Introduction

This document contains instructions on onboarding a CSS(Cloudera Semantic Search) cluster on the Private cloud (K8s On ECS).

Prerequisites

Helm (v 3.14.x)	Installing Helm
Kubectl	Install Tools | Kubernetes
curl	Linux curl commands
Experience Cluster / Data Service Cluster	How to create an Experience Cluster in ycloud using jenkins?
Private Cloud Base Cluster	How to create a Base Cluster in ycloud using jenkins?
Steps to create manual instances	CSS on PvC DS - Prod deployment

Assumptions

1. You have access to Jenkins https://master-01.jenkins.cloudera.com
2. You have access to Cloudcat https://cloudcat.infra.cloudera.com/provisionedInstanceGroup/ec2Create
3. You have access to the docker-private registry https://docker-private.infra.cloudera.com/
4. You have ~/.ssh and ~/.kube directory in your computer.

Deploying CSS Cluster on ECS

To deploy the CSS cluster we need to deploy the cert manager and a self-sign certificate first. After this, we can deploy the helm charts specific to CSS. In this CSS deployment, we will have 3 master pods, 1 data pod, 1 dashboard pod, 1 ml pod, 1 ingest pod. And the coordinator node will get deployed along with the data node.

Step 1: Download the Kube config

To Download the kube config we need to sftp `/etc/rancher/rke2/rke2.yaml` file from the node with the ECS master role. You also need to move this rke2.yaml file to ops-cluster-configs file. We will use this file in future for kubeconfig settings.

```
Unset
sftp root@<ecs master host>
passwd: <root password>
sftp> get /etc/rancher/rke2/rke2.yaml
// Need to come out of sftp (ctrl+D)
mv rke2.yaml $HOME/.kube/ops-cluster-configs
```

Step 2: Pull and unpack the charts

A. Pull the charts

You need to use 0.1.0-b28 version or more while pulling the charts, here I am using the charts from *docker-private.infra.cloudera.com* repo, this may change in prod. Note: Always advice to create a directory and pull the charts in it.

```
Unset
mkdir css-helm-charts
cd css-helm-charts
```

Example command to pull a chart, you need to pull 2 charts here `solr/opensearch, solr/opensearch-dashboards`

```
Unset
helm pull oci://docker-private.infra.cloudera.com/cloudera-helm/solr/opensearch --version
0.1.0-b28
helm pull oci://docker-private.infra.cloudera.com/cloudera-helm/solr/opensearch-dashboards
--version 0.1.0-b28
```

B. Unpack helm charts bundle

Below is the sample command. You need to unpack all the charts, example as shown below

```
Unset
for file in *.tgz; do tar -vxf "$file"; done
```

After this, your css-helm-charts will have 2 more directories

```
Unset
opensearch                      opensearch-dashboards
```

Note: You may need to execute these steps in prod based on the registry you are using

Step 3: Export the environment variables

ECS_SERVER_HOST, KUBECONFIG, HELM_CHARTS_DIRECTORY. You need to set the proper values to these variables. Below is the example to set one of these variables.

```
Unset
export ECS_SERVER_HOST="example.vpc.cloudera.com"
export HELM_CHARTS_DIRECTORY="/tmp/css-helm-charts"
```

Note:

1. *You can find the value for the ECS_SERVER_HOST from here.*
2. *KUBECONFIG is the file location, which has been downloaded as part of this step, mostly it should be $HOME/.kube/ops-cluster-configs*
3. *HELM_CHARTS_DIRECTORY value should be the complete path of the helm charts pull directory. As per the steps followed it should be the full path till this directory css-helm-charts*

Step 4: Run the install script

You can download the script files deployCSSHelmCharts.sh from this location [PrivateCloud] and give execute permissions to the script and run it. In this case I gave full permission [777]. Example below

```
Unset
chmod 777 deployCSSHelmCharts.sh
./deployCSSHelmCharts.sh
```

This script will install all the necessary helm charts.

Step 5: Open dashboard access by Port forwarding

To be able to connect to the dashboards service hosted on the pod, you will need to do

```
Unset
kubectl port-forward service/opensearch-dashboards 5601 -n css
```

After running this command, you should be able to access the web interface on localhost:5601

Validate CSS Cluster on ECS

Please check this section on how to validate the CSS cluster deployments.

Validation Steps For Dashboard

Please check this section on how to validate the CSS Dashboard.

Delete CSS Cluster on ECS

Step 1: Export the environment variables

You need to export the KUBECONFIG variable. KUBECONFIG is the file location, which has been downloaded as part of this step, mostly it should be `$HOME/.kube/ops-cluster-configs`

Step 2: Run the delete script

You can download the script files deleteInstallation.sh from this location [PrivateCloud] and give execute permissions to the script and run it. In this case I gave full permission [777]. Example below

```
Unset
chmod 777 deleteInstallation.sh
./deleteInstallation.sh
```

This script will uninstall all the necessary helm charts.

Additional Steps to use prod registry

A. Modify the helm charts to use Cloudera's public repository.

Change the values.yaml for opensearch:

- For image field, Replace docker-private.infra.cloudera.com with container.repository.cloudera.com
- For imagePullSecrets field, Replace [] with [{"name": "jfrog-dev"}]

```
Unset
vi opensearch/values.yaml

# replace docker-private.infra.cloudera.com with container.repository.cloudera.com
image:
  repository: "container.repository.cloudera.com/cloudera/opensearch"
  tag: "" ## Use Release version from Chart
  pullPolicy: "IfNotPresent"

# also replace [] with [{"name": "jfrog-dev"}]
imagePullSecrets: [{"name": "jfrog-dev"}]
```

Change the values.yaml for opensearch-dashboards:

- For image field, Replace docker-private.infra.cloudera.com with container.repository.cloudera.com
- For imagePullSecrets field, Replace [] with [{"name": "jfrog-dev"}]

```
Unset
vi opensearch-dashboards/values.yaml

# replace docker-private.infra.cloudera.com with container.repository.cloudera.com
image:
  repository: "container.repository.cloudera.com/cloudera/opensearch-dashboards"
  tag: "" ## Use Release version from Chart
  pullPolicy: "IfNotPresent"

# also replace [] with [{"name": "jfrog-dev"}]
imagePullSecrets: [{"name": "jfrog-dev"}]
```

Additional Links

1. Runbook to create ingest pipeline for neural search
2. For RAG Demo script you can follow this link RAG_NS_Demo