

Foglight® for Container Management 1.2.0  
**User and Administration Guide**



**© 2019 Quest Software Inc.**

**ALL RIGHTS RESERVED.**

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

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 LLC 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 trademark 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.



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

Foglight for Container Management User and Administration Guide  
Updated - May 2019  
Foglight Version - 5.9.4  
Software Version - 1.2.0

# Contents

<b>Understanding Foglight® for Container Management</b>	<b>5</b>
About Foglight for Container Management	5
Architecture	6
Sizing Your Monitored Environment	6
Foglight Management Server Requirements	7
Kubernetes Agent Requirements	7
Docker Swarm Agent Requirements	7
Getting Started	8
Prerequisite	8
Creating and Activating Agent	18
Configuring data collection interval	23
<b>Using Foglight for Container Management</b>	<b>24</b>
Kubernetes	24
Monitoring Kubernetes Pods	24
Monitoring Kubernetes Nodes	27
Monitoring Kubernetes Clusters	30
Monitoring Kubernetes Other Components	32
Alarms	36
Administration	36
Docker Swarm	37
Monitoring Docker Containers	37
Monitoring Docker Hosts	39
Monitoring Docker Swarm Clusters	43
Monitoring Docker Swarm Services	43
Alarms	44
Analytics	45
Kubernetes analytics	45
Docker Swarm analytics	49
Metrics	52
Kubernetes metrics	52
Docker Swarm metrics	53
<b>About Us</b>	<b>55</b>
We are more than just a name	55
Our brand, our vision. Together.	55
Contacting Quest	55
Technical support resources	55

---

# Understanding Foglight<sup>®</sup> for Container Management

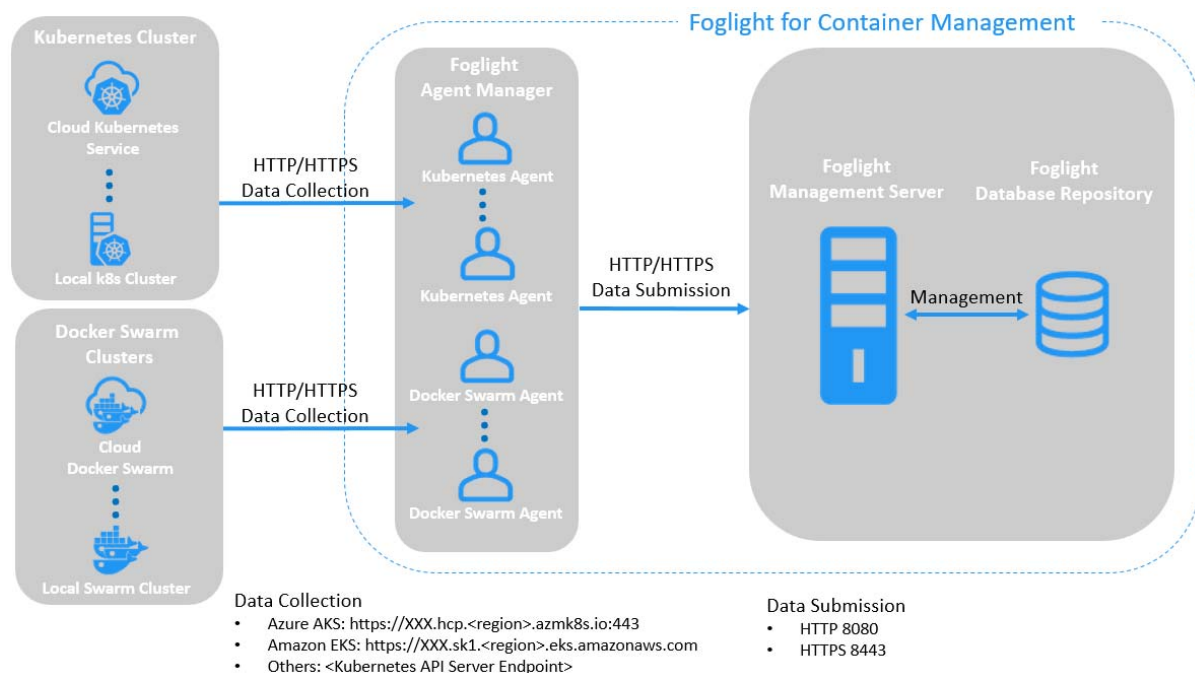
- [About Foglight for Container Management](#)
- [Architecture](#)
- [Sizing Your Monitored Environment](#)
  - [Foglight Management Server Requirements](#)
  - [Kubernetes Agent Requirements](#)
  - [Docker Swarm Agent Requirements](#)
- [Getting Started](#)
  - [Prerequisite](#)
  - [Creating and Activating Agent](#)
  - [Configuring data collection interval](#)

## 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<sup>®</sup> 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



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](#)
    - [Preparing the Kubernetes credential](#)
    - [Enabling Heapster service in monitored environment](#)
  - [Docker Swarm Agent](#)
    - [Preparing Docker Swarm Agent credentials](#)
    - [Enabling Docker Remote API for monitored docker host](#)
    - [Uploading Docker Swarm Agent credentials](#)
- [Creating and Activating Agent](#)
  - [Creating and Activating a Kubernetes Agent](#)
  - [Creating and Activating a Docker Swarm Agent](#)
- [Configuring data collection interval](#)

## 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:

**i | 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)

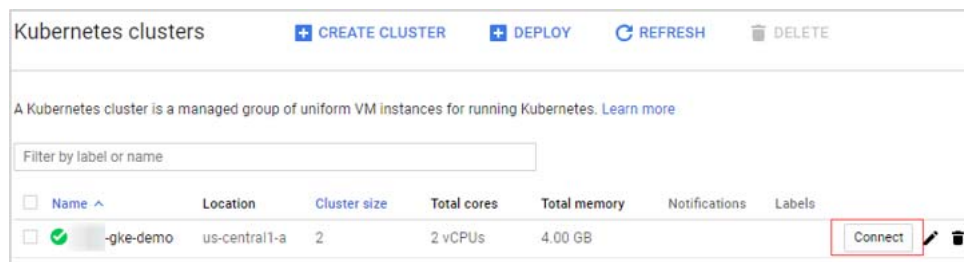
Follow the Amazon EKS official guide [Getting Started with Amazon EKS](#). Follow the guide and complete [Create a kubeconfig for Amazon EKS](#). in the end of the guide.

**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.



- 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-208510
```

[Run in Cloud Shell](#)

- 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.G1zuBVkzkoVclVUV 7yQM50DpQ7z7ahGzFA f2e08FhxZjDlCXRFdAw5yt8c9dHBT90yYk
        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.

- `kubectl create serviceaccount <service account name>`
- `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.

- `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>
```

- `kubectl describe secret <secret name>`

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

[illegible]

- e Open the intermediate Kubernetes configuration file under `<USER_HOME>/k8s/config`, and then add the user and change the token to the new one.

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURDekNDQWZP20F3SUJBZ01RV2Q...
    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.G1zuBVkzkoVcl1VUV_7yXM50DpQ7z7ahGzFA_f2o08FhxZjD1CXRFdAw5ytBc9dHBT90yYk8...
        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.eyJpc3MiOiJrdWU1cm5ldGZvLS3N1cnZpZ2VhY2NwdW50Iiwia3...
```

## 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 `.yaml` file after you performing the steps.

Clusters / kube-demo-cluster

kube-demo-cluster Expires in a month ● Normal

**Access** Overview Worker Nodes Worker Pools Services

### 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": "LS0tLS1CRUdJITI2DRUJSUJZJ0FURStL
S0tCk1JSUJZUENDQXkyZ0F3SUJBZ01KQUx5SndFOUQ2N2N2TUeWR0NTc0dTSWZiRFFkM3U0FNRGct4T
pBHMJm1YVQkPNTUxkUkPNa114WUdZGESHRRpPUFEXITJGau1RUTJabU1618pjd0YUaGhZUEF6TFd0p
U1Eua114WUdZGESHRRpPUFEXITJGau1RUTJabU1618pjd0YUaGhZUEF6TFd0pU1Eua114WUdZGESH
RRTURUNGN31FZFRFZURUReqUcK16STJNU0ZtTURSae16agB0UE5oVUyGhE5tUmpNeK0:TURNF1KRXdNe
TFyZFdKbGNTNvKkR1Z6TFd0aE1JSUMKSVpBTkja3Foa21HOXcuqkFRRUZBQUYDQMc4QU1JSUNDZ0tDQ
ddFQX11dk4x1puc09CcKJHT1PT0uvenBpRuo5M3ReYUUYeGtYmNlvcjMzclZUUSaY1R6a21xQjUMU
z0BRkFkTUFQNEUVRGJmZnpHcKx3M20ZkxDL2huMhuCjNRQ100U25421RRU1ErK0R2MUK3UkQzcD1pT
ZzP2UanErK0R3g1NPFearBompU0Gcx1U1ZJZSS:EvduRy1pw3U4Q0USSEBxOHFZZAKKQ3JJa
FNIdEu0K2N3KXQuNEZyQ3dRb1N1TUUNUXBYUFREKx1d05yeXpX0UkZBompM2FrahQ30S:SV211U1BGR
3BqbfFBZx05:jBQ0X5eJDenFNG1DTHJxREZ6M3UMUUM3TFhNEFhMys20EUxCKM3UzK0NUhBTU96U
GppSZZU21v0U53UFBDC0pKVT1PM09Iq110b25nQn1EK1BZcExkRzdT20N3WTFJaXc1ZzQKau10dU04S
zRcKtPSnluK2ItRl4d6QUPc194cjk2d9y4Tn14Uk11UkNFC2RzTUG1N3hPS3pgN1Joh2E3bGpPZRoYL
Zzhb1Z20Kpx1TUPU9U0UchobitJS09ZZ105cmh2c0xkentGT1Jz0Fp1BD1qDnk4U0RncE4ZL2ZobTKRt
pUcKduVlcxcKdUWZCZRbc3cG1cKREZSU0w0CtGvafXbtGUTZ5d2NKO1FqGnZ6St1uZEp6ZnQ4Zk9IZ
kRncjBteFqKRO5aN1pH5U1FU1ZZU3JNTUc2RCMT011S2BqRH1hRUpaVn80dU14R3YvU3NLMUZ2cB1UW
UJvbbdBaD1v3RCF9pCpNUU3SK11NkcL2V3UGFhc3NDMTZBN2J0Z1Z0Nk9o0UUsTXpQUh1UWR6ZndRR
3dxcjB4NH8M1U2c302U1dFC1BZSEUmr0JmEY4M5KcEmzeFUDQKdFQUFHT1FNRT1R3SFFZFRZSME9CQ
11FRkcxUUYeQxZ21Tcnd1dK1tRkMdxdebm7cESNqjhm9TFUZE13Uu1NqNFBKcxeUEYacQxZ21Tc
nd1dK1tRkM5N2xtc2UvTkiB40dBMUkR4RRgpNQ10CQY4d0RRUuPlh1pJahZ1tkFRRUx0UFEZ24Uq
kFGc29JhVfYdjiRcKc41TNKaJQ5andUhb2N0d21U1JTC1NFd1RuanRvhwGdEd0dU9EVIUMhMmSjRUA
3g5U9nRkUx20Yz3ZPh1NFVFI4T316UGxT24vY9kbUUsQUuKaJBHUUPoUEhCdG1HUUdtSy91ck1Ta
21Z0UJjQ3RvYU1oeUoZT1dn0Xh1U3p2H2RQe1Ex0utnazESYUZSVYkRQv0U1UnpuQ011cUpCTIdZ0H2tU
U1ieThmMDR1N19pN1JZR1R411FUIUZO89VTGZnaJNg0XA2Q21cMz1Re11wU95CjZPejd1bU0pL0YvM
pUadnRSRCNt1ZQn1dnaDE4dUR0d2tnhne0T0p0b360D9M00dESnRtlyeUeNcFOU0UUNNcKMUUS1TEZ1
F11OW1k4T1Uv511Hh3pBS08zc0UCNng4cTUZY2J0cMXhhZaEvR1Bvc1N25nNBNEGuUFD2Rk8vUahNcQp5U
0Z1MjBRHc1aK3hH02U1U1k10VjFkY2x6akt0FQK3JXK25wT19FU0Z1Z1B1Vnd1UW9R3djiYit6SjZha
HRIcmJUTUz2aU5ZS0pCQdmUj1KTE0vcFZnTGIYSDdZkM4QjFaQ1N0cE2Z40Z0cXNqYtk2RD1Id2tnS
GpZU1J2eUuKAVSRF24eH8G0GouVTRk2hdancvZnFpU1ELJUMTUNZRjFTU2NZT1Bd1UuoUGN2cUNFe
Rd1Z23RREZcJ11Hq2YnHh121xcpn1ZUJNnjgZ4UURER1azJXWesU1ZcQn13B3Y4QUNhPhcY2NqU
0Z0NU10R1d1jeUdWdjFZUUM4ckR0dE0HTk1hR1ZMSY1eUJhB25TMBJ3Z25RUTH2RMVzNGu3a29TZk1LR
1KxZMM05DhwaU2a13pyMct0Ue1xakMKOUBoNFTZS11xa2Zc10tLS0tR05E1ENFU1LRJk1DQURFLS0tL
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>
```

- [illegible]

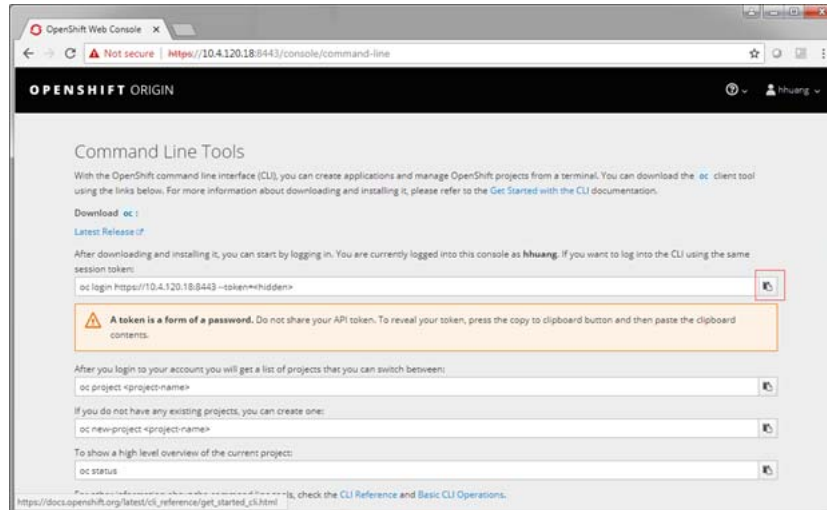
- ```
apiVersion: v1
clusters:
- name: kube-demo-cluster
  cluster:
    certificate-authority: ca-me101-kube-demo-cluster.pem
    server: https://130.198.66.34:30244
contexts:
- name: kube-demo-cluster
  context:
    cluster: kube-demo-cluster
    user: Jane.Wang@quest.com
    namespace: default
current-context: 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: eyJraWQ1O1IyMDEzMTAzMCOwMDowMDowMCIsImFaZjY1NiJTNHjU2...
        idp-issuer-url: https://iam.ng.bluemix.net/kubernetes
        refresh-token: J1B9J8mGdOcQbYvbmh7csxFZ5jC9QEJxgWP7E57db_w...
```

- ```
apiVersion: v1
clusters:
- name: kube-demo-cluster
  cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS1R
      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: eyJhbGciOiJIUzU1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJrdWJl
```

- 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.



- 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.

```

1  apiVersion: v1
2  clusters:
3  - cluster:
4      insecure-skip-tls-verify: true
5      server: <cluster-ip-port>
6      name: <config-cluster-name>
7  contexts:
8  - context:
9      cluster: <config-cluster-name>
10     namespace: <project-name>
11     user: <config-user-name>
12     name: <config-context-name>
13     current-context: <config-context-name>
14  kind: Config
15  preferences: {}
16  users:
17  - name: <config-user-name>
18     user:
19         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.eyJpc3MiOiJrdWJlcm5ldGVzL3N1cnZpY2VhbnQyZW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNIjWVudC9uYW1lc3BhY2UiOiJkZWZhZDw0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNIjWVudC9zZWNIeXZOUmFtZSI6Im9zLWFkb
```



```
WluLXRva2VuLWY0a2ZsIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLW
FjY291bnQubmFtZSI6Im9zLWFKbWluIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9
zZXJ2aWNlLWVjY291bnQudWlkIjoiodmZNGU0NTQtNzQ1Yy0xMWU4LWVmNmEtMDA1MDU2YjY3
NDFhIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6b3MtYWRTaW4ifQ.RW
H_AoXy2U1elkHN_Bs9IR1xo0zNCJlwcY0h3zuQnrkOFi8gVpX1I77uhAPp7oIjPqDSWkUAN9F
6mP_tNdGwJsqrMhYEMotCLnnIM61BYxIcABvwr66aOZ3Gn0D7EM5M_7XgKDC16ON3W5NaH0D8
DpVTYqgkQ49u3qt4gqrcjVCaSSDNWlgGxY4K0IDrUbKkdgaRKzeD9o4Bv9VbYICqyxwoUebku
JACHiXGICse-ozS_zroPiltT5HW-RY0Pn3Fp3zBnydiokna0-mXot5lqoYc-
R6ElU9YSrAOhWm9Q8ipiut6OczXbmLPM4DYve6dmHi_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
```

**i | 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
```

**i | 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.

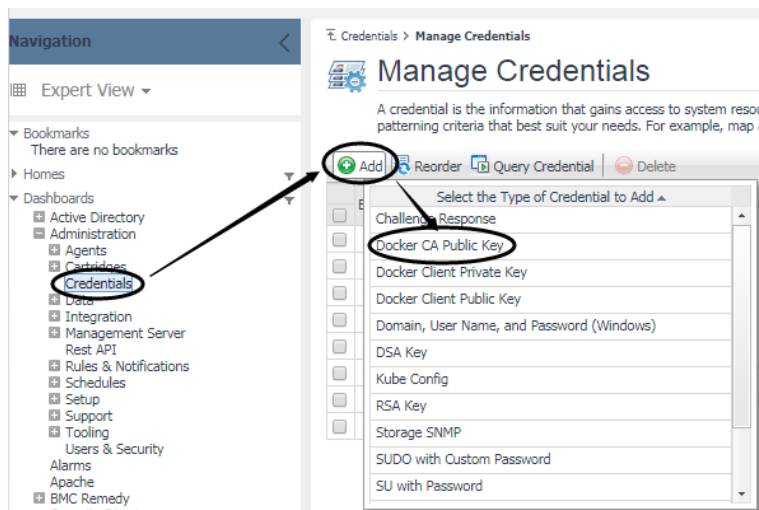
When complete [Preparing Docker Swarm Agent credentials](#) on page 15, following credentials should be generated.

- 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.

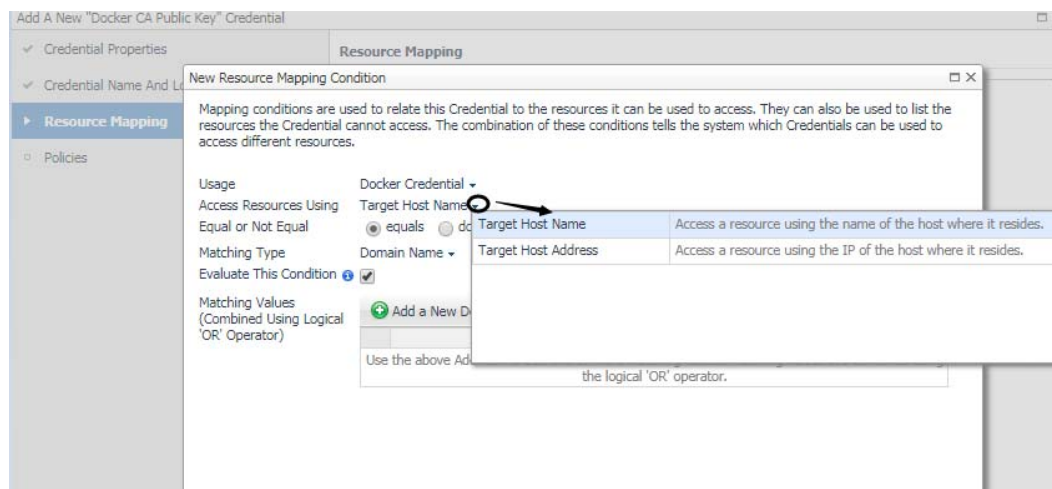
On the **Administration > Credentials > Manage Credentials** dashboard, click **Add**, and then select Docker CA Public Key or Docker Client Public Key or Docker Client Private Key to upload related credentials. Take Docker CA Public Key as an example.



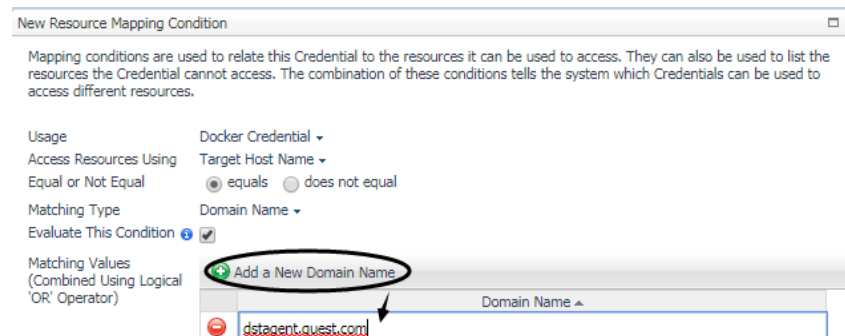


In the **Add a New “Docker CA Public Key” Credential** dialog box, specify the following values:

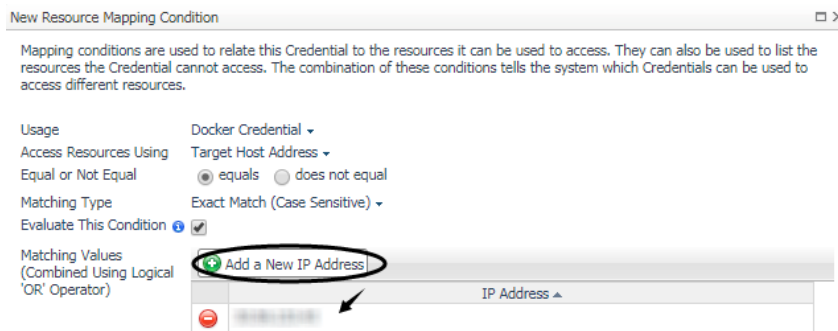
- Credential Properties: Click **Load from file** to import Docker CA Public Key, and then click **Next**.
- Credential Name And Lockbox: Specify a unique name for this credential, and then click **Next**.
- Resource Mapping: Click **Add**. In the **New Resource Mapping Condition** dialog box, choose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



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



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



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

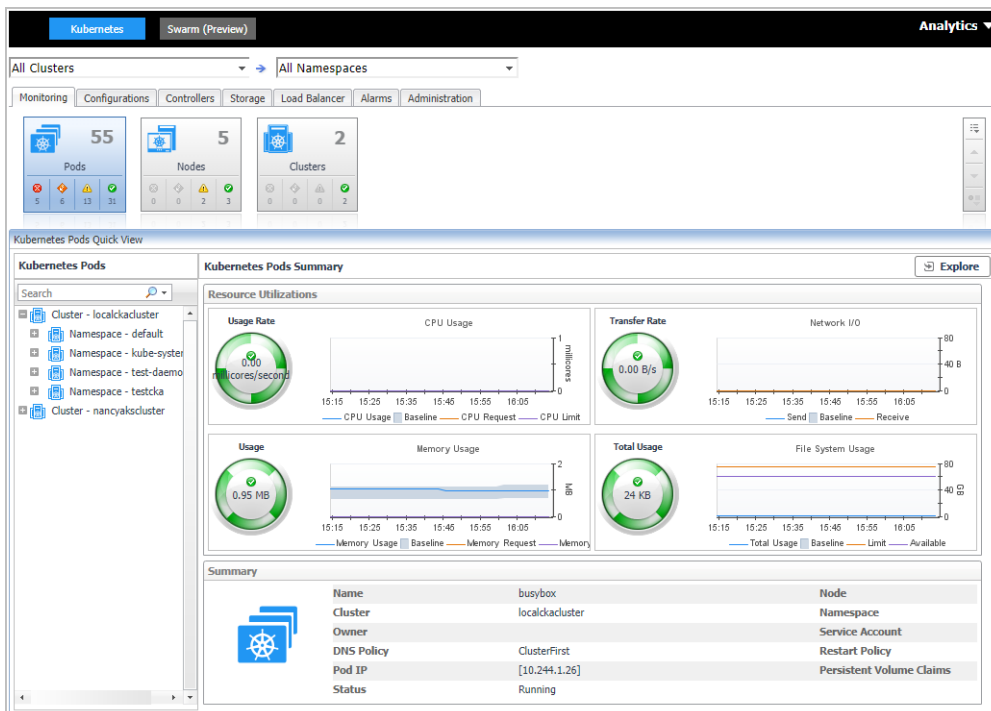
Foglight for Container Management supports Kubernetes Agent and Docker Swarm Agent.

- [Creating and Activating a Kubernetes Agent](#)
- [Creating and Activating a Docker Swarm Agent](#)

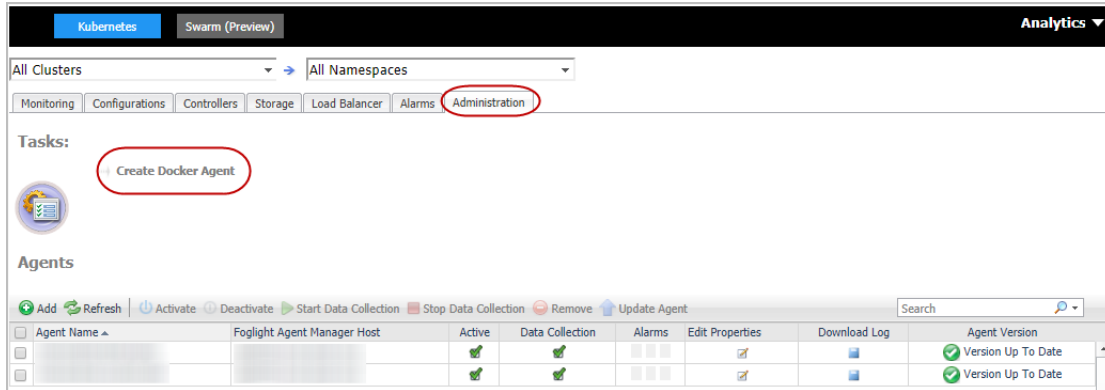
## Creating and Activating a Kubernetes Agent

**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.



- In the Container dashboard, click **Administration** tab, and then click **Create Docker Agent**. The **Create Docker Agent** wizard opens.



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

The screenshot shows the 'Create Docker Agent' wizard, specifically the 'Orchestration and Agent Manager' step. The left sidebar has a tree view with 'Orchestration and Agent Manager' selected, and sub-items for 'Agent Properties', 'Credential Verification', and 'Summary'. The main area contains the following fields:

- Cluster Name:** A text input field with the value 'nancyakcluster'.
- Agent Manager:** A dropdown menu with a single visible option.
- Orchestration Type:** A dropdown menu with the value 'Kubernetes'.

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

- Agent Properties**

The screenshot shows the 'Create Docker Agent' wizard, specifically the 'Agent Properties' step. The left sidebar has a tree view with 'Agent Properties' selected, and sub-items for 'Orchestration and Agent Manager', 'Credential Verification', and 'Summary'. The main area contains the following fields:

- Agent Name:** A text input field with the value 'Monitor@nancyakcluster'.
- Kubernetes API Service End Point:** A text input field with a placeholder value.
- Kubernetes Version:** A text input field with the value '1.7'.
- Heapster Service Namespace:** A text input field with the value 'kube-system'.
- Heapster Service Name:** A text input field with the value 'heapster'.
- Enable Proxy:** A checkbox that is checked.
- Proxy Type:** A dropdown menu with the value 'HTTP'.
- Proxy Server Address:** A text input field with a placeholder value.
- Proxy Server Port:** A text input field with the value '8081'.
- Collector Configuration:** A dropdown menu with the value '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>`.
- **Enable Proxy:** To enable the proxy, select the checkbox. Enter the *Proxy Server Address* and *Proxy Server Port* information.

## 6 Credential Verification

- Do not configure a credential: click **Next**.
- **Add cluster to a new credential.**
  - **Credential Type:** **Kube Config** is chosen by default. Click **Next**.

Create Docker Agent

- ✓ Orchestration and Agent Manager
- ✓ Agent Properties
- ▶ **Credential Verification**
  - ▶ **Credential Type**
  - Credential Properties
  - Credential Name and Lockbox

**Credential Type**

Select the credential type from the list below.

Credential Type	Description
<input checked="" type="radio"/> 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**.

Create Docker Agent

- ✓ Orchestration and Agent Manager
- ✓ Agent Properties
- ▶ **Credential Verification**
  - ✓ **Credential Type**
  - ▶ **Credential Properties**
  - Credential Name and Lockbox
  - Resource Mapping
  - Policies
  - Summary

**Credential Properties**

Enter the properties for this credential.

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

Kube Config

```
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data:
  LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUV5RENDQXIDZ0F3SUJBZ0lSQXwa
  MnFOOFU3TFp4MnZvZnF6NmNGL1I3RFFZSkvWkldodmNOQVFTEJRQXcKRRFRFE1Ba
  OdBMVVFQXhNlQ1kyRXdtAGNOTVRneE1ERTVNRGMwTORVMFdoY05NakF4TURFNE1E
  YzBPRFJwV2p8TgpNlXN3Q1FZRFZRUURFd0pqWVRDQ0FpSXdEUVIKS29aSWh2Y05B
  UUVVCQlFBRGdnSVBRENDRQWdvQ2dnSUJBTGh6CnR1ek1tUUp1M1M1WtZmx5CTQzR1BE
  NXZ2Z20hTODYzTldQU0hRTZUWElzeFF45WQZfJ1K2c4WGdvQW13eDRLN0QKeSs1d
  EZwRzhwTkNNM1hVlpsbEZnRzFrndpKZkFCdHlaYzRRT21ZWUY3MlBZeGE5S3BlTY1
  bnAxZGROU2Q0MQ0M1NTcGU3RFB6eWNFcmxYN3d1aVdWc2pnebnOXVIZnk5VFla
  eTBGOXlPR2NmahhMVHewNk9OeWV3R1JBRnpvCnM4Y0x2aTdWVfJuUmRTWJDFeJRH
  NzRsZihsejV0SzhFaExLQ21jTlplckZyMjhpSGRhTERWeTZTY21ja2tzcFAKTUdVkrRyVWF
  Haxh1dkRpQ1kxYVxwEwG9pUFB1Yk1vdlF4WkN3KzdyRmR1WTZOMU9UMXh2TmFsUvU1
```

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

Create Docker Agent

- ✓ Orchestration and Agent Manager
- ✓ Agent Properties
- ▶ **Credential Verification**
  - ✓ **Credential Type**
  - ✓ **Credential Properties**
  - ▶ **Credential Name and Lockbox**
  - Resource Mapping
  - Policies
  - Summary

**Credential Name and Lockbox**

These properties identify the credential on the Management Server.

Please provide a unique name to identify this credential.

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

Lockbox	Password Required
<input checked="" type="radio"/> System	No

- **Resource Mapping:** click **Next**.

Create Docker Agent

- Orchestration and Agent Manager
- Agent Properties
- Credential Verification**
  - Credential Type
  - Credential Properties
  - Credential Name and Lockbox
  - Resource Mapping**
  - Policies

**Resource Mapping**

Select the appropriate resource mapping option below.

☒ Provided host name (only)  
☐ All hosts  
☐ Hosts with name containing

Host Name: eusckubec-euscluster-a4d903-e68d03a0.hcp.eastus.az  
mk8s.io

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

Create Docker Agent

- Orchestration and Agent Manager
- Agent Properties
- Credential Verification**
  - Credential**
  - Resource Mapping
  - Summary

**Credential**

Select the Credential that will contain this resource.

Credential	Password Required
System	No
nancyakskubeconfig	

- Resource Mapping: click **Next**.

Create Docker Agent

- Orchestration and Agent Manager
- Agent Properties
- Credential Verification**
  - Credential Type
  - Credential Properties
  - Credential Name and Lockbox
  - Resource Mapping**
  - Policies

**Resource Mapping**

Select the appropriate resource mapping option below.

☒ Provided host name (only)  
☐ All hosts  
☐ Hosts with name containing

Host Name: eusckubec-euscluster-a4d903-e68d03a0.hcp.eastus.az  
mk8s.io

- 7 Summary: click **Finish**.

Create Docker Agent

- Orchestration and Agent Manager
- Agent Properties
- Credential Verification
- Summary**

**Summary**

Cluster	nancyakskcluster
Agent Manager	...
Orchestration Type	Kubernetes
Agent Name	Monitor@nancyakskcluster
Kubernetes API Service End Point	https://192.168.1.10:443
Kubernetes Version	1.7
Heapster Service Namespace	kube-system
Heapster Service Name	heapster
Enable Proxy	true
Proxy Type	HTTP
Proxy Server Address	...
Proxy Server Port	8,081
Collector Configuration	defaultSchedule
Credential	kubeconfig.nancyakskcluster
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	Type	Tags
testagent	192.168.1.100	DockerSwarmAgent	

This agent is currently using properties for DockerSwarmAgent agents.  
⚙️ Modify properties for this agent only.  
⚙️ Modify the properties for all DockerSwarmAgent agents.

**Configuration**

Name:

Host Name:

Docker Remote API End Point:

**Swarm**

Swarm Name:

**Data Collection Scheduler**

Collector Config:

- *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.
- *Swarm Name*: specify the swarm cluster name for display. If the swarm name is kept as “default”, then the cluster name will be displayed as “default (cluster ID)” on the dashboard. If a customized name is input here, then the customized cluster name will be displayed on the dashboard.

**i** | **NOTE:** Ensure that the docker host inside the same cluster has the same configuration for Swarm Name.

- 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.

**i** | **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**.



# Using Foglight for Container Management

- [Kubernetes](#)
  - [Monitoring Kubernetes Pods](#)
  - [Monitoring Kubernetes Nodes](#)
  - [Monitoring Kubernetes Clusters](#)
  - [Monitoring Kubernetes Other Components](#)
  - [Alarms](#)
- [Docker Swarm](#)
  - [Monitoring Docker Containers](#)
  - [Monitoring Docker Hosts](#)
  - [Monitoring Docker Swarm Clusters](#)
  - [Monitoring Docker Swarm Services](#)
  - [Alarms](#)
- [Analytics](#)
  - [Kubernetes analytics](#)
    - [Heatmap analytics](#)
    - [Scatter Plot analytics](#)
  - [Docker Swarm analytics](#)
    - [Heatmap analytics](#)
    - [Scatter Plot analytics](#)
- [Metrics](#)
  - [Kubernetes metrics](#)
  - [Docker Swarm metrics](#)

## 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:

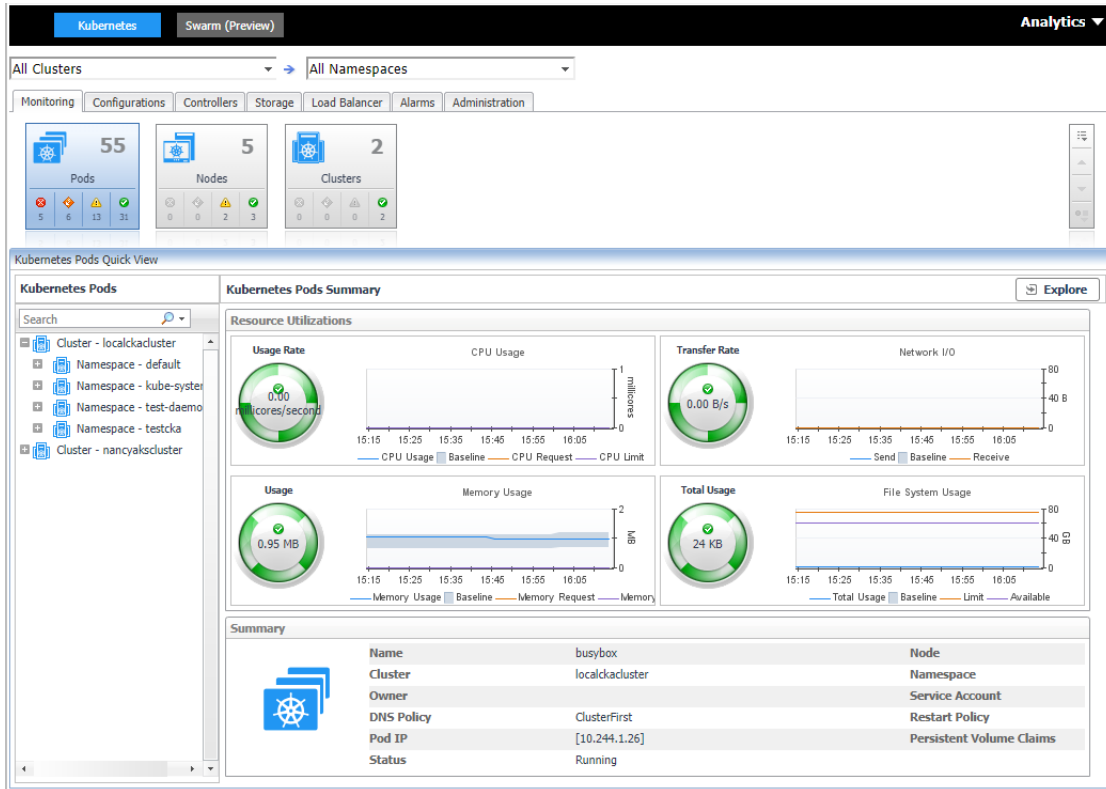


- The **Kubernetes Pods** tree view, which appears on the left of *Kubernetes Pods Quick View*, lists the pods existing in the monitored Kubernetes environment.
- The [Kubernetes Pods Summary view](#), which appears on the right after you select an individual pod in the **Kubernetes Pods** tree view.

## 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



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*.

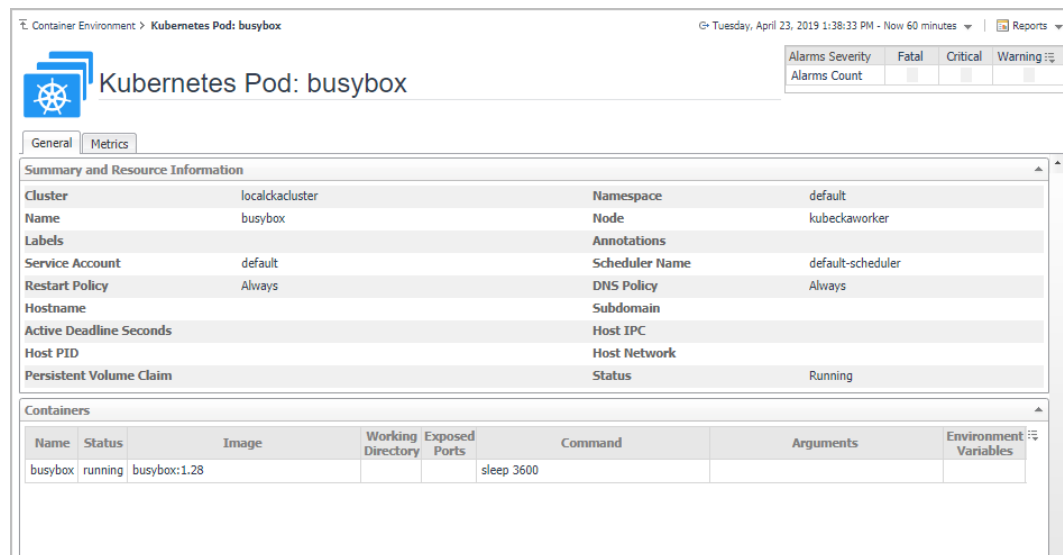
Click **Explore** on the upper right of the **Kubernetes Pods Summary** view to open the **Pods Explorer view**, which shows more detailed information about this Kubernetes cluster.

## Pods Explorer view

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

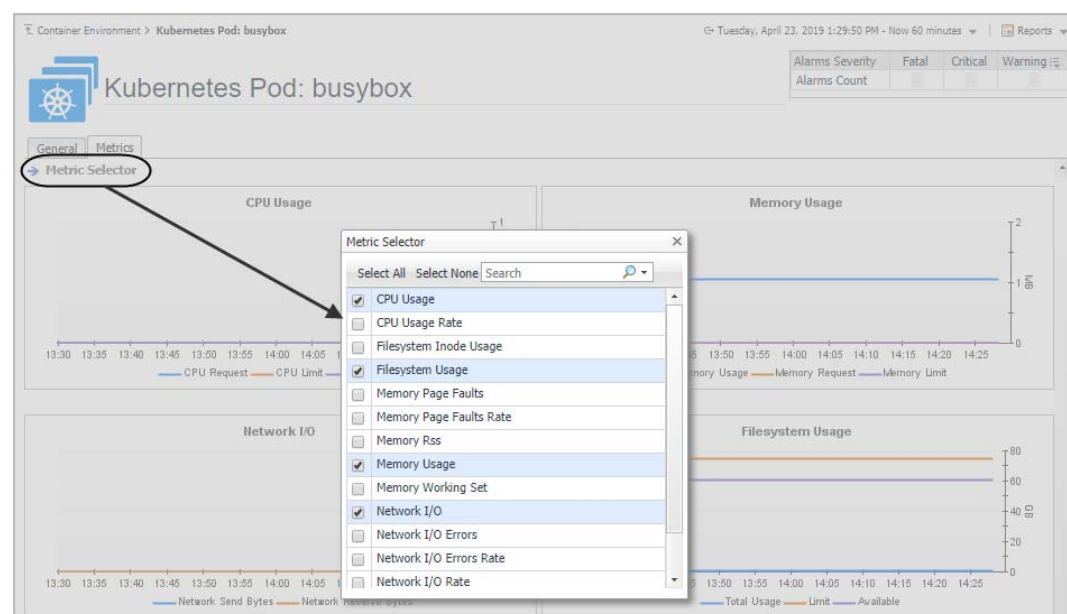
- *General tab*: The *General* tab displays the overall information of the selected Kubernetes Pod over a selected period of time, including the *Summary and Resource Information* table, the *Containers* table, and the *Init Containers* table. For more information, see [Pod metrics on page 52](#).

**Figure 3. Kubernetes Pods Explorer view General Tab**



- *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**



# Monitoring Kubernetes Nodes

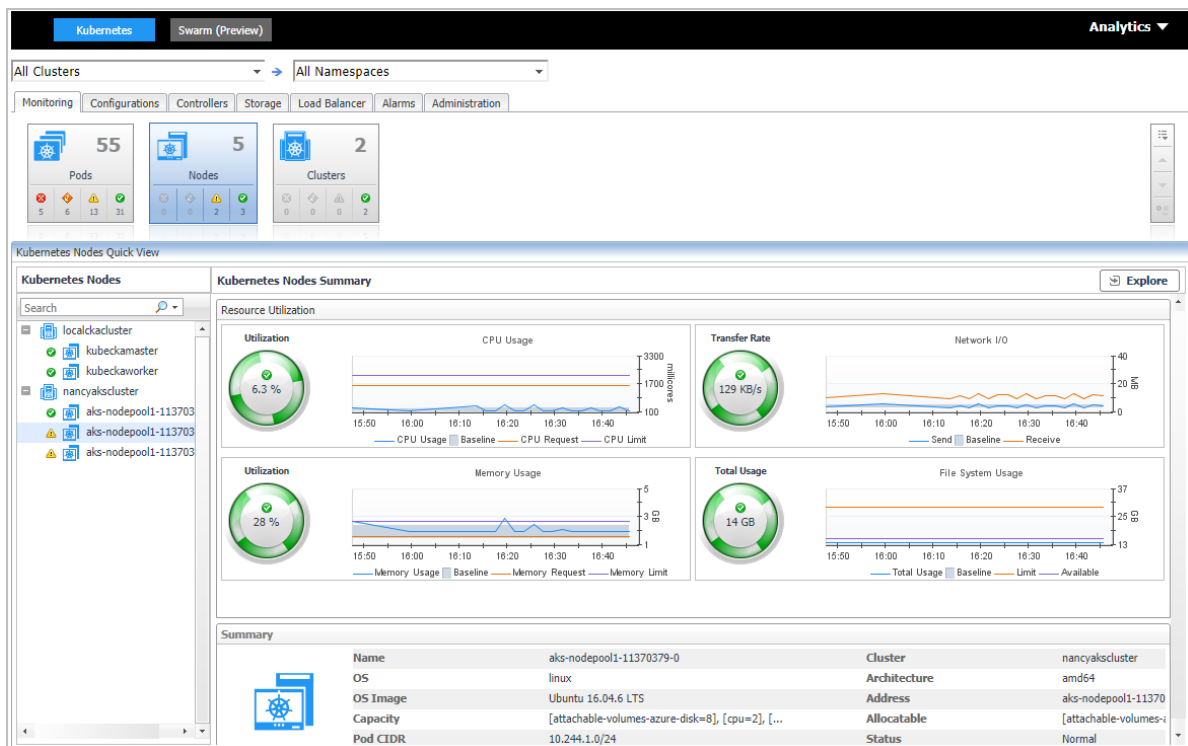
A node, previously known as a minion, is a worker machine in Kubernetes. A node may be a VM or physical machine, depending on the cluster. Each node has the services necessary to run pods and is managed by the master components. The *Kubernetes Nodes Quick View*, which appears after clicking **Monitoring > 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



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

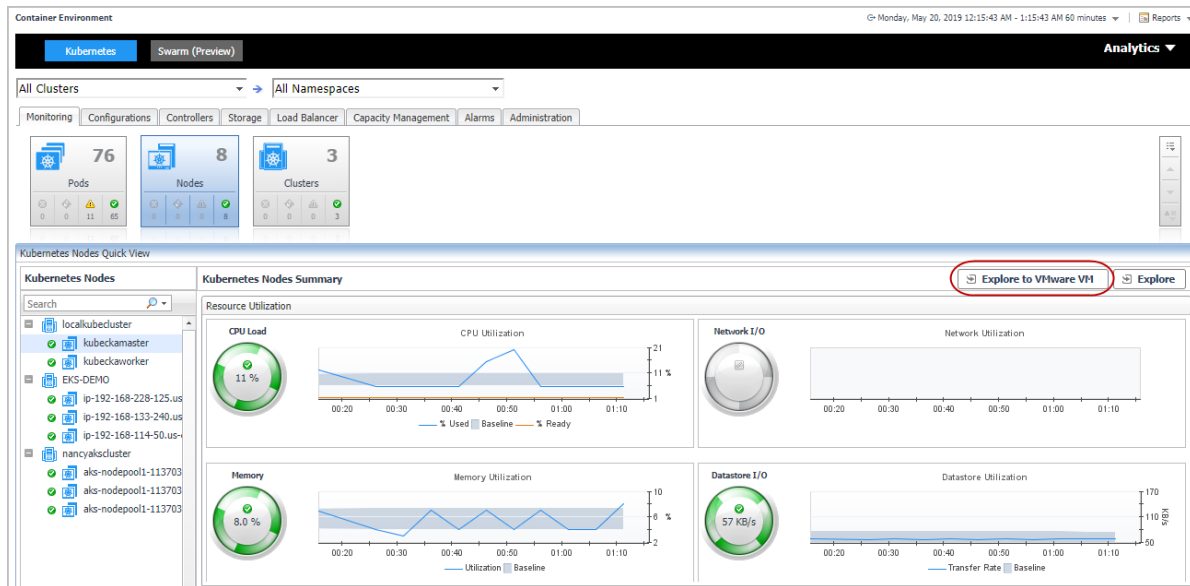
- **Resource Utilizations:** The resource utilization for the selected Kubernetes node over a selected period of time, which includes the following:
  - **CPU Utilization:** Shows the CPU utilization summary for the selected Kubernetes node based on its total capacity during a selected time period.
  - **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.

**NOTE:** If the virtual machines belong to the Container cluster, and they are monitored by other cartridges at the same time, for example, VMware, AWS, or Azure. Then, the collected data will come from that cartridge directly and the **Kubernetes Nodes Summary** view will be different from the screenshot above. See [Figure 6](#) for detailed information.

Figure 6. Kubernetes Nodes Summary view for VMware



- **Explore to VMware VM:** Click the button to open the *VMware Explorer* view, which is the same view from VMware cartridge. The *Explore to xx* button varies from the cartridge that is monitoring the machines. Currently, the supported cartridges include: VMware, AWS, and Azure.
- **Resource Utilizations:** The displayed metrics will be slight different among different monitoring cartridges.

## 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 53.

Container Environment > Kubernetes Node: aks-nodepool1-11370379-0
Tuesday, April 23, 2019 1:40:35 PM - Now 60 minutes Reports

## Kubernetes Node: aks-nodepool1-11370379-0

Alarms Severity	Fatal	Critical	Warning
Alarms Count			1

General
Metrics

### Summary and Resource Information

Name	aks-nodepool1-11370379-0	Kubelet Endpoint	10.250
Architecture	amd64	Operating System	linux
Kernel Version	4.15.0-1041-azure	Container Runtime Version	docker://3.0.4
Kube Proxy Version	v1.12.6	Kubelet Version	v1.12.6
Boot ID	f3ad1377-088d-42e8-a625-c37f25e58841	Machine ID	31e6dcdd55b47faa9ad59ce858bcf18
Pod CIDR	10.244.1.0/24	External ID	
Provider ID	azure:///subscriptions/a4d9037c-b740-4205-ad1...	Unschedulable	false
Labels	[beta.kubernetes.io/os=linux], [kubernetes.azure.com/role=node-agent]	Annotations	[volumes.kubernetes.io/controller-managed-attach-detach=true]
Capacity	[attachable-volumes-azure-disk=8], [cpu=2], [...]	Allocatable	[attachable-volumes-azure-disk=8], [cpu=1.9999999999999999]
Status	Normal		
Addresses			
Hostname	aks-nodepool1-11370379-0	InternalIP	10.240.0.4

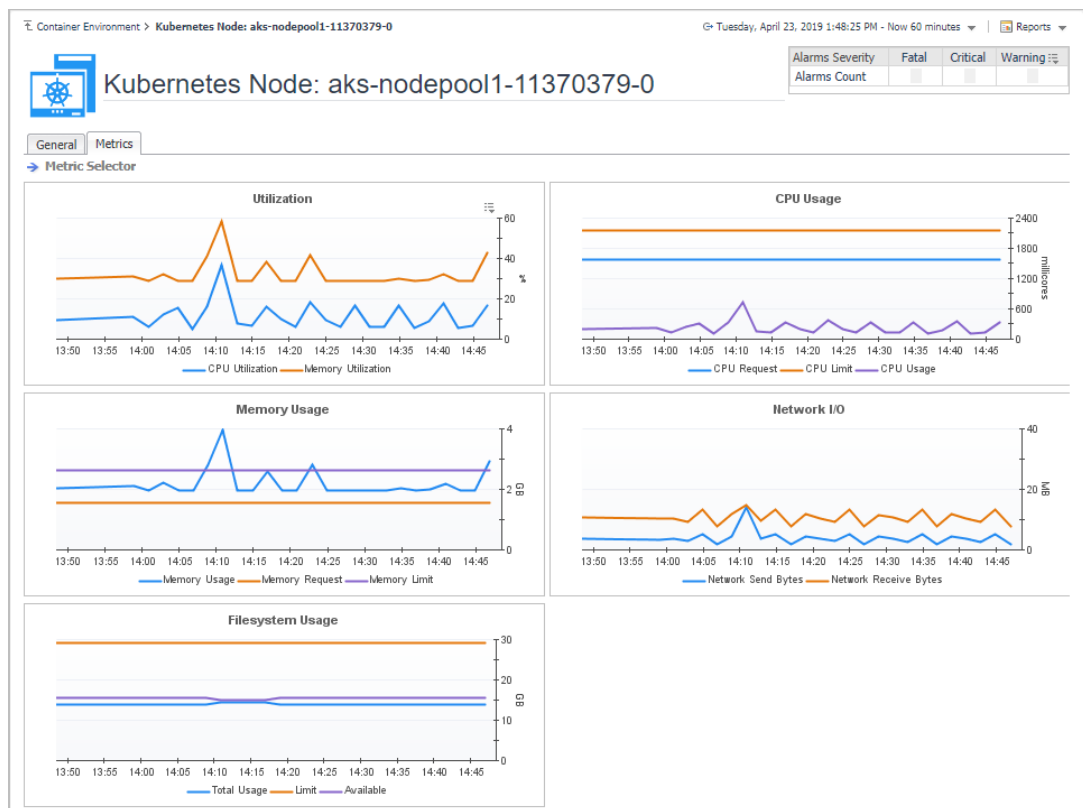
### Pods

Search

Name	Status	Cluster	Namespace	Containers	Init Containers
kibana-logging-7444956bf8-fnmps	Running	nancyakscluster	kube-system	1	0
tunnelfront-86695c8b65-776xp	Running	nancyakscluster	kube-system	1	0
coredns-79c89b8f4-vkfsh	Running	nancyakscluster	kube-system	1	0
fve-768d8c4678-6cbtw	Running	nancyakscluster	questfve	1	0
kube-proxy-rstrq	Running	nancyakscluster	kube-system	1	0
kube-svc-redirect-bqb9q	Running	nancyakscluster	kube-system	2	0
coredns-79c89b8f4-glvrb	Running	nancyakscluster	kube-system	1	0
elasticsearch-2	Failed	nancyakscluster	elasticsearch	0	0
heapster-6d879b9dc8-qycmc	Running	nancyakscluster	kube-system	2	0
invalidimage-bb487f87-55xhc	Pending	nancyakscluster	test	1	0

- Foglight for Container Management 1.2.0 User and Administration Guide  
Using Foglight for Container Management

Figure 8. Kubernetes Nodes Explorer view Metrics Tab



## Monitoring Kubernetes Clusters

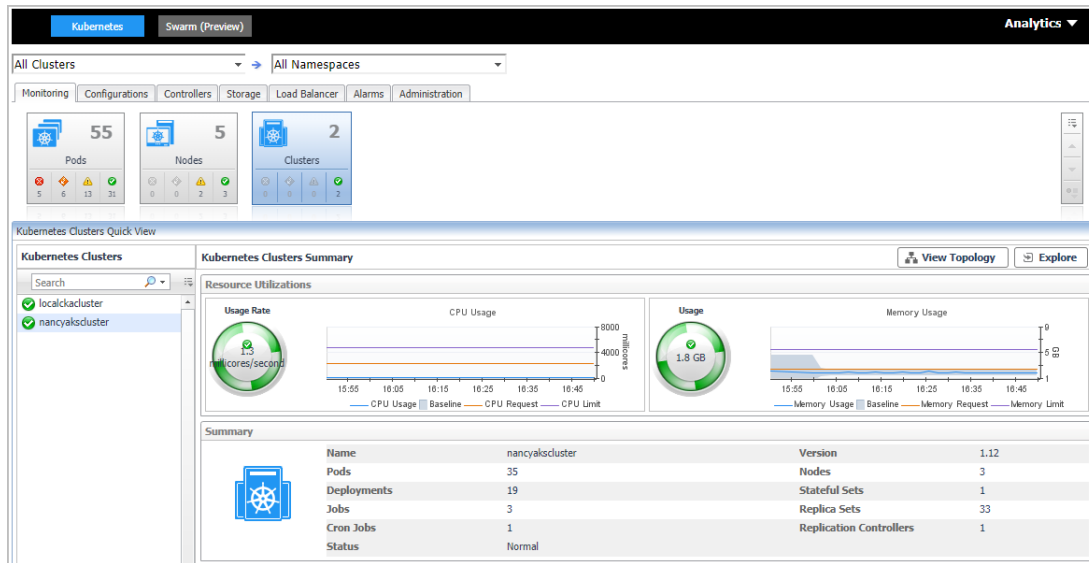
Kubernetes cluster is a group of kubernetes resources. There are two kinds of nodes inside a cluster, Kubernetes master and Kubernetes nodes. Kubernetes master is responsible for maintaining the desired state of your cluster which Kubernetes node is responsible to run your application and cloud workflows. 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 9. Kubernetes Clusters Summary view



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.

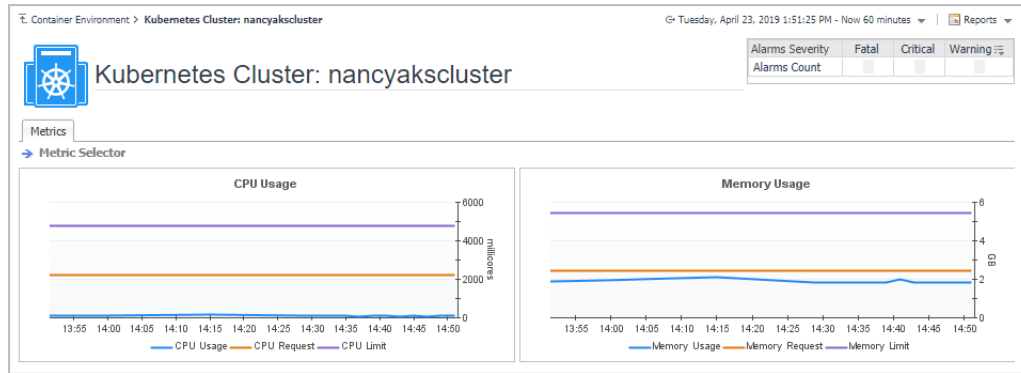
Click **View Topology** on the upper right of the **Kubernetes Clusters Summary** view to open the [Cluster Topology view](#), which shows the topology graph from the application accessible aspect.

## Cluster Explorer view

The *Cluster Explorer* view opens when you click **Explore** in the [Kubernetes Clusters Summary view](#), which includes the following tabs:

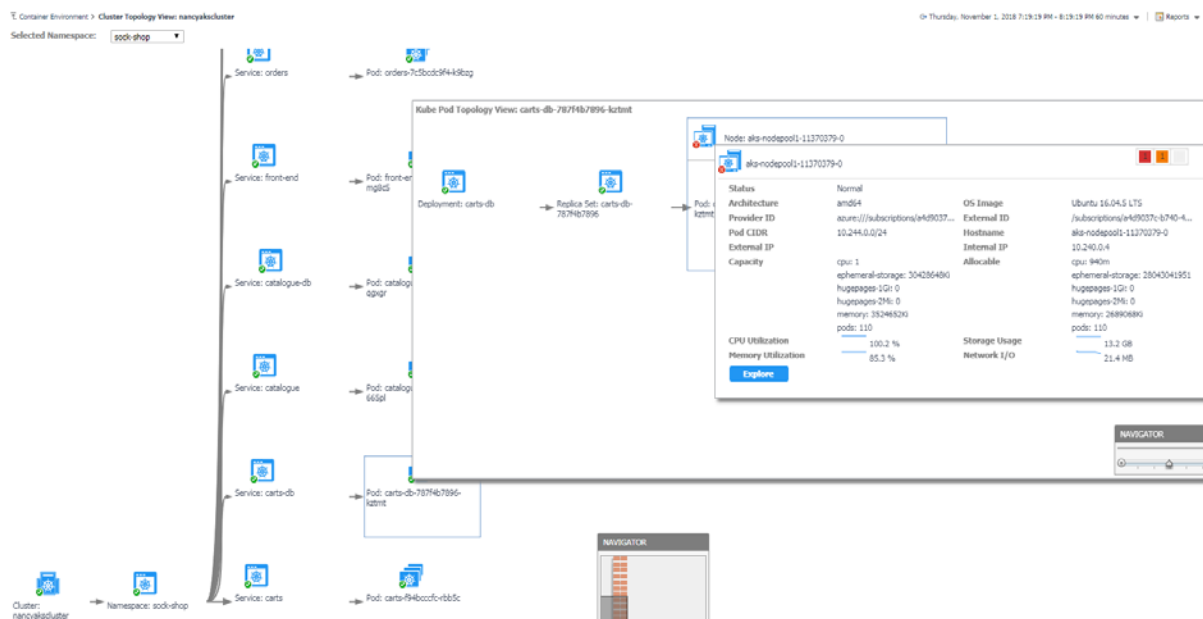
- **Metrics tab:** The *Metrics* tab displays a *Metric Selector* allowing you to choose the metrics to be plotted on this dashboard. Charts of *CPU Usage* and *Memory Usage* are presented by default.

Figure 10. Kubernetes Clusters Explorer view Metrics tab



## Cluster Topology view

Figure 11. Kubernetes Clusters Topology view



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 12. Kubernetes Configuration Dashboard

Name	Cluster	Namespace	Labels	Annotations	Configured Data Keys
cluster-info	localk8cluster	kube-public			kubeconfig
coredns	localk8cluster	kube-system			Corefile
extension-apiserver-authentication	localk8cluster	kube-system			client-ca-file, requestheader-extra-headers-prefix, requestheader-die...
kube-flannel-cfg	localk8cluster	kube-system	[app=flannel], [tier=node]		net-conf.json, cni-conf.json
kube-proxy	localk8cluster	kube-system	[app=kube-proxy]		config.conf, kubeconfig.conf
kubeadm-config	localk8cluster	kube-system			ClusterStatus, ClusterConfiguration
kubelet-config-1.13	localk8cluster	kube-system			kubelet
metrics-server-config	localk8cluster	kube-system	[addonmanager.kubernetes.io/m...		NannyConfiguration
fair-lambkin-elasticsearch-curator-config	nancyakcluster	default	[heritage=Tiller], [app=fair-lamb...		config.yml, action_file.yml
impressive-llama-mariadb-master	nancyakcluster	default	[heritage=Tiller], [app=mariadb...		my.cnf
impressive-llama-mariadb-slave	nancyakcluster	default	[component=slave], [release=im...		my.cnf
impressive-llama-mariadb-tests	nancyakcluster	default			run.sh
metricbeat-config	nancyakcluster	default	[k8s-app=metricbeat], [app=fair...		metricbeat.yml
metricbeat-modules	nancyakcluster	default	[component=fair-lambkin-elastic...		system.yml, kubernetes.yml
sysdig-agent	nancyakcluster	default		[kubecti.kubernetes.io/last-appli...	dragent.yml
understood-zebra-elasticsearch-curator-config	nancyakcluster	default	[release=understood-zebra], [he...		action_file.yml, config.yml
aks-nodepool1-11370379-0-config-5fgt4dhcbf	nancyakcluster	kube-system			kubelet

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 13. Kubernetes Controllers Dashboard

Kubernetes

Swarm (Preview)

Analytics

All Clusters

All Namespaces

Monitoring

Configurations

Controllers

Storage

Load Balancer

Alarms

Administration

Deployment

Replica Set

Replication Controller

Daemon Set

Stateful Set

Job

Cron Job

Search

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	fglam	localkcacluster	default	1	1	1	False		2,147,483,647	2,147,483,647		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	five-app	localkcacluster	default	1	1	2	False	5	600	10		[RollingUpdate]Max Surge:25%,Max Unavailable:25%
	Normal	postgresql	localkcacluster	default	1	1	1	False		2,147,483,647	2,147,483,647		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	coredns	localkcacluster	kube-system	2	2	1	False		600	10		[RollingUpdate]Max Surge:25%,Max Unavailable:1
	Normal	heapster	localkcacluster	kube-system	1	1	1	False		2,147,483,647	2,147,483,647		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Abnormal	metrics-server-v0.3.1	localkcacluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:25%,Max Unavailable:25%
	Normal	[Status Detail]									10		[RollingUpdate]Max Surge:25%,Max Unavailable:25%
	Normal	Progressing:True, Reason:NewReplicaSetAvailable, Message:ReplicaSet "metrics-server-v0.3.1-9455848d4c" has successfully progressed.									10		[RollingUpdate]Max Surge:25%,Max Unavailable:25%
	Normal	Available:False, Reason:MinimumReplicasUnavailable, Message:Deployment does not have minimum availability.									10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	[Replicas Detail]									10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	Replicas:1									10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	Unavailable Replicas:1									10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	alternating-jackal-nginx-ingress-default-backend	nancyakcluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	coredns	nancyakcluster	kube-system	2	2	2	False		2,147,483,647	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	coredns-autoscaler	nancyakcluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	heapster	nancyakcluster	kube-system	1	1	2	False		2,147,483,647	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	kibana-logging	nancyakcluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:25%,Max Unavailable:25%
	Normal	killed-unicorn-nginx-ingress-controller	nancyakcluster	kube-system	2	2	1	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	killed-unicorn-nginx-ingress-default-backend	nancyakcluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	kubernetes-dashboard	nancyakcluster	kube-system	1	1	4	False		600	10		[RollingUpdate]Max Surge:0,Max Unavailable:1
	Normal	metrics-server	nancyakcluster	kube-system	1	1	4	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	tiller-deploy	nancyakcluster	kube-system	1	1	1	False		600	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	tunnelfront	nancyakcluster	kube-system	1	1	2	False		2,147,483,647	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Abnormal	invalidimage	nancyakcluster	test	2	2	1	False		2,147,483,647	10		[RollingUpdate]Max Surge:1,Max Unavailable:1
	Normal	unabletoschedule	nancyakcluster	test	1	1	1	False		2,147,483,647	10		[RollingUpdate]Max Surge:1,Max Unavailable: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*, *Replica Set*, *Replication Controller*, *Daemon Set*, *Stateful Set*, *Job*, and *Cron Job*.

## Storage

Figure 14. Kubernetes Storage Dashboard

Kubernetes

Swarm (Preview)

Analytics

All Clusters

All Namespaces

Monitoring

Configurations

Controllers

Storage

Load Balancer

Alarms

Administration

Persistent Volume

Persistent Volume Claim

Storage Class

Search

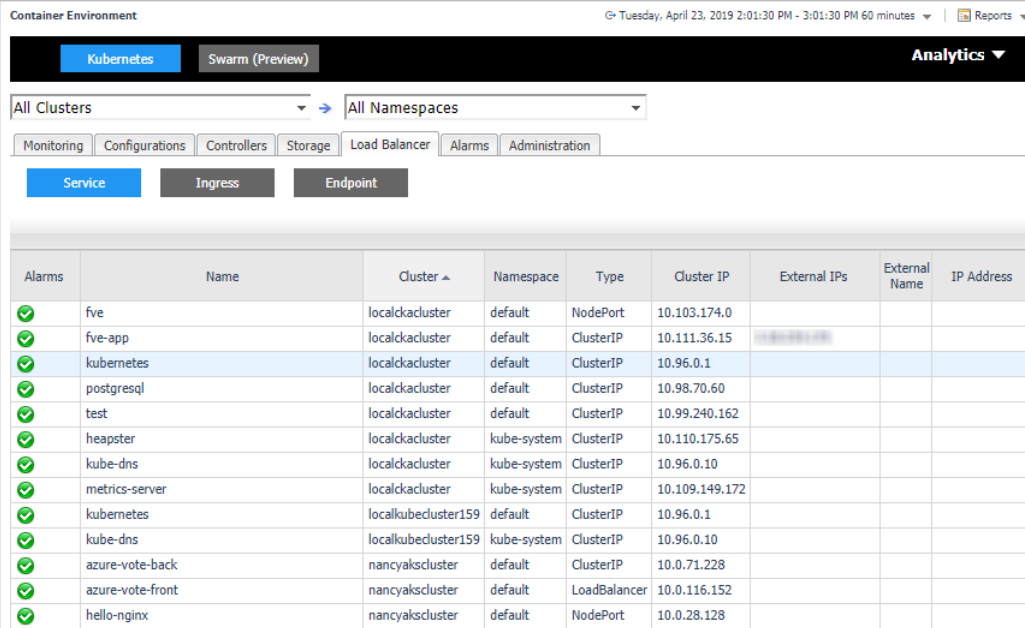
Alarms	Status	Name	Cluster	Reclaim Policy	Claim	Storage Class	Source Type
✓	Bound	pvc-sc-no-sc-customize	localkcacluster	Retain	pvc-sc-pv-customize-sc	no-sc-customize	HostPath
✓	Available	pvc-sc-default	localkcacluster	Retain		default	HostPath
✓	Available	pvc-sc-invalid-provisioner	localkcacluster	Retain		sc-invalid-provisio...	HostPath
✓	Bound	pvc-invalid-nfs	localkcacluster	Recycle	pvc-invalid-sc-pv	slow	NFS
✓	Available	pvc-pvc-oversize	localkcacluster	Retain		sc-oversize	HostPath
✓	Available	pvc-pvc-acm1	localkcacluster	Retain		sc-pvc-acm1	HostPath
✓	Bound	pvc-2b95e22d-dc28-11e8-b2ed-befa22179703	nancyakcluster	Delete	data-mehdb-1	default	AzureDisk
✓	Bound	pvc-45f1fe1e-5f54-11e9-b660-16063de8b09f	nancyakcluster	Delete	data-elasticsearch-2	default	AzureDisk
✓	Bound	pvc-59cb23a5-fd17-11e8-adf4-de8994810bc3	nancyakcluster	Delete	data-elasticsearch-0	default	AzureDisk
✓	Bound	pvc-70499cb8-fd17-11e8-adf4-de8994810bc3	nancyakcluster	Delete	data-elasticsearch-1	default	AzureDisk
✓	Bound	pvc-90f76a94-2e94-11e9-810c-0a130f143c9f	nancyakcluster	Delete	alertmanager-prom-prometheus-operator-alertmanager-db-al...	default	AzureDisk
✓	Bound	pvc-9727fba6-2e94-11e9-810c-0a130f143c9f	nancyakcluster	Delete	prometheus-prom-prometheus-operator-prometheus-db-prom...	default	AzureDisk

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 15. Kubernetes Load Balancer Dashboard

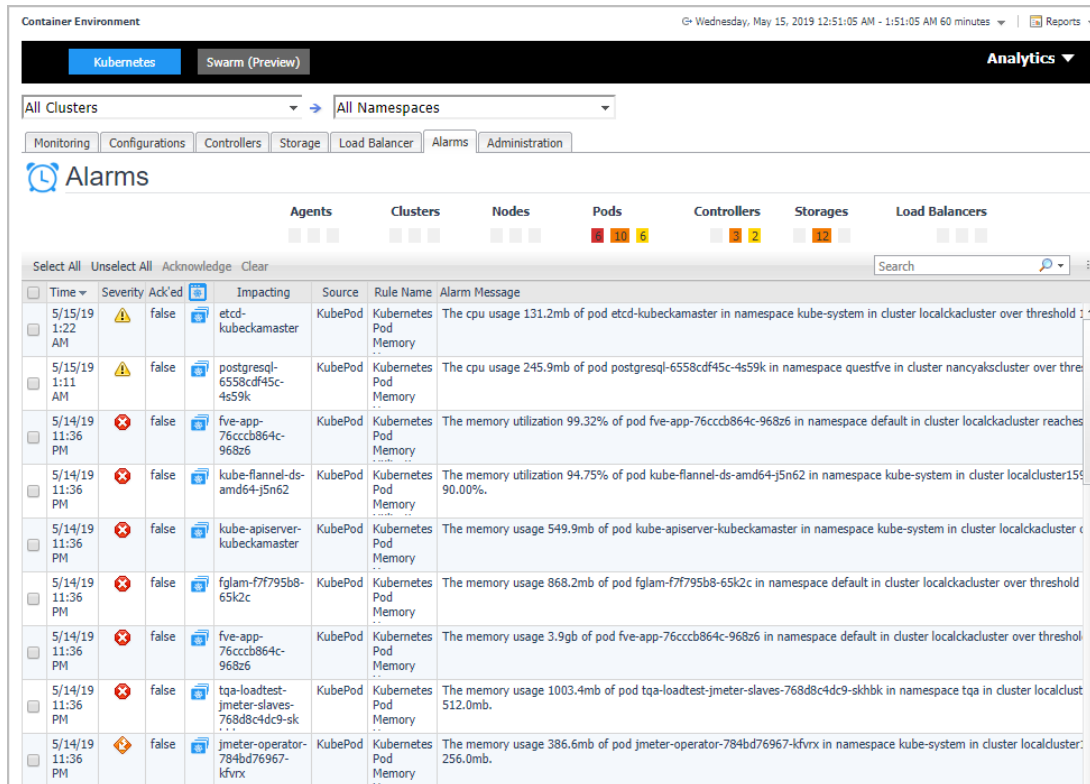


Alarms	Name	Cluster	Namespace	Type	Cluster IP	External IPs	External Name	IP Address
✓	fve	localkcluster	default	NodePort	10.103.174.0			
✓	fve-app	localkcluster	default	ClusterIP	10.111.36.15			
✓	kubernetes	localkcluster	default	ClusterIP	10.96.0.1			
✓	postgresql	localkcluster	default	ClusterIP	10.98.70.60			
✓	test	localkcluster	default	ClusterIP	10.99.240.162			
✓	heapster	localkcluster	kube-system	ClusterIP	10.110.175.65			
✓	kube-dns	localkcluster	kube-system	ClusterIP	10.96.0.10			
✓	metrics-server	localkcluster	kube-system	ClusterIP	10.109.149.172			
✓	kubernetes	localkubecuster159	default	ClusterIP	10.96.0.1			
✓	kube-dns	localkubecuster159	kube-system	ClusterIP	10.96.0.10			
✓	azure-vote-back	nancyakcluster	default	ClusterIP	10.0.71.228			
✓	azure-vote-front	nancyakcluster	default	LoadBalancer	10.0.116.152			
✓	hello-nginx	nancyakcluster	default	NodePort	10.0.28.128			

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

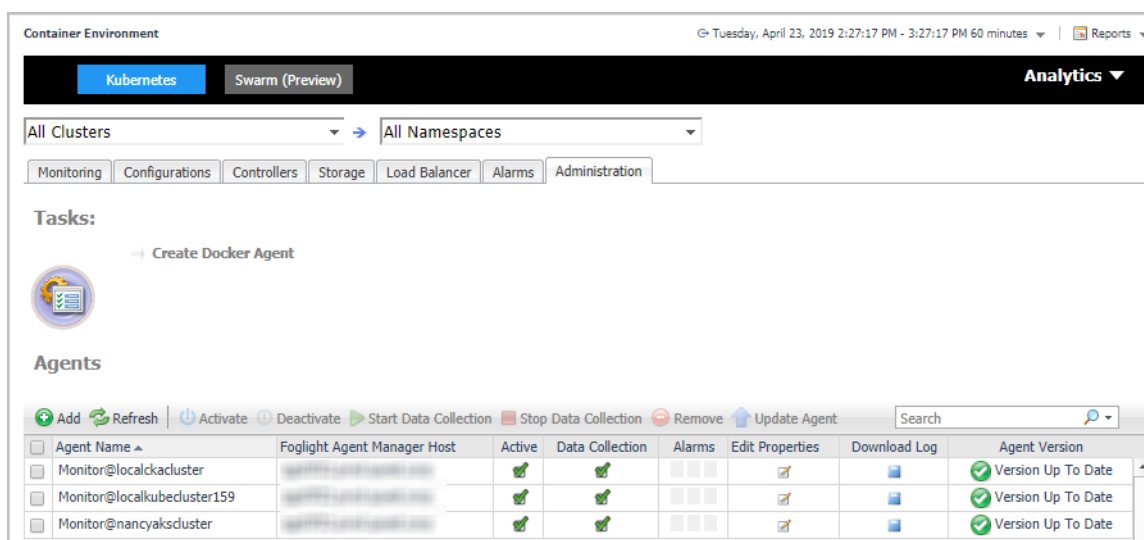
Figure 16. Kubernetes Alarms Dashboard



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.

# Administration

Figure 17. Kubernetes Administration Dashboard



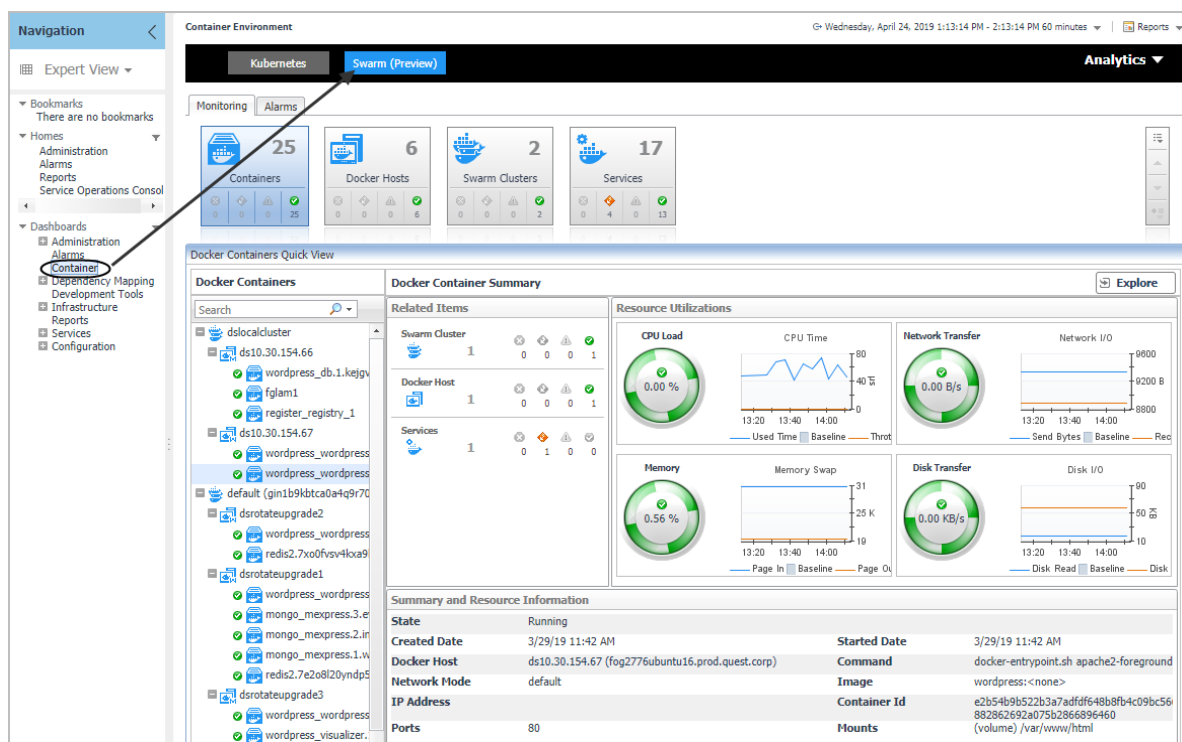
The *Administration* dashboard supports new agent creation and existing agents management. Use *Create Docker Agent* or *Add* to create a new agent. See [Creating and Activating a Kubernetes Agent](#) on page 18 for more information. Use *Activate*, *Deactivate*, *Start Data Collection*, *Stop Data Collection*, *Remove*, and *Update Agent* to manage the agent. Click *Edit Properties* to update the properties of the chosen agent.

**NOTE:** The *Administration* dashboard can be accessed only when the users have both the Administrator role and the Container Administrator role. To grant the users with the Container Administrator role, go to **Administration > Users & Security** management under *Administer Server > Manage Users, Groups, Roles > Roles* tab.

# Docker Swarm

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

Figure 18. Docker Swarm Dashboard



## Monitoring Docker Containers

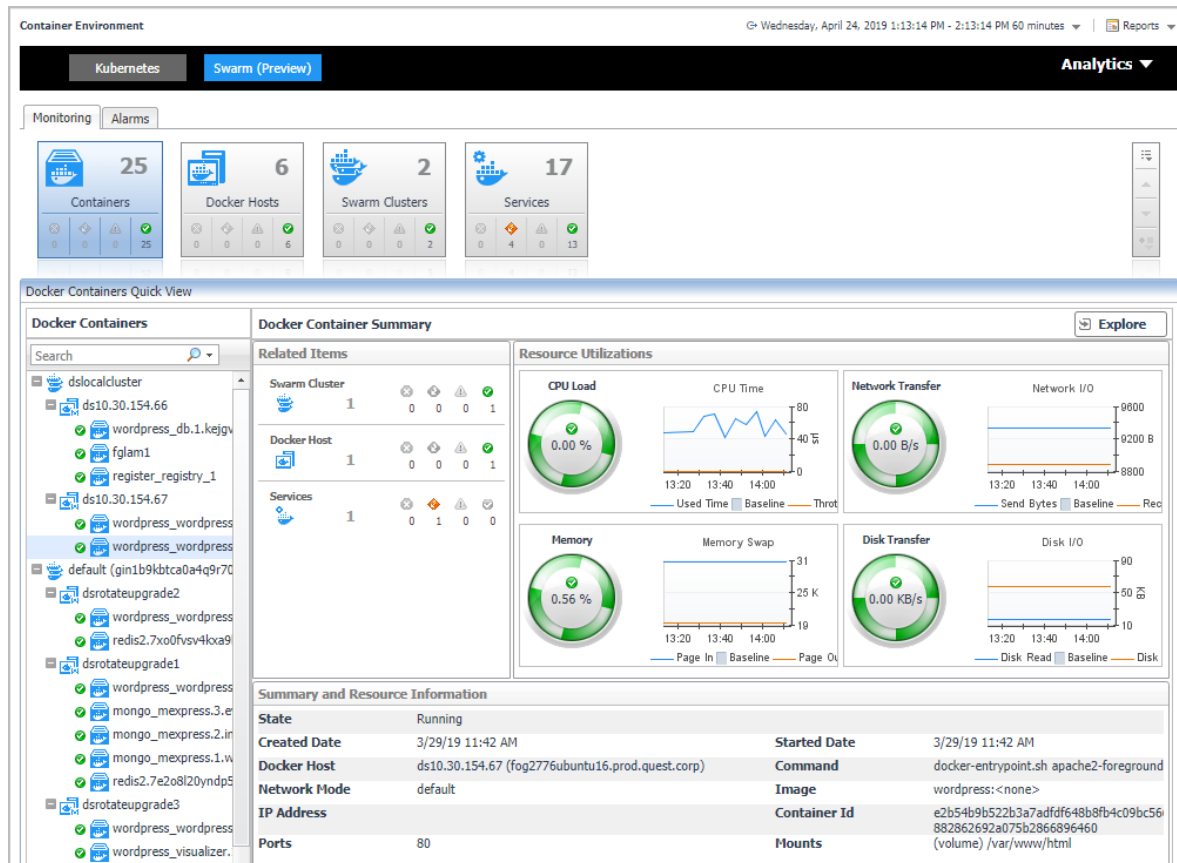
This view consists of the following two panes:

- 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 19. Docker Container Summary view



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.

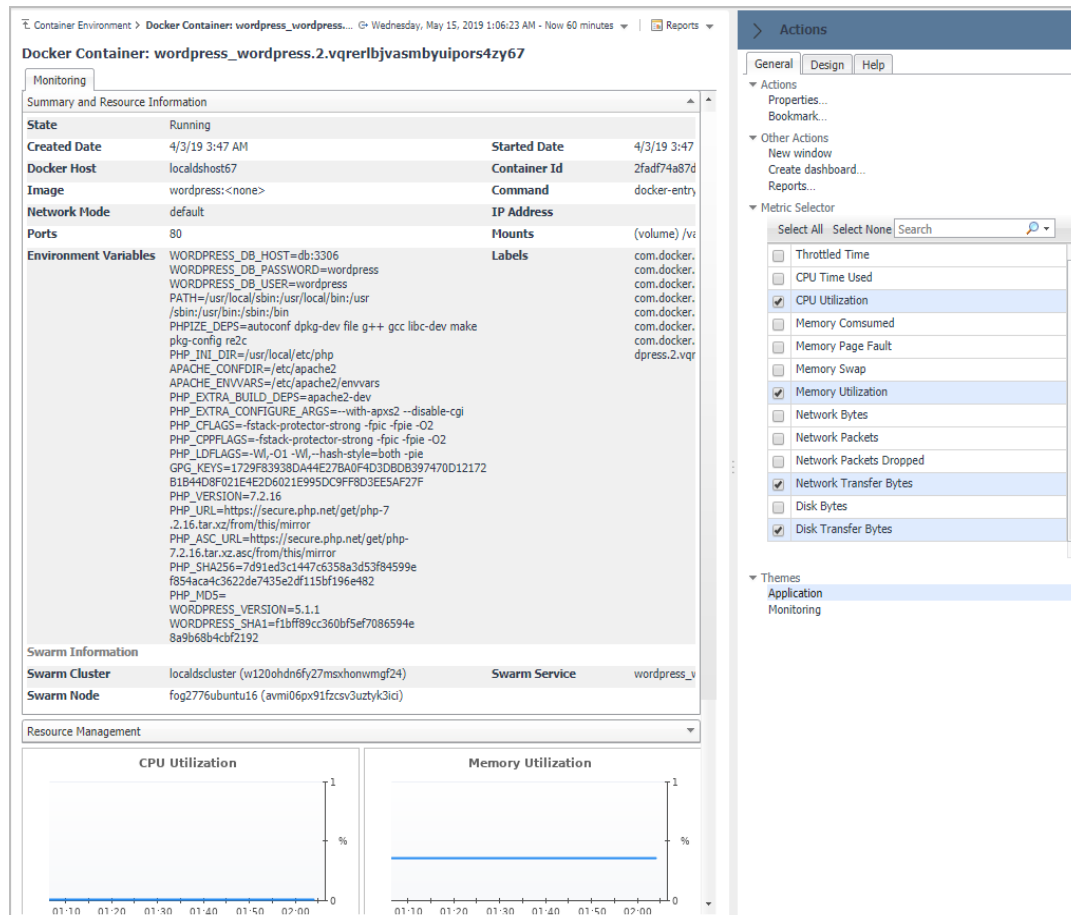
Click **Explore** on the upper right of the **Docker Container Summary** view to open the [Container Explorer](#) view, which shows more detailed information about this container.

## 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. To set the Metrics list displayed, go to **Action > General > Metric Selector**. For more information, see [Container metrics](#) on page 53.

Figure 20. Docker Container Explorer view Monitoring Tab



## Monitoring Docker Hosts

This view consists of the following two panes:

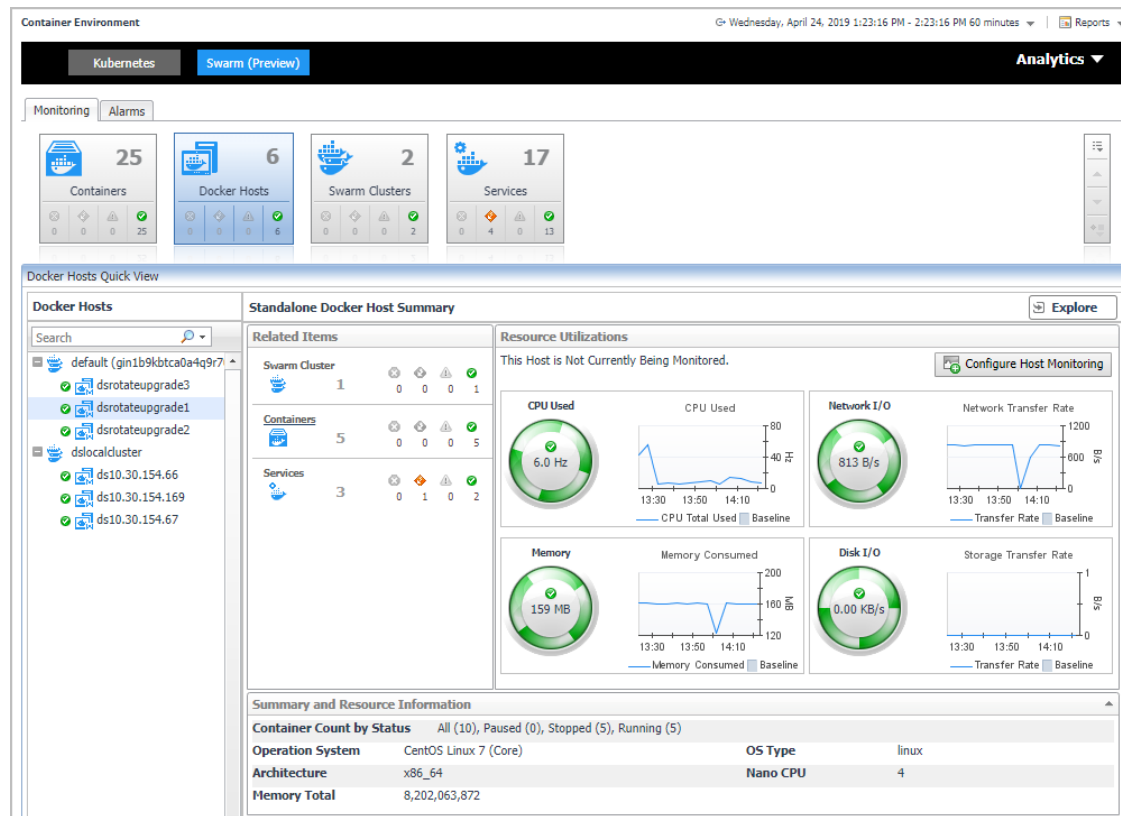
- The **Docker Hosts** tree view, which appears on the left of *Docker Hosts Quick View*, lists the docker hosts existing in the monitored *Docker* environment. The docker hosts in the tree view are grouped by **cluster > docker host**.
- The [Docker Host Summary](#) view appears on the right after you 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 21. Docker Host Summary view



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, *Containers* table, *Images* table, and *Volumes* table.

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

Figure 22. Docker Host Explorer view Monitoring Tab

Container Environment > Docker Host: dsrotateupgrade3

Thursday, April 25, 2019 10:39:16 AM - Now 60 minutes

Reports

▼

Docker Host: dsrotateupgrade3

Monitoring

Metrics

Summary and Resource Information

Container Count by Status

All (22), Paused (0), Stopped (9), Running (13)

Operation System

CentOS Linux 7 (Core)

Host

dsrotateupgrade3 (10.4.117.155)

Memory

1.8 GB

Nano CPU

1

Docker Version

18.09.2

Cgroup Driver

cgroupfs

Docker Host Swarm Information

Swarm Cluster

default (gin1b9kbtca0a4q9r70aym1je)

Swarm Role

Manager

Swarm Node

dsrotateupgrade3 (xypq2vtwnxctzdyo16cjg9v3)

Swarm Node Status

[ready]

Containers

Alarms

	Name	Id	Image	Netw Mod
✓	voting_result.1.m1ab9gkqai8lkr24oudykdt	aa25171d29a8076e9a0a076340a967927a237cccd066b6e460c4e4d1ad2979	dockersamples/examplevotingapp_result:<none>	default
✓	voting_vote.2.ua654o33tdvp7fn3pvvqo0rb	133ae0269871f9e90b894be92719216e9beef2d46328c5d849a5ace025dd4b4	dockersamples/examplevotingapp_vote:<none>	default
✓	wordpress_wordpress.3.3d78luku8t9evlq3mpt8ehr	7b6b79b24d5c7efa67e4ac707fc1efcc696daf36c6205dae4fd1645123d3dec2	wordpress:<none>	default
✓	voting_redis.1.avis611hu72gyxl5h5epjy6v	288e8a1dde3a0e893913559c7e2202e96f52a88fadfe5b862d4b6988a735d996	redis:<none>	default
✓	voting_vote.1.deivdnggt0nrlxb8qo0b8r1hd	f5eca4b5aec9294745fe742a2eb9e12ee26b13699deb4c73cc6b334d9e78bc5d	dockersamples/examplevotingapp_vote:<none>	default
✓	redis2.xypq2vtwnxctzdyo16cjg9v3.vrubg5q34jlaq8fih21mg6vex	149fa7578f49f332fed8ac292d61cd19dc8eafecdf45a1b3482201312fadf0e6	redis:<none>	default
✓	mongo_mongo.1.95ak9qe8gp158etdue4pvrwz2w	78a7e75c36a9056c9abc9eba4d636f2a1558ff236873d67bd114f21df12aad2d	mongo:<none>	default
✓	mysql_phpmymadmin.1.0c2df4k6ghuo4n463mf8q7m	f89221041f8ddcbb2c594bf9a30af3924d8b1f26700ecc794f173b52e47bd7	phpmyadmin/phpmyadmin:<none>	default
✓	wordpress_db.1.7c7rtt36ojtscpb6zr1tjoh	1711d54e6f15255a20e37685cfc85ad6a3656bae84bf9ce44cb763d29d734a2	mysql:<none>	default
✓	wordpress_visualizer.1.drldgtulabzbe5eriegdiq7	d1af858af95546c4db25fc7218dc89bfe4c99b73980d746a2eea9dd74a89719	dockersamples/visualizer:<none>	default
✓	voting_visualizer.1.nwq4fjmb8wpyvmh9wn30qddy	16b274edf0a972a313b61f785e3473571cfa6f380ef1a57991cc64fc53dddbb9	dockersamples/visualizer:<none>	default
✓	mysql_mysql.1.mroqco2fs7qm5pjk9y4bnjke	2c2718742e082060de9cb9da6d2e9dac440cdbc50ffa626ed45157698cc722	mysql:<none>	default
✓	voting_db.1.8eol3f95f710l50jrvpq4j9hu	7a60a959fb3afa27006ed3383d69e966ae25270c15029f95127c6696ae226db	postgres:<none>	default

Images

State	Name	Id	Size	Virtual Size	Comment	Containers	Not Updated Duration
✓	dockersamples/examplevotingapp_result:<none>	sha256:e10df791f13c3ac17efa123dfce57e3297fcea05a34b3bbf305749f22a9b3c83	216.0 MB	216.0 MB		1	2 minute(s)
✓	phpmyadmin/phpmyadmin:<none>	sha256:c6ba363e7c9bba3bc96aa490e31da3e26e6f7e5d8c525fb8a36df2544c2aa54	158.2 MB	158.2 MB		1	2 minute(s)
✓	mongo:<none>	sha256:0d183f48c313d863d26aed97c27a0fb73833674c87da2576b6282de2439a144c	389.5 MB	389.5 MB		1	2 minute(s)
🗑	nate/dockviz:latest	sha256:93b5259c1e18862e143ae39678640cbdd555d1b8e2742bc6f4da9c2b78acd8ab	6.3 MB	6.3 MB		0	
✓	mysql:<none>	sha256:7bb2586065cd50457e315a5dab0732a87c45c5fad619c017732f5a13e58b51dd	454.8 MB	454.8 MB		2	2 minute(s)
✓	redis:<none>	sha256:d4deec2c521cd4e0450218bd53c69611bacd2eb10838057a5de7dcb341c66cf5	144.2 MB	144.2 MB		1	2 minute(s)
✓	postgres:<none>	sha256:d7cf98b297166b40efca50ff11ef9c7e801d45a0f6c1ba316854984229667578	214.9 MB	214.9 MB		1	2 minute(s)
🗑	alpine:<none>	sha256:5cb3aa0f89934411fba5c063a9bc98ace75bdf92e77d0029543d9f2ef4ad0	5.2 MB	5.2 MB		0	





- *Containers* table: Includes the containers on this docker host.
- *Images* table: Includes the images pulled onto this docker host.
  - : Indicates this image is using by a container.
  - : Indicates no container is using this image and the image can be recycled.
- *Volumes* table: Includes the volumes created on this docker host.
  - : Indicates this volume is using by a container.
  - : Indicates no container is using this volume and the volume can be recycled.

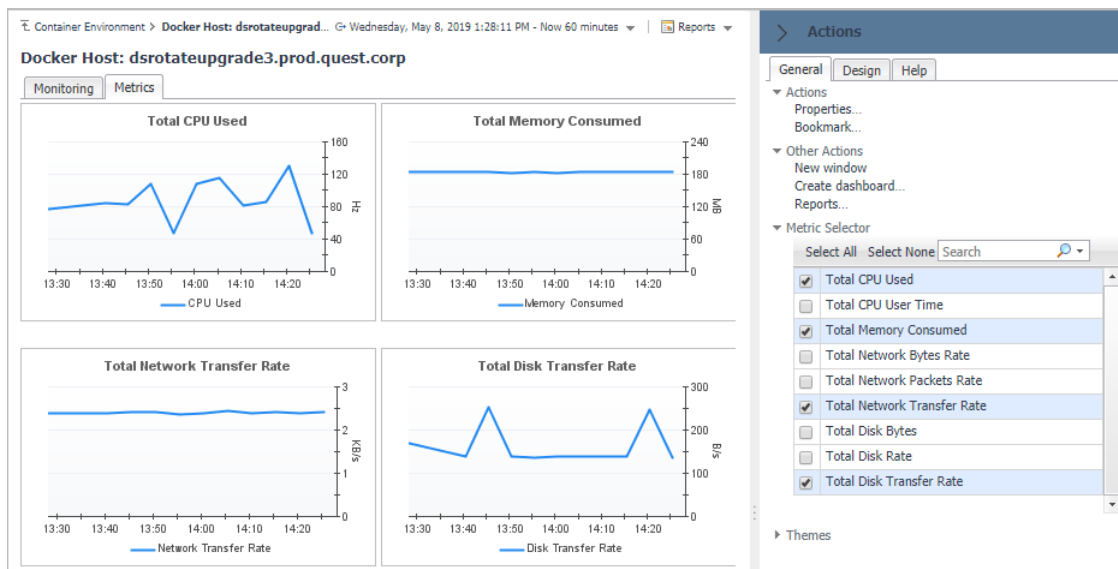
Figure 23. Docker Host Explorer view Images table and Volumes table under Monitoring tab

The screenshot shows the Docker Host Explorer interface. The top section displays the 'Images' table with columns: State, Name, Id, Size, Virtual Size, Comment, Containers, and Not Updated Duration. The bottom section displays the 'Volumes' table with columns: State, Name, Id, Size, Virtual Size, Comment, Containers, and Not Updated Duration. A modal window titled 'Docker Host Explore Containers' is open, showing a list of containers with columns: Alarms, Name, and Id. A red circle highlights the number '2' in the 'Containers' column of the Images table, which corresponds to the modal window.

By clicking the number in the *Containers* column, a *Docker Host Explore Containers* view will open, which lists the containers using this image or this volume. Click the Name or ID of the container and an explore page of the container will appear.

- **Metrics** tab: The *Metrics* tab displays the Metrics list. To set the Metrics list displayed, go to **Action > General > Metric Selector**. For more information about the description of the metrics, see [Container metrics](#) on page 53.

Figure 24. Docker Host Explorer view Metrics Tab



# Monitoring Docker Swarm Clusters

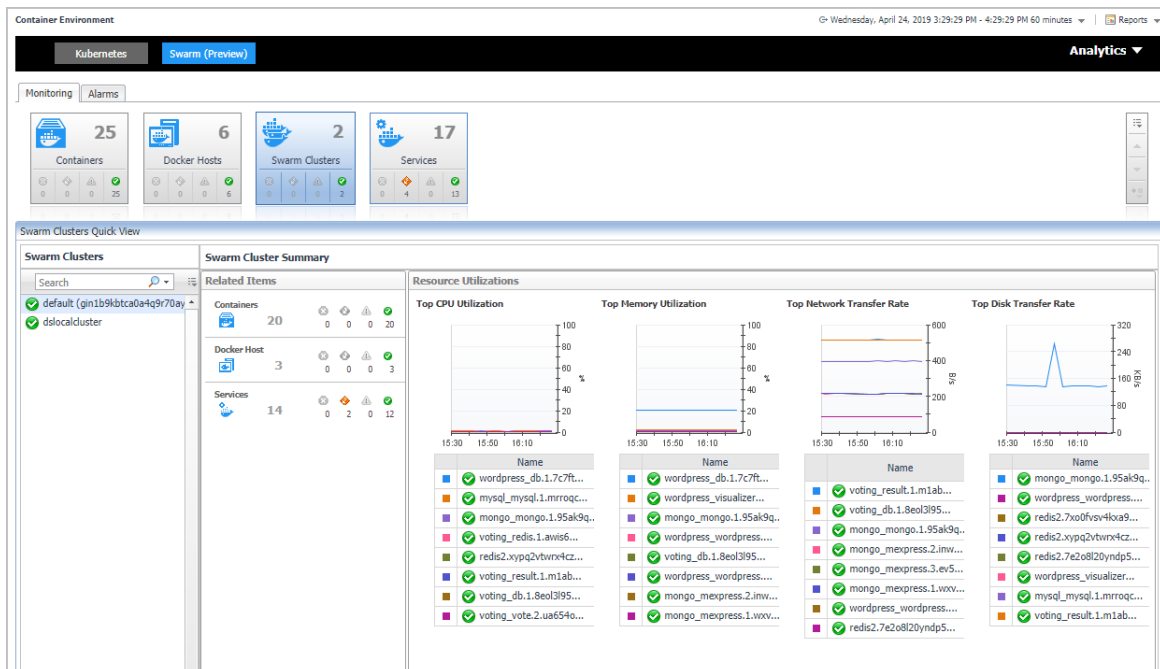
This view consists of the following two panes:

- 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 25. Docker Swarm Cluster Summary view



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

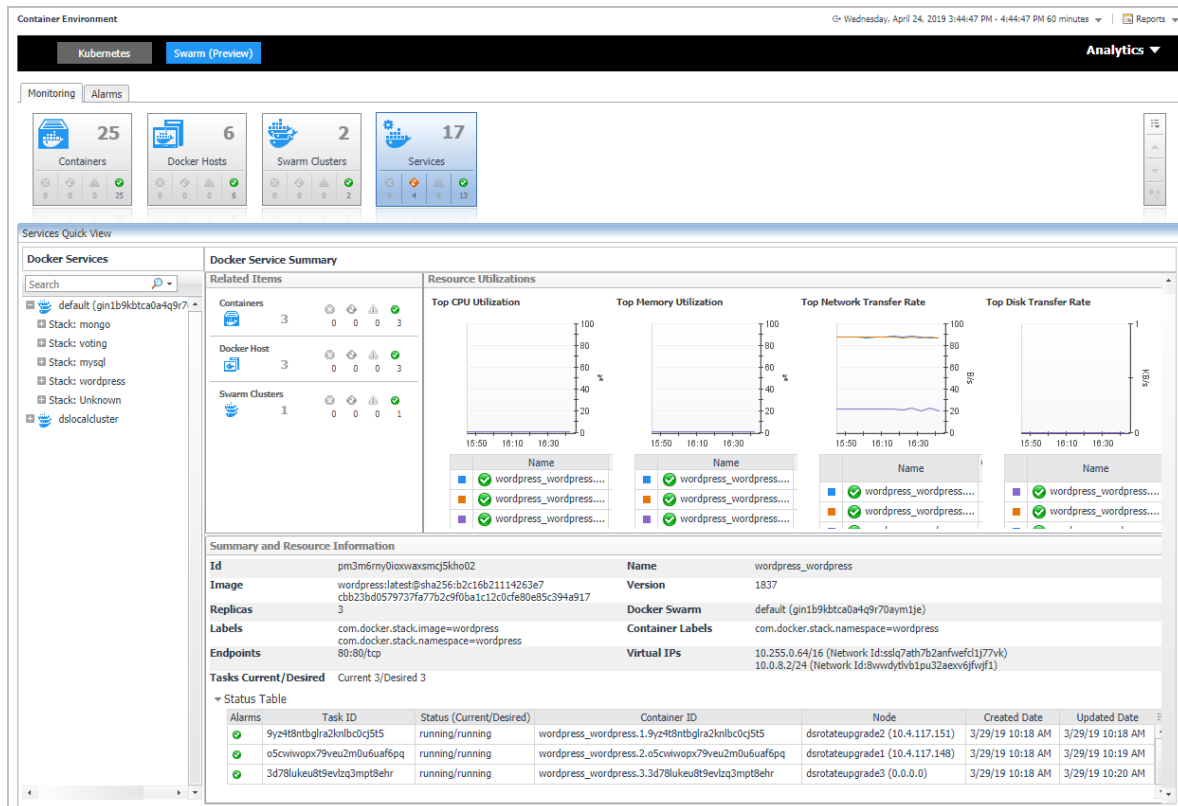
This view consists of the following two panes:

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

Figure 26. Docker Service Summary view

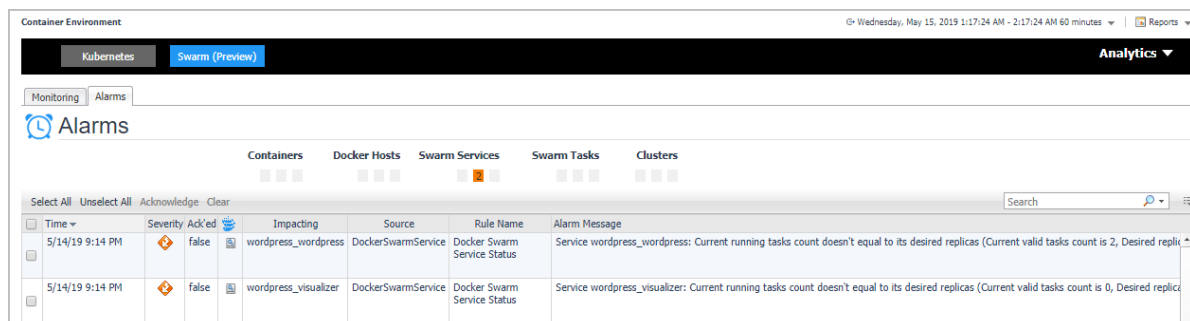


The **Docker Service Summary** view displays the following data:

- **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 27. Docker Swarm Alarms Dashboard



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.

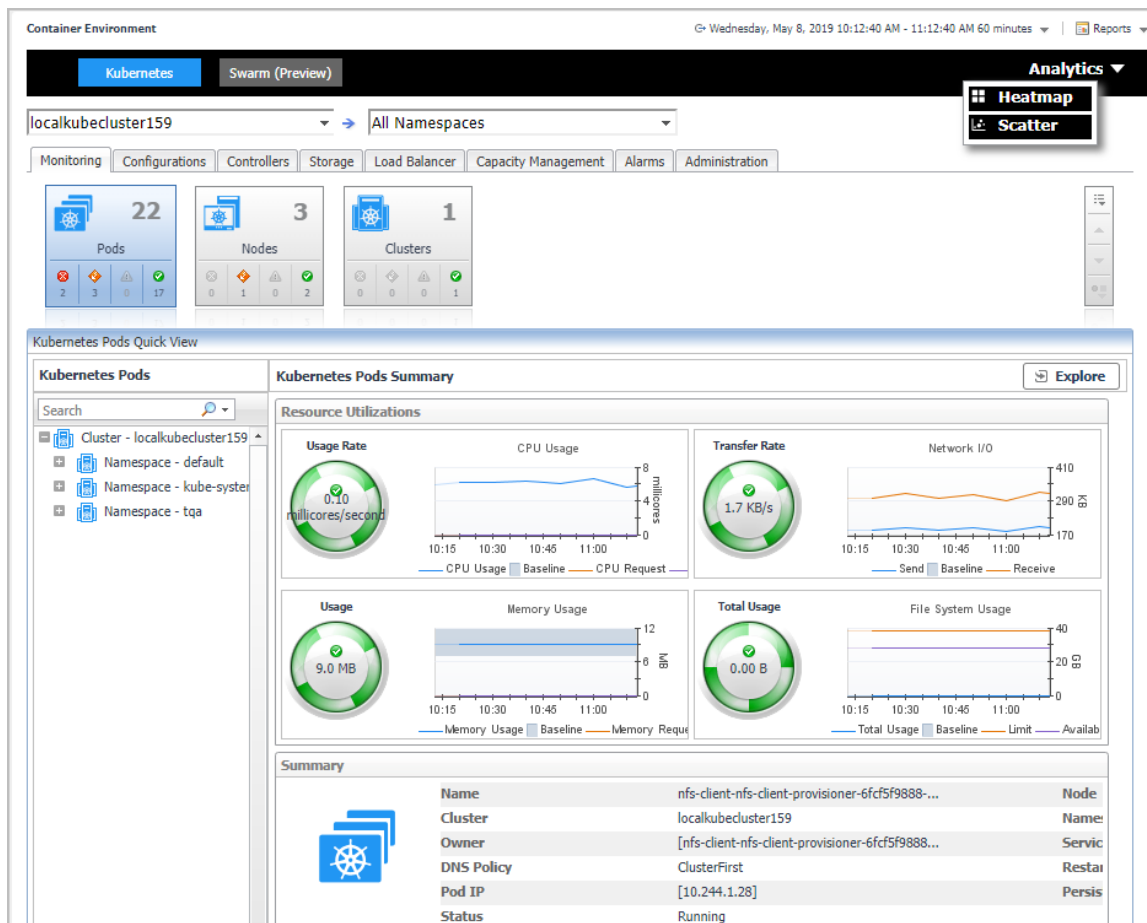
Scatter Plot is used to display values in points using two variables for a set of data. The points is color-coded also, Color Metric can be used to display one additional variable.

- [Kubernetes analytics](#)
  - [Heatmap analytics](#)
  - [Scatter Plot analytics](#)
- [Docker Swarm analytics](#)
  - [Heatmap analytics](#)
  - [Scatter Plot analytics](#)

## Kubernetes analytics

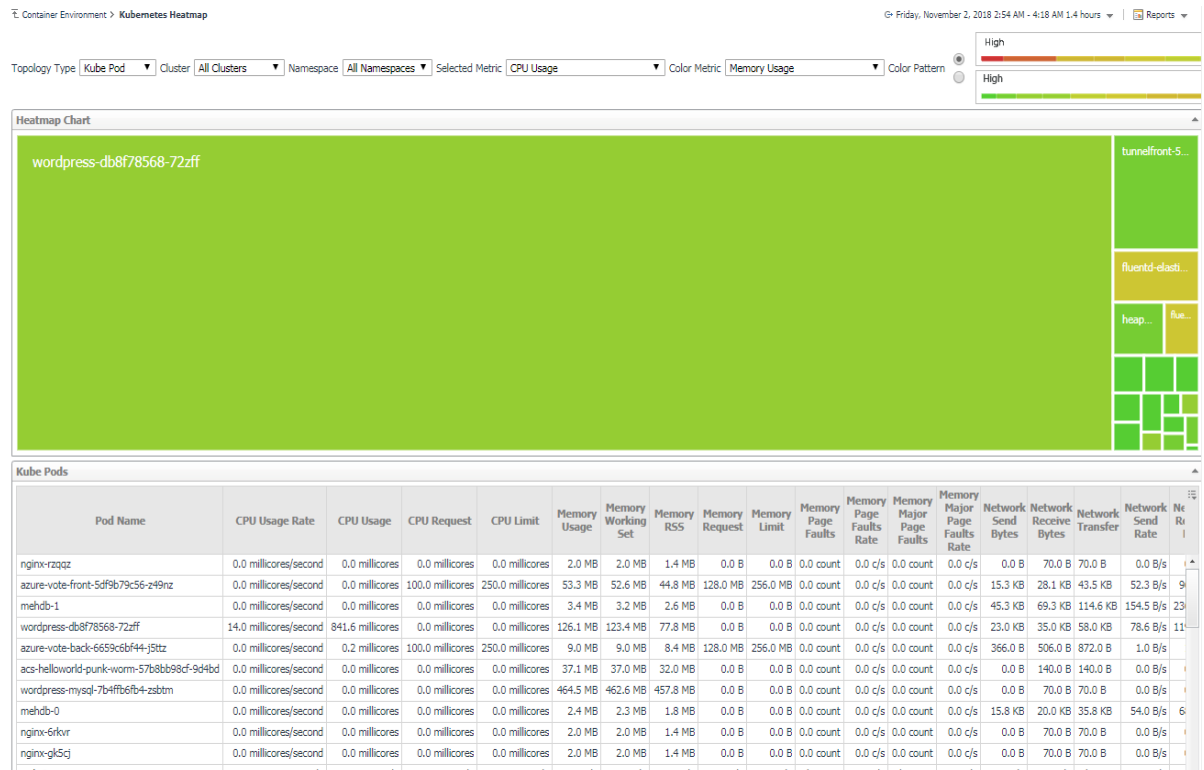
In the Container dashboard, choose **Kubernetes** from the header. Then click **Analytics** from the header, a drop down view will display with **Heatmap** and **Scatter** on it. Click **Heatmap** will navigate to the Kubernetes **Heatmap Analytics** dashboard, while click **Scatter** will navigate to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

Figure 28. Kubernetes analytics Navigation



# Heatmap analytics

Figure 29. Kubernetes Heatmap Analytics Dashboard



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 52](#).
- **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).

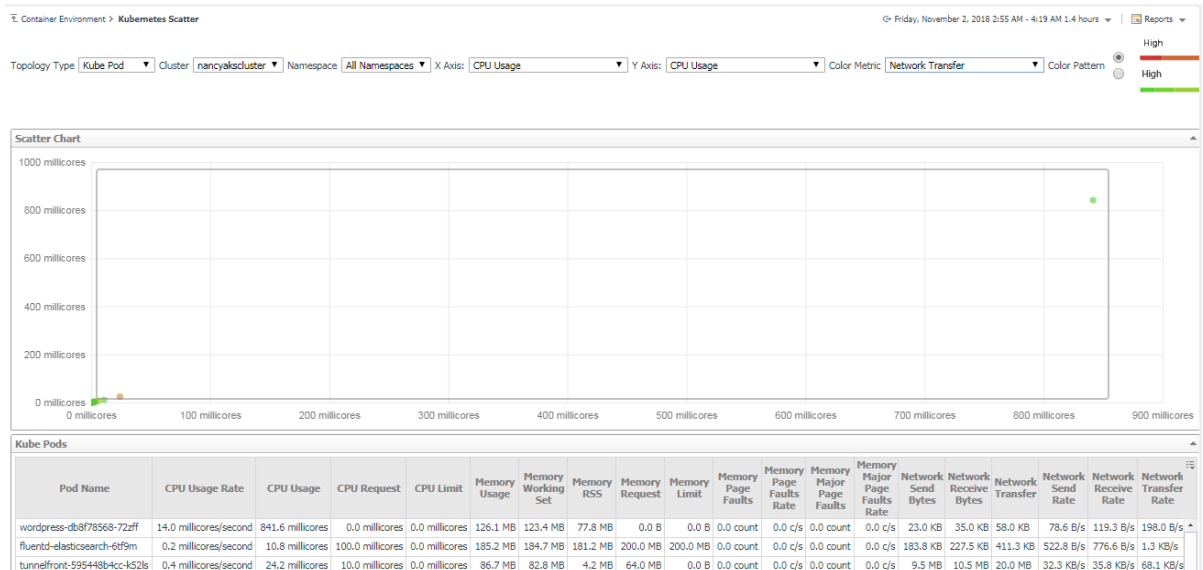
Figure 29 shows an example of heat map. This sample diagram represents the “wordpress-db8f78568-72zff” has the maximum amounts of CPU usage, while “fluentd-elastic-ef455uh68-72cfe” has a higher Memory Usage. If you switch the Color Pattern, then “wordpress-db8f78568-72zff” will turn to red. Clicking 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 52
- [Nodes Explorer view](#) on page 28
- [Node metrics](#) on page 53
- [Cluster Explorer view](#) on page 31

- [Cluster metrics](#) on page 53

## Scatter Plot analytics

Figure 30. Kubernetes Scatter Plot Analytics Dashboard



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 30 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:

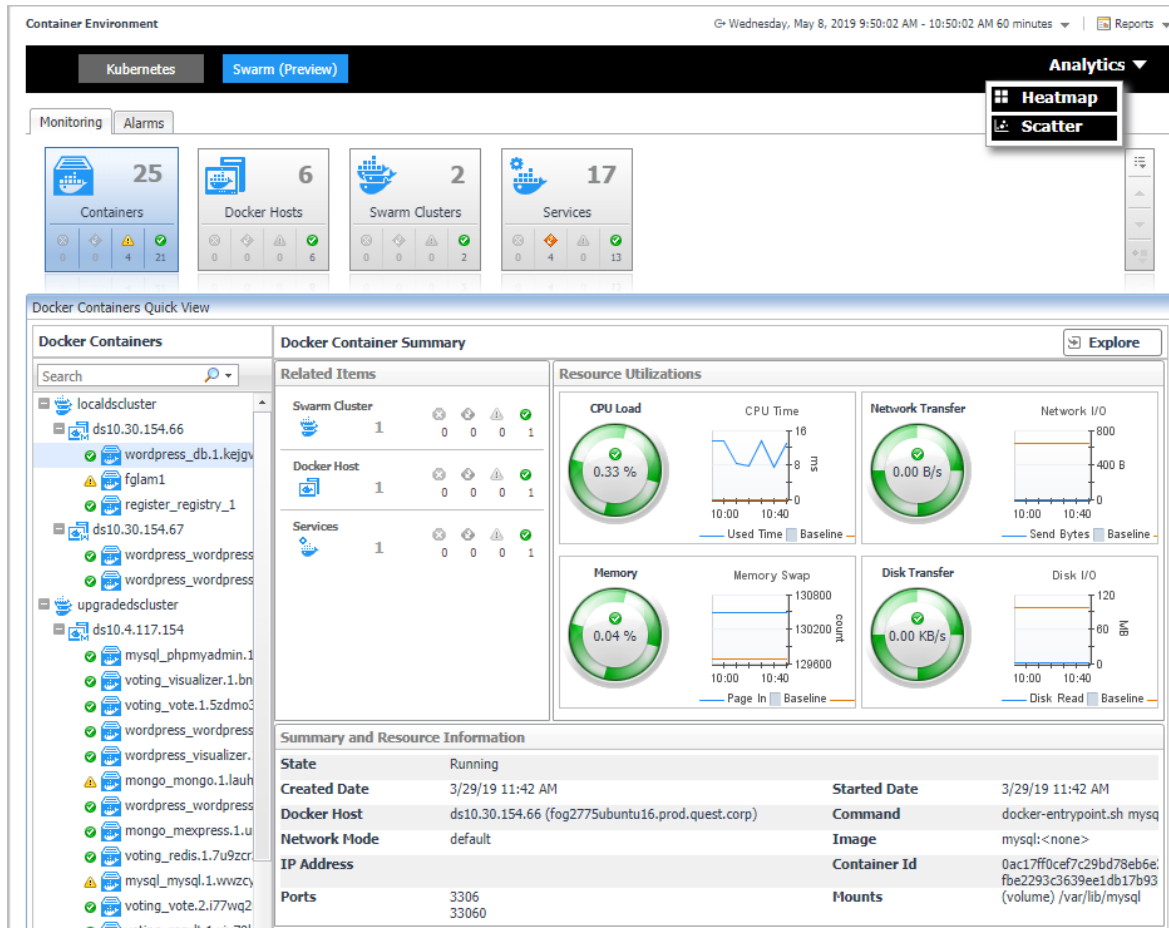
- [Pods Explorer view](#) on page 26
- [Pod metrics](#) on page 52
- [Nodes Explorer view](#) on page 28
- [Node metrics](#) on page 53
- [Cluster Explorer view](#) on page 31
- [Cluster metrics](#) on page 53



# Docker Swarm analytics

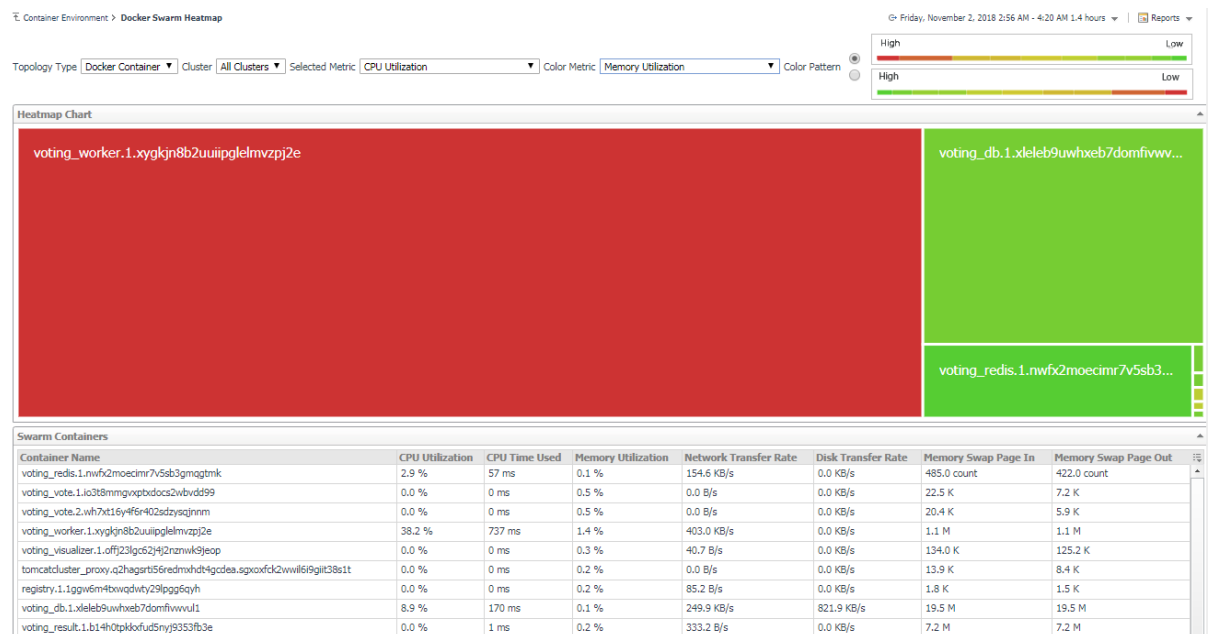
In the Container dashboard, choose **Docker Swarm** from the header. Then click **Analytics** from the header, a drop down view will display with **Heatmap** and **Scatter** on it. Click **Heatmap** will navigate to the Docker Swarm **Heatmap Analytics** dashboard, while click **Scatter** will navigate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

Figure 31. Docker Swarm Analytics Navigation



# Heatmap analytics

Figure 32. Docker Swarm Heatmap Analytics Dashboard



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

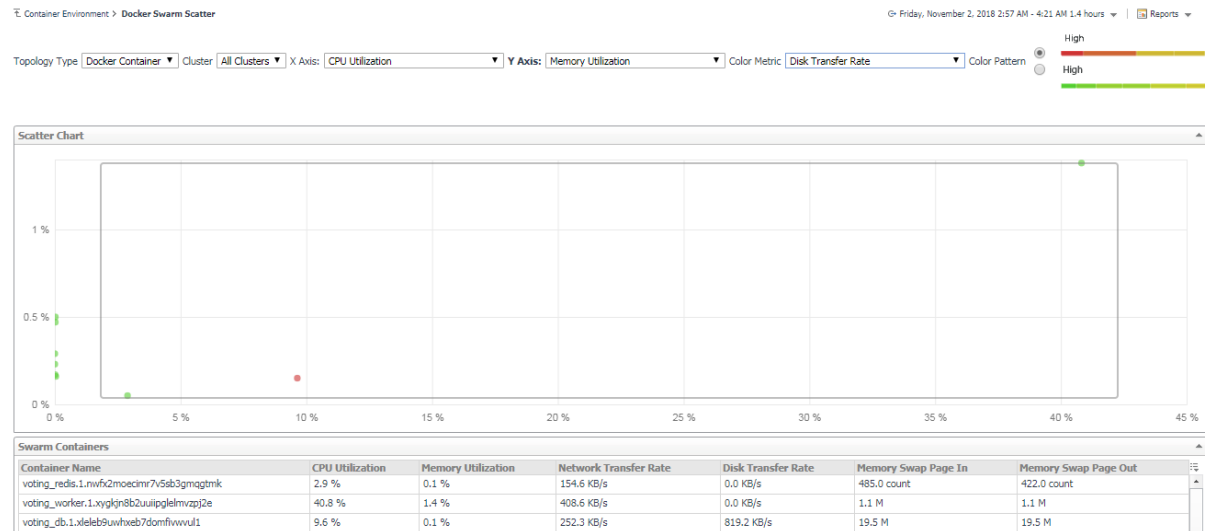
- **Topology Type:** Indicates the monitored topology object, including Docker Container and Docker Host.
- **Cluster:** Lists all clusters available in the monitored Docker Swarm environment.
- **Selected Metric:** Populates a rectangle based upon the selected metrics. For example, if you select *Memory 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 53.
- **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 32 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 39
- [Docker Host Explorer view](#) on page 41
- [Container metrics](#) on page 53

# Scatter Plot analytics

Figure 33. Docker Swarm Scatter Plot Analytics Dashboard



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 33 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 39
- [Docker Host Explorer view](#) on page 41
- [Container metrics](#) on page 53

# 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.
CPU Utilization	Percentage of CPU usage / CPU limit if user configured CPU limit for this pod.
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	Percentage of Memory usage / Memory limit if user configured Memory limit for this pod.
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**

<b>Metric name</b>	<b>Description</b>
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 Outbound 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.

## 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.