Lab: Kubernetes and OpenShift Command-line Interfaces and APIs

Find detailed information about your OpenShift cluster and assess its health by querying its Kubernetes resources.

Outcomes

- Use the command line to retrieve information about the cluster resources.
- Identify cluster operators and API resources.
- List the available namespaced resources.
- Identify the resources that belong to the core API group.
- List the resource types that the `oauth.openshift.io` API group provides.
- List the resource usage of containers in a pod.
- Use the JSONPath filter to get the number of allocatable pods and compute resources for a node.
- List the memory and CPU usage of all pods in the cluster.
- Use `jq` filters to retrieve the `conditions` status of a pod.
- View cluster events and alerts.

As the `student` user on the `workstation` machine, use the `lab` command to prepare your system for this exercise.

```
[student@workstation ~]$ lab start cli-review
```

Instructions

The API URL of your OpenShift cluster is https://api.ocp4.example.com:6443, and the `oc` command is already installed on your `workstation` machine.

Log in to the OpenShift cluster as the `developer` user with the `developer` password. Use the `cli-review` project for your work.

1. Log in to the OpenShift cluster and create the `cli-review` project.

   Log in to the OpenShift cluster.

   ```
   [student@workstation ~]$ oc login -u developer -p developer \
     https://api.ocp4.example.com:6443
   ...output omitted...
   ```

Create the `cli-review` project.

```
[student@workstation ~]$ oc new-project cli-review
Now using project "cli-review" on server "https://api.ocp4.example.com:6443".
...output omitted...
```

2. Use the `oc` command to list the following information for the cluster:

- Retrieve the cluster version.
- Identify the supported API versions.
- Identify the fields for the `pod.spec.securityContext` object.

Identify the cluster version.

```
[student@workstation ~]$ oc version
Client Version: 4.18.6
Kustomize Version: v5.4.2
Kubernetes Version: v1.31.6
```

Identify the supported API versions.

```
[student@workstation ~]$ oc api-versions
admissionregistration.k8s.io/v1
admissionregistration.k8s.io/v1beta1
apiextensions.k8s.io/v1
apiregistration.k8s.io/v1
apiserver.openshift.io/v1
apps.openshift.io/v1
apps/v1
...output omitted...
```

Identify the fields for the `pod.spec.securityContext` object.

```
[student@workstation ~]$ oc explain pod.spec.securityContext
KIND:     Pod
VERSION:  v1

FIELD: securityContext <PodSecurityContext>

DESCRIPTION:
...output omitted...
```

3. From the terminal, log in to the OpenShift cluster as the `admin` user with the `redhatocp` password. Then, use the command line to identify the following cluster resources:

   - List the cluster operators.
   - Identify the available namespaced resources.
   - Identify the resources that belong to the core API group.
   - List the resource types that the `oauth.openshift.io` API group provides.
   - List the events in the `openshift-kube-controller-manager` namespace.

   Log in to the OpenShift cluster.

   ```
   [student@workstation ~]$ oc login -u admin -p redhatocp \
     https://api.ocp4.example.com:6443
   ...output omitted...
   ```

   List the cluster operators.

   ```
   [student@workstation ~]$ oc get clusteroperators
   NAME                       VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE
   authentication             4.18.6    True        False         False      12h
   baremetal                  4.18.6    True        False         False      31d
   cloud-controller-manager   4.18.6    True        False         False      31d
   cloud-credential           4.18.6    True        False         False      31d
   cluster-autoscaler         4.18.6    True        False         False      31d
   config-operator            4.18.6    True        False         False      31d
   console                    4.18.6    True        False         False      31d
   ...output omitted...
   ```

   List the available namespaced resources.

   ```
   [student@workstation ~]$ oc api-resources --namespaced
   NAME                     SHORTNAMES   APIVERSION   NAMESPACED   KIND
   bindings                              v1           true         Binding
   configmaps               cm           v1           true         ConfigMap
   endpoints                ep           v1           true         Endpoints
   events                   ev           v1           true         Event
   limitranges              limits       v1           true         LimitRange
   persistentvolumeclaims   pvc          v1           true         PersistentVolumeClaim
   pods                     po           v1           true         Pod
   ...output omitted...
   ```

   Identify the resources that belong to the core API group.

```
[student@workstation ~]$ oc api-resources --api-group ''
NAME                     SHORTNAMES   APIVERSION   NAMESPACED   KIND
bindings                              v1           true         Binding
componentstatuses        cs           v1           false        ComponentStatus
configmaps               cm           v1           true         ConfigMap
endpoints                ep           v1           true         Endpoints
events                   ev           v1           true         Event
limitranges              limits       v1           true         LimitRange
namespaces               ns           v1           false        Namespace
nodes                    no           v1           false        Node
...output omitted...
```

List the resource types that the `oauth.openshift.io` API group provides.

```
[student@workstation ~]$ oc api-resources --api-group oauth.openshift.io
NAME                ...  APIVERSION            NAMESPACED KIND
oauthaccesstokens        oauth.openshift.io/v1 false      OAuthAccessToken
oauthauthorizationtokens oauth.openshift.io/v1 false      OAuthAuthorizati...
...output omitted...
```

Retrieve the events for the `openshift-kube-controller-manager` namespace.

```
[student@workstation ~]$ oc get events -n openshift-kube-controller-manager
LAST SEEN TYPE    REASON                  OBJECT                                  ...
...output omitted...
```

4. Identify the following information about the cluster services and its nodes:

   - Retrieve the `conditions` status of the `etcd-master01` pod in the `openshift-etcd` namespace by using `jq` filters to limit the output.
   - List the compute resource usage of the containers in the `etcd-master01` pod in the `openshift-etcd` namespace.
   - Get the number of allocatable pods for the `master01` node by using a JSONPath filter.
   - List the memory and CPU usage of all pods in the cluster.
   - Retrieve the compute resource consumption of the `master01` node.
   - Retrieve the capacity and allocatable CPU for the `master01` node by using a JSONPath filter.

   Retrieve the `conditions` status of the `etcd-master01` pod in the `openshift-etcd` namesp
   the `.status.conditions` attribute of the pod.

```
[student@workstation ~]$ oc get pods etcd-master01 -n openshift-etcd \
  -o json | jq .status.conditions
[
  {
    "lastProbeTime": null,
    "lastTransitionTime": "2023-03-12T16:40:35Z",
    "status": "True",
    "type": "Initialized"
  },
  {
    "lastProbeTime": null,
    "lastTransitionTime": "2023-03-12T16:40:47Z",
    "status": "True",
    "type": "Ready"
  },
  {
    "lastProbeTime": null,
    "lastTransitionTime": "2023-03-12T16:40:47Z",
    "status": "True",
    "type": "ContainersReady"
  },
  {
    "lastProbeTime": null,
    "lastTransitionTime": "2023-03-12T16:40:23Z",
    "status": "True",
    "type": "PodScheduled"
  }
]
```

List the resource usage of the containers in the `etcd-master01` pod in the `openshift-etc`

```
[student@workstation ~]$ oc adm top pods etcd-master01 \
  -n openshift-etcd --containers
POD             NAME           CPU(cores)   MEMORY(bytes)
etcd-master01   etcd           101m         323Mi
etcd-master01   etcd-metrics   8m           32Mi
etcd-master01   etcd-readyz    3m           46Mi
etcd-master01   etcd-rev       1m           33Mi
etcd-master01   etcdctl        0m           0Mi
```

Use a JSONPath filter to determine the number of allocatable pods for the `master01` nod

```
[student@workstation ~]$ oc get node master01 \
  -o jsonpath='{.status.allocatable.pods}{"\n"}'
250
```

List the memory and CPU usage of all pods in the cluster. Use the `--sum` option to print t
resource usage on your system probably differs.

```
[student@workstation ~]$ oc adm top pods -A --sum
NAMESPACE        NAME                                    CPU(cores)  MEMORY(bytes)
metallb-system   controller-5f6dfd8c4f-ddr8v             0m          56Mi
metallb-system   metallb-operator-controller-manager-... 0m          50Mi
metallb-system   metallb-operator-webhook-server-...     0m          26Mi
metallb-system   speaker-2dds4                           9m          210Mi
...output omitted...
                                                         --------    --------
                                                         386m         11929Mi
```

Retrieve the resource consumption of the `master01` node.

```
[student@workstation ~]$ oc adm top node
NAME       CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
master01   1199m        15%    12555Mi         66%
```

Use a JSONPath filter to determine the capacity and allocatable CPU for the `master01` no

```
[student@workstation ~]$ oc get node master01 -o jsonpath=\
'Allocatable: {.status.allocatable.cpu}{"\n"}'\
'Capacity: {.status.capacity.cpu}{"\n"}'
Allocatable: 7500m
Capacity: 8
```

5. Retrieve debugging information for the cluster. Specify the `/home/student/DO180/labs/cli-review/debugging` directory as the destination directory.

   Then, generate debugging information for the `kube-apiserver` cluster operator. Specify the `/home/student/DO180/labs/cli-review/inspect` directory as the destination directory. Limit the debugging information to the last five minutes.

   Retrieve debugging information for the cluster. Save the output to the `/home/student/D` `review/debugging` directory.

   ```
   [student@workstation ~]$ oc adm must-gather \
     --dest-dir /home/student/DO180/labs/cli-review/debugging
   [must-gather      ] OUT Using must-gather plug-in image: quay.io/openshift-r...
   ...output omitted...
   Reprinting Cluster State:
   When opening a support case, bugzilla, or issue please include the following...
   ClusterID: 6c8c6eed-26ed-4911-9df9-b081404842c8
   ClientVersion: 4.18.6
   ClusterVersion: Stable at "4.18.6"
   ClusterOperators:
           All healthy and stable
   ```

Generate debugging information for the `kube-apiserver` cluster operator. Save the outp
the `/home/student/DO180/labs/cli-review/inspect` directory, and limit the debugging i

```
[student@workstation ~]$ oc adm inspect clusteroperator kube-apiserver \
  --dest-dir /home/student/DO180/labs/cli-review/inspect --since 5m
Gathering data for ns/metallb-system...
...output omitted...
Wrote inspect data to /home/student/DO180/labs/cli-review/inspect.
```

Evaluation

As the `student` user on the `workstation` machine, use the `lab` command to grade your work. Correct any reported failures and rerun the command until successful.

```
[student@workstation ~]$ lab grade cli-review
```

Finish

As the `student` user on the `workstation` machine, use the `lab` command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish cli-review
```