Foglight® for Container Management 1.0

User and Administration Guide

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept.
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our website (https://www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal.

Trademarks

Quest, the Quest logo, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit https://www.quest.com/legal/trademark-information.aspx. “Apache HTTP Server”, Apache, “Apache Tomcat” and “Tomcat” are trademarks of the Apache Software Foundation. Google is a registered trademark of Google Inc. Android, Chrome, Google Play, and Nexus are trademarks of Google Inc. Red Hat, JBoss, the JBoss logo, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the U.S. and other countries. CentOS is a trademark of Red Hat, Inc. in the U.S. and other countries. Fedora and the Infinity design logo are trademarks of Red Hat, Inc. Microsoft, .NET, Active Directory, Internet Explorer, Hyper-V, Office 365, SharePoint, Silverlight, SQL Server, Visual Basic, Windows, Windows Vista and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. AIX, IBM, PowerPC, PowerVM, and WebSphere are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Java, Oracle, Oracle Solaris, PeopleSoft, Siebel, Sun, WebLogic, and ZFS are trademarks or registered trademarks of Oracle and/or its affiliates in the United States and other countries. SPARC is a registered trademark of SPARC International, Inc. in the United States and other countries. Products bearing the SPARC trademarks are based on an architecture developed by Oracle Corporation. OpenLDAP is a registered trademark of the OpenLDAP Foundation. HP is a registered trademark that belongs to Hewlett-Packard Development Company, L.P. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries. Novell and eDirectory are registered trademarks of Novell, Inc., in the United States and other countries. VMware, ESX, ESXi, vSphere, vCenter, vMotion, and vCloud Director are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Sybase is a registered trademark of Sybase, Inc. The X Window System and UNIX are registered trademarks of The Open Group. Mozilla and Firefox are registered trademarks of the Mozilla Foundation. “Eclipse”, “Eclipse Foundation Member”, “EclipseCon”, “Eclipse Summit”, “Built on Eclipse”, “Eclipse Ready” “Eclipse Incubation”, and “Eclipse Proposals” are trademarks of Eclipse Foundation, Inc. IOS is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. Apple, iPad, iPhone, Mac OS, Safari, Swift, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Ubuntu is a registered trademark of Canonical Ltd. Symantec and Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. OpenSUSE, SUSE, and YAST are registered trademarks of SUSE LCC in the United States and other countries. Citrix, AppFlow, NetScaler, XenApp, and XenDesktop are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. AlertSite and DéjàClick are either trademarks or registered trademarks of Boca Internet Technologies, Inc. Samsung, Galaxy S, and Galaxy Note are registered trademarks of Samsung Electronics America, Inc. and/or its related entities. MOTOROLA is a registered trademarks of Motorola Trademark Holdings, LLC. The Trademark BlackBerry Bold is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries. Quest is not endorsed, sponsored, affiliated with or otherwise authorized by Research In Motion Limited. Ixia and the Ixia four-petal logo are registered trademarks or trademarks of Ixia. Opera, Opera Mini, and the O logo are trademarks of Opera Software ASA. Tevron, the Tevron logo, and CitraTest are registered trademarks of Tevron, LLC. PostgreSQL is a registered trademark of the PostgreSQL Global Development Group. MariaDB is a trademark or registered trademark of MariaDB Corporation Ab in the European Union and United States of America and/or other countries. Vormetric is a registered trademark of Vormetric, Inc. Intel, Itanium, Pentium, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Debian is a registered trademark of Software in the Public Interest, Inc. OpenStack is a trademark of the OpenStack Foundation. Amazon Web Services, the “Powered by Amazon Web Services” logo, and “Amazon RDS” are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries. Infobright, Infobright Community Edition and Infobright Enterprise Edition are trademarks of Infobright Inc. POLYCOM®, RealPresence® Collaboration Server, and RMX® are registered trademarks of Polycom, Inc. All other trademarks and registered trademarks are property of their respective owners.

Legend

🟥 **WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.**

! **CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.**

i **IMPORTANT NOTE**, **NOTE**, **TIP**, **MOBILE**, or **VIDEO:** An information icon indicates supporting information.

Foglight for Container Management User and Administration Guide
Updated - November 2018
Foglight Version - 5.9.4
Software Version - 1.0

Contents

1

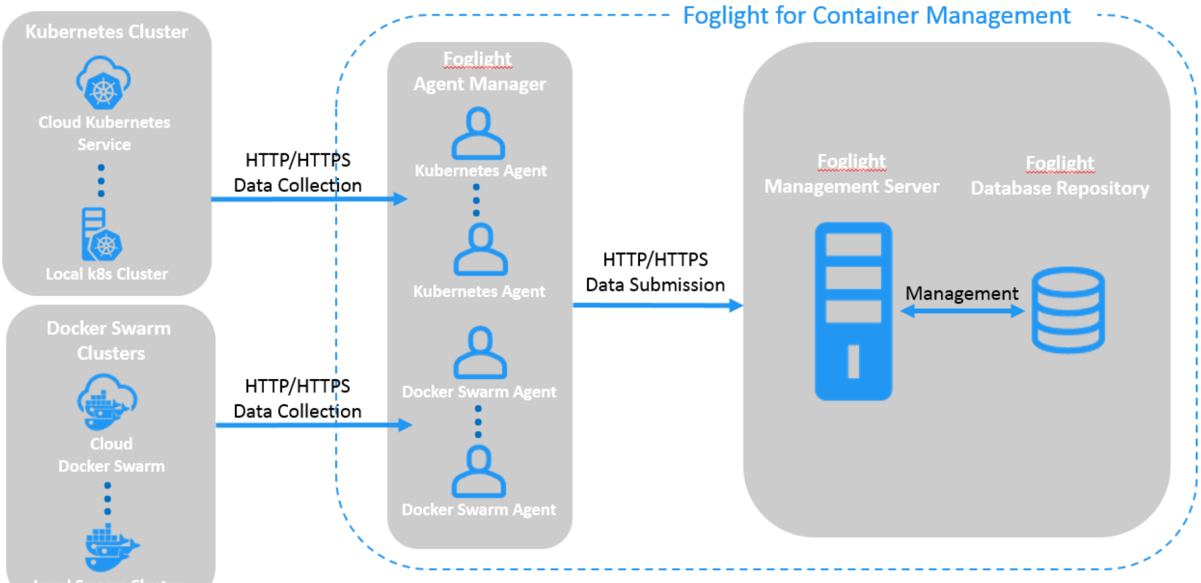
Understanding Foglight® for Container Management

About Foglight for Container Management

Containers are a method of operating system virtualization that allow you to run an application and its dependencies in resource-isolated processes. Foglight® for Container Management simplifies this process by tracking each container, the resources it consumes, and the remaining compute of the container host, as well as providing you with the cluster information and pre-configured rules with notifications identifying the problem of your clusters.

Architecture

Figure 1. Components of Foglight for Container Management



Data Collection

- Azure AKS: https://XXX.hcp.<region>.azmk8s.io:443
- Amazon EKS: https://XXX.sk1.<region>.eks.amazonaws.com
- Others: <Kubernetes API Server Endpoint>

Data Submission

- HTTP 8080
- HTTPS 8443

Foglight for Container Management consists of three main components:

- Foglight Management Server and Foglight Database Repository — Responsible for managing, alerting, and viewing the collected data. Both components can be set to run on the same machine or reside on separate machines.
- Agent Manager — Hosts the monitoring Kubernetes agents.
- Docker Swarm clusters — Manages containerized applications in a clustered environment.
- Kubernetes clusters — Manages containerized applications in a clustered environment.

Sizing Your Monitored Environment

Consider the possibility of a great amount of objects being collected, analyzed, and maintained by the application, several aspects of the underlying server must be taken into account. The sizing of the supporting clusters and containers depends on the complexity of the underlying environment. Sufficient processing power and CPU memory are required to support effective collection, server data handling, and analytics.

NOTE: Currently Quest validates the environment with up to 10000 containers. If your environment beyonds this scale, contact Quest Support.

Foglight Management Server Requirements

The minimum system requirements of the Foglight Management Server vary from the scale of clusters. The scale of clusters is determined by running containers.

Table 1. Foglight Management Server requirements

Operating System	Maximum Containers	Foglight		Agent Manager	
		JVM Settings	# of CPUs	JVM Settings	# of CPUs
Windows 64-bit	1000	Xms|Xmx=4G	2	Xms|Xmx=4G	2
	5000	Xms|Xmx=8G	4	Xms|Xmx=8G	4
	10000	Xms|Xmx=12G	6	Xms|Xmx=12G	6
Linux 64-bit	1000	Xms|Xmx=4G	2	Xms|Xmx=4G	2
	5000	Xms|Xmx=8G	4	Xms|Xmx=8G	4
	10000	Xms|Xmx=12G	6	Xms|Xmx=12G	6

If you are using an embedded Agent Manager, make sure to use the sum resources of both Foglight and Agent Manager.

Kubernetes Agent Requirements

Kubernetes Agent collects inventory and metrics every 5 minutes by default. Refer to Configuring data collection interval for details about how to change the collection interval.

Table 2. Kubernetes Agent requirements

Maximum Containers	Kubernetes Agent Collection Interval (minutes)	
	Inventory	Metrics
500	5	5
1000	10	10
5000	30	30
10000	60	60

Table 2 is the recommendations for local Kubernetes clusters. If you deploy Kubernetes clusters on the Cloud Provider Kubernetes Service, consider your network rate and change your configurations based on different Cloud Provider and different region/zone of your cluster.

Docker Swarm Agent Requirements

Docker Swarm Agent collects inventory and metrics every 5 minutes by default. Refer to Configuring data collection interval for details about how to change the collection interval.

Table 3. Docker Swarm Agent requirements

Maximum Containers	Docker Swarm Agent Collection Interval (minutes)	
	Inventory	Metrics
500	5	5
1000	10	10
5000	30	30

Table 3 is the recommendations for local Docker Swarm clusters. For cloud environment, consider network rate and change configurations based on different Cloud Provider and different region/zone.

Getting Started

Prerequisite

Kubernetes Agent

Each Kubernetes Agent monitors the assets inside the selected Kubernetes Service Providers. To enable the data collection, complete the following prerequisites before create agent.

- Preparing the Kubernetes credential
- Enabling Heapster service in monitored environment

Preparing the Kubernetes credential

The Kubernetes configuration file named *KubeConfig* is a standard configuration of Kubernetes and is required for Kubernetes agents to access the cluster. Foglight for Container Management verifies and supports the local Kubernetes and the following Cloud Kubernetes Service Providers. Based upon your environment, select either of approaches to get your *KubeConfig* file:

> **NOTE:** Data from different Kubernetes Agents with the same cluster name will be merged into one cluster.

- Local Kubernetes
- Azure Kubernetes Service (AKS)
- Amazon Elastic Container Service for Kubernetes (EKS)
- Google Cloud Platform Container Engine (GKE)
- IBM Cloud Kubernetes Service
- Openshift Origin

Local Kubernetes

If you build a Kubernetes cluster locally, find this *KubeConfig* file under the `/etc/kubernetes/admin.kubeconfig` on your master node.

Azure Kubernetes Service (AKS)

Before generating the Kubernetes credentials, record the following information:

- Azure Username
- Azure Password
- Azure Subscription Number
- The name of your AKS Cluster Resource Group
- The name of your AKS cluster

Download the Azure Command Line Interface and install it in your local platform, and then follow steps below to generate your Kubernetes credential:

1 Run the command *`az login`*.

   Then a browser shows up, directing you to the Azure Portal where you should enter your Azure Username and Password to complete the authentication.

2 Run the command: *`az account set --subscription <azure subscription number>`*

3 Run the command: *`az aks get-credentials --resource-group <azure resource group name> --name <azure cluster name>`*

4 Find the Kubernetes configuration file under *`<USER_HOME>/.kube/config`* on your local platform.

   NOTE: The token in this Kubernetes configuration file will get expired after two years. If you don’t want the credential gets expired, refer to Foglight Container Tools for detail.

Amazon Elastic Container Service for Kubernetes (EKS)

Follow the Amazon EKS offical guide Getting Started with Amazon EKS. Follow the guide and complete Create a kubeconfig for Amazon EKS. in the end of the guide.

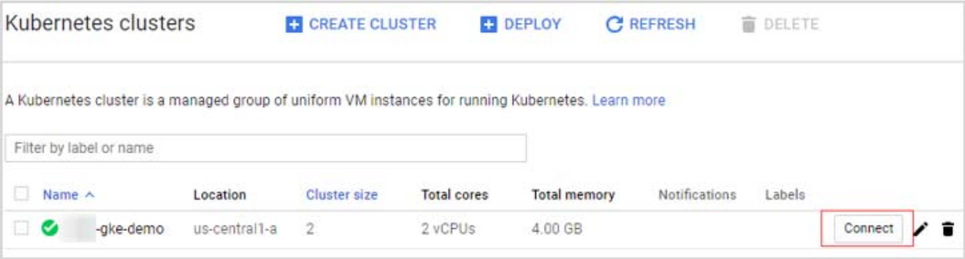
NOTE: If you don’t want the credential gets expired, refer to Foglight Container Tools for detail.

Google Cloud Platform Container Engine (GKE)

Download the Google Cloud Client tool and install it in your local platform, and then follow steps below to generate your Kubernetes credential:

1 Generate the intermediate Kubernetes credential for your cluster.

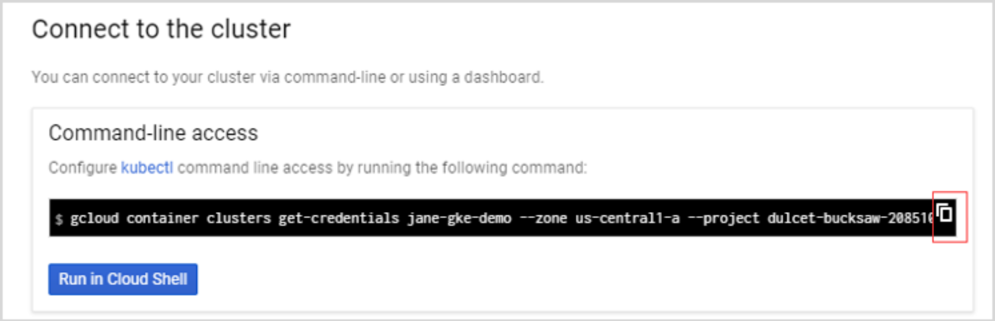
   a Log into your Kubernetes cluster, click **Connect** next to your cluster name.



A Kubernetes cluster is a managed group of uniform VM instances for running Kubernetes. Learn more

	Name	Location	Cluster size	Total cores	Total memory	Notifications	Labels	
□	✓ [redacted]-gke-demo	us-central1-a	2	2 vCPUs	4.00 GB			Connect

   b Click to copy the command below, and then run this command.



Connect to the cluster

You can connect to your cluster via command-line or using a dashboard.

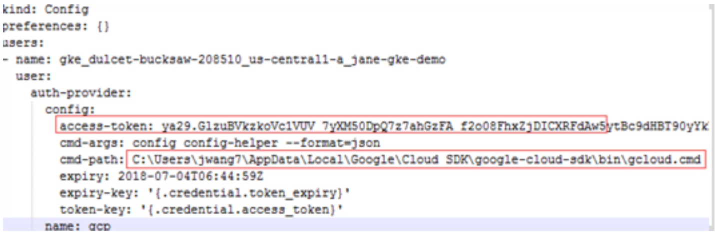
Command-line access

Configure kubectl command line access by running the following command:

```
$ gcloud container clusters get-credentials jane-gke-demo --zone us-central1-a --project dulcet-bucksaw-20851(
```

c Find the intermediate Kubernetes configuration file under `<USER_HOME>/.kube/config` on your local platform. The following is the example of this intermediate Kubernetes configuration file.

> **NOTE:** This Kubernetes configuration file cannot be used as the agent credential because the token in this file will get expired soon and "*cmd-path*" of the token directs to your local platform.



```
kind: Config
preferences: {}
users:
- name: gke_dulcet-bucksaw-208510_us-central1-a_jane-gke-demo
  user:
    auth-provider:
      config:
        access-token: ya29.GlzuBVkzkoVc1VUV 7yXM50DpQ7z7ahGzFA f2o08FhxZjDICXRFdAw5ytBc9dHBT90yYk
        cmd-args: config config-helper --format=json
        cmd-path: C:\Users\jwang7\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin\gcloud.cmd
        expiry: 2018-07-04T06:44:59Z
        expiry-key: '{.credential.token_expiry}'
        token-key: '{.credential.access_token}'
      name: gcp
```

d Open Google Cloud Client tool and run the following commands to create a Kubernetes service account that grants with the *cluster-admin* role and the access to your Google Kubernetes Engine (GKE) cluster.

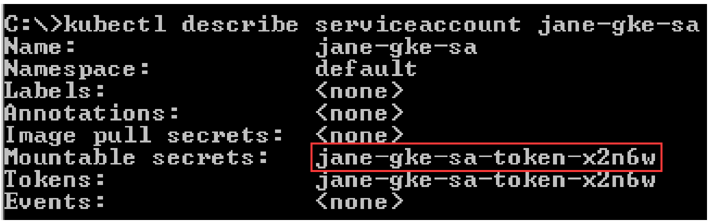
a *kubectl create serviceaccount <service account name>*

b *kubectl create clusterrolebinding <cluster role binding name> --clusterrole=cluster-admin -serviceaccount=default:<service account name>*

"default" in the above command is the namespace name of this service account name. The name space name will be "default" if you do not change it. You can also change to other namespace names, as needed.

c *kubectl describe serviceaccount <service account name>*

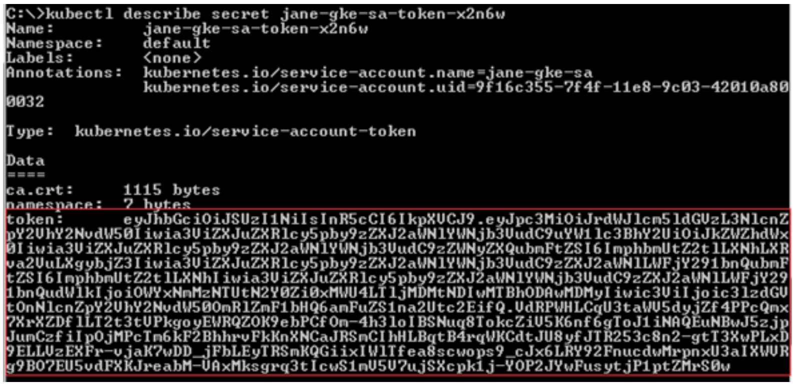
You will get the response similar as below. Record the <secret name> for later use.



```
C:\>kubectl describe serviceaccount jane-gke-sa
Name:                     jane-gke-sa
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Image pull secrets:       <none>
Mountable secrets:        jane-gke-sa-token-x2n6w
Tokens:                   jane-gke-sa-token-x2n6w
Events:                   <none>
```

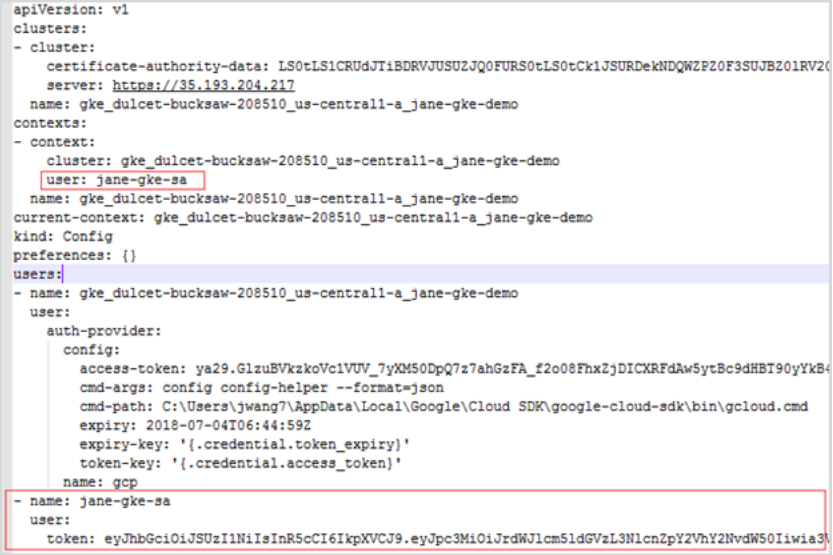
d *kubectl describe secret <secret name>*

You will get response similar as below. Record the token value (exclude "token:") for later use.



```
C:\>kubectl describe secret jane-gke-sa-token-x2n6w
Name:         jane-gke-sa-token-x2n6w
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/service-account.name=jane-gke-sa
              kubernetes.io/service-account.uid=9f16c355-7f4f-11e8-9c03-42010a80
0032

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1115 bytes
namespace:  7 bytes
token:      eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZ
pY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx
0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6ImphbmUtZ2tlLXNhLXR
va2VuLXgybjZ3Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmF
tZSI6ImphbmUtZ2tlLXNhIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY29
1bnQudWlkIjoiOWYxNmMzNTUtN2Y0Zi0xMWU4LTljMDMtNDIwMTBhODAwMDMyIiwic3ViIjoic3lzdGV
tOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6amFuZS1na2Utc2EifQ.VdRPWHLCqU3taWU5dyjZf4PPcQmx
7XrXZDf1LT2t3tVPkgoyEWRQZOK9ebPCfOm-4h3loIBSNuq8TokcZiU5K6nf6gToJ1iNAQEuNBwJ5zjp
JumCzfiIpOjMPcTm6kF2BhhrvFkKnXNCaJRSmCIhHLBqtB4rqWKCdtJU8yfJTR253c8n2-gtT3XwPLxD
9ELLUzEXFr-vjaK7wDD_jFbLEyTRSmKQGiixIW1Tfea8scwops9_cJx6LRY92FnucdwMrpnxU3aIXWUR
g9BO7EU5vdFXKJreabM-UAxMksgrq3tIcwS1mU5U7ujSXcpk1j-YOP2JYwFusytjP1ptZMrS0w
```

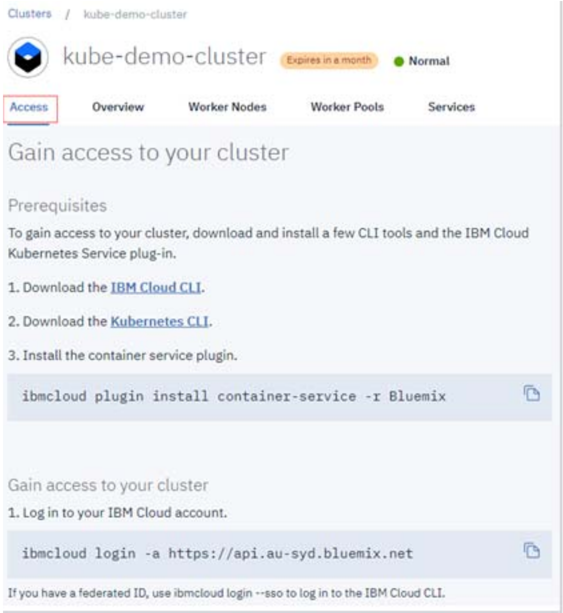
e   Open the intermediate Kubernetes configuration file under <USER_HOME>/.kube/config, and then add the user and change the token to the new one.



```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURDekNDQWZPZ0F3SUJBZ01RV2
    server: https://35.193.204.217
  name: gke_dulcet-bucksaw-208510_us-central1-a_jane-gke-demo
contexts:
- context:
    cluster: gke_dulcet-bucksaw-208510_us-central1-a_jane-gke-demo
    user: jane-gke-sa
  name: gke_dulcet-bucksaw-208510_us-central1-a_jane-gke-demo
current-context: gke_dulcet-bucksaw-208510_us-central1-a_jane-gke-demo
kind: Config
preferences: {}
users:
- name: gke_dulcet-bucksaw-208510_us-central1-a_jane-gke-demo
  user:
    auth-provider:
      config:
        access-token: ya29.GlzuBVkzkoVc1VUV_7yXM50DpQ7z7ahGzFA_f2o08FhxZjDICXRFdAw5ytBc9dHBT90yYkB
        cmd-args: config config-helper --format=json
        cmd-path: C:\Users\jwang7\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin\gcloud.cmd
        expiry: 2018-07-04T06:44:59Z
        expiry-key: '{.credential.token_expiry}'
        token-key: '{.credential.access_token}'
      name: gcp
- name: jane-gke-sa
  user:
    token: eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3
```

IBM Cloud Kubernetes Service

If you have created your cluster on IBM Cloud Kubernetes Service, get the access from the console as described on the cluster’s *Access* view. You will get a .pem file and a .yml file after you performing the steps.



Gain access to your cluster

Prerequisites

To gain access to your cluster, download and install a few CLI tools and the IBM Cloud Kubernetes Service plug-in.

1. Download the IBM Cloud CLI.

2. Download the Kubernetes CLI.

3. Install the container service plugin.

```
ibmcloud plugin install container-service -r Bluemix
```

Gain access to your cluster

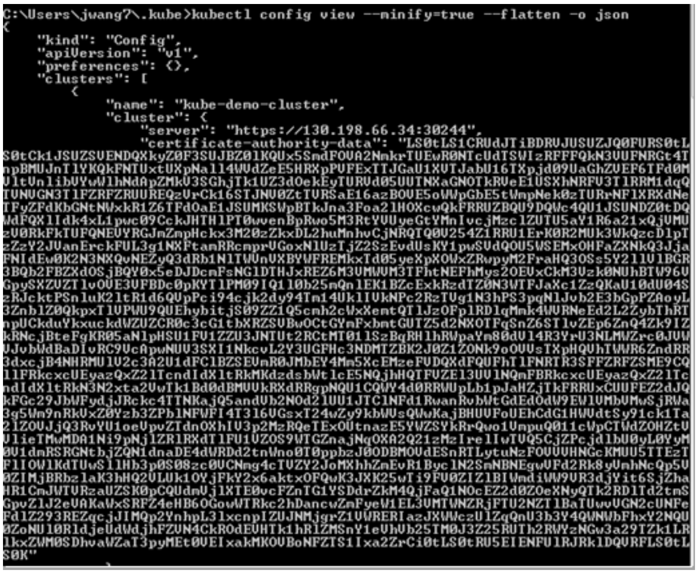
1. Log in to your IBM Cloud account.

```
ibmcloud login -a https://api.au-syd.bluemix.net
```

If you have a federated ID, use ibmcloud login --sso to log in to the IBM Cloud CLI.

By default IBM Cloud Kubernetes Service uses certificate authority file and token/refresh token. However, certificate authority data and service account token should be used in the Kubernetes Agent credential. After you successfully test your connection through "kubectl get nodes", follow the steps below to generate the Kubernetes Agent credential.

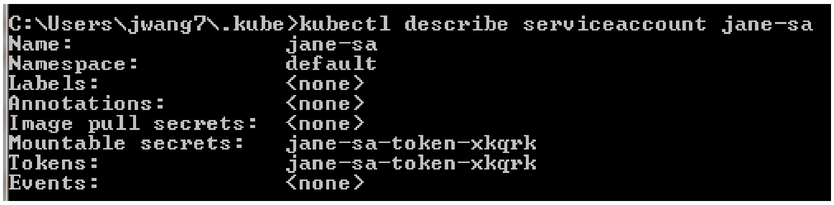
1 Run the command *kubectl config view -minify=true -flatten -o json*. You will get an output similar as below, then record the *<certificate authority data>* for later use.



```
C:\Users\jwang7\.kube>kubectl config view --minify=true --flatten -o json
{
    "kind": "Config",
    "apiVersion": "v1",
    "preferences": {},
    "clusters": [
        {
            "name": "kube-demo-cluster",
            "cluster": {
                "server": "https://130.198.66.34:30244",
                "certificate-authority-data": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tL
S0tCk1JSUZSVENDQXkyZ0F3SUJBZ01KQUx5SmdFOVA2NmkrTUEwR0NTcUdTSWIzRFFFQkN3VUFNRGt4T
npBMUJnTlYKQkFNTUxtUXpNall4WVdZeE5HRXpPVFExTTJGaU1XVTJabU16TXpjd09UaGhZVEF6TFd0M
VltVnlibVYwWlhNdApZMkU3SGhjTk1UZ3dOekEyTURVd05UUTNXaGNOTkRVeE1USXhNRFU3TlRRM1dqQ
TVNVGN3T1FZRFZRUUREQzVrCk16STJNV0ZtTVRSaE16azBOVE5oWWpGbE5tWmpNek0zTURrNF1XRXdNe
TFyZFdKbGNtNWxkR1Z6TFdOaE1JSUMKSWpBTkJna3Foa2lHOXcwQkFRRUZBQU9DQWc4QU1JSUNDZ0tDQ
WdFQX1Idk4xL1pwc09CckJHTH1PT0wvenBpRwo5M3RtYVUyeGtYMnIvcjMzc1ZUTU5aY1R6a21xQjVMU
zV0RkFkTUFQNEVYRGJmZmpHckx3M20zZkxDL2huMnhvCjNRQTQ0V254Z1RRU1ErK0R2MUk3WkQzcDlpT
zZzY2JVanErckFUL3g1NXFtamRRcmprVGoxNlUzTjZ2SzEvdUsKY1pwSVdQOU5WSEMxOHFaZXNkQ3Jja
FNIdEw0K2N3NXQvNEZyQ3dRb1N1TWVnUXBYWFREMkxTd05yeXpXOWxZRwpyM2FraHQ3OSs5Y2llV1BGR
3BQb2FBZXdOSjBQY0x5eDJDcmFsNG1DTHJxREZ6M3VMWVM3TFhtNEFhMys2OEVxCkM3Vzk0NUhBTW96V
GpySXZUZTlvOVE3VFBDc0pKYTlPM09IQ1l0b25mQnlEK1BZcExkRzdTZ0N3WTFJaXc1ZzQKaU10dU04S
zRJcktPSnluK2ltR1d6QVpPci94cjk2dy94Tm14UklIVkNPc2RzTVg1N3hPS3pqN1Jvb2E3bGpPZAoyL
3Znb1Z0QkpxT1VPWU9QUEhybitjS09ZZ1Q5cmh2cWxXemtQT1JzOFplRDlqMmk4WVRNeEd2L2ZybThRT
npUCkduYkxucFZUZUZCR0c3cG1tbXRZSVBwOCtGYmFxbmtGUTZ5d2NXOTFqSnZ6STlvZEp6ZnQ4Zk9IZ
kRNcjBteFgKR05aNlpHSU1FV1ZZU3JNTUt2RCtMT011SzBqRH1hRWpaYm80dV14R3YrU3NLMWZrc0JUW
UJvbWdBaDIvRC9VcApwNUV3SXI1NkcvL2Y3UGFHc3NDMTZBK2J0Z1ZONk9oOVVsTXpHQVhTWWR6ZndRR
3dxcjB4NHRMU1V2c3A2U1dFC1BZSEVmR0JMbEY4Mm5XcEMzeFVDQXdFQUFhT1FNRTR3SFFZRFZSME9CQ
11FRkcxcUEyazQxZ21TcndIdX1tRkMKdzdsbWtlcE5NQjhHQTFVZE13UV1NQmFBRkcxcUEyazQxZ21Tc
ndIdX1tRkN3N2xta2VwTk1Bd0dBMVVkRXdRRgpNQU1CQWY4d0RRWUpLb1pJaHZjTkFRRUxCUUFFZ2dJQ
kFGc29JbWFydjJRckc4TTNKajQ5andVb2NOd21UU1JTC1NFd1RwanRvbWtGdEdOdW9EV1VMbVMwSjRWa
3g5Wm9nRkVxZ0Yzb3ZPb1NFWFI4T316VGsxT24wZy9kbWVsQWwKajBHUVFoUEhCdG1HWVdtSy91ck1Ta
21ZOUJjQ3RvYU1oeVpvZTdnOXhIV3p2MzRQeTExOUtnazE5YWZSYkRrQwo1VmpuQ011cWpCTWdZOHZtV
UlieTMwMDA1Ni9pNj1ZR1RXdT1FU1VZOS9WTGZnajNqOXA2Q21zMzIrelIwTVQ5CjZPcjd1bU0yL0YyM
0V1dmRSRGNtbjZQN1dnaDE4dWRDd2tnWno0T0ppbzJ0ODBMOVdESnRTLytuNzFOVVVHNGcKMUU5TTEzT
F1IOW1KdTUwS11Hb3p0S08zc0VCNmg4cTVZY2JoMXhhZmEvR1BycIN2SmNBNEgwVFd2Rk8yUmhNcQp5V
0ZIMjBRbz1aK3hHQ2VLUk1OYjFkY2x6aktxOFQwK3JXK25wTi9FV0ZIZ1BIWmdiWW9VR3djYit6SjZha
HR1CmJWTVRzaUZSK0pCQUdmVj1XTE0vcFZnTG1YSDdrZkM4QjFaQ1NOcEZ2d0ZOeXNyQTk2RD1Td2tmS
GpvZ1J2eVAKaWxSRFZ4eHB6OGowWTRkc2hDancwZmFyeW1EL3VMTWNZRjFTU2NZT1BaTUwvUGN2cUNFe
Fd1Z293REZqcjJIMQp2YnhpL31xcnpIZUJNMjgrZ1VWRERIazJXWWczU1ZqQnU3b3Y4QWNWbFhxY2NQU
0ZoNU10R1djeUdWdjhFZUN4CkROdEVHTk1hR1ZMSnY1eVhVb25TM0J3Z25RUTh2RWYzNGw3a29TZk1LR
1kxZWM0SDhvaWZaT3pyMEt0VEIxakMKOVBoNFZTS1Ixa2ZrCi0tLS0tRU5EIENFUlRJRk1DQVRFLS0tL
S0K"
```

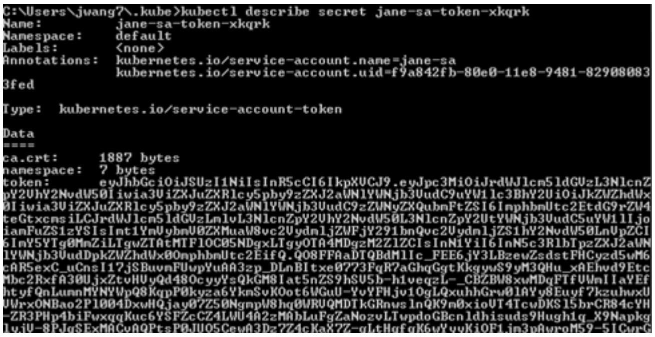
2 Run the command *kubectl create serviceaccount <service account>*.

3 Run the command *kubectl describe serviceaccount <service account>*. You will get a response similar as below, then record <service account secret> (in this sample, it is jane-sa-token-xkqrk) for later use.



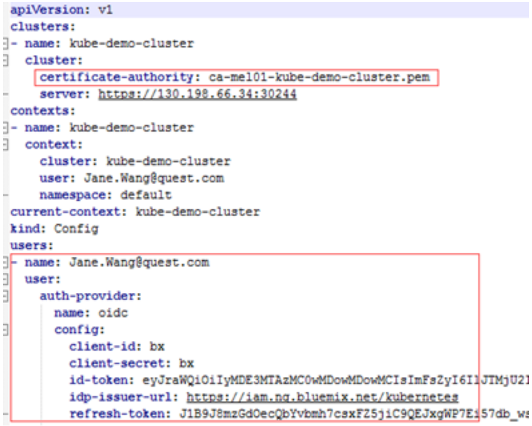
```
C:\Users\jwang7\.kube>kubectl describe serviceaccount jane-sa
Name:                jane-sa
Namespace:           default
Labels:              <none>
Annotations:         <none>
Image pull secrets:  <none>
Mountable secrets:   jane-sa-token-xkqrk
Tokens:              jane-sa-token-xkqrk
Events:              <none>
```

4 Run the command *kubectl describe secret <service account secret>*. You will get a response similar as below, then record <service account token> for later use.



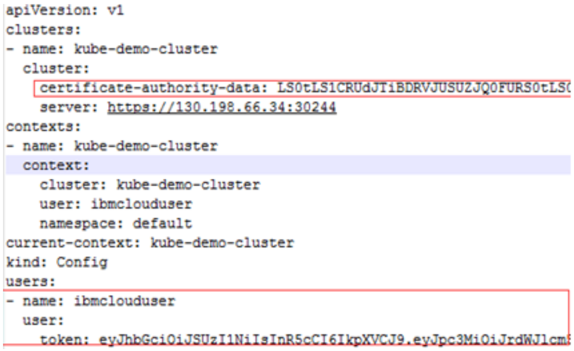
```
C:\Users\jwang7\.kube>kubectl describe secret jane-sa-token-xkqrk
Name:         jane-sa-token-xkqrk
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/service-account.name=jane-sa
              kubernetes.io/service-account.uid=f9a842fb-80e0-11e8-9481-82908083
3fed

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1887 bytes
namespace:  7 bytes
token:      eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZ
pY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx
0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6ImphbmUtc2EtdG9rZW4
teGtxcmsiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjo
iamFuZS1zYSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50LnVpZCI
6ImY5YTg0MmZiLTgwZTAtMTFlOC05NDgxLTgyOTA4MDgzM2ZlZCIsInN1YiI6InN5c3RlbTpzZXJ2aWN
lYWNjb3VudDpkZWZhdWx0OmphbmUtc2EifQ.QO8FFAaDTQBdMlIc_FEE6jY3LBzewZsdstFHCyzd5wM6
cAR5exC_uCnsI17jSBuvmFUwpYuAA3zp_DLnBItxe0773FqR7aGhqGgtKkgywS9yM3QHu_xAEhvd9Etc
Mbc2RxfA30UjxZtvHUyQd48OcyyYsQkGM8Iat5nZS9hSU5b-h1veqzL-_CBZBW8xwMDqFTfUWmIIaYEf
htyfQnLumnMYNYWpQ8KqpP0kyza6YkmSvKOot6WGuU-YvYFHjv1OgLQxuhhGrw0lAYy8Euyf7kzuhwxU
UWrxONBao2Pl004DxwHQjay07Z50NgmpW8hq0WRUQMDTkGRnwslnQK9m0xioUT4TcwDKS15brCR84cYH
-ZR3PHp4biFwxqqKuc6YSFZcCZ4LWU4A2zMAbLuFgZaNozvLTwpdoGBcnldhisuds9Hugh1q_X9Napkg
1vjU-8PJgSExMACvAQPtsP0JU05CewA3Dz7Z4cKaX7Z-gLtHgfgK6wYvvKiOF1jm3pAwroM59-5ICwrG
```

5 Open the .yml file generated previously, which looks like below.



```
apiVersion: v1
clusters:
- name: kube-demo-cluster
  cluster:
    certificate-authority: ca-mel01-kube-demo-cluster.pem
    server: https://130.198.66.34:30244
contexts:
- name: kube-demo-cluster
  context:
    cluster: kube-demo-cluster
    user: Jane.Wang@quest.com
    namespace: default
current-context: kube-demo-cluster
kind: Config
users:
- name: Jane.Wang@quest.com
  user:
    auth-provider:
      name: oidc
      config:
        client-id: bx
        client-secret: bx
        id-token: eyJraWQiOiIyMDE3MTAzMC0wMDowMDowMCIsImFsZyI6IlJTMjU2I
        idp-issuer-url: https://iam.ng.bluemix.net/kubernetes
        refresh-token: J1B9J8mzGdOecQbYvbmh7csxFZ5jiC9QEJxgWP7Ei57db_ws
```

6 Change the certificate authority to the data <certificate authority data> of this authority and change the users section to use <service account token>. Save your changes, and then you will get a credential file like below. This file will be used as the Kubernetes Agent credential to connect to your IBM cloud Kubernetes service cluster.



```
apiVersion: v1
clusters:
- name: kube-demo-cluster
  cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0
    server: https://130.198.66.34:30244
contexts:
- name: kube-demo-cluster
  context:
    cluster: kube-demo-cluster
    user: ibmclouduser
    namespace: default
current-context: kube-demo-cluster
kind: Config
users:
- name: ibmclouduser
  user:
    token: eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJrdWJlcm5
```

Openshift Origin

If you could access the `/etc/origin/master/admin.kubeconfig` on the master node, download this file which can be used as the Kubernetes Agent credential.

If you could not access the `/etc/origin/master/admin.kubeconfig` on the master node, follow instructions below to generate a permanent credential file.

Before generating the permanent Kubernetes credentials, record the following information and ensure you have granted the privilege for accessing the cluster-wide resources:

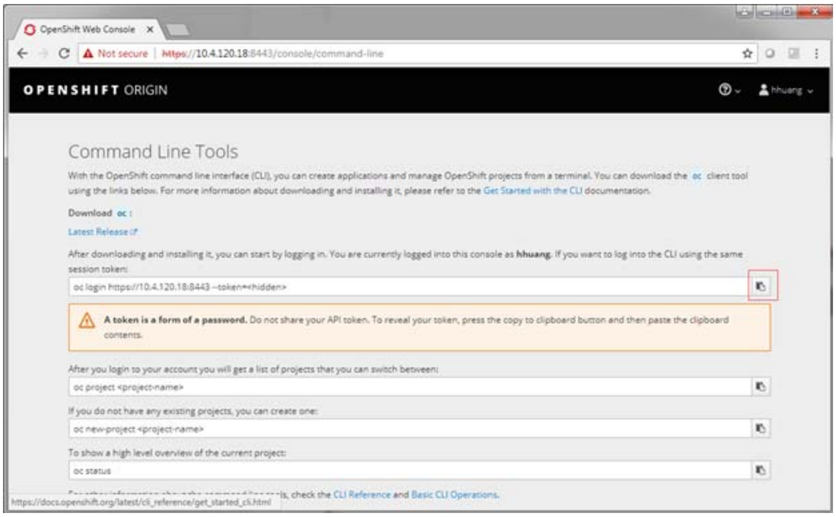
- Openshift Username
- Openshift Password

Download the Openshift Command Line Interface and install it in your local platform, and then follow steps below to generate your Kubernetes credential:

1 Log into Openshift and generate an intermediate Kubernetes configuration file.

   1 After logging into Openshift, click **Command Line Tools** on the upper right.

   2 Click the button next to the *Session token* field, copy the command, and then paste it in your local Command Line Tool. Make sure to find the intermediate Kubernetes configuration file under *`<USER_HOME>/.kube/config`* on your local platform.



Command Line Tools

With the OpenShift command line interface (CLI), you can create applications and manage OpenShift projects from a terminal. You can download the oc client tool using the links below. For more information about downloading and installing it, please refer to the Get Started with the CLI documentation.

Download oc :

Latest Release

After downloading and installing it, you can start by logging in. You are currently logged into this console as **hhuang**. If you want to log into the CLI using the same session token:

```
oc login https://10.4.120.18:8443 --token=<hidden>
```

A token is a form of a password. Do not share your API token. To reveal your token, press the copy to clipboard button and then paste the clipboard contents.

After you login to your account you will get a list of projects that you can switch between:

```
oc project <project-name>
```

If you do not have any existing projects, you can create one:

```
oc new-project <project-name>
```

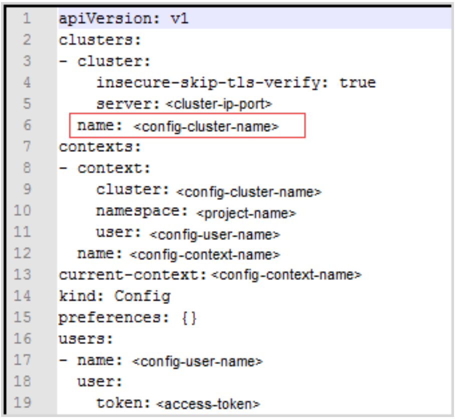
To show a high level overview of the current project:

```
oc status
```

For other information about the command line tools, check the CLI Reference and Basic CLI Operations.

https://docs.openshift.org/latest/cli_reference/get_started_cli.html

   3 On your local platform, browse to open this configuration file. You may see the context similar to the following. Record ***<config-cluster-name>*** for later use.



```
apiVersion: v1
clusters:
- cluster:
    insecure-skip-tls-verify: true
    server: <cluster-ip-port>
  name: <config-cluster-name>
contexts:
- context:
    cluster: <config-cluster-name>
    namespace: <project-name>
    user: <config-user-name>
  name: <config-context-name>
current-context: <config-context-name>
kind: Config
preferences: {}
users:
- name: <config-user-name>
  user:
    token: <access-token>
```

2 The token generated in step 1 will be expired after 4 hours, however Foglight for Container Management needs a permanent Kubernetes credential. So you need to create a service account with “**cluster-admin**” role, and then get the authorization code (not expired) of this service account to generate our permanent Kubernetes credential.

   1 Run the command *`oc project`* ***`<project-name>`***.

   2 Run the command *`oc create serviceaccount`* ***`<service-account-name>`***.

   You can check if your service account has been created successfully using the command: *`kubectl get serviceaccounts`*

   3 Run the command *`oc serviceaccounts get-token`* ***`<service-account-name>`***. Then you will get a token <*service-account-token*> like below. Record this token for later use.

   `"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6Im9zLWFkb`

```
WluLXRva2VuLWY0a2ZsIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLW
FjY291bnQubmFtZSI6Im9zLWFkbWluIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9
zZXJ2aWNlLWFjY291bnQudWlkIjoiODMzNGU0NTQtNzQ1Yy0xMWU4LWFmNmEtMDA1MDU2YjY3
NDFhIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6b3MtYWRtaW4ifQ.RW
H_AoXy2U1elkHN_Bs9IR1xo0zNCJlwcY0h3zuQnrkOFi8gVpX1I77uhAPp7oIjPqDSWkUAN9F
6mP_tNdGwJsqRmHYEMOtCLnnIM61BYxIcABvwr66aOZ3Gn0D7EM5M_7XgKDCl6ON3W5NaH0D8
DpVTYqxkQ49u3qt4gqrcjVCaSsDNWlgGxY4KOIDrUbKkdgaRKzeD9o4Bv9VbYICqyxwoUebku
JAcHiXGIcSe-ozS_zroPi1tT5HW-RY0Pn3Fp3zBnydiokna0-mXot5lqoYc-
R6E1U9YSrAOhWm9Q8ipiut6OczXbmLPM4DYve6dmHi_j5FquCqhod-QlA7aPw"
```

4. Run the following command to grant your service account with the “**cluster-admin**” privilege: *kubectl create clusterrolebinding* ***<cluster-role-binding-name>*** *--clusterrole=cluster-admin --serviceaccount=default:****<service-account-name>***.

3 Generate a permanent Kubernetes configuration file and save it under *<USER_HOME>/.kube/config file/credential*.

1. Open and edit the intermediate configuration file.
2. Use kubectl to add user credentials, create new context, in the end change the existing contexts to the ones that you added in step 2. For example,

   kubectl config set-credentials <credential-name> --token=<service-account-token>

   kubectl config set-context <new-context-name> --cluster=<config-cluster-name> --user=<credential-name> --namespace=<project-name>

   kubectl config use-context <new-context-name>
3. Save the current Kubernetes configuration file.

Enabling Heapster service in monitored environment

There are various approaches to enable Heapster on your Kubernetes cluster. Visit Heapster official website to determine the approach that you are going to deploy your Heapster service, or you can follow instructions in https://github.com/foglight/container to deploy your service.

Some of the cloud platform Kubernetes service has enabled Heapster service for the cluster. If you have connected to the cluster, run the following command to check: *kubectl cluster-info*

Docker Swarm Agent

Each Docker Swarm Agent monitors the assets in one docker host. Docker Remote API needs to be enabled for the Docker Swarm Agent collecting data from the docker host. If TLS is enabled to secure the Docker Remote API, credential for Docker Swarm Agent needs to be prepared. Complete the following prerequisites before create agent.

- Preparing Docker Swarm Agent credentials
- Enabling Docker Remote API for monitored docker host
- Uploading Docker Swarm Agent credentials

Preparing Docker Swarm Agent credentials

If TLS enabled to secure Docker Remote API, then complete the following guide to get the credentials for Docker Swarm Agent for the docker host. Otherwise, continue with Enabling Docker Remote API for monitored docker host on page 16

Refer to the official guide to generate the keys. Be aware that, during generating the keys, the Foglight Agent Manager host address should be in the allow access list.

Docker Swarm Agent needs following credentials, you can get them when you finish the official guide.

- CA Public Key (ca.pem in official guide)
- Client Public Key (cert.pem in official guide)
- Client Private Key (key.pem in official guide)

Enabling Docker Remote API for monitored docker host

Change `ExecStart` in docker service startup script as below.

Non-TLS secured

```
ExecStart=/usr/bin/dockerd –H tcp://0.0.0.0:2375 –H
unix:///var/run/docker.sock
```

NOTE: Access should be allowed to the TCP port 2375

TLS secured

If TLS enabled, complete Preparing Docker Swarm Agent credentials on page 15 first, then you will get the ca.pem, server-cert.pem and server-key.pem mentioned in the official guide.

```
ExecStart=/usr/bin/dockerd --tlsverify --tlscacert=ca.pem --tlscert=server-
cert.pem --tlskey=server-key.pem -H tcp://0.0.0.0:2375 -H
unix:///var/run/docker.sock
```

NOTE: Access should be allowed to the TCP port 2375

Then restart docker service.

Uploading Docker Swarm Agent credentials

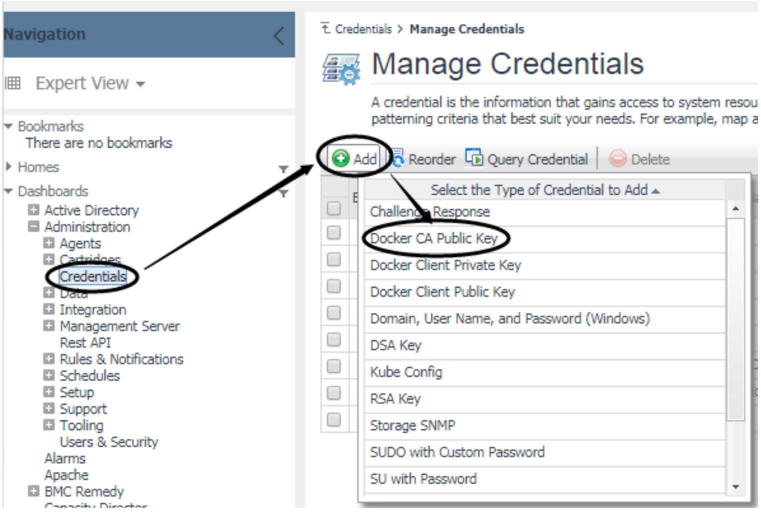
If TLS is enabled to secure Docker Remote API, go through this section to upload the credential for Docker Swarm Agent. Otherwise, skip this section.

When complete Preparing Docker Swarm Agent credentials on page 15, following credentials should be generated.

- CA Public Key
- Client Public Key
- Client Private Key

These are the credentials for Docker Swarm Agent, complete the following steps to upload the credentials.

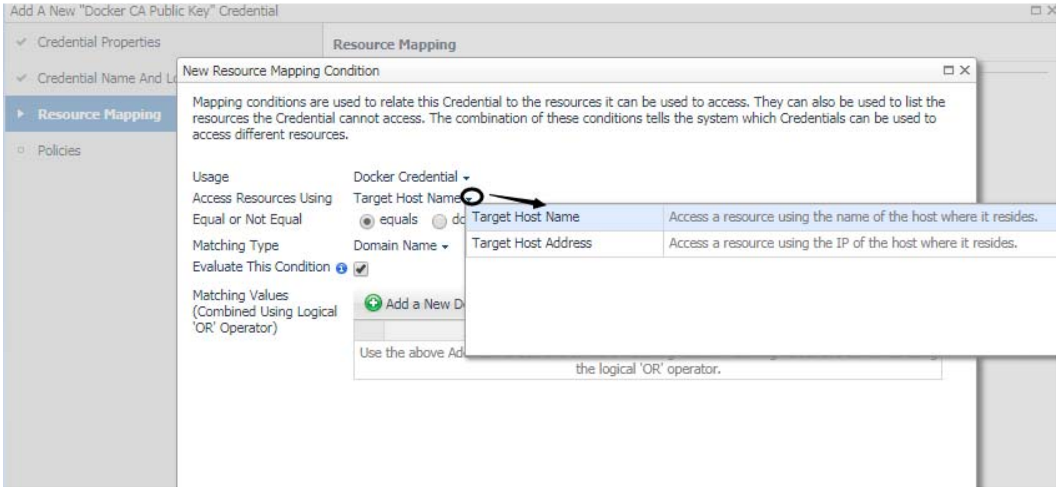
On the **Administration > Credentials > Manage Credentials** dashboard, click **Add**, and then select Docker CA Public Key or Docker Client Public Key or Docker Client Private Key to upload related credentials. Take Docker CA Public Key as an example.



A credential is the information that gains access to system resou
patterning criteria that best suit your needs. For example, map a

In the **Add a New "Docker CA Public Key" Credential** dialog box, specify the following values:

- Credential Properties: Click **Load from file** to import Docker CA Public Key, and then click **Next**.
- Credential Name And Lockbox: Specify a unique name for this credential, and then click **Next**.
- Resource Mapping: Click **Add**. In the **New Resource Mapping Condition** dialog box, choose Target Host Name or Target Host Address for the monitored docker host.



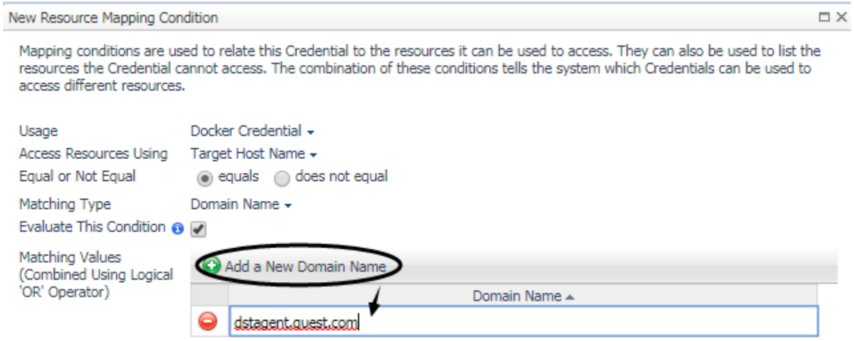
Mapping conditions are used to relate this Credential to the resources it can be used to access. They can also be used to list the resources the Credential cannot access. The combination of these conditions tells the system which Credentials can be used to access different resources.

Usage: Docker Credential
Access Resources Using: Target Host Name
Equal or Not Equal: ● equals ○ do
Matching Type: Domain Name
Evaluate This Condition: ☑
Matching Values (Combined Using Logical 'OR' Operator)

Target Host Name	Access a resource using the name of the host where it resides.
Target Host Address	Access a resource using the IP of the host where it resides.

Use the above Ad ... the logical 'OR' operator.

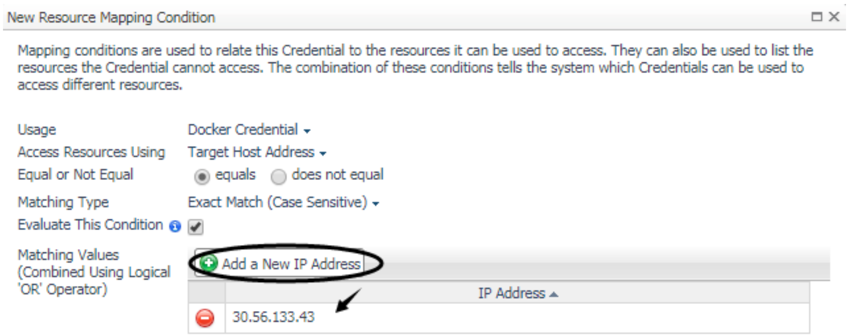
If choose **Target Host Name**, then enter the host name of the monitored docker host.



Mapping conditions are used to relate this Credential to the resources it can be used to access. They can also be used to list the resources the Credential cannot access. The combination of these conditions tells the system which Credentials can be used to access different resources.

Usage: Docker Credential
Access Resources Using: Target Host Name
Equal or Not Equal: ● equals ○ does not equal
Matching Type: Domain Name
Evaluate This Condition: ☑
Matching Values (Combined Using Logical 'OR' Operator)
Domain Name: dstagent.quest.com

If choose Target Host Address, then enter the IP address of the monitored docker host.



New Resource Mapping Condition

Mapping conditions are used to relate this Credential to the resources it can be used to access. They can also be used to list the resources the Credential cannot access. The combination of these conditions tells the system which Credentials can be used to access different resources.

Usage: Docker Credential

Access Resources Using: Target Host Address

Equal or Not Equal: ● equals ○ does not equal

Matching Type: Exact Match (Case Sensitive)

Evaluate This Condition: ☑

Matching Values (Combined Using Logical 'OR' Operator)

IP Address
30.56.133.43

Then click **Add** to finish editing **New Resource Mapping Condition** and back to **Resource Mapping**. Then click **Finish**.

Then **Docker CA Public Key** has been uploaded and mapped to the docker host. To monitor this docker host, **Docker Client Public Key** and **Docker Client Private Key** also need to be uploaded following the above steps.

Creating and Activating Agent

Foglight for Container Management supports Kubernetes Agent and Docker Swarm Agent.

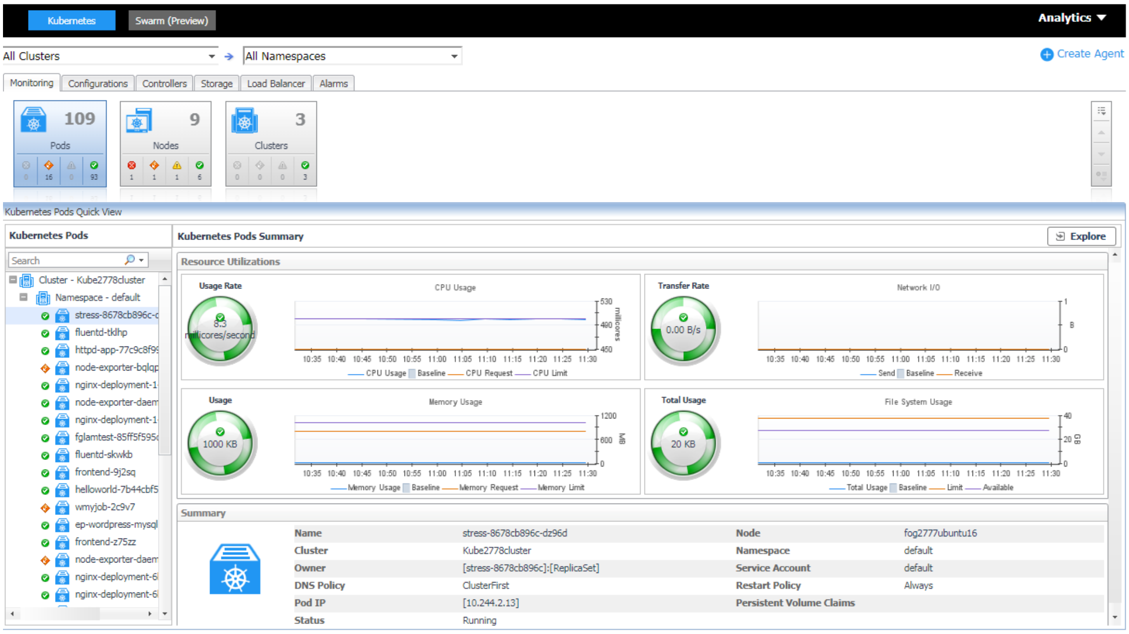
- Creating and Activating a Kubernetes Agent
- Creating and Activating a Docker Swarm Agent

Creating and Activating a Kubernetes Agent

To create a Kubernetes agent on a monitored host:

1 Log in to the Foglight browser interface and make sure the left navigation panel is open.

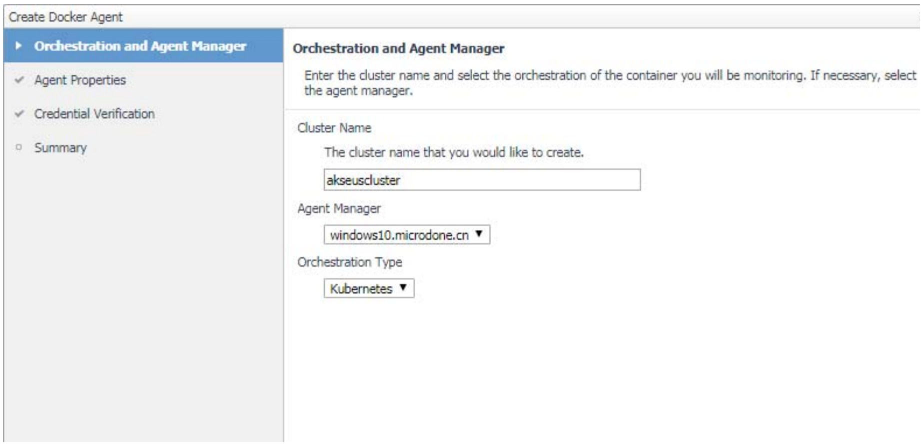
2 On the navigation panel, from **Standard View** click **Container Environment** or from **Expert View** click **Dashboards > Container**. Then the Container dashboard will display as below.



Name	stress-8678cb896c-dz96d	Node	fog2777ubuntu16
Cluster	Kube2778cluster	Namespace	default
Owner	[stress-8678cb896c]:[ReplicaSet]	Service Account	default
DNS Policy	ClusterFirst	Restart Policy	Always
Pod IP	[10.244.2.13]	Persistent Volume Claims	
Status	Running		

3 In the Container dashboard, click **Create Agent** on the top right. The **Create Agent** wizard opens.

4 *Orchestration and Agent Manager: specify the following values, and then click **Next**.*



Orchestration and Agent Manager

Enter the cluster name and select the orchestration of the container you will be monitoring. If necessary, select the agent manager.

Cluster Name

The cluster name that you would like to create.

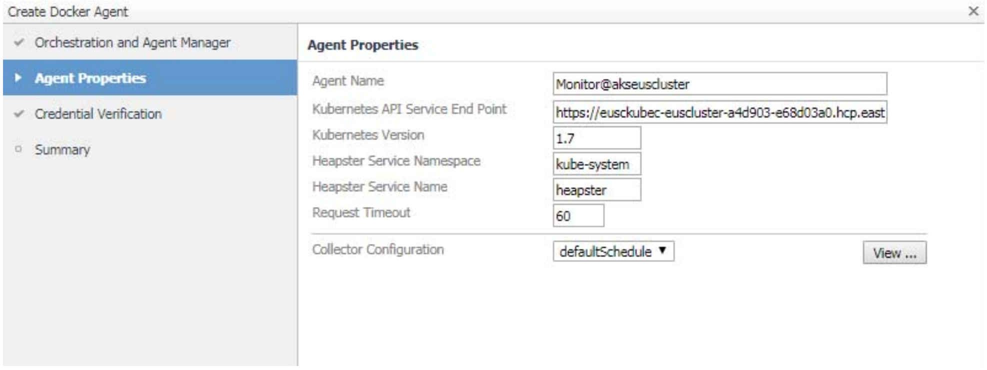
akseuscluster

Agent Manager: windows10.microdone.cn ▼

Orchestration Type: Kubernetes ▼

- Cluster Name: unique name for the monitored cluster.
- Agent Manager: select an Agent Manager which manages the agent.
- Orchestration Type: container orchestration.

5 *Agent Properties*



Agent Properties

Agent Name	Monitor@akseuscluster
Kubernetes API Service End Point	https://eusckubec-euscluster-a4d903-e68d03a0.hcp.east
Kubernetes Version	1.7
Heapster Service Namespace	kube-system
Heapster Service Name	heapster
Request Timeout	60
Collector Configuration	defaultSchedule ▼

- *Kubernetes API Service End Point*: Get this information from the *KubeConfig* file. For more information, see Enabling Heapster service in monitored environment on page 15.
- *Kubernetes Version: 1.7 by default.*

  NOTE: Only need to change for OpenShift clusters.

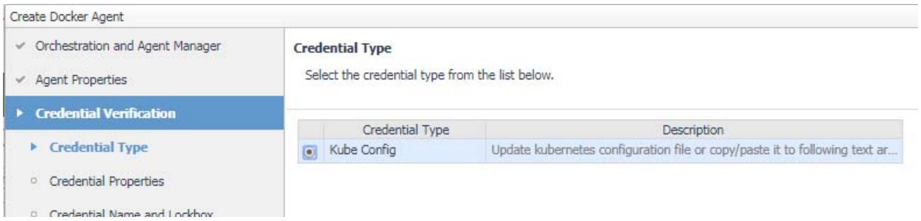
- *Heapster Service Namespace/Heapster Service Name*: Get both values from the Heapster service configuration. For more information, see Enabling Heapster service in monitored environment on page 15.

  If you deploy the Heapster service using:

  - HTTP: *Namespace of Heapster* should be <*heapster service namespace*> and *Heapster Service Name* should be <*heapster service name*>.
  - HTTPS: *Namespace of Heapster* should be <*heapster service namespace*> and *Heapster Service Name* should be *https:<heapster service namespace>:*.

6 *Credential Verification*

- *No credential: click **Next**.*
- Add cluster to a new credential
  - Credential Type: choose **Kube Config**, and click **Next.**

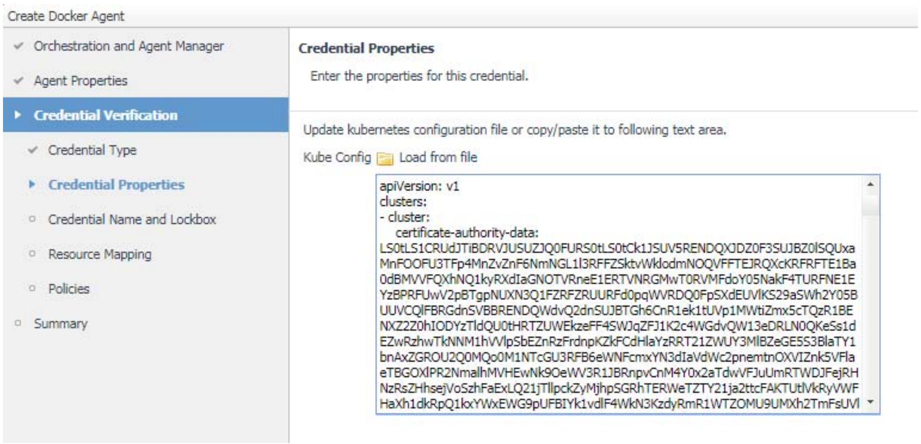


Credential Type

Select the credential type from the list below.

	Credential Type	Description
●	Kube Config	Update kubernetes configuration file or copy/paste it to following text ar...

- Credential Properties: click **Load from file** to upload the credential, and click **Next**.



Credential Properties

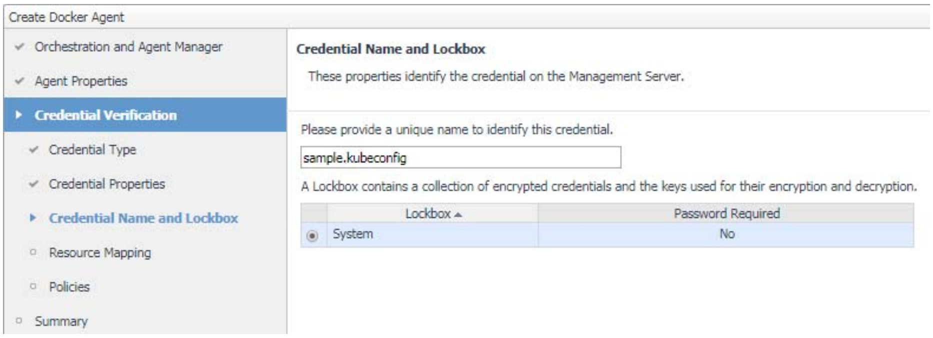
Enter the properties for this credential.

Update kubernetes configuration file or copy/paste it to following text area.

Kube Config Load from file

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV5RENDQXJDZ0F3SUJBZ0lSQUxa
MnFOOFU3TFp4MnZvZnF6NmNGL1l3RFFZSktvWklodmNOQVFFTEJRQXcKRFRFTE1Ba
0dBMVVFQXhNQ1kyRXdIaGNOTVRneE1ERTVNRGMwT0RVMFdoY05NakF4TURFNE1E
YzBPRFUwV2pBTgpNUXN3Q1FZRFZRUURFd0pqWVRDQ0FpSXdEUVlKS29aSWh2Y05B
UUVCQlFBRGdnSVBBRENDQWdvQ2dnSUJBTGh6CnR1ek1tUVp1MWtiZmx5cTQzR1BE
NXZ2Z0hIODYzTldQU0tHRTZUWEkzeFF4SWJqZFJ1K2c4WGdvQW13eDRLN0QKeSs1d
EZwRzhwTkNNM1hVVlpSbEZnRzFrdnpKZkFCdHlaYzRRT21ZWUY3MlBZeGE5S3BlaTY1
bnAxZGROU2Q0MQo0M1NTcGU3RFB6eWNFcmxYN3dIaVdWc2pnemtnOXVIZnk5VFla
eTBGOXlPR2NmalhMVHEwNk9OeWV3R1JBRnpvCnM4Y0x2aTdwVFJuUmRTWDJFejRH
NzRsZHhsejVoSzhFaExLQ21jTllpckZyMjhpSGRhTERWeTZTY21ja2ttcFAKTUtlVkRyVWF
HaXh1dkRpQ1kxYWxEWG9pUFBIYk1vdlF4WkN3KzdyRmR1WTZOMU9UMXh2TmFsUVl
```

- Credential Name and Lockbox: give a name for the credential, choose a lockbox, then click **Next**.



Credential Name and Lockbox

These properties identify the credential on the Management Server.

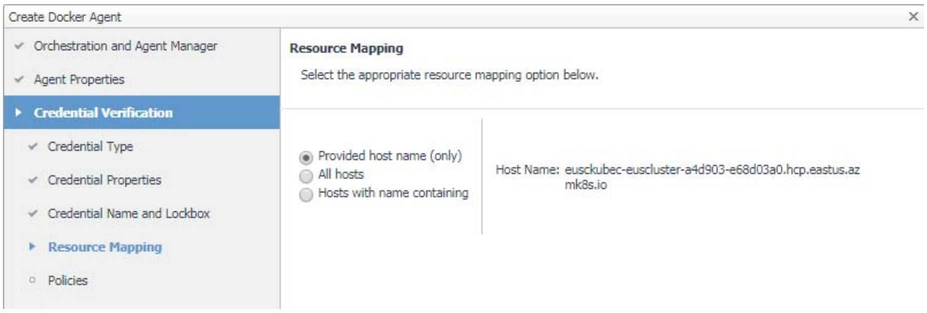
Please provide a unique name to identify this credential.

sample.kubeconfig

A Lockbox contains a collection of encrypted credentials and the keys used for their encryption and decryption.

	Lockbox ▲	Password Required
●	System	No

- Resource Mapping: click **Next**.



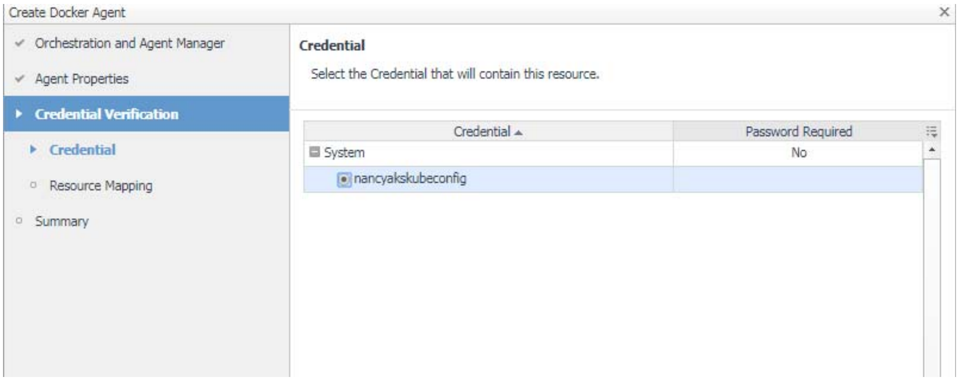
Resource Mapping

Select the appropriate resource mapping option below.

● Provided host name (only)
○ All hosts
○ Hosts with name containing

Host Name: euscubec-euscluster-a4d903-e68d03a0.hcp.eastus.azmk8s.io

- Add cluster to an existing credential
  - Credential: choose credential, then click **Next**.

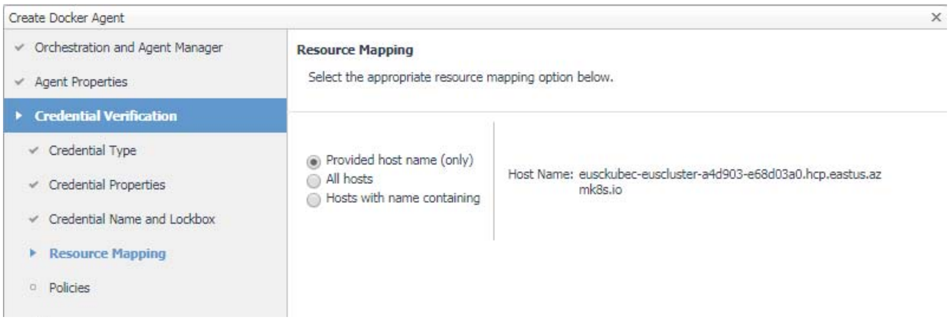


Credential

Select the Credential that will contain this resource.

Credential ▲	Password Required
⊟ System	No
● nancyakskubeconfig	

- Resource Mapping: click **Next**.



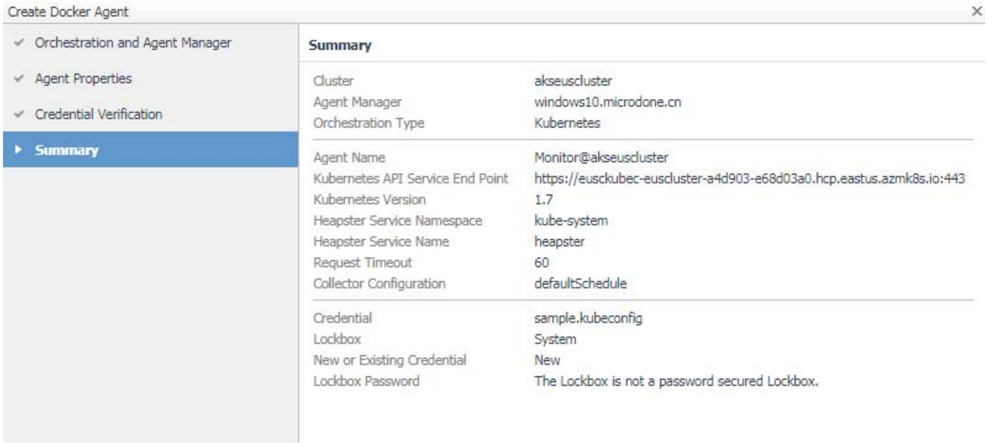
Resource Mapping

Select the appropriate resource mapping option below.

● Provided host name (only)
○ All hosts
○ Hosts with name containing

Host Name: euscubec-euscluster-a4d903-e68d03a0.hcp.eastus.azmk8s.io

7 *Summary: click **Finish**.*



Summary

Cluster	akseuscluster
Agent Manager	windows10.microdone.cn
Orchestration Type	Kubernetes
Agent Name	Monitor@akseuscluster
Kubernetes API Service End Point	https://eusckubec-euscluster-a4d903-e68d03a0.hcp.eastus.azmk8s.io:443
Kubernetes Version	1.7
Heapster Service Namespace	kube-system
Heapster Service Name	heapster
Request Timeout	60
Collector Configuration	defaultSchedule
Credential	sample.kubeconfig
Lockbox	System
New or Existing Credential	New
Lockbox Password	The Lockbox is not a password secured Lockbox.

8 Then the agent will be created and activated automatically.

Creating and Activating a Docker Swarm Agent

Each Docker Swarm Agent monitored one docker host. If the docker host belongs to a Docker Swarm cluster, it will be considered as a manager/worker node. Otherwise, it will be considered to be a standalone docker host.

> **NOTE:** For a Docker Swarm cluster, you should create one Docker Swarm Agent for one host in the cluster, and if you want to monitor the whole cluster environment, you need to create all the Docker Swarm Agents for all the hosts in the cluster.

To create a Docker Swarm agent on a monitored host:

1 Login in to the Foglight browser interface and make sure the left navigation panel is open.

2 On the navigation panel, under **Dashboards**, click **Administration > Agents > Agent Status**.

The **Agent Status** dashboard opens.

3 In the **Agent Status** dashboard, click **Create Agent**.

The **Create Agent** wizard opens.

4 *Host Selector*: Select the monitored host that you want to monitor with the Docker Swarm agent instance that you are about to create, and then click **Next**.

> **NOTE:** In order to select the host, the Foglight Agent Manager must be installed and running on the monitored host.

5 *Agent Type and Instance Name*: Specify the following values, and then click **Next**.

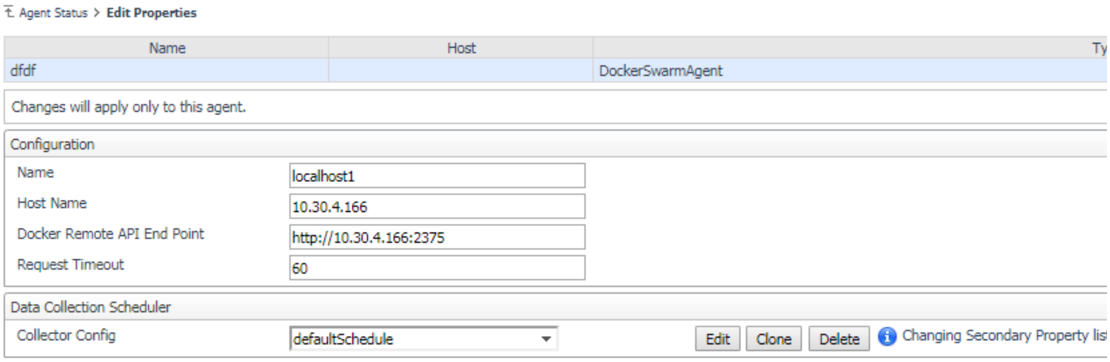
- *Agent Type*: Select DockerSwarmAgent from the agent type list.
- *Agent Name*: Specify the name of the agent instance that you are about to create using either of the following approaches:
  - *Generic Name*: This option is selected by default. A generic name is a combination of the host name and the agent type and uses the following syntax: `agent_type@host_name`.
  - *Specify Name*: Type that name in the *Name* field. For example, `MyAgent`.

6 On the **Summary** page, review the choices you have made, and then click ***Finish***.

The *Agents* table refreshes automatically, showing the new Docker Swarm Agent.

7 On the *Agents* table, select the Docker Swarm Agent that you create, click **Edit Properties**, and then click **Modify the private properties for this agent**.

8 In the *Agents* properties view, check if the following values have been configured based upon your environment:



Agent Status > Edit Properties

Name	Host	Ty
dfdf		DockerSwarmAgent

Changes will apply only to this agent.

Configuration

Name	localhost1
Host Name	10.30.4.166
Docker Remote API End Point	http://10.30.4.166:2375
Request Timeout	60

Data Collection Scheduler

Collector Config: defaultSchedule

- *Name*: give a name to the monitored docker host, it should be unique.
- *Host Name*: IP address or host name of the monitored docker host.
- *Docker Remote API End Point*: Docker Remote API endpoint of the monitored docker host. For more information, see Enabling Docker Remote API for monitored docker host on page 16.

9 Return back to the *Agents* table, select the above property changed Docker Swarm Agent, and then click **Activate**.

The new Docker Swarm Agent is created and data will be shown on the **Monitoring** tab after a few minutes.

Configuring data collection interval

The default data collection interval of agents is set to 5 minutes by default. Foglight for Container Management enables you to change this collection interval as needed.

> **NOTE:** Changing the data collection interval will take effect for all Kubernetes agents and Docker Swarm agents.

To configure the data collection interval:

1 On the navigation panel, under **Dashboards**, select **Administration > Agents > Agent Status**.

2 On the *Agent Status* dashboard, select the Kubernetes agent that you use to monitoring the container environment, and then click **Edit Properties**.

3 In the *Edit Properties* dashboard, click **Edit** next to the *Collector Config* field.

4 In the KubernetesAgent or DockerSwarmAgent Collector Config dialog box, change the following values, as needed:

- *Inventory Collector*: Specifies the interval for collecting components.
- *Metrics Collector*: Specifies the interval for collecting metrics.

5 Click **Save**.

2

Using Foglight for Container Management

Kubernetes

Monitoring Kubernetes Pods

A pod contains one or multiple containers, such as Docker containers, which contains storage/network and the specification about how to run the containers. The *Kubernetes Pods Quick View*, which appears after clicking

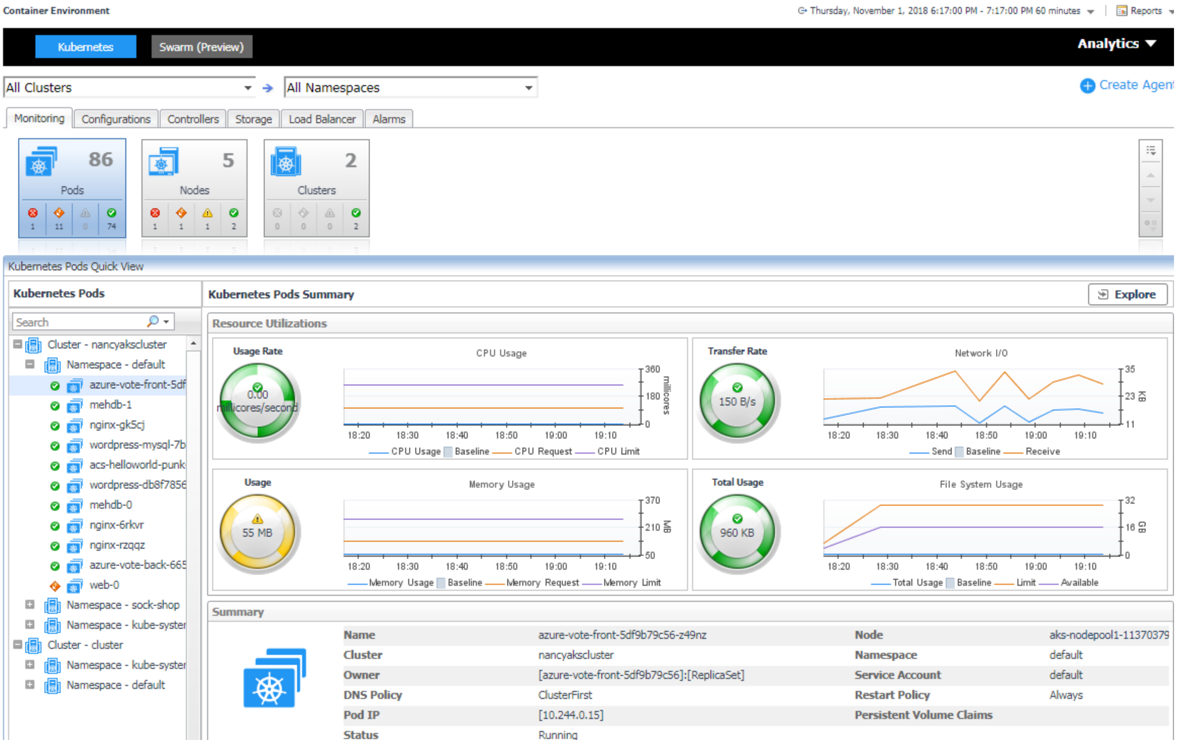
Monitoring > Pods, shows the data collected about the selected clusters and namespaces. This view consists of the following two panes:

- The **Kubernetes Pods** tree view, which appears on the left of *Kubernetes Pods Quick View*, lists the pods existing in the monitored Kubernetes environment.
- The Kubernetes Pods Summary view, which appears on the right after you select an individual pod in the **Kubernetes Pods** tree view.

Kubernetes Pods Summary view

The **Kubernetes Pods Summary** view appears on the right when you select a cluster in the **Kubernetes Pods** tree view.

Figure 2. Kubernetes Pods Summary view



Name	azure-vote-front-5df9b79c56-z49nz	**Node**	aks-nodepool1-11370379
Cluster	nancyakscluster	**Namespace**	default
Owner	[azure-vote-front-5df9b79c56]:[ReplicaSet]	**Service Account**	default
DNS Policy	ClusterFirst	**Restart Policy**	Always
Pod IP	[10.244.0.15]	**Persistent Volume Claims**	
Status	Running		

The **Kubernetes Pods Summary** view displays the following data:

- *Resource Utilizations*: The resource utilization for the selected Kubernetes Pod over a selected period of time, which includes the following:
  - *CPU Usage*: Shows the CPU utilization summary for the selected Kubernetes Pod based on its total capacity during a selected time period.
  - *Transfer Rate*: Shows the network utilization summary for the selected Kubernetes Pod, including the average rate of network throughput, and the amounts of data sent to and received from the network.
  - *Memory Usage*: Shows the physical memory utilization summary for the selected Kubernetes Pod, broken into the amounts of memory that is swapped to disk, actively used, and allocated, all during a selected time period.
  - File System Usage: Shows the file system resource utilization summary for the selected Kubernetes Pod, including the available/total/limited file system resource.
- *Summary*: Displays the detailed information about the selected Kubernetes Pod, including *Name, Node, Cluster, Namespace, Owner, Pod IP, Service Account, DNS Policy, Restart Policy,* and *Status*.

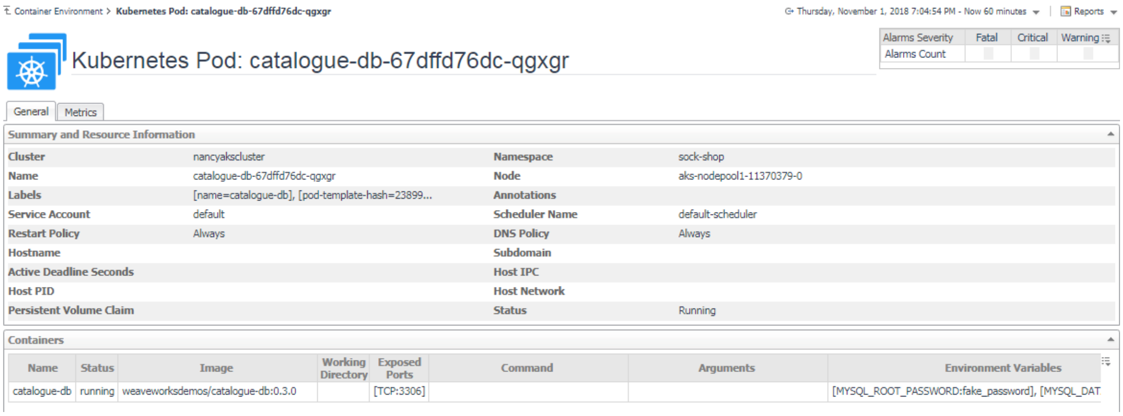
Click **Explore** on the upper right of the **Kubernetes Pods Summary** view to open the Pods Explorer view, which shows more detailed information about this Kubernetes cluster.

Pods Explorer view

The *Pods Explorer* view opens when you click **Explore** in the Kubernetes Pods Summary view, which includes the following tabs:

- *General tab:* The *General* tab displays the overall information of the selected Kubernetes Pod over a selected period of time, including the *Summary and Resource Information* table, the *Containers* table, and the *Init Containers* table. For more information, see Pod metrics *on page 48.*

Figure 3. Kubernetes Pods Explorer view General Tab



Container Environment > **Kubernetes Pod: catalogue-db-67dffd76dc-qgxgr**

Thursday, November 1, 2018 7:04:54 PM - Now 60 minutes

Reports

Alarms Severity	Fatal	Critical	Warning
Alarms Count			

Kubernetes Pod: catalogue-db-67dffd76dc-qgxgr

General | Metrics

Summary and Resource Information

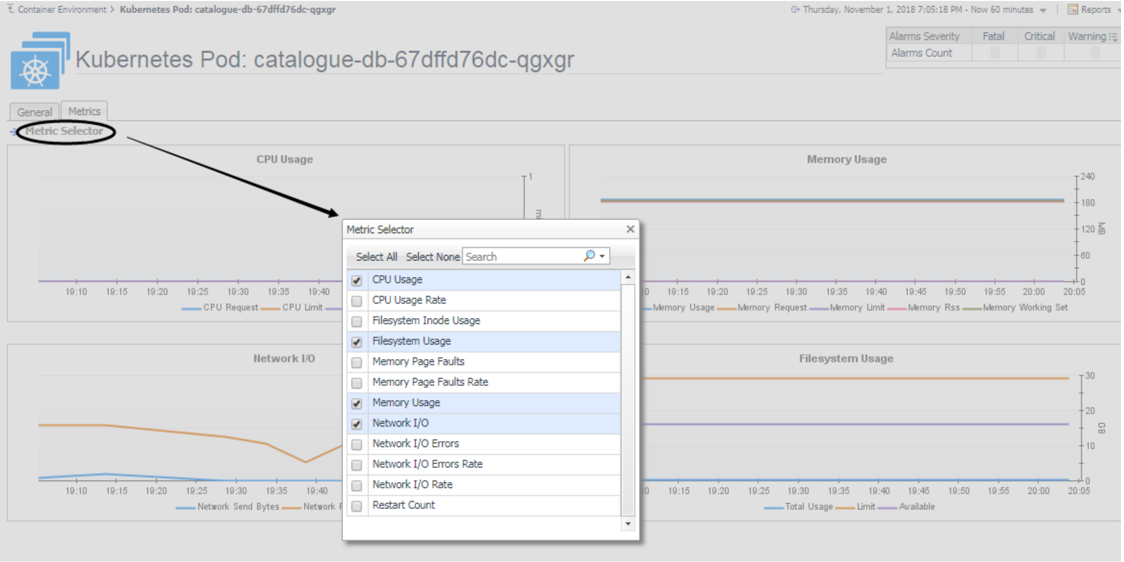
Cluster	nancyakscluster	**Namespace**	sock-shop
Name	catalogue-db-67dffd76dc-qgxgr	**Node**	aks-nodepool1-11370379-0
Labels	[name=catalogue-db], [pod-template-hash=23899...	**Annotations**	
Service Account	default	**Scheduler Name**	default-scheduler
Restart Policy	Always	**DNS Policy**	Always
Hostname		**Subdomain**	
Active Deadline Seconds		**Host IPC**	
Host PID		**Host Network**	
Persistent Volume Claim		**Status**	Running

Containers

Name	Status	Image	Working Directory	Exposed Ports	Command	Arguments	Environment Variables
catalogue-db	running	weaveworksdemos/catalogue-db:0.3.0		[TCP:3306]			[MYSQL_ROOT_PASSWORD:fake_password], [MYSQL_DAT

- *Metrics tab:* The *Metrics* tab displays a *Metric Selector* allowing you to choose the metrics to be plotted on this dashboard. Charts of *CPU Usage, Memory Usage, and Network I/O* are presented by default.

Figure 4. Kubernetes Pods Explorer view Metrics Tab



Container Environment > **Kubernetes Pod: catalogue-db-67dffd76dc-qgxgr**

Thursday, November 1, 2018 7:05:18 PM - Now 60 minutes

Reports

Alarms Severity	Fatal	Critical	Warning
Alarms Count			

Kubernetes Pod: catalogue-db-67dffd76dc-qgxgr

General | Metrics

Metric Selector

Metric Selector

Select All Select None Search

- ☑ CPU Usage
- □ CPU Usage Rate
- □ Filesystem Inode Usage
- ☑ Filesystem Usage
- □ Memory Page Faults
- □ Memory Page Faults Rate
- ☑ Memory Usage
- ☑ Network I/O
- □ Network I/O Errors
- □ Network I/O Errors Rate
- □ Network I/O Rate
- □ Restart Count

Monitoring Kubernetes Nodes

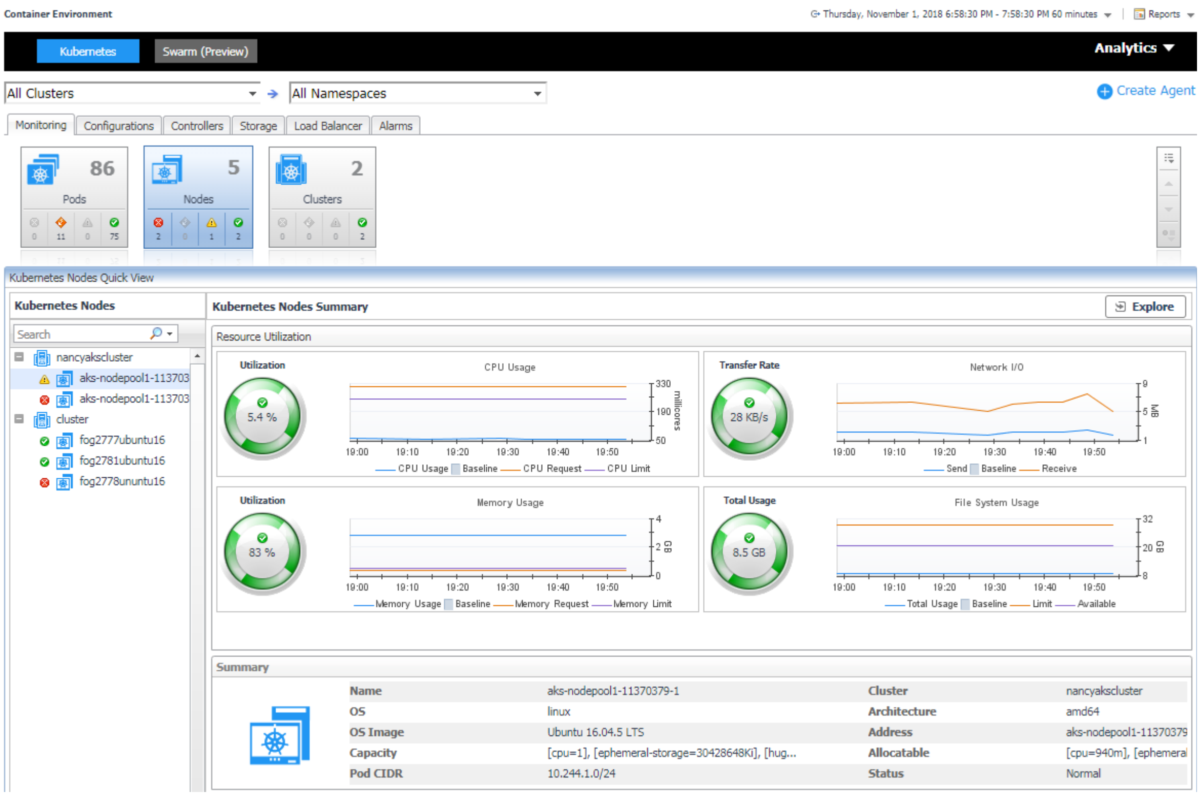
A node, previously known as a minion, is a worker machine in Kubernetes. A node may be a VM or physical machine, depending on the cluster. Each node has the services necessary to run pods and is managed by the master components. The *Kubernetes Nodes Quick View*, which appears after clicking **Monitoring > Nodes**, shows the data collected about the selected clusters and namespaces. This view consists of the following two panes:

- The **Kubernetes Nodes** tree view, which appears on the left of *Kubernetes Nodes Quick View*, lists the nodes existing in the monitored Kubernetes environment.
- The Kubernetes Nodes Summary view, which appears on the right after you select an individual node in the **Kubernetes Nodes** tree view.

Kubernetes Nodes Summary view

The **Kubernetes Nodes Summary** view appears on the right when you select a node in the **Kubernetes Nodes** tree view.

Figure 5. Kubernetes Nodes Summary view



All Clusters → All Namespaces

Name	aks-nodepool1-11370379-1	**Cluster**	nancyakscluster
OS	linux	**Architecture**	amd64
OS Image	Ubuntu 16.04.5 LTS	**Address**	aks-nodepool1-11370379
Capacity	[cpu=1], [ephemeral-storage=30428648Ki], [hug...	**Allocatable**	[cpu=940m], [ephemeral
Pod CIDR	10.244.1.0/24	**Status**	Normal

The **Kubernetes Nodes Summary** view displays the following data:

- *Resource Utilizations*: The resource utilization for the selected Kubernetes node over a selected period of time, which includes the following:
  - *CPU Utilization*: Shows the CPU utilization summary for the selected Kubernetes node based on its total capacity during a selected time period.
  - *Transfer Rate*: Shows the network utilization summary for the selected Kubernetes node, including the average rate of network throughput, and the amounts of data sent to and received from the network.
  - *Memory Utilization*: Shows the physical memory utilization summary for the selected Kubernetes node, broken into the amounts of memory that is swapped to disk, actively used, and allocated, all during a selected time period.
  - File System Usage: Shows the file system resource utilization summary for the selected Kubernetes Pod, including the available/total/limited file system resource.

- *Summary*: Displays the detailed information about the selected Kubernetes node, including *Name, Pod CIDR, OS, Architecture, OS Image, Address, Capacity, Allocatable,* and *Status*.

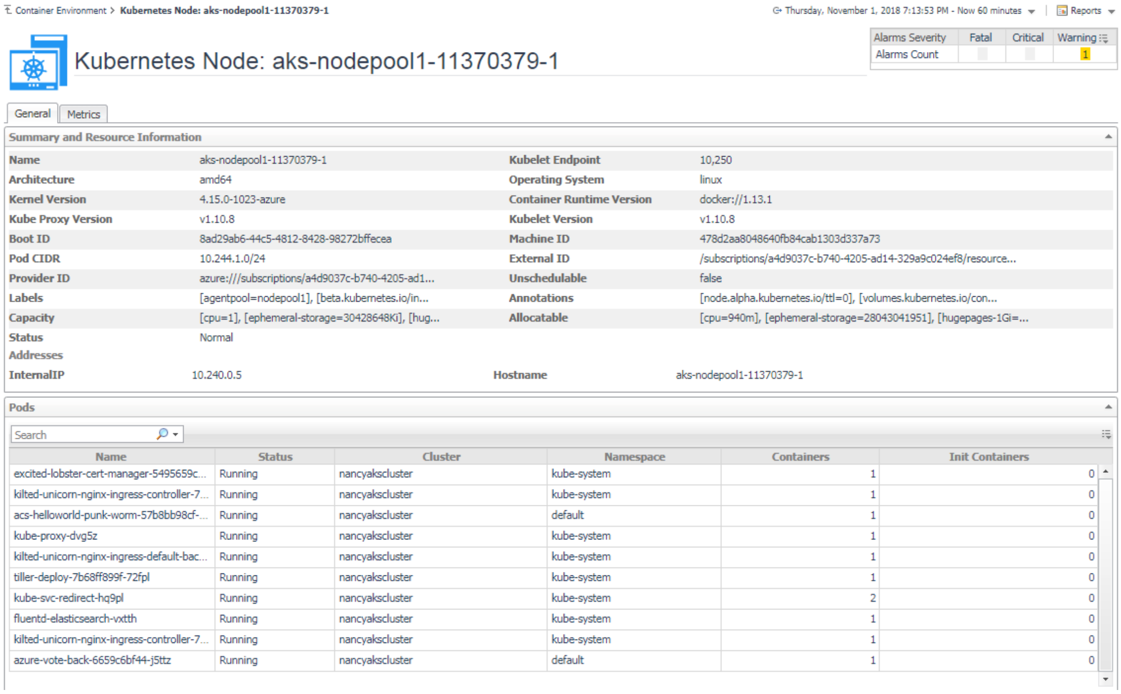
Click **Explore** on the upper right of the **Kubernetes Nodes Summary** view to open the Nodes Explorer view, which shows more detailed information about this Kubernetes node.

Nodes Explorer view

The *Nodes Explorer* view opens when you click **Explore** in the Kubernetes Nodes Summary view, which includes the following tabs:

- *General tab:* The *General* tab displays the overall information of the selected Kubernetes node over a selected period of time, including the *Summary and Resource Information* table and the *Pods* table. For more information, see Node metrics *on page 49.*

Figure 6. Kubernetes Nodes Explorer view General Tab



Container Environment > Kubernetes Node: aks-nodepool1-11370379-1
Thursday, November 1, 2018 7:13:53 PM - Now 60 minutes
Reports

Alarms Severity	Fatal	Critical	Warning
Alarms Count			1

Kubernetes Node: aks-nodepool1-11370379-1

Summary and Resource Information

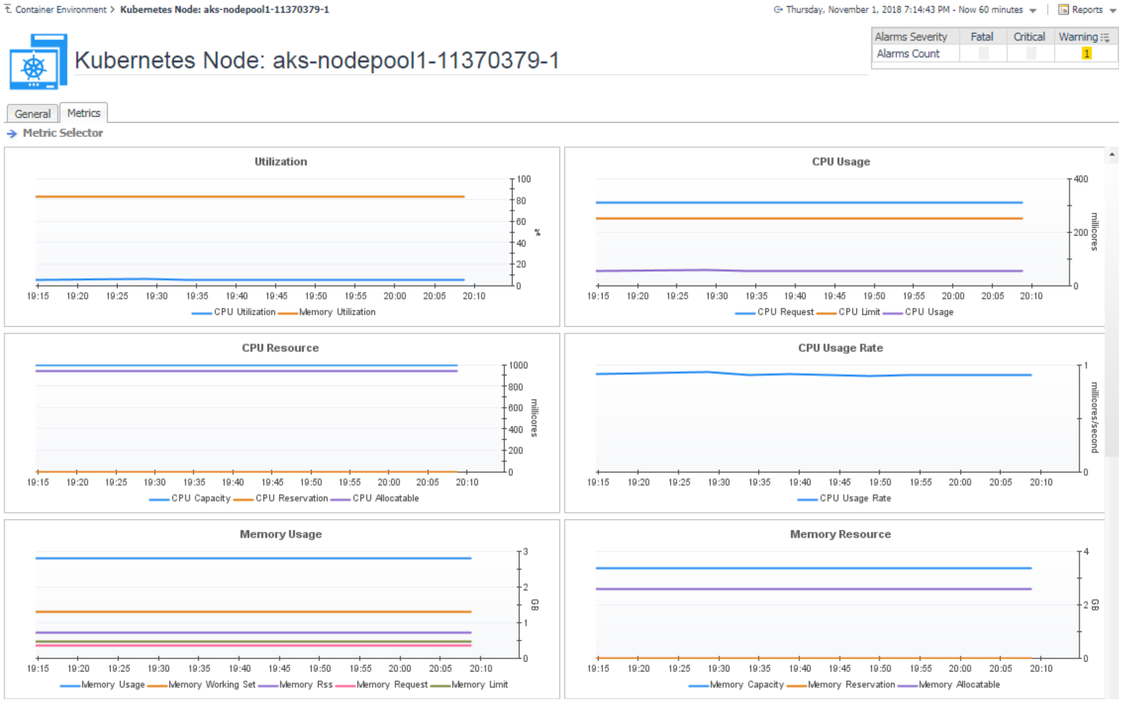
Name	aks-nodepool1-11370379-1	**Kubelet Endpoint**	10,250
Architecture	amd64	**Operating System**	linux
Kernel Version	4.15.0-1023-azure	**Container Runtime Version**	docker://1.13.1
Kube Proxy Version	v1.10.8	**Kubelet Version**	v1.10.8
Boot ID	8ad29ab6-44c5-4812-8428-98272bffecea	**Machine ID**	478d2aa8048640fb84cab1303d337a73
Pod CIDR	10.244.1.0/24	**External ID**	/subscriptions/a4d9037c-b740-4205-ad14-329a9c024ef8/resource...
Provider ID	azure:///subscriptions/a4d9037c-b740-4205-ad1...	**Unschedulable**	false
Labels	[agentpool=nodepool1], [beta.kubernetes.io/in...	**Annotations**	[node.alpha.kubernetes.io/ttl=0], [volumes.kubernetes.io/con...
Capacity	[cpu=1], [ephemeral-storage=30428648Ki], [hug...	**Allocatable**	[cpu=940m], [ephemeral-storage=28043041951], [hugepages-1Gi=...
Status	Normal		
Addresses			
InternalIP	10.240.0.5	**Hostname**	aks-nodepool1-11370379-1

Pods

Name	Status	Cluster	Namespace	Containers	Init Containers
excited-lobster-cert-manager-5495659c...	Running	nancyakscluster	kube-system	1	0
kilted-unicorn-nginx-ingress-controller-7...	Running	nancyakscluster	kube-system	1	0
acs-helloworld-punk-worm-57b8bb98cf-...	Running	nancyakscluster	default	1	0
kube-proxy-dvg5z	Running	nancyakscluster	kube-system	1	0
kilted-unicorn-nginx-ingress-default-bac...	Running	nancyakscluster	kube-system	1	0
tiller-deploy-7b68ff899f-72fpl	Running	nancyakscluster	kube-system	1	0
kube-svc-redirect-hq9pl	Running	nancyakscluster	kube-system	2	0
fluentd-elasticsearch-vxtth	Running	nancyakscluster	kube-system	1	0
kilted-unicorn-nginx-ingress-controller-7...	Running	nancyakscluster	kube-system	1	0
azure-vote-back-6659c6bf44-j5ttz	Running	nancyakscluster	default	1	0

- *Metrics tab:* The *Metrics* tab displays a *Metric Selector* allowing you to choose the metrics to be plotted on this dashboard. Charts of *CPU Usage, Utilization, Memory Usage, and Network I/O* are presented by default.

Figure 7. Kubernetes Nodes Explorer view Metrics Tab



Container Environment > **Kubernetes Node: aks-nodepool1-11370379-1**

Thursday, November 1, 2018 7:14:43 PM - Now 60 minutes

Kubernetes Node: aks-nodepool1-11370379-1

Alarms Severity	Fatal	Critical	Warning
Alarms Count			1

Monitoring Kubernetes Clusters

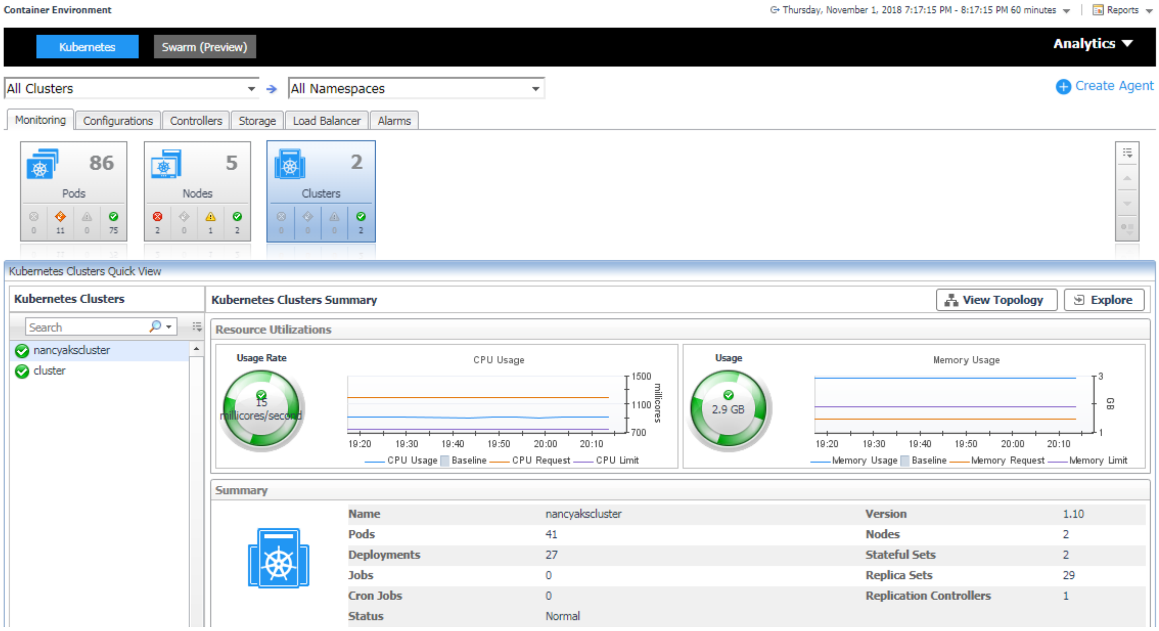
Kubernetes cluster is a group of kubernetes resources. There are two kinds of nodes inside a cluster, Kubernetes master and Kubernetes nodes. Kubernetes master is responsible for maintaining the desired state of your cluster which Kubernetes node is responsible to run your application and cloud workflows.The *Kubernetes Cluster Quick View*, which appears after clicking **Monitoring > Clusters**, shows the data collected about the selected clusters and namespaces. This view consists of the following two panes:

- The **Kubernetes Clusters** tree view, which appears on the left of *Kubernetes Clusters Quick View*, lists the clusters existing in the monitored Kubernetes environment.
- The Kubernetes Clusters Summary view, which appears on the right after you select an individual cluster in the **Kubernetes Clusters** tree view.

Kubernetes Clusters Summary view

The **Kubernetes Clusters Summary** view appears on the right when you select a node in the **Kubernetes Clusters** tree view.

Figure 8. Kubernetes Clusters Summary view



Name	nancyakscluster	**Version**	1.10
Pods	41	**Nodes**	2
Deployments	27	**Stateful Sets**	2
Jobs	0	**Replica Sets**	29
Cron Jobs	0	**Replication Controllers**	1
Status	Normal		

The **Kubernetes Clusters Summary** view displays the following data:

- *Resource Utilizations*: The resource utilization for the selected Kubernetes cluster over a selected period of time, which includes the following:
  - *Usage Rate*: Shows the CPU usage summary for the selected Kubernetes cluster based on its total capacity during a selected time period.
  - *Memory Usage*: Shows the physical memory utilization summary for the selected Kubernetes node, broken into the amounts of memory that is swapped to disk, actively used, and allocated, all during a selected time period.
- *Summary*: Displays the detailed information about the selected Kubernetes cluster, including *Name, Version, Pods, Nodes, Deployments, Stateful Sets, Jobs,* and *Replica Sets*.

Click **Explore** on the upper right of the **Kubernetes Clusters Summary** view to open the Cluster Explorer view, which shows more detailed information about this Kubernetes cluster.

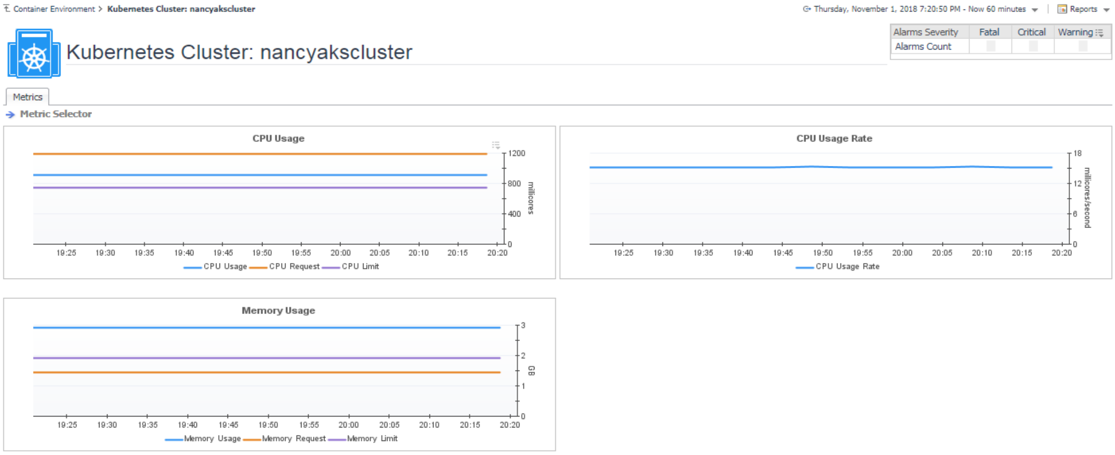
Click **View Topology** on the upper right of the **Kubernetes Clusters Summary** view to open the Cluster Topology view, which shows the topology graph from the application accessible aspect.

Cluster Explorer view

The *Cluster Explorer* view opens when you click **Explore** in the Pod metrics, which includes the following tabs:

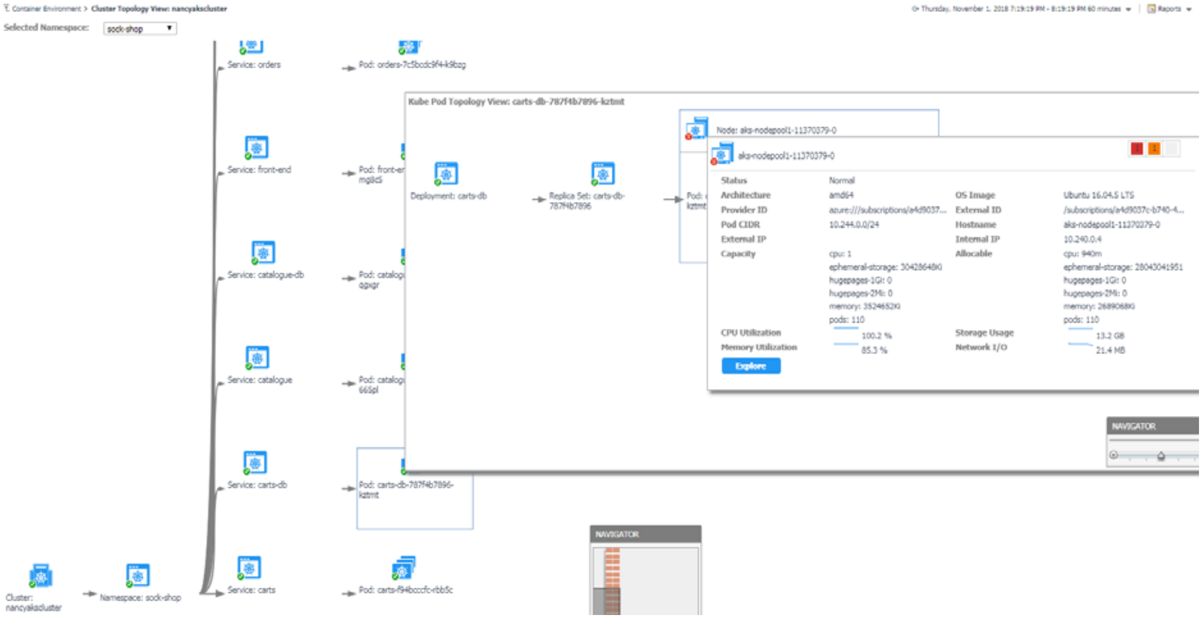
- *Metrics tab:* The *Metrics* tab displays a *Metric Selector* allowing you to choose the metrics to be plotted on this dashboard. Charts of *CPU Usage* and *Memory Usage* are presented by default.

Figure 9. Kubernetes Clusters Explorer view Metrics tab



Cluster Topology view

Figure 10. Kubernetes Clusters Topology view


Container Environment > Cluster Topology View: nancyakscluster
Thursday, November 1, 2018 7:19:19 PM - 8:19:19 PM 60 minutes
Reports
Selected Namespace:
sock-shop
Service: orders
Pod: orders-7c5bcdc9f4-k9bzg
Service: front-end
Pod: front-er mg8c6
Service: catalogue-db
Pod: catalog ggxgr
Service: catalogue
Pod: catalog 66Spl
Service: carts-db
Pod: carts-db-787f4b7896-kztmt
Service: carts
Pod: carts-f94bccfc-rbb5c
Cluster: nancyakscluster
Namespace: sock-shop
Kube Pod Topology View: carts-db-787f4b7896-kztmt
Deployment: carts-db
Replica Set: carts-db-787f4b7896
Pod: c kztmt
Node: aks-nodepool1-11370379-0
1
1
Explore
NAVIGATOR
NAVIGATOR

aks-nodepool1-11370379-0

Status	Normal		
Architecture	amd64	OS Image	Ubuntu 16.04.5 LTS
Provider ID	azure:///subscriptions/a4d9037...	External ID	/subscriptions/a4d9037c-b740-4...
Pod CIDR	10.244.0.0/24	Hostname	aks-nodepool1-11370379-0
External IP		Internal IP	10.240.0.4
Capacity	cpu: 1 ephemeral-storage: 30428648Ki hugepages-1Gi: 0 hugepages-2Mi: 0 memory: 3524652Ki pods: 110	Allocable	cpu: 940m ephemeral-storage: 28043041951 hugepages-1Gi: 0 hugepages-2Mi: 0 memory: 2689068Ki pods: 110
CPU Utilization	100.2 %	Storage Usage	13.2 GB
Memory Utilization	85.3 %	Network I/O	21.4 MB

The *topology view* visualizes the relationships between the objects from the pods accessible aspect in your environment through an interactive dependency map. The map illustrates how different components relate to each other, and the levels of the available resources available to them. Click on Pod, another sub topology view will popup to show the relationship from pods controller to storage for the selected Pod. Click other components or click the Pod inside the sub topology view, an information view will popup to show alarms, basic information, some metrics. From the information popup view of Pod, Node and Cluster, click the Explore button will navigate to the explorer view of the selected Pod/Node/Cluster. The **NAVIGATOR** in the bottom-right corner allows you to easily set the zoom level by dragging the slider into the appropriate position.

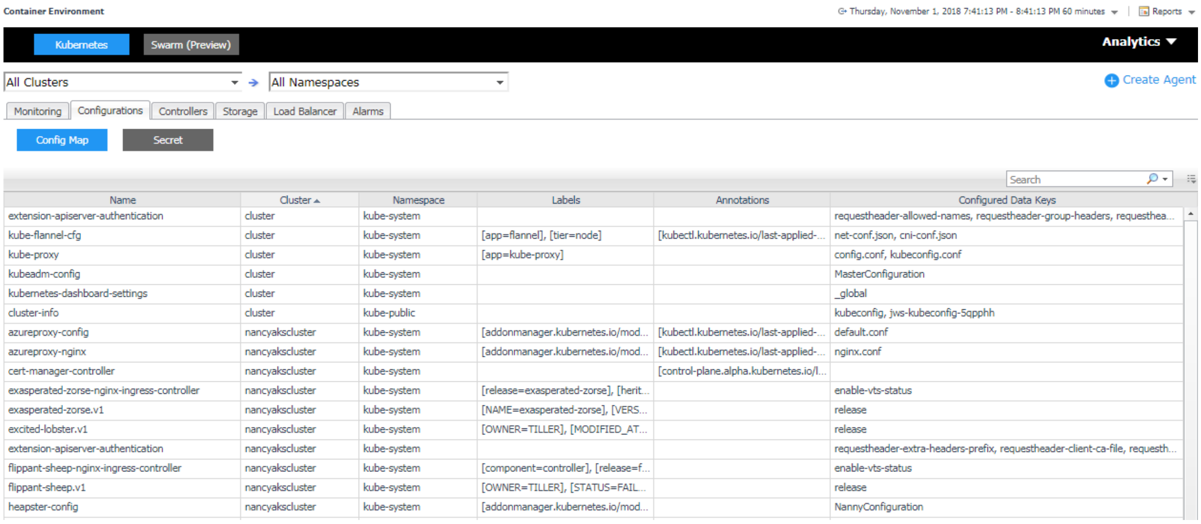
Monitoring Kubernetes Other Components

Kubernetes other components here including pods controllers, services, ingresses, persistent volumes, secrets and so on. All these components are grouped and displayed in tabs.

- Configurations
- Controllers
- Storage
- Load Balancer

Configurations

Figure 11. Kubernetes Configuration Dashboard



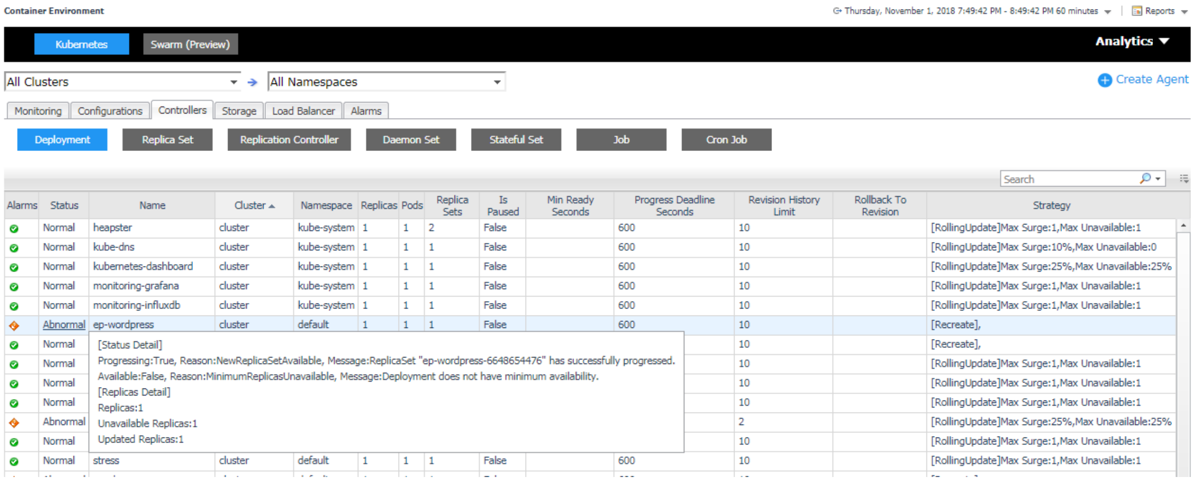
Name	Cluster	Namespace	Labels	Annotations	Configured Data Keys
extension-apiserver-authentication	cluster	kube-system			requestheader-allowed-names, requestheader-group-headers, requesthea...
kube-flannel-cfg	cluster	kube-system	[app=flannel], [tier=node]	[kubectl.kubernetes.io/last-applied-...	net-conf.json, cni-conf.json
kube-proxy	cluster	kube-system	[app=kube-proxy]		config.conf, kubeconfig.conf
kubeadm-config	cluster	kube-system			MasterConfiguration
kubernetes-dashboard-settings	cluster	kube-system			_global
cluster-info	cluster	kube-public			kubeconfig, jws-kubeconfig-5qpphh
azureproxy-config	nancyakscluster	kube-system	[addonmanager.kubernetes.io/mod...	[kubectl.kubernetes.io/last-applied-...	default.conf
azureproxy-nginx	nancyakscluster	kube-system	[addonmanager.kubernetes.io/mod...	[kubectl.kubernetes.io/last-applied-...	nginx.conf
cert-manager-controller	nancyakscluster	kube-system		[control-plane.alpha.kubernetes.io/l...	
exasperated-zorse-nginx-ingress-controller	nancyakscluster	kube-system	[release=exasperated-zorse], [herit...		enable-vts-status
exasperated-zorse.v1	nancyakscluster	kube-system	[NAME=exasperated-zorse], [VERS...		release
excited-lobster.v1	nancyakscluster	kube-system	[OWNER=TILLER], [MODIFIED_AT...		release
extension-apiserver-authentication	nancyakscluster	kube-system			requestheader-extra-headers-prefix, requestheader-client-ca-file, requesth...
flippant-sheep-nginx-ingress-controller	nancyakscluster	kube-system	[component=controller], [release=f...		enable-vts-status
flippant-sheep.v1	nancyakscluster	kube-system	[OWNER=TILLER], [STATUS=FAIL...		release
heapster-config	nancyakscluster	kube-system	[addonmanager.kubernetes.io/mod...		NannyConfiguration

The *Configurations* dashboard includes Kubernetes secret and config map.

- A Kubernetes secret is an object that contains a small amount of sensitive data such as a password, a token, or a key. Such information might otherwise be put in a Pod specification or in an image; putting it in a Secret object allows for more control over how it is used, and reduces the risk of accidental exposure.
- A Kubernetes config map binds configuration files, command-line arguments, environment variables, port numbers, and other configuration artifacts to your Pods' containers and system components at runtime. Config maps allow you to separate your configurations from your Pods and components, which helps keep your workloads portable, makes their configurations easier to change and manage, and prevents hardcoding configuration data to Pod specifications.

Controllers

Figure 12. Kubernetes Controllers Dashboard



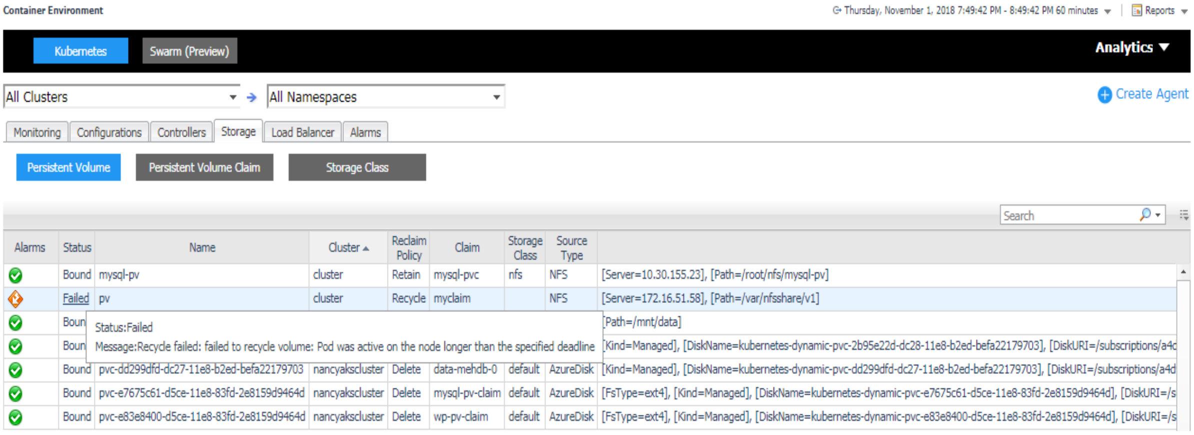
Alarms	Status	Name	Cluster ▲	Namespace	Replicas	Pods	Replica Sets	Is Paused	Min Ready Seconds	Progress Deadline Seconds	Revision History Limit	Rollback To Revision	Strategy
✓	Normal	heapster	cluster	kube-system	1	1	2	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
✓	Normal	kube-dns	cluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:10%,Max Unavailable:0
✓	Normal	kubernetes-dashboard	cluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:25%,Max Unavailable:25%
✓	Normal	monitoring-grafana	cluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
✓	Normal	monitoring-influxdb	cluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
◆	Abnormal	ep-wordpress	cluster	default	1	1	1	False		600	10		[Recreate],
✓	Normal										10		[Recreate],
✓	Normal										10		[RollingUpdate]Max Surge:1,Max Unavailable:1
✓	Normal										10		[RollingUpdate]Max Surge:1,Max Unavailable:1
✓	Normal										10		[RollingUpdate]Max Surge:1,Max Unavailable:1
◆	Abnormal										2		[RollingUpdate]Max Surge:25%,Max Unavailable:25%
✓	Normal										10		[RollingUpdate]Max Surge:1,Max Unavailable:1
✓	Normal	stress	cluster	default	1	1	1	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1

[Status Detail]
Progressing:True, Reason:NewReplicaSetAvailable, Message:ReplicaSet "ep-wordpress-6648654476" has successfully progressed.
Available:False, Reason:MinimumReplicasUnavailable, Message:Deployment does not have minimum availability.
[Replicas Detail]
Replicas:1
Unavailable Replicas:1
Updated Replicas:1

A controller manages a set of pods and ensures that the cluster is in the specified state. Instead of manually creating a pod, controllers can be used to create pods and to manage the pods. For example, the pods maintained by a replication controller are automatically replaced if they fail, get deleted, or are terminated. The *Controllers* dashboard presents the information related to the following controller types: *Deployment, Replication Set, Replication Controller, Daemon Set, Stateful Set, Job,* and *Cron Job*.

Storage

Figure 13. Kubernetes Storage Dashboard



Alarms	Status	Name	Cluster ▲	Reclaim Policy	Claim	Storage Class	Source Type	
✓	Bound	mysql-pv	cluster	Retain	mysql-pvc	nfs	NFS	[Server=10.30.155.23], [Path=/root/nfs/mysql-pv]
◆	Failed	pv	cluster	Recycle	myclaim		NFS	[Server=172.16.51.58], [Path=/var/nfsshare/v1]
✓	Boun							[Path=/mnt/data]
✓	Boun							[Kind=Managed], [DiskName=kubernetes-dynamic-pvc-2b95e22d-dc28-11e8-b2ed-befa22179703], [DiskURI=/subscriptions/a4c
✓	Bound	pvc-dd299dfd-dc27-11e8-b2ed-befa22179703	nancyakscluster	Delete	data-mehdb-0	default	AzureDisk	[Kind=Managed], [DiskName=kubernetes-dynamic-pvc-dd299dfd-dc27-11e8-b2ed-befa22179703], [DiskURI=/subscriptions/a4d
✓	Bound	pvc-e7675c61-d5ce-11e8-83fd-2e8159d9464d	nancyakscluster	Delete	mysql-pv-claim	default	AzureDisk	[FsType=ext4], [Kind=Managed], [DiskName=kubernetes-dynamic-pvc-e7675c61-d5ce-11e8-83fd-2e8159d9464d], [DiskURI=/s
✓	Bound	pvc-e83e8400-d5ce-11e8-83fd-2e8159d9464d	nancyakscluster	Delete	wp-pv-claim	default	AzureDisk	[FsType=ext4], [Kind=Managed], [DiskName=kubernetes-dynamic-pvc-e83e8400-d5ce-11e8-83fd-2e8159d9464d], [DiskURI=/s

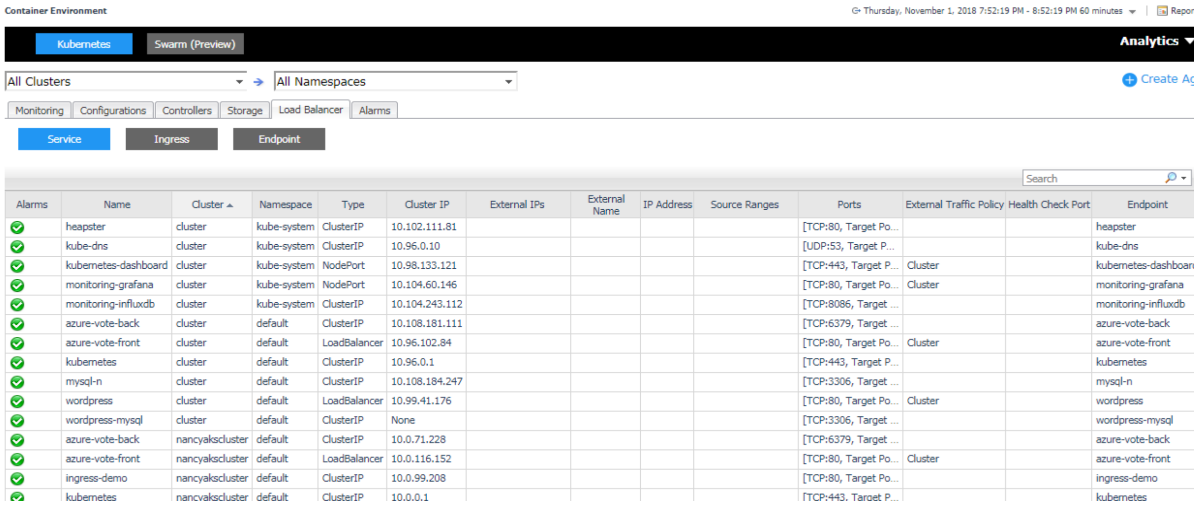
Status:Failed
Message:Recycle failed: failed to recycle volume: Pod was active on the node longer than the specified deadline

The Kubernetes storage contains volumes, storage class, persistent volume, and persistent volume claim. Volumes are on-disk files used by the containers for persistent their data as well as sharing with other containers.The *Storage* dashboard shows the information about the following storage classes:

- Storage Class provides a way for the administrator to describe the "class" of storage they offer.
- Persistent Volume subsystem provides an API for users and administrators that abstracts details of how storage is provided from how it is consumed.
- Persistent Volume Claim is used for dynamic volume provisioning which allow storage volumes to be created on-demand.

Load Balancer

Figure 14. Kubernetes Load Balancer Dashboard



Container Environment

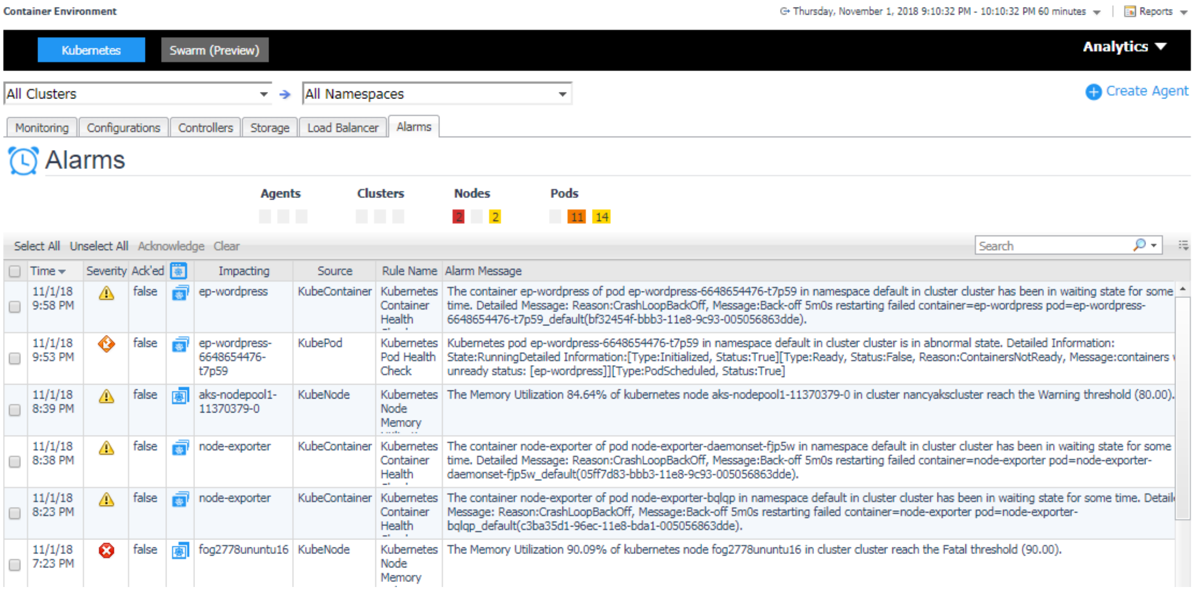
Thursday, November 1, 2018 7:52:19 PM - 8:52:19 PM 60 minutes | Repor

Alarms	Name	Cluster ▲	Namespace	Type	Cluster IP	External IPs	External Name	IP Address	Source Ranges	Ports	External Traffic Policy	Health Check Port	Endpoint
✓	heapster	cluster	kube-system	ClusterIP	10.102.111.81					[TCP:80, Target Po...			heapster
✓	kube-dns	cluster	kube-system	ClusterIP	10.96.0.10					[UDP:53, Target P...			kube-dns
✓	kubernetes-dashboard	cluster	kube-system	NodePort	10.98.133.121					[TCP:443, Target P...	Cluster		kubernetes-dashboar
✓	monitoring-grafana	cluster	kube-system	NodePort	10.104.60.146					[TCP:80, Target Po...	Cluster		monitoring-grafana
✓	monitoring-influxdb	cluster	kube-system	ClusterIP	10.104.243.112					[TCP:8086, Target ...			monitoring-influxdb
✓	azure-vote-back	cluster	default	ClusterIP	10.108.181.111					[TCP:6379, Target ...			azure-vote-back
✓	azure-vote-front	cluster	default	LoadBalancer	10.96.102.84					[TCP:80, Target Po...	Cluster		azure-vote-front
✓	kubernetes	cluster	default	ClusterIP	10.96.0.1					[TCP:443, Target P...			kubernetes
✓	mysql-n	cluster	default	ClusterIP	10.108.184.247					[TCP:3306, Target ...			mysql-n
✓	wordpress	cluster	default	LoadBalancer	10.99.41.176					[TCP:80, Target Po...	Cluster		wordpress
✓	wordpress-mysql	cluster	default	ClusterIP	None					[TCP:3306, Target ...			wordpress-mysql
✓	azure-vote-back	nancyakscluster	default	ClusterIP	10.0.71.228					[TCP:6379, Target ...			azure-vote-back
✓	azure-vote-front	nancyakscluster	default	LoadBalancer	10.0.116.152					[TCP:80, Target Po...	Cluster		azure-vote-front
✓	ingress-demo	nancyakscluster	default	ClusterIP	10.0.99.208					[TCP:80, Target Po...			ingress-demo
✓	kubernetes	nancyakscluster	default	ClusterIP	10.0.0.1					[TCP:443, Target P...			kubernetes

The *Load Balancer* dashboard includes information about Kubernetes service, endpoint, and ingress. A Kubernetes ingress can provide load balancing, SSL termination and name-based virtual hosting. A Kubernetes service is an abstraction which defines a logical set of pods and a policy by which to access them - sometime called micro-services. Kubernetes will update the endpoint whenever the set of pods in a service changes

Alarms

Figure 15. Kubernetes Alarms Dashboard



Container Environment

Thursday, November 1, 2018 9:10:32 PM - 10:10:32 PM 60 minutes | Reports

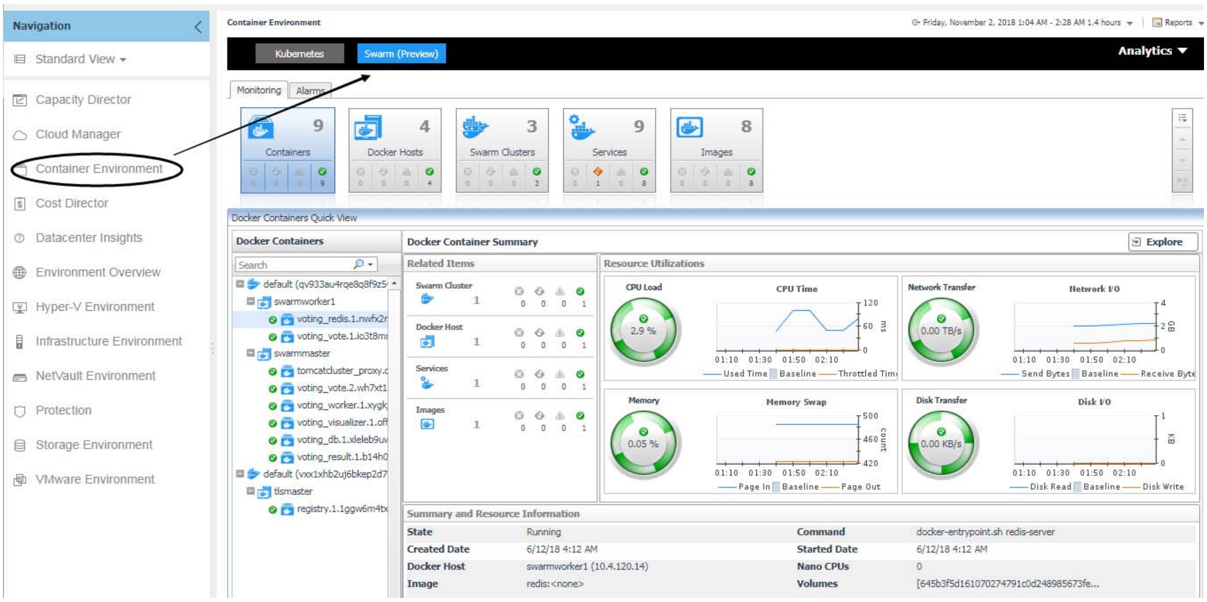
□	Time ▾	Severity	Ack'ed		Impacting	Source	Rule Name	Alarm Message
□	11/1/18 9:58 PM	⚠	false		ep-wordpress	KubeContainer	Kubernetes Container Health	The container ep-wordpress of pod ep-wordpress-6648654476-t7p59 in namespace default in cluster cluster has been in waiting state for some time. Detailed Message: Reason:CrashLoopBackOff, Message:Back-off 5m0s restarting failed container=ep-wordpress pod=ep-wordpress-6648654476-t7p59_default(bf32454f-bbb3-11e8-9c93-005056863dde).
□	11/1/18 9:53 PM	◆	false		ep-wordpress-6648654476-t7p59	KubePod	Kubernetes Pod Health Check	Kubernetes pod ep-wordpress-6648654476-t7p59 in namespace default in cluster cluster is in abnormal state. Detailed Information: State:RunningDetailed Information:[Type:Initialized, Status:True][Type:Ready, Status:False, Reason:ContainersNotReady, Message:containers unready status: [ep-wordpress]][Type:PodScheduled, Status:True]
□	11/1/18 8:39 PM	⚠	false		aks-nodepool1-11370379-0	KubeNode	Kubernetes Node Memory	The Memory Utilization 84.64% of kubernetes node aks-nodepool1-11370379-0 in cluster nancyakscluster reach the Warning threshold (80.00).
□	11/1/18 8:38 PM	⚠	false		node-exporter	KubeContainer	Kubernetes Container Health	The container node-exporter of pod node-exporter-daemonset-fjp5w in namespace default in cluster cluster has been in waiting state for some time. Detailed Message: Reason:CrashLoopBackOff, Message:Back-off 5m0s restarting failed container=node-exporter pod=node-exporter-daemonset-fjp5w_default(05ff7d83-bbb3-11e8-9c93-005056863dde).
□	11/1/18 8:23 PM	⚠	false		node-exporter	KubeContainer	Kubernetes Container Health	The container node-exporter of pod node-exporter-bqlqp in namespace default in cluster cluster has been in waiting state for some time. Detail Message: Reason:CrashLoopBackOff, Message:Back-off 5m0s restarting failed container=node-exporter pod=node-exporter-bqlqp_default(c3ba35d1-96ec-11e8-bda1-005056863dde).
□	11/1/18 7:23 PM	✖	false		fog2778ununtu16	KubeNode	Kubernetes Node Memory	The Memory Utilization 90.09% of kubernetes node fog2778ununtu16 in cluster cluster reach the Fatal threshold (90.00).

The *Alarms* dashboard displays a list of alarms generated against the monitored Kubernetes environment. Use this view to quickly identify any potential problems related to a specific Kubernetes component.

Docker Swarm

The *Docker Container Quick View*, which appears after clicking **Monitoring > Containers**, click **Swarm (preview)** from the header on top to switch to Docker Swarm dashboard.

Figure 16. Docker Swarm Dashboard



Summary and Resource Information

State	Running	Command	docker-entrypoint.sh redis-server
Created Date	6/12/18 4:12 AM	Started Date	6/12/18 4:12 AM
Docker Host	swarmworker1 (10.4.120.14)	Nano CPUs	0
Image	redis:<none>	Volumes	[645b3f5d161070274791c0d248985673fe...

Monitoring Docker Containers

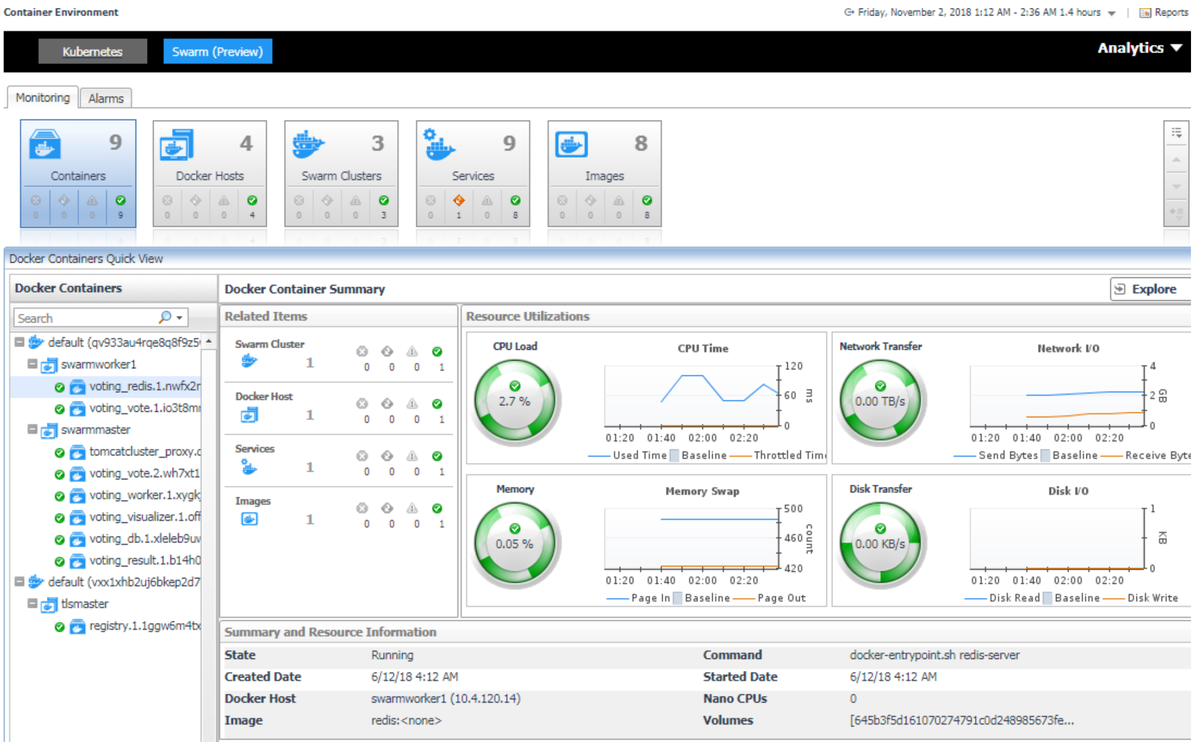
This view consists of the following two panes:

- The **Docker Containers** tree view, which appears on the left of *Docker Containers Quick View*, lists the containers existing in the monitored *Docker* environment. The containers in the tree view are grouped by **cluster > docker host > container**.
- The Docker Container Summary view, which appears on the right after you select an individual container in the **Docker Containers** tree view.

Docker Container Summary view

The **Docker Container Summary** view appears on the right when you select a container in the **Docker Containers** tree view.

Figure 17. Docker Container Summary view



Summary and Resource Information

State	Running	Command	docker-entrypoint.sh redis-server
Created Date	6/12/18 4:12 AM	Started Date	6/12/18 4:12 AM
Docker Host	swarmworker1 (10.4.120.14)	Nano CPUs	0
Image	redis:<none>	Volumes	[645b3f5d161070274791c0d248985673fe...

The **Docker Container Summary** view displays the following data:

- *Related Items: Shows the related Docker components grouped by type as well as the associated alarms.*
- *Resource Utilizations*: The resource utilization for the selected Docker Container over a selected period of time, which includes the following:
  - *CPU Load*: Shows the CPU utilization of the selected container.
  - CPU Time: Shows the used time and throttled time of the selected container.
  - Network Transfer: Shows the transfer bytes rate of the selected container over a selected period of time.
  - Network I/O: Shows the total send/receive bytes of the selected container.
  - Memory: Shows the memory utilization of the selected container.
  - Memory Swap: Shows the mounts of memory pages that are swapped to disk.
  - Disk Transfer: Shows the disk transfer bytes rate of the selected container over a selected period of time.
  - Disk I/O: Shows the disk read/write bytes of the selected container.
- *Summary and Resource Information*: Displays the detailed information about the selected Container, including *State, Command, Created Time, Started Time, Image and so on*.

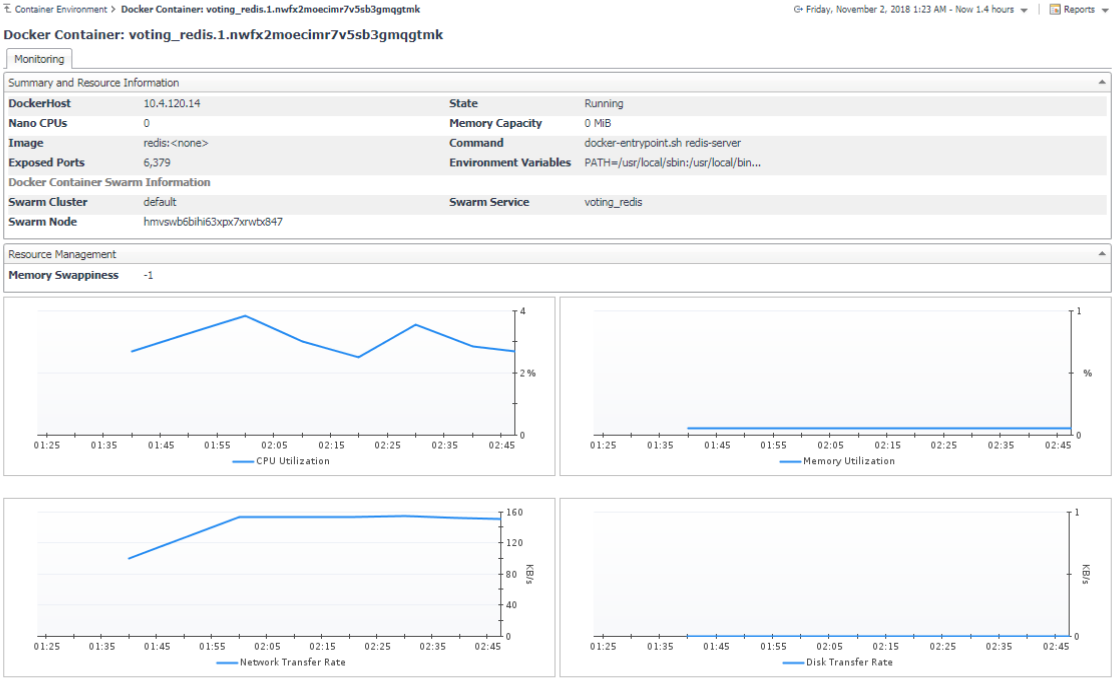
Click **Explore** on the upper right of the **Docker Container Summary** view to open the Container Explorer view, which shows more detailed information about this container.

Container Explorer view

The *Container Explorer* view opens when you click **Explore** in the Docker Container Summary view, which includes the following tabs:

- *Monitoring tab:* The *Monitoring* tab displays the overall information of the selected container over a selected period of time, including the *Summary and Resource Information* table, Resource Management table as well as the Metrics list. For more information, see Container metrics *on page 49.*

Figure 18. Docker Container Explorer view Monitoring Tab



Container Environment > Docker Container: voting_redis.1.nwfx2moecimr7v5sb3gmqgtmk

Friday, November 2, 2018 1:23 AM - Now 1.4 hours

Reports

Docker Container: voting_redis.1.nwfx2moecimr7v5sb3gmqgtmk

Monitoring

Summary and Resource Information

DockerHost	10.4.120.14	**State**	Running
Nano CPUs	0	**Memory Capacity**	0 MiB
Image	redis:<none>	**Command**	docker-entrypoint.sh redis-server
Exposed Ports	6,379	**Environment Variables**	PATH=/usr/local/sbin:/usr/local/bin...
Docker Container Swarm Information			
Swarm Cluster	default	**Swarm Service**	voting_redis
Swarm Node	hmvswb6bihi63xpx7xrwtx847		

Resource Management

Memory Swappiness	-1

Monitoring Docker Hosts

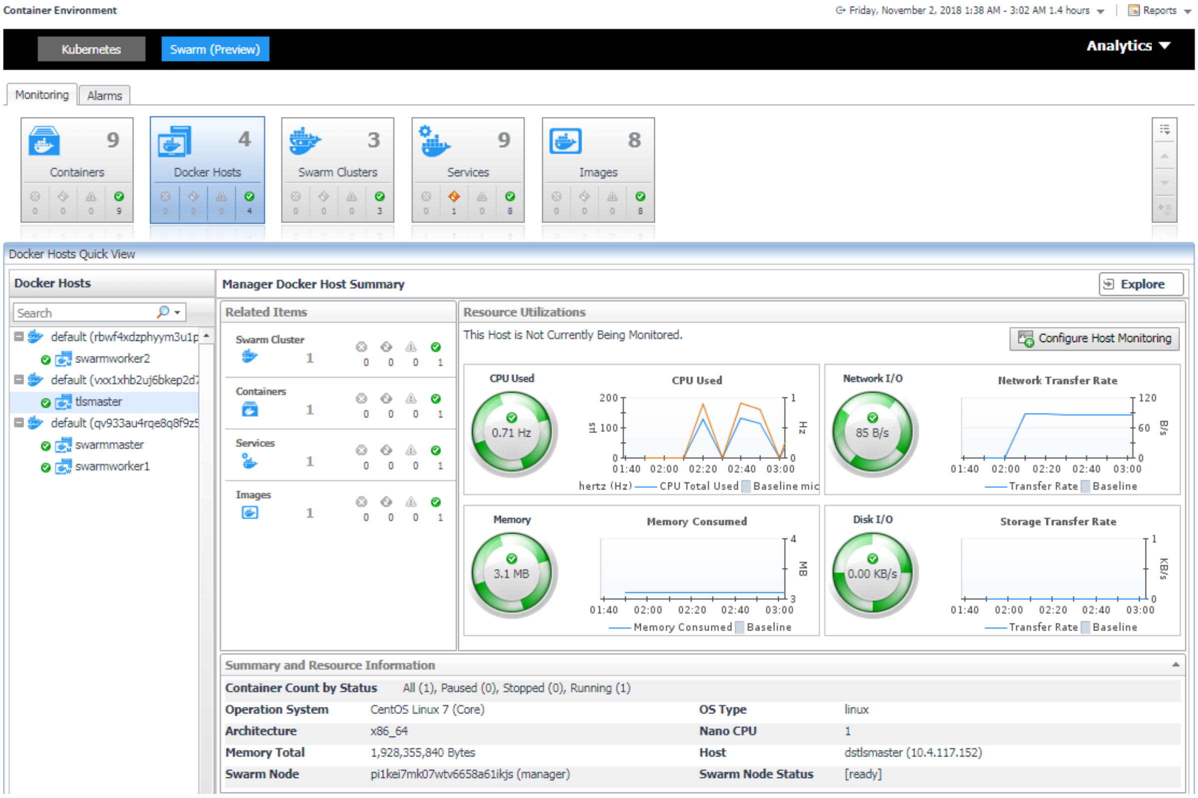
This view consists of the following two panes:

- The **Docker Hosts** tree view, which appears on the left of *Docker Hosts Quick View*, lists the docker hosts existing in the monitored *Docker* environment. The docker hosts in the tree view are grouped by **cluster > docker host**.
- The Docker Host Summary view, which appears on the right after you select an individual docker host in the **Docker Hosts** tree view.

Docker Host Summary view

The **Docker Host Summary** view appears on the right when you select a docker host in the **Docker Hosts** tree view.

Figure 19. Docker Host Summary view



Summary and Resource Information

Container Count by Status	All (1), Paused (0), Stopped (0), Running (1)		
Operation System	CentOS Linux 7 (Core)	**OS Type**	linux
Architecture	x86_64	**Nano CPU**	1
Memory Total	1,928,355,840 Bytes	**Host**	dstlsmaster (10.4.117.152)
Swarm Node	pi1kei7mk07wtv6658a61ikjs (manager)	**Swarm Node Status**	[ready]

The **Docker Host Summary** view displays the following data:

- *Related Items: Shows the related Docker components grouped by type as well as the associated alarms.*
- *Resource Utilizations*: The resource utilization for the selected docker host over a selected period of time, which includes the following:
  - *CPU Load*: Shows the CPU utilization of the selected docker host.
  - CPU Used: Shows the used CPU resources aggregated from the containers running on the docker host.
  - Network I/O and Network Transfer Rate: Shows the transfer bytes rate of the selected docker host aggregated from the containers running on the docker host over a selected period of time.
  - Memory and Memory Consumed: Shows the memory consumed bytes aggregated from the containers running on the docker host.
  - Disk I/O and Disk Transfer: Shows the disk transfer bytes rate of the selected docker host aggregated from the containers running on the docker host over a selected period of time.
- *Summary and Resource Information*: Displays the detailed information about the selected docker host, including *Container Count by Status, Operating System, Memory Total and so on*.

Click **Explore** on the upper right of the **Docker Host Summary** view to open the Docker Host Explorer view, which shows more detailed information about this container.

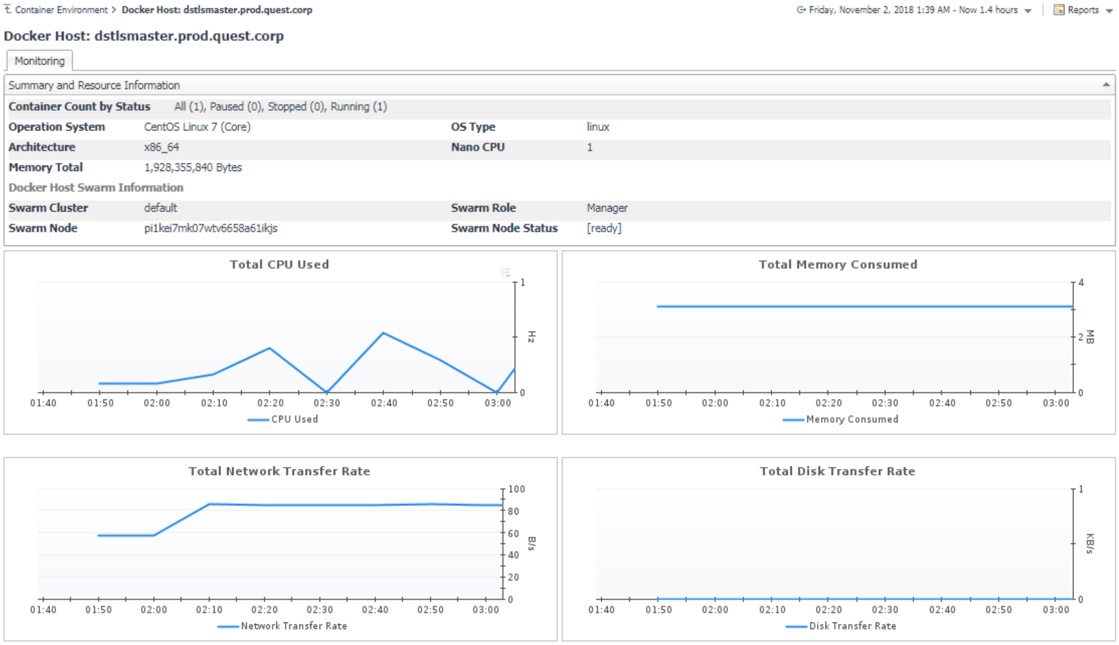
Docker Host Explorer view

The *Docker Host Explorer* view opens when you click **Explore** in the Docker Host Summary view, which includes the following tabs:

- *Monitoring tab:* The *Monitoring* tab displays the overall information of the selected docker host over a selected period of time, including the *Summary and Resource Information* table, Resource Management table as well as the Metrics list. For more information, see Container metrics *on page 49.*

  NOTE: All the docker host metrics are calculated from the aggregated metrics of the containing containers on the docker host.

Figure 20. Docker Host Explorer view Monitoring Tab



Container Environment > **Docker Host: dstlsmaster.prod.quest.corp**

Friday, November 2, 2018 1:39 AM - Now 1.4 hours | Reports

Docker Host: dstlsmaster.prod.quest.corp

Monitoring

Summary and Resource Information

Container Count by Status	All (1), Paused (0), Stopped (0), Running (1)		
Operation System	CentOS Linux 7 (Core)	**OS Type**	linux
Architecture	x86_64	**Nano CPU**	1
Memory Total	1,928,355,840 Bytes		
Docker Host Swarm Information			
Swarm Cluster	default	**Swarm Role**	Manager
Swarm Node	pi1kei7mk07wtv6658a61ikjs	**Swarm Node Status**	[ready]

Monitoring Docker Images

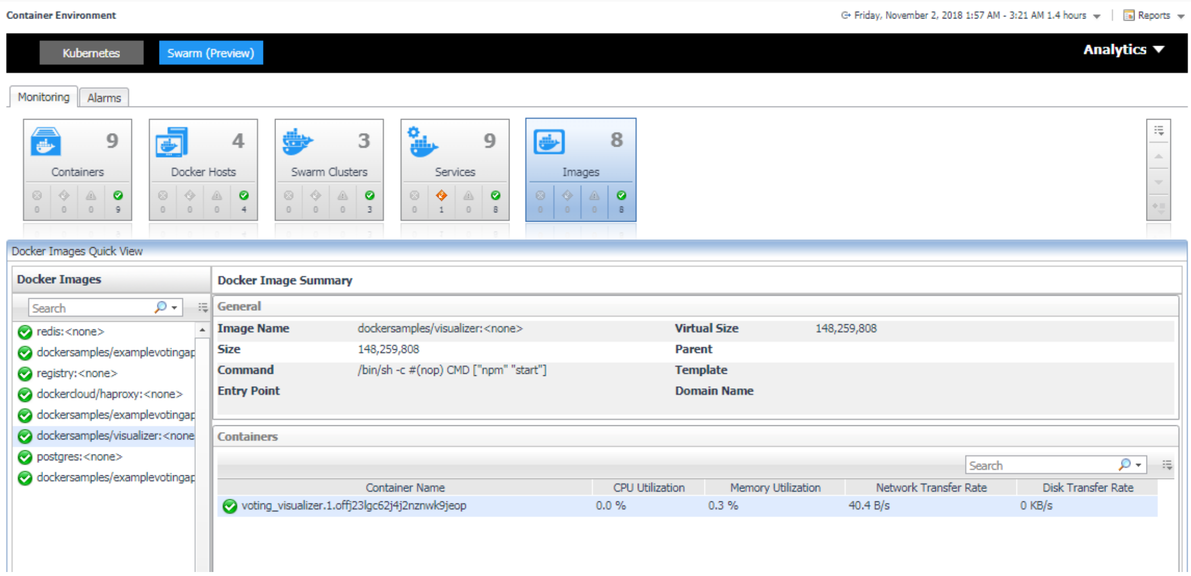
This view consists of the following two panes:

- The **Docker Images** tree view, which appears on the left of *Docker Images Quick View*, lists the docker images existing in the monitored *Docker* environment.
- The Docker Image Summary view, which appears on the right after you select an individual docker image in the **Docker Images** tree view.

Docker Image Summary view

The **Docker Image Summary** view appears on the right when you select a docker image in the **Docker Images** tree view.

Figure 21. Docker Image Summary view



redis:<none>
dockersamples/examplevotingap
registry:<none>
dockercloud/haproxy:<none>
dockersamples/examplevotingap
dockersamples/visualizer:<none
postgres:<none>
dockersamples/examplevotingap

General

Image Name	dockersamples/visualizer:<none>	**Virtual Size**	148,259,808
Size	148,259,808	**Parent**	
Command	/bin/sh -c #(nop) CMD ["npm" "start"]	**Template**	
Entry Point		**Domain Name**	

Containers

Container Name	CPU Utilization	Memory Utilization	Network Transfer Rate	Disk Transfer Rate
✓ voting_visualizer.1.offj23lgc62j4j2nznwk9jeop	0.0 %	0.3 %	40.4 B/s	0 KB/s

The **Docker Image Summary** view displays the following data:

- *General: Shows the general information of the selected docker image, including Size, Command, Entry Point and so on.*
- *Containers: The table list the containers with useful metrics that are created based on the selected docker image.*

Monitoring Docker Swarm Clusters

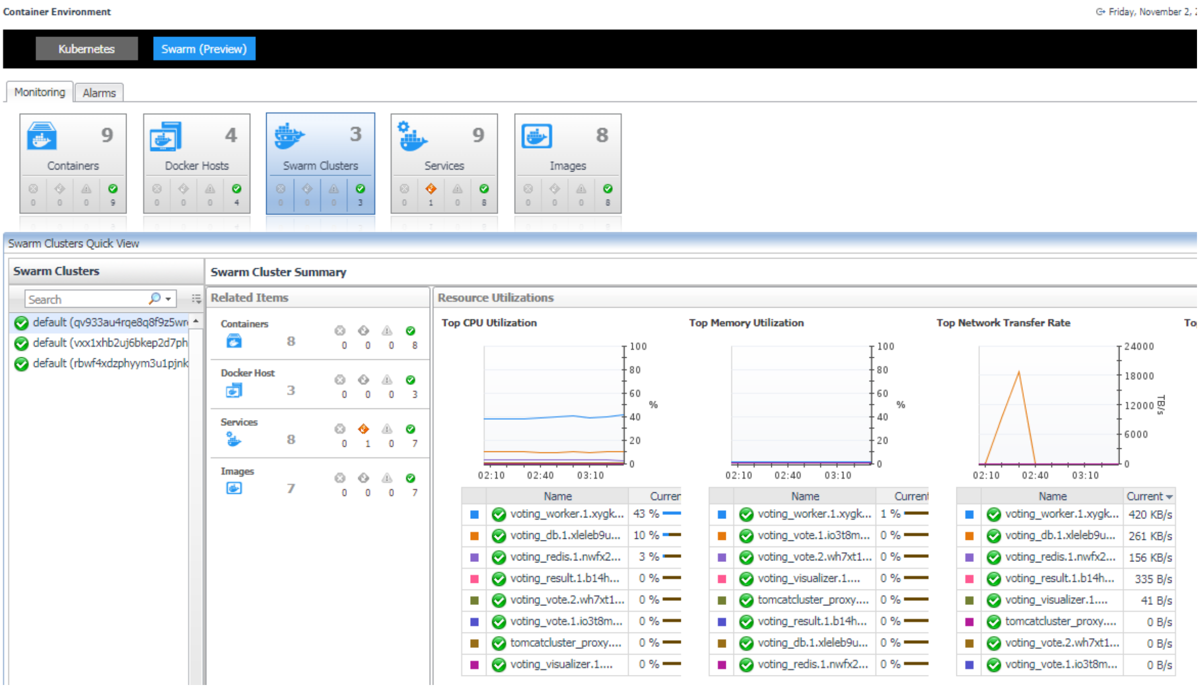
This view consists of the following two panes:

- The **Swarm Clusters** tree view, which appears on the left of *Swarm Clusters Quick View*, lists the docker swarm clusters existing in the monitored *Docker* environment.
- The Docker Swarm Cluster Summary view, which appears on the right after you select an individual docker swarm cluster in the **Swarm Clusters** tree view.

Docker Swarm Cluster Summary view

The **Docker Swarm Cluster Summary** view appears on the right when you select a docker swarm cluster in the **Swarm Clusters** tree view.

Figure 22. Docker Swarm Cluster Summary view



	Name	Currer
🟦	✓ voting_worker.1.xygk...	43 %
🟧	✓ voting_db.1.xleleb9u...	10 %
🟪	✓ voting_redis.1.nwfx2...	3 %
🟥	✓ voting_result.1.b14h...	0 %
🟩	✓ voting_vote.2.wh7xt1...	0 %
🟦	✓ voting_vote.1.io3t8m...	0 %
🟫	✓ tomcatcluster_proxy....	0 %
🟪	✓ voting_visualizer.1....	0 %

	Name	Current
🟦	✓ voting_worker.1.xygk...	1 %
🟧	✓ voting_vote.1.io3t8m...	0 %
🟪	✓ voting_vote.2.wh7xt1...	0 %
🟥	✓ voting_visualizer.1....	0 %
🟩	✓ tomcatcluster_proxy....	0 %
🟦	✓ voting_result.1.b14h...	0 %
🟫	✓ voting_db.1.xleleb9u...	0 %
🟪	✓ voting_redis.1.nwfx2...	0 %

	Name	Current
🟦	✓ voting_worker.1.xygk...	420 KB/s
🟧	✓ voting_db.1.xleleb9u...	261 KB/s
🟪	✓ voting_redis.1.nwfx2...	156 KB/s
🟥	✓ voting_result.1.b14h...	335 B/s
🟩	✓ voting_visualizer.1....	41 B/s
🟪	✓ tomcatcluster_proxy....	0 B/s
🟫	✓ voting_vote.2.wh7xt1...	0 B/s
🟦	✓ voting_vote.1.io3t8m...	0 B/s

The **Docker Swarm Cluster Summary** view displays the following data:

- *Related Items: Shows the related Docker components grouped by type as well as the associated alarms.*
- *Resource Utilizations: Shows CPU Utilization, Memory Utilization, Network Transfer Rate, Disk Transfer Rate metrics of the containers running in this docker swarm cluster in descending order.*

Monitoring Docker Swarm Services

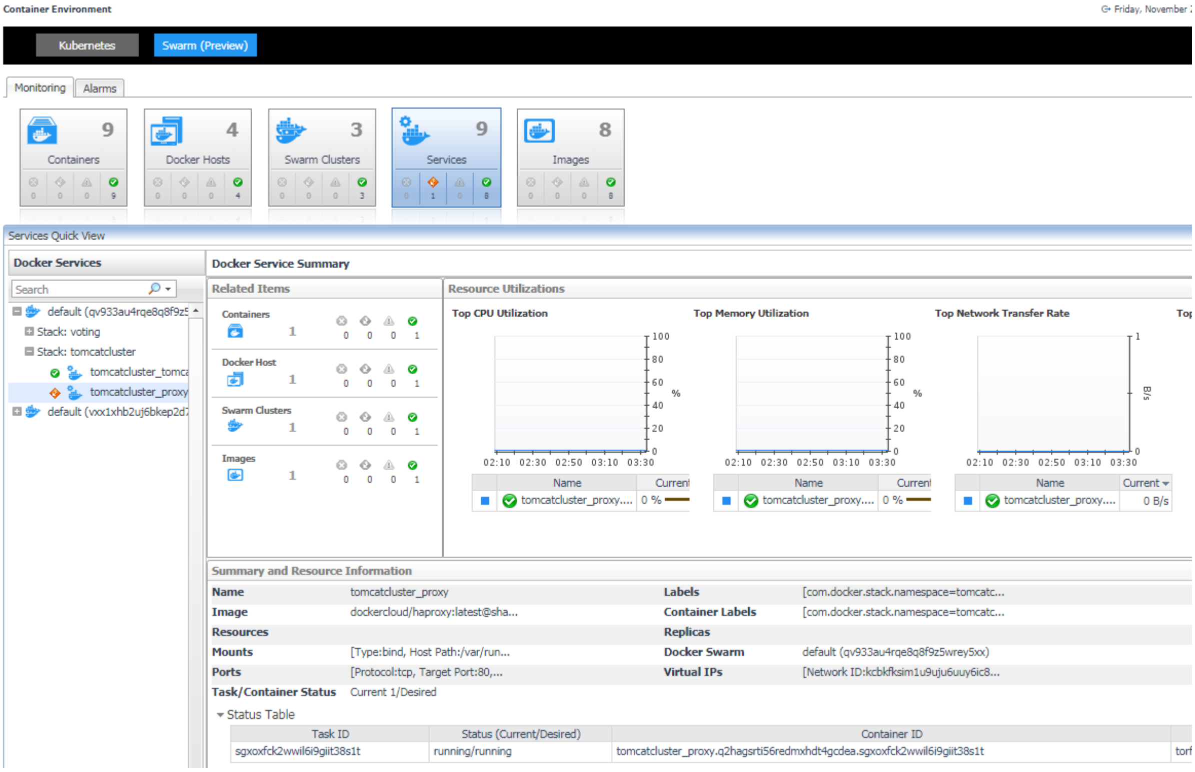
This view consists of the following two panes:

- The **Swarm Services** tree view, which appears on the left of *Swarm Services Quick View*, lists the docker swarm services existing in the monitored *Docker* environment.
- The Docker Swarm Service Summary view, which appears on the right after you select an individual docker swarm service in the **Swarm Services** tree view.

Docker Swarm Service Summary view

The **Docker Swarm Service Summary** view appears on the right when you select a docker swarm service in the **Swarm Services** tree view.

Figure 23. Docker Swarm Service Summary view



	Name	Current
■	tomcatcluster_proxy....	0 %

	Name	Current
■	tomcatcluster_proxy....	0 %

	Name	Current
■	tomcatcluster_proxy....	0 B/s

Summary and Resource Information

Name	tomcatcluster_proxy	**Labels**	[com.docker.stack.namespace=tomcatc...
Image	dockercloud/haproxy:latest@sha...	**Container Labels**	[com.docker.stack.namespace=tomcatc...
Resources		**Replicas**	
Mounts	[Type:bind, Host Path:/var/run...	**Docker Swarm**	default (qv933au4rqe8q8f9z5wrey5xx)
Ports	[Protocol:tcp, Target Port:80,...	**Virtual IPs**	[Network ID:kcbkfksim1u9uju6uuy6ic8...
Task/Container Status	Current 1/Desired		

Status Table

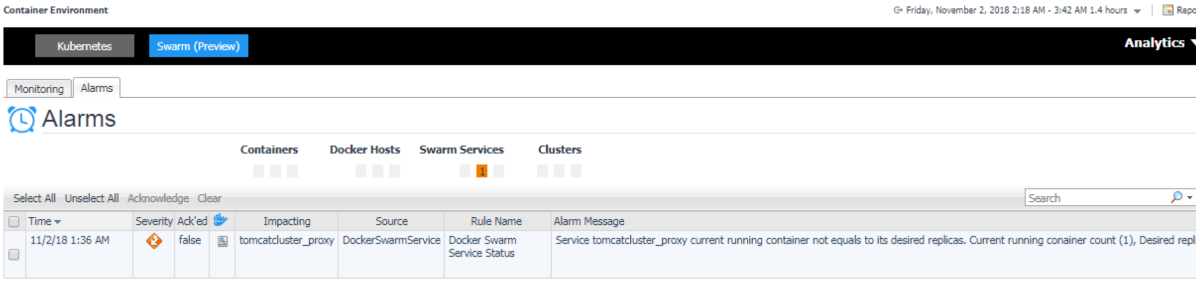
Task ID	Status (Current/Desired)	Container ID	
sgxoxfck2wwil6i9giit38s1t	running/running	tomcatcluster_proxy.q2hagsrti56redmxhdt4gcdea.sgxoxfck2wwil6i9giit38s1t	torf

The **Docker Swarm Service Summary** view displays the following data:

- *Related Items: Shows the related Docker components grouped by type as well as the associated alarms.*
- *Resource Utilizations: Shows CPU Utilization, Memory Utilization, Network Transfer Rate, Disk Transfer Rate metrics of the containers running in this docker swarm service in descending order.*
- *Summary and Resource Information: Shows the summary information of the docker swarm service, including Labels, Image, Mount Volumes, Ports, Container Status and so on.*

Alarms

Figure 24. Docker Swarm Alarms Dashboard



□	Time	Severity	Ack'ed		Impacting	Source	Rule Name	Alarm Message
□	11/2/18 1:36 AM	◆	false		tomcatcluster_proxy	DockerSwarmService	Docker Swarm Service Status	Service tomcatcluster_proxy current running container not equals to its desired replicas. Current running conainer count (1), Desired repl

The *Alarms* dashboard displays a list of alarms generated against the monitored Docker environment. Use this view to quickly identify any potential problems related to a specific Docker component.

Analytics

Foglight for Container Management provide analytics feature for Kubernetes and Docker Swarm.

Heat Map is a two-dimensional representation of data in which values are represented by colors. Showing collected metrics with elaborate heat maps allows you to understand complex data sets and the monitored cluster environment well.

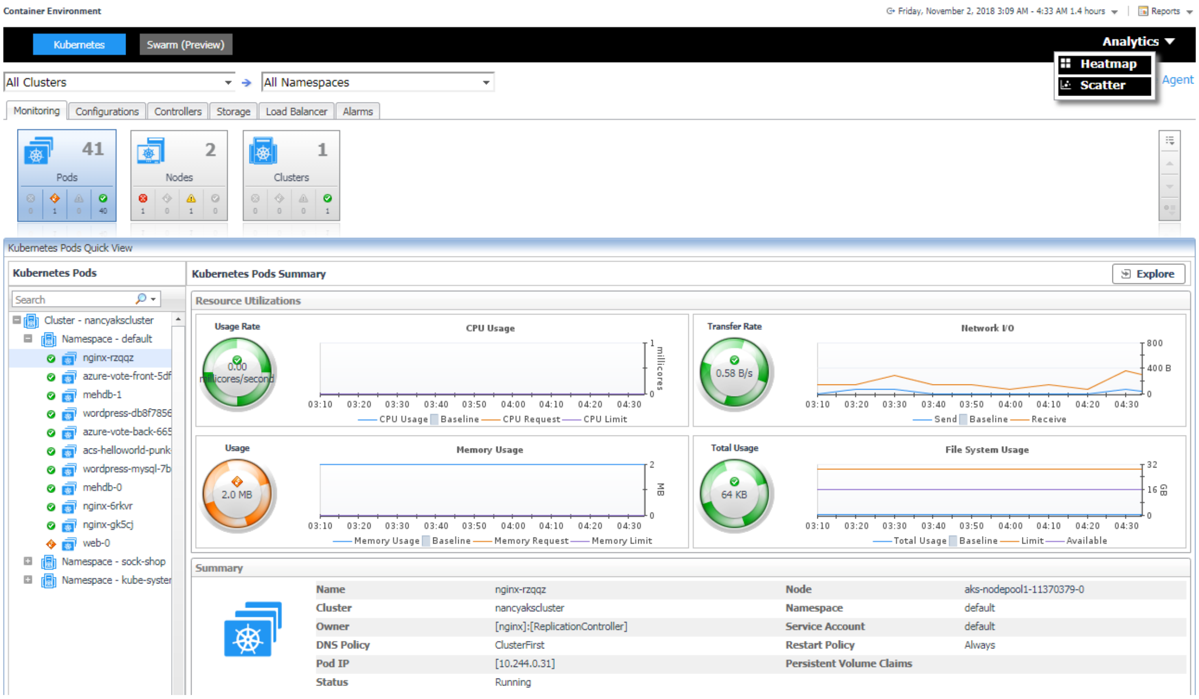
Scatter Plot is used to display values in points using two variables for a set of data. The points is color-coded also, Color Metric can be used to display one additional variable.

- Kubernetes analytics
  - Heatmap analytics
  - Scatter Plot analytics
- Docker Swarm analytics
  - Heatmap analytics
  - Scatter Plot analytics

Kubernetes analytics

In the Container dashboard, choose **Kubernetes** from the header. Then click **Analytics** from the header, a drop down view will display with **Heatmap** and **Scatter** on it. Click **Heatmap** will navigate to the Kubernetes **Heatmap Analytics** dashboard, while click **Scatter** will navigate to the Kubernetes **Scatter Plot Analytics** dashboard.

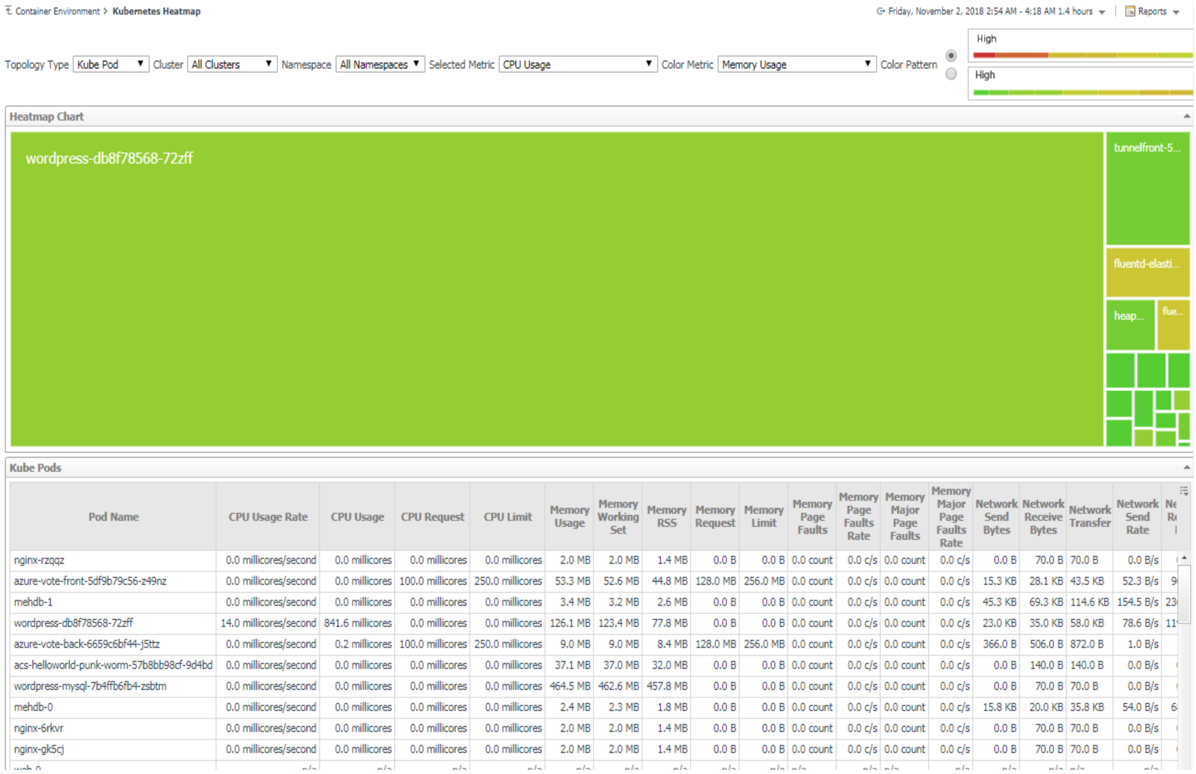
Figure 25. Kubernetes analytics Navigation



Name	nginx-rzqqz	Node	aks-nodepool1-11370379-0
Cluster	nancyakscluster	Namespace	default
Owner	[nginx]:[ReplicationController]	Service Account	default
DNS Policy	ClusterFirst	Restart Policy	Always
Pod IP	[10.244.0.31]	Persistent Volume Claims	
Status	Running		

Heatmap analytics

Figure 26. Kubernetes Heatmap Analytics Dashboard



Container Environment > Kubernetes Heatmap
Friday, November 2, 2018 2:54 AM - 4:18 AM 1.4 hours
Reports
High
High
Heatmap Chart
wordpress-db8f78568-72zff
tunnelfront-5...
fluentd-elasti...
heap...
flue...
Kube Pods

Topology Type: Kube Pod; Cluster: All Clusters; Namespace: All Namespaces; Selected Metric: CPU Usage; Color Metric: Memory Usage; Color Pattern: ● ○

Pod Name	CPU Usage Rate	CPU Usage	CPU Request	CPU Limit	Memory Usage	Memory Working Set	Memory RSS	Memory Request	Memory Limit	Memory Page Faults	Memory Page Faults Rate	Memory Major Page Faults	Memory Major Page Faults Rate	Network Send Bytes	Network Receive Bytes	Network Transfer	Network Send Rate	Ne R
nginx-rzqqz	0.0 millicores/second	0.0 millicores	0.0 millicores	0.0 millicores	2.0 MB	2.0 MB	1.4 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	0.0 B	70.0 B	70.0 B	0.0 B/s	
azure-vote-front-5df9b79c56-z49nz	0.0 millicores/second	0.0 millicores	100.0 millicores	250.0 millicores	53.3 MB	52.6 MB	44.8 MB	128.0 MB	256.0 MB	0.0 count	0.0 c/s	0.0 count	0.0 c/s	15.3 KB	28.1 KB	43.5 KB	52.3 B/s	9
mehdb-1	0.0 millicores/second	0.0 millicores	0.0 millicores	0.0 millicores	3.4 MB	3.2 MB	2.6 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	45.3 KB	69.3 KB	114.6 KB	154.5 B/s	23
wordpress-db8f78568-72zff	14.0 millicores/second	841.6 millicores	0.0 millicores	0.0 millicores	126.1 MB	123.4 MB	77.8 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	23.0 KB	35.0 KB	58.0 KB	78.6 B/s	11
azure-vote-back-6659c6bf44-j5ttz	0.0 millicores/second	0.2 millicores	100.0 millicores	250.0 millicores	9.0 MB	9.0 MB	8.4 MB	128.0 MB	256.0 MB	0.0 count	0.0 c/s	0.0 count	0.0 c/s	366.0 B	506.0 B	872.0 B	1.0 B/s	
acs-helloworld-punk-worm-57b8bb98cf-9d4bd	0.0 millicores/second	0.0 millicores	0.0 millicores	0.0 millicores	37.1 MB	37.0 MB	32.0 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	0.0 B	140.0 B	140.0 B	0.0 B/s	
wordpress-mysql-7b4ffb6fb4-zsbtm	0.0 millicores/second	0.0 millicores	0.0 millicores	0.0 millicores	464.5 MB	462.6 MB	457.8 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	0.0 B	70.0 B	70.0 B	0.0 B/s	
mehdb-0	0.0 millicores/second	0.0 millicores	0.0 millicores	0.0 millicores	2.4 MB	2.3 MB	1.8 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	15.8 KB	20.0 KB	35.8 KB	54.0 B/s	6
nginx-6rkvr	0.0 millicores/second	0.0 millicores	0.0 millicores	0.0 millicores	2.0 MB	2.0 MB	1.4 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	0.0 B	70.0 B	70.0 B	0.0 B/s	
nginx-gk5cj	0.0 millicores/second	0.0 millicores	0.0 millicores	0.0 millicores	2.0 MB	2.0 MB	1.4 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	0.0 B	70.0 B	70.0 B	0.0 B/s	
web-0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	

Heat maps will be refreshed automatically when you change either of the following fields:

- *Topology Type:* Indicates the monitored topology object, including Kubernetes Pod, Kubernetes Node, and Kubernetes Cluster.
- *Cluster:* Lists all clusters available in the monitored Kubernetes environment.
- *Namespace:* Lists all namespaces available in the monitored Kubernetes environment.
- *Selected Metric:* Populates a rectangle based upon the selected metrics. For example, if you select *Memory Usage* from the *Selected Metric* drop-down list, the rectangle area will be populated based on the used memory for the selected topology object. For more information about metrics, refer to Kubernetes metrics *on page 48*.
- Rendering related metrics: For example, if you select *CPU Usage Rate* and Red to Green, the rectangle of the topology object that has larger value of CPU Usage Rate will be rendered in red.
  - *Color Metric*: Renders the color of rectangle based upon the selected color metric.
  - *Color Pattern*: Offers two patterns, Red to Green (larger value shows in red) or Green to Red (larger value shows in green).

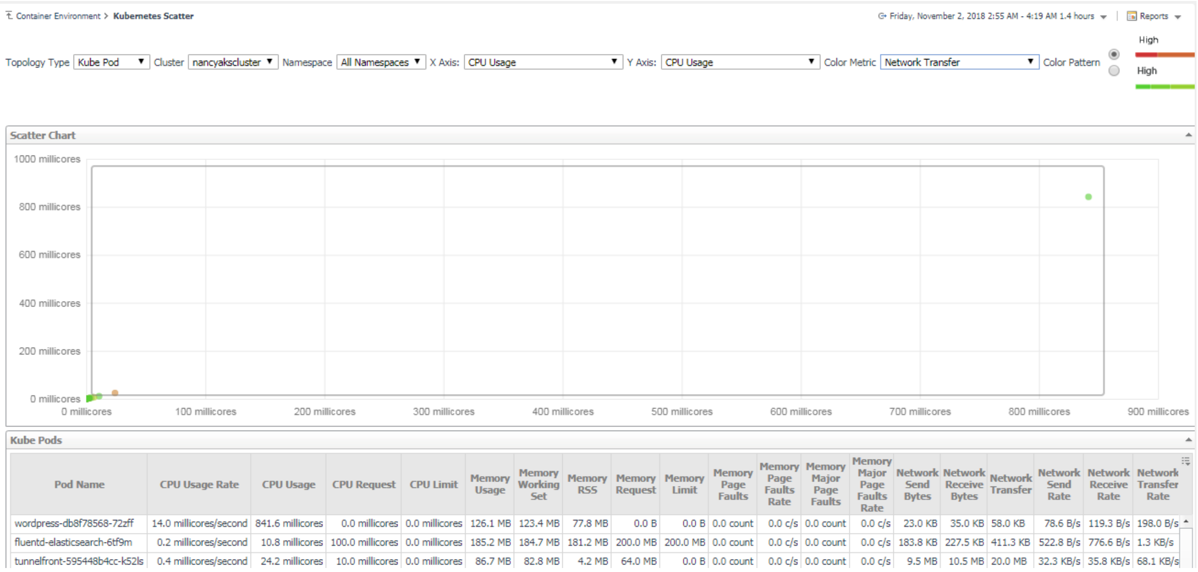
Figure 26 shows an example of heat map. This sample diagram represents the “wordpress-db8f78568-72zff” has the maximum amounts of CPU usage, while “fluentd-elastic-ef455uh68-72cfe” has a higher Memory Usage. If you switch the Color Pattern, then “wordpress-db8f78568-72zff” will turn to red. Clicking the object name on the heat map directs you to the relevant object *Explorer* dashboard. For more information, see:

- Pods Explorer view on page 26
- Pod metrics on page 48
- Nodes Explorer view on page 28
- Node metrics on page 49
- Cluster Explorer view on page 30

- Cluster metrics on page 49

Scatter Plot analytics

Figure 27. Kubernetes Scatter Plot Analytics Dashboard



Container Environment > **Kubernetes Scatter**

Friday, November 2, 2018 2:55 AM - 4:19 AM 1.4 hours

Topology Type: Kube Pod | Cluster: nancyakscluster | Namespace: All Namespaces | X Axis: CPU Usage | Y Axis: CPU Usage | Color Metric: Network Transfer | Color Pattern: ● High 🟥 ○ High 🟩

Scatter Chart

Kube Pods

Pod Name	CPU Usage Rate	CPU Usage	CPU Request	CPU Limit	Memory Usage	Memory Working Set	Memory RSS	Memory Request	Memory Limit	Memory Page Faults	Memory Page Faults Rate	Memory Major Page Faults	Memory Major Page Faults Rate	Network Send Bytes	Network Receive Bytes	Network Transfer	Network Send Rate	Network Receive Rate	Network Transfer Rate
wordpress-db8f78568-72zff	14.0 millicores/second	841.6 millicores	0.0 millicores	0.0 millicores	126.1 MB	123.4 MB	77.8 MB	0.0 B	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	23.0 KB	35.0 KB	58.0 KB	78.6 B/s	119.3 B/s	198.0 B/s
fluentd-elasticsearch-6tf9m	0.2 millicores/second	10.8 millicores	100.0 millicores	0.0 millicores	185.2 MB	184.7 MB	181.2 MB	200.0 MB	200.0 MB	0.0 count	0.0 c/s	0.0 count	0.0 c/s	183.8 KB	227.5 KB	411.3 KB	522.8 B/s	776.6 B/s	1.3 KB/s
tunnelfront-595448b4cc-k52ls	0.4 millicores/second	24.2 millicores	10.0 millicores	0.0 millicores	86.7 MB	82.8 MB	4.2 MB	64.0 MB	0.0 B	0.0 count	0.0 c/s	0.0 count	0.0 c/s	9.5 MB	10.5 MB	20.0 MB	32.3 KB/s	35.8 KB/s	68.1 KB/s

The points on the chart will be refreshed automatically when you change either of the following fields:

- *Topology Type:* Indicates the monitored topology object, including Kubernetes Pod, Kubernetes Node, and Kubernetes Cluster.
- *Cluster:* Lists all clusters available in the monitored Kubernetes environment.
- *Namespace:* Lists all namespaces available in the monitored Kubernetes environment.
- *X Axis:* Indicates which metrics will be plotted on X axis.
- *Y Axis:* Indicates which metrics will be plotted on Y axis.
- Rendering related metrics:
  - Color Metric: Renders the color of circle based upon the selected metrics.
  - *Color Pattern*: Offers two patterns, Red to Green (larger value shows in red) or Green to Red (larger value shows in green).

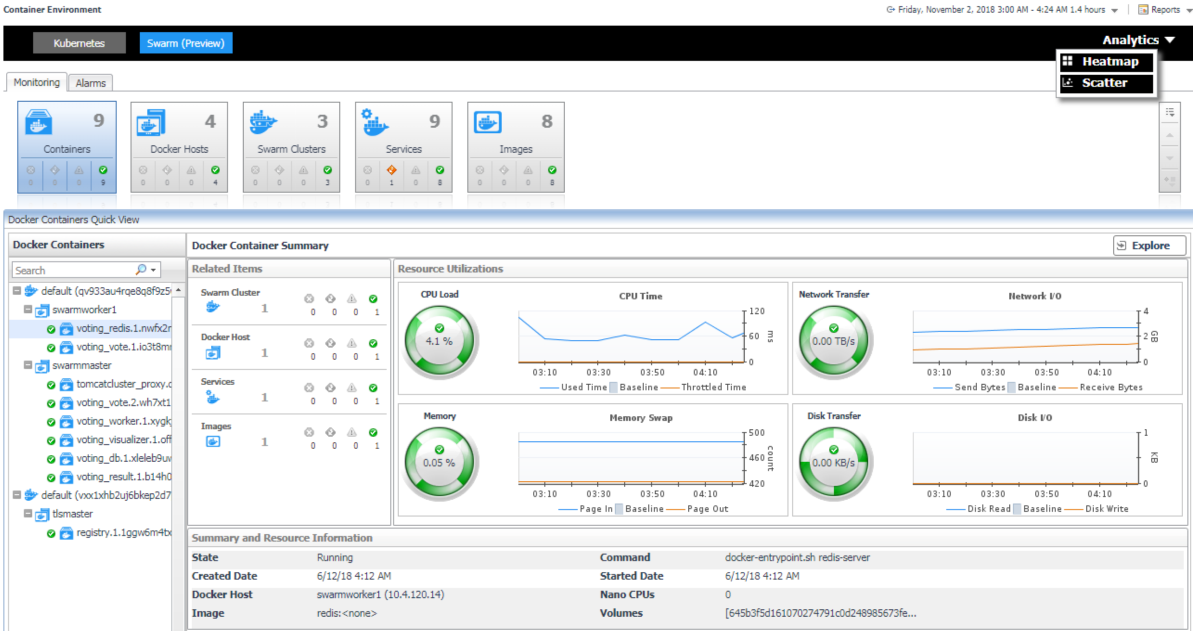
Figure 27 shows an example of Scatter Plot analytics. The purple circle in the middle represents the following: “wordpress-db8f78568-72zff” CPU Usage is around 0.85 cores, its Memory Usage is around 121MB, and its value of Network Transfer Bytes is not high. For more information, see:

- Pods Explorer view on page 26
- Pod metrics on page 48
- Nodes Explorer view on page 28
- Node metrics on page 49
- Cluster Explorer view on page 30
- Cluster metrics on page 49

Docker Swarm analytics

In the Container dashboard, choose **Docker Swarm** from the header. Then click **Analytics** from the header, a drop down view will display with **Heatmap** and **Scatter** on it. Click **Heatmap** will navigate to the Docker Swarm **Heatmap Analytics** dashboard, while click **Scatter** will navigate to the Docker Swarm **Scatter Plot Analytics** dashboard.

Figure 28. Docker Swarm Analytics Navigation

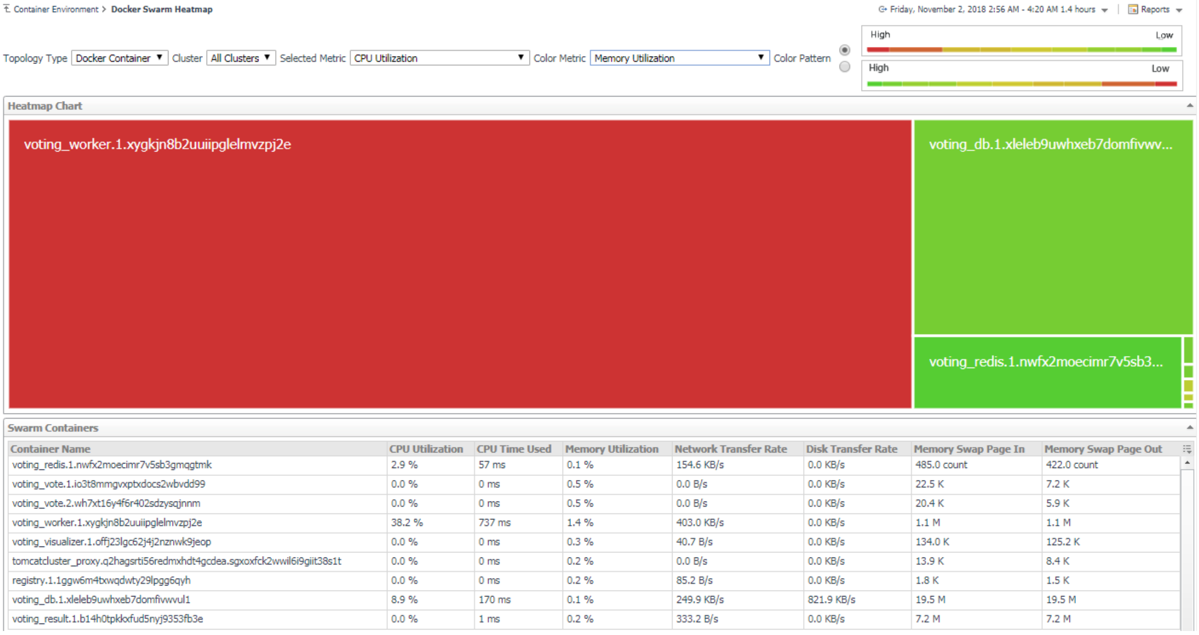


Summary and Resource Information

State	Running	**Command**	docker-entrypoint.sh redis-server
Created Date	6/12/18 4:12 AM	**Started Date**	6/12/18 4:12 AM
Docker Host	swarmworker1 (10.4.120.14)	**Nano CPUs**	0
Image	redis:<none>	**Volumes**	[645b3f5d161070274791c0d248985673fe...

Heatmap analytics

Figure 29. Docker Swarm Heatmap Analytics Dashboard


Container Environment > Docker Swarm Heatmap
Friday, November 2, 2018 2:56 AM - 4:20 AM 1.4 hours
Reports
High
Low
Heatmap Chart
voting_worker.1.xygkjn8b2uuiipglelmvzpj2e
voting_db.1.xleleb9uwhxeb7domfivwv...
voting_redis.1.nwfx2moecimr7v5sb3...

Topology Type: Docker Container; Cluster: All Clusters; Selected Metric: CPU Utilization; Color Metric: Memory Utilization; Color Pattern

Swarm Containers

Container Name	CPU Utilization	CPU Time Used	Memory Utilization	Network Transfer Rate	Disk Transfer Rate	Memory Swap Page In	Memory Swap Page Out
voting_redis.1.nwfx2moecimr7v5sb3gmqgtmk	2.9 %	57 ms	0.1 %	154.6 KB/s	0.0 KB/s	485.0 count	422.0 count
voting_vote.1.io3t8mmgvxptxdocs2wbvdd99	0.0 %	0 ms	0.5 %	0.0 B/s	0.0 KB/s	22.5 K	7.2 K
voting_vote.2.wh7xt16y4f6r402sdzysqjnnm	0.0 %	0 ms	0.5 %	0.0 B/s	0.0 KB/s	20.4 K	5.9 K
voting_worker.1.xygkjn8b2uuiipglelmvzpj2e	38.2 %	737 ms	1.4 %	403.0 KB/s	0.0 KB/s	1.1 M	1.1 M
voting_visualizer.1.offj23lgc62j4j2nznwk9jeop	0.0 %	0 ms	0.3 %	40.7 B/s	0.0 KB/s	134.0 K	125.2 K
tomcatcluster_proxy.q2hagsrti56redmxhdt4gcdea.sgxoxfck2wwil6i9giit38s1t	0.0 %	0 ms	0.2 %	0.0 B/s	0.0 KB/s	13.9 K	8.4 K
registry.1.1ggw6m4txwqdwty29lpgg6qyh	0.0 %	0 ms	0.2 %	85.2 B/s	0.0 KB/s	1.8 K	1.5 K
voting_db.1.xleleb9uwhxeb7domfivwvul1	8.9 %	170 ms	0.1 %	249.9 KB/s	821.9 KB/s	19.5 M	19.5 M
voting_result.1.b14h0tpkkxfud5nyj9353fb3e	0.0 %	1 ms	0.2 %	333.2 B/s	0.0 KB/s	7.2 M	7.2 M

Heat maps will be refreshed automatically when you change either of the following fields:

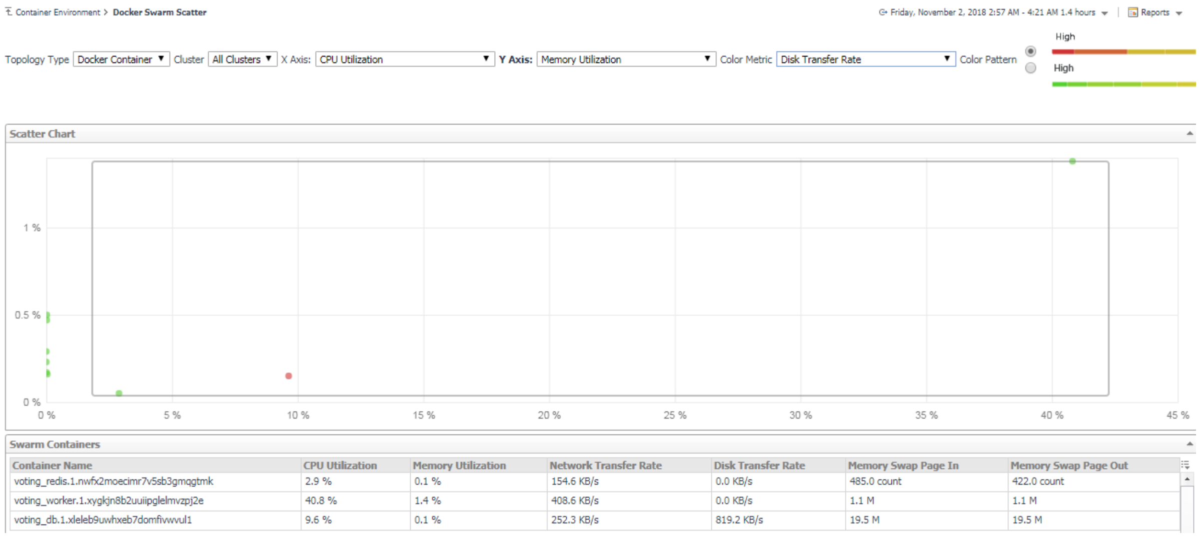
- *Topology Type:* Indicates the monitored topology object, including Docker Container and Docker Host.
- *Cluster:* Lists all clusters available in the monitored Docker Swarm environment.
- *Selected Metric:* Populates a rectangle based upon the selected metrics. For example, if you select *Memory Time Used* from the *Selected Metric* drop-down list, the rectangle area will be populated based on the used CPU time for the selected topology object. For more information about metrics, refer to Docker Swarm metrics *on page 49*.
- Rendering related metrics: For example, if you select *CPU Utilization* and Red to Green, the rectangle of the topology object that has larger value of CPU Utilization will be rendered in red.
  - *Color Metric*: Renders the color of rectangle based upon the selected color metric.
  - *Color Pattern*: Offers two patterns, Red to Green (larger value shows in red) or Green to Red (larger value shows in green).

Figure 29 shows an example of heat map. This sample diagram represents the “voting_redis.1.nwfx2moecimr7v5sb3gmqgtmk” has the maximum amounts of CPU Utilization which is the largest in size, and also it has the higher Memory Utilization since it is in Red. If you switch the Color Pattern, then “voting_redis.1.nwfx2moecimr7v5sb3gmqgtmk” will turn to green. Clicking the object name on the heat map directs you to the relevant object *Explorer* dashboard. For more information, see:

- Container Explorer view on page 36
- Docker Host Explorer view on page 38
- Container metrics on page 49

Scatter Plot analytics

Figure 30. Docker Swarm Scatter Plot Analytics Dashboard


Container Environment > Docker Swarm Scatter
Friday, November 2, 2018 2:57 AM - 4:21 AM 1.4 hours
Reports
High
High
Scatter Chart
1 %
0.5 %
0 %
0 %
5 %
10 %
15 %
20 %
25 %
30 %
35 %
40 %
45 %
Swarm Containers

Topology Type: Docker Container | Cluster: All Clusters | X Axis: CPU Utilization | Y Axis: Memory Utilization | Color Metric: Disk Transfer Rate | Color Pattern: ● High 🟥 ○ High 🟩

Container Name	CPU Utilization	Memory Utilization	Network Transfer Rate	Disk Transfer Rate	Memory Swap Page In	Memory Swap Page Out
voting_redis.1.nwfx2moecimr7v5sb3gmqgtmk	2.9 %	0.1 %	154.6 KB/s	0.0 KB/s	485.0 count	422.0 count
voting_worker.1.xygkjn8b2uuiipglelmvzpj2e	40.8 %	1.4 %	408.6 KB/s	0.0 KB/s	1.1 M	1.1 M
voting_db.1.xleleb9uwhxeb7domfivwvul1	9.6 %	0.1 %	252.3 KB/s	819.2 KB/s	19.5 M	19.5 M

The points on the chart will be refreshed automatically when you change either of the following fields:

- *Topology Type:* Indicates the monitored topology object, including Docker Container and Docker Host.
- *Cluster:* Lists all clusters available in the monitored Docker Swarm environment.
- *X Axis:* Indicates which metrics will be plotted on X axis.
- *Y Axis:* Indicates which metrics will be plotted on Y axis.
- Rendering related metrics:
  - Color Metric: Renders the color of circle based upon the selected metrics.
  - *Color Pattern*: Offers two patterns, Red to Green (larger value shows in red) or Green to Red (larger value shows in green).

Figure 30 shows an example of Scatter Plot analytics. The purple circle in the middle represents the following: “voting_redis.1.nwfx2moecimr7v5sb3gmqgtmk” CPU Utilization is 2.9%, its Memory Usage is 0.1%, and its value of Network Transfer Bytes is not high. For more information, see:

- Container Explorer view on page 36
- Docker Host Explorer view on page 38
- Container metrics on page 49

Metrics

Kubernetes metrics

Pod metrics

Table 4. Pod metrics

Metric name	Description
CPU Usage	CPU usage on all cores in millicores
CPU Usage Rate	CPU usage rate on all cores in millicores/second
CPU Request	CPU request (the guaranteed amount of resources) in millicores
CPU Limit	CPU hard limit in millicores
Memory Usage	Total memory usage in bytes
Memory Working Set	Total working set usage. Working set is the memory being used and not easily dropped by the kernel.
Memory Rss	RSS memory usage
Memory Request	Memory request (the guaranteed amount of resources) in bytes
Memory Limit	Memory hard limit in bytes
Memory Page Faults	Number of page faults
Memory Major Page Faults	Number of major page faults
Memory Page Faults Rate	Number of page faults per second
Memory Major Page Faults Rate	Number of major page faults per second
Network Send	Total send bytes
Network Receive	Total receive bytes
Network Send Rate	Total send bytes per second
Network Receive Rate	Total receive bytes per second
Network Send Errors	Total send errors count
Network Receive Errors	Total receive errors count
Network Send Errors Rate	Total send errors count per second
Network Receive Errors Rate	Total receive errors count per second
Network Transfer	Total send and receive bytes
Network Transfer Rate	Total send and receive bytes per second

Node metrics

Table 5. Node metrics

Metric name	Description
CPU Usage	CPU usage on all cores in millicores
CPU Usage Rate	CPU usage rate on all cores in millicores/second
CPU Request	CPU request (the guaranteed amount of resources) in millicores
CPU Limit	CPU hard limit in millicores
CPU Utilization	CPU utilization as a share of node allocatable
Memory Usage	Total memory usage in bytes
Memory Working Set	Total working set usage. Working set is the memory being used and not easily dropped by the kernel.
Memory Rss	RSS memory usage
Memory Request	Memory request (the guaranteed amount of resources) in bytes
Memory Limit	Memory hard limit in bytes
Memory Page Faults	Number of page faults
Memory Major Page Faults	Number of major page faults
Memory Page Faults Rate	Number of page faults per second
Memory Major Page Faults Rate	Number of major page faults per second
Memory Utilization	Memory utilization as a share of memory allocatable

Cluster metrics

Table 6. Cluster metrics

Metric name	Description
CPU Usage	CPU usage on all cores in millicores
CPU Usage Rate	CPU usage rate on all cores in millicores/second
CPU Request	CPU request (the guaranteed amount of resources) in millicores
CPU Limit	CPU hard limit in millicores
Memory Usage	Total memory usage in bytes
Memory Request	Memory request (the guaranteed amount of resources) in bytes
Memory Limit	Memory hard limit in bytes

Docker Swarm metrics

Container metrics

Table 7. Container metrics

Metric name	Description
CPU Utilization	CPU utilization.
CPU Time Used	Total CPU time that a container used.
CPU Throttled Time	Total time that a container’s CPU usage was throttled.
Memory Page Fault	Total page fault count of a container’s Memory.
Memory Consumed	Total memory consumed of a container in bytes.

Table 7. Container metrics

Metric name	Description
Memory Utilization	Memory utilization.
Memory PageIn Rate	Total page in count of a container’s Memory.
Memory PageOut Rate	Total page out count of a container’s Memory.
Disk Read Bytes	Total disk read bytes.
Disk Write Bytes	Total disk write bytes.
Disk Transfer Rate	Sum of total disk read and write bytes.
Network Send Packets	Total network send packets count.
Network Receive Packets	Total network receive packets count.
Network Send Bytes	Total network send bytes.
Network Receive Bytes	Total network receive bytes.
Network Inbound Dropped Packets	Total dropped packet count of all the packets coming into the container.
Network Outbond Dropped Packets	Total dropped packet count of all the packets going out from the container.
Network Transfer Rate	Sum of network send bytes and receive bytes per seconds during a specific period.

About Us

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit https://www.quest.com/company/contact-us.aspx/.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.