



Tanium™ Containers Deployment Guide

Version 1.0.1

July 22, 2021

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://docs.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 services provided 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 to make interaction with Tanium software more intuitive and to accelerate the time to success. To ensure high accessibility standards, Tanium complies with the U.S. Federal regulations - specifically Section 508 of the Rehabilitation Act of 1998. We have conducted third-party accessibility assessments over the course of product development for many years, and most recently a comprehensive audit against the WCAG 2.1 / VPAT 2.3 standards for all major product modules was completed in September 2019. 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 best efforts to address any gaps quickly, as is feasible, given the severity of the issue and scope of the changes. These objectives are factored 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 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, email Tanium Support at support@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 their respective owners.

© 2021 Tanium Inc. All rights reserved.

Table of contents

- Overview** 5
 - Tanium™ Client Container 5
 - Operating modes 5
 - Integration with other Tanium products 6
 - Trends 6
- Getting started** 7
 - Step 1: Review the requirements 7
 - Step 2: Obtain the Tanium™ Client Container 7
 - Step 3: Import the Containers solution on the Tanium Server 7
 - Step 4: Install and configure the Tanium Client Container 7
 - Step 5: Verify the installation 7
 - Step 6: (Optional) Import the Containers board in Tanium™ Trends 7
 - Step 7: Explore the Containers solution 7
 - Step 8: Monitor Containers metrics 8
- Tanium Containers requirements** 9
 - Tanium dependencies 9
 - Third-party software 9
- Installing Tanium Containers** 10
 - Before you begin 10
 - Obtain the Tanium Client Container 10
 - Import the Containers solution on the Tanium Server 10
 - Install and configure the Tanium Client Container 10
 - Unzip the Tanium Client Container ZIP file 11
 - Push the Tanium Client Container to the image registry 11
 - Configure the Tanium Client Container 12
 - configMAP 12
 - secret 12

DaemonSet	13
Deploy the Tanium Client Container	15
Verify Containers	15
Import the Containers board in Trends	16
What to do next	16
Troubleshooting	17
Unable to view or select content	17
Gather details for the Tanium Client Container	17
Uninstall the Tanium Client Container	17
Uninstall the Tanium Containers solution	18
Contact Tanium Support	18
Reference: Tanium Containers sensors	19
Container Host Operating System	19
Container Image	19
Container Image Name	20
Container Labels	21
Container Name with Image Hash	21
Container Network	21
Container PID Count	22
Container Running Processes	23
Container Runtime	24
Container Stats	24
Container Uptime	25
Is Managed Container Host	26
Is Tanium Client Container	26
Kubernetes Environment	26
Kubernetes Pods	27
Running Containers	27

Overview

With Tanium™ Containers, you can extend the visibility of the Tanium™ Core Platform to containers that run on the endpoints in your environment. Tanium Containers provides:

- Container orchestration software versions
- Cloud-based container service information
- Runtime visibility to containers
- Validation that the correct container images are in use
- Insight into container configuration and permissions
- Visibility into container network connectivity

Tanium™ Client Container

To use the Tanium Core Platform to monitor containers on endpoints in an enterprise deployment, install and configure the Tanium™ Client Container on those endpoints. The Tanium Client Container is a containerized version of the Tanium Client that provides visibility into running containers in orchestrated worker environments. The Tanium Client Container also includes tools to query and parse data from the running containers to provide data to the sensors from the Containers solution.

The Tanium Client Container runs directly on container nodes and is compliant with the Open Container Initiative (OCI).



NOTE

The Tanium Client that runs inside the Tanium Client Container is not upgradable. To switch to a new version of the Tanium Client in the Tanium Client Container, download a new version of the Tanium Client Container image, load it into your registry, and re-apply the Tanium Client Container DaemonSet described in [Installing Tanium Containers on page 10](#).

Operating modes

The Tanium Client Container runs in one of two modes: *client mode* and *tools mode*. The Tanium Client Container automatically chooses a mode at runtime.

Client mode

The Tanium Client Container operates in client mode if the Kubernetes worker node does not already have a Tanium Client. In client mode, the Tanium Client Container communicates directly with the Tanium Server as a Tanium Client.



When in client mode, the Tanium Client Container only responds to sensors in the Tanium Containers solution. This prevents the Tanium Server from treating the Tanium Client Container as a traditional endpoint. The Tanium Client Container is a Tanium Client but, as a container, it is not a traditional endpoint that runs packages or contains endpoint tools installed by Tanium solutions.

Tools mode

The Tanium Client Container operates in tools mode if the Kubernetes node already contains a Tanium Client. In tools mode, the Tanium Client Container provides tools to query and parse data from running containers to the existing Tanium Client. The Tanium Client Container continues to run as a paused container. In this mode, the existing Tanium Client responds to container sensors in addition to general (non-container) sensors.

Integration with other Tanium products

Containers has integration with Tanium™ Trends for additional reporting of related data.

Trends

Trends can feature a **Containers** board that shows container usage across the environment. The following panels are in the **Containers** board:

- Running Containers
- Running Pods
- Vendor
- Kubernetes Service
- Kubernetes Version
- Node Operating System
- Container Runtime
- Container Runtime Version
- Container Image Hash
- Privileged Containers
- Container Breaching Paving Policy
- Multi-Process Containers

You manually import the **Containers** board. For more information, see [Import the Containers board in Trends on page 16](#).

Getting started

Step 1: Review the requirements

Review the Tanium requirements and supported container configurations. See [Tanium Containers requirements on page 9](#).

Step 2: Obtain the Tanium™ Client Container

To use the Tanium Core Platform to monitor containers on endpoints in an enterprise deployment, install and configure the Tanium Client Container on those endpoints. [Contact Tanium Support](#) to obtain the Tanium Client Container ZIP file.

Step 3: Import the Containers solution on the Tanium Server

To ask container-related questions through Tanium™ Interact and the Tanium™ Console, import the Containers solution. See [Import the Containers solution on the Tanium Server on page 10](#).

Step 4: Install and configure the Tanium Client Container

Set up and configure the Tanium Client Container on your container environment nodes. See [Install and configure the Tanium Client Container on page 10](#).

Step 5: Verify the installation

Ask a question that uses a sensor from the Containers solution to verify the hosts with the Tanium Client Container respond to the Tanium Server. See [Verify Containers on page 15](#).

Step 6: (Optional) Import the Containers board in Tanium™ Trends

Import the Containers board in Trends to view container metrics in your environment. See [Import the Containers board in Trends on page 16](#).

Step 7: Explore the Containers solution

Explore the sensors in the Containers solution to see which questions are available in Interact and the Tanium Console. See [Reference: Tanium Containers sensors on page 19](#).

Step 8: Monitor Containers metrics

From the Trends menu, go to Boards and then click **Containers** to view the number of **Running Containers** and **Running Pods** and the **Container Inventory** and **Container Hygiene** sections.

Tanium Containers requirements

Review the requirements before you install and use Tanium Containers.

Tanium dependencies

Component	Requirement
Tanium™ Core Platform ¹	7.4.1 or later (requires access to the <code>tanium-init.dat</code>)
Tanium products	Tanium Interact 2.4.50 or later to query the Container sensors (Optional) Trends 3.6 or later to view the Containers board

¹ The license entitlement for the Tanium Core Platform includes the Containers solution.

Third-party software

Tanium Containers supports the following container versions in on-premises and cloud environments.



Confirm that the Tanium Client Container is hosted on a private container registry to securely provide the Tanium Client Container image. Do not host the image on a public container registry.

Software	Requirement	Supported runtime environments
Kubernetes	1.15 or later	<ul style="list-style-type: none">Use Linux-based worker nodes with the following operating systems (OSes):<ul style="list-style-type: none">BottlerocketCoreOSUbuntuAny Linux OS supported by the Tanium Client. For more information, see Tanium Client Management User Guide: Client version and host system requirements.Use a private container registry or similar to provide the Tanium Client Container to the worker nodes.Use Containerd, cRIO, or Docker as the container runtime.
Red Hat OpenShift	3.x or later	<ul style="list-style-type: none">Red Hat Enterprise Linux (RHEL)Red Hat Enterprise Linux CoreOS (RHCOS)

Installing Tanium Containers

Perform the following steps to import the Containers solution on the Tanium™ Server, and to obtain, install, and configure the Tanium Client Container on endpoints with container images.

Before you begin

- Read the [release notes](#).
- Review the [Tanium Containers requirements on page 9](#).

Obtain the Tanium Client Container

[Contact Tanium Support](#) to obtain a download link for the Tanium Client Container ZIP file.



After download, verify the SHA256 checksum of the ZIP file matches the SHA256 checksum listed on the Tanium download link.

BEST PRACTICE

Import the Containers solution on the Tanium Server

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



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. Otherwise, perform these steps on each Tanium Server.

1. Sign in to the Tanium Console with an administrator account.
2. From the Main menu, go to **Administration > Configuration > Solutions**.
3. In the **Content** section, select the checkbox for **Containers** and click **Install**.
4. Review the content to import and click **Begin Install**.
5. If prompted, click **Yes** to confirm the action.

Install and configure the Tanium Client Container

Use the following steps to set up and configure the Tanium Client Container on your container environment nodes. The steps are the same for both nodes that contain the Tanium Client and nodes that do not have an existing Tanium Client. The Tanium Client Container automatically detects an existing Tanium Client on the host and selects the appropriate operating mode. For more information, see [Operating modes on page 5](#).



The commands provided in this section are examples. Make sure to adjust your own commands to match your environment.



The following examples use an Amazon Elastic Kubernetes Service (EKS) environment in region `us-west-1` with the account `12345678` and the AWS username `awsadmin`. The concepts apply to any Kubernetes environment. Additionally, the examples use `tanium/tcc` as the name of the Tanium Client Container image and `tcc` for the name of the Kubernetes app. Adjust your own commands accordingly.

Unzip the Tanium Client Container ZIP file

Move or copy the ZIP file into your preferred directory or folder, and then extract the contents of the file.

Docker example:

```
docker image load --input tanium-client-container-2.0.1-7.4.5.1204.tar
```

CTR example:

```
ctr image import "Tanium-client-container-2.0.1-7.4.5.1204.tar"
```

Push the Tanium Client Container to the image registry

Use the following steps to register the Tanium Client Container image with your private container registry.

1. Authenticate your local Docker command with the EKS registry. For example:

```
$ aws ecr get-login-password --region us-west-1 | docker login --username  
awsadmin --password-stdin 12345678.dkr.ecr.us-west-1.amazonaws.com
```

2. Tag the Tanium Client Container image in the registry. For example:

```
$ docker tag tanium/tcc:latest 12345678.dkr.ecr.us-west-  
1.amazonaws.com/tcc:latest
```

3. Push the image to the registry. For example:

```
$ docker push 12345678.dkr.ecr.us-west-1.amazonaws.com/tcc:latest
```



Some registries require you to create the repository beforehand and do not allow you to push images that are not configured.

Configure the Tanium Client Container

Perform the following steps to configure your Kubernetes environment.

CONFIGMAP

The Tanium Client Container requires two environment variables: `CONTAINER_RUNTIME` and `CONTAINER_RUNTIME_ENDPOINT`.

- The `CONTAINER_RUNTIME` variable must be `docker`, `containerd`, or `crio`. The value must match your Kubernetes environment.
- The `CONTAINER_RUNTIME_ENDPOINT` variable must point to the CRI-compatible container socket that is used by your container runtime.

Create a `configmap.yaml` file such as the following example to declare the metadata and environment variables for the Tanium Client Container. You can also use the configuration file to apply ENV variables to the Tanium Client as well as the log level.

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: tcc-config
  namespace: default
  labels:
    app: tcc
data:
  CONTAINER_RUNTIME: "docker"
  CONTAINER_RUNTIME_ENDPOINT: "unix:///var/run/docker.sock"
```

SECRET

The Tanium Client Container requires the `tanium-init.dat` initialization file from the Tanium Server. The `tanium-init.dat` file allows Tanium Clients to register with the Tanium Server and use the Tanium Zone Server settings. For instructions on how to download the `tanium-init.dat` initialization file from the Tanium Server, see [Tanium Client Management User Guide: Configure client settings](#).

After you download the `tanium-init.dat` initialization file, use the following command to verify the Tanium Servers in the `server name list` in the file:

```
# TaniumClient pki show ./tanium-init.dat --verbose
```

To securely allow the Tanium Client Container access to the contents of the `tanium-init.dat` file, generate a Kubernetes secret. For example:

```
$ kubectl create secret generic tanium-init --from-file tanium-init.dat --
output=yaml --dry-run=client > secret-tanium-init.yaml
```



Be careful not to allow the `tanium-init.dat` file to be distributed or stored outside of your organization, such as in a publicly accessible source code repository or any other location accessible from the public internet. Limit the distribution to specific use in the deployment of Tanium Clients and the Tanium Client Container.

Though the `tanium-init.dat` file does not contain private keys and cannot be used to provide control over a Tanium environment, a user with malicious intent could use the file to connect an unapproved client and use this unauthorized access to learn how your organization uses Tanium.



In Tanium Core Platform 7.4.1 or later, you can also retrieve the `tanium-init.dat` file from the Tanium Server through the REST API.

DAEMONSET

A Kubernetes DaemonSet is a special container configuration that is automatically created for each node. The DaemonSet is commonly used for metrics, logging, and security tooling.

The DaemonSet configuration declares how the Tanium Client Container runs and combines data from the `configmap` and `secret`.



The Tanium Client Container must run in privileged mode; be sure to limit access to the Tanium Client Container.

Create a `daemonset.yaml` file that declares essential configurations and volume mounts to allow the Tanium Client Container to function properly. For example:

```
---
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: tcc
  namespace: default
  labels:
    app: tcc
spec:
  selector:
    matchLabels:
```

```
    app: tcc
template:
  metadata:
    labels:
      app: tcc
  spec:
    hostIPC: false
    hostPID: true
    hostNetwork: true
    restartPolicy: Always
  containers:
    - name: tcc
      image: 12345678.dkr.ecr.us-west-1.amazonaws.com/tcc:latest
      imagePullPolicy: Always
      volumeMounts:
        - name: tanium-init-volume
          mountPath: /opt/Tanium/init
          readOnly: true
        - name: host-var-run
          mountPath: /host/var/run
        - name: host-run
          mountPath: /host/run
        - name: host-root
          mountPath: /host/root
          readOnly: true
      env:
        - name: CONTAINER_RUNTIME
          valueFrom:
            configMapKeyRef:
              name: tcc-config
              key: CONTAINER_RUNTIME
        - name: CONTAINER_RUNTIME_ENDPOINT
          valueFrom:
            configMapKeyRef:
              name: tcc-config
              key: CONTAINER_RUNTIME_ENDPOINT
      securityContext:
        runAsUser: 0
        runAsGroup: 0
        privileged: true
      volumes:
        - name: tanium-init-volume
```

```
secret:
  secretName: tanium-init
  defaultMode: 0400
- name: host-var-run
  hostPath:
    path: /var/run
    type: Directory
- name: host-run
  hostPath:
    path: /run
    type: Directory
- name: host-root
  hostPath:
    path: /
    type: Directory
```

Deploy the Tanium Client Container

With the **kubect1** command configured for your cluster environment, apply each of the YAML files. For example:

```
$ kubect1 apply --filename="secret-tanium-init.yaml"
```

```
$ kubect1 apply --filename="configmap.yaml"
```

```
$ kubect1 apply --filename="daemonset.yaml" --selector="app=tcc"
```

When complete, the Tanium Client Container should be applied to your Kubernetes environment, each existing node creates a container with the Tanium Client Container, and each new node now runs a Tanium Client Container container as part of the creation process. You can verify the DaemonSet of the Tanium Client Container with the following command:

```
$ kubect1 get --selector="app=tcc" daemonsets
```

Verify Containers

After you install the Containers solution on the Tanium Server and install the Tanium Client Container on at least one container host, use the **Is Managed Container Host** sensor to verify the Tanium Server retrieves results from the Tanium Client Container.

1. Sign in to the Tanium Server as a user with the Administrator reserved role, or a user with the **Ask Dynamic Questions** permission.
2. On the Tanium **Home** page, enter the following question in the **Explore Data** field:

```
Get Is Managed Container Host
```

3. Click **Search**.

The **Question Results** page opens to show answers from endpoints.

- Endpoints that are container hosts with the Tanium Client Container respond with **True**.
- Endpoints that are not container hosts with the Tanium Client Container do not respond and appear as **[no results]**.

Verify that there are one or more **True** responses to confirm that the Tanium Client Container responds.

Import the Containers board in Trends

Import the **Containers** board in Trends to provide you with charts on container usage in the environment. The Tanium Client Container ZIP file includes the `tanium-trends-boards-containers.json` file for the **Containers** board. For information on how to import a board from a JSON file in Trends, see [Tanium Trends User Guide: Import boards, sections, and panels](#).

To view the **Containers** board in Trends, you must have the following permissions:

- **Trends** show permission
- **Trends API Board** read for the Trends content set
- **Trends Data** read for the Trends content set

What to do next

- In Trends, click **Boards > Containers** to monitor metrics.
- See [Reference: Tanium Containers sensors on page 19](#) for a list of sensors in the Containers solution.

Troubleshooting

If you encounter unexpected behavior with Tanium Containers, use the information contained here to troubleshoot the issue.



NOTE

The troubleshooting examples use `tanium/tcc` as the name of the Tanium Client Container image and `tcc` for the name of the Kubernetes app. Adjust your own commands accordingly.

Unable to view or select content

In environments that enable role-based access control (RBAC), users cannot access content to which they do not have permission. Sensors are among those objects that are managed through RBAC. If you are unable to access sensors in the Tanium Containers solution, make sure your user account has sufficient permission to the **Containers** content set.

- You must have read permission to the **Containers** content set to view sensors in the Tanium Containers solution.
- You must have write permission to the **Containers** content set to add, edit, or delete sensors in the Tanium Containers solution.
- You must have the **Trends API Board** read, **Trends Data** read, and **Trends** show permissions to the **Trends** content set to view the Containers board in Trends.

Gather details for the Tanium Client Container

If you experience issues when you deploy or run the Tanium Client Container on endpoints, use the **describe** command to view details for the Tanium Client Container. For example:

```
kubectl describe daemonset.apps/tcc
```

For more information and options, see the [describe command in the Kubernetes command reference](#).

Uninstall the Tanium Client Container

Run the following commands to uninstall the Tanium Client Container from the Kubernetes nodes:

```
kubectl delete daemonset.apps/tcc --wait=true --cascade=foreground
```

```
kubectl delete configmap/tcc --wait=true
```

```
kubectl delete secret/tanium-init --wait=true
```

Uninstall the Tanium Containers solution

Perform the following steps to remove the Tanium Containers solution from the Tanium Server.



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. Otherwise, perform these steps on each Tanium Server.

1. Sign in to the 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 checkbox for **Containers** and click **Uninstall**.
4. Review the summary and click **Yes**.

Contact Tanium Support

To contact Tanium Support for help, sign in to <https://support.tanium.com>.

Reference: Tanium Containers sensors

Use the sensors contained in the Containers solution to retrieve information from the containers in the environment.

- Tanium Client Containers that run in client mode only respond to sensors in the Containers solution.
- Tanium Client Containers that run in tools mode respond to the sensors in the Containers solution, while the Tanium Clients on the Kubernetes worker nodes respond to non-container sensors.



Because containers are intended to be temporary, the sensors in the Containers solution cannot be registered with the Tanium Data Service. For more information on the Tanium Data Service, see [Tanium Console User Guide: Manage sensor results collection](#).

Container Host Operating System

Category: Containers

Returns the Operating System generation of a managed container host.

Columns

Name	Description	Type	Hidden
Container Host Operating System		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Image

Category: Containers

Returns information about the images used to instantiate running containers.

Parameters

Name	Description	Type	Possible / Default values
Container ID	If specified, only return data for the specified container ID. Otherwise, return data for all containers.	Text	

Columns

Name	Description	Type	Hidden
Container ID		Text	No
Name		Text	No
Image SHA256		Text	No
Image Location		Text	No
POD ID		Text	No
Privileged?		Text	No
Labels		Text	No
Process Path		Text	No
Process Args		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Image Name

Category: Containers

Returns the names of images used to instantiate running containers.

Supported Platforms

Platform	Query Type
Linux	Shell

Container Labels

Category: Containers

Returns labels defined for running containers.

Columns

Name	Description	Type	Hidden
Container ID		Text	No
Labels		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Name with Image Hash

Category: Containers

Returns the names and hashes of images (not containers, but the template used to instantiate the container).

Columns

Name	Description	Type	Hidden
Container		Text	No
Image SHA256		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Network

Category: Containers

Returns network details for running containers.

Parameters

Name	Description	Type	Possible / Default values
Container ID	If specified, only return data for the specified container ID. Otherwise, return data for all containers.	Text	

Columns

Name	Description	Type	Hidden
Container ID		Text	No
Protocol		Text	No
Local Address		Text	No
Remote Address		Text	No
Created		Text	No
State		Text	No
PID		Text	No
Application		Text	No
Command Line		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container PID Count

Category: Containers

Returns the number of Process IDs (PIDs) for running containers.

Parameters

Name	Description	Type	Possible / Default values
Container ID	If specified, only return data for the specified container ID. Otherwise, return data for all containers.	Text	

Columns

Name	Description	Type	Hidden
Container ID		Text	No
Name		Text	No
PID Count		Numeric	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Running Processes

Category: Containers

Returns process details for running containers.

Parameters

Name	Description	Type	Possible / Default values
Container ID	If specified, only return data for the specified container ID. Otherwise, return data for all containers.	Text	

Columns

Name	Description	Type	Hidden
Container ID		Text	No
Executable Path		Text	No
Command		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Runtime

Category: Containers

Provides detail regarding the executor of the containers, the "Container Runtime."

Columns

Name	Description	Type	Hidden
Container Runtime Name		Text	No
Container Runtime Version		Text	No
Container Runtime API Version		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Stats

Category: Containers

Provides runtime resource utilization statistics for running containers.

Parameters

Name	Description	Type	Possible / Default values
Container ID	If specified, only return data for the specified container ID. Otherwise, return data for all containers.	Text	

Columns

Name	Description	Type	Hidden
Container ID		Text	No
Name		Text	No
CPU Percentage		Numeric	No
Memory Percentage		Numeric	No
Memory Limit		File Size	No
Network TX		File Size	No
Network RX		File Size	No
Disk Read		File Size	No
Disk Write		File Size	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Uptime

Category: Containers

Provides information regarding the age of running containers.

Parameters

Name	Description	Type	Possible / Default values
Container ID	If specified, only return data for the specified container ID. Otherwise, return data for all containers.	Text	

Columns

Name	Description	Type	Hidden
Container ID		Text	No
Name		Text	No
Uptime		Time Duration	No

Supported Platforms

Platform	Query Type
Linux	Shell

Is Managed Container Host

Category: Containers

Identifies managed endpoints that are container hosts and have the TCC/TCC tools.

Supported Platforms

Platform	Query Type
Linux	Shell

Is Tanium Client Container

Category: Containers

Returns **True** if the Tanium Client runs in a Tanium Client Container, **False** otherwise. Windows, macOS, Solaris, and AIX endpoints always return **False**.

Supported Platforms

Platform	Query Type
Linux	Shell
macOS	Shell
Windows	VBScript

Kubernetes Environment

Category: Containers

Identifies the Kubernetes environment details, typically of the cloud provider.

Columns

Name	Description	Type	Hidden
Infrastructure Provider		Text	No
Kubernetes Product		Text	No

Name	Description	Type	Hidden
Kubernetes Version		Text	No
Kubernetes Service Host		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Kubernetes Pods

Category: Containers

Enumerates all Kubernetes running pods including those typically hidden from view.

Columns

Name	Type	Description
Pod ID	Text	
Name	Text	
Namespace	Text	
Status	Text	
Created	Text	
Attempt	Text	
Runtime	Text	

Supported Platforms

Platform	Query Type
Linux	Shell

Running Containers

Category: Containers

Identifies all running containers, including those hidden and unknown to the orchestration layer (such as System or Rogue containers).

Parameters

Name	Description	Type	Possible / Default values
Show unorchestrated only	Show containers that are running on the host, but not reported by the orchestrator.	Checkbox	Unchecked
Hide pause containers	Hide pause containers /pause and /usr/bin/pod	Checkbox	Unchecked

Columns

Name	Description	Type	Hidden
Container ID		Text	No
Runtime		Text	No
Source		Text	No
Status		Text	No
Created		Text	No
Pid		Text	No
MD5Sum		Text	No
RootFS		Text	No
OS		Text	No
Pid Count		Integer	No
LWP Count		Integer	No
Arguments		Text	No
Orchestrated		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell