Monitoring Docker

Docker PowerPack version 106

Table of Contents

Chapter 1

Introduction

Overview

This manual describes how to monitor the Docker platform in SL1 using the *Docker* PowerPack.

For more information about monitoring Docker, watch the video at https://sciencelogic.com/product/resources/sl1-kubernetes-and-docker-container-monitoring.

The following sections provide an overview of the Docker platform and the *Docker* PowerPack:

NOTE: ScienceLogic provides this documentation for the convenience of ScienceLogic customers. Some of the configuration information contained herein pertains to third-party vendor software that is subject to change without notice to ScienceLogic. ScienceLogic makes every attempt to maintain accurate technical information and cannot be held responsible for defects or changes in third-party vendor software. There is no written or implied guarantee that information contained herein will work for all third-party variants. See the End User License Agreement (EULA) for more information.

What Does the Docker PowerPack Monitor?

The *Docker* PowerPack includes:

- An example Basic/Snippet Credential and an example SSH/Key Credential for discovering Docker devices

- Dynamic Applications and Run Book Actions to discover, model, and monitor the following Docker component devices:
    - Hosts
    - Containers
    - Swarms
    - Stacks
    - Services
- Device Classes for each type of Docker component device SL1 monitors
- Event Policies that are triggered when Docker component devices meet certain status criteria

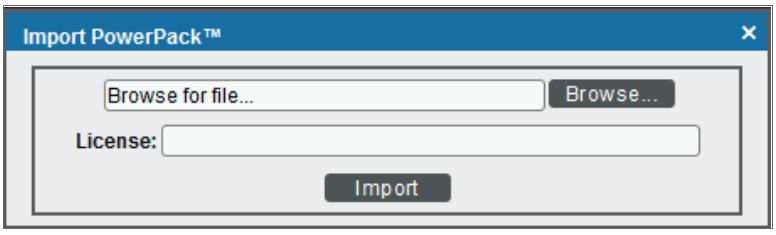
Installing the Docker PowerPack

Before completing the steps in this manual, you must import and install the latest version of the *Docker* PowerPack.

TIP: By default, installing a new version of a PowerPack overwrites all content from a previous version of that PowerPack that has already been installed on the target system. You can use the ***Enable Selective PowerPack Field Protection*** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. (For more information, see the ***System Administration*** manual.)

To download and install a PowerPack:

1. Download the PowerPack from the ScienceLogic Support Site.
2. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).
3. In the **PowerPack Manager** page, click the **[Actions]** button, then select *Import PowerPack*.
4. The **Import PowerPack** dialog box appears:



5. Click the **[Browse]** button and navigate to the PowerPack file.
6. When the **PowerPack Installer** modal appears, click the **[Install]** button to install the PowerPack.

NOTE: If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPack Manager** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

Chapter 2

Configuration and Discovery

Overview

The following sections describe how to configure and discover the Docker platform and its component devices for monitoring by SL1 using the *Docker* PowerPack:

Prerequisites for Monitoring Docker

If you are using Secure Shell (SSH) to monitor Docker or Kubernetes nodes in conjunction with the *Kubernetes* PowerPack, you must install cURL 7.40 or greater on all of the Docker hosts that you want to monitor, prior to discovery. You must then run the following cURL commands on each of those hosts:

- `curl --unix-socket /var/run/docker.sock http://docker/containers/json`
- `curl --unix-socket /var/run/docker.sock http://docker/containers/\[container_id]/json`
- `curl --unix-socket /var/run/docker.sock http://docker/containers/\[container_id]/stats?stream=0`

If you are using a Basic/Snippet credential, before you can monitor the Docker platform and its component devices in SL1 using the *Docker* PowerPack, you must first follow the instructions in the *Enabling the Docker API* section. These steps enable the Dynamic Applications in the *Docker* PowerPack to communicate with and gather data from the Docker API.

NOTE: You do not need to enable the API if you are using SSH to monitor Docker.

WARNING: If you choose to enable the API when monitoring Docker versions through 18.06.1-ce-rc2, be aware that a vulnerability exists. The API endpoints behind the 'docker cp' command are vulnerable to a symlink-exchange attack. (CVE-2018-15664).

Enabling the Docker API

Before you discover Docker components using the *Docker* PowerPack, you must first enable the Docker API. This section describes how to do so for Windows, CentOS, Red Hat Enterprise Linux (RHEL), and Oracle Linux operating systems.

NOTE: If you are using SSH to monitor Docker, skip this section and go to the *Creating an SSH/Key Credential* section.

Windows

To enable the Docker API for Windows using the Docker Toolbox:

1. Start Docker Quickstart Terminal.
2. To determine the IP address of the Docker host machine, type the following command:

```
$ docker-machine ip
```

3. Log in to the host machine:

```
$ docker-machine ssh
```

4. Navigate to Boot2Docker:

```
$ cd /var/lib/boot2docker
```

5. Edit the Boot2Docker profile:

```
$ sudo vi profile
```

6. In the profile, change "DOCKER_HOST" to "DOCKER_HOST='-H tcp://0.0.0.0:*[port number]*'", and set DOCKER_TLS=no.
7. Exit the SSH session, and then restart Docker:

```
$ exit
$ docker-machine restart
```

8. To verify that the Docker API is accessible, open a browser and navigate to http:*[IP address]*:*[port number]*/version.

   If the Docker API is successfully enabled, the version returns something similar to the following:

```
{"Version":"17.10.0-ce","ApiVersion":"1.33","MinAPIVersion":"1.12","GitCommit":
"f4ffd25","GoVersion":"go1.8.3","Os":"linux","Arch":"amd64","KernelVersion":
"4.4.93-boot2docker","BuildTime":"2017-10-17T19:05:23.000000000+00:00"}
```

CentOS

To enable the Docker API for CentOS:

1. Log in to the command-line interface of the server running Docker and navigate to systemd/system:

```
$ cd /etc/systemd/system
```

2. Create a new "docker.service.d" folder, then navigate to that folder:

```
$ mkdir docker.service.d
$ cd docker.service.d
```

3. Create a new docker.conf file:

```
$ vi docker.conf
```

4. Type the following:

```
INSERT
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:[port number] -H
unix://var/run/docker.sock
```

5. Reload daemon, restart Docker, and open the port on the firewall by typing the following:

```
$ systemctl daemon-reload
$ systemctl restart docker
$ firewall-cmd --add-port=[port number]/tcp
```

6. Verify that the Docker API is accessible by typing the following:

```
$ *curl http://localhost:[port number]/version*
```

If the Docker API is successfully enabled, the version returns something similar to the following:

```
{"Version":"17.06.1-ce","ApiVersion":"1.30","MinAPIVersion":"1.12","GitCommit"
:"874a737","GoVersion":"go1.8.3","Os":"linux","Arch":"amd64","KernelVersion":
"3.10.0-514.26.2.el7.x86_64","BuildTime":"2017-08-17T23:01:50.155177940+00:00"}
```

RHEL 7 and Oracle Linux 7

To enable the Docker API for RHEL 7 or Oracle Linux 7:

1. Log in to the command-line interface of the server running Docker and navigate to systemd/system:

```
$ cd /etc/systemd/system
```

2. Edit the service.docker file:

```
$ sudo vi docker.service
```

3. Create or edit the file to ensure that it has a [Service] section and a line that starts with "`ExecStart=/usr/bin/dockerd`". Add "`-H tcp://0.0.0.0:[port number] -H unix:///var/run/docker.sock`" so that the updated line looks like this:

```
ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:4243 -H unix:///var/run/docker.sock
```

4. Open the firewall port, if needed, and then reload daemon and restart restart Docker by typing the following:

```
$ sudo firewall-cmd --add-port=[port number]/tcp
$ sudo firewall-cmd --reload
$ sudo systemctl daemon-reload
$ sudo systemctl restart docker
```

5. Verify that the Docker API is accessible by typing the following:

```
$ curl http://[IP address]:[port number]/version
```

If the Docker API is successfully enabled, the version returns something similar to the following:

```
{"Version":"17.06.2-ee-4","ApiVersion":"1.30","MinAPIVersion":"1.12","GitCommit":
"dd2c358","GoVersion":"go1.8.3","Os":"linux","Arch":"amd64","KernelVersion":
"3.10.0-514.el7.x86_64","BuildTime":"2017-10-12T16:19:56.386620861+00:00"}
```

NOTE: For Linux distributions, some versions of the firewall require the "--permanent" flag. This is likely the case if the first attempt at automatic discovery fails and manually aligned Dynamic Applications are not collecting.

Configuring a Docker Credential

The *Docker* PowerPack includes an example Basic/Snippet Credential and an example SSH/Key Credential for your use. You can modify these to create your own Credentials that will enable SL1 to discover your Docker devices.

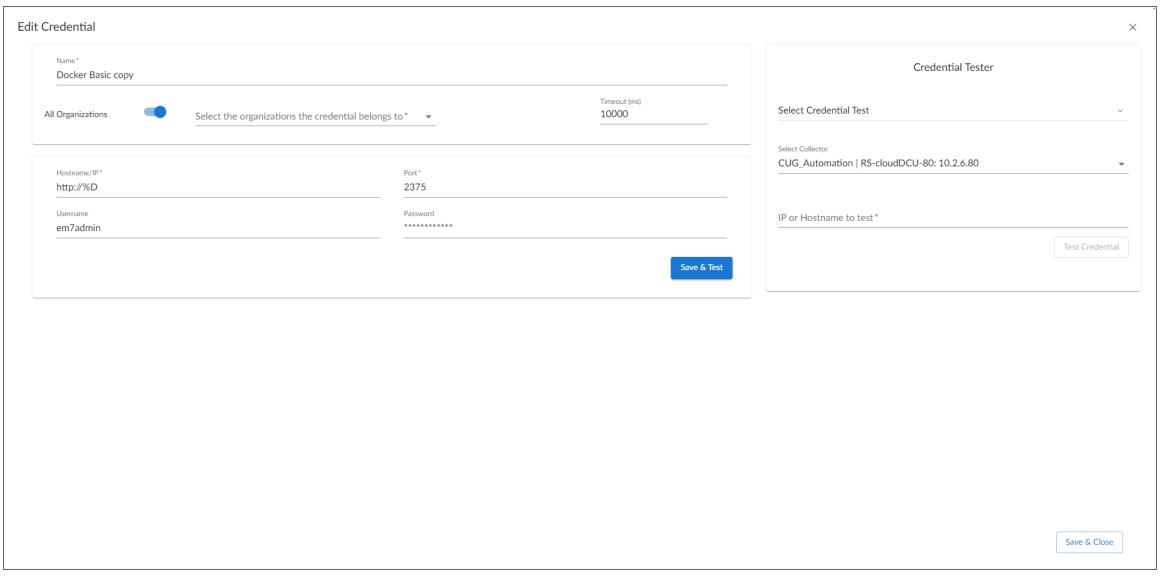
Creating a Basic/Snippet Credential

To configure SL1 to monitor the Docker platform using the Docker API, you must create a Basic/Snippet credential that allows the Dynamic Applications in the *Docker* PowerPack to connect with Docker hosts or swarms. An example Basic/Snippet credential that you can edit for your own use is included in the *Docker* PowerPack.

NOTE: If you are using an SL1 system prior to version 11.1.0, the new user interface does not include the ***Duplicate*** option for sample credential(s). ScienceLogic recommends that you use *the classic user interface and the Save As button* to create new credentials from sample credentials. This will prevent you from overwriting the sample credential(s).

To create a Basic/Snippet credential to access Docker hosts or swarms:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the example **Docker Basic** credential, click its **[Actions]** icon (⋯) and select ***Duplicate***. A copy of the credential, called **Docker Basic copy** appears.
3. Click the **[Actions]** icon (⋯) for the **Docker Basic copy** credential and select ***Edit***. The **Edit Credential** modal page appears:



4. Supply values in the following fields:

- ***Name***. Type a new name for the Docker credential.
- ***All Organizations***. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the ***What organization manages this service?*** drop-down field to align the credential with those specific organizations.
- ***Timeout(ms)***. Type "10000".
- ***Hostname/IP***. Type "%D".
- ***Port***. Type the port number you specified when you *enabled the Docker API*.
- ***Username***. Type a value for the username.
- ***Password***. Type a value for the password.

> **NOTE:** The Docker platform does not require a specific username and password to access the platform, but SL1 does require the ***Username*** and ***Password*** fields to have values when using Basic/Snippet credentials to monitor Docker. Therefore, those fields must have entries, but the values themselves do not matter.

5. Click the **[Save & Close]** button.

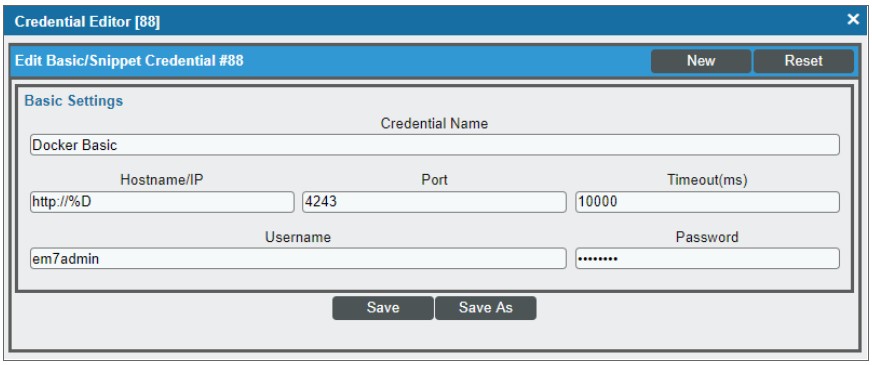
Creating a Basic/Snippet Credential in the SL1 Classic User Interface

To configure SL1 to monitor the Docker platform using the Docker API, you must create a Basic/Snippet credential that allows the Dynamic Applications in the *Docker* PowerPack to connect with Docker hosts or swarms. An example Basic/Snippet credential that you can edit for your own use is included in the *Docker* PowerPack.

To create a Basic/Snippet credential to access Docker hosts or swarms:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the example **Docker Basic** credential, and then click its wrench icon (). The **Edit Basic/Snippet Credential** modal page appears.

3. Complete the following fields:



- ***Credential Name***. Type a new name for the Docker credential.
- ***Hostname/IP***. Type "%D".
- ***Port***. Type the port number you specified when you *enabled the Docker API*.
- ***Timeout(ms)***. Type "10000".
- ***Username***. Type a value for the username.
- ***Password***. Type a value for the password.

NOTE: The Docker platform does not require a specific username and password to access the platform, but SL1 does require the ***Username*** and ***Password*** fields to have values when using Basic/Snippet credentials to monitor Docker. Therefore, those fields must have entries, but the values themselves do not matter.

4. Click the **[Save As]** button.
5. When the confirmation message appears, click **[OK]**.

Creating an SSH/Key Credential

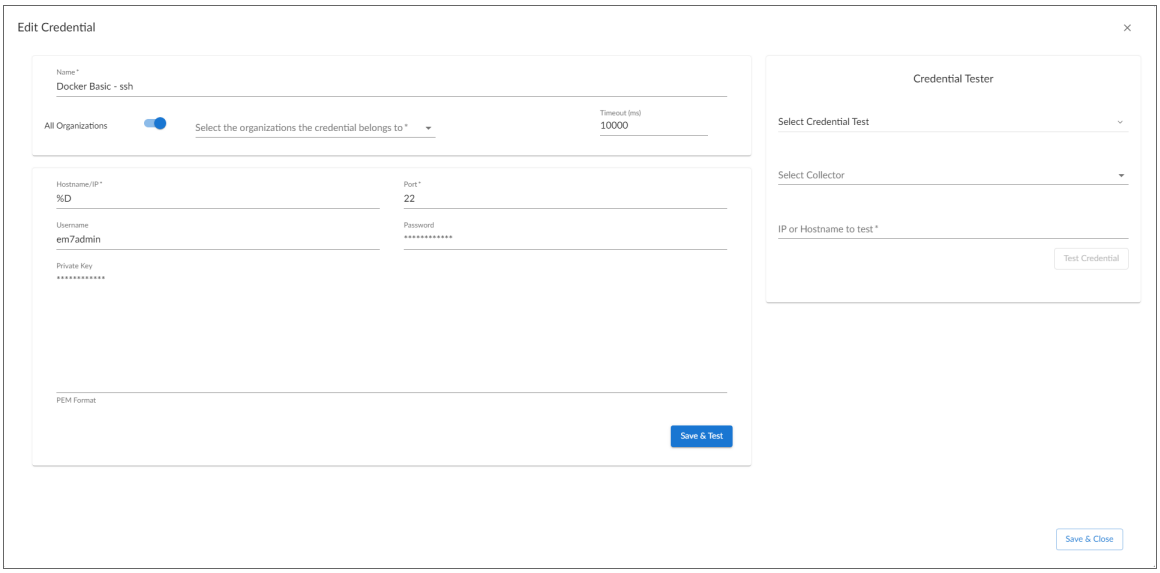
If you are using SSH to monitor Docker swarms, then you must create an SSH/Key credential that allows the Dynamic Applications in the *Docker* PowerPack to connect with Docker swarms. An example SSH/Key credential that you can edit for your own use is included in the *Docker* PowerPack.

NOTE: You can also use an SSH credential in conjunction with the *Kubernetes* PowerPack to monitor the Docker infrastructure for a Kubernetes cluster.

NOTE: If you are using an SL1 system prior to version 11.1.0, the new user interface does not include the ***Duplicate*** option for sample credential(s). ScienceLogic recommends that you use *the classic user interface and the Save As button* to create new credentials from sample credentials. This will prevent you from overwriting the sample credential(s).

To create an SSH/Key credential to monitor Docker containers:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the example **Docker Basic - SSH** credential, click its **[Actions]** icon (⋯) and select ***Duplicate***. A copy of the credential, called **Docker Basic - SSH copy** appears.
3. Click the **[Actions]** icon (⋯) for the **Docker Basic - SSH copy** credential and select ***Edit***. The **Edit Credential** modal page appears:



- ***Name***. Type a new name for the Docker credential.
- ***All Organizations***. Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the ***What organization manages this service?*** drop-down field to align the credential with those specific organizations.
- ***Timeout(ms)***. Type "10000".
- ***Hostname/IP***. Type "%D".
- ***Port***. Type the SSH port number for the Docker swarm you want to monitor.
- ***Username***. Type the username for a user with SSH access to the Docker swarm command line interface.
- ***Password***. Type the user's password.
- ***PEM Format***. Keep this field blank.

4. Click the **[Save & Close]** button.

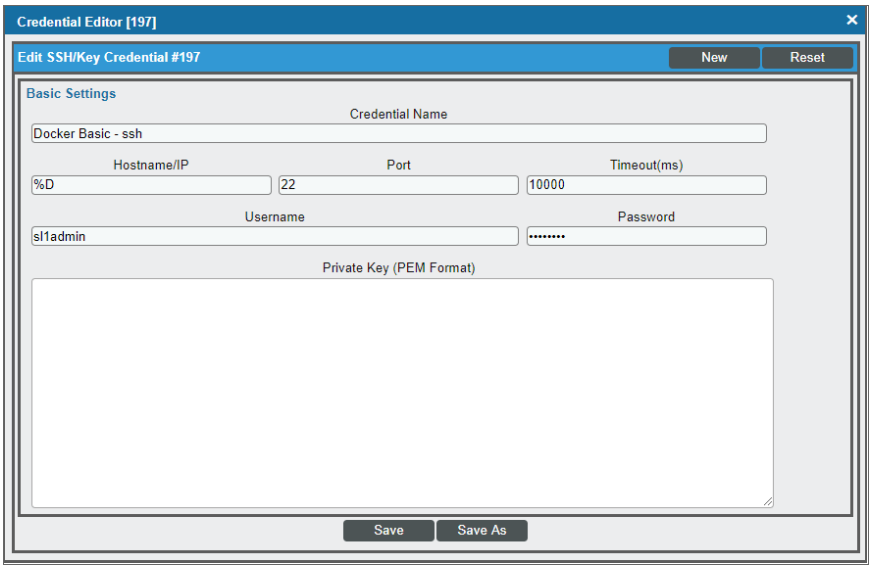
Creating an SSH/Key Credential in the SL1 Classic User Interface

If you are using SSH to monitor Docker swarms, then you must create an SSH/Key credential that allows the Dynamic Applications in the *Docker* PowerPack to connect with Docker swarms. An example SSH/Key credential that you can edit for your own use is included in the *Docker* PowerPack.

> **NOTE:** You can also use an SSH credential in conjunction with the *Kubernetes* PowerPack to monitor the Docker infrastructure for a Kubernetes cluster.

To create an SSH/Key credential to monitor Docker containers:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the example **Docker Basic - SSH** credential, and then click its wrench icon (). The **Edit SSH/Key Credential** modal page appears.
3. Complete the following fields:



- ***Credential Name***. Type a new name for the Docker credential.
- ***Hostname/IP***. Type "%D".
- ***Port***. Type the SSH port number for the Docker swarm you want to monitor.
- ***Timeout(ms)***. Type "10000".

- ***Username***. Type the username for a user with SSH access to the Docker swarm command line interface.
- ***Password***. Type the user's password.
- ***Private Key (PEM Format)***. Keep this field blank.

4. Click the **[Save As]** button.
5. When the confirmation message appears, click **[OK]**.

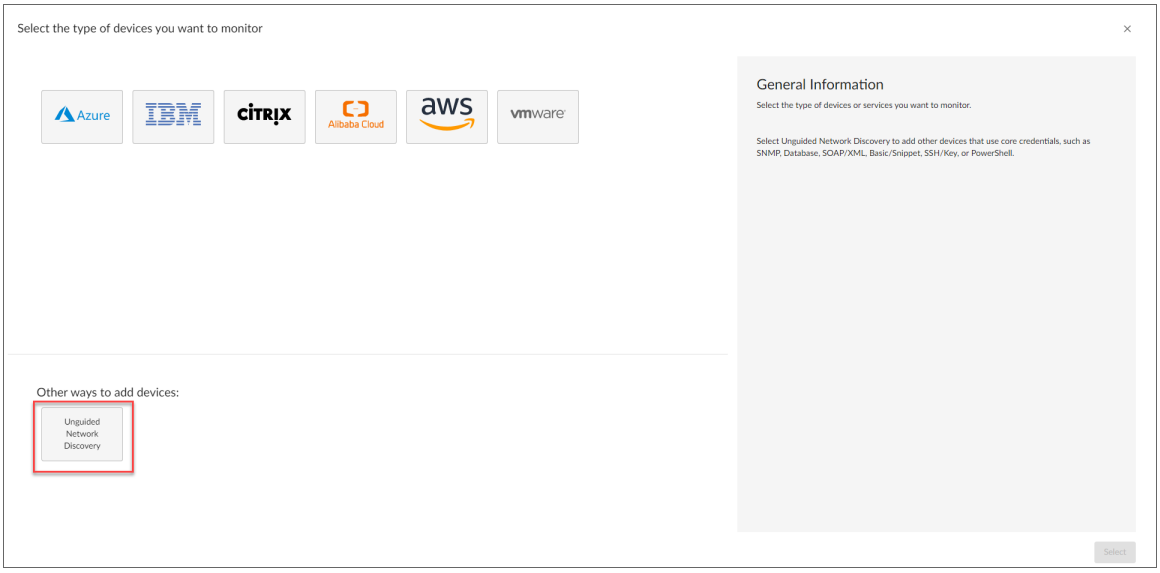
Discovering Docker Components

To discover and model your Docker component devices for monitoring, you must run a discovery session. The discovery session will discover the Docker hosts and swarms that SL1 will use as the root devices for monitoring the Docker components.

Several minutes after the discovery session has completed, the Dynamic Applications in the *Docker* PowerPack will automatically align to the Docker root devices. These Dynamic Applications will discover, model, and monitor the remaining components in your Docker system.

To discover Docker components, perform the following steps:

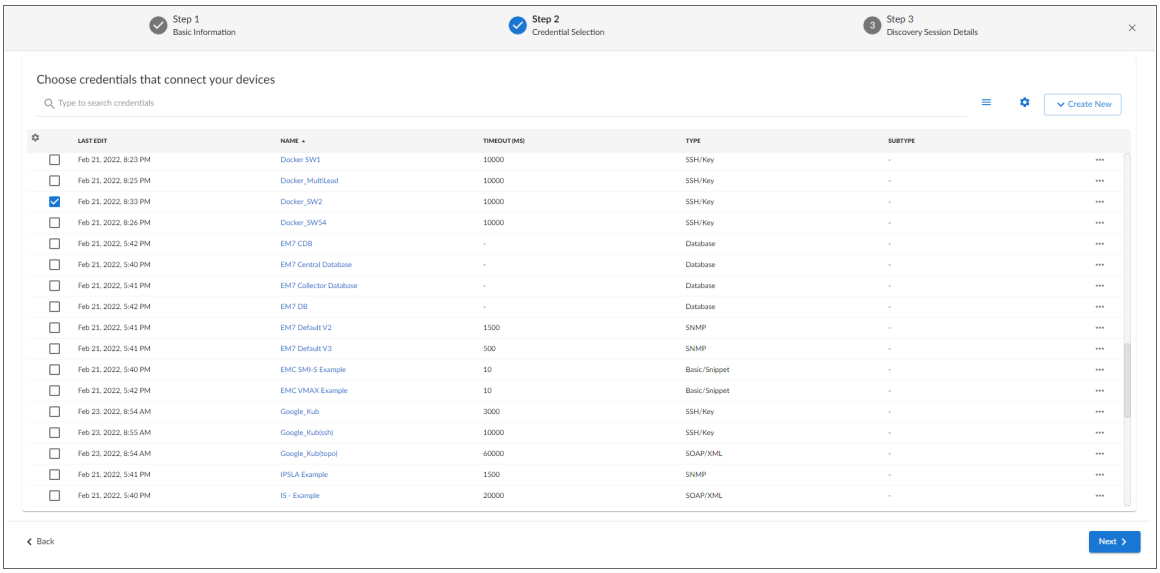
1. On the **Devices** page ( ) or the **Discovery Sessions** page (Devices > Discovery Sessions), click the **[Add Devices]** button. The **Select** page appears:



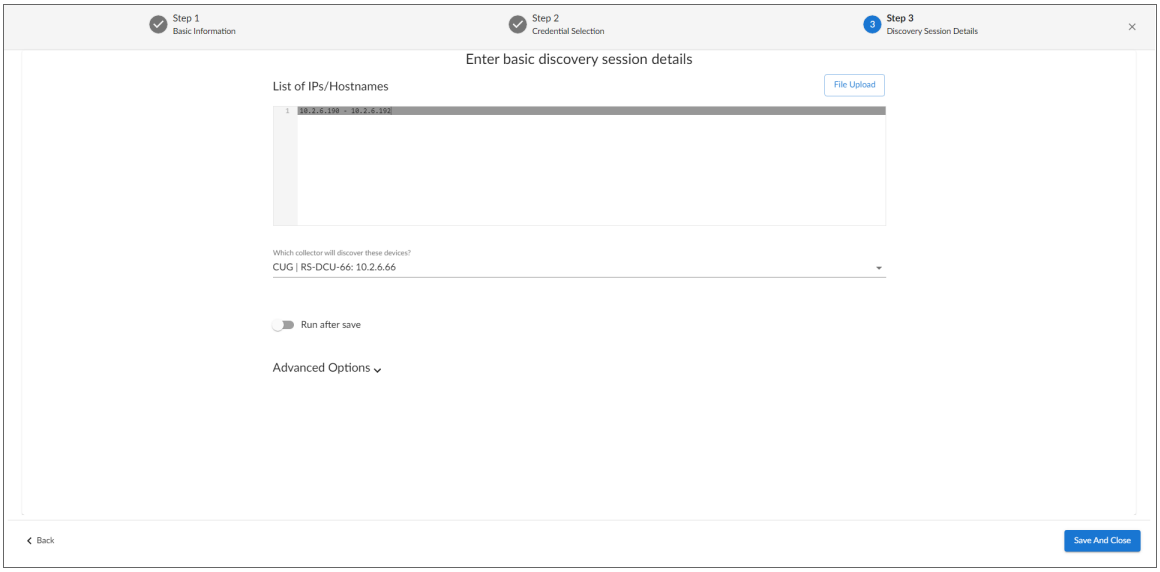
2. Click the **[Unguided Network Discovery]** button. Additional information about the requirements for discovery appears in the **General Information** pane to the right.
3. Click **[Select]**. The **Add Devices** page appears.
4. Complete the following fields:

- ***Name***. Type a unique name for this discovery session. This name is displayed in the list of discovery sessions on the **[Discovery Sessions]** tab.
- ***Description***. Optional. Type a short description of the discovery session. You can use the text in this description to search for the discovery session on the **[Discovery Sessions]** tab.
- ***Select the organization to add discovered devices to***. Select the name of the organization to which you want to add the discovered devices

5. Click **[Next]**. The **Credentials** page of the **Add Devices** wizard appears:



6. On the **Credentials** page, locate and select the ***Basic/Snippet*** or ***SSH/Key*** credential you created for Docker.
7. Click **[Next]**. The **Discovery Session Details** page of the **Add Devices** wizard appears:



8. Complete the following fields:
    - ***List of IPs/Hostnames***. Type the IP addresses for all of the Docker hosts in the swarm that you want to discover.

NOTE: Swarms are created only when the swarm leader is discovered. To monitor a Docker Swarm, you must discover all nodes included in the cluster by SSH connections.

NOTE: You must have both Docker Swarms and Docker Hosts (Managers and Workers) discovered on the same Data Collector on which the Docker Swarm Leader is discovered to keep the cache on the Docker Swarm device. If there is maintenance or a failure on the Data Collector that is hosting the Swarm cluster and Docker Hosts, a run book action will move all the Hosts and the Swarm cluster to the same Data Collector that the Leader has been moved to. If a device is moved to a different Data Collector, the same run book action will keep the Host with the Leader. Data gaps in collection may appear during this process.

    - ***Which collector will monitor these devices?***. Required. Select an existing collector to monitor the discovered devices.
    - ***Run after save***. Select this option to run this discovery session as soon as you save the session.

      In the **Advanced options** section, click the down arrow icon ( ) to complete the following fields:

        - ***Discover Non-SNMP***. Enable this setting.
        - ***Model Devices***. Enable this setting.
9. Click **[Save and Run]** if you enabled the Run after save setting, or **[Save and Close]** to save the discovery session. The **Discovery Sessions** page (Devices > Discovery Sessions) displays the new discovery session.
10. If you selected the **Run after save** option on this page, the discovery session runs, and the **Discovery Logs** page displays any relevant log messages. If the discovery session locates and adds any devices, the **Discovery Logs** page includes a link to the **Device Investigator** page for the discovered device.

Discovering Docker Components in the SL1 Classic User Interface

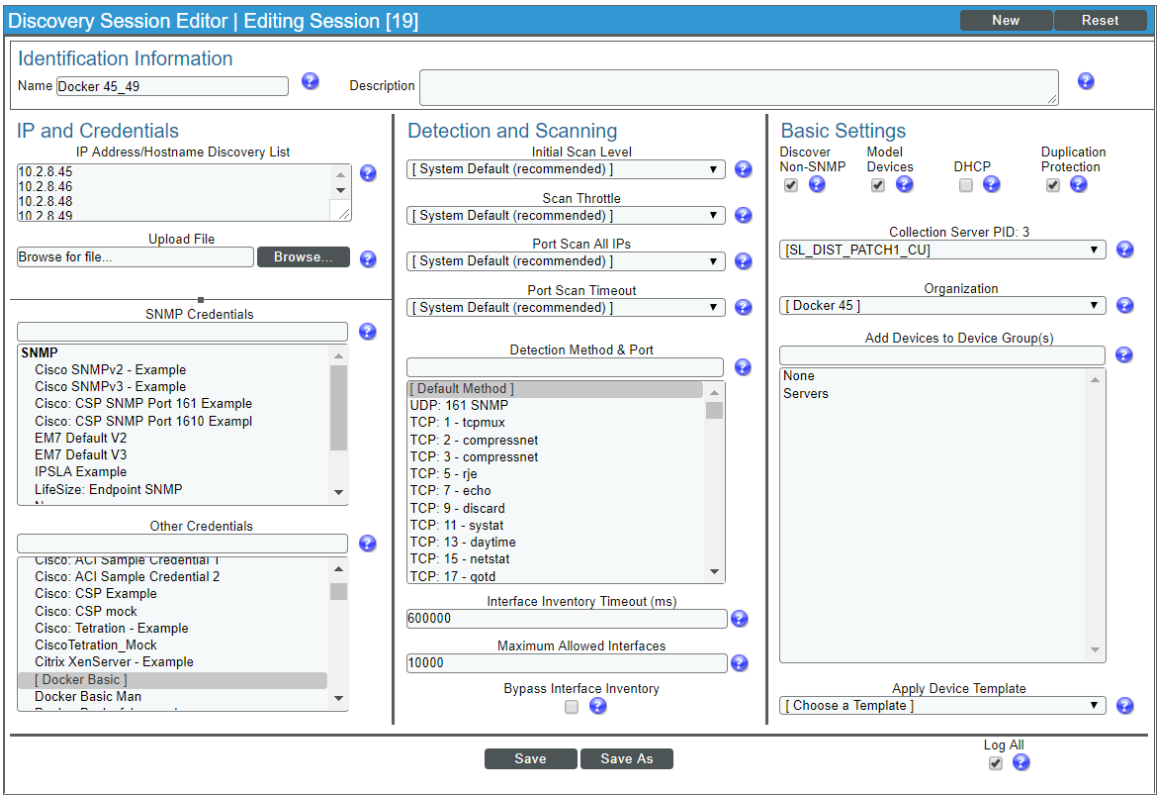
To discover and model your Docker component devices for monitoring, you must run a discovery session. The discovery session will discover the Docker hosts and swarms that SL1 will use as the root devices for monitoring the Docker components.

Several minutes after the discovery session has completed, the Dynamic Applications in the *Docker* PowerPack will automatically align to the Docker root devices. These Dynamic Applications will discover, model, and monitor the remaining components in your Docker system.

To discover Docker components, perform the following steps:

1. Go to the **Discovery Control Panel** page (System > Manage > Classic Discovery), and then click the **[Create]** button. The **Discovery Session Editor** page appears.

2. In the **Discovery Session Editor** page, complete the following fields:



- ***Name***. Type a name for your discovery session.
- ***IP Address/Hostname Discovery List***. Type the IP addresses for all of the Docker hosts in the swarm that you want to discover.

NOTE: Swarms are created only when the swarm leader is discovered. To monitor a Docker Swarm, you must discover all nodes included in the cluster by SSH connections.

NOTE: You must have both Docker Swarms and Docker Hosts (Managers and Workers) discovered on the same Data Collector on which the Docker Swarm Leader is discovered to keep the cache on the Docker Swarm device. If there is maintenance or a failure on the Data Collector that is hosting the Swarm cluster and Docker Hosts, a run book action will move all the Hosts and the Swarm cluster to the same Data Collector that the Leader has been moved to. If a device is moved to a different Data Collector, the same run book action will keep the Host with the Leader. Data gaps in collection may appear during this process.

- ***Other Credentials***. Select the *Basic/Snippet or SSH/Key credential(s)* you created for Docker.
- ***Discover Non-SNMP***. Select this checkbox.

- ***Model Devices***. Select this checkbox.

3. Optionally, you can enter values in the other fields on this page. For more information about the other fields on this page, see the ***Discovery & Credentials*** manual.
4. Click the **[Save]** button to save the discovery session, and then close the **Discovery Session Editor** window.
5. The discovery session you created displays at the top of the **Discovery Control Panel** page. Click its lightning-bolt icon ( ) to run the discovery session.
6. The **Discovery Session** window appears. When a root device is discovered, click its device icon ( ) to view the **Device Properties** page for that device.

Manually Aligning Dynamic Applications

To verify that SL1 has automatically aligned the correct Dynamic Applications during discovery:

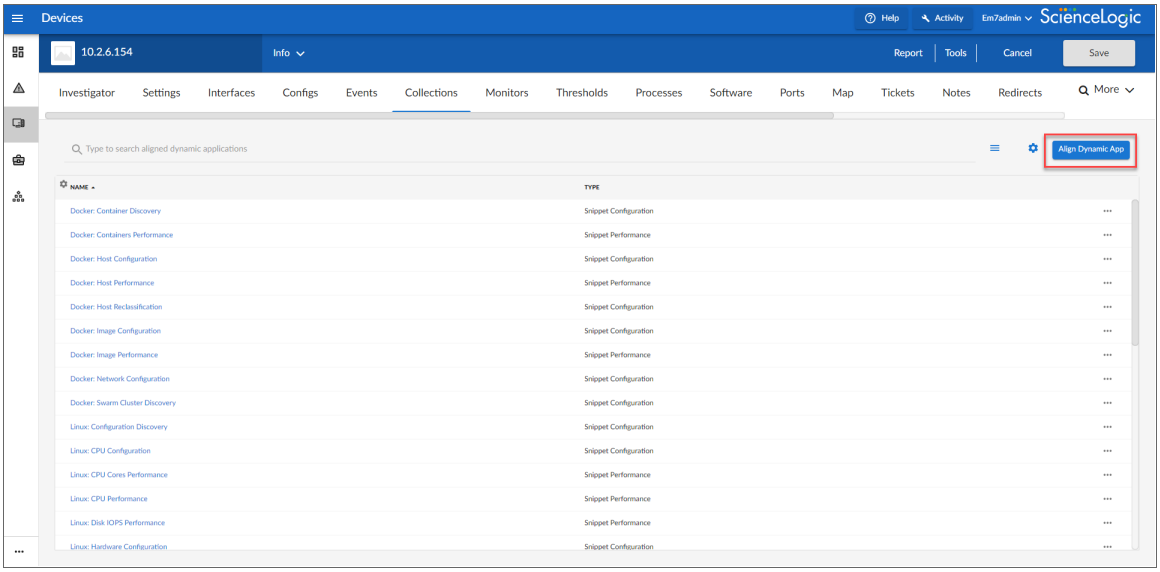
1. From the **Device Investigator** page for the Docker root device, click the **[Collections]** tab.
2. The following Dynamic Applications should appear in the list of aligned Dynamic Applications:

    - For Docker Hosts:
        - Docker: Container Discovery
        - Docker: Containers Performance
        - Docker: Host Configuration
        - Docker: Host Performance
        - Docker: Host Reclassification
        - Docker: Image Configuration
        - Docker: Image Performance
        - Docker: Network Configuration
        - Docker: Swarm Cluster Discovery
    - For Docker Swarms:
        - Docker: Stack Discovery
        - Docker: Swarm Configuration
        - Docker: Swarm Performance
        - Docker: Swarm Service Discovery

> **NOTE:** It can take several minutes after discovery for Dynamic Applications to display on the **Dynamic Application Collections** page. If the listed Dynamic Applications do not display on this page, try clicking the **[Reset]** button.

If the Dynamic Applications have not been automatically aligned, you can align them manually. To do so, perform the following steps:

1. Go to the **Device Investigator** page for the Docker root device and click the **[Collections]** tab.
2. Click the **[Edit]** button and then click the **[Align Dynamic App]** button.On the **Dynamic Application Collections** page, click the **[Action]** button and then select *Add Dynamic Application* from the menu. The **Dynamic Application Alignment** page appears.

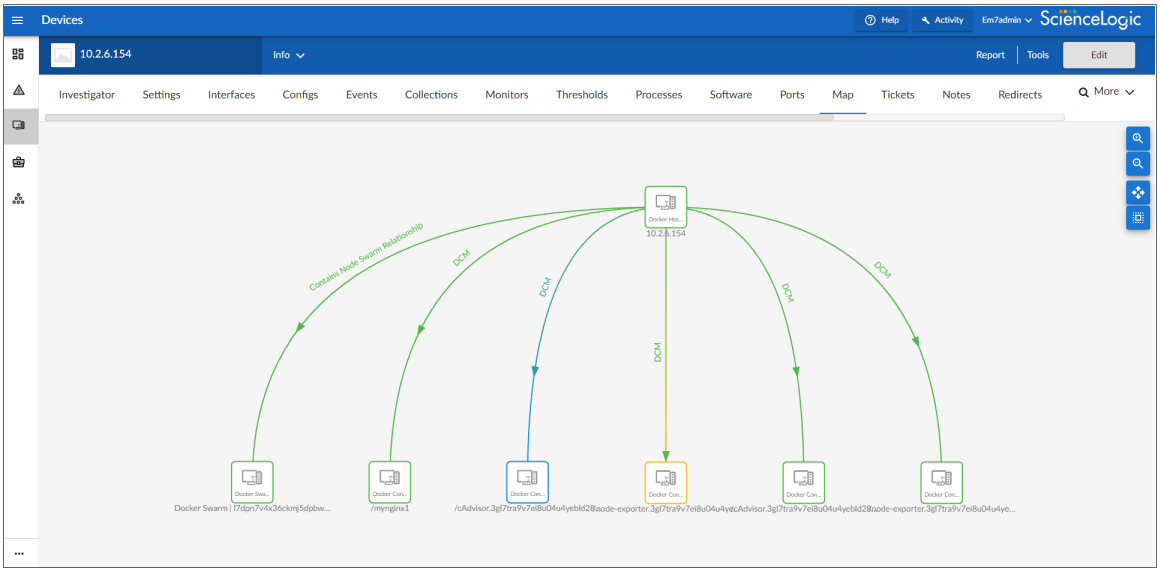


3. In the **Align Dynamic Application** window, click *Choose Dynamic Application*.
4. In the **Choose Dynamic Application** window, select the Dynamic Application you want to align and click **[Select]**. The name of the selected Dynamic Application appears in the **Align Dynamic Application** window.
5. Uncheck the box next to ***Use Device SNMP Credential*** and click *Choose Credential*. The **Choose Credential** window appears.
6. Select the ***Basic/Snippet credential*** you created for Docker and then click the **[Select]** button. The name of the Docker credential you selected appears in the **Align Dynamic Application** window.
7. Click the **[Align Dynamic App]** button. When the Dynamic Application is successfully aligned, it is added to the Collections tab, and a confirmation message appears at the bottom of the tab.
8. Repeat steps 2-7 as needed to align any additional Dynamic Applications.

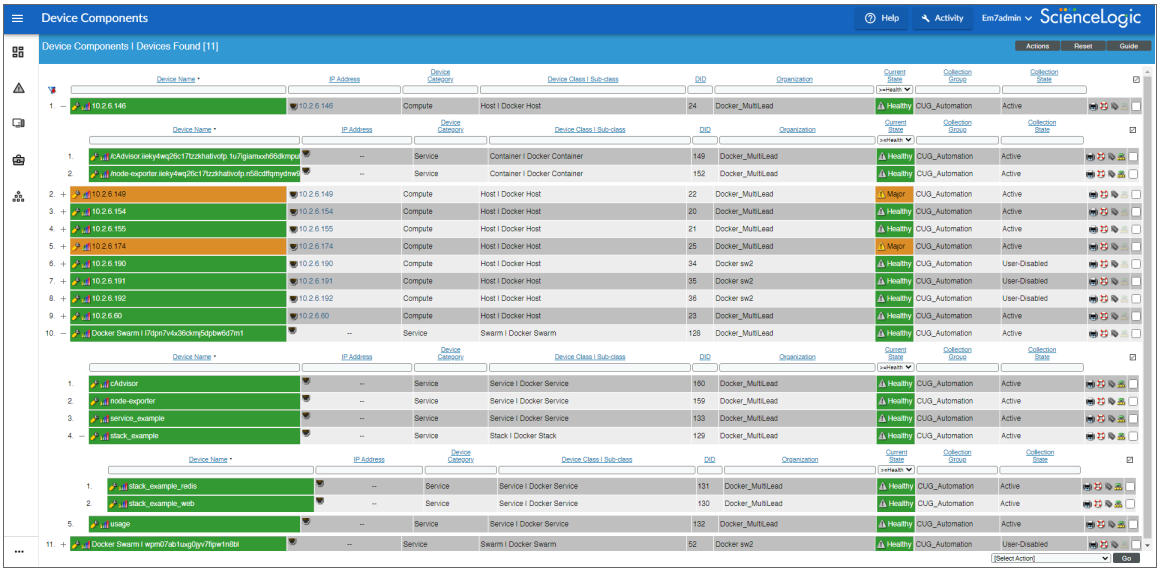
Viewing Docker Component Devices

In addition to the **Devices** page, you can view the Docker platform and all of its component devices in the following places in the user interface:

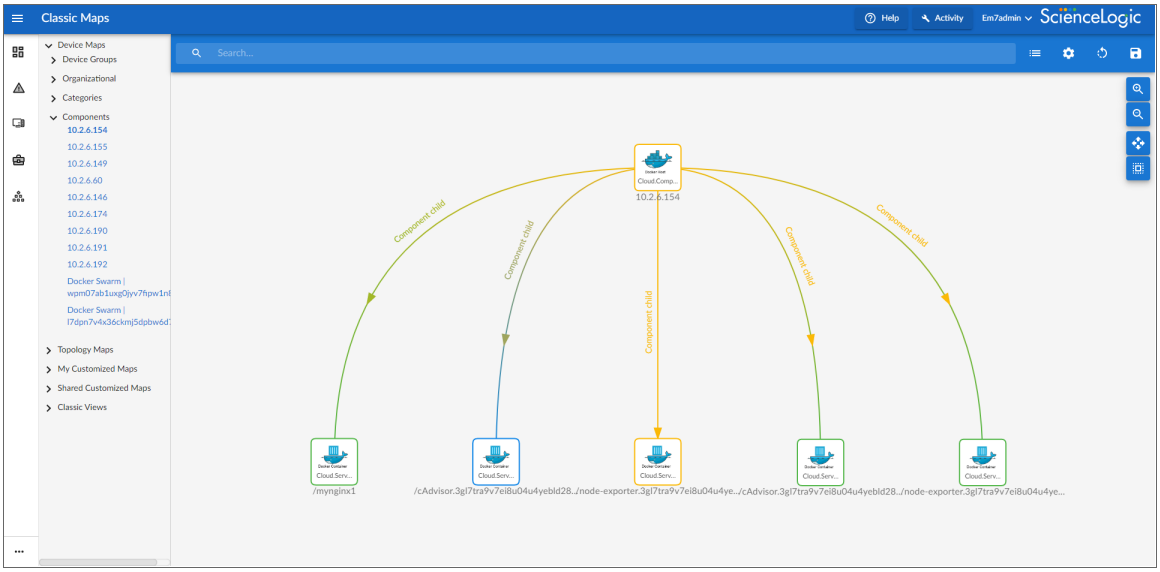
- The **Device Investigator** Map page (click **Map** in the **Device Investigator** page) displays a map of a particular device and all of the devices with which it has parent-child relationships. Double-clicking any of the listed devices reloads the page to make the selected device the primary device.



- The **Device Components** page (Registry > Devices > Device Components) displays a list of all root devices and component devices discovered by SL1 in an indented view, so you can easily view the hierarchy and relationships between child devices, parent devices, and root devices. To view the component devices associated with Docker, find the Docker Host or Docker Swarm device and click its plus icon (+):



- The **Component Map** page (Classic Maps > Device Maps > Components) allows you to view devices by root node and view the relationships between root nodes, parent components, and child components in a map. This makes it easy to visualize and manage root nodes and their components. SL1 automatically updates the **Component Map** as new component devices are discovered. The platform also updates each map with the latest status and event information. To view the map for a Docker device, go to the **Component Map** page and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the ***Maps*** manual.



Relationships Between Component Devices

In addition to parent/child relationships between component devices, SL1 also creates relationships between the following component devices:

- Swarms and Nodes
- Services and Containers

You can also use the *Docker* PowerPack in conjunction with the *Kubernetes* PowerPack when monitoring Kubernetes systems. When you do so, SL1 creates relationships between Docker Swarms and Containers and their underlying Kubernetes Nodes.

Chapter 3

Docker Dashboards in the SL1 Classic User Interface

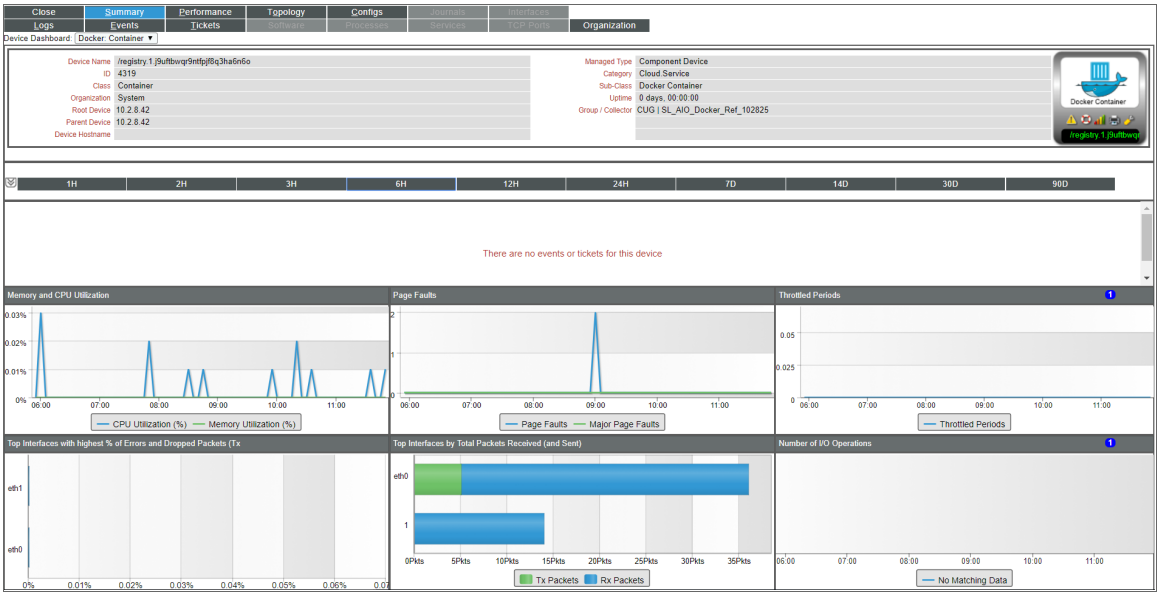
Overview

The following sections describe the device dashboards that are included in the *Docker* PowerPack and can be accessed in the SL1 classic user interface:

Device Dashboards

The *Docker* PowerPack includes device dashboards that provide summary information for Docker component devices. Each of the device dashboards in the *Docker* PowerPack is set as the default device dashboard for the equivalent device class.

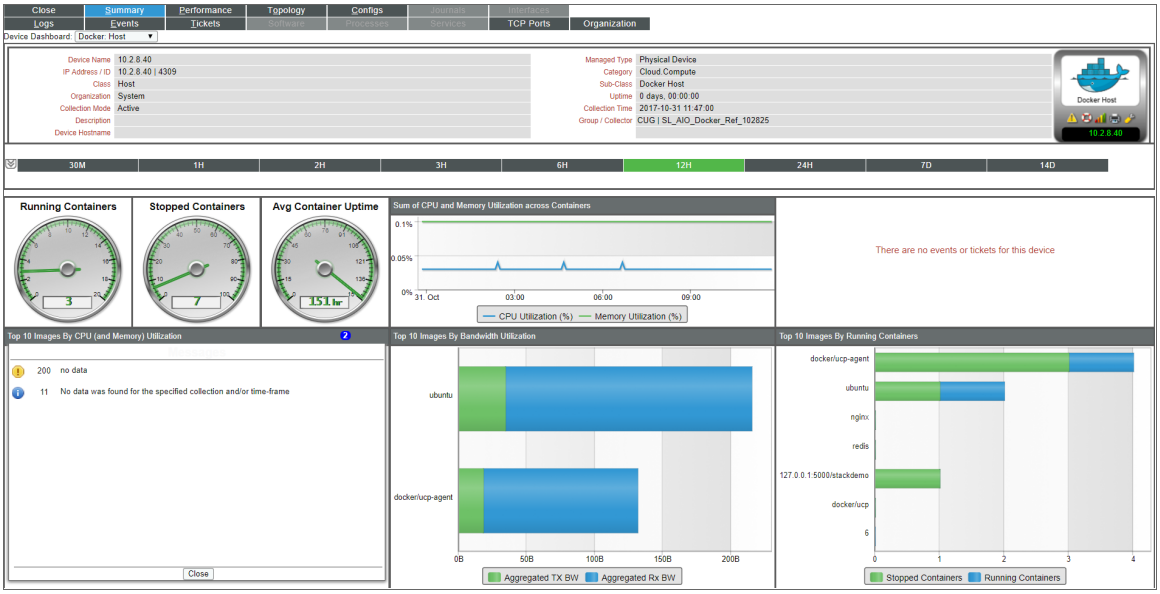
Docker: Container



The Docker: Container device dashboard displays the following information:

- Events and tickets for the device
- Memory and CPU utilization over a specified period of time
- Page faults over a specified period of time
- Throttled periods over a specified period of time
- Top interfaces with the highest percentage of errors and dropped packets over a specified period of time
- Top interfaces by total packets received over a specified period of time
- Number of input and output operations over a specified period of time

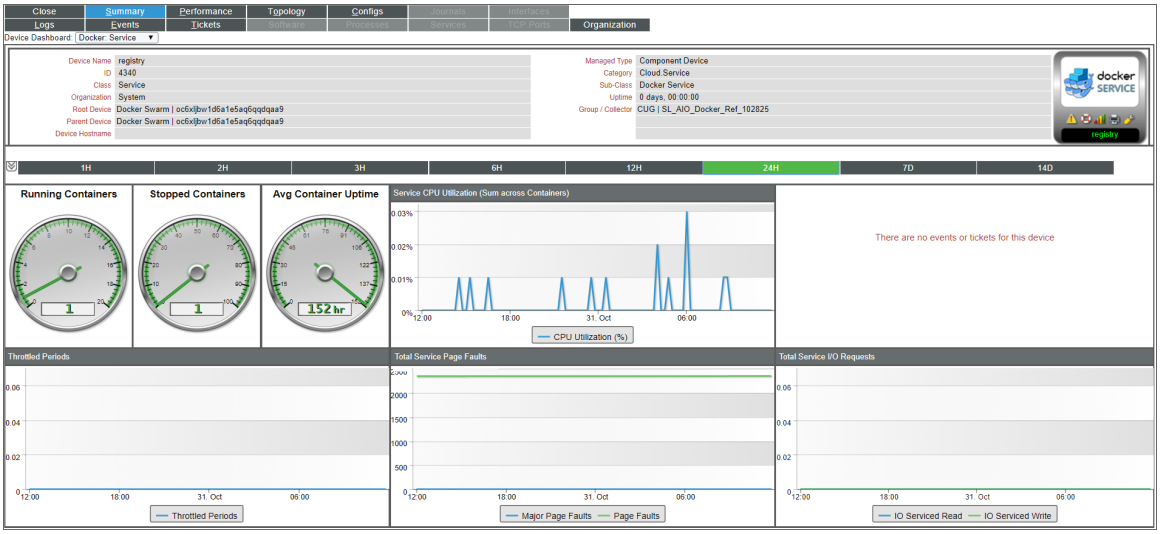
Docker: Host



The Docker: Host device dashboard displays the following information:

- Containers running and stopped over a specified period of time, as well as average container uptime
- Total CPU utilization and memory across all containers over a specified period of time
- Events and tickets for the device
- Top 10 images by CPU and memory utilization over a specified period of time
- Top 10 images by bandwidth utilization over a specified period of time
- Top 10 images by running containers over a specified period of time

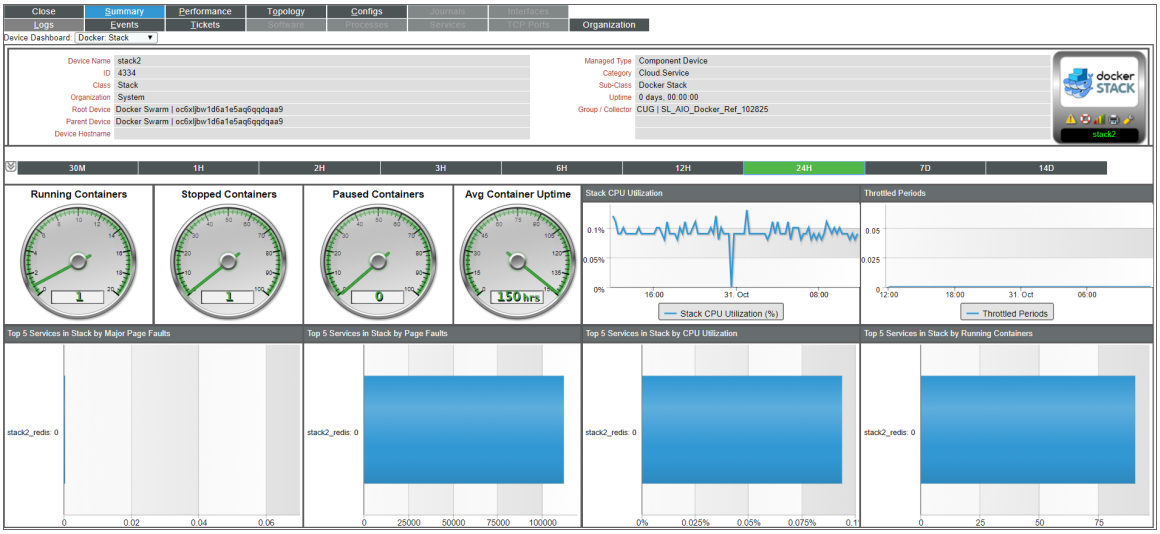
Docker: Service



The Docker: Service device dashboard displays the following information:

- Containers running and stopped over a specified period of time, as well as average container uptime
- Service CPU utilization over a specified period of time
- Events and tickets for the device
- Throttled periods over a specified period of time
- Top service page faults over a specified period of time
- Total service input and output requests over a specified period of time

Docker: Stack



The Docker: Stack device dashboard displays the following information:

- Containers running, stopped, and paused over a specified period of time, as well as average container uptime
- Stack CPU utilization over a specified period of time
- Throttled periods over a specified period of time
- Top 5 services in the stack over a specified period of time, based on the number of major page faults
- Top 5 services in the stack over a specified period of time, based on the number of page faults
- Top 5 services in the stack over a specified period of time, based on CPU utilization
- Top 5 services in the stack over a specified period of time, based on the number of running containers

Docker: Swarm

The Docker: Swarm device dashboard displays the following information:

- Containers running, stopped, and paused over a specified period of time, as well as average container uptime
- Number of stacks, services, images, total nodes, and available nodes over a specified period of time
- Top 5 stacks in the swarm over a specified period of time, based on CPU utilization
- Top 5 services in the swarm over a specified period of time, based on CPU utilization
- Top 5 services in the swarm over a specified period of time, based on the number of major page faults and total page faults
- Overall swarm CPU utilization
- Overall swarm memory utilization
- Overall swarm input and output

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@sciencelogic.com

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010