Monitoring Kubernetes

Kubernetes PowerPack version 105

Table of Contents

Chapter 1

Introduction

Overview

This manual describes how to monitor Kubernetes clusters in SL1 using the *Kubernetes* PowerPack.

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon (☰).
- To view a page containing all of the menu options, click the Advanced menu icon (⋯).

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

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

This chapter covers the following topics:

> **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 is Kubernetes?

Kubernetes is an open-source platform that automates the deployment, scaling, and operation of application containers. The Kubernetes platform is deployed in ***clusters*** that consist of compute nodes. These nodes can take on the following roles:

- ***Master***. The master runs on one of the physical computers in the cluster and manages the cluster. It oversees all cluster activities such as scheduling, maintaining, and scaling applications, as well as executing updates.
- ***Nodes***. Nodes are physical computers or virtual machines (VMs) that run applications and perform other tasks in a Kubernetes cluster. Nodes are controlled by the master.

Kubernetes manages containers through a series of objects that represent your system, including ***Pods***, ***Services***, ***Volumes***, and ***Namespaces***. Kubernetes also uses a series of Controller objects that provide additional features and functionality; these include ***ReplicaSets***, ***Deployments***, ***StatefulSets***, ***DaemonSets***, ***Jobs***, ***CronJobs***, and ***IngressControllers***.

NOTE: For more information about these Kubernetes concepts, consult the Kubernetes documentation.

What Does the Kubernetes PowerPack Monitor?

The *Kubernetes* PowerPack enables you to monitor Kubernetes clusters, nodes, namespaces and controllers.

NOTE: The *Kubernetes* PowerPack can leverage the capabilities of the *Linux Base Pack* PowerPack to provide a comprehensive view of the Kubernetes cluster nodes, including their underlying hardware. If you want to do this, you must install and run the most recent version of this PowerPack, create an SSH credential, and include the Credential ID in the Kubernetes credential. For more information about using this PowerPack, see the ***Monitoring Linux*** manual.

NOTE: The *Kubernetes* PowerPack has been validated on the Cloud Native Computing Foundation (CNCF) version of Kubernetes.

The *Kubernetes* PowerPack includes the following features:

- Dynamic Applications that perform the following tasks:
  - Discover and monitor the Kubernetes cluster, nodes, namespaces, and controllers
  - Collect and present data about the underlying Linux operating system of the cluster nodes (Only if an SSH Credential ID is included in the Kubernetes credential). For more information, see the ***Monitoring Linux*** manual.

- Device Classes for each of the Kubernetes devices the *Kubernetes* PowerPack models
- Event Policies and corresponding alerts that are triggered when Kubernetes devices meet certain status criteria
- Guided Discovery and a Universal Credential to discover Kubernetes Cluster devices
- Run Book Action and Automation policies do the following:
    - Align Dynamic Applications from the *Linux Base Pack* PowerPack to Kubernetes nodes and report back to the ScienceLogic Data Collector or All-in-One Appliance if the Dynamic Applications were successfully aligned
    - Ensure that Namespaces (and their children) have a 1-hour vanishing timer, to properly reflect topology changes

Installing the Kubernetes PowerPack

Before completing the steps in this manual, you must import and install the latest version of the *Kubernetes* 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 section on ***Global Settings***.

To download and install the PowerPack:

1. Search for and download the PowerPack from the **PowerPacks** page (Product Downloads > PowerPacks & SyncPacks) at the ScienceLogic Support Site.
2. In SL1, go to the **PowerPacks** page (System > Manage > PowerPacks).
3. Click the **[Actions]** button and choose *Import PowerPack*. The **Import PowerPack** dialog box appears.
4. Click **[Browse]** and navigate to the PowerPack file from step 1.
5. Select the PowerPack file and click **[Import]**. The **PowerPack Installer** modal displays a list of the PowerPack contents.
6. Click **[Install]**. The PowerPack is added to the **PowerPacks** page.

> **NOTE:** If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPacks** 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

Use the following menu options to navigate the SL1 user interface:

- To view a pop-out list of menu options, click the menu icon (☰).
- To view a page containing all of the menu options, click the Advanced menu icon (•••).

The following sections describe how to configure and discover Kubernetes clusters for monitoring by SL1 using the *Kubernetes* PowerPack:

This chapter covers the following topics:

Prerequisites for Monitoring Kubernetes Clusters

Before you can monitor Kubernetes clusters using the *Kubernetes* PowerPack, you must first meet the following requirements:

Required:

- ***Authentication Token***: Create a Service Account Token that SL1 will use to authenticate with the Kubernetes API. This Service Account Token must have the minimum permissions set in the ***Required Permissions for the Service Account Token*** section.
- IP or Endpoint of the Kubernetes API
- Kubernetes API Port
- Installed Metrics-Server in the cluster. See https://kubernetes-sigs.github.io/metrics-server/ for more information.

Optional

These requirements must be met if you want to monitor the underlying Linux OS of the nodes using the Linux Base Pack:

- Import and install the Linux Base Pack PowerPack version 111 or later.
- Configure SSH credentials (username/password or username/private-key) on the Kubernetes cluster nodes. These credentials must be the same on all nodes.
- Create a SSH Credential in SL1 (username/password or username/private-key). Once the credential is created, copy the Credential ID, which will be used during Guided Discovery. See Creating A SSH Credential (Optional) for more information.

Required Permissions for the Service Account Token

To create a token with the minimum permissions, copy the code below in a .yaml file with the name `slmonitor-config.yaml` and apply it. It will create a ClusterRole, ServiceAccount, bind the ClusterRole and the ServiceAccount, and create a secret with an authentication token.

To apply the file using kubectl use the command:

```
kubectl apply -f slmonitor-config.yaml
```

To get the authentication token use the command:

```
kubectl describe secret slmonitor-readonly-sa
```

Config File:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: slmonitor-readonly-clusterrole
rules:
- apiGroups:
  - ""
  - apps
  - batch
  - metrics.k8s.io
  - networking.k8s.io
  - autoscaling
  resources:
  - nodes
  - pods
  - componentstatuses
  - namespaces
  - persistentvolumes
  - events
  - replicationcontrollers
  - services
  - deployments
  - statefulsets
  - replicasets
  - daemonsets
  - cronjobs
  - jobs
  - ingresses
  - horizontalpodautoscalers
  verbs:
  - get
  - list
  - watch
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: slmonitor-readonly-sa
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: slmonitor-readonly-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: slmonitor-readonly-clusterrole
subjects:
- kind: ServiceAccount
  name: slmonitor-readonly-sa
  namespace: default
---
apiVersion: v1
kind: Secret
metadata:
  name: slmonitor-readonly-sa
  annotations:
    kubernetes.io/service-account.name: slmonitor-readonly-sa
type: kubernetes.io/service-account-token
```

Creating Credentials for Kubernetes Clusters

Creating a Kubernetes Credential

To define a Kubernetes credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Click on the **[Create New]** button and select *Create Kubernetes Credential*.
3. Supply values in the following fields:
    - ***Name***. Name of the credential. Can be any combination of alphanumeric characters, up to 64 characters.
    - ***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)***. Time, in milliseconds, after which SL1 will stop trying to communicate with the device from which you want to retrieve data.
    - ***Kubernetes IP/Hostname***. Enter the Kubernetes API IP, hostname, or endpoint.
    - ***Port***. Enter the Kubernetes API port.
    - ***Kubernetes Service Account Token***. Enter the Kubernetes Service Account Token created previously, which is used to authenticate in the Kubernetes API.
    - ***Enable Linux Monitoring***. (Optional) If toggled on, will enable you to monitor and collect data about the underlying Linux operating systems of the cluster nodes. If enabled, enter the *ID of an SSH credential* in the field that appears below.
4. Click **[Save & Test]**.

Creating an SSH Credential (Optional)

You only need to create an SSH credential if you plan to enable Linux Monitoring using the Linux Base Pack PowerPack. To define an SSH Credential:

1. Go to the **Credentials** page (Manage > Credentials).
2. Click on the **[Create New]** button and select *Create SSH/Key Credential*.
3. Supply values in the following fields:
    - ***Name***. Name of the credential. Can be any combination of alphanumeric characters, up to 64 characters.
    - ***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)***. Time, in milliseconds, after which SL1 will stop trying to communicate with the device from which you want to retrieve data.
    - ***Hostname/IP***. Type *%D*.
    - ***Username***. Type the SSH account username. This will be used to connect to the nodes.
    - ***Password***. Type the password for the SSH account.
    - ***Private Key (PEM Format)***. Type the SSH private key.

NOTE: The available combinations for authentication are Username/Password, Username/Private Key, or Username/Private Key/Password.

4. Click **[Save & Test]**.

NOTE: After Creating the SSH Credential, you need to enter the ***Credential ID*** in the ***Enable Linux Monitoring*** option in the Kubernetes Credential.

Discovering a Kubernetes Cluster

To create and run a discovery session that will discover a Kubernetes Cluster, 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 on the **[Kubernetes]** button.
3. Select the Kubernetes credential created previously or click the **[Create New]** button to ***define a new Kubernetes credential***.
4. Click on the **[Next]** button.
5. Complete the required fields:
   - ***Root Device Name***. The virtual device name that will be created with this Guided Discovery.
   - ***Organization***. Select the organization where you want to discover the virtual device.
   - ***Collector Group Name***. Select the collector group for the virtual device.
6. Click on the **[Next]** button.
7. Once the discovery process finishes, click on the **[Close]** button. Guided Discovery will create a virtual device and align the "Kubernetes: Cluster Discovery" Dynamic Application with the credential to start monitoring the Kubernetes Cluster.

Customizing Event Policies

The "Kubernetes: Event Configuration" Dynamic Application is a Journal Dynamic Application that collects the events reported by Kubernetes. The "Kubernetes: Normal event" and "Kubernetes: Warning event" are general event policies that are enabled by default.

Users can enable more specific event policies in the PowerPack after disabling the ""Kubernetes: Normal event" and "Kubernetes: Warning event" policies. To enable these event policies, perform the following steps:

1. Go to the **Event Policy Manager** page (Registry > Events > Event Manager).
2. Search for the "Kubernetes: Normal event" in the ***Event Policy Name*** field.

3. Select the wrench icon (🔧) for the event policy to open the **Event Policy Editor** page.
4. In the ***Operational State*** drop-down, select *Disabled* and then click the **[Save]** button. Repeat these steps for the "Kubernetes: Warning event" event policy.
5. Once these event policies are disabled, find the event policies you want to use and enable them. You can enable more than one event policy at a time by selecting their checkboxes in the **Event Policy Manager** page, selecting *ENABLE these event policies* in the ***Select Action*** menu, and then clicking the **[Go]** button.

The following event policies are available:

Event Policy	Device	Severity	State
Kubernetes: Normal event	Any	Notice	Disabled
Kubernetes: Warning event	Any	Major	Enabled
Kubernetes: Network Failure	Any	Critical	Enabled
Kubernetes: Error Image Never Pull	Pod	Major	Enabled
Kubernetes: Failed to Create Pod Container	Pod	Major	Enabled
Kubernetes: Failed to Kill Pod	Pod	Major	Enabled
Kubernetes: Failed Start Hook	Pod	Major	Enabled
Kubernetes: Failed Sync	Pod	Major	Enabled
Kubernetes: Failed Validation	Pod	Major	Enabled
Kubernetes: Free Disk Space Failed	Pod	Major	Enabled
Kubernetes: Image Pull Backoff	Pod	Major	Enabled
Kubernetes: Pod Container Created	Pod	Notice	Disabled
Kubernetes: Pod Exceeded Grace Period	Pod	Major	Enabled
Kubernetes: Pod Failed	Pod	Notice	Enabled

Event Policy	Device	Severity	State
Kubernetes: Pod Image Inspect Failed	Pod	Notice	Enabled
Kubernetes: Pod Image Pulled	Pod	Notice	Disabled
Kubernetes: Pod Image Pulling	Pod	Major	Disabled
Kubernetes: Pod Killing	Pod	Major	Enabled
Kubernetes: Pod Network Not Ready	Pod	Major	Enabled
Kubernetes: Pod Preempting	Pod	Major	Enabled
Kubernetes: Pod Started	Pod	Notice	Disabled
Kubernetes: Pod Unhealthy	Pod	Major	Enabled
Kubernetes: Prestop Hook	Pod	Major	Enabled
Kubernetes: Probe Warning	Pod	Major	Enabled
Kubernetes: Already Mounted Volume	Node	Notice	Enabled
Kubernetes: Container GC Failed	Node	Major	Disabled
Kubernetes: Failed Attach Volume	Node	Critical	Enabled
Kubernetes: Failed Create Pod Sandbox	Node	Notice	Enabled
Kubernetes: Failed Map Volume	Node	Major	Enabled
Kubernetes: Failed Mount	Node	Major	Enabled
Kubernetes: Failed Node Allocatable Enforcement	Node	Major	Enabled
Kubernetes: Failed Pod Sandbox Status	Node	Notice	Enabled

Event Policy	Device	Severity	State
Kubernetes: File System Resize Failed	Node	Major	Enabled
Kubernetes: File System Resize Successful	Node	Notice	Enabled
Kubernetes: Image GC Failed	Node	Major	Enabled
Kubernetes: Invalid Disk Capacity	Node	Major	Enabled
Kubernetes: Kubelet Setup Failed	Node	Critical	Enabled
Kubernetes: Node Allocatable Enforced	Node	Notice	Enabled
Kubernetes: Node Not Ready	Node	Major	Enabled
Kubernetes: Node Not Schedulable	Node	Major	Enabled
Kubernetes: Node Ready	Node	Notice	Enabled
Kubernetes: Node Schedulable	Node	Notice	Disabled
Kubernetes: Rebooted	Node	Critical	Enabled
Kubernetes: Sandbox Changed	Node	Notice	Enabled
Kubernetes: Starting	Node	Notice	Enabled
Kubernetes: Successful Attach Volume	Node	Notice	Disabled
Kubernetes: Successful Mount Volume	Node	Notice	Disabled
Kubernetes: Volume Resize Failed	Node	Major	Enabled
Kubernetes: Volume Resize Successful	Node	Notice	Enabled
Kubernetes: Node Condition Healthy	Node	Healthy	Enabled

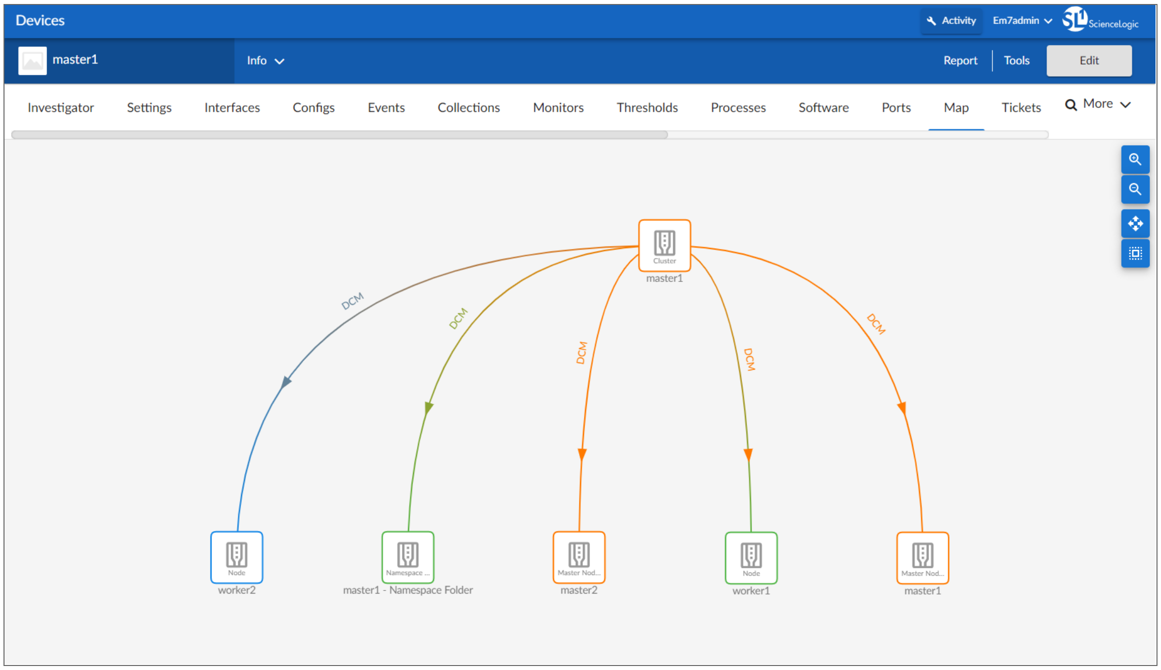
Event Policy	Device	Severity	State
Kubernetes: Node Condition Unhealthy	Node	Major	Enabled
Kubernetes: Node Condition Unknown	Node	Major	Enabled
Kubernetes: Cluster Creation	Pingable	Notice	Enabled
Kubernetes: Component Healthy State	Cluster	Healthy	Enabled
Kubernetes: Component No Healthy State	Cluster	Major	Enabled
Kubernetes: Node Status Changed	Cluster	Notice	Enabled
Kubernetes: Persistent Volume status Healthy	Cluster	Healthy	Enabled
Kubernetes: Persistent Volume Status Unhealthy	Cluster	Major	Enabled
Kubernetes: Restart Count Exceeded Threshold	Deployment/Daemon Set	Minor	Enabled
Kubernetes: Restart Count Returned to Normal	Deployment/Daemon Set	Healthy	Enabled

Viewing Component Devices

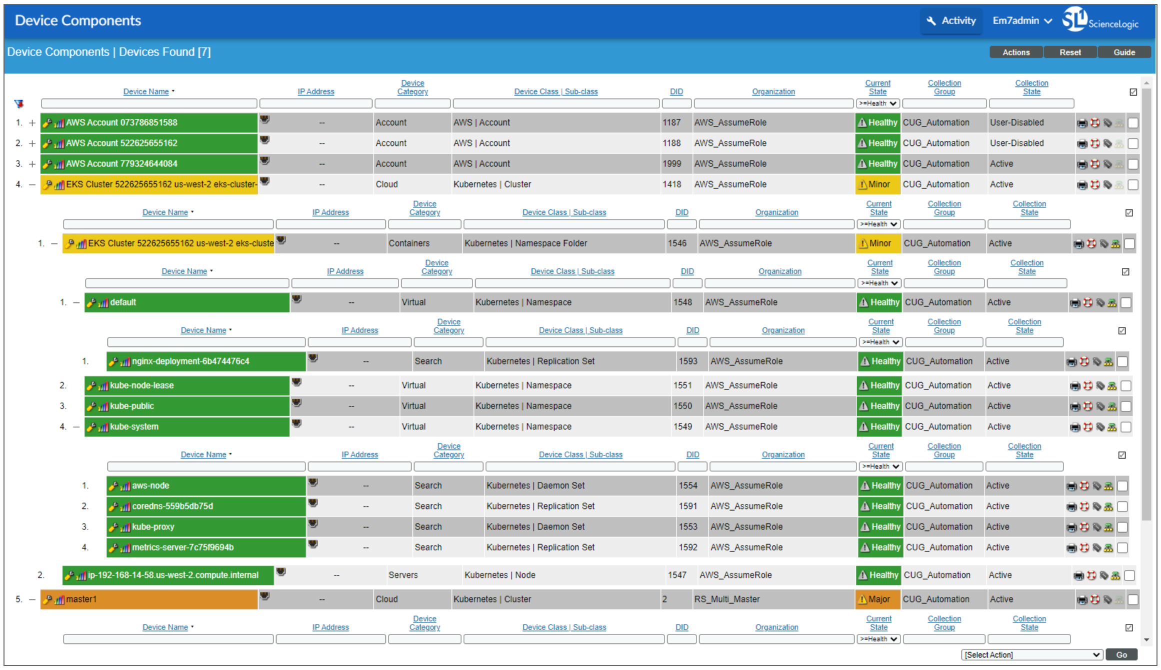
When SL1 performs collection for the Kubernetes cluster, SL1 will create component devices that represent each device and align other Dynamic Applications to those component devices. Some of the Dynamic Applications aligned to the component devices will also be used to create additional component devices. All component devices appear in the **Devices** page just like devices discovered using the ScienceLogic discovery process.

In addition to the **Devices** page, you can view the Kubernetes cluster and all associated 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



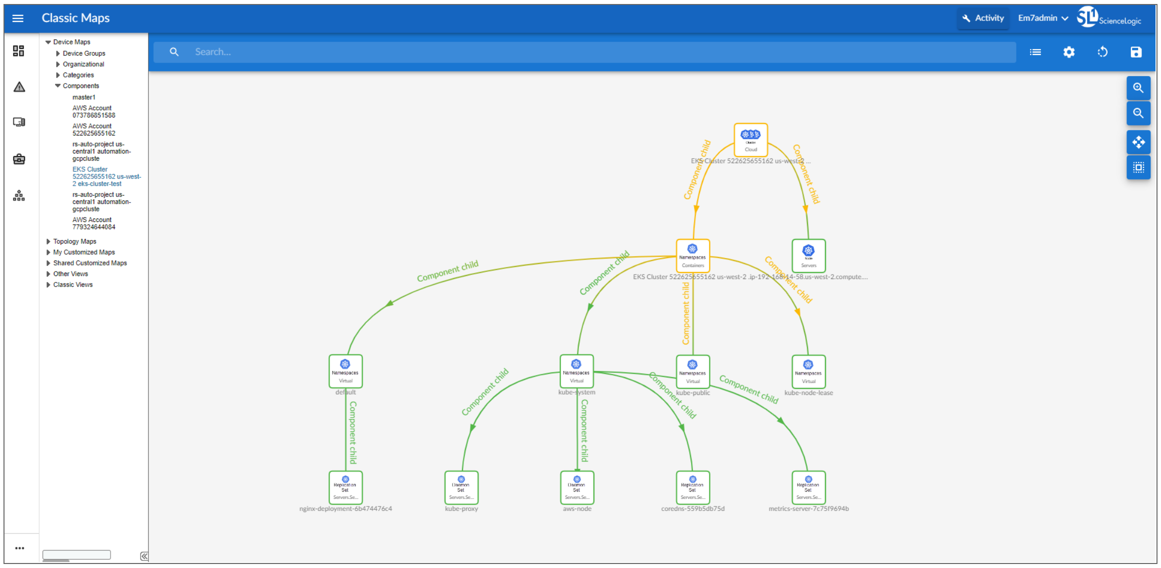
- The **Device Components** page (Devices > Device Components) displays a list of all root devices and component devices discovered by SL1 in an indented view, so you can easily view the hierarchy and relationships between child devices, parent devices, and root devices. To view the component devices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



Device Components | Devices Found [7]

	Device Name	IP Address	Device Category	Device Class | Sub-class	DID	Organization	Current State	Collection Group	Collection State
1. +	AWS Account 073786851588	--	Account	AWS | Account	1187	AWS_AssumeRole	Healthy	CUG_Automation	User-Disabled
2. +	AWS Account 522625655162	--	Account	AWS | Account	1188	AWS_AssumeRole	Healthy	CUG_Automation	User-Disabled
3. +	AWS Account 779324644084	--	Account	AWS | Account	1999	AWS_AssumeRole	Healthy	CUG_Automation	Active
4. –	EKS Cluster 522625655162 us-west-2 eks-cluster-	--	Cloud	Kubernetes | Cluster	1418	AWS_AssumeRole	Minor	CUG_Automation	Active
	Device Name	IP Address	Device Category	Device Class | Sub-class	DID	Organization	Current State	Collection Group	Collection State
1. –	EKS Cluster 522625655162 us-west-2 eks-cluste	--	Containers	Kubernetes | Namespace Folder	1546	AWS_AssumeRole	Minor	CUG_Automation	Active
	Device Name	IP Address	Device Category	Device Class | Sub-class	DID	Organization	Current State	Collection Group	Collection State
1. –	default	--	Virtual	Kubernetes | Namespace	1548	AWS_AssumeRole	Healthy	CUG_Automation	Active
	Device Name	IP Address	Device Category	Device Class | Sub-class	DID	Organization	Current State	Collection Group	Collection State
1.	nginx-deployment-6b474476c4	--	Search	Kubernetes | Replication Set	1593	AWS_AssumeRole	Healthy	CUG_Automation	Active
2.	kube-node-lease	--	Virtual	Kubernetes | Namespace	1551	AWS_AssumeRole	Healthy	CUG_Automation	Active
3.	kube-public	--	Virtual	Kubernetes | Namespace	1550	AWS_AssumeRole	Healthy	CUG_Automation	Active
4. –	kube-system	--	Virtual	Kubernetes | Namespace	1549	AWS_AssumeRole	Healthy	CUG_Automation	Active
	Device Name	IP Address	Device Category	Device Class | Sub-class	DID	Organization	Current State	Collection Group	Collection State
1.	aws-node	--	Search	Kubernetes | Daemon Set	1554	AWS_AssumeRole	Healthy	CUG_Automation	Active
2.	coredns-559b5db75d	--	Search	Kubernetes | Replication Set	1591	AWS_AssumeRole	Healthy	CUG_Automation	Active
3.	kube-proxy	--	Search	Kubernetes | Daemon Set	1553	AWS_AssumeRole	Healthy	CUG_Automation	Active
4.	metrics-server-7c75f9694b	--	Search	Kubernetes | Replication Set	1592	AWS_AssumeRole	Healthy	CUG_Automation	Active
2.	ip-192-168-14-58.us-west-2.compute.internal	--	Servers	Kubernetes | Node	1547	AWS_AssumeRole	Healthy	CUG_Automation	Active
5. –	master1	--	Cloud	Kubernetes | Cluster	2	RS_Multi_Master	Major	CUG_Automation	Active
	Device Name	IP Address	Device Category	Device Class | Sub-class	DID	Organization	Current State	Collection Group	Collection State

- 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. SL1 also updates each map with the latest status and event information. To view the map for a Kubernetes cluster, go to Classic Maps > Device Maps > Components, and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the ***Maps*** manual.



Chapter 3

Dashboards

Overview

The following sections describe the device dashboards that are included in the *Kubernetes* PowerPack:

This chapter covers the following topics:

Device Dashboards

The *Kubernetes* PowerPack includes device dashboards that provide summary information for Kubernetes component devices. The following device dashboards in the *Kubernetes* PowerPack are aligned as the default device dashboard for the equivalent device class.

NOTE: If the device dashboards are not populating data for your selected time frame, change the time frame then click back to your desired time frame and the data will populate.

Kubernetes Cluster

The Kubernetes Cluster device dashboard displays the following information:

- The basic information about the device
- Six gauges that display the following metrics:
  - Total number of nodes
  - Total number of ready nodes
  - Number of CPUs
  - Number of Controllers
  - Number of Pods
  - Number of ScienceLogic devices
- The average pod lifetime
- Top nodes, sorted by CPU utilization
- Top nodes, sorted by the number of Pods
- Top nodes, sorted by the number of TCP segments received

Kubernetes Node

The Kubernetes Node device dashboard displays the following information:

- The basic information about the device
- Number of active pods
- Average number of containers per pod
- Memory utilization
- CPU utilization

- Top file systems, sorted by utilization
- Top 10 interfaces, sorted by the number of inbound packets.

Kubernetes Namespace

The Kubernetes Namespace device dashboard displays the following information:

- The basic information about the device
- Top controllers, sorted by the number of pods
- Top controllers, sorted by the number of restarts
- Top controllers, sorted by memory utilization
- Top controllers, sorted by CPU utilization

Kubernetes Controllers

The Kubernetes Controllers device dashboard displays the following information:

- The basic information about the device
- Controller memory utilization
- Controller CPU utilization
- Controller pod and container count
- Container restart count
- AutoScale count
- Resource CPU requests

Appendix

3

Kubernetes API Endpoints

Overview

This appendix describes the list of API endpoints being requested from and their matching Dynamic Applications.

Kubernetes API Endpoints

The Kubernetes API Endpoints listed below are requested by the listed Dynamic Applications.

- `/api/v1/componentstatuses`
  - Kubernetes: Component Status
- `/apis/apps/v1/replicasets`
  - Kubernetes: Component Count Performance
  - Kubernetes: Controller Performance
  - Kubernetes: Controller Pod Configuration
  - Kubernetes: Controller Discovery
- `/apis/batch/v1/jobs`
  - Kubernetes: Component Count Performance
  - Kubernetes: Controller Discovery
- `/api/v1/services`
  - Kubernetes: Ingress Controller Configuration
  - Kubernetes: Service Configuration

- `/apis/networking.k8s.io/v1/ingresses`
    - Kubernetes: Ingress Controller Configuration
    - Kubernetes: Ingress Controller Discovery
- `/apis/metrics.k8s.io/v1beta1/nodes`
    - Kubernetes: Node Performance
- `/api/v1/pods`
    - Kubernetes: Cluster Performance
    - Kubernetes: Component Count Performance
    - Kubernetes: Controller Performance
    - Kubernetes: Controller Pod Configuration
    - Kubernetes: Pod Configuration
    - Kubernetes: Pod Performance (Node)
    - Kubernetes: Controller Discovery
- `/apis/apps/v1/daemonsets`
    - Kubernetes: Component Count Performance
    - Kubernetes: Controller Discovery
- `/apis/apps/v1/deployments`
    - Kubernetes: Component Count Performance
    - Kubernetes: Controller Performance
    - Kubernetes: Pod Configuration
    - Kubernetes: Controller Discovery
- `/api/v1/events`
    - Kubernetes: Events Configuration
- `/apis/apps/v1/statefulsets`
    - Kubernetes: Component Count Performance
    - Kubernetes: Controller Discovery
- `/api/v1/persistentvolumes`
    - Kubernetes: Persistent Volume Configuration

- `/api/v1/nodes`
  - Kubernetes: Cluster Performance
  - Kubernetes: Component Count Performance
  - Kubernetes: Node Configuration
  - Kubernetes: Node Discovery
  - Kubernetes: Node Performance
- `/apis/autoscaling/v2/horizontalpodautoscalers`
  - Kubernetes: Controller Pod Performance
  - Kubernetes: Horizontal Pod Autoscaler Configuration
- `/api/v1/replicationcontrollers`
  - Kubernetes: Component Count Performance
  - Kubernetes: Controller Discovery
- `/apis/batch/v1/cronjobs`
  - Kubernetes: Component Count Performance
  - Kubernetes: Controller Discovery
- `/apis/metrics.k8s.io/v1beta1/pods`
  - Kubernetes: Controller Pod Performance
- `/api/v1/namespaces`
  - Kubernetes: Namespace Discovery
  - Kubernetes: Namespace Folder Discovery

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

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. For more information, see https://sciencelogic.com/company/legal.

ScienceLogic

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

International: +1-703-354-1010