Managing Kubernetes Clusters

The Cisco Container Platform web interface allows you to manage Kubernetes clusters by using the **Kubernetes Dashboard**. Once you set up the **Kubernetes Dashboard**, you can deploy applications on the authorized Kubernetes clusters, and manage the application and the cluster itself.

This chapter contains the following topic:

- Setting up Kubernetes Dashboard, on page 1
- Monitoring Health of Cluster Deployments, on page 1
- Monitoring Logs from Cluster Deployments, on page 2

Setting up Kubernetes Dashboard

Step 1 From the left pane, click **Clusters**.

Step 2 From the drop-down list displayed under the **ACTIONS** column, choose **Download Token** to get the Kubernetes configuration (Kubeconfig) file of the cluster that you want to access using the Kubernetes Dashboard.

Step 3 Use the `Kubeconfig` file from Step 2 to login to the Kubernetes Dashboard.

Monitoring Health of Cluster Deployments

It is recommended to continuously monitor the health of your cluster deployment to improve the probability of early detection of failures and avoid any significant impact from a cluster failure.

Cisco Container Platform is deployed with Prometheus and Grafana configured to start monitoring and logging services automatically when a Kubernetes cluster is created.

Prometheus is an open-source systems monitoring and alerting toolkit and Grafana is an open source metric analytics and visualization suite.

Prometheus collects the data from the cluster deployment, and Grafana provides a general purpose dashboard for displaying the collected data. Grafana offers a highly customizable and user-friendly dashboard for monitoring purposes.

Note A user with *Administrator* role can view all the cluster deployments, but a user with *User* role can view only those clusters for which the user has permission to view.

Step 1 Access the Kubernetes cluster master node using ssh.

```
ssh -l <username> <IP address of master node>
```

Note Once you create a Kubernetes cluster, it may take a few minutes for the necessary services to start. If ssh to a cluster fails, we recommend that you try again after a few minutes.

Step 2 Obtain the password for Grafana, which is stored as a Kubernetes secret.

```
kubectl -n ccp get secrets ccp-addons-grafana -o yaml | grep admin-password | awk '{print $2}' |
base64 --decode
```

Note When you run the command on a Control Plane cluster, you can skip **-n ccp** in the command as these services run in the default namespace.

Step 3 Access the Grafana UI using a web browser.

```
https://<VIP>/grafana
```

Where <*VIP*> is the Virtual IP address of the control or tenant cluster as the case may be. In case of a tenant cluster, <*VIP*> is the Virtual IP address that is used by the cluster Ingress as described in Services and Networking.

Step 4 Log in to the Grafana UI of your Kubernetes cluster using your username, and the password that you obtained in Step 2.

Note It is important to either change or retain the original login credentials since the secret that was used to initialize the Grafana login may be lost or changed with future upgrades.

Step 5 Add Prometheus as the data source and configure the Grafana dashboard to monitor the health of your cluster deployments.

Monitoring Logs from Cluster Deployments

The Elasticsearch, Fluentd, and Kibana (EFK) stack enables you to collect and monitor log data from containerized applications for troubleshooting or compliance purposes. These components are automatically installed when you install Cisco Container Platform.

Fluentd is an open source data collector. It works at the backend to collect and forward the log data to Elasticsearch.

Kibana is an open source analytics and visualization platform designed to work with Elasticsearch. It allows you to create rich visualizations and dashboards with the aggregated data.

Note A user with the *Administrator* role can view all logs, but a user with *User* role can view logs for only those clusters for which the user has permission to view.

Viewing EFK Logs Using Kibana (Tenant Cluster)

Before you begin

Ensure that you have installed the `kubectl` utility.

Step 1 Download the Kubeconfig file of the cluster whose logs you want to view, see Downloading Kubeconfig File.

Step 2 Copy the contents of the downloaded Kubeconfig file to:

- Your local host `~/.kube/config`
- A local file and export KUBECONFIG=<*Downloaded Kubeconfig file*>

Step 3 Create a port-forward using kubectl to access Kibana from outside a cluster.

a) Determine the pod.

```
kubectl -n ccp get pods
```

For example, `kibana-logging-7db596d7f6-g9pxv`

b) Open a port-forward.

```
kubectl port-forward -n ccp
```

For example, `kibana-logging-7db596d7f6-g9pxv 5601:5601`

Step 4 Access the Kibana UI and view the data from the target tenant cluster using a web browser.

http://localhost:5601/app/kibana

For more information on customizing the Kibana UI, refer to the latest Kibana documentation.

Viewing EFK Logs Using Kibana (Control Plane Cluster)

Before you begin

Ensure that you have installed the `kubectl` utility.

Step 1 Access the Kubernetes cluster master node using ssh.

```
ssh ccpuser@control plane master node
sudo cat /etc/kubernetes/admin.conf
```

Step 2 Copy the contents of the downloaded Kubeconfig file to:

- Your local host `~/.kube/config`
- A local file and export KUBECONFIG=<*Full path of the Kubeconfig local file*>

For more information on setting Kubeconfig, see Configure Access to Multiple Clusters.

Step 3 Create a port-forward using kubectl to access Kibana from outside a cluster.

a) Determine the pod.

```
kubectl get pods
```

For example, `kibana-logging-7db596d7f6-g9pxv`

b) Open a port-forward.

```
kubectl port-forward
```

For example, `kibana-logging-7db596d7f6-g9pxv 5601:5601`

Step 4 Access the Kibana UI and view the data from the target tenant cluster using a web browser.

http://localhost:5601/app/kibana

For more information on customizing the Kibana UI, refer to the latest Kibana documentation.

Step 5

What to do next

Forwarding Logs to External Elasticsearch Server

Use the following Curl commands to configure forwarding of logs to an external Elasticsearch server:

Step 1 Open a terminal that has a curl client installed.

Step 2 Configure Cisco Container Platform login credentials.

```
export MGMT_HOST=https://<Cisco Container Platform IP address>:<Port>
export CCP_USER=<Username>
export CCP_PASSPHRASE=<Passphrase>
```

Step 3 Login to Cisco Container Platform and save the session cookie for future requests into the cookies.txt local file.

```
curl -k -j -c cookies.txt -X POST -H "Content-Type:application/x-www-form-urlencoded" -d
"username=$CCP_USER&password=$CCP_PASSWORD" $MGMT_HOST/2/system/login/
```

Step 4 Get the list of cluster names.

```
curl -s -k -b cookies.txt -H "Content-Type: application/json"
$MGMT_HOST/2/clusters/ | jq -r '.[].name'
```

Step 5 Set the CLUSTER_NAME environment variable to the cluster that you are working on.

```
export CLUSTER_NAME="<A cluster name from Step 2>"
```

Step 6 Configure the cluster UUID.

```
export CLUSTER_UUID=$(curl -s -k -b cookies.txt -H "Content-Type: application/json"
$MGMT_HOST/2/clusters/$CLUSTER_NAME | jq -r '.uuid')
```

Step 7 Configure the Elasticsearch server IP address and port number.

```
export EFK_SERVER=<IP address of Elasticsearch server>
export EFK_PORT=<Port number of Elasticsearch server>
```

Step 8 Install the helm chart to configure the custom Elasticsearch server.

```
curl -s -k -b cookies.txt -X POST --header 'Content-Type: application/json' --header 'Accept:
application/json' -d '{"chart_url": "/opt/ccp/charts/ccp-agent.tgz", "name": "ccpagent", "options":
```

```
"cop-efk.localLogForwarding.enabled=False,cop-efk.localLogForwarding.elasticsearchHost='$EFK_SERVER',cop-efk.localLogForwarding.elasticsearchPort='EFK_PORT'"}'
 $MGMT_HOST/2/clusters/$CLUSTER_UUID/helmcharts
```