About this task

The `provisionUpdateRetirementAction` mutation requires Provision 1.19 or later and the **Provision Retirement Actions write** permission.

©2026 Tanium Inc. All Rights Reserved

The following mutation updates the `comment`, `requestedBy`, and `ticketNumber` values associated with the retirement action that matches the ID `12`.



Note: This mutation does not support updating other retirement action fields.

```
mutation exampleProvisionUpdateRetirementActionNotes($input:
ProvisionUpdateRetirementActionInput!) {
  provisionUpdateRetirementAction(input: $input) {
    retirementAction {
      id
      definitionID
      status
      eventLogs {
        edges {
          node {
            id
            stepName
            status
            details
            actionID
            messageID
            createdTime
            endpointActionCode
          }
        }
        totalRecords
      }
      notes {
        comment
        requestedBy
        ticketNumber
      }
      manualEntry
      ipAddress
      natIPAddress
      createdTime
      updatedTime
      createdBy {
        persona {
          id
          name
        }
        user {
          id
          username
          displayName
        }
      }
      updatedBy {
        persona {
          id
          name
        }
        user {
          id
          username
          displayName
        }
```

©2026 Tanium Inc. All Rights Reserved

```
      }
    }
    error {
      message
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "12",
    "notes": {
      "requestedBy": "updated requested by",
      "ticketNumber": null,
      "comment": "updated new comment"
    }
  }
}
```

Example response:

```
{
  "data": {
    "provisionUpdateRetirementAction": {
      "retirementAction": {
        "id": "12",
        "definitionID": "1",
        "status": "pending",
        "eventLogs": {
          "edges": [],
          "totalRecords": 0
        },
        "notes": {
          "comment": "updated new comment",
          "requestedBy": "updated requested by",
          "ticketNumber": null
        },
        "manualEntry": true,
        "ipAddress": null,
        "natIPAddress": null,
        "createdTime": "2025-02-17T15:44:51.771139Z",
        "updatedTime": "2025-02-17T15:48:06.389774Z",
        "createdBy": {
          "persona": {
            "id": "0",
            "name": null
          },
          "user": {
            "id": "1",
            "username": "admin",
            "displayName": null
          }
        },
        "updatedBy": {
          "persona": {
            "id": "0",
            "name": null
          },
          "user": {
```

©2026 Tanium Inc. All Rights Reserved

```
            "id": "1",
            "username": "admin",
            "displayName": null
          }
        }
      },
      "error": null
    }
  }
}
```

(1.0 - Experimental (Early Development)) Update retirement action definition

(mutation.provisionUpdateRetirementActionDefinition)

About this task

The `provisionUpdateRetirementActionDefinition` mutation requires Provision 1.19 or later and the **Provision Retirement Action Definitions write** permission.

Update the retirement action definition that matches an input ID. Configure retirement action options based on the following:

- If `userLock` is set to `true`, `userUnlock` must be set to `false`.
- If `userUnlock` is set to `true`, `userLock`, `deleteFiles`, and `bitlockerFreeze` must all be set to `false`.
- If `deleteFiles` is set to `true`, `userUnlock` must be set to `false`.

  Tip: For the best results, if `deleteFiles` is set to `true`, set `userLock` and `bitlockerFreeze` to `true`, and assign a value to `bitlockerMessage`.

- If `bitlockerFreeze` is set to `true`, `userUnlock` must be set to `false`, and you must assign a value to `bitlockerMessage`.

  Tip: For the best results, if `bitlockerFreeze` is set to `true`, set `userLock` to `true`.

©2026 Tanium Inc. All Rights Reserved

Note: Device retirement actions are supported on Windows endpoints only.

Note: Set `enabled` to `true` to allow users to create a retirement action using this retirement action definition.

The following mutation updates the retirement action definition that matches the ID `1` with different retirement action settings:

```
mutation exampleProvisionUpdateRetirementActionDefinition($input:
ProvisionUpdateRetirementDefinitionInput!) {
  provisionUpdateRetirementActionDefinition(input: $input) {
    actionDefinition {
      id
      name
      description
      endpointActionOptions {
        userLock
        userUnlock
        deleteFiles
        bitlockerFreeze
        bitlockerMessage
      }
      enabled
      deletedTime
      createdTime
      updatedTime
      createdBy {
        user {
          username
          id
          displayName
        }
        persona {
          name
          id
        }
      }
      updatedBy {
        user {
          username
          id
          displayName
        }
        persona {
          name
          id
        }
      }
      notesDefinition {
        requestedBy {
          notesRequired
          label
        }
        ticketNumber {
```

©2026 Tanium Inc. All Rights Reserved

```
          notesRequired
          label
        }
        comment {
          notesRequired
          label
        }
      }
      retirementActionsCount
    }
    error {
      message
      retryable
    }
  }
}
```

Include the input variables in the **Variables** panel or in your variables dictionary:

```
{
  "input": {
    "id": "1",
    "name": "Example retirement action definition update",
    "description": "example description update",
    "endpointActionOptions": {
      "userLock": true,
      "userUnlock": false,
      "deleteFiles": true,
      "bitlockerFreeze": true,
      "bitlockerMessage": "example BitLocker message"
    },
    "enabled": true,
    "notesDefinition": {
      "requestedBy": {
        "label": "requestedBy label new",
        "notesRequired": "REQUIRED"
      },
      "comment": {
        "label": "comment label new",
        "notesRequired": "OPTIONAL"
      },
      "ticketNumber": {
        "notesRequired": "REQUIRED",
        "label": "new label"
      }
    }
  }
}
```

Example response:

```
{
  "data": {
    "provisionUpdateRetirementActionDefinition": {
      "actionDefinition": {
        "id": "1",
        "name": "Example retirement action definition update",
        "description": "example description update",
        "endpointActionOptions": {
          "userLock": true,
          "userUnlock": false,
```

©2026 Tanium Inc. All Rights Reserved

```
          "deleteFiles": true,
          "bitlockerFreeze": true,
          "bitlockerMessage": "example BitLocker message"
        },
        "enabled": true,
        "deletedTime": null,
        "createdTime": "2025-02-17T15:10:36.447904Z",
        "updatedTime": "2025-02-17T15:15:50.940297Z",
        "createdBy": {
          "user": {
            "username": "admin",
            "id": "1",
            "displayName": null
          },
          "persona": {
            "name": null,
            "id": "0"
          }
        },
        "updatedBy": {
          "user": {
            "username": "admin",
            "id": "1",
            "displayName": null
          },
          "persona": {
            "name": null,
            "id": "0"
          }
        },
        "notesDefinition": {
          "requestedBy": {
            "notesRequired": "REQUIRED",
            "label": "requestedBy label new"
          },
          "ticketNumber": {
            "notesRequired": "REQUIRED",
            "label": "new label"
          },
          "comment": {
            "notesRequired": "OPTIONAL",
            "label": "comment label new"
          }
        },
        "retirementActionsCount": 0
      },
      "error": null
    }
  }
}
```


Reporting examples

The following queries and mutation require Reporting and retrieve reports or report results, export and import report definitions, and delete reports and Reporting dashboards.

©2026 Tanium Inc. All Rights Reserved

Delete a report by ID

(mutation.reportDelete)

About this task

The `reportDelete` mutation requires Reporting 1.32 or later and the **Report** write permission.

The following mutation deletes the report that matches the referenced ID.

```
mutation exampleDeleteReport($id: ID!) {
  reportDelete(ref: {id: $id}) {
    id
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the report ID variable in the **Variables** panel or in your variables dictionary.

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef"
}
```

Example response:

```
{
  "data": {
    "reportDelete": {
      "id": "12345678-90ab-cdef-1234-567890abcdef",
      "error": null
    }
  }
}
```

Delete a Reporting dashboard by ID

(mutation.reportingDashboardDelete)

About this task

The `reportingDashboardDelete` mutation requires Reporting 1.32 or later and the **Dashboard** write permission.

The following mutation deletes the Reporting dashboard that matches the referenced ID.

```
mutation exampleDeleteReportingDashboard($id: ID!) {
```

©2026 Tanium Inc. All Rights Reserved

```
  reportingDashboardDelete(ref: {id: $id}) {
    id
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the Reporting dashboard ID variable in the **Variables** panel or in your variables dictionary.

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef"
}
```

Example response:

```
{
  "data": {
    "reportingDashboardDelete": {
      "id": "12345678-90ab-cdef-1234-567890abcdef",
      "error": null
    }
  }
}
```

Get a report by ID

(query.report)

About this task

The following query retrieves the report that matches the referenced ID.

```
query exampleGetReportByID($id: ID!) {
  report(ref: {id: $id}) {
    id
    author {
      user {
        id
      }
      persona {
        id
      }
    }
    contentSet {
      id
    }
    createdTime
    description
    favorite
    labels
    lastModifiedBy {
      user {
        id
      }
```

©2026 Tanium Inc. All Rights Reserved

```
      persona {
        id
      }
    }
    modifiedTime
    moduleName
    name
    viewDetails {
      columns {
        name
      }
    }
  }
}
```

Include the report ID variable in the **Variables** panel or in your variables dictionary.

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef"
}
```

Example response:

```
{
  "data": {
    "report": {
      "id": "12345678-90ab-cdef-1234-567890abcdef",
      "author": {
        "user": {
          "id": "1"
        },
        "persona": {
          "id": "0"
        }
      },
      "contentSet": {
        "id": "21"
      },
      "createdTime": "2022-01-01T05:33:35.838686969Z",
      "description": "",
      "favorite": false,
      "labels": null,
      "lastModifiedBy": {
        "user": {
          "id": "1"
        },
        "persona": {
          "id": "0"
        }
      },
      "modifiedTime": "2022-06-01T05:33:35.838686969Z",
      "moduleName": null,
      "name": "Example report",
      "viewDetails": {
        "columns": [
          {
            "name": "Example report column"
          }
        ]
      }
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Get an ID of a report

(query.reports)

About this task

The following query retrieves reports that have the text "Example report" in the name or description, and returns the ID for each report.

```
query exampleGetReportIDs($first: Int, $text: String) {
  reports(first: $first, filter: {text: $text}) {
    edges {
      node {
        id
        description
        name
      }
    }
  }
}
```

Include the pagination and text variables in the **Variables** panel or in your variables dictionary.

```
{
  "first": 2,
  "text": "Example report"
}
```

Example response:

```
{
  "data": {
    "reports": {
      "edges": [
        {
          "node": {
            "id": "12345678-90ab-cdef-1234-567890abcdef",
            "description": "Example description",
            "name": "Example report"
          }
        },
        {
          "node": {
            "id": "abcdef12-3456-7890-abcd-ef1234567890",
            "description": "Example report 2",
            "name": "Report example"
          }
        }
      ]
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Get reports

(query.reports)

About this task

The following query retrieves the first 2 reports.

```
query exampleGetReports($first: Int) {
  reports(first: $first) {
    edges {
      node {
        id
        author {
          user {
            id
          }
          persona {
            id
          }
        }
        contentSet {
          id
        }
        createdTime
        description
        favorite
        labels
        lastModifiedBy {
          user {
            id
          }
          persona {
            id
          }
        }
        modifiedTime
        moduleName
        name
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary.

```
{
  "first": 2
}
```

Example response:

```
{
  "data": {
    "reports": {
      "edges": [
        {
```

©2026 Tanium Inc. All Rights Reserved

```
        "node": {
          "id": "12345678-90ab-cdef-1234-567890abcdef",
          "author": {
            "user": {
              "id": "1"
            },
            "persona": {
              "id": "0"
            }
          },
          "contentSet": {
            "id": "21"
          },
          "createdTime": "2022-01-01T05:33:35.838686969Z",
          "description": "Example description",
          "favorite": false,
          "labels": null,
          "lastModifiedBy": {
            "user": {
              "id": "1"
            },
            "persona": {
              "id": "0"
            }
          },
          "modifiedTime": "2022-03-17T05:33:35.838686969Z",
          "moduleName": null,
          "name": "Example report"
        }
      },
      {
        "node": {
          "id": "abcdef12-3456-7890-abcd-ef1234567890",
          "author": {
            "user": {
              "id": "2"
            },
            "persona": {
              "id": "1"
            }
          },
          "contentSet": {
            "id": "123"
          },
          "createdTime": "2022-01-01T19:58:08.175603337Z",
          "description": "Example report 2",
          "favorite": false,
          "labels": null,
          "lastModifiedBy": {
            "user": {
              "id": "3"
            },
            "persona": {
              "id": "0"
            }
          },
          "modifiedTime": "2022-06-01T19:58:08.175603337Z",
          "moduleName": null,
          "name": "Example report 2"
        }
      }
    ]
  }
}
```


©2026 Tanium Inc. All Rights Reserved

```
}
```

Get report results by report ID

(query.reportResultData)

About this task

Using `reportResultData` in a request requires:

- Reporting 1.12.144 or later to submit the query
- Reporting 1.35.36 or later to use `query.reportResultData (summarize)` and `query.reportResultData.edges.node.count` to retrieve row counts in the response

Important: The cursor refresh functionality (`query.reportResultData.collectionInfo.startCursor`, `query.reportResultData (refresh)`) is deprecated and no longer functions. If a query request using `query.reportResultData` fails, resubmit the original query request at least 60 seconds after the initial failed response.

The following query retrieves column `values` for the first 2 report results, and summarizes the results by returning a `count` of duplicate rows. The query also retrieves report definition details, including report column `name`, `sourceName`, and `sourceColumnName`.

```
query exampleGetReportResultData($id: ID!, $first: Int, $summarize: Boolean!) {
  reportResultData(id: $id, first: $first, summarize: $summarize) {
    edges {
      node {
        columns {
          values
        }
        count
      }
    }
    viewDetails {
      columns {
        name
        sourceName
        sourceColumnName
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the ID, pagination, and summary variables in the **Variables** panel or in your variables dictionary.

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "first": 2,
  "summarize": true
}
```

Example response:


```
{
  "data": {
    "reportResultData": {
      "edges": [
        {
          "node": {
            "columns": [
              {
                "values": [
                  "Npcap OEM"
                ]
              },
              {
                "values": [
                  "1.10"
                ]
              },
              {
                "values": [
                  "Windows 10 Enterprise"
                ]
              }
            ],
            "count": 10
          }
        },
        {
          "node": {
            "columns": [
              {
                "values": [
                  "Microsoft Visual C++ 2013 Redistributable (x86) - 12.0.21005"
                ]
              },
              {
                "values": [
                  "12.0.21005.1"
                ]
              },
              {
                "values": [
                  "Windows 10 Enterprise"
                ]
              }
            ],
            "count": 10
          }
        }
      ],
      "viewDetails": {
```


©2026 Tanium Inc. All Rights Reserved

```
        "columns": [
          {
            "name": "Name",
            "sourceName": "Installed Applications",
            "sourceColumnName": "Name"
          },
          {
            "name": "Version",
            "sourceName": "Installed Applications",
            "sourceColumnName": "Version"
          },
          {
            "name": "Operating System",
            "sourceName": "Operating System",
            "sourceColumnName": "Operating System"
          }
        ]
      }
    }
  }
}
```

For reports with multiple results, you can retrieve a page of data and a page end cursor, then submit your request, defining `after` in the input equal to the page end cursor to retrieve the next page of results. You can repeat this until `hasNextPage` equals `false` in the response, signifying that no more pages of data exist.

The following query retrieves column `values` for the first 2 report results, and summarizes the results by returning a `count` of duplicate rows. This query also retrieves `hasNextPage`, to determine whether at least one more page of data exists, and `endCursor`, to pass this cursor value in a separate request and retrieve the next page of results.

```
query exampleGetPaginatedResultData($id: ID!, $first: Int, $summarize: Boolean!) {
  reportResultData(id: $id, first: $first, summarize: $summarize) {
    edges {
      node {
        columns {
          values
        }
        count
      }
    }
    viewDetails {
      columns {
        name
        sourceName
        sourceColumnName
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the ID, pagination, and summary variables in the **Variables** panel or in your variables dictionary.

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "first": 2,
  "summarize": true
}
```

Example response:

```
{
  "data": {
    "reportResultData": {
      "edges": [
        {
          "node": {
            "columns": [
              {
                "values": [
                  "example-endpoint-1"
                ]
              },
              {
                "values": [
                  "192.0.2.10"
                ]
              }
            ],
            "count": 1
          }
        },
        {
          "node": {
            "columns": [
              {
                "values": [
                  "example-endpoint-2"
                ]
              },
              {
                "values": [
                  "192.0.2.20"
                ]
              }
            ],
            "count": 1
          }
        }
      ],
      "viewDetails": {
        "columns": [
          {
            "name": "Computer Name",
            "sourceName": "Computer Name",
            "sourceColumnName": "Computer Name"
          },
          {
            "name": "IP Address",
            "sourceName": "IP Address",
            "sourceColumnName": "IP Address"
```

©2026 Tanium Inc. All Rights Reserved

```
          }
        ]
      },
      "pageInfo": {
        "hasPreviousPage": false,
        "hasNextPage": true,
        "startCursor": "MjMwNDUzOTM6MA==",
        "endCursor": "MjMwNDUzOTM6MQ=="
      }
    }
  }
}
```

The following query defines `after` in the input equal to the prior `endCursor` value of `MjMwNDUzOTM6MQ==` to retrieve the next page of results.

```
query exampleGetPaginatedResultData($id: ID!, $first: Int, $after: Cursor, $summarize:
Boolean!) {
  reportResultData(id: $id, first: $first, after: $after, summarize: $summarize) {
    edges {
      node {
        columns {
          values
        }
        count
      }
    }
    viewDetails {
      columns {
        name
        sourceName
        sourceColumnName
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
  }
}
```

Include the ID, pagination, and summary variables in the **Variables** panel or in your variables dictionary.

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "first": 2,
  "after": "MjMwNDUzOTM6MQ==",
  "summarize": true
}
```

Example response:

```
{
  "data": {
    "reportResultData": {
```

©2026 Tanium Inc. All Rights Reserved

```
      "edges": [
        {
          "node": {
            "columns": [
              {
                "values": [
                  "example-endpoint-3"
                ]
              },
              {
                "values": [
                  "192.0.2.30"
                ]
              }
            ],
            "count": 1
          }
        },
        {
          "node": {
            "columns": [
              {
                "values": [
                  "example-endpoint-4"
                ]
              },
              {
                "values": [
                  "192.0.2.40"
                ]
              }
            ],
            "count": 1
          }
        }
      ],
      "viewDetails": {
        "columns": [
          {
            "name": "Computer Name",
            "sourceName": "Computer Name",
            "sourceColumnName": "Computer Name"
          },
          {
            "name": "IP Address",
            "sourceName": "IP Address",
            "sourceColumnName": "IP Address"
          }
        ]
      },
      "pageInfo": {
        "hasPreviousPage": true,
        "hasNextPage": true,
        "startCursor": "MjMwNDUzOTM6Mg==",
        "endCursor": "MjMwNDUzOTM6Mw=="
      }
    }
  }
}
```


©2026 Tanium Inc. All Rights Reserved

Export a report by ID

(query.reportExport)

About this task

The following query exports the report definition that matches the referenced ID. You can import this definition using the `importReport` mutation.

```
query exampleExportReportDefinition($id: ID!) {
  reportExport(ref: {id: $id}) {
    reportDefinition
  }
}
```

Include the report ID variable in the **Variables** panel or in your variables dictionary.

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef"
}
```

Example response:


```
{
  "data": {
    "reportExport": {
      "reportDefinition": "{\"id\":\"12345678-90ab-cdef-1234-567890abcdef\",
\"displayName\":\"Example Report\", \"description\":\"Example API Gateway Reporting
functionality\", \"contentSet\":{\"name\":\"Reporting\"}, \"sourceView\":{\"id\":\"source
data\", \"querySpec\":{\"hasChanged\":true, \"sources\":{\"Computer Name\":{\"id\":\"Computer
Name\", \"sensor\":{\"sensorName\":\"Computer Name\", \"flatten\":true}}, \"IP
Address\":{\"id\":\"IP Address\", \"sensor\":{\"sensorName\":\"IP Address\",
\"flatten\":false}}}, \"selects\":[{\"id\":\"1\", \"sourceColumn\":{\"sourceId\":\"Computer
Name\", \"columnName\":\"Computer Name\"}, \"displayName\":\"Computer Name\",
\"display\":{}}, {\"id\":\"2\", \"sourceColumn\":{\"sourceId\":\"IP Address\",
\"columnName\":\"IP Address\"}, \"displayName\":\"IP Address\", \"display\":{}}]},
\"summary\":{\"countId\":\"summary_count_id\", \"displayName\":\"Count\"}},
\"display\":{\"advanced_filters_enabled\":true, \"available_additional_filters\":[],
\"available_quick_filters\":[\"quick:base:computerGroup\", \"quick:base:endpointLastSeen\",
\"quick:base:os\"], \"version\":1}, \"categories\":[{\"name\":\"Test\"}],
\"meta\":{\"class\":\"USER\", \"type\":\"V2\"}}"
    }
  }
}
```


Import a report definition

(mutation.reportImport)

About this task

The following query imports a report definition that you exported using the `reportExport` query.

©2026 Tanium Inc. All Rights Reserved

```
mutation exampleImportReportDefinition($allowDuplicate: Boolean!, $reportDefinition: String!)
{
  reportImport(
    input: {allowDuplicate: $allowDuplicate, reportDefinition: $reportDefinition}
  ) {
    report {
      id
    }
    error {
      message
      retryable
      timedOut
    }
  }
}
```

Include the report ID variable in the **Variables** panel or in your variables dictionary.

```
{
  "allowDuplicate": false,
  "reportDefinition": "{\"id\":\"12345678-90ab-cdef-1234-567890abcdef\",
\"displayName\":\"Example report import\", \"description\":\"Example import functionality\",
\"contentSet\":{\"name\":\"Reporting\"}, \"sourceView\":{\"id\":\"source data\",
\"querySpec\":{\"hasChanged\":true, \"sources\":{\"Computer Name\":{\"id\":\"Computer Name\",
\"sensor\":{\"sensorName\":\"Computer Name\", \"flatten\":true}}, \"IP Address\":{\"id\":\"IP
Address\", \"sensor\":{\"sensorName\":\"IP Address\", \"flatten\":false}}},
\"selects\":[{\"id\":\"1\", \"sourceColumn\":{\"sourceId\":\"Computer Name\",
\"columnName\":\"Computer Name\"}, \"displayName\":\"Computer Name\", \"display\":{}},
{\"id\":\"2\", \"sourceColumn\":{\"sourceId\":\"IP Address\", \"columnName\":\"IP Address\"},
\"displayName\":\"IP Address\", \"display\":{}}]},
\"summary\":{\"countId\":\"summary_count_id\", \"displayName\":\"Count\"}},
\"display\":{\"advanced_filters_enabled\":true, \"available_additional_filters\":[],
\"available_quick_filters\":[\"quick:base:computerGroup\", \"quick:base:endpointLastSeen\",
\"quick:base:os\"], \"version\":1}, \"categories\":[{\"name\":\"Test\"}],
\"meta\":{\"class\":\"USER\", \"type\":\"V2\"}}"
}
```

Example response:

```
{
  "data": {
    "reportImport": {
      "report": {
        "id": "abcdef12-3456-7890-abcd-ef1234567890"
      },
      "error": null
    }
  }
}
```

Retrieve report results collection start cursor

(query.reportResultData.collectionInfo.startCursor, query.reportResultData (refresh))

About this task

Important: The cursor refresh functionality (`query.reportResultData.collectionInfo.startCursor, query.reportResultData (refresh)`) is deprecated and no longer functions. If a query request using `query.reportResultData` fails, resubmit the original query request at least 60 seconds after the initial failed response.

The following query retrieves report details. The query requests `collectionInfo.startCursor`, which is populated in the response only if the data is not ready within 30 seconds. Use the `collectionInfo.startCursor` value for refreshing the request.

```
query exampleGetReportResultCursor($id: ID!, $first: Int) {
  reportResultData(id: $id, first: $first) {
    edges {
      node {
        columns {
          values
        }
      }
    }
    collectionInfo {
      active
      startCursor
    }
  }
}
```

Include the pagination and report ID variables in the **Variables** panel or in your variables dictionary.

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "first": 2
}
```

Example response for an ongoing collection with a `startCursor` value, meaning that the data is not yet ready to be returned:

```
{
  "data": {
    "reportResultData": {
      "edges": [],
      "collectionInfo": {
        "active": true,
        "startCursor": "4t2cB0ZGNPqy4J4P7O7opwz/24/5zgPi6OinDAA=:0"
      }
    }
  }
}
```

Refresh report results collection using starting cursor

The following query refreshes the collection using the returned `startCursor` value:

©2026 Tanium Inc. All Rights Reserved

```
query exampleRefreshReportResultCursor($id: ID!, $first: Int, $refresh: Cursor) {
  reportResultData(id: $id, first: $first, refresh: $refresh) {
    edges {
      node {
        columns {
          values
        }
      }
    }
    collectionInfo {
      active
      startCursor
    }
  }
}
```

Include the pagination and report ID variables in the **Variables** panel or in your variables dictionary:

```
{
  "id": "12345678-90ab-cdef-1234-567890abcdef",
  "first": 2,
  "refresh": "4t2cB0ZGNPqy4J4P7O7opwz/24/5zgPi6OinDAA=:0"
}
```

Example response with a completed collection:

```
{
  "data": {
    "reportResultData": {
      "edges": [
        {
          "node": {
            "columns": [
              {
                "values": [
                  "Example computer 1"
                ]
              },
              {
                "values": [
                  "2001:DB8::1234",
                  "192.0.2.10"
                ]
              }
            ]
          }
        },
        {
          "node": {
            "columns": [
              {
                "values": [
                  "Example computer 2"
                ]
              },
              {
                "values": [
                  "2001:DB8::1234",
                  "192.0.2.20"
                ]
              }
```

©2026 Tanium Inc. All Rights Reserved

```
            ]
          }
        }
      ],
      "collectionInfo": {
        "active": false,
        "startCursor": null
      }
    }
  }
}
```

Sensor examples

The following queries and mutation retrieve sensors and sensor parameters, and register a sensor for harvest.

Get normal sensors

(query.sensors)

Normal sensors are scripts that run on endpoints to compute a response to a Tanium question. This data can be sourced from either the Tanium Server or Tanium Data Service if a sensor is being harvested. The following query retrieves the first 5 normal sensors. For more information about normal sensors, see Tanium Console User Guide: Managing sensors.

```
query exampleGetNormalSensors($first: Int, $path: String, $op: FieldFilterOp!, $value:
String) {
  sensors(first: $first, filter:{path: $path, op: $op, value: $value}) {
    edges {
      node {
        contentSetName
        description
        endpointQueryPaths
        harvested
        lastUpdatedBy
        name
        parameterizations {
          harvested
          values {
            name
            value
          }
        }
        parameters {
          defaultValue
          name
        }
        virtual
      }
    }
    pageInfo {
      endCursor
      startCursor
```

©2026 Tanium Inc. All Rights Reserved

```
      hasNextPage
      hasPreviousPage
    }
  }
}
```

Include pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 5,
  "path": "virtual",
  "op": "EQ",
  "value": "false"
}
```

Example response:

```
{
  "data": {
    "sensors": {
      "edges": [
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "The full Active Directory distinguished name for the computer\
nExample: CN=Win8-test5,CN=Computers,DC=corp,DC=com",
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "AD Distinguished Name",
            "parameterizations": null,
            "parameters": null,
            "virtual": false
          }
        },
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "The Active Directory domain name (if any) that the computer is
joined to.\nExample: intra.company.com",
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "AD Domain",
            "parameterizations": null,
            "parameters": null,
            "virtual": false
          }
        },
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "Returns the name of the Active Directory Forest that a machine is
a member of.  This may produce the same value that the Sensor named AD Domain
produces.\nExample: corp.domain.com",
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "AD Forest",
            "parameterizations": null,
            "parameters": null,
```

©2026 Tanium Inc. All Rights Reserved

```
            "virtual": false
          }
        },
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "The Active Directory organizational unit (OU) where the machine
is located.\nExample: CN=Computers,DC=corp,DC=com",
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "AD Organizational Unit",
            "parameterizations": null,
            "parameters": null,
            "virtual": false
          }
        },
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "Returns the short, NetBIOS name of a machine's domain.\nExample:
CORP",
            "endpointQueryPaths": [],
            "harvested": true,
            "lastUpdatedBy": "taniumconsole",
            "name": "AD Short Domain",
            "parameterizations": null,
            "parameters": null,
            "virtual": false
          }
        }
      ],
      "pageInfo": {
        "endCursor": "NTc2NTM4MDoxOQ==",
        "startCursor": "NTc2NTM4MDow",
        "hasNextPage": true,
        "hasPreviousPage": false
      }
    }
  }
}
```

Get virtual sensors

(query.sensors)

Tanium solutions can insert data into Tanium Data Service (TDS) directly, in contrast to a normal sensor running a script on an endpoint to collect data. This data is sourced from TDS using virtual sensors. The following query retrieves the first 5 virtual sensors registered in TDS. For more information about virtual sensors, see Tanium Reporting User Guide: Data sources.

```
query exampleGetVirtualSensors($first: Int, $path: String, $op: FieldFilterOp!, $value:
String) {
  sensors(first: $first, filter:{path: $path, op: $op, value: $value}) {
    edges {
      node {
        contentSetName
        description
```

©2026 Tanium Inc. All Rights Reserved

```
        endpointQueryPaths
        harvested
        lastUpdatedBy
        name
        parameterizations {
          harvested
          values {
            name
            value
          }
        }
        parameters {
          defaultValue
          name
        }
        virtual
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 5,
  "path": "virtual",
  "op": "EQ",
  "value": "true"
}
```

Example response:

```
{
  "data": {
    "sensors": {
      "edges": [
        {
          "node": {
            "contentSetName": "Tanium Data Service",
            "description": null,
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "Computer Groups",
            "parameterizations": null,
            "parameters": null,
            "virtual": true
          }
        },
        {
          "node": {
            "contentSetName": "Tanium Data Service",
            "description": null,
            "endpointQueryPaths": [],
            "harvested": false,
```

©2026 Tanium Inc. All Rights Reserved

```
            "lastUpdatedBy": "taniumconsole",
            "name": "EID First Seen",
            "parameterizations": null,
            "parameters": null,
            "virtual": true
          }
        },
        {
          "node": {
            "contentSetName": "Tanium Data Service",
            "description": null,
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "EID Last Changed",
            "parameterizations": null,
            "parameters": null,
            "virtual": true
          }
        },
        {
          "node": {
            "contentSetName": "Tanium Data Service",
            "description": null,
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "EID Last Seen",
            "parameterizations": null,
            "parameters": null,
            "virtual": true
          }
        },
        {
          "node": {
            "contentSetName": "Tanium Data Service",
            "description": null,
            "endpointQueryPaths": [
              [
                "id"
              ]
            ],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "Endpoint ID",
            "parameterizations": null,
            "parameters": null,
            "virtual": true
          }
        }
      ],
      "pageInfo": {
        "startCursor": "NTc2NTM4MDow",
        "endCursor": "NTc2NTM4MDoxOQ==",
        "hasPreviousPage": false,
        "hasNextPage": true
      }
    }
  }
}
```


©2026 Tanium Inc. All Rights Reserved

Get sensors available to Tanium Data Service

(query.sensors)

The following query retrieves virtual sensors registered in TDS and sensors registered for harvest in TDS. For more information about virtual sensors, see Tanium Reporting User Guide: Data sources.

```
query exampleGetTDSData($filter: FieldFilter) {
  sensors(
    filter: $filter
  ) {
    edges {
      node {
        contentSetName
        description
        endpointQueryPaths
        harvested
        lastUpdatedBy
        name
        parameterizations {
          harvested
          values {
            name
            value
          }
        }
        parameters {
          defaultValue
          name
        }
        virtual
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the filter variable in the **Variables** panel or in your variables dictionary:

```
{
  "filter":
{"any":true,"filters":[{"path":"virtual","op":"EQ","negated":false,"value":"true"},{"path":"h
arvested","op":"EQ","negated":false,"value":"true"}]}
}
```

Example response:

```
{
  "data": {
    "sensors": {
      "edges": [
        {
```

©2026 Tanium Inc. All Rights Reserved

```
          "node": {
            "contentSetName": "Core Content",
            "description": "Returns the name of the Active Directory Forest that a machine is
a member of.  This may produce the same value that the Sensor named AD Domain
produces.\nExample: corp.domain.com",
            "endpointQueryPaths": [],
            "harvested": true,
            "lastUpdatedBy": "taniumconsole",
            "name": "AD Forest",
            "parameterizations": null,
            "parameters": null,
            "virtual": false
          }
        },
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "The Active Directory organizational unit (OU) where the machine
is located.\nExample: CN=Computers,DC=corp,DC=com",
            "endpointQueryPaths": [],
            "harvested": true,
            "lastUpdatedBy": "taniumconsole",
            "name": "AD Organizational Unit",
            "parameterizations": null,
            "parameters": null,
            "virtual": false
          }
        }
      ],
      "pageInfo": {
        "startCursor": "c2Vuc29yczow",
        "endCursor": "c2Vuc29yczozMjM=",
        "hasPreviousPage": false,
        "hasNextPage": false
      }
    }
  }
}
```


Get hidden sensors

(query.sensors)

Hidden sensors are used by the Tanium Core Platform, Tanium solutions, or Tanium shared services. By default, `query.sensors` requests do not return hidden sensors. You can retrieve hidden sensors in the response using the `includeHidden` argument on `query.sensors`.



Note: Using or updating hidden sensors might negatively impact Tanium functionality.

The following query retrieves the first `2` hidden sensors, using the `(includeHidden: true)` argument and the `(filter: {path: "hidden", "op": EQ, "value": true})` filter argument:

©2026 Tanium Inc. All Rights Reserved

```
query exampleGetOnlyHiddenSensors($first: Int, $hidden: Boolean! $path: String, $op:
FieldFilterOp!, $value: String) {
  sensors(
    first: $first
    includeHidden: $hidden
    filter: {path: $path, op: $op, value: $value}
  ) {
    edges {
      node {
        hidden
        contentSetName
        description
        endpointQueryPaths
        harvested
        lastUpdatedBy
        name
        category
        maxAgeSeconds
        valueType
        created
        updated
      }
    }
    pageInfo {
      endCursor
      startCursor
      hasNextPage
      hasPreviousPage
    }
  }
}
```

Include pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "hidden": true,
  "path": "hidden",
  "op": "EQ",
  "value": "true"
}
```

Example response:

```
{
  "data": {
    "sensors": {
      "edges": [
        {
          "node": {
            "hidden": true,
            "contentSetName": "Health Check Content Set",
            "description": "The serial number, if available, provided by the computer
manufacturer.\nExample: 123ABC1",
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "Health Check - Computer Serial Number",
            "category": "Health Check",
            "maxAgeSeconds": 86400,
            "valueType": "String",
```

©2026 Tanium Inc. All Rights Reserved

```
            "created": null,
            "updated": "2023-01-11T18:48:23Z"
          }
        },
        {
          "node": {
            "hidden": true,
            "contentSetName": "Health Check Content Set",
            "description": "System unique identifier UUID.\nExample:
3e6be9de-8139-11d1-9106-a43f08d823a6",
            "endpointQueryPaths": [],
            "harvested": false,
            "lastUpdatedBy": "taniumconsole",
            "name": "Health Check - System UUID",
            "category": "Health Check",
            "maxAgeSeconds": 86400,
            "valueType": "String",
            "created": null,
            "updated": "2023-04-24T21:26:25Z"
          }
        }
      ],
      "pageInfo": {
        "endCursor": "c2Vuc29yczozOQ==",
        "startCursor": "c2Vuc29yczoyMA==",
        "hasNextPage": true,
        "hasPreviousPage": true
      }
    }
  }
}
```

The following request retrieves both hidden and non-hidden sensors, using the `(includeHidden: true)` argument:

```
query exampleGetHiddenAndNonHiddenSensors($first: Int, $hidden: Boolean!) {
  sensors(
    first: $first
    includeHidden: $hidden
  ) {
    edges {
      node {
        hidden
        contentSetName
        description
        endpointQueryPaths
        harvested
        lastUpdatedBy
        name
        category
        maxAgeSeconds
        valueType
        created
        updated
      }
    }
    pageInfo {
      endCursor
      startCursor
      hasNextPage
      hasPreviousPage
    }
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Include pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "hidden": true,
}
```

Get sensor parameters (query.getSensorParameters)

GET SENSOR PARAMETERS

The following query retrieves sensors whose description contains the text `custom tag`, and returns both the sensor's parameters and sensor parameterizations, or sets of parameterized values harvested within Tanium Data Service or otherwise registered.

```
query exampleGetSensorParameters($path: String, $op: FieldFilterOp!, $value: String) {
  sensors(filter: {path: $path, op: $op, value: $value}) {
    edges {
      node {
        contentSetName
        description
        lastUpdatedBy
        name
        parameterizations {
          harvested
          values {
            name
            value
          }
        }
        parameters {
          defaultValue
          name
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "path": "description",
  "op": "CONTAINS",
  "value": "custom tag"
}
```

©2026 Tanium Inc. All Rights Reserved

Example response:


```
{
  "data": {
    "sensors": {
      "edges": [
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "Checks to see if a given custom tag exists on the endpoint. The
input can either be a substring or an exact match, and the check is case
insensitive.\nExample: True",
            "lastUpdatedBy": "taniumconsole",
            "name": "Custom Tag Exists",
            "parameterizations": [
              {
                "harvested": true,
                "values": [
                  {
                    "name": "exactMatch",
                    "value": "1"
                  },
                  {
                    "name": "tag",
                    "value": "testCustomTag"
                  }
                ]
              },
              {
                "harvested": true,
                "values": [
                  {
                    "name": "exactMatch",
                    "value": "0"
                  },
                  {
                    "name": "tag",
                    "value": "testedTag"
                  }
                ]
              },
              {
                "harvested": true,
                "values": [
                  {
                    "name": "exactMatch",
                    "value": "1"
                  },
                  {
                    "name": "tag",
                    "value": "customTagApplied"
                  }
                ]
              }
            ],
            "parameters": [
              {
                "defaultValue": "1",
                "name": "tag"
              },
              {
                "defaultValue": "1",
                "name": "exactMatch"
```


©2026 Tanium Inc. All Rights Reserved

```
              }
            ]
          }
        },
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "Any specified custom tags that have been set for this machine.
See the Custom Tagging Dashboard.\nExample:  Development, Test-Machines",
            "lastUpdatedBy": "taniumconsole",
            "name": "Custom Tags",
            "parameterizations": null,
            "parameters": null
          }
        }
      ],
      "pageInfo": {
        "startCursor": "NTc2NTM4MDow",
        "endCursor": "NTc2NTM4MDoxOQ==",
        "hasPreviousPage": false,
        "hasNextPage": false
      }
    }
  }
}
```

GET SENSOR PARAMETERS SHOWING HIDDEN STATUS

The following query retrieves columns associated with the Application Crashes in Last X Days sensor, and includes whether the column is hidden or not hidden.

```
query exampleGetSensorParametersHidden($path: String, $op: FieldFilterOp!, $value: String) {
  sensors(
    filter: {path: $path, value: $value, op: $op}
  ) {
    edges {
      node {
        contentSetName
        description
        lastUpdatedBy
        name
        columns {
          name
          hidden
          ignoreCase
          valueType
        }
        parameterizations {
          harvested
          values {
            name
            value
          }
        }
        parameters {
          defaultValue
          name
        }
      }
    }
```

©2026 Tanium Inc. All Rights Reserved

```
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "path": "name",
  "op": "EQ",
  "value": "Application Crashes in Last X Days"
}
```

Example response:

```
{
  "data": {
    "sensors": {
      "edges": [
        {
          "node": {
            "contentSetName": "Core Content",
            "description": "A parameterized Sensor that queries for any processes that have
crashed in the last X days.\nExample: chrome.exe",
            "lastUpdatedBy": "taniumconsole",
            "name": "Application Crashes in Last X Days",
            "columns": [
              {
                "name": "Process",
                "hidden": false,
                "ignoreCase": true,
                "valueType": ""
              },
              {
                "name": "Date",
                "hidden": true,
                "ignoreCase": true,
                "valueType": ""
              }
            ],
            "parameterizations": [
              {
                "harvested": true,
                "values": [
                  {
                    "name": "days",
                    "value": "5"
                  }
                ]
              }
            ],
            "parameters": [
              {
                "defaultValue": "5",
                "name": "days"
              }
            ]
```

©2026 Tanium Inc. All Rights Reserved

```
          }
        }
      ],
      "pageInfo": {
        "startCursor": "c2Vuc29yczow",
        "endCursor": "c2Vuc29yczow",
        "hasPreviousPage": false,
        "hasNextPage": false
      }
    }
  }
}
```

Register sensor for harvest (mutation.sensorHarvest)

REGISTER A SENSOR FOR HARVEST

The following mutation registers a sensor for harvest by Tanium Data Service.



Note: Wait 10 minutes after you create a sensor before submitting this mutation request.

Registration takes several minutes. The first time you submit this request, the response may contain a cursor, indicating an ongoing registration. If your response contains a cursor, wait several minutes, then pass the cursor as a `sensorHarvest` argument in the request to retrieve a success message or error message.



Best Practice: For the best results, pass the `integrationName` value in the format `<companyName>-<productIntegration>`, where `<companyName>` represents the company responsible for the integration, and `<productIntegration>` is the integration name or integration purpose.

```
mutation exampleHarvestSensor($harvest: Boolean!, $integrationName: String!, $name: String!)
{
  sensorHarvest(
    input: {harvest: $harvest, integrationName: $integrationName, name: $name}
  ) {
    cursor
    error {
      message
      retryable
      timedOut
    }
    success
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the harvest, integration name, and sensor name variables in the **Variables** panel or in your variables dictionary:

```
{
  "harvest": true,
  "integrationName": "examplecompany-integratetanium",
  "name": "test-sensor"
}
```

Example response with a cursor:

```
{
  "data": {
    "sensorHarvest": {
      "cursor": "NTc2NTM4MDow",
      "error": null,
      "success": null
    }
  }
}
```

REGISTER A SENSOR FOR HARVEST USING A CURSOR

If the response contains a cursor, update your mutation request to pass the cursor:

```
mutation exampleHarvestSensor2($cursor: Cursor, $harvest: Boolean!, $integrationName:
String!, $name: String!) {
  sensorHarvest(
    input: {cursor: $cursor, harvest: $harvest, integrationName: $integrationName, name:
$name}
  ) {
    cursor
    error {
      message
      retryable
      timedOut
    }
    success
  }
}
```

Update your variables to include the cursor:

```
{
  "cursor": "NTc2NTM4MDow",
  "harvest": true,
  "integrationName": "examplecompany-integratetanium",
  "name": "test-sensor"
}
```

Example response after sensor registration completes:

```
{
  "data": {
    "sensorHarvest": {
```

©2026 Tanium Inc. All Rights Reserved

```
      "cursor": null,
      "error": null,
      "success": true
    }
  }
}
```

Threat Response example

The following mutation requires Threat Response and assigns the **Resolved** status to a Threat Response alert.

Assign Resolved status to alert

(mutation.threatResponseAlertResolve)

About this task

The following request assigns the **Resolved** status to the Threat Response alert matching the GUID.

```
mutation exampleResolveTHRAlert($guid: ID) {
  threatResponseAlertResolve(ref: {guid: $guid}) {
    resolved
    guid
    error
  }
}
```

Include the GUID variable in the **Variables** panel or in your variables dictionary:

```
{
  "guid": "12345678-90ab-cdef-1234-567890abcdef"
}
```

Example response:

```
{
  "data": {
    "threatResponseAlertResolve": {
      "resolved": true,
      "guid": "12345678-90ab-cdef-1234-567890abcdef",
      "error": null
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Reference: Deprecated fields

Fields might be updated to stability index `0 - Deprecated` in favor of other fields for improved performance, functionality support, or other reasons. Deprecated fields receive support and maintenance for a minimum of 12 months from the initial deprecation announcement. Starting with the next release after that 12 month period, deprecated fields might be removed from the Gateway schema.

After a field deprecation announcement, update your requests to replace deprecated fields with the recommended replacement fields.

March 2026 deprecated fields

The following fields are changed to stability `0 - Deprecated`; support for these fields ends April 2027.

March 2026 deprecated fields

Deprecated field	Replacement field
`input.PlaybookStartInput.taskArgumentOverrides`	`input.PlaybookStartInput.stepOverrides.taskArgument`
`input.PlaybookScheduleUpsertInput.taskArgumentOverrides`	`input.PlaybookScheduleUpsertInput.stepOverrides.taskArgument`
`object.PlaybookSchedule.taskArgumentOverrides`	`object.PlaybookSchedule.stepOverrides.taskArgument`
`object.PlaybookStepTaskArgumentOverride`	`object.PlaybookStepOverride`
`query.playbookRuns.edges.node.steps.error`	`query.playbookRuns.edges.node.steps.errorV2`
`query.playbookRun.steps.error`	`query.playbookRun.steps.errorV2`
`query.playbookRuns.filter.isScheduled`	`query.playbookRuns.filter.scheduleID`

August 2025 deprecated fields

The following fields are changed to stability `0 - Deprecated`; support for these fields ends March 2027.

©2026 Tanium Inc. All Rights Reserved

August 2025 deprecated fields

Deprecated field	Replacement field
`query.playbookRun.steps.progress.details.actionID`	`query.playbookRun.steps.taskResult.actionID`
`query.playbookRuns.edges.node.steps.progress.details.actionID`	`query.playbookRuns.edges.node.steps.taskResult.actionID`

July 2025 deprecated fields

July 2025 deprecated fields

Deprecated field	Replacement field
`query.endpoints(source.tds.allNamespaces)` `query.endpoints(source.tds.namespaces)`	These fields will remain, but the behavior of these fields is deprecated as of July 2025. Currently, you can include the `allNamespaces` and `namespaces` fields in the `source` input to `query.endpoints`. After support ends in December 2026, these fields become mutually exclusive; if you include both fields, a validation error occurs. To continue using these fields, only use one of the fields in each query.

June 2025 deprecated fields

The following fields are changed to stability `0 - Deprecated`; support for these fields ends December 2026.

June 2025 deprecated fields

Deprecated field	Replacement field
`query.playbookRun.steps.confirmationResult`	`query.playbookRun.steps.taskResult`
`query.playbookRun.steps.progress`	`query.playbookRun.steps.taskResult`
`query.playbookRuns.edges.node.steps.confirmationResult`	`query.playbookRuns.edges.node.steps.taskResult`
`query.playbookRuns.edges.node.steps.progress`	`query.playbookRuns.edges.node.steps.taskResult`

©2026 Tanium Inc. All Rights Reserved

April 2025 deprecated fields

The following fields are changed to stability `0 - Deprecated`; support for these fields ends November 2026.

April 2025 deprecated fields

Deprecated field	Replacement field
`mutation.playbookUpsert.playbook.canRun`	This field does not map directly to a replacement field. Instead, use `mutation.playbookStart` with (`input.dryRun`) set to `true` . The `dryRun` field consumes fewer resources to determine whether the current user can create a playbook.
`mutation.playbookStart.playbookRun.playbook.canRun`	
`query.playbookRuns.edges.node.canRun`	
`query.playbooks.edges.node.canRun`	
`query.playbookSchedules.edges.node.playbook.canRun`	
`mutation.playbookStart.playbookRun.steps.playbookStep.currentUserMissingPermissions`	
`mutation.playbookUpsert.playbook.runs.edges.node.steps.playbookStep.currentUserMissingPermissions`	
`query.playbookRuns.edges.node.steps.currentUserMissingPermissions`	
`query.playbooks.edges.node.steps.currentUserMissingPermissions`	
`query.playbookSchedules.edges.node.playbook.steps.currentUserMissingPermissions`	

November 2024 deprecated fields

The following field is changed to stability `0 - Deprecated`; support for this field ends June 2026.

November 2024 deprecated fields

Deprecated field	Replacement field
`mutation.assetsImport`	`mutation.assetUpsertEndpoints`

August 2024 deprecated fields

The following fields are changed to stability `0 - Deprecated`; support for these fields ends March

©2026 Tanium Inc. All Rights Reserved

2026.

August 2024 deprecated fields

Deprecated field	Replacement field	Notes
`mutation.deleteConfigurationItemElement`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.deleteRelationship`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.importConfigurationItemEntities`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.mergeConfigurationItemElements`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.syncAssets`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.updateConfigurationItemProperties`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.upsertRelationship`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.configurationItemEntities`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.configurationItemProperties`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
		integration.
`query.configurationItemRelationships`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.endpoints.edges.node.configurationItem`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.endpoints.edges.node.risk`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.assetCriticality`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.criticalityScore`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.riskLevel`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.totalScore`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.vectors`	This field retrieves only Risk Score 1.0 information. Gateway	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
	does not provide a replacement field.	more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.relationshipTypes`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.

July 2023 deprecated fields

Note: This release introduces the stability index. Fields deprecated in prior releases are changed to stability `0 - Deprecated`.

The following field is changed to stability `0 - Deprecated`; support for this field ends February 2025.

July 2023 deprecated fields

Deprecated field	Replacement field	Notes
`query.endpointLastSeen`	`query.endpoints (filter: {path: id, op: EQ, value: "endpoint-id"})` and include `query.endpoints.edges.node.eidLastSeen` in the request	Use `query.endpoints (filter: {path: id, op: EQ, value: "endpoint-id"})` with the `query.endpoints.edges.node.eidLastSeen` field instead. This allows retrieving the last seen time for multiple endpoints in a single request. The `query.endpointLastSeen (eids)` argument does not map directly to a replacement argument. Instead, create a `query.endpoints` filter argument, using `{path: id, op: EQ, value: "endpoint-id"}` to filter the request to return selected endpoints based on ID, then also return `query.endpoints.edges.node.eidLastSeen` in the response to view an

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
		endpoint's last seen time. For more information, see an `endpoints` query with an endpoint ID filter returning `eidlastSeen` Reference: Filter syntax on page 562 .

May 2023 deprecated fields

Support for these fields ends November 2024.

May 2023 deprecated fields

Deprecated field	Replacement field	Notes
`mutation.closeDirectConnection`	`mutation.directConnectClose`	Use `mutation.directConnectClose` instead. The `directConnect` APIs provide better support for tracking connections that take longer to establish. The `mutation.closeDirectConnection (input)` argument maps one-to-one to the `mutation.directConnectClose (input)` argument. For more information, see a `mutation.directConnectClose` Reference: Gateway examples on page 63 .
`mutation.killProcess`	`mutation.directConnectProcessTerminate`	Use `mutation.directConnectProcessTerminate` instead. The `directConnect` APIs provide better support for tracking connections that take longer to establish. Certain arguments for

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
		`mutation.killProcess (input)` map one-to-one to `mutation.directConnectProcessTerminate (input)` arguments: • `name` • `pid` • `signal` The `mutation.killProcess (input: {connectionID: ID})` optional argument maps to the `mutation.directConnectProcessTerminate (input: {connectionID: ID!}` required argument. The `mutation.killProcess (input: {endpoint: {endpointID: ID}})` argument does not map directly to a replacement argument. Instead of passing an endpoint ID, pass an open connection ID in the `mutation.directConnectProcessTerminate (input: {connectionID: ID!})` argument. For more information, see a `mutation.directConnectProcessTerminate` Reference: Gateway examples on page 63 .
`mutation.openDirectConnection`	`mutation.directConnectOpen`	Use `mutation.DirectConnectOpen` instead. The `directConnect` APIs provide better support for tracking connections that take longer to establish. The `mutation.openDirectConnection (input {endpointID: ID})` optional argument maps to the `mutation.`

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
		`directConnectOpen (input {endpointID: ID!})` required argument. For more information, see a `mutation.directConnectOpen` Reference: Gateway examples on page 63 .
`mutation.pingDirectConnection`	`mutation.directConnectPing`	Use `mutation.DirectConnectPing` instead. The `directConnect` APIs provide better support for tracking connections that take longer to establish. The `mutation.PingDirectConnection (input)` argument maps one-to-one to the `mutation.directConnectPing (input)` argument. For more information, see a `mutation.directConnectPing` Reference: Gateway examples on page 63 .
`query.directEndpoint`	`query.directConnectEndpoint`	Use `query.directConnectEndpoint` instead. The `directConnect` APIs provide better support for tracking connections that take longer to establish. The `query.directEndpoint (input: {endpointID: ID})` does not map directly to a replacement argument. Instead of passing an endpoint ID, pass an open connection ID in the `query.directConnectEndpoint (input: {connectionID: ID!})` argument. For more information, see a `query.directConnectEndpoint.performance` Reference: Gateway examples on page 63 , a `query.`

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
		`directConnectEndpoint.processes` Reference: Gateway examples on page 63 , or a `query.directConnectEndpoint.alerts` Reference: Gateway examples on page 63 .

November 2022 deprecated fields

Support for these fields ends May 2024.

November 2022 deprecated fields

Deprecated field	Replacement field	Notes
`query.endpoints.edges.node.software`	`query.endpoints.edges.node.installedApplications` , `query.endpoints.edges.node.deployedSoftwarePackages`	Use `query.endpoints.edges.node.installedApplications` instead for improved filter support when retrieving a list of software installed on the endpoint. Use `query.endpoints.edges.node.deployedSoftwarePackages` instead for improved filter support when retrieving software that is or can be deployed on the endpoint. The `query.endpoints.edges.node.software` field retrieves both installed software and software that is or can be deployed on an endpoint, which might cause confusion when reviewing the results. The `query.endpoints.edges.node.software.name` field is deprecated in favor of the `query.endpoints.edges.node.installedApplications.name` field. The `query.endpoints.edges.node.software.installedVersions` field is deprecated in favor of the

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
		`query.endpoints.edges.node.installedApplications.version` field. The `query.endpoints.edges.node.software.installableVersions` field and `query.endpoints.edges.node.software.updateToVersions` are deprecated in favor of the `query.endpoints.edges.node.deployedSoftwareVersions.version` field.

August 2022 deprecated fields

Support for these fields ends February 2024.

August 2022 deprecated fields

Deprecated field	Replacement field	Notes
`query.directEndpoint.performance.memoryUsagePercent`	`query.directEndpoint.performance.memoryUsedPercent`	Use `query.directEndpoint.performance.memoryUsedPercent` instead, which provides a usage percentage, as opposed to `query.directEndpoint.performance.memoryUsagePercent` which provides a remaining memory percentage and is inconsistent with other fields that provide a usage percentage. The `query.directEndpoint.performance.`

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
		`memoryUsagePercent` field maps one-to-one to `query.directEndpoint.performance.memoryUsedPercent`.
`query.lastActionDetails (id)`	`query.scheduledAction (ref: {id}).lastAction`	Use `query.scheduledAction (ref: {id}).lastAction` instead, which returns information about a scheduled action and the most recent action created by the scheduled action. Child fields for `query.lastActionDetails` map one-to-one to child fields for `query.scheduledAction.lastAction`. For more information, see a `query.scheduledAction` Get scheduled action information on page 149.
`query.lastActionResults (id)`	`query.scheduledAction (ref: {id}).lastAction.results`	Use `query.scheduledAction.lastAction.results` instead, which returns information about a scheduled action and the most recent action created by the scheduled action. Child fields for `query.lastActionResults` map one-to-one to child fields for `query.scheduledAction.lastAction.results`. For more information, see a `query.scheduledAction` Get scheduled action information on page 149.

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
`mutation.createAction (action: {<arguments>})`	`mutation.actionPerform (input: {<arguments>})`, `mutation. scheduledActionCreate (input: {<arguments>})`,	Use `mutation.actionPerform (input: {<arguments>})` instead, which allows you to better define action functionality beyond the constrained set available in this mutation. Certain arguments for `mutation.createAction` map one-to-one to `mutation.actionPerform` arguments: • `changeClientSetting` • `collectActiveDirectoryInfo` • `reboot` • `restartService` • `startService` • `stopService` For more information, see a `mutation.actionPerform` Perform an action operation on page 156. The `mutation.createAction (name)` argument maps one-to-one to the `mutation. scheduledActionCreate (name)`. Certain `mutation.createAction` arguments map to the following `mutation. scheduledActionCreate` arguments: • `description` deprecated, use `comment` instead

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
		• `schedule.distributeOverSeconds` deprecated, use `distributeSeconds` instead • `schedule.end` deprecated, use `endTime` instead • `schedule.reissueSeconds` deprecated, use `issueSeconds` instead • `schedule.start` deprecated, use `startTime` instead • `target` deprecated, use `targets` instead For more information, see a `mutation.scheduledActionCreate` Create scheduled action on page 135 . The `mutation.createAction(replaceExisting)` argument does not map directly to a replacement argument. Instead, retrieve scheduled actions, filtering on the name you want to replace (`query.scheduledActions`), then delete the scheduled actions by ID (`mutation.scheduledActionDelete`). For more information, see a `query.scheduledActions` Get scheduled actions (filter by name) on page 154 and a `mutation.scheduledActionDelete` Delete action or scheduled action on page 140 .

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
`mutation.deleteAction (name)`	`mutation. scheduledActionDelete (ref: {id})`	Use `mutation. scheduledActionDelete (ref: {id})` instead, which allows you to more precisely delete an action by ID Pass the action ID instead of the action name to delete an action. For more information, see a `mutation. scheduledActionDelete` Delete action or scheduled action on page 140 .

December 2021 deprecated fields

Support for these fields ends June 2023.

December 2021 deprecated fields

Deprecated field	Replacement field
`query.assets`	Use `query.endpoints` , which provides better pagination and filtering.

November 2021 deprecated fields

These fields were deprecated before Gateway released to the public. Support for these fields ends May 2023.

November 2021 deprecated fields

Deprecated field	Replacement field
`mutation.installSoftware`	Use `mutation.manageSoftware` instead, for improved functionality in creating a software package deployment to install, update, or remove a software package.

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field
`mutation.removeSoftware`	Use `mutation.manageSoftware` instead, for improved functionality in creating a software package deployment to install, update, or remove a software package.
`mutation.upsertEntity`	This field was deprecated before the first official Gateway release. Use `mutation.importConfigurationItemEntities` instead.
`query.directConnection`	Use `query.directConnectEndpoint` instead, for improved performance and stability.
`query.endpoints.edges.node.diskSpace`	`query.endpoints.edges.node.disks` instead, which supports retrieving multiple disks for a given host, as opposed to `query.endpoints.edges.node.diskSpace` which returns information for only one disk drive.
`query.endpoints.edges.node.memory.used`	`query.endpoints.edges.node.memory.total` instead, for improved support and data.
`query.endpoints.edges.node.memory.usedPercentage`	
`query.entities`	This field was deprecated before the first official Gateway release. Use `query.configItemEntities` instead.
`query.relationships`	This field was deprecated before the first official Gateway release. Use `query.configItemRelationships` instead.
`query.software`	Use `query.softwarePackages` instead, for improved pagination and support.
`query.directEndpoint.performance.memoryUsagePercent`	Use `query.directEndpoint.performance.memoryUsedPercent` instead, which provides a usage percentage, as opposed to `query.directEndpoint.performance.memoryUsagePercent` which provides a remaining memory percentage and is inconsistent with other fields that provide a usage percentage.

©2026 Tanium Inc. All Rights Reserved

Reference: Filter syntax

Filter query requests to constrain records returned in a response, or mutation requests to constrain a created group's endpoint membership, or a created action's targeted groups.

Filters compare a field value or group membership, passed in the request, to the corresponding field value or group membership on a record. If the record matches the filter, the record is added to the collection.

Filters generally evaluate one of the following comparisons:

- `path` and `value` - a record field and the associated value
- `sensor` and `value` - a sensor evaluation and associated result value, compared to the record
- `memberOf` - endpoint group membership

Filters that compare values use relational operators, enumerated in the `op` field. If a filter does not define `op`, the default is `EQ`. For more information on available relational operators, see `FieldFilterOp` in the **Documentation Explorer** pane of the query explorer.



Note: Relational operators might be restricted to certain data types, data sources, or field data type comparisons.

The following relational operators are available for filters, subject to restrictions:

- `CONTAINS` - for string data type fields, the record field value contains the filter value
- `ENDS_WITH` - for string data type fields, the record field value ends with the filter value
- `EQ` - the record field value equals the filter value
- `GT` - for data types that can be compared, the record field value is greater than the filter value

©2026 Tanium Inc. All Rights Reserved

- `GTE` - for data types that can be compared, the record field value is greater than or equal to the filter value
- `LT` - for data types that can be compared, the record field value is less than the filter value
- `LTE` - for data types that can be compared, the record field value is less than or equal to the filter value
- `MATCHES` - for string data type fields, the record field value matches the filter value; the filter value is evaluated as a Perl syntax regular expression
- `READ_AFTER` - for certain data sources, the record field value is read after the filter date and time value
- `STARTS_WITH` - for string data type fields, the record field value starts with the filter value
- `UPDATED_AFTER` - for certain data sources, the record field value is updated after the filter date and time value

Filters also use logical operators to evaluate whether a filter or combination of filters is true or false.

- `negated` (`NOT` operator) - If a filter contains `negated: true`, records that do not match a filter statement are included in a collection. If a filter contains `negated: false`, records that match a filter statement are included in a collection. The default is `negated: false` (matching records belong to the collection).
- `any` (`AND`, `OR` operators) - If a filter includes multiple filter statements and contains `any: false`, a record must match every field filter to be included in the collection. If a filter includes multiple filter statements and contains `any: true`, a record belongs to the collection if it matches one or more of the field filters. The default is `any: false` (a record must match every field filter).



Note: Query filter syntax does not support the `get` statement functionality that Interact supports to retrieve specific fields and information from records in the returned collection. Query filters function like the `from` clause when asking a question, to limit the records in the returned collection. Depending on the query, you might retrieve field information as object metadata. When querying endpoints, use the `endpoints.edges.node.sensorReadings` field to retrieve sensor results for endpoints.

©2026 Tanium Inc. All Rights Reserved

Actions query filter syntax

Actions queries (`query.actions`) support the `path` and `value` filter fields.

For more information on syntax, see `FieldFilter` in the **Documentation Explorer** pane of the query explorer.

Single filter field

You can filter on a single field value. Unless otherwise specified, define a filter object using the syntax:

```
{path: "field-path", negated: "boolean", op: "operator", value: "filter-value"}
```

For example, the following request filters and returns only actions whose name does not contain `production`:

```
query getActionsFiltered($path: String, $negated: Boolean!, $op: FieldFilterOp!, $value:
String) {
  actions(
    filter: {path: $path, negated: $negated, op: $op, value: $value}
  ) {
    edges {
      node {
        approver {
          persona {
            id
            name
          }
          user {
            id
            name
          }
        }
        comment
        creationTime
        creator {
          persona {
            id
            name
          }
          user {
            id
            name
          }
        }
        distributeSeconds
        expirationTime
        expireSeconds
        id
        name
        package {
          id
```

©2026 Tanium Inc. All Rights Reserved

```
          name
          params
          sensorSourcedParams {
            name
            value
          }
        }
        results {
          completed
          downloading
          expired
          failed
          failedVerification
          id
          pendingVerification
          running
          verified
          waiting
          waitingToRetry
        }
        scheduledAction {
          approved
          comment
          distributeSeconds
          endTime
          expireSeconds
          id
          issueCount
          issueSeconds
          lastAction {
            id
          }
          lastStartTime
          name
          nextStartTime
          package {
            id
          }
          startTime
          status
        }
        startTime
        status
        stopped
        stoppedFlag
      }
    }
  }
}
```

Include the filter-related variables in the **Variables** panel:

```
{
  "path": "name",
  "negated": true,
  "op": "CONTAINS",
  "value": "production"
}
```

©2026 Tanium Inc. All Rights Reserved

Multiple filter fields

You can filter on multiple field values. Define a filter object using the syntax:

```
{any: boolean,
  filters: [
    {path: "field-path-1", negated: "boolean-1", op: "operator-1", value: "filter-value-1"},
    {path: "field-path-2", negated: "boolean-2", op: "operator-2", value: "filter-value-2"},
    {path: "field-path-n", negated: "boolean-n", op: "operator-n", value: "filter-value-n"}
  ]
```

For example, the following request filters and returns only scheduled actions whose name does not contain `production`, and whose comment contains `test`:

```
query actionsMultipleFilter($path1: String, $negated1: Boolean!, $op1: FieldFilterOp!,
$value1: String, $path2: String, $negated2: Boolean!, $op2: FieldFilterOp!, $value2: String)
{
  actions(
    filter: {any: false, filters: [{path: $path1, negated: $negated1, op: $op1, value:
$value1}, {path: $path2, negated: $negated2, op: $op2, value: $value2}]}
  ) {
    edges {
      node {
        approver {
          persona {
            id
            name
          }
          user {
            id
            name
          }
        }
        comment
        creationTime
        creator {
          persona {
            id
            name
          }
          user {
            id
            name
          }
        }
        distributeSeconds
        expirationTime
        expireSeconds
        id
        name
        package {
          id
          name
          params
          sensorSourcedParams {
            name
            value
          }
        }
```

©2026 Tanium Inc. All Rights Reserved

```
        results {
          completed
          downloading
          expired
          failed
          failedVerification
          id
          pendingVerification
          running
          verified
          waiting
          waitingToRetry
        }
        scheduledAction {
          approved
          comment
          distributeSeconds
          endTime
          expireSeconds
          id
          issueCount
          issueSeconds
          lastAction {
            id
          }
          lastStartTime
          name
          nextStartTime
          package {
            id
          }
          startTime
          status
        }
        startTime
        status
        stopped
        stoppedFlag
      }
    }
  }
}
```

Include the filter-related variables in the **Variables** panel:

```
{
  "path1": "name",
  "negated1": true,
  "op1": "CONTAINS",
  "value1": "production",
  "path2": "comment",
  "negated2": false,
  "op2": "CONTAINS",
  "value2": "test"
}
```

Asset query filter syntax

Filter Asset queries when retrieving Asset products (`query.assetProducts`) or endpoints

©2026 Tanium Inc. All Rights Reserved

(`query.assetProductEndpoints`).

Asset product queries (`assetProducts`) support the following filter fields:

- `search`
- `states`
- `vendors`

For more information on syntax, see `assetProductsFilter` in the **Documentation Explorer** pane of the query explorer.

Endpoint queries using `assetProductEndpoints` support the following filter fields:

- `name`
- `vendor`
- `version`
- `usage`

For more information on syntax, see `assetProductEndpointsFilter` in the **Documentation Explorer** pane of the query explorer.

Asset products

You can retrieve Asset products and filter based on product metadata. Define a filter object using the syntax:

```
filter {
  vendors: [
    "vendor-name-1",
    "vendor-name-2",
    "vendor-name-n..."
  ],
  search: "search-string",
  states: [Cataloged | Ignored | Tracked]
}
```

For example, the following request retrieves only Asset products with a vendor name of `Example vendor`, an Asset state of `Cataloged`, and the search string `Product` in either the vendor name or product name:

©2026 Tanium Inc. All Rights Reserved

```
query exampleGetAssetProductsFiltered($first: Int, $vendor: String!, $search: String,
$states: AssetProductState!) {
  assetProducts(
    first: $first, filter: {
      vendors: [$vendor],
      search: $search,
      states: [$states]
    }
  ) {
    edges {
      node {
        vendor
        name
        installation {
          installedCount
          usedCount
          unusedCount
          pendingUsage
        }
        usage {
          usageNotDetected
          notInstalled
          baselining
          limited
          normal
          high
        }
        tracking {
          state
          reportingPeriodDays
          normalMinutesUsedPerDay
          highMinutesUsedPerDay
          baselinePeriodDays
        }
        versions {
          version
          installs
        }
      }
    }
  }
}
```

Include the filter variables in the **Variables** panel:

```
{
  "first": 2,
  "vendor": "Example vendor",
  "search": "Product",
  "states": "Cataloged"
}
```

Endpoints based on Asset products

You can retrieve endpoints and filter the returned endpoints based on Asset product information. Define a filter object using the syntax:

```
filter {
  vendors: [
```

©2026 Tanium Inc. All Rights Reserved

```
    "vendor-name-1",
    "vendor-name-2",
    "vendor-name-n..."
  ],
  name: "name-string",
  version: "version-string"
}
```

For example, the following request retrieves only endpoints that have installed the `Example name` product, with a vendor of `Example vendor` and version of `1.2.3`:

```
query exampleGetEndpointsAssetProduct($first: Int, $vendor: String, $name: String, $version:
String) {
  assetProductEndpoints(
    first: $first
    filter: {vendor: $vendor, name: $name, version: $version, usage: Normal}
  ) {
    edges {
      node {
        id
        eid
        computerName
        computerId
        serialNumber
        osPlatform
        operatingSystem
        servicePack
        manufacturer
        ipAddress
        userName
        createdAt
        updatedAt
      }
    }
  }
}
```

Include the filter variables in the **Variables** panel:

```
{
  "first": 2,
  "vendor": "Example vendor",
  "name": "Example name",
  "version": "1.2.3"
}
```

Automate query filter syntax

Automate queries that request playbook information (`query.playbooks`) support the following filter fields:

- `authorUserIDs`
- `contentSets`

©2026 Tanium Inc. All Rights Reserved

- `createdAfterTime`
- `createdBeforeTime`
- `hasDisabledSchedules`
- `ids`
- `isScheduled`
- `name`

Automate queries that request playbook information support sorting the response on the following filter fields in ascending (`asc`) or descending (`desc`) order:

- `activeRunsCount` based on number count
- `name` based on alphabetical order
- `schedulesCount` based on number count

If you do not define a sort order, the query defaults to sorting on `name` in `asc` order.

Automate queries that request playbook run information (`query.playbookRuns`) support the following filter fields:

- `createdAfterTime`
- `createdBeforeTime`
- `hasErrors`
- `startReason.startTypes`
- `playbookIDs`
- `playbookName`
- `runAsUserIDs`
- `statuses`

©2026 Tanium Inc. All Rights Reserved

Automate queries that request playbook run information support sorting the response on the following filter fields in ascending (`asc`) or descending (`desc`) order:

- `endTime` based on chronological order
- `name` based on alphabetical order
- `scheduleStartTime` based on chronological order
- `startTime` based on chronological order
- `status` based on playbook run lifecycle order, from creation to completion

If you do not define a sort order, the query defaults to sorting on `status` in `asc` order.

Automate queries that request playbook schedule information (`query.playbookSchedules`) support the following filter fields:

- `excludedScheduledByIDs`
- `isEnabled`
- `playbookIDs`
- `scheduledByIDs`

The `playbooks`, `playbookRuns`, and `playbookSchedules` queries require Automate 1.3 or later and the **Automate Playbook read** permission.

For more information on filter syntax, see `PlaybookFieldFilter` and `PlaybookRunFieldFilter` in the **Documentation Explorer** pane of the query explorer. For more information on sort syntax, see `PlaybookFieldSort`, `PlaybookSortField`, `PlaybookRunFieldSort`, and `PlaybookRunSortField` in the **Documentation Explorer** pane of the query explorer.

Playbooks belonging to a content set

You can retrieve playbooks belonging to a content set by content set ID or name. Define a filter object using the following syntax to filter by content set ID:

```
filter {
```

©2026 Tanium Inc. All Rights Reserved

```
  contentSets:
    [
      {id: content-set-id-value-1},
      {id: content-set-id-value-2},
      {id: content-set-id-value-n}
    ]
}
```

Define a filter object using the following syntax to filter by content set name:

```
filter {
  contentSets:
    [
      {name: content-set-name-value-1},
      {name: content-set-name-value-2},
      {name: content-set-name-value-n}
    ]
}
```

For example, the following query retrieves the first two playbooks belonging to the content set with the ID `123`:

```
query exampleGetPlaybooksContentSetID($first: Int, $contentSetID: ID) {
  playbooks(first: $first, filter: {contentSets: [{id: $contentSetID}]}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "contentSetID": "123"
}
```

©2026 Tanium Inc. All Rights Reserved

The following query retrieves the first two playbooks belonging to the content set with the name `Example content set`:

```
query exampleGetPlaybooksContentSetName($first: Int, $contentSetName: String) {
  playbooks(first: $first, filter: {contentSets: [{name: $contentSetName}]}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "contentSetName": "Example content set"
}
```

Playbooks or playbook runs created after a time

You can retrieve playbooks and playbook runs created after a date. If you retrieve a playbook, you can also retrieve associated playbook runs created after a date. Define a filter object using the following syntax:

```
filter {
  createdAfterTime: time-value
}
```

For example, the following query retrieves the first two playbooks created after June 1, 2023:

```
query exampleGetPlaybooksAfter($first: Int, $create: Time) {
  playbooks(first: $first, filter: {createdAfterTime: $create}) {
    edges {
      node {
        createdTime
```

©2026 Tanium Inc. All Rights Reserved

```
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "create": "2023-06-01T01:02:03Z"
}
```

The following query retrieves the first two playbook runs created after June 1, 2023:

```
query exampleGetPlaybookRunsAfter($first: Int, $create: Time) {
  playbookRuns(first: $first, filter: {createdAfterTime: $create}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
```

```
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
          }
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "create": "2023-06-01T01:02:03Z"
}
```

Playbooks or playbook runs created before a time

You can retrieve playbooks and playbook runs created before a date. If you retrieve a playbook, you can also retrieve associated playbook runs created before a date. Define a filter object using the following syntax:

```
filter {
  createdBeforeTime: time-value
}
```

For example, the following query retrieves the first two playbooks created before June 1, 2023:

```
query exampleGetPlaybooksBefore($first: Int, $create: Time) {
```

©2026 Tanium Inc. All Rights Reserved

```
  playbooks(first: $first, filter: {createdBeforeTime: $create}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "create": "2023-06-01T00:00:00Z"
}
```

The following query retrieves the first two playbook runs created before June 1, 2023:

```
query exampleGetPlaybookRunsBefore($first: Int, $create: Time) {
  playbookRuns(first: $first, filter: {createdBeforeTime: $create}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
        playbook {
          activeRunsCount
          author {
            id
          }
```

©2026 Tanium Inc. All Rights Reserved

```
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
          }
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "create": "2023-06-01T00:00:00Z"
}
```

Playbooks configured with a playbook schedule

You can retrieve playbooks associated with at least one configured playbook schedule. Define a filter object using the following syntax:

```
filter {
  isScheduled: boolean-value
}
```

For example, the following query retrieves the first two playbooks with a configured schedule:

©2026 Tanium Inc. All Rights Reserved

```
query exampleGetPlaybooksScheduled($first: Int, $isScheduled: Boolean) {
  playbooks(first: $first, filter: {isScheduled: $isScheduled}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "isScheduled": true
}
```

Playbooks configured with at least one disabled playbook schedule

You can retrieve playbooks associated with at least one configured playbook schedule that is also disabled. Define a filter object using the following syntax:

```
filter {
  hasDisabledSchedules: boolean-value
}
```

For example, the following query retrieves the first two playbooks with at least one configured schedule that is also disabled:

```
query exampleGetPlaybooksScheduleDisabled($first: Int, $hasDisabledSchedules: Boolean) {
  playbooks(first: $first, filter: {hasDisabledSchedules: $hasDisabledSchedules}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
```

©2026 Tanium Inc. All Rights Reserved

```
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "hasDisabledSchedules": true
}
```

Playbooks with a referenced name

You can retrieve playbooks configured with a given name. Define a filter object using the following syntax:

```
filter {
  name: name-value
}
```

For example, the following query retrieves playbooks with the name `Example playbook`:

```
query exampleGetPlaybooksNamed($name: String) {
  playbooks(filter: {name: $name}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
```

©2026 Tanium Inc. All Rights Reserved

```
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variable in the **Variables** panel or in your variables dictionary:

```
{
  "name": "Example playbook"
}
```

Playbooks with a referenced author ID

You can retrieve playbooks created by a given user ID. Define a filter object using the following syntax:

```
filter {
  authorUserIDs: [user-id-value-1,user-id-value-2,user-id-value-n]
}
```

For example, the following query retrieves the first two playbooks created by user ID `123`:

```
query exampleGetPlaybooksAuthor($first: Int, $authorUserIDs: [ID!]) {
  playbooks(first: $first, filter: {authorUserIDs: $authorUserIDs}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variable in the **Variables** panel or in your variables dictionary:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "first": 2,
  "authorUserIDs": ["123"]
}
```

Playbooks with a referenced ID

You can retrieve playbooks with a given ID. Define a filter object using the following syntax:

```
filter {
  ids: [
    playbook-id-value-1,
    playbook-id-value-2,
    playbook-id-value-n
  ]
}
```

For example, the following query retrieves the playbooks with the IDs `12345678-90ab-cdef-1234-567890abcdef` and `fedcba09-8765-4321-fedc-ba0987654321`:

```
query exampleGetPlaybooksByID($playbookID1: ID!, $playbookID2: ID!) {
  playbooks(filter: {ids: [$playbookID1, $playbookID2]}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "playbookID1": "12345678-90ab-cdef-1234-567890abcdef",
  "playbookID2": "fedcba09-8765-4321-fedc-ba0987654321"
}
```

Playbook runs and playbook schedules associated with a playbook ID

©2026 Tanium Inc. All Rights Reserved

You can retrieve playbook runs and playbook schedules associated with a playbook, based on the playbook's ID. Define a filter object using the following syntax:

```
filter {
  playbookIDs: [
    playbook-id-1,
    playbook-id-2,
    playbook-id-n
  ]
}
```

The following query retrieves the first two playbook runs associated with the playbook ID `12345678-90ab-cdef-1234-567890abcdef`:

```
query exampleGetPlaybookRunsPlaybookID($first: Int, $playbookID: ID!) {
  playbookRuns(first: $first, filter: {playbookIDs: [$playbookID]}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
```

©2026 Tanium Inc. All Rights Reserved

```
          }
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "playbookID": "12345678-90ab-cdef-1234-567890abcdef"
}
```

Playbook runs associated with a named playbook

You can retrieve playbook runs associated with a playbook, based on the playbook's name. This matches either the full playbook name or a string within the playbook name. For example, if you filter on `example playbook` in the playbook name, the playbook names `example playbook` and `testing example playbook step` both match. Define a filter object using the following syntax:

```
filter {
  playbookName: playbook-name
}
```

The following query retrieves the first two playbook runs associated with the playbook named `Example named playbook`:

```
query exampleGetPlaybookRunsPlaybookName($first: Int, $playbookName: String) {
  playbookRuns(first: $first, filter: {playbookName: $playbookName}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
```

©2026 Tanium Inc. All Rights Reserved

```
          }
          name
        }
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
          }
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "playbookName": "Example named playbook"
}
```

Playbook runs associated with a `runAs` user ID

You can retrieve playbook runs whose `runAs` user matches the filter. If you retrieve a playbook, you can also retrieve associated playbook runs whose `runAs` user matches the filter. Define a filter object using the following syntax:

```
filter {
  runAsUserIDs: [
    user-ID-value-1,
    user-ID-value-2,
    user-ID-value-n..
  ]
}
```

The following query retrieves the first two playbook runs associated with the `runAs` user `1234`:

```
query exampleGetPlaybookRunsUser($first: Int, $runAsUserID: ID!) {
  playbookRuns(first: $first, filter: {runAsUserIDs: [$runAsUserID]}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
```

©2026 Tanium Inc. All Rights Reserved

```
          }
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "runAsUserID": "1234"
}
```

Playbook runs by status

You can retrieve playbook runs based on their current status. If you retrieve a playbook, you can also retrieve associated playbook runs based on current status. Define a filter object using the following syntax:

```
filter {
  statuses: [
    enumerated-status-value-1,
    enumerated-status-value-2,
    enumerated-status-value-n
  ]
}
```

For information on playbook run status values, see `PlaybookRunStatus` in the **Documentation Explorer** pane of the query explorer.

The following query retrieves the first two playbook runs that are in progress:

```
query exampleGetPlaybookRunsStatus($first: Int, $status: [PlaybookRunStatus!]) {
  playbookRuns(first: $first, filter: {statuses: $status}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
```

©2026 Tanium Inc. All Rights Reserved

```
            id
          }
          user {
            id
          }
        }
        name
      }
      id
      playbook {
        activeRunsCount
        author {
          id
        }
        createdTime
        description
        id
        lastModifiedBy {
          displayName
          id
          username
        }
        name
        revision
        steps {
          description
          name
          taskArgument {
            __typename
          }
        }
        updatedTime
      }
      runAs {
        persona {
          id
        }
        user {
          id
        }
      }
      scheduleStartTime
      startTime
      status
    }
  }
  pageInfo {
    hasPreviousPage
    hasNextPage
    startCursor
    endCursor
  }
  totalRecords
 }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "status": "IN_PROGRESS"
}
```

©2026 Tanium Inc. All Rights Reserved

Playbook runs initiated from a playbook schedule

You can retrieve playbook runs that were started due to a configured playbook schedule. Define a filter object using the following syntax:

```
filter {
  startReason: {startTypes: [SCHEDULE]}
}
```

The following query retrieves the first two playbook runs that started due to a configured playbook schedule:

```
query exampleGetPlaybookRunsStartedBySchedule($first: Int) {
  playbookRuns(first: $first, filter: {startReason: {startTypes: [SCHEDULE]}}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
          }
```

©2026 Tanium Inc. All Rights Reserved

```
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Playbook runs initiated by a user or trigger

You can retrieve playbook runs started due to a user action or from a trigger. Define a filter object using the following syntax:

```
filter {
  startReason: {startTypes: [USER, TRIGGER]}
}
```

The following query retrieves the first two playbook runs that started due to either a user action or a trigger:

```
query exampleGetPlaybookRunsStartedByUserOrTrigger($first: Int) {
  playbookRuns(first: $first, filter: {startReason: {startTypes: [USER, TRIGGER]}}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
        playbook {
          activeRunsCount
```

©2026 Tanium Inc. All Rights Reserved

```
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
          }
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Playbook runs that generated errors

You can retrieve playbook runs that generated an error while running. If you retrieve a playbook, you can also retrieve associated playbook runs that generated an error. Define a filter object using the following syntax:

```
filter {
  hasErrors: boolean-value
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

The following query retrieves the first two playbook runs that generated an error while evaluating steps:

```
query exampleGetPlaybookRunsError($first: Int, $hasErrors: Boolean) {
  playbookRuns(first: $first, filter: {hasErrors: $hasErrors}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
          }
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
```

©2026 Tanium Inc. All Rights Reserved

```
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "hasErrors": true
}
```

Playbook schedules that are enabled

You can retrieve playbook schedules that are enabled. Define a filter object using the following syntax:

```
filter {
  isEnabled: boolean-value
}
```

The following query retrieves the first two enabled playbook schedules:

```
query exampleGetEnabledPlaybookSchedules($first: Int, $isEnabled: Boolean) {
  playbookSchedules(first: $first, filter: {isEnabled: $isEnabled}) {
    edges {
      node {
        createdTime
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
      }
```

```
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "isEnabled": true
}
```

Playbook schedules that are created by a referenced user ID

You can retrieve playbook schedules that are created by a user based on user ID. Define a filter object using the following syntax:

```
filter {
  scheduledByIDs: [
    user-id-1,
    user-id-2,
    user-id-n
  ]
}
```

The following query retrieves the first two playbook schedules created by a user with the ID `123`:

```
query exampleGetPlaybookScheduleUserID($first: Int, $scheduledByID: ID!) {
  playbookSchedules(first: $first, filter: {scheduledByIDs: [$scheduledByID]}) {
    edges {
      node {
        createdTime
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
```

©2026 Tanium Inc. All Rights Reserved

```
            __typename
          }
        }
        updatedTime
      }
    }
  }
  pageInfo {
    hasPreviousPage
    hasNextPage
    startCursor
    endCursor
  }
  totalRecords
 }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "scheduledByID": "123"
}
```

Playbook schedules that are not created by a referenced user ID

You can retrieve playbook schedules that are not created by a user based on user ID. Define a filter object using the following syntax:

```
filter {
  excludedScheduledByIDs: [
    user-id-1,
    user-id-2,
    user-id-n
  ]
}
```

The following query retrieves the first two playbook schedules that are not created by a user with the ID `123`:

```
query exampleGetPlaybookScheduleNotUserID($first: Int, $excludedScheduledByID: ID!) {
  playbookSchedules(first: $first, filter: {excludedScheduledByIDs:
[$excludedScheduledByID]}) {
    edges {
      node {
        createdTime
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
```

©2026 Tanium Inc. All Rights Reserved

```
            username
          }
          name
          revision
          steps {
            description
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2,
  "excludedScheduledByID": "123"
}
```

Playbooks or playbook runs results sorting

You can sort playbooks and playbook runs results. If you retrieve a playbook, you can also sort associated playbook runs. Define a sort object using the following syntax:

```
sort {
  field: field-name
  order: sort-order
}
```

For example, the following query sorts playbook results in descending order based on active run count:

```
query exampleGetPlaybooksSort($field: PlaybookSortField!, $order: SortOrder!) {
  playbooks(sort: {field: $field, order: $order}) {
    edges {
      node {
        createdTime
        description
        id
        name
        revision
        steps {
          description
          name
          taskArgument {
```

©2026 Tanium Inc. All Rights Reserved

```
            __typename
          }
        }
        updatedTime
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "field": "activeRunsCount",
  "order": "desc"
}
```

The following query sorts playbook run results in ascending order based on playbook run end time:

```
query exampleGetPlaybookRunsSort($field: PlaybookRunSortField!, $order: SortOrder!) {
  playbookRuns(sort: {field: $field, order: $order}) {
    edges {
      node {
        createdTime
        endTime
        events {
          createdTime
          initiator {
            persona {
              id
            }
            user {
              id
            }
          }
          name
        }
        id
        playbook {
          activeRunsCount
          author {
            id
          }
          createdTime
          description
          id
          lastModifiedBy {
            displayName
            id
            username
          }
          name
          revision
          steps {
            description
```

```
            name
            taskArgument {
              __typename
            }
          }
          updatedTime
        }
        runAs {
          persona {
            id
          }
          user {
            id
          }
        }
        scheduleStartTime
        startTime
        status
      }
    }
    pageInfo {
      hasPreviousPage
      hasNextPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "field": "endTime",
  "order": "asc"
}
```

Comply query filter syntax

Endpoint queries that request Comply-related information using `complianceFindings` or `cveFindings` support the `path` and `value` filter fields. Comply 2.16.97 or later is required to submit a request that filters specific System Vulnerability and System Compliance risk vector fields.

For more information on syntax, see `FieldFilter` in the **Documentation Explorer** pane of the query explorer.

Single column

You can filter on field values. Define a filter object using one of the following syntaxes:

```
complianceFindings(filter:
  {
    path: "field-path",
```

©2026 Tanium Inc. All Rights Reserved

```
    op: "operator-value",
    value: "field-value"
  }
)
```

```
cveFindings(filter:
  {
    path: "field-path",
    op: "operator-value",
    value: "field-value"
  }
)
```

Use dot notation for sub-fields not at the root level. You can filter the endpoints returned and filter the endpoints based on Comply information in a single request.

For example, the following query retrieves the first endpoint with a compliance findings `category` of `Error` and returns that endpoint's associated compliance findings:

```
query exampleFilterComplianceFindings($first: Int) {
  endpoints(first: $first) {
    edges {
      node {
        name
        ipAddress
        compliance {
          complianceFindings(filter: {
            path: "category"
            value: "Error"
          }) {
            category
            id
            profile
            profileVersion
            rule
            ruleId
            standard
            standardVersion
            state
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the cursor variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

©2026 Tanium Inc. All Rights Reserved

Multiple columns

You can filter on multiple Comply-related columns. Define a filter object using one of the following syntaxes:

```
complianceFindings(filter:
  {any: boolean,
    filters: [
      {path: "field-path-1", op: "operator-1", value: "filter-value-1"},
      {path: "field-path-2", op: "operator-2", value: "filter-value-2"},
      {path: "field-path-n", op: "operator-n", value: "filter-value-n"}
    ]
  }
)
```

```
cveFindings(filter:
  {any: boolean,
    filters: [
      {path: "field-path-1", op: "operator-1", value: "filter-value-1"},
      {path: "field-path-2", op: "operator-2", value: "filter-value-2"},
      {path: "field-path-n", op: "operator-n", value: "filter-value-n"}
    ]
  }
)
```

Use dot notation for sub-fields not at the root level. You can filter the endpoints returned and filter the endpoints based on Comply information in a single request.

For example, the following query retrieves the first endpoint that is a member of the All Mac computer group, with a vulnerability findings `cveYear` of `2020` and `cvssScore` value of `5`, and returns that endpoint's associated vulnerability findings:

```
query exampleFilterMacOSEndpointVulnerability($first: Int!) {
  endpoints(first: $first, filter: {memberOf: {name: "All Mac"}}) {
    edges {
      node {
        name
        ipAddress
        compliance {
          cveFindings(
            filter: {any: false, filters: [{path: "cveYear", value: "2020"}, {path:
"cvssScore", value: "5"}]}
          ) {
            cveId
            cveYear
            cvssScoreV3
            firstFound
            lastFound
            severityV3
            summary
          }
        }
      }
    }
    pageInfo {
      startCursor
      endCursor
```

©2026 Tanium Inc. All Rights Reserved

```
      hasPreviousPage
      hasNextPage
    }
  }
}
```

Include the cursor variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1
}
```

Computer group creation filter syntax

Filter computer groups when creating actions or scheduled actions, to target certain computer groups, or when constraining endpoint membership in a created computer group.

Computer group creation and action creation mutations support the following filter fields:

- `memberOf`
- `sensor` and `value`

For more information on syntax, see `computerGroupFilter` in the **Documentation Explorer** pane of the query explorer.

Computer group membership

You can define computer group membership based on named computer groups. Define a filter object using the syntax:

```
{memberOf: {name: "computer-group-name"}}
```

For example, the following request creates a computer group whose members are endpoints that are members of the All Linux computer group:

```
mutation createComputerGroup($name: String!, $filterGroup: String, $mrEnabled: Boolean!) {
  computerGroupCreate(
    input: {name: $name, filter: {memberOf: {name: $filterGroup}}, managementRightsEnabled:
$mrEnabled}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Include the computer group name variable in the **Variables** panel:

```
{
    "name": "Example computer group",
    "filterGroup": "All Linux",
    "mrEnabled": false
}
```

The following request creates an action that targets endpoints that are members of the All Linux computer group:

```
mutation createAction($comment: String, $distributeSeconds: Int, $expirationTime: Time,
$expireSeconds: Int, $name: String, $packageID: ID, $cgName: String) {
  actionCreate(
    input: {comment: $comment, distributeSeconds: $distributeSeconds, expirationTime:
$expirationTime, expireSeconds: $expireSeconds, name: $name, package: {id: $packageID},
targets: {targetGroup: {filter: {memberOf: {name: $cgName}}}}}
  ) {
    action {
      id
    }
    error {
      message
    }
  }
}
```

Include the computer group name variable in the **Variables** panel:

```
{
  "comment": "Example created action",
  "distributeSeconds": 60,
  "expirationTime": "2023-01-01T00:00:00+00:00",
  "expireSeconds": 120,
  "name": "Example created action",
  "packageID": 12345,
  "cgName": "All Linux"
}
```

Sensors

You can define computer group membership based on readings of arbitrary named sensors. Define a filter object using the syntax:

```
{sensor: {name: "sensor-name", negated: "boolean", op: "operator", value: "filter-value"}}
```

Only the `name` and `value` arguments are required. If you do not define the `negated` argument, the default value is `false`. If you do not define the `op` argument, the default value is `EQ` (equals).

For example, the following request creates a group based on the Is Virtual sensor returning a value of

©2026 Tanium Inc. All Rights Reserved

`false` for endpoints, and returns only those endpoints:

```
mutation createComputerGroupVirtual($name: String!, $sensor: String, $negated: Boolean!, $op:
FieldFilterOp!, $value: String, $mrEnabled: Boolean!) {
  computerGroupCreate(
    input: {name: $name, filter: {sensor: {name: $sensor}, negated: $negated, op: $op, value:
$value}, managementRightsEnabled: $mrEnabled}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the sensor name and value variables in the **Variables** panel:

```
{
  "name": "Example computer group",
  "sensor": "Is Virtual",
  "negated": false,
  "op": "EQ",
  "value": "false",
  "mrEnabled": false
}
```

The following request creates an action that targets endpoints based on the Is Virtual sensor returning a value of `false` for endpoints, targeting only those endpoints:

```
mutation createActionVirtualEndpoints($comment: String, $distributeSeconds: Int,
$expirationTime: Time, $expireSeconds: Int, $name: String, $packageID: ID, $sensor: String,
$negated: Boolean!, $op: FieldFilterOp!, $value: String) {
  actionCreate(
    input: {comment: $comment, distributeSeconds: $distributeSeconds, expirationTime:
$expirationTime, expireSeconds: $expireSeconds, name: $name, package: {id: $packageID},
targets: {targetGroup: {filter: {sensor: {name: $sensor}, negated: $negated, op: $op, value:
$value}}}}
  ) {
    action {
      id
    }
    error {
      message
    }
  }
}
```

Include the sensor name and value variables in the **Variables** panel:

```
{
  "comment": "Example created action",
  "distributeSeconds": 60,
  "expirationTime": "2023-01-01T00:00:00+00:00",
  "expireSeconds": 120,
  "name": "Example created action",
  "packageID": 12345,
```

©2026 Tanium Inc. All Rights Reserved

```
  "sensor": "Is Virtual",
  "negated": false,
  "op": "EQ",
  "value": "false"
}
```

Parameterized sensors

You can create a computer group based on parameterized sensors. Define a filter object using the syntax:

```
{sensor:
  {name: "sensor-name",
  params:[
    {name: "parameter-1-name", value: "parameter-1-value"},
    {name: "parameter-2-name", value: "parameter-2-value"},
    {name: "parameter-n-name", value: "parameter-n-value"}
  ]
},
  value: "filter-value"
}
```

Add a `params` `name`/`value` object for every parameter.

For example, the following request creates a computer group whose membership is based on the Custom Tag Exists sensor. Only endpoints tagged with the custom tag `the-tag` are returned by the sensor, and are members of this computer group:

```
mutation createComputerGroupTag($name: String!, $sensorName: String, $paramName: String!,
$paramValue: String!, $sensorValue: String, $mrEnabled: Boolean!) {
  computerGroupCreate(
    input: {name: $name, filter: {sensor: {name: $sensorName, params: [{name: $paramName,
value: $paramValue}]}, value: $sensorValue}, managementRightsEnabled: $mrEnabled}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the sensor name, value, and parameter variables in the **Variables** panel:

```
{
  "name": "Example tagged endpoints group",
  "sensorName": "Custom Tag Exists",
  "sensorValue": "true",
  "paramName": "tag",
  "paramValue": "the-tag",
  "mrEnabled": false
}
```

©2026 Tanium Inc. All Rights Reserved

The following request creates an action that targets a computer group whose membership is based on the Custom Tag Exists sensor. Only endpoints tagged with the custom tag `the-tag` are returned by the sensor, and are members of this targeted computer group:

```
mutation createActionTargetTags($comment: String, $distributeSeconds: Int, $expirationTime:
Time, $expireSeconds: Int, $name: String, $packageID: ID, $sensorName: String, $paramName:
String!, $paramValue: String!, $sensorValue: String) {
  actionCreate(
    input: {comment: $comment, distributeSeconds: $distributeSeconds, expirationTime:
$expirationTime, expireSeconds: $expireSeconds, name: $name, package: {id: $packageID},
targets: {targetGroup: {filter: {sensor: {name: $sensorName, params: [{name: $paramName,
value: $paramValue}]}, value: $sensorValue}}}}
  ) {
    action {
      id
    }
    error {
      message
    }
  }
}
```

Include the sensor name, value, and parameter variables in the **Variables** panel:

```
{
  "comment": "Example action tagged endpoints",
  "distributeSeconds": 60,
  "expirationTime": "2023-01-01T00:00:00+00:00",
  "expireSeconds": 120,
  "name": "Example action tagged endpoints",
  "packageID": 12345,
  "sensorName": "Custom Tag Exists",
  "sensorValue": "true",
  "paramName": "tag",
  "paramValue": "the-tag"
}
```

Single column from multi-column sensors

You can create a computer group based on a single column from multi-column sensors by specifying the column name. Define a filter object using the syntax:

```
{sensor: {
  name: "sensor-name",
  column: "sensor-column-name"},
  value: "filter-value"}
}
```

For example, the following request creates a computer group based on the CPU Details sensor returning endpoints with an `Intel` CPU. Those endpoints are members of the computer group:

```
mutation createComputerGroupIntel($name: String!, $sensorName: String, $sensorColumn: String,
$sensorValue: String, $mrEnabled: Boolean!) {
  computerGroupCreate(
```

©2026 Tanium Inc. All Rights Reserved

```
    input: {name: $name, filter: {sensor: {name: $sensorName, column: $sensorColumn}, value:
$sensorValue}, managementRightsEnabled: $mrEnabled}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the sensor name, column, and value variables in the **Variables** panel:

```
{
  "name": "Example computer group Intel",
  "sensorName": "CPU Details",
  "sensorColumn": "CPU",
  "sensorValue": "Intel",
  "mrEnabled": false
}
```

The following request creates an action that targets endpoints based on the CPU Details sensor returning endpoints with an `Intel` CPU. Those endpoints are members of the computer group and targeted by this action:

```
mutation createActionTargetIntel($comment: String, $distributeSeconds: Int, $expirationTime:
Time, $expireSeconds: Int, $name: String, $packageID: ID, $sensorName: String, $sensorColumn:
String, $sensorValue: String) {
  actionCreate(
    input: {comment: $comment, distributeSeconds: $distributeSeconds, expirationTime:
$expirationTime, expireSeconds: $expireSeconds, name: $name, package: {id: $packageID},
targets: {targetGroup: {filter: {sensor: {name: $sensorName, column: $sensorColumn}, value:
$sensorValue}}}}
  ) {
    action {
      id
    }
    error {
      message
    }
  }
}
```

Include the sensor name, column, and value variables in the **Variables** panel:

```
{
  "comment": "Example action Intel endpoints",
  "distributeSeconds": 60,
  "expirationTime": "2023-01-01T00:00:00+00:00",
  "expireSeconds": 120,
  "name": "Example action Intel endpoints",
  "packageID": 12345,
  "sensorName": "CPU Details",
  "sensorColumn": "CPU",
  "sensorValue": "Intel"
}
```

©2026 Tanium Inc. All Rights Reserved

Multiple columns from multi-column sensors

If you want to create a computer group based on multiple columns of each row of a multi-column sensor, define a `filter.filters` list, then define a list element for each column filter, using the following syntax:

```
filters: [
  {sensor: {column: "column-1-name"}, value: "column-1-value"},
  {sensor: {column: "column-2-name"}, value: "column-2-value"},
  {sensor: {column: "column-n-name"}, value: "column-n-value"}
]
```

You can only use a single level of child filters, and can only use the default `EQ` operator.

For example, the following request creates a computer group based on the CPU Details sensor returning endpoints with `4` total cores and an `Intel` CPU:

```
mutation createComputerGroup4CoreIntel($name: String!, $sensorName: String, $sensorColumn1:
String, $sensorValue1: String, $sensorColumn2: String, $sensorValue2: String, $mrEnabled:
Boolean!) {
  computerGroupCreate(
    input: {name: $name, filter: {sensor: {name: $sensorName}, filters: [{sensor: {column:
$sensorColumn1}, value: $sensorValue1}, {sensor: {column: $sensorColumn2}, value:
$sensorValue2}]}, managementRightsEnabled: $mrEnabled}
  ) {
    group {
      id
    }
    error {
      message
    }
  }
}
```

Include the sensor name, column, and value variables in the **Variables** panel:

```
{
  "name": "Example computer group 4 core Intel",
  "sensorName": "CPU Details",
  "sensorColumn1": "CPU",
  "sensorValue1": "Intel",
  "sensorColumn2": "Total Cores",
  "sensorValue2": "4",
  "mrEnabled": false
}
```

In this form, the sensor is identified in the top filter and the columns are identified in the child filters. You can only use a single level of child filters, and can only use the default `EQ` operator.

The following request creates an action that targets endpoints based on the CPU Details sensor returning endpoints with `4` total cores and an `Intel` CPU:

©2026 Tanium Inc. All Rights Reserved

```
mutation createActionTarget4CoreIntel($comment: String, $distributeSeconds: Int,
$expirationTime: Time, $expireSeconds: Int, $name: String, $packageID: ID, $sensorName:
String, $sensorColumn1: String, $sensorValue1: String, $sensorColumn2: String, $sensorValue2:
String) {
  actionCreate(
    input: {comment: $comment, distributeSeconds: $distributeSeconds, expirationTime:
$expirationTime, expireSeconds: $expireSeconds, name: $name, package: {id: $packageID},
targets: {targetGroup: {filter: {sensor: {name: $sensorName}, filters: [{sensor: {column:
$sensorColumn1}, value: $sensorValue1}, {sensor: {column: $sensorColumn2}, value:
$sensorValue2}]}}}}
  ) {
    action {
      id
    }
    error {
      message
    }
  }
}
```

Include the sensor name, column, and value variables in the **Variables** panel:

```
{
  "comment": "Example action 4 core Intel endpoints",
  "distributeSeconds": 60,
  "expirationTime": "2023-01-01T00:00:00+00:00",
  "expireSeconds": 120,
  "name": "Example action Intel endpoints",
  "packageID": 12345,
  "sensorName": "CPU Details",
  "sensorColumn1": "CPU",
  "sensorValue1": "Intel",
  "sensorColumn2": "Total Cores",
  "sensorValue2": "4"
}
```

Computer group query filter syntax

Computer group queries support the following filter fields:

- `filterEnabled`
- `managementRightsEnabled`
- `type`

For more information on syntax, see `computerGroupsFilter` in the **Documentation Explorer** pane of the query explorer.

©2026 Tanium Inc. All Rights Reserved

Manual or standard computer group

You can filter based on whether the computer group is standard (membership is based on the results of a sensor filter expression) or manual (membership is based on a manually entered list of computer names or fully qualified domain names (FQDNs)). To retrieve standard computer groups, define a filter object using the syntax:

```
{type: STANDARD}
```

To retrieve manual computer groups, define a filter object using the syntax:

```
{type: MANUAL}
```

For example, the following request filters and returns computer groups based on type, with the specific type defined in the variable definition:

```
query getGroupsByType($type: ComputerGroupType) {
  computerGroups(filter: {type: $type}) {
    edges {
      node {
        id
        name
        contentSet {
          id
          name
        }
        expression
        type
        filterEnabled
        managementRightsEnabled
      }
    }
  }
}
```

To retrieve manual computer groups, include the following type variable in the **Variables** panel or in your variables dictionary:

```
{
  "type": "MANUAL"
}
```

To retrieve standard computer groups, include the following type variable in the **Variables** panel or in your variables dictionary:

```
{
  "type": "STANDARD"
}
```

©2026 Tanium Inc. All Rights Reserved

Filter group

You can filter based on whether the computer group is a filter group. To retrieve filter computer groups, define a filter object using the syntax:

```
{filterEnabled: true}
```

For example, the following request filters and returns filter groups:

```
query getFilterGroups($filterEnabled: Boolean) {
  computerGroups(filter: {filterEnabled: $filterEnabled}) {
    edges {
      node {
        id
        name
        contentSet {
          id
          name
        }
        expression
        type
        filterEnabled
        managementRightsEnabled
      }
    }
  }
}
```

Include the filter enabled variable in the **Variables** panel or in your variables dictionary:

```
{
  "filterEnabled": true
}
```

Management rights group

You can filter based on whether the computer group is a management rights group. To retrieve management rights computer groups, define a filter object using the syntax:

```
{managementRightsEnabled: true}
```

For example, the following request filters and returns management rights groups:

```
query getMRGroups($mrEnabled: Boolean) {
  computerGroups(filter: {managementRightsEnabled: $mrEnabled}) {
    edges {
      node {
        id
        name
        contentSet {
          id
          name
        }
```

©2026 Tanium Inc. All Rights Reserved

```
        expression
        type
        filterEnabled
        managementRightsEnabled
      }
    }
  }
}
```

Include the filter enabled variable in the **Variables** panel or in your variables dictionary:

```
{
  "mrEnabled": true
}
```

Discover query filter syntax

Discover queries that request labels or interfaces using `discoverLabels` or `discoverInterfaces` support the `path` and `value` filter fields.

For more information on syntax, see `FieldFilter` in the **Documentation Explorer** pane of the query explorer.

Single column

You can filter on field values. Define a filter object using one of the following syntaxes:

```
discoverInterfaces(filter:
  {
    path: "field-path",
    op: "operator-value",
    value: "field-value"
  }
)
```

```
discoverLabels(filter:
  {
    path: "field-path",
    op: "operator-value",
    value: "field-value"
  }
)
```

Use dot notation for sub-fields not at the root level.

For example, the following query retrieves labels with a `createdTime value` greater than `2023-01-01` and returns the labels:

```
query exampleGetLabels {
  discoverLabels(filter: {path: "createdTime", op: GT, value: "2023-01-01"}) {
    edges {
```

©2026 Tanium Inc. All Rights Reserved

```
      node {
        name
        type
        createdTime
        modifiedTime
      }
    }
  }
}
```

Multiple columns

You can filter on multiple Discover-related columns. Define a filter object using one of the following syntaxes:

```
discoverInterfaces(filter:
  {any: boolean,
    filters: [
      {path: "field-path-1", op: "operator-1", value: "filter-value-1"},
      {path: "field-path-2", op: "operator-2", value: "filter-value-2"},
      {path: "field-path-n", op: "operator-n", value: "filter-value-n"}
    ]
  }
)
```

```
discoverLabels(filter:
  {any: boolean,
    filters: [
      {path: "field-path-1", op: "operator-1", value: "filter-value-1"},
      {path: "field-path-2", op: "operator-2", value: "filter-value-2"},
      {path: "field-path-n", op: "operator-n", value: "filter-value-n"}
    ]
  }
)
```

Use dot notation for sub-fields not at the root level.

For example, the following query retrieves interfaces associated with the label `Example_label` and an osPlatform of `Linux`, and returns matching interfaces:

```
query exampleGetInterfaces {
  discoverInterfaces(
    filter: {any: false, filters: [{path: "labels.name", op: EQ, value: "Example_label"},
{path: "osPlatform", op: EQ, value: "Linux"}]}
  ) {
    edges {
      node {
        id
        labels
        osPlatform
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Endpoint query filter syntax

Endpoint queries support the following filter fields:

- `memberOf`
- `path` and `value`
- `sensor` and `value`

> **Note:** For the best results, when filtering using `sensor` and `value`, if `query.sensors.edges.node.endpointQueryPaths` contains a path value, create a filter using `path` set equal to the `endpointQueryPaths` value for improved stability and performance.

Endpoint queries with `sensorReadings` support the `column` and `value` filter fields to filter based on sensor results.

For more information on syntax, see `endpointFieldFilter` in the **Documentation Explorer** pane of the query explorer.

Fields associated with a sensor (`path` and `value`)

For endpoint queries, you can filter on field values for fields associated with a sensor in the Gateway schema. Unless otherwise specified, define a filter object using the syntax:

```
{path: "field-path", op: "EndpointFieldFilter-value", value: "field-value"}
```

Note the following:

- If you define a field not associated with a sensor, the request returns an error.
- Use dot notation for sub-fields not at the root level. For example, to filter on service names, define `path: "services.name"`.
- The `UPDATED_AFTER` and `READ_AFTER` operators apply only to fields associated with sensors, such as any of the `services` child fields.
- All other operators apply to fields associated with a single-column sensor, or a single column

©2026 Tanium Inc. All Rights Reserved

of a multi-column sensor.

For example, the following request filters and returns only endpoints whose primary user email address is `user@example.com`:

```
query getEndpointsByEmail ($path:String, $value:String){
  endpoints(filter: {path: $path, value: $value}) {
    edges {
      node {
        id
        primaryUser {
          email
        }
      }
    }
  }
}
```

Include the path and value variables in the **Variables** panel:

```
{
  "path": "primaryUser.email",
  "value": "user@example.com"
}
```

Computer group membership

You can filter on membership in named computer groups. To filter by computer group name, define a filter object using the syntax:

```
{memberOf: {name: "computer-group-name"}}
```

To filter by computer group ID, define a filter object using the syntax:

```
{memberOf: {id: "computer-group-id"}}
```

For example, the following request filters and returns only endpoints that are members of the All Linux computer group, based on name:

```
query getLinuxEndpoints ($cgname: String){
  endpoints(filter: {memberOf: {name: $cgname}}) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the computer group name variable in the **Variables** panel:

```
{
```

©2026 Tanium Inc. All Rights Reserved

```
  "cgname": "All Linux"
}
```

The following request filters and returns only endpoints that are members of the computer group with the ID of `123`:

```
query filterEndpointsComputerGroupID ($id: ID){
  endpoints(filter: {memberOf: {id: $id}}) {
    edges {
      node {
        computerID
                name
                ipAddress
      }
    }
  }
}
```

Include the computer group ID variable in the **Variables** panel:

```
{
  "id": 123
}
```

Endpoint IPv4 addresses within a range

Endpoint queries can filter based on the IPv4 address falling within a range of IPv4 addresses. Define a filter object using the syntax:

```
{
  filters: [
    {path: "ipAddress", op: GTE, value: lower-ipv4-address},
    {path: "ipAddress", op: LTE, value: upper-ipv4-address}
  ]
}
```

For example, the following request filters based on IPv4 addresses between 192.0.2.0 and 192.0.2.255:

```
query endpointsInIpRange($ipv4RangeStart: String!, $ipv4RangeEnd: String!) {
  endpoints(
    filter: {filters: [{path: "ipAddress", op: GTE, value: $ipv4RangeStart}, {path:
"ipAddress", op: LTE, value: $ipv4RangeEnd}]}
  ) {
    edges {
      node {
        id
        name
        ipAddress
      }
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the IPv4 address variables in the **Variables** panel:

```
{
  "ipv4RangeStart": "192.0.2.0",
  "ipv4RangeEnd": "192.0.2.255"
}
```

Endpoints first seen or last seen

Endpoint queries can filter on endpoint IDs, then return the first seen or last seen time. To filter based on the endpoint ID value, define a filter object using the syntax:

```
{path: "id", value: "endpoint-id", op: "FieldFilterOp-value"}
```

For ID fields, you can use the `EQ`, `LT`, `LTE`, `GT`, or `GTE` values for the `op` argument.

In the request, define the `query.endpoints.edges.node.eidLastSeen` or `query.endpoints.edges.node.eidFirstSeen` field to determine the time the endpoint was last seen or first seen, respectively.

For example, the following request filters based on endpoint ID 123, and returns the `eidLastSeen` value in the response:

```
query getEndpointsLastSeen ($path:String, $value:String, $op: FieldFilterOp!){
  endpoints(filter: {path: $path, value: $value, op: $op}) {
    edges {
      node {
        id
        eidLastSeen
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel:

```
{
  "path": "id",
  "value": "123",
  "op": "EQ"
}
```

Endpoint queries can also filter on the time an endpoint was first seen or last seen. To filter based on the `eidFirstSeen` value, define a filter object using the syntax:

```
{path: "eidFirstSeen", value: "time-value", op: "FieldFilterOp-value"}
```

To filter based on the `eidLastSeen` value, define a filter object using the syntax:

©2026 Tanium Inc. All Rights Reserved

```
{path: "eidLastSeen", value: "time-value", op: "FieldFilterOp-value"}
```

For time fields, you can use the `EQ`, `LT`, `LTE`, `GT`, `GTE`, `UPDATED_AFTER`, or `READ_AFTER` values for the `op` argument.

For example, the following request filters based on endpoints last seen after January 1, 2022:

```
query getEndpointsLastSeen ($path:String, $value:String, $op: FieldFilterOp!){
  endpoints(filter: {path: $path, value: $value, op: $op}) {
    edges {
      node {
        id
        eidLastSeen
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel:

```
{
  "path": "eidLastSeen",
  "value": "2022-01-01T00:00:00.000Z",
  "op": "GT"
}
```

Sensors

Endpoint queries can filter on readings of arbitrary named sensors. Define a filter object using the syntax:

```
 {sensor: {name: "sensor-name", value: "filter-value"}}
```

Note: For the best results, when filtering using `sensor` and `value`, if `query.sensors.edges.node.endpointQueryPaths` contains a path value, create a filter using `path` set equal to the `endpointQueryPaths` value for improved stability and performance.

For example, the following request filters based on the Total Memory sensor returning a value of `16384 MB` for endpoints, and returns only those endpoints:

```
query get16GBMemEndpoints($sensor: String, $value: String) {
  endpoints(filter: {sensor: {name: $sensor}, value: $value}) {
    edges {
      node {
        id
      }
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Include the sensor name and value variables in the **Variables** panel:

```
{
  "sensor": "Total Memory",
  "value": "16384 MB"
}
```

Sensor values updated after a given date

Endpoint queries can filter on sensor values updated after a given date. Define a filter object using the syntax:

```
 {sensor: {name: "sensor-name", op: UPDATED_AFTER, value: "rfc-3339-timestamp"}}
```

For example, the following request filters based on whether any endpoints had their Total Memory sensor value updated after January 1, 2022:

```
query get16GBMemEndpoints($sensor: String, $value: String) {
  endpoints(filter: {sensor: {name: $sensor}, op: UPDATED_AFTER, value: $value}) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel:

```
{
  "sensor": "Total Memory",
  "value": "2022-01-01T00:00:00.000Z"
}
```

Sensor maximum age

Endpoint queries that filter using sensor values can override the sensor Max Sensor Age setting, and force sensor reevaluations if the cached sensor results are older than this setting. Define a filter object using the syntax:

```
 {sensor: {name: "sensor-name", op: operator-value, value: "column-value"}, maxAge: integer-
seconds}
```

©2026 Tanium Inc. All Rights Reserved

Note: You can only use `maxAge` with the Tanium Server (`ts`) data source. The response returns a validation error if you try to use `maxAge` with the Tanium Data Service (`tds`) data source.

Note: Setting this value to very small periods of time might cause excessive resource usage on endpoints and frequent sensor evaluation.

For example, the following request filters based on endpoints with 8000 MB of total memory, and forces sensor script reexecution if the cached results are older than 300 seconds:

```
query exampleGet8GBMaxAgeEndpoints($sensor: String, $value: String, $maxAge: Int) {
  endpoints(
    source: {ts: {stableWaitTime: 10}}
    filter: {sensor: {name: $sensor}, op: EQ, value: $value, maxAge: $maxAge}
  ) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the filter variables in the **Variables** panel:

```
{
  "sensor": "Total Memory",
  "value": "8000 MB",
  "maxAge": 300
}
```

Parameterized sensors

You can also filter on parameterized sensors. Define a filter object using the syntax:

```
{sensor:
  {name: "sensor-name",
  params:[
    {name: "parameter-1-name", value: "parameter-1-value"},
    {name: "parameter-2-name", value: "parameter-2-value"},
    {name: "parameter-n-name", value: "parameter-n-value"}
  ]
},
  value: "filter-value"
}
```

Add a `params` `name`/`value` object for every parameter.

©2026 Tanium Inc. All Rights Reserved

For example, the following request filters based on the Custom Tag Exists sensor, returning endpoints tagged with the custom tag `the-tag`, and returns only those endpoints:

```
query getEndpointsWithCustomTag($sensorName: String, $sensorValue: String, $paramName:
String!, $paramValue: String!) {
  endpoints(
    filter: {sensor: {name: $sensorName, params: [{name: $paramName, value: $paramValue}]},
value: $sensorValue}
  ) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the sensor name, value, and parameter variables in the **Variables** panel:

```
{
  "sensorName": "Custom Tag Exists",
  "sensorValue": "true",
  "paramName": "tag",
  "paramValue": "the-tag"
}
```

Parameterized sensor in the root filter

You can also filter on parameterized sensors in the root filter. Use the same parameters in the root filter and in `sensorReadings` for that sensor.



Note: For TDS queries, parameters must match a parameterized sensor instance that is already registered with Tanium Data Service. If the parameters do not match an existing parameterized sensor in Tanium Data Service, Gateway returns a response that the validation is unregistered or that parameters must be an exact match. Replace the file path with a value your environment has expanded, or ensure that Tanium Data Service has collected sensor results for the matching parameterized sensor. For more information about Tanium Data Service, see Tanium Interact User Guide: Managing sensor results collection through TDS.

For example, the following request returns endpoints where `File Exists` matches the regex for that file path. Use the same file parameter in the root filter and in `sensorReadings`:

```
query exampleParameterFilterRoot($fileParameter: String!) {
  endpoints(
    source: {tds: {excludeNoResults: true, excludeErrors: true}}
    filter: {any: false, filters: [{sensor: {name: "File Exists", params: [{name: "file",
value: $fileParameter}]}, value: ".*File Exists.*", op: MATCHES, negated: false}]}
    first: 500
```

©2026 Tanium Inc. All Rights Reserved

```
  ) {
    totalRecords
    pageInfo {
      hasNextPage
      endCursor
    }
    edges {
      node {
        name
        sensorReadings(
          sensors: [{name: "Short Hostname"}, {name: "File Exists", params: [{name: "file",
value: $fileParameter}]}]
        ) {
          columns {
            name
            values
          }
        }
      }
    }
  }
}
```

Include the file name variable in the **Variables** panel:

```
{
  "fileParameter": "C:\\nobackup\\example.txt"
}
```

Field path

You can filter endpoints by a field path. For example the following query filters to show only endpoints whose `os.generation` contains `Windows`:

```
query exampleWindowsFilter($parameterPath: String, $parameterPathValue: String) {
  endpoints(
    first: 20
    filter: {path: $parameterPath, op: CONTAINS, value: $parameterPathValue}
  ) {
    totalRecords
    edges {
      node {
        name
        os {
          generation
        }
      }
    }
  }
}
```

Include the path and value variable in the **Variables** panel:

```
{
  "parameterPath": "os.generation",
  "parameterPathValue": "Windows"
}
```

©2026 Tanium Inc. All Rights Reserved

Sensor readings

You can filter endpoints by sensor readings. For example, the following query returns endpoints whose `CPU Details` has `Total Cores` greater than or equal to `2`, and also retrieves `CPU Speed`:

```
query exampleEndpointsSensorReadings($sensorName: String!, $column1: String!, $column1Value:
String, $column2: String!) {
  endpoints(first: 50, source: {tds: {}}) {
    edges {
      node {
        name
        sensorReadings(
          sensors: [{name: $sensorName, columns: [$column1, $column2], filter: {column:
$column1, op: GTE, value: $column1Value, any: false}}]
        ) {
          columns {
            name
            values
          }
        }
      }
    }
  }
}
```

Include the sensor and column variables in the **Variables** panel:

```
{
  "sensorName": "CPU Details",
  "column1": "Total Cores",
  "column1Value": "2",
  "column2": "CPU Speed"
}
```

Root filter sensor row match

Many times, the match value is a row key such as `Application Name|Version`. You can build this query from a prior `sensorReadings` query for `Installed Applications` on a known endpoint.

For example, the following query returns endpoints where `Installed Applications` contains a specific sensor name and value:

```
query exampleRootFilter($sensorName: String, $sensorValue: String) {
  endpoints(
    first: 20
    source: {tds: {excludeNoResults: true, excludeErrors: true}}
    filter: {sensor: {name: $sensorName}, op: CONTAINS, value: $sensorValue}
  ) {
    totalRecords
    edges {
```

©2026 Tanium Inc. All Rights Reserved

```
      node {
        name
      }
    }
  }
}
```

Include the sensor variables in the **Variables** panel:

```
{
  "sensorName": "Installed Applications",
  "sensorValue": "Google Chrome|145.0.7632.160"
}
```

Single column from multi-column sensors

You can filter a column from multi-column sensors by specifying the column name. Define a filter object using the syntax:

```
{sensor: {
  name: "sensor-name",
  column: "sensor-column-name"},
  value: "filter-value"}
}
```

For example, the following request filters based on the CPU Details sensor returning endpoints with an `Intel` CPU, and returns only those endpoints:

```
query getIntelCPUEndpoints($sensorName: String, $sensorColumn: String, $sensorValue: String)
{
  endpoints(
    filter: {sensor: {name: $sensorName, column: $sensorColumn}, value: $sensorValue}
  ) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the sensor name, column, and value variables in the **Variables** panel:

```
{
  "sensorName": "CPU Details",
  "sensorColumn": "CPU",
  "sensorValue": "Intel"
}
```

Multiple columns from multi-column sensors

If you want to select across multiple columns of each row of a multi-column sensor, define a

©2026 Tanium Inc. All Rights Reserved

`filter.filters` list, then define a list element for each column filter, using the following syntax:

```
filters: [
  {sensor: {column: "column-1-name"}, value: "column-1-value"},
  {sensor: {column: "column-2-name"}, value: "column-2-value"},
  {sensor: {column: "column-n-name"}, value: "column-n-value"}
]
```

You can only use a single level of child filters, and can only use the default `EQ` operator.

For example, the following request filters based on the CPU Details sensor returning endpoints with `4` total cores and an `Intel` CPU, and returns only those endpoints:

```
query get4CoreIntelCPUEndpoints($sensorName: String, $sensorColumn1: String, $sensorValue1:
String, $sensorColumn2: String, $sensorValue2: String) {
  endpoints(
    filter: {sensor: {name: $sensorName}, filters: [{sensor: {column: $sensorColumn1}, value:
$sensorValue1}, {sensor: {column: $sensorColumn2}, value: $sensorValue2}]}
  ) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the sensor name, column, and value variables in the **Variables** panel:

```
{
  "sensorName": "CPU Details",
  "sensorColumn1": "CPU",
  "sensorValue1": "Intel",
  "sensorColumn2": "Total Cores",
  "sensorValue2": "4"
}
```

In this form, the sensor is identified in the top filter and the columns are identified in the child filters. You can only use a single level of child filters, and can only use the default `EQ` operator.

Sensor readings from one or multiple sensors

After you filter endpoints, you can filter your results to return data from additional sensors. As a child of the node object, define the sensor reading filter using the following syntax:

```
sensorReadings
  (sensors: [{name: "sensor-name"}], includeHiddenColumns: boolean) {
    columns {
      sensor {
        name
      }
      name
      values
```

©2026 Tanium Inc. All Rights Reserved

```
    }
```

For example, the following request first filters endpoints and only returns endpoints with `Firefox` as an installed application, then for each endpoint, returns the results of the Is Linux sensor:

```
query GetLinuxFirefoxEndpoints($sensorName1: String, $sensorColumn1: String, $sensorOp1:
FieldFilterOp!, $sensorValue1: String, $sensorName2: String!) {
  endpoints(
    filter: {sensor: {name: $sensorName1, column: $sensorColumn1}, op: $sensorOp1, value:
$sensorValue1}
  ) {
    edges {
      node {
        name
        ipAddress
        sensorReadings(sensors: [{name: $sensorName2}]) {
          columns {
            sensor {
              name
            }
            name
            values
          }
        }
      }
    }
  }
}
```

Include the sensor name, column, operator, and value variables in the **Variables** panel:

```
{
  "sensorName1": "Installed Applications",
  "sensorColumn1": "Name",
  "sensorOp1": "CONTAINS",
  "sensorValue1": "Firefox",
  "sensorName2": "Is Linux"
}
```

Single column from multi-column sensor readings

You can filter a column from sensor readings for multi-column sensors by specifying the column name. Define a filter object using the syntax:

```
sensorReadings
  (sensors: [{name: "sensor-name", filter: {column: "column-name", op: "operator-value",
value: "column-value", restrictOwner: "boolean-value"}}}], includeHiddenColumns: boolean) {
    columns {
      sensor {
        name
      }
      name
      values
    }
  }
```

©2026 Tanium Inc. All Rights Reserved

You can filter the set of endpoints returned and filter based on the sensor readings in a single request.

For example, the following request retrieves endpoints, then filters based on the Installed Applications sensor returning endpoints with the `tar` application installed, and is limited to the returned field values, then returns only those endpoints:

```
query getEndpointsInstalledTar ($first: Int) {
  endpoints (first: $first){
    edges {
      node {
        computerID
        sensorReadings(
          sensors: [{name: "Installed Applications", filter: {column: "Name", op: EQ, value:
"tar", restrictOwner: false}}]
        ) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

Multiple columns from multi-column sensor readings

If you want to select across multiple columns of each row of multi-column sensor readings, define a `filter.filters` list, then define a list element for each column filter, using the following syntax:

```
sensorReadings(
  sensors: [
    {
      name: "sensor-name",
      filter: {
        restrictOwner: "boolean-value",
        filters: [
         {column: "column-name-1", op: "operator-value-1", value: "column-value-1"},
          {column: "column-name-2", op: "operator-value-2", value: "column-value-2"},
         {column: "column-name-n", op: "operator-value-n", value: "column-value-n"}]
      }
    }
  ], includeHiddenColumns: boolean
) {
    columns {
```

©2026 Tanium Inc. All Rights Reserved

```
      sensor {
        name
      }
      name
      values
    }
  }
```

You can only use a single level of child filters. You can filter the set of endpoints returned and filter based on the sensor readings in a single request.

For example, the following query retrieves the first 2 endpoints that are members of the `All Linux` computer group, and returns the application `Name` and `Version` columns for only the endpoints that have the `tar` application version `1.30` installed, based on the results from the Installed Applications sensor.

```
query getLinuxEndpointsInstalledTar130 ($first: Int){
  endpoints(first: $first, filter: {memberOf: {name: "All Linux"}}) {
    edges {
      node {
        computerID
        sensorReadings(
          sensors: [{name: "Installed Applications",
            filter: {
              restrictOwner: false,
              filters: [
              {column: "Name", op: EQ, value: "Tar"},
              {column: "Version" op: EQ, value: "1.30"}]
          }
            }]
        ) {
          columns {
            name
            values
            sensor {
              name
            }
          }
        }
      }
    }
  }
}
```

Include the pagination variable in the **Variables** panel or in your variables dictionary:

```
{
  "first": 2
}
```

You can also define a delimited list of values for filtering. Define a filter object using the syntax:

```
{
  sensor: {
    name: "sensor-name",
  },
```

©2026 Tanium Inc. All Rights Reserved

```
  op: "MATCHES",
  value: "column-value-1|column-value-2|column-value-n..."
}
```

Define the `value` field value as multiple column values, with each value separated by the sensor delimiter, such as an escaped vertical line (`\|`). The delimited list of values does not need to start with the first sensor column. If you want to match on any column value for a column, define a wildcard (`.*`).

Best Practice: For the best results, define a `filter.filters` list, then define a list element for each column filter, as described in the preceding example.

The following request filters based on the Client Extensions - Status sensor, returning only endpoints with the sensor column values of `version` and `1.2.345`:

```
query exampleFilterDelimitedValues($maxWaitTime: Int!, $filter: EndpointFieldFilter) {
  endpoints(
    source: {ts: {maxWaitTime: $maxWaitTime}}
    filter: $filter
  ) {
    totalRecords
    collectionInfo {
      contributedTotal
      respondedTotal
    }
    edges {
      node {
        name
        id
      }
    }
  }
}
```

Include the pagination, wait time, and filter variables in the **Variables** panel:

```
{
  "maxWaitTime": 30,
  "filter": {
    "sensor": {
      "name": "Client Extensions - Status"
    },
    "op": "MATCHES",
    "value": ".*\\|version\\|1\\.2\\.345",
    "negated": false,
    "any": false
  }
}
```

©2026 Tanium Inc. All Rights Reserved

(1.2 - Experimental (Release Candidate)) Sensor reading for one sensor, converted data type

After you filter endpoints, you can filter your results to return data from an additional sensor, and convert the sensor result data type in the response. For example, if a sensor with a string return type returns an integer ID, you can convert the data type to integer in the response. As a child of the node object, define the sensor reading filter using the following syntax:

```
sensorReading
  (sensor: {
    columns: ["column-name"],
    filter: {column: "filter-column-name", op: "operator-value", value: "column-value",
restrictOwner: "boolean-value"},
    maxAge: "maximum-data-age-seconds",
    name: "sensor-name",
    params: [name: "param-name", value: "param-value"]
  }
)
```

For example, the following request first filters endpoints and only returns endpoints with `Firefox` as an installed application, then for each endpoint, returns the results of the Is Linux sensor:

```
query GetLinuxFirefoxEndpoints($sensorName1: String, $sensorColumn1: String, $sensorOp1:
FieldFilterOp!, $sensorValue1: String, $sensorName2: String!, $sensorColumn2: String!) {
  endpoints(
    filter: {sensor: {name: $sensorName1, column: $sensorColumn1}, op: $sensorOp1, value:
$sensorValue1}
  ) {
    edges {
      node {
        name
        ipAddress
        sensorReading(sensor: {name: $sensorName2, columns: [$sensorColumn2]}) {
          columns {
            sensor {
              name
            }
            name
            values
          }
        }
      }
    }
  }
}
```

Include the sensor name, column, operator, and value variables in the **Variables** panel:

```
{
  "sensorName1": "Installed Applications",
  "sensorColumn1": "Name",
  "sensorOp1": "CONTAINS",
  "sensorValue1": "Firefox",
  "sensorName2": "Is Linux",
  "sensorColumn2": "Is Linux"
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

You can define one sensor and sensor column for each instance of `sensorReading`. Use aliases in your request to request information for multiple sensors, and convert the data type as needed. For example, the following request returns values from the Custom Tags sensor, without converting the data type, and the Total Cores numeric value from the CPU Details sensor, converted to float values:

```
query endpointWithSensorReading {
  endpoints {
    edges {
      node {
        id
        customTags: sensorReading(sensor: {name: "Custom Tags"}) {
          values
        }
        totalCpuCores: sensorReading(
          sensor: {name: "CPU Details", columns: ["Total Cores"]}
        ) {
          floatValues
        }
      }
    }
  }
}
```

Package spec query filter syntax

Package spec queries support the `path` and `value` filter fields.

For more information on syntax, see `FieldFilter` in the **Documentation Explorer** pane of the query explorer.

Single filter field

You can filter on a single field value. Unless otherwise specified, define a filter object using the syntax:

```
{path: "field-path", negated: "boolean", op: "operator", value: "filter-value"}
```

For example, the following request filters and returns only package spec objects whose name does not contain `production`:

```
query getPackageSpecsFiltered($path: String, $negated: Boolean!, $op: FieldFilterOp!, $value:
String) {
  packageSpecs(filter: {path: $path, negated: $negated, op: $op, value: $value}) {
    edges {
      node {
        id
        name
```

©2026 Tanium Inc. All Rights Reserved

```
        contentSetName
        command
        commandTimeoutSeconds
        expireSeconds
        params {
          key
          label
          defaultValue
        }
        sensorSourcedParams {
          name
        }
      }
    }
  }
}
```

Include the filter-related variables in the **Variables** panel:

```
{
  "path": "name",
  "negated": true,
  "op": "CONTAINS",
  "value": "production"
}
```

Multiple filter fields

You can filter on multiple field values. Define a filter object using the syntax:

```
{any: boolean,
  filters: [
    {path: "field-path-1", negated: "boolean-1", op: "operator-1", value: "filter-value-1"},
    {path: "field-path-2", negated: "boolean-2", op: "operator-2", value: "filter-value-2"},
    {path: "field-path-n", negated: "boolean-n", op: "operator-n", value: "filter-value-n"}
  ]
```

For example, the following request filters and returns only package spec objects whose name does not contain `production`, and whose content set name contains `test`:

```
query packageSpecsMultipleFilters($path1: String, $negated1: Boolean!, $op1: FieldFilterOp!,
$value1: String, $path2: String, $negated2: Boolean!, $op2: FieldFilterOp!, $value2: String)
{
  packageSpecs(
    filter: {any: false, filters: [{path: $path1, negated: $negated1, op: $op1, value:
$value1}, {path: $path2, negated: $negated2, op: $op2, value: $value2}]}
  ) {
    edges {
      node {
        id
        name
        contentSetName
        command
        commandTimeoutSeconds
        expireSeconds
        params {
          key
```

©2026 Tanium Inc. All Rights Reserved

```
          label
          defaultValue
        }
        sensorSourcedParams {
          name
        }
      }
    }
  }
}
```

Include the filter-related variables in the **Variables** panel:

```
{
  "path1": "name",
  "negated1": true,
  "op1": "CONTAINS",
  "value1": "production",
  "path2": "contentSetName",
  "negated2": false,
  "op2": "CONTAINS",
  "value2": "test"
}
```

Patch query filter syntax

Patch queries to retrieve patch definitions support the following filter field:

- `cveIds`
- `classification`
- `hasCve`
- `isSuperseded`
- `platform`
- `product`
- `releasedOnOrBeforeDate`
- `releasedOnOrAfterDate`
- `severity`

For more information on syntax, see `patchDefinitionFieldFilter` in the **Documentation Explorer** pane of the query explorer.

©2026 Tanium Inc. All Rights Reserved

CVE ID

You can filter patch definitions based on an associated CVE ID. Define a filter object using the syntax:

```
{cveIds: [cve-id]}
```



Note: The filter supports one, and only one, CVE ID to filter the returned patch definitions.

For example, the following request filters and returns only patch definitions with a CVE ID of `CVE-2014-123456`:

```
query exampleGetPatchDefinitions($first: Int, $filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(first: $first, filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "first": 1,
  "filter": {
    "cveIds": [
      "CVE-2014-123456"
    ]
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Classification

©2026 Tanium Inc. All Rights Reserved

You can filter patch definitions based on a patch definition classification. Define a filter object using the syntax:

```
{classification: classification-type}
```

For example, the following request filters and returns only patch definitions with a classification of `Security Updates`:

```
query exampleGetPatchDefinitions($filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        classification
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "classification": [
      "Security Updates"
    ]
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Has association with any CVE

You can filter patch definitions based on whether the patch definition is associated with any CVE. Define a filter object using the syntax:

```
{hasCve: boolean-value}
```

©2026 Tanium Inc. All Rights Reserved

For example, the following request filters and returns only patch definitions associated with any CVE:

```
query exampleGetPatchDefinitions($filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        classification
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "hasCve": true
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Is superseded by another patch definition

You can filter patch definitions based on whether the patch definition is superseded by another patch definition. Define a filter object using the syntax:

```
{isSuperseded: boolean-value}
```

For example, the following request filters and returns only patch definitions superseded by another patch definition:

```
query exampleGetPatchDefinitions($filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(filter: $filter, sort: $sort) {
    edges {
      node {
        id
```

©2026 Tanium Inc. All Rights Reserved

```
        createdDate
        classification
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "isSuperseded": true
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Operating System platform

You can filter patch definitions based on whether the patch definition applies to an operating system. Define a filter object using the syntax:

```
{platform: PatchEndpointPlatform-enum-value}
```

Supported values include `MACOS`, `LINUX`, and `WINDOWS`.

For example, the following request filters and returns only patch definitions that apply to Linux operating systems:

```
query exampleGetPatchDefinitions($filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        classification
        cveIds
        isSuperseded
        platform
        releaseDate
```

©2026 Tanium Inc. All Rights Reserved

```
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "platform": "LINUX"
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Product

You can filter patch definitions based on the product to which the patch definition applies to an operating system. Define a filter object using the syntax:

```
{product: product-name}
```

For example, the following request filters and returns only patch definitions for Windows Server 2019:

```
query exampleGetPatchDefinitions($filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        classification
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
```

©2026 Tanium Inc. All Rights Reserved

```
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "product": "Windows Server 2019"
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Released on or before a date

You can filter patch definitions based on the patch definition release date occurring prior to or equal to a date in ISO 8601 format. Define a filter object using the syntax:

```
{releasedOnOrBeforeDate: iso-8601-date}
```

For example, the following request filters and returns only patch definitions released before or on December 31, 2020:

```
query exampleGetPatchDefinitions($filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        classification
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

©2026 Tanium Inc. All Rights Reserved

```
{
  "filter": {
    "releasedOnOrBeforeDate": "2020-12-31"
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Released on or after a date

You can filter patch definitions based on the patch definition release date occurring on or after a date in ISO 8601 format. Define a filter object using the syntax:

```
{releasedOnOrAfterDate: iso-8601-date}
```

For example, the following request filters and returns only patch definitions released on or after December 31, 2023:

```
query exampleGetPatchDefinitions($filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        classification
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "releasedOnOrAfterDate": "2023-12-31"
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Severity

You can filter patch definitions based on the patch definition severity. Define a filter object using the syntax:

```
{severity: severity-value}
```

The filter comparison is case-insensitive. Supported severity values include `critical`, `important`, `low`, `medium`, `moderate`, and `none`

For example, the following request filters and returns only patch definitions assigned a `medium` severity:

```
query exampleGetPatchDefinitions($filter: PatchDefinitionFieldFilter, $sort:
PatchDefinitionFieldSort) {
  patchDefinitions(filter: $filter, sort: $sort) {
    edges {
      node {
        id
        createdDate
        classification
        cveIds
        isSuperseded
        platform
        releaseDate
        severity
        sizeInBytes
        title
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the pagination and filter variables in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "severity": "medium"
  },
  "sort": {
    "field": "releaseDate",
    "order": "desc"
  }
}
```

Provision query filter syntax

©2026 Tanium Inc. All Rights Reserved

Provision queries that request retirement action information (`query.provisionRetirementActions`) support the following filter fields:

- `computerName`
- `manufacturer`
- `model`
- `serialNumber`
- `status`

Provision queries that request retirement action information support sorting the response on the following filter fields in ascending (`asc`) or descending (`desc`) order:

- `computerName` based on alphabetical order
- `createdAt` based on chronological order
- `manufacturer` based on alphabetical order
- `model` based on alphabetical order
- `retirementActionDefinition` based on alphabetical order
- `serialNumber` based on alphanumerical order
- `status` based on alphabetical order
- `updatedAt` based on chronological order

If you do not define a sort order, the query defaults to sorting on `updatedAt` in `asc` order.

For more information on filter syntax, see `ProvisionFieldFilter` and `ProvisionFilterField` in the **Documentation Explorer** pane of the query explorer. For more information on sort syntax, see `ProvisionRetirementActionFieldSort` and `ProvisionRetirementActionSortField` in the **Documentation Explorer** pane of the query explorer.

Retirement actions targeting a manufacturer

©2026 Tanium Inc. All Rights Reserved

You can retrieve retirement actions that target a manufacturer, including partial matches. Define a filter object with the following syntax:

```
{
  filterField: manufacturer,
  value: filter-value
}
```

For example, the following query retrieves retirement actions that target manufacturers that contain `example`:

```
query exampleGetMultipleProvisionRetirementActions($filter: [ProvisionFieldFilter!]) {
  provisionRetirementActions(filter: $filter) {
    edges {
      cursor
      node {
        id
        definitionID
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variable in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "filterField": "manufacturer",
    "value": "example"
  }
}
```

Retirement actions targeting a hardware model

You can retrieve retirement actions that target a hardware model, including partial matches. Define a filter object with the following syntax:

```
{
  filterField: model,
  value: filter-value
}
```

For example, the following query retrieves retirement actions that target hardware models that contain `example`:

```
query exampleGetMultipleProvisionRetirementActions($filter: [ProvisionFieldFilter!]) {
  provisionRetirementActions(filter: $filter) {
```

©2026 Tanium Inc. All Rights Reserved

```
    edges {
      cursor
      node {
        id
        definitionID
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variable in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "filterField": "model",
    "value": "example"
  }
}
```

Retirement actions targeting an endpoint serial number

You can retrieve retirement actions that target an endpoint serial number, including partial matches. Define a filter object with the following syntax:

```
{
  filterField: serialNumber,
  value: filter-value
}
```

For example, the following query retrieves retirement actions that target endpoint serial numbers that contain `example`:

```
query exampleGetMultipleProvisionRetirementActions($filter: [ProvisionFieldFilter!]) {
  provisionRetirementActions(filter: $filter) {
    edges {
      cursor
      node {
        id
        definitionID
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the variable in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "filterField": "serialNumber",
    "value": "example"
  }
}
```

Retirement actions that are at a specific lifecycle status

You can retrieve retirement actions that target a specific lifecycle status, including partial matches. Retirement action statuses include `pending`, `issued`, `information`, `in progress`, `canceled`, and `completed`. Define a filter object with the following syntax:

```
{
  filterField: status,
  value: filter-value
}
```

For example, the following query retrieves retirement actions at a status of `pending`:

```
query exampleGetMultipleProvisionRetirementActions($filter: [ProvisionFieldFilter!]) {
  provisionRetirementActions(filter: $filter) {
    edges {
      cursor
      node {
        id
        definitionID
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variable in the **Variables** panel or in your variables dictionary:

```
{
  "filter": {
    "filterField": "status",
    "value": "pending"
  }
}
```

Retirement action results sorting

You can sort retirement action results. Define a sort object with the following syntax:

©2026 Tanium Inc. All Rights Reserved

```
sort: {
    field: field-name,
    order: sort-order
}
```

For example, the following query sorts retirement actions in descending order based on computer name:

```
query exampleGetMultipleProvisionRetirementActions($sort:
ProvisionRetirementActionFieldSort!) {
  provisionRetirementActions(sort: $sort) {
    edges {
      cursor
      node {
        id
        definitionID
      }
    }
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    totalRecords
  }
}
```

Include the variables in the **Variables** panel or in your variables dictionary:

```
{
  "sort": {
    "field": "computerName",
    "order": "desc"
  }
}
```

Reporting query filter syntax

Reporting queries support the following filter fields:

- `authorNames`
- `contentSetNames`
- `createdAfter`
- `createdBefore`
- `labels`

©2026 Tanium Inc. All Rights Reserved

- `moduleNames`
- `text`

For more information on syntax, see `reportFieldFilter` in the **Documentation Explorer** pane of the query explorer.

Report author

You can filter reports based on the author name. Define a filter object using the syntax:

```
{authorNames: [name-1, name-2, name-...n]}
```

For example, the following request filters and returns only report definitions whose author is either `testadmin-1` or `testadmin-2`:

```
query exampleGetReportsByAuthor($author1: String!, $author2: String!) {
  reports(filter: {authorNames: [$author1, $author2]}) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the count and time variables in the **Variables** panel or in your variables dictionary:

```
{
  "author1": "testadmin-1",
  "author2": "testadmin-2"
}
```

Content set membership

You can filter reports based on the report content set membership. Define a filter object using the syntax:

```
{contentSetNames: [content-set-name-1, content-set-name-2, content-set-name-...n]}
```

If you define multiple solutions, a report only needs to match one of the content sets to pass the filter.

For example, the following request filters and returns only report definitions belonging to the Base or Default content sets:

```
query exampleGetReportsByContentSet($contentSet1: String!, $contentSet2: String!) {
  reports(filter: {contentSetNames: [$contentSet1, $contentSet2]}) {
```

©2026 Tanium Inc. All Rights Reserved

```
    edges {
      node {
        id
      }
    }
  }
}
```

Include the content set variables in the **Variables** panel or in your variables dictionary:

```
{
  "contentSet1": "Base",
  "contentSet2": "Default"
}
```

Creation time

You can filter reports based on the report creation time, either before or after a specified time. To filter reports created before a specified time, define a filter object using the syntax:

```
{createdBefore: rfc-3339-time}
```

For example, the following request filters and returns only report definitions created before 2022-01-01T00:00:00Z:

```
query exampleGetReportsBefore($timeBefore: Time) {
  reports(filter: {createdBefore: $timeBefore}) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the timestamp variable in the **Variables** panel or in your variables dictionary:

```
{
  "timeBefore": "2022-01-01T00:00:00Z"
}
```

To filter reports created after a specified time, define a filter object using the syntax:

```
{createdAfter: rfc-3339-time}
```

For example, the following request filters and returns only report definitions created after 2022-01-01T00:00:00Z:

```
query exampleGetReportsAfter($timeAfter: Time) {
  reports(filter: {createdAfter: $timeAfter}) {
    edges {
      node {
        id
```

©2026 Tanium Inc. All Rights Reserved

```
      }
    }
  }
}
```

Include the timestamp variable in the **Variables** panel or in your variables dictionary:

```
{
  "timeAfter": "2022-01-01T00:00:00Z"
}
```

Report labels

You can filter reports based on the associated labels. Define a filter object using the syntax:

```
{labels: [label-1, label-2, label-...n]}
```

If you define multiple labels, a report only needs to match one of the labels to pass the filter.

For example, the following request filters and returns only report definitions with a label of `testlabel-1` or `testlabel-2`:

```
query exampleGetReportsByLabel($label1: String!, $label2: String!) {
  reports(filter: {labels: [$label1, $label2]}) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the label variables in the **Variables** panel or in your variables dictionary:

```
{
  "label1": "testlabel-1",
  "label2": "testlabel-2"
}
```

Associated Tanium solution

You can filter reports based on the associated Tanium solution. Define a filter object using the syntax:

```
{moduleNames: [solution-1, solution-2, solution-...n]}
```

If you define multiple solutions, a report only needs to match one of the solutions to pass the filter.

For example, the following request filters and returns only report definitions associated with Tanium Interact or Tanium Asset:

```
query exampleGetReportsBySolution($solution1: String!, $solution2: String!) {
```

©2026 Tanium Inc. All Rights Reserved

```
  reports(filter: {moduleNames: [$solution1, $solution2]}) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the solution name variables in the **Variables** panel or in your variables dictionary:

```
{
  "solution1": "Tanium Interact",
  "solution2": "Tanium Asset"
}
```

Name or description text

You can filter reports based on text in the report name or report description. Define a filter object using the syntax:

```
{text: text-string}
```

For example, the following request filters and returns only report definitions that have `test-string` either in the name or the definition:

```
query exampleGetReportsByText($text: String) {
  reports(filter: {text: $text}) {
    edges {
      node {
        id
      }
    }
  }
}
```

Include the text variable in the **Variables** panel or in your variables dictionary:

```
{
  "text": "test-string"
}
```

Scheduled actions query filter syntax

Scheduled actions queries (`query.scheduledActions`) support the `path` and `value` filter fields.

For more information on syntax, see `FieldFilter` in the **Documentation Explorer** pane of the query explorer.

©2026 Tanium Inc. All Rights Reserved

Single filter field

You can filter on a single field value. Unless otherwise specified, define a filter object using the syntax:

```
{path: "field-path", negated: "boolean", op: "operator", value: "filter-value"}
```

For example, the following request filters and returns only scheduled actions whose name does not contain `production`:

```
query getScheduledActionsFiltered($path: String, $negated: Boolean!, $op: FieldFilterOp!,
$value: String) {
  scheduledActions(
    filter: {path: $path, negated: $negated, op: $op, value: $value}
  ) {
    edges {
      node {
        comment
        id
        name
        nextStartTime
        package {
          id
          name
          params
          sensorSourcedParams {
            name
            value
          }
        }
        targets {
          actionGroup {
            id
            name
          }
          targetGroup {
            id
            name
          }
        }
      }
    }
  }
}
```

Include the filter-related variables in the **Variables** panel:

```
{
  "path": "name",
  "negated": true,
  "op": "CONTAINS",
  "value": "production"
}
```

©2026 Tanium Inc. All Rights Reserved

Multiple filter fields

You can filter on multiple field values. Define a filter object using the syntax:

```
{any: boolean,
  filters: [
    {path: "field-path-1", negated: "boolean-1", op: "operator-1", value: "filter-value-1"},
    {path: "field-path-2", negated: "boolean-2", op: "operator-2", value: "filter-value-2"},
    {path: "field-path-n", negated: "boolean-n", op: "operator-n", value: "filter-value-n"}
  ]
```

For example, the following request filters and returns only scheduled actions whose name does not contain `production`, and whose comment contains `test`:

```
query scheduledActionsMultipleFilter($path1: String, $negated1: Boolean!, $op1:
FieldFilterOp!, $value1: String, $path2: String, $negated2: Boolean!, $op2: FieldFilterOp!,
$value2: String) {
  scheduledActions(
    filter: {any: false, filters: [{path: $path1, negated: $negated1, op: $op1, value:
$value1}, {path: $path2, negated: $negated2, op: $op2, value: $value2}]}
  ) {
    edges {
      node {
        comment
        id
        name
        nextStartTime
        package {
          id
          name
          params
          sensorSourcedParams {
            name
            value
          }
        }
        targets {
          actionGroup {
            id
            name
          }
          targetGroup {
            id
            name
          }
        }
      }
    }
  }
}
```

Include the filter-related variables in the **Variables** panel:

```
{
  "path1": "name",
  "negated1": true,
  "op1": "CONTAINS",
  "value1": "production",
```

©2026 Tanium Inc. All Rights Reserved

```
  "path2": "comment",
  "negated2": false,
  "op2": "CONTAINS",
  "value2": "test"
}
```

Sensor query filter syntax

Sensor queries support the `includeHidden`, `path`, and `value` filter fields.

For more information on syntax, see `FieldFilter` in the **Documentation Explorer** pane of the query explorer.

Hidden sensors

Sensor queries do not include hidden sensors by default. You can filter a request to return both hidden sensors and non-hidden sensors. Unless otherwise specified, define an `includeHidden` argument using the syntax:

```
(includeHidden: "boolean")
```

For example, the following request filters and returns both hidden and non-hidden sensors:

```
query getHiddenAndNonHiddenSensors($first: Int, $hidden: Boolean!) {
  sensors(
    first: $first
    includeHidden: $hidden
  ) {
    edges {
      node {
        hidden
        contentSetName
        description
        endpointQueryPaths
        harvested
        name
        category
        maxAgeSeconds
        valueType
        created
        updated
      }
    }
    pageInfo {
      endCursor
      startCursor
      hasNextPage
      hasPreviousPage
    }
  }
}
```

©2026 Tanium Inc. All Rights Reserved

Include the filter-related variables in the **Variables** panel:

```
{
  "first": 2,
  "hidden": true,
}
```

©2026 Tanium Inc. All Rights Reserved

Reference: Integration with Insomnia API Client

You can upload a Tanium Gateway example requests json file to the Insomnia API client, configure communications between Insomnia and the Tanium Server, and submit example requests and receive responses. Insomnia can also parse the example requests and generate code samples based on the requests.

Prerequisites

- Download the Insomnia API client and install it on an endpoint that can bidirectionally communicate with the Tanium Server. Ensure that your Insomnia API client version supports importing the Insomnia v4 json file format.
- Download one of the following JSON files that contains Tanium Gateway Insomnia example requests:
    - Tanium Gateway Insomnia example requests collection (json file) through January 2026
    - new Tanium Gateway Insomnia example requests (json file) added after February 2025 (as of August 2025)
    - Tanium Gateway Insomnia example requests collection (json file) through August 2025
    - Tanium Gateway Insomnia example requests collection (json file) through January 2025
    - new Tanium Gateway Insomnia example requests (json file) added after August 2024 (as of January 2025)
    - Tanium Gateway Insomnia example requests collection (json file) through November 2024

©2026 Tanium Inc. All Rights Reserved

    - new Tanium Gateway Insomnia example requests (json file) added after May 2024 (as of August 2024)
    - Tanium Gateway Insomnia example requests collection (json file) through August 2024

- In Tanium Console, create a user account for the integration. Add this user to the **Gateway User** role, and any other roles based on the Tanium solutions against which you want to submit requests. For more information about how to create a user, see Tanium Console User Guide: Managing users. For more information about supported Tanium solutions, see Solution dependencies on page 35.

  > **Note:** If you create an integration user in Tanium Console, you must also create a matching integration user in the identity provider that you use for Tanium. For more information, see Tanium Console User Guide: Create a user and Tanium Console User Guide: User authentication.

- In Tanium Console, create an API token for this user account and copy the token to your clipboard. For more information, see Tanium Console User Guide: Create an API token.

Configure Insomnia to communicate with Tanium

About this task

> **Important:** Back up your workspace before importing a json file. If you previously imported a Tanium Gateway example requests json file and made updates in the workspace, then import another json file into the project, it might overwrite your workspace changes.

1. In the Insomnia client, click **+** next to **Projects**.
2. Enter a project name, then click **Create**.
3. From the new project, select **Create** > **Request Collection**.
4. Enter a collection name, then click **Create**.
5. Select **collection_name** > **Import/Export** > **Import**.
6. Select the Tanium Gateway Insomnia example requests file, then click **Scan**.

©2026 Tanium Inc. All Rights Reserved

7. Click **Import**.
8. Select **project_name** > **collection_name**. Confirm that the import succeeded and the example syntax is loaded.
   If you previously imported a Tanium Gateway Insomnia example requests file into a project or collection, the import might overwrite your existing workspace.
9. From the new project, select **No Environment** > **Manage Environments**.
10. If table view is enabled, click **Table Edit** to enable json editing.
11. Copy the following json object, then paste it in the **Base Environment** pane:

```
{
  "url": "https://",
  "path": "/plugin/products/gateway/graphql",
  "api_token": "token-string"
}
```

12. Update the `url` value to the Tanium Server URL.
13. Update the `api_token` value to the API token string associated with your integration user. Click **Close**.

Verify integration

1. In the Insomnia client, select **project_name** > **collection_name**.
2. Expand **General examples** and then click **Time**. Verify that the request path is `POST _.url_.path`.
3. Click **Header** and verify that the `session` header is set to the `_.api_token` variable, and that the `Content-Type` header is set to `application/json`.
4. Click **Send**. The Gateway service responds with the current system time, such as the following:

```
{
  "data": {
    "now": "2021-11-08T19:22:03Z"
  }
```

©2026 Tanium Inc. All Rights Reserved

```
}
```

Generate code example from example request

1. In the Insomnia client, right-click an example, then select **Generate Code**.
2. Select a language from the dropdown list, then an associated library from the dropdown list. The code example updates as you select a language and library.

©2026 Tanium Inc. All Rights Reserved

Reference: Paginated query syntax

Query pagination allows you to retrieve a portion of a records collection (a page) per query response, preserving system performance but enabling full record retrieval through multiple responses. You can adjust the number of records returned per page, and retrieve different pages in the collection.

> **Note:** For the best results, due to the variability in size of Tanium deployments, retrieve 100 records per page for an initial test of your paginated query, and adjust the number of records returned per page based on performance and response time.

When you submit a paginated query request, the Gateway service assigns an opaque string that is valid for that query and collection (a cursor) to each record in the collection. Cursors generally persist for five minutes after the most recent use of any cursor in that collection.

For example, endpoints queries support pagination. If you submit an endpoints query request and receive a cursor in the response:

- You can reference this cursor in a successive query of that records collection to retrieve pages of records before or after a given cursor. You can repeat this process to retrieve all pages in the collection.
- You cannot reference this endpoints-associated cursor in a sensors query. The cursor is valid only for the endpoints collection.
- If you wait more than five minutes, then submit another endpoints query request referencing a cursor from the initial query request, the response returns an error because the cursors associated with your initial query are no longer valid.

Paginated queries generally support the following syntax:

- `<field>.edges` - retrieve information about a record in the collection
- `<field>.pageInfo` and `<field>.totalRecords` - retrieve information about pages and cursors in the collection, and collection total size

©2026 Tanium Inc. All Rights Reserved

- `<field> (arguments)` syntax to configure the number of records returned, and a cursor in the collection to determine which records to retrieve

Note: A field might not support all of these fields and arguments. For more information on specific fields, see the **Documentation Explorer** pane of the query explorer.

Edges

Fields that support query pagination define an `edge` as a record within the collection. For each edge, you can retrieve the following fields:

- `<field>.edges.node` - specific information about the object in that record
- `<field>.edges.cursor` - the record cursor

Edge query syntax

Use the following syntax to retrieve records for a field that supports query pagination:

```
query {
  <field> {
    edges {
      node {
        <record-field-1>
        <record-field-2>
        <record-field-n>
      }
      cursor
    }
  }
}
```

For example, the following query retrieves endpoint edges; for each edge, the response returns the endpoint name and IP address, and the cursor associated with that edge. The request does not specify pagination arguments, so the response by default returns the first 20 records from the collection.

```
query endpointsQuery {
  endpoints {
    edges {
      node {
        name
        ipAddress
      }
      cursor
```

©2026 Tanium Inc. All Rights Reserved

```
    }
  }
}
```

Page information

Fields that support query pagination allow you to retrieve information about a page of records in the collection using the following fields:

- `<field>.pageInfo.endCursor` - the last cursor in the current page
- `<field>.pageInfo.hasNextPage` - whether there are pages after the current page in the collection

Fields that support query pagination might also support the following fields:

- `<field>.pageInfo.hasPreviousPage` - whether there are pages before the current page in the collection
- `<field>.pageInfo.startCursor` - the first cursor in the current page
- `<field>.totalRecords` - the total number of records in the collection

Page information query syntax

Use the following syntax to retrieve page information for a field that supports query pagination:

```
query {
  <field> {
    edges {
      node {
        <record-field-1>
        <record-field-2>
        <record-field-n>
      }
      cursor
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
    totalRecords
  }
}
```

For example, the following query retrieves endpoints and also returns page information, including

©2026 Tanium Inc. All Rights Reserved

the first cursor, last cursor, and whether there are pages before and after the current page in the collection. The query also returns the total number of records in the collection. The request does not specify pagination arguments, so the response by default returns the first 20 records from the collection.

```
query endpointsQueryPageInfo {
  endpoints {
    edges {
      node {
        name
        ipAddress
      }
      cursor
    }
    pageInfo {
          startCursor
          endCursor
          hasPreviousPage
          hasNextPage
    }
    totalRecords
  }
}
```

Paginated collection traversal arguments

Fields that support query pagination have arguments that control the number of records returned per page and which page is returned. If a query does not specify these arguments, the default settings generally return 20 records forward from the first page at the start of the collection.

Use the following arguments to traverse forward in a records collection:

- `<field> (after)` - retrieve records after the specified cursor
- `<field> (first)` - retrieve the first number of records after a cursor, or at the beginning of the collection if no cursor is specified

Use the following arguments to traverse backward in a records collection:

- `<field> (before)` - retrieve records before the specified cursor
- `<field> (last)` - retrieve the last number of records before a cursor

©2026 Tanium Inc. All Rights Reserved

Note: In a single query request, you can retrieve results either forward from a cursor or backwards from a cursor. You cannot specify both `before` and `after` in a single query request.

Forward traversal argument syntax

Use the following syntax to retrieve the first number of results from the start of a paginated records collection:

```
<field>(first: <Int>)
```

Use the following syntax to retrieve the first number of results without specifying a cursor (`null`), such as when you first retrieve records and pages from a paginated records collection:

```
<field>(after: null, first: <Int>)
```

Use the following syntax to retrieve the first number of results after a specified cursor in a paginated records collection, such as when you retrieve `<field>.pageInfo.endCursor` in a prior query and want to traverse forward to retrieve the next page of records:

```
<field>(after: <Cursor>, first: <Int>)
```

For example, the following query retrieves the first 20 endpoint records from the beginning of the collection, and also returns cursor information for this first page, including `endpoints.pageInfo.endCursor` for the last cursor on the page:

```
query endpointsQueryNullCursor {
  endpoints (after: null, first: 20){
    edges {
      node {
        name
        ipAddress
      }
      cursor
    }
    pageInfo {
          startCursor
          endCursor
          hasPreviousPage
          hasNextPage
    }
    totalRecords
  }
}
```

To retrieve the next page of 20 records in this collection, pass the `endpoints.pageInfo.endCursor` value from the prior query as the `after` argument value:

```
query endpointsQueryNextPage {
```

©2026 Tanium Inc. All Rights Reserved

```
  endpoints (after: <endpoints.pageInfo.endCursor-value>, first: 20){
    edges {
      node {
        name
        ipAddress
      }
      cursor
    }
    pageInfo {
          startCursor
          endCursor
          hasPreviousPage
          hasNextPage
    }
    totalRecords
  }
}
```

Continue submitting requests with the `endpoints.pageInfo.endCursor` value from the prior query response to retrieve successive pages in the collection. If a response includes `endpoints.pageInfo.hasNextPage` equals `false`, you retrieved the last page in the collection.

Backward traversal argument syntax

Use the following syntax to retrieve the prior number of results before a specified cursor in a paginated records collection, such as when you retrieve `<field>.pageInfo.startCursor` in the prior query and want to traverse backward to retrieve the prior page of records:

```
<field>(last: <Int>, before: <Cursor>)
```

For example, if you retrieve the `endpoints.pageInfo.startCursor` value from a query and `endpoints.pageInfo.hasPreviousPage` equals `true`, a page of results precedes the page in the response. To retrieve the prior page of 20 records in this collection, pass the `endpoints.pageInfo.startCursor` value from the prior query as the `before` argument value:

```
query endpointsQueryPreviousPage {
  endpoints (before: <endpoints.pageInfo.startCursor-value>, last: 20){
    edges {
      node {
        name
        ipAddress
      }
      cursor
    }
    pageInfo {
          startCursor
          endCursor
          hasPreviousPage
          hasNextPage
    }
    totalRecords
```

©2026 Tanium Inc. All Rights Reserved

```
  }
}
```

Continue submitting requests with the `endpoints.pageInfo.startCursor` value from the prior query response to retrieve prior pages in the collection. If a response includes `endpoints.pageInfo.hasPreviousPage` equals `false`, you retrieved the first page in the collection.

Paginated endpoints collection refresh argument

If you want to obtain updated information from an endpoints collection, such as if you expect endpoints to respond several minutes after your initial query request, you can refresh the collection. If your query request refreshes the collection, the Gateway service updates collection results with current information, invalidates the prior set of cursors, and assigns new cursors to the records. The `endpoints.collectionInfo` field returns additional information about endpoints in the collection, including information about endpoints that responded to the query, and an `endpoints.collectionInfo.startCursor` cursor value. This value always points to the first cursor in the collection; you can reference this cursor value to refresh a collection.

Use the `endpoints (refresh)` argument to refresh an endpoints collection and the associated cursors.

Collection and cursor refresh argument syntax

Use the following syntax to refresh a paginated endpoint records collection and the associated cursors, such as when you expect the set of endpoints to update after your initial query, and you retrieved `endpoints.collectionInfo.startCursor` in a prior query:

```
endpoints (refresh: <Cursor>)
```

The following example returns the `startCursor` value and additional endpoint information:

```
query endpointsQueryCollectionInfo {
  endpoints {
    edges {
      node {
        name
        id
      }
      cursor
    }
    pageInfo {
      startCursor
      endCursor
```

©2026 Tanium Inc. All Rights Reserved

```
      hasPreviousPage
      hasNextPage
    }
    totalRecords
    collectionInfo {
      active
      contributedTotal
      expectedTotal
      respondedPercentage
      respondedTotal
      startCursor
      success
    }
  }
}
```

After you obtain the `endpoints.collectionInfo.startCursor` value, and more endpoints respond to the query, you can refresh the collection of endpoints by passing `endpoints.collectionInfo.startCursor` as the `refresh` argument value:

```
query endpointsQueryRefresh {
  endpoints (refresh: <endpoints.collectionInfo.startCursor-value>){
    edges {
      node {
        name
        id
      }
      cursor
    }
    pageInfo {
      startCursor
      endCursor
      hasPreviousPage
      hasNextPage
    }
    totalRecords
    collectionInfo {
      active
      contributedTotal
      expectedTotal
      respondedPercentage
      respondedTotal
      startCursor
      success
    }
  }
}
```

Because the collection refresh also refreshes the collection's cursors, you must pass updated cursors the next time you want to query this updated collection.

©2026 Tanium Inc. All Rights Reserved

Reference: Stability index updates

Gateway schema fields are indexed to a level of stability, which correlates with the level of support and maintenance Tanium provides for that field.

Stability levels

Stability level	Description	Support provided
`0 - Deprecated`	Deprecated fields stop receiving support. Fields might be deprecated in favor of other fields for improved performance, functionality support, or other reasons.	Deprecated fields receive support and maintenance for a minimum of 12 months from the initial deprecation announcement. Starting with the next release after that 12 month period, deprecated fields might be removed from the Gateway schema.
`1.0 - Experimental (Early Development)`	Experimental fields in early development are not stable, and might change significantly as functionality is developed and tested.	Experimental fields are subject to update or removal, without guarantee of backwards compatibility. After an evaluation period of indeterminate length, an experimental field's stability index might be updated and the field kept in the Gateway schema, or the field might be removed from the Gateway schema.
`1.1 - Experimental (Active Development)`	Experimental fields in active development are not stable, and the implemented functionality is closer to meeting requirements than experimental fields in early development.	
`1.2 - Experimental (Release Candidate)`	Experimental fields as release candidate are not stable, and the functionality is ready for evaluation. Functionality or other changes might occur in response to the evaluation.	
`2 - Legacy`	Legacy fields are not actively	Legacy fields receive support and maintenance for

©2026 Tanium Inc. All Rights Reserved

Stability level	Description	Support provided
	developed. Alternative fields in the Gateway schema provide improved functionality.	existing functionality, but do not receive ongoing feature functionality updates.
`3 - Stable`	Stable fields have active support.	Stable fields receive ongoing feature functionality updates as needed, and support and maintenance.

Fields are not required to advance sequentially through each stability, nor are they required to be indexed to each stability level.

Note: For the best results, if a field's stability changes to `0 - Deprecated` or `2 - Legacy`, update your requests to replace the deprecated or legacy field with the recommended replacement.

July 2026 stability index changes

The following field changed to stability `1.1 - Experimental (Active Development)`.

Field	Notes
`object.TaniumPlatformUser.availablePersonas`	list of personas available to the current user

June 2026 stability index changes

The following fields changed to stability `1.1 - Experimental (Active Development)`.

Field	Notes
`query.connectSQLServerStoredProcedures(input:)`	query argument; retrieves the stored procedures defined in one or more SQL Server schemas within a database, including parameter

©2026 Tanium Inc. All Rights Reserved

Field	Notes
	names, data types, and `EXECUTE` permission for the configured user
`ConnectSQLServerSchemaProcedures.schemaName`	schema name; matches one of the schema names in the request input
`ConnectSQLServerSchemaProcedures.procedures`	stored procedures defined in the schema; empty when the schema has no procedures or does not exist
`ConnectSQLServerStoredProcedure.name`	procedure name
`ConnectSQLServerStoredProcedure.hasExecute`	whether the configured user has `EXECUTE` permission on the procedure
`ConnectSQLServerStoredProcedure.parameters`	declared parameters, in declaration order
`ConnectSQLServerStoredProcedureParameter.name`	parameter name
`ConnectSQLServerStoredProcedureParameter.dataType`	SQL Server type name; matches the value of `sys.types.name`

May 2026 stability index changes

The following field changed to stability `1.1 - Experimental (Active Development)`.

Field	Notes
`PlaybookSchedule.deleted`	whether a playbook schedule has been deleted

The following fields changed to stability `1.2 - Experimental (Release Candidate)`.

Field	Notes
`AutomateErrorCode.SERVICE_UNAVAILABLE`	error code: service unavailable
`AutomateErrorCode.SESSION_EXPIRED`	error code: session expired

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`AutomateErrorCode.SESSION_UNAUTHORIZED`	error code: session unauthorized
`AutomateErrorCode.UNKNOWN`	error code: unknown
`AutomateTaskResultDeploySoftwareDeployment.expectedCompletionTime`	expected completion time for a deploy software deployment task result
`AutomateTaskResultPatchInstallDeployment.expectedCompletionTime`	expected completion time for a patch install deployment task result
`AutomateTaskResultVerifyCondition.expectedCompletionTime`	expected completion time for a verify condition task result
`AutomateTaskResultVerifyCondition.progress`	progress for a verify condition task result
`AutomateTaskResultWithEndpointExecutionChildStepDetails.skippedEndpoints`	endpoints skipped during an endpoint execution child step
`AutomateTaskResultWithExpectedCompletionTime.expectedCompletionTime`	expected completion time for an Automate task result
`AutomateTaskResultWithProgress.progress`	progress for an Automate task result with progress
`DeploymentPlan`	deployment plan type and all its fields
`DeploymentPlanFieldFilter`	filter input type for deployment plans and all its fields
`DeploymentPlanFieldSort`	sort input type for deployment plans and all its fields
`mutation.deploymentPlanPerformOperation(input:)`	mutation argument
`mutation.deploymentPlanUpsert(input:)`	mutation argument
`mutation.ringDeploymentChangeWait(input:)`	mutation argument
`mutation.ringDeploymentPerformOperation(input:)`	mutation argument

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`mutation.ringDeploymentSendSignal(input:)`	mutation argument
`mutation.ringDeploymentStart(input:)`	mutation argument
`nextRunTimes(count:)`	field argument for retrieving the next scheduled run times
`PlaybookNotificationErrorCode.EMAIL_NOTIFICATION_TIMEOUT`	playbook email notification error code: email notification timeout
`PlaybookNotificationErrorCode.EMAIL_SERVER_DOES_NOT_EXIST`	playbook email notification error code: email server does not exist
`PlaybookNotificationErrorCode.EMAIL_SERVER_UNAVAILABLE`	playbook email notification error code: email server unavailable
`PlaybookNotificationEvent.errorCode`	error code associated with a playbook notification event
`PlaybookRun.startDetails`	details about how a playbook run was started
`PlaybookRun.startType`	start type of a playbook run (schedule-initiated or user-initiated)
`PlaybookRunFieldFilter.startReason`	filter playbook runs by start reason
`PlaybookRunStartDetailsSchedule.schedule`	schedule associated with a schedule-initiated playbook run
`PlaybookRunStartDetailsSchedule.scheduleStartTime`	start time of the schedule that triggered the playbook run
`PlaybookRunStartReasonFilter.scheduleSpecTypes`	filter playbook runs by schedule spec type
`PlaybookRunStartReasonFilter.startTypes`	filter playbook runs by start type
`PlaybookRunStartType.SCHEDULE`	playbook run start type: initiated by a schedule
`PlaybookRunStartType.USER`	playbook run start type: initiated by a user

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`PlaybookRunStepError`	playbook run step error interface fields
`PlaybookRunStepErrorInvalidSession`	playbook run step error type for invalid sessions and all its fields
`PlaybookRunStepErrorServiceUnavailable`	playbook run step error type for service unavailability and all its fields
`progressionConditionResults(includedStates:)`	field argument
`progressionConditionResults(includeStates:)`	field argument
`ProgressionCondition`	progression condition interface and subtypes
`ProgressionConditionInput`	progression condition input type and subtypes
`ProgressionCriteriaSet`	progression criteria set type and all its fields
`query.deploymentPlan(id:)`	query and its argument
`query.deploymentPlans`	query and all pagination, filter, and sort arguments
`query.ringDeployment(id:)`	query and its argument
`query.ringDeployments`	query and all pagination, filter, and sort arguments
`RingDeployment.notificationSettings`	notification settings for a ring deployment
`RingDeploymentGuideNotificationSettings.events`	guide notification event types for a ring deployment
`RingDeploymentGuideNotificationSettings.userIds`	user IDs to notify for ring deployment guide notifications
`RingDeploymentGuideNotificationSettingsInput.events`	guide notification event types input for a ring deployment
`RingDeploymentGuideNotificationSettingsInput.userIds`	user IDs input for ring deployment guide notifications

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`RingDeploymentNotificationEventType.DEPLOYMENT_FINISHED`	ring deployment notification event type: deployment finished
`RingDeploymentNotificationEventType.FINAL_RING_STARTED`	ring deployment notification event type: final ring started
`RingDeploymentNotificationEventType.MANUAL_OVERRIDE`	ring deployment notification event type: manual override
`RingDeploymentNotificationEventType.RING_STARTED`	ring deployment notification event type: ring started
`RingDeploymentNotificationEventType.RING_STOPPED`	ring deployment notification event type: ring stopped
`RingDeploymentNotificationSettings.guide`	guide notification settings for a ring deployment
`RingDeploymentNotificationSettingsInput.guide`	guide notification settings input for a ring deployment
`RingDeploymentScheme`	ring deployment scheme type and all its fields
`RingDeploymentStartInput.notificationSettings`	notification settings input when starting a ring deployment



Note: `PlaybookNotificationErrorCode.EMAIL_AUTH_FAILED` remains at stability `1.1 - Experimental (Active Development)`.

The following fields changed to stability `3 - Stable`.

Fields updated in May 2026 to stability level 3

Field	Notes
`AutomateTaskResultWithConfirmation.confirmationResult`	confirmation result for an Automate task result with confirmation

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`AutomateTaskResultConfirmation.confirmationResult`	confirmation result for an Automate task result confirmation
`AutomateTaskResultInstruction.confirmationResult`	confirmation result for an Automate task result instruction

April 2026 stability index changes

The following fields are changed to stability `0 - Deprecated`; support for these fields ends May 2027.

April 2026 deprecated fields

Deprecated field	Replacement field
`object.PlaybookStepOverride.stepId`	`object.PlaybookStepOverride.stepID`

March 2026 stability index changes

The following fields are added at stability `1.0 - Experimental (Early Development)`.

Fields added in March 2026 at stability level 1.0

Field	Notes
`query.playbookRuns.filter.startReason.startType`	filter playbook runs by start reason

The following fields changed to stability `3 - Stable`.

Fields updated in March 2026 to stability level 3

Field	Notes
`mutation.integrityMonitorMonitorCreate`	creates a monitor
`mutation.integrityMonitorMonitorMarkForDeletion`	deletes a monitor by ID
`mutation.integrityMonitorMonitorUpdate`	updates a monitior by ID

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`mutation.integrityMonitorStartMonitorDeployment`	deploys all monitors
`mutation.integrityMonitorStartWatchlistDeployment`	deploys all watchlists
`mutation.integrityMonitorWatchlistCreate`	creates a Windows-style watchlist
`mutation.integrityMonitorWatchlistMarkForDeletion`	deletes a watchlist by ID
`mutation.integrityMonitorWatchlistPathDelete`	deletes a watchlist path by ID
`mutation.integrityMonitorWatchlistUnixPathCreate`	creates a Unix-style watchlist
`mutation.integrityMonitorWatchlistUnixPathUpdate`	updates a Unix-style watchlist path
`mutation.integrityMonitorWatchlistUpdate`	updates a watchlist by ID
`mutation.integrityMonitorWatchlistWindowsFilePathCreate`	creates a Windows-style watchlist file path
`mutation.integrityMonitorWatchlistWindowsFilePathUpdate`	creates a Windows-style watchlist file path
`mutation.integrityMonitorWatchlist WindowsRegistryPathCreate`	creates a Windows-style watchlist registry path
`mutation.integrityMonitorWatchlist WindowsRegistryPathUpdate`	creates a Windows-style watchlist registry path
`query.integrityMonitorMonitor`	retrieves a monitor by ID
`query.integrityMonitorMonitors`	retrieves multiple monitors
`query.integrityMonitorStatus`	retrieves the Integrity Monitor service status
`query.integrityMonitorWatchlist`	retrieves a watchlist by ID
`query.integrityMonitorWatchlists`	retrieves multiple watchlists

The following fields are changed to stability `0 - Deprecated`; support for these fields ends April 2027.

©2026 Tanium Inc. All Rights Reserved

March 2026 deprecated fields

Deprecated field	Replacement field
`input.PlaybookStartInput.taskArgumentOverrides`	`input.PlaybookStartInput.stepOverrides.taskArgument`
`input.PlaybookScheduleUpsertInput.taskArgumentOverrides`	`input.PlaybookScheduleUpsertInput.stepOverrides.taskArgument`
`object.PlaybookSchedule.taskArgumentOverrides`	`object.PlaybookSchedule.stepOverrides.taskArgument`
`object.PlaybookStepTaskArgumentOverride`	`object.PlaybookStepOverride`
`query.playbookRuns.edges.node.steps.error`	`query.playbookRuns.edges.node.steps.errorV2`
`query.playbookRun.steps.error`	`query.playbookRun.steps.errorV2`
`query.playbookRuns.filter.isScheduled`	`query.playbookRuns.filter.scheduleID`

December 2025 stability index changes

The following fields are added at stability `1.2 - Experimental (Release Candidate)`.

Fields added in December 2025 at stability level 1.2

Field	Notes
`query.endpoints.edges.node.compliance.cveFindings.cwes.id`	retrieve the ID of the Common Weakness Enumeration (CWE)
`query.endpoints.edges.node.compliance.cveFindings.cwes.name`	retrieve the name of the CWE
`query.endpoints.edges.node.compliance.cveFindings.mitreAttacks.id`	retrieve the ID of the MITRE ATT&CK
`query.endpoints.edges.node.compliance.cveFindings.mitreAttacks.name`	retrieve the name of the MITRE ATT&CK
`query.endpoints.edges.node.compliance.cveFindings.cvssTemporalScoreV3`	retrieve the CVSS v3 temporal score of the CVE
`query.endpoints.edges.node.compliance.cveFindings.alias`	retrieve the alias of the CVE
`query.endpoints.edges.node.compliance.cveFindings.epssScore`	retrieve the Exploit Prediction Scoring System (EPSS) score of the CVE
`query.endpoints.edges.node.compliance.cveFindings.epssPercentile`	retrieve the EPSS percentile of the CVE

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`query.endpoints.edges.node.compliance.cveFindings.epssPercentileRange`	retrieve the EPSS percentile range of the CVE
`query.endpoints.edges.node.compliance.cveFindings.maxMaturity`	retrieve the max maturity of exploits of the CVE
`query.endpoints.edges.node.compliance.cveFindings.cvssScoreV4`	retrieve the CVSS v4 score of the CVE
`query.endpoints.edges.node.compliance.cveFindings.cvssSeverityV4`	retrieve the CVSS v4 severity of the CVE
`query.endpoints.edges.node.compliance.cveFindings.cvssThreatScoreV4`	retrieve the CVSS v4 Threat Score of the CVE
`query.endpoints.edges.node.compliance.cveFindings.cvssVectorV4`	retrieve the CVSS v4 vector of the CVE
`query.endpoints.edges.node.compliance.cveFindings.createdDate`	retrieve the creation date of the CVE
`query.endpoints.edges.node.compliance.cveFindings.modifiedDate`	retrieve the last modification date of the CVE
`query.endpoints.edges.node.compliance.cveFindings.mitreLink`	retrieve the MITRE link of the CVE
`query.endpoints.edges.node.compliance.cveFindings.nistLink`	retrieve the NIST link of the CVE
`query.endpoints.edges.node.compliance.cveFindings.secpodLink`	retrieve the SecPod link of the CVE
`query.endpoints.edges.node.compliance.cveFindings.solutionLinks`	retrieve a list of solution links of the CVE
`query.endpoints.edges.node.compliance.cveFindings.remediation`	retrieve the remediation text of the CVE

August 2025 stability index changes

The following fields are changed to stability `0 - Deprecated`; support for these fields ends March 2027.

August 2025 deprecated fields

Deprecated field	Replacement field
`query.playbookRun.steps.progress.details.actionID`	`query.playbookRun.steps.taskResult.actionID`
`query.playbookRuns.edges.node.steps.progress.details.actionID`	`query.playbookRuns.edges.node.steps.taskResult.actionID`

©2026 Tanium Inc. All Rights Reserved

July 2025 stability index changes

The behavior of the following fields is deprecated, but the fields remain with new behavior.

Fields with deprecated behavior in July 2025

Deprecated field	Replacement field
`query.endpoints (source.tds.allNamespaces)` `query.endpoints (source.tds.namespaces)`	These fields will remain, but the behavior of these fields is deprecated as of July 2025. Currently, you can include the `allNamespaces` and `namespaces` fields in the `source` input to `query.endpoints`. After support ends in December 2026, these fields become mutually exclusive; if you include both fields, a validation error occurs. To continue using these fields, only use one of the fields in each query.

The following fields are added at stability `1.2 - Experimental (Release Candidate)`.

Fields added in July 2025 at stability level 1.2

Field	Notes
`query.endpoints.edges.node.compliance.cveFindings.detectedCPEs`	retrieve CPEs that were detected on an endpoint during a Comply scan
`query.endpoints.edges.node.compliance.cveFindings.detectedProducts`	retrieve products that were detected on an endpoint during a Comply scan

June 2025 stability index changes

The following fields are changed to stability `0 - Deprecated`; support for these fields ends December 2026.

Fields deprecated in June 2025

Deprecated field	Replacement field
`query.playbookRun.steps.confirmationResult`	`query.playbookRun.steps.taskResult`
`query.playbookRun.steps.progress`	`query.playbookRun.steps.taskResult`
`query.playbookRuns.edges.node.steps.confirmationResult`	`query.playbookRuns.edges.node.steps.taskResult`

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field
`query.playbookRuns.edges.node.steps.progress`	`query.playbookRuns.edges.node.steps.taskResult`

April 2025 stability index changes

The following fields are changed to stability `0 - Deprecated`; support for these fields ends November 2026.

Fields deprecated in April 2025

Deprecated field	Replacement field
`mutation.playbookUpsert.playbook.canRun`	This field does not map directly to a replacement field. Instead, use `mutation.playbookStart` with (`input.dryRun`) set to `true` . The `dryRun` field consumes fewer resources to determine whether the current user can create a playbook.
`mutation.playbookStart.playbookRun.playbook.canRun`	
`query.playbookRuns.edges.node.canRun`	
`query.playbooks.edges.node.canRun`	
`query.playbookSchedules.edges.node.playbook.canRun`	
`mutation.playbookStart.playbookRun.steps.playbookStep.currentUserMissingPermissions`	
`mutation.playbookUpsert.playbook.runs.edges.node.steps.playbookStep.currentUserMissingPermissions`	
`query.playbookRuns.edges.node.steps.currentUserMissingPermissions`	
`query.playbooks.edges.node.steps.currentUserMissingPermissions`	
`query.playbookSchedules.edges.node.playbook.steps.currentUserMissingPermissions`	

The following fields are added at stability `1.1 - Experimental (Active Development)`.

Fields added in April 2025 at stability level 1.1

Field	Notes
`query.playbookRuns.initatorUserIDs`	retrieve the user IDs of all users that started a playbook run for each playbook run in the collection
`query.playbookRuns.initiatorUsers`	retrieve the users that started a playbook run for each playbook run in the collection

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`query.playbookRuns.playbooks`	retrieve the playbook ID and name for each playbook associated with a playbook run in the collection

The following fields changed to stability `1.2 - Experimental (Release Candidate)`.

Fields updated in April 2025 to stability level 1.2

Field	Notes
`mutation.playbookContentSetUpdate`	update only the content set associated with a playbook
`mutation.playbookPerformOperation`	perform a playbook operation, such as delete a playbook
`mutation.playbookPreview`	preview the results of creating a playbook without creating the playbook
`mutation.playbookRunPerformOperation`	perform a playbook run operation, such as pause a playbook run
`mutation.playbookSchedulePerformOperation`	perform a playbook schedule operation, such as delete a playbook schedule

The following fields changed to stability `3 - Stable`.

Fields updated in April 2025 to stability level 3

Field	Notes
`mutation.playbookScheduleUpsert`	create a new playbook schedule, or update an existing playbook schedule that matches the ID
`mutation.playbookStart`	create a playbook run from a playbook
`mutation.playbookUpsert`	create a playbook, or update an existing playbook that matches the ID
`query.playbook`	retrieve the playbook that matches the ID
`query.playbooks`	retrieve multiple playbooks
`query.playbookRun`	retrieve the playbook run that matches the ID

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`query.playbookRuns`	retrieve multiple playbook runs
`query.playbookSchedule`	retrieve the playbook schedule that matches the ID
`query.playbookSchedules`	retrieve multiple playbook schedules

January 2025 stability index changes

The following field is added at stability `1.1 - Experimental (Active Development)`.

Fields added in January 2025 at stability level 1.1

Field	Notes
`query.currentUserContext.persona.name`	retrieve the active persona name for the current user

November 2024 stability index changes

The following field is changed to stability `0 - Deprecated`; support for this field ends June 2026.

Fields deprecated in November 2024

Deprecated field	Replacement field
`mutation.assetsImport`	`mutation.assetUpsertEndpoints`

The following field is added at stability `1.1 - Experimental (Active Development)`.

Fields added in November 2024 at stability level 1.1

Field	Notes
`query.managementRightsGroups`	retrieve the management rights groups that a user and active persona can access

©2026 Tanium Inc. All Rights Reserved

September 2024 stability index changes

The following field is added at stability `1.2 - Experimental (Release Candidate)`.

Fields added in September 2024 at stability level 1.2

Field	Notes
`query.computerGroups.edges.node.modifiedTime`	retrieve the time that a computer group was last updated
`query.computerGroup.modifiedTime`	retrieve the time that the computer group was last updated

August 2024 stability index changes

The following fields are changed to stability `0 - Deprecated`; support for these fields ends March 2026.

Fields deprecated in August 2024

Deprecated field	Replacement field	Notes
`mutation.deleteConfigurationItemElement`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.deleteRelationship`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.importConfigurationItemEntities`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.mergeConfigurationItemElements`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.syncAssets`	Gateway does not provide a	This field is no longer supported. This field

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
	replacement field.	was developed for an earlier Salesforce integration.
`mutation.updateConfigurationItemProperties`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`mutation.upsertRelationship`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.configurationItemEntities`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.configurationItemProperties`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.configurationItemRelationships`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.endpoints.edges.node.configurationItem`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.
`query.endpoints.edges.node.risk`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.assetCriticality`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .

©2026 Tanium Inc. All Rights Reserved

Deprecated field	Replacement field	Notes
`query.endpoints.edges.node.risk.criticalityScore`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.riskLevel`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.totalScores`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.endpoints.edges.node.risk.vectors`	This field retrieves only Risk Score 1.0 information. Gateway does not provide a replacement field.	Use Risk Score 2.0 instead, which aligns to the newest Benchmark product data. For more information, see Tanium Benchmark User Guide: Evaluating your risk score .
`query.relationshipTypes`	Gateway does not provide a replacement field.	This field is no longer supported. This field was developed for an earlier Salesforce integration.

June 2024 stability index changes

The following fields are added at stability `1.0 - Experimental (Early Development)`.

Fields added in June 2024 at stability level 1.0

Field	Notes
`query.endpointCounts (input.refresh)`	refresh the endpoint counts collection identified by the given cursor, to retrieve results that might be slow to populate
`query.endpointCounts.collectionInfo`	retrieve information about the endpoint count collection, including the

©2026 Tanium Inc. All Rights Reserved

Field	Notes
	counts of endpoints that contribute to the collection, and cursor information

May 2024 stability index changes

The following field is added at stability `1.1 - Experimental (Active Development)`.

Fields added in May 2024 at stability level 1.1

Field	Notes
`query.endpoints.collectionInfo.question.consoleURL`	retrieve the URL of the **Question Results** page for this question in Tanium Console

The following field is updated to stability `3 - Stable`.

Fields updated to stability level 3 in May 2024

Field	Notes
`query.endpoints (sort)`	sort the returned endpoints based on a single field or sensor column

March 2024 stability index changes

The following fields are added at stability `1.0 - Experimental (Early Development)`.

Fields added in March 2024 at stability level 1.0

Field	Notes
`query.endpointCounts`	retrieve endpoint counts matching the input sensor column values
`query.endpointCounts.edges.node.columns`	retrieve the sensor columns and values associated with the matching endpoint count

©2026 Tanium Inc. All Rights Reserved

Field	Notes
`query.endpointCounts.edges.node.count`	retrieve the endpoint count matching the input sensor column values
`query.endpoints.collectionInfo.question.id`	retrieve the ID of the question used to query the Tanium Server (`ts`) data source for endpoint information

The following field is added at stability `1.1 - Experimental (Active Development)`.

Fields added in March 2024 at stability level 1.1

Field	Notes
`query.endpoints.edges.node.sensorReading`	retrieve endpoint data from a single sensor, and convert the sensor result data type

The following field is added at stability `1.2 - Experimental (Release Candidate)`.

Fields added in March 2024 at stability level 1.2

Field	Notes
`query.endpoints (sort)`	sort the returned endpoints based on a single field or sensor column

The following fields are added at stability `3 - Stable`.

Fields added in March 2024 at stability level 3

Field	Notes
`query.packageSpecs.edges.node.lastUpdatedBy`	retrieve the last user to update a package spec object
`query.sensors.edges.node.lastUpdatedBy`	retrieve the last user to update a sensor

February 2024 stability index changes

The following field is added at stability `3 - Stable`.

©2026 Tanium Inc. All Rights Reserved

Fields added in February 2024 at stability level 3

Field	Notes
`query.endpoints (filter.maxAge)`	force the referenced filter sensor to reevaluate data if the cached sensor data is older than this value

January 2024 stability index changes

The following field is added at stability `1.1 - Experimental (Active Development)`.

Fields added in January 2024 at stability level 1.1

Field	Notes
`query.currentUserContext`	retrieve the current user and active persona

The following field is added at stability `1.2 - Experimental (Release Candidate)`.

Fields added in January 2024 at stability level 1.2

Field	Notes
`query.actions`	retrieve multiple actions

November 2023 stability index changes

The following fields are added at stability `1.2 - Experimental (Release Candidate)`.

Fields added in November 2023 at stability level 1.2

Field	Notes
`query.endpoints.edges.node.compliance. complianceFindings.excepted`	retrieve whether this compliance finding is excepted
`query.endpoints.edges.node.compliance.cveFindings. affectedProducts`	retrieve the affected products associated with a CVE
`query.endpoints.edges.node.compliance.cveFindings. cisaDateAdded`	retrieve the date the vulnerability was added to the

©2026 Tanium Inc. All Rights Reserved

Field	Notes
	CISA KEV catalog
`query.endpoints.edges.node.compliance.cveFindings.cisaDueDate`	retrieve the CISA KEV entry due date for taking action
`query.endpoints.edges.node.compliance.cveFindings.cisaNotes`	retrieve the CISA KEV entry notes
`query.endpoints.edges.node.compliance.cveFindings.cisaProduct`	retrieve the CISA KEV entry product information
`query.endpoints.edges.node.compliance.cveFindings.cisaRequiredAction`	retrieve the CISA KEV entry listed action to address the vulnerability
`query.endpoints.edges.node.compliance.cveFindings.cisaShortDescription`	retrieve the CISA KEV entry short description
`query.endpoints.edges.node.compliance.cveFindings.cisaVendor`	retrieve the CISA KEV entry vendor or project name
`query.endpoints.edges.node.compliance.cveFindings.cisaVulnerabilityName`	retrieve the CISA KEV entry vulnerability name
`query.endpoints.edges.node.compliance.cveFindings.cpes`	retrieve Common Platform Enumerations (CPE) associated with the CVE
`query.endpoints.edges.node.compliance.cveFindings.excepted`	retrieve whether this CVE finding is excepted
`query.endpoints.edges.node.compliance.cveFindings.isCisaKev`	retrieve whether this CVE finding is included in the CISA KEV catalog
`query.endpoints.edges.node.compliance.cveFindings.scanType`	retrieve the method used to discover the vulnerability finding

The following field is changed to stability `2 - Legacy`.

Fields updated to stability level 2 in November 2023

Legacy field	Replacement field	Notes
`query.packages`	`query.packageSpecs`	The `query.packageSpecs` request offers improved pagination and parameter support for retrieving packages compared to

©2026 Tanium Inc. All Rights Reserved

Legacy field	Replacement field	Notes
		`query.packages`. The following `query.packages.items` arguments map one-to-one to `query.packageSpecs` arguments: • `query.packages.items.command` to `query.packageSpecs.command` • `query.packages.items.commandTimeout` to `query.packageSpecs.commandTimeoutSeconds` • `query.packages.items.contentSet.name` to `query.packageSpecs.contentSetName` • `query.packages.items.expireSeconds` to `query.packageSpecs.expireSeconds` • `query.packages.items.id` to `query.packageSpecs.id` • `query.packages.items.name` to `query.packageSpecs.name` • `query.packages.items.parameters.key` to `query.packageSpecs.params.key` Unlisted `query.packages` arguments do not map directly to a replacement argument. For more information, see a `query.packageSpecs` Get package spec information on page 147.

©2026 Tanium Inc. All Rights Reserved

July 2023 stability index changes

DRAFT COMMENT Include only for stability index values where this is true in this and future entries. - No fields changed to stability [`0 - Deprecated`, `1 - Experimental`, `2 - Legacy`, `3 - Stable`].

This release introduces the stability index.

- Existing fields in active support are changed to stability `3 - Stable`.
- Unless otherwise stated below, fields deprecated in prior releases are changed to stability `0 - Deprecated`.

The following field is changed to stability `0 - Deprecated`; support for this field ends February 2025.

Fields deprecated in July 2023

Deprecated field	Replacement field	Notes
`query.endpointLastSeen`	`query.endpoints (filter: {path: id, op: EQ, value: "endpoint-id"})` and include `query.endpoints.edges.node.eidLastSeen` in the request	The `query.endpointLastSeen (eids)` argument does not map directly to a replacement argument. Instead, create a `query.endpoints` filter argument, using `{path: id, op: EQ, value: "endpoint-id"}` to filter the request to return selected endpoints based on ID, then also return `query.endpoints.edges.node.eidLastSeen` in the response to view an endpoint's last seen time. For more information, see an `endpoints` query with an endpoint ID filter returning `eidlastSeen` Reference: Filter syntax on page 562 .

The following fields are changed to stability `1.0 - Experimental (Early Development)`, `1.1 - Experimental (Active Development)`, or `1.2 - Experimental (Release Candidate)`.

©2026 Tanium Inc. All Rights Reserved

Fields updated to stability level 1.2 in July 2023

Experimental field	Notes
`query.directConnectEndpoint.performance._dev_query (1.2 - Experimental (Release Candidate))`	

The following fields are changed to stability `2 - Legacy`.

Fields updated to stability level 2 in July 2023

Legacy field	Replacement field	Notes
`mutation.closeDirectConnection`	`mutation.directConnectClose`	The `mutation.closeDirectConnection (input)` argument maps one-to-one to the `mutation.directConnectClose (input)` argument. For more information, see a `mutation.directConnectClose` Reference: Gateway examples on page 63 .
`mutation.killProcess`	`mutation.directConnectProcessTerminate`	Certain arguments for `mutation.killProcess (input)` map one-to-one to `mutation.directConnectProcessTerminate (input)` arguments: • `name` • `pid` • `signal` The `mutation.killProcess (input: {connectionID: ID})` optional argument maps to the `mutation.directConnectProcessTerminate (input: {connectionID: ID!}` required argument. The `mutation.killProcess (input: {endpoint: {endpointID: ID}})`

©2026 Tanium Inc. All Rights Reserved

Legacy field	Replacement field	Notes
		argument does not map directly to a replacement argument. Instead of passing an endpoint ID, pass an open connection ID in the `mutation.directConnectProcessTerminate (input: {connectionID: ID!})` argument. For more information, see a `mutation.directConnectProcessTerminate` Reference: Gateway examples on page 63 .
`mutation.openDirectConnection`	`mutation.directConnectOpen`	The `mutation.openDirectConnection (input {endpointID: ID})` optional argument maps to the `mutation.directConnectOpen (input {endpointID: ID!})` required argument. For more information, see a `mutation.directConnectOpen` Reference: Gateway examples on page 63 .
`mutation.pingDirectConnection`	`mutation.directConnectPing`	The `mutation.PingDirectConnection (input)` argument maps one-to-one to the `mutation.directConnectPing (input)` argument. For more information, see a `mutation.directConnectPing` Reference: Gateway examples on page 63 .
`query.directEndpoint`	`query.directConnectEndpoint`	The `query.directEndpoint (input: {endpointID: ID})` does not map directly to a replacement argument. Instead of passing an endpoint ID, pass an open connection ID in the `query.directConnectEndpoint (input: {connectionID: ID!})` argument.

©2026 Tanium Inc. All Rights Reserved

Legacy field	Replacement field	Notes
		For more information, see a `query.directConnectEndpoint.performance` Reference: Gateway examples on page 63 , a `query.directConnectEndpoint.processes` Reference: Gateway examples on page 63 , or a `query.directConnectEndpoint.alerts` Reference: Gateway examples on page 63 .

©2026 Tanium Inc. All Rights Reserved