Tanium™ Containers Deployment Guide

Version 1.0.0

July 09, 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.

Table of contents

Overview **6**

Tanium™ Client Container 6

Operating modes 6

Integration with other Tanium products 7

Trends 7

Getting started **8**

Step 1: Review the requirements 8

Step 2: Obtain the Tanium™ Client Container 8

Step 3: Install and configure the Tanium Client Container 8

Step 4: Verify the installation 8

Step 5: Explore the Containers solution 8

Tanium Containers requirements **9**

Third-party software 9

Installing Tanium Containers **10**

Before you begin 10

Obtain the Tanium Client Container 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 11

configMAP 11

secret 12

DaemonSet 13

Deploy the Tanium Client Container 14

Verify Containers 15

What to do next 15

Troubleshooting **16**

Unable to view or select content ... 16
Gather details for the Tanium Client Container ... 16
Uninstall the Tanium Client Container ... 16
Uninstall the Tanium Containers solution ... 17
Contact Tanium Support ... 17
Reference: Tanium Containers sensors ... 18
Container Host Operating System ... 18
Container Image ... 18
Container Labels ... 19
Container Name with Image Hash ... 20
Container Network ... 20
Container PID Count ... 21
Container Running Processes ... 22
Container Runtime ... 22
Container Stats ... 23
Container Uptime ... 24
Image Name ... 24
Is Managed Container Host ... 25
Is Tanium Client Container ... 25
Kubernetes Environment ... 25
Kubernetes Pods ... 26
Running Containers ... 26

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).

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 Tanium as a Service (TaaS) as a Tanium Client.

NOTE When in client mode, the Tanium Client Container only responds to sensors in the Tanium Containers solution. This prevents TaaS 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 built in integration with Tanium™ Trends for additional reporting of related data.

Trends

Trends features 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

NOTE The **Containers** board is not yet available in a released version of Trends. If necessary, you can manually import the **Containers** board that is included as a JSON file with the Tanium Client Container. For information on how to import a board from a JSON file in Trends, see Tanium Trends User Guide: Import boards, sections, and panels.

Getting started

Step 1: Review the requirements

Review the 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: 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 4: 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 5: 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 18.

Tanium Containers requirements

Review the requirements before you install and use Tanium Containers.

Third-party software

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

A private container registry is required to securely provide the Tanium Client Container image.

Software	Requirement	Supported runtime environments

Kubernetes	1.15 or later	• Use Linux-based worker nodes with the following operating systems (OSes):
		  • Bottlerocket
		  • CoreOS
		  • Ubuntu
		  • Any 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	• Red Hat Enterprise Linux (RHEL) • Red Hat Enterprise Linux CoreOS (RHCOS)

Installing Tanium Containers

Perform the following steps 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.

> NOTE Tanium Containers requires the Containers solution to be in Tanium as a Service (TaaS). TaaS automatically handles installations and upgrades for the Containers solution. You only need to install and configure the Tanium Client Container on endpoints with containers.

Obtain the Tanium Client Container

Contact Tanium Support to obtain a download link for the Tanium Client Container ZIP file.

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

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 6.

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

> NOTE 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-2.0.0-7.4.5.1204.tar
```

CTR example:

```
ctr image import "Tanium-client-container-2.0.0-7.4.5.1204.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/dockershim.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. For instructions on how to download the `tanium-init.dat` initialization file from TaaS, see Tanium Console User Guide: Download infrastructure configuration files (keys).

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.

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

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

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

```
$ kubectl 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:

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

Verify Containers

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

1. Sign in to TaaS as a user with the Admin 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.

What to do next

See Reference: Tanium Containers sensors on page 18 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.

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

To uninstall the Tanium Containers solution from TaaS, contact Contact Tanium Support on page 17.

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 container hosts

Columns

Name	Description	Type	Hidden
Container Host Operating System		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Image

Category: Containers

Report the unique container image names

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 Labels

Category: Containers

Container Labels

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

Container Name with Image Hash

Columns

Name	Description	Type	Hidden
Container		Text	No
Image SHA256		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Network

Category: Containers

Container Network

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

Name	Description	Type	Hidden
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

Container PID Count

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

Container Running Processes.

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

Container Runtime

Columns

Name	Description	Type	Hidden
Container Runtime Name		Text	No

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

Supported Platforms

Platform	Query Type
Linux	Shell

Container Stats

Category: Containers

Report container statistics

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

Name	Description	Type	Hidden
Disk Write		File Size	No

Supported Platforms

Platform	Query Type
Linux	Shell

Container Uptime

Category: Containers

Report the uptime for each container.

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

Image Name

Category: Containers

Image Name

Supported Platforms

Platform	Query Type
Linux	Shell

Is Managed Container Host

Category: Containers

Is Managed Container Host.

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

Kubernetes Environment

Columns

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

Name	Description	Type	Hidden
Kubernetes Service Host		Text	No

Supported Platforms

Platform	Query Type
Linux	Shell

Kubernetes Pods

Category: Containers

Enumerate Pods From ContainerD

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

Running containers, including orchestrated and unorchestrated 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