Environment Overview

You will be interacting with an OpenShift 4 cluster that is running on {{ ENVIRONMENT }}. During the lab you will also install OpenShift Container Storage, based on Rook/Ceph.

The basics of the OpenShift 4 installation have been completed in advance. The OpenShift cluster is essentially set to all defaults and looks like the following:

- 3 master nodes
- 3 worker nodes
- 1 bastion host

Homeroom is a solution that provides this integrated lab guide, terminal, and web console pane. It is actually running inside the cluster that you will be interacting with.

Conventions

You will see various code and command blocks throughout these exercises. Some of the command blocks can be executed directly. Others will require modification of the command before execution. If you see a command block with a red border (see below), the command will copy to clipboard for slight required modification.

The icon beside the command blocks should tell you if the commands will be executed or copied.

- This command block will be copied to your clipboard for modification.

```
some command to modify
```

> To paste the copied command try the following
>
> - Cmd + V *tested to work in Chrome on macOS*
> - Ctrl + Shift + V *tested to work in Chrome and Firefox on Windows 10*
> - Right click + paste in the terminal window *tested to work on Edge on Windows 10*

- This will execute in the console

```
echo Hello World\!
```

Most command blocks support auto highlighting or executing with a click. If you hover over the command block above and left-click, it should automatically highlight all the text to make for easier copying. Look at the symbol next to the block to see if it will copy or execute.

Cluster Admin Authentication

The login you provided to access this lab guide actually has nothing to do with the terminal or web console you will interact with. We use a feature of Kubernetes called `ServiceAccounts` which are non-human user accounts. The terminal and web console tabs are interacting with the OpenShift API using one of these `ServiceAccounts`, and that account has been given the *cluster-admin* `ClusterRole`. This allows the terminal and web console to perform administrative / privileged actions against the APIs.

Privileges in OpenShift are controlled through a set of roles, policies, and bindings which you will learn more about in one of the exercises in this workshop.

As a quick example, you can execute the following to learn more about what a `Role` is:

```
oc explain Role
```

Inspect how `ClusterRole` differs:

```
oc explain ClusterRole
```

You can execute the following to learn more about `RoleBinding`:

```
oc explain RoleBinding
```

Inspect how `ClusterRoleBinding` differs:

```
oc explain ClusterRoleBinding
```

You can always use `oc explain [RESOURCE]` to get more explanation about what various objects are.

Let's look at PolicyRules defined in the `ClusterRole` *cluster-admin*:

```
oc get clusterrole cluster-admin -o yaml
```

Notice how under rules, an account with the*cluster-admin* role has wildcard `*` access to all `resources` and `verbs` of an apiGroup and all `verbs` in `nonResourceURLs`.

`verbs` are actions that you perform against resources. Things like `delete` and `get` are `verbs` in OpenShift.

To learn more about certain verbs, run `oc [verb] --help`

Let's learn more about the `verb` *whoami*:

```
oc whoami --help
```

We will now run `oc whoami` to see what account you will be using today:

```
oc whoami
```

Let's inspect *dashboard-cluster-admin* `ClusterRoleBinding` that gave our `ServiceAccount` *cluster-admin* `ClusterRole`:

```
oc get clusterrolebinding dashboard-cluster-admin -o yaml
```

Notice that our `ServiceAccount` is a subject in this `ClusterRoleBinding` with a role referenced being the *cluster-admin* `ClusterRole`

As a cluster-admin throughout the exercises, you will be able to do anything with the cluster as you have noted earlier, so follow instructions carefully.

Last updated 2020-12-08 08:49:42 +0100