Installing MapR and Kubernetes Software on Separate Nodes

This section describes how to install the configuration files for the MapR Data Fabric for Kubernetes. In this configuration, MapR and Kubernetes software must be installed on separate nodes.

To install the MapR Data Fabric for Kubernetes, you must download the configuration files and use the Kubernetes `kubectl` interface to install the namespace, RBAC, plug-in, and provisioner `.yaml` files.

Before Installation

Before installing the MapR Data Fabric for Kubernetes, note these preinstallation best practices:

- You must install the configuration files in the order shown in the steps below. Using a different installation order can cause problems.
- Ensure that all Kubernetes nodes use the same Linux distribution. For example, all nodes can be CentOS nodes, or all nodes can be Ubuntu nodes. But a cluster with a mixture of CentOS and Ubuntu nodes is not supported.
- This procedure does not allow you to install the MapR Data Fabric for Kubernetes on a Kubernetes node that is also a node in a MapR cluster. If a Kubernetes node already has MapR software installed, installing the MapR Data Fabric for Kubernetes can cause issues with the running MapR cluster. See Installing MapR and Kubernetes Software on the Same Nodes (kdf_installation_same_nodes.html#concept_ogr_sxc_qcb).
- Do not install the MapR client on a node where the volume plug-in configuration file is installed. The MapR client can be installed on a node in the Kubernetes cluster, but it must be installed **before** the MapR Data Fabric for Kubernetes is installed on the same Kubernetes cluster.

Installation Steps

Use these steps to install the configuration files:

1. Download the following configuration (`.yaml`) files from `https://package.mapr.com/tools/KubernetesDataFabric/v<version>/` to a directory on a node in the Kubernetes cluster:

File	Description
`kdf-namespace.yaml`	Configuration file for the `mapr-system` namespace, under which all MapR components are installed.
`kdf-rbac.yaml`	RBAC configuration file. This file enables the provisioner to call the Kubernetes APIs that it needs to function. | - `kdf-plugin-centos.yaml` - `kdf-plugin-ubuntu.yaml` - `kdf-plugin-azure.yaml` [1] - `kdf-plugin-openshift.yaml` [2] - `kdf-plugin-gke.yaml` [3]	Configuration files used to install the plug-in. Download the plug-in file that matches your environment. You can use the CentOS configuration file for Red Hat, CentOS, or SUSE Kubernetes hosts.
`kdf-provisioner.yaml`	Configuration file used to install the provisioner inside the Kubernetes cluster.

[1]Before installing the `kdf-plugin-azure.yaml` , see Azure AKS Considerations (kdf_azure_aks_considerations.html#concept_mdd_2gn_5cb).

[2]To install the `kdf-plugin-openshift.yaml` , see OpenShift Considerations (kdf_installation_openshift.html#concept_hxy_hnt_5cb).

[3]To install the `kdf-plugin-gke.yaml` , see Google Kubernetes Engine (GKE) Considerations (kdf_gke_considerations.html#concept_ntw_php_vcb).

2. In Kubernetes, use the `kubectl create` command with the `-f` option to create the namespace for the plug-in and provisioner:

   Note: The examples in this procedure assume that you are running each `kubectl create` command from the directory containing the downloaded configuration files.

   ```
   kubectl create -f kdf-namespace.yaml
   ```

3. In Kubernetes, use the `kubectl create` command with the `-f` option to install the RBAC file:

   Note: Do not apply the RBAC file in OpenShift environments. See OpenShift Considerations (kdf_installation_openshift.html#concept_hxy_hnt_5cb).

   ```
   kubectl create -f kdf-rbac.yaml
   ```

4. In the plug-in configuration file that you downloaded in step 1, set the Kubernetes service location and the FlexVolume plug-in path. To specify the Kubernetes service location, specify the external location and port of your API server. You can find the correct values by doing a `kubectl config view` and looking at the current context and then looking at the cluster selected for that context. This information is used to look up tickets:

   ```
   - name : KUBERNETES_SERVICE_LOCATION
     value: "changeme!:6443"
   ```

   If your Kubernetes environment has a nonstandard location for FlexVolume plug-ins (for example, Azure environments sometimes use a nonstandard location), specify the FLEXVOLUME_PLUGIN_PATH by changing the directory in the `value:` field:

   ```
   - name : FLEXVOLUME_PLUGIN_PATH
     value: "/usr/libexec/kubernetes/kubelet-plugins/volume/exec"
   ```

5. Use the `kubectl create` command with the `-f` option to install the plug-in. The plug-in that you specify in the `create` command depends on your operating system environment:

   Note: When you issue the `kubectl create -f` command, a daemon set copies the plug-in to every node in the Kubernetes cluster.

   ```
   kubectl create -f kdf-plugin-centos.yaml
   ```

   or

   ```
   kubectl create -f kdf-plugin-ubuntu.yaml
   ```

   or

```
kubectl create -f kdf-plugin-azure.yaml
```

or

```
kubectl create -f kdf-plugin-openshift.yaml
```

or

```
kubectl create -f kdf-plugin-gke.yaml
```

6. In Kubernetes, use the `kubectl create` command with the `-f` option to install the provisioner on a single node of the Kubernetes cluster. Kubernetes determines the node on which to install the provisioner.

   ```
   kubectl create -f kdf-provisioner.yaml
   ```

7. To begin using the MapR Data Fabric for Kubernetes, see MapR Data Fabric for Kubernetes (kdf_using_and_troubleshooting.html#concept_ogr_sxc_qcb).