Lesson 01 Demo 02

Executing crictl Commands

Objective: To execute crictl commands for performing container runtime operations

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

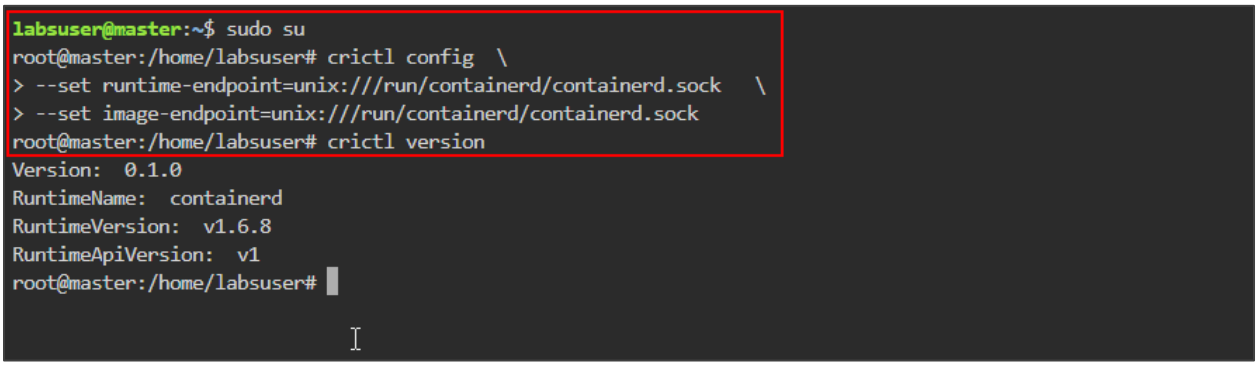
Steps to be followed:

1. Configure the container runtime environment
2. Perform container runtime operations

Step 1: Configure the container runtime environment

1.1 Execute the following commands to get root access, configure the runtime endpoint, and configure the image endpoint for containerd:

```
sudo su
crictl config  \
--set runtime-endpoint=unix:///run/containerd/containerd.sock   \
--set image-endpoint=unix:///run/containerd/containerd.sock
crictl version
```



Note: Here, the command **crictl config** is used to configure the environment for using the container runtime endpoint and image endpoint provided by the containerd.

Step 2: Perform container runtime operations

2.1 List and view the information of the pod in the container runtime environment using the following command:
crictl pods

```
root@master:/home/labsuser# crictl version
Version:  0.1.0
RuntimeName:  containerd
RuntimeVersion:  v1.6.8
RuntimeApiVersion:  v1
root@master:/home/labsuser# crictl pods
POD ID              CREATED             STATE               NAME                                      NAMESPACE           ATTEMPT             RUNTIME
e6835e48b8944       28 minutes ago      Ready               calico-kube-controllers-7ddc4f45bc-chwjv  kube-system         16                  (default)
41545e9c6f33e       28 minutes ago      Ready               coredns-5dd5756b68-s7jkb                  kube-system         16                  (default)
524fc0b0b57d4       28 minutes ago      Ready               coredns-5dd5756b68-6lb9j                  kube-system         16                  (default)
969018e988d77       28 minutes ago      Ready               calico-node-q7vnv                         kube-system         16                  (default)
a6d80f95ed284       28 minutes ago      Ready               kube-proxy-bvgrj                          kube-system         16                  (default)
38cdd83b3539a       29 minutes ago      Ready               kube-apiserver-master.example.com         kube-system         16                  (default)
914a106327462       29 minutes ago      Ready               etcd-master.example.com                   kube-system         16                  (default)
147a21d2f804a       29 minutes ago      Ready               kube-controller-manager-master.example.com kube-system        16                  (default)
10043edd89017       29 minutes ago      Ready               kube-scheduler-master.example.com         kube-system         16                  (default)
a9fab6d223aba       29 minutes ago      Ready               lab-scheduler-master.example.com          kube-system         16                  (default)
```

2.2 Retrieve images to view available container images in the runtime environment using the following command:
crictl images

```
root@master:/home/labsuser# crictl images
IMAGE                                TAG                 IMAGE ID            SIZE
docker.io/calico/cni                 v3.25.0             d70a5947d57e5       88MB
docker.io/calico/cni                 v3.26.1             9dee260ef7f59       93.4MB
docker.io/calico/kube-controllers    v3.25.0             5e785d005ccc1       31.3MB
docker.io/calico/kube-controllers    v3.26.1             1919f2787fa70       32.8MB
docker.io/calico/node                v3.25.0             08616d26b8e74       87.2MB
docker.io/calico/node                v3.26.1             8065b798a4d67       86.6MB
docker.io/library/nginx              1.16-alpine         5fad07aba15a1       8.89MB
docker.io/library/redis              3.2-alpine          6e94a98d3442e       9.43MB
k8s.gcr.io/pause                     3.6                 6270bb605e12e       302kB
registry.k8s.io/coredns/coredns      v1.10.1             ead0a4a53df89       16.2MB
registry.k8s.io/coredns/coredns      v1.11.1             cbb01a7bd410d       18.2MB
registry.k8s.io/etcd                 3.5.12-0            3861cfcd7c04c       57.2MB
registry.k8s.io/etcd                 3.5.9-0             73deb9a3f7025       103MB
registry.k8s.io/kube-apiserver       v1.28.2             cdcab12b2dd16       34.7MB
```

2.3 Obtain a simplified list of **IMAGE ID** for the available container images in the runtime environment using the following command:
crictl images –q

```
root@master:/home/labsuser# crictl images -q
sha256:9dee260ef7f5990aaf6e8f6767b767366c27a6abbf79ba8dba45ff3290bd5de0
sha256:1919f2787fa7098a4c0dec514cf385f90a79c9f6c5effecda570fa3b8c771a3d
sha256:8065b798a4d6729605e3706c202db657bfbcb8109127ece6af5bfb6da106adb7
sha256:4c5693dacb42b96f09609967740l3048a68ea67790f9066bc6aa490e2b40a499
sha256:6270bb605e12e581514ada5fd5b3216f727db55dc87d5889c790e4c760683fee
sha256:ead0a4a53df89fd173874b46093b6e62d8c72967bbf606d672c9e8c9b601a4fc
sha256:73deb9a3f702532592a4167455f8bf2e5f5d900bcc959ba2fd2d35c321de1af9
sha256:cdcab12b2dd16cce4efc5dd43c082469364f19ad978e922d110b74a42eff7cce
sha256:55f13c92defb1eb854040a76e366da866bdcb1cc31fd97b2cde94433c8bf3f57
sha256:c120fed2beb84b861c2382ce81ab046c0ae612e91264ef7c9e61df5900fa0bb0
sha256:7a5d9d67a13f6ae031989bc2969ec55b06437725f397e6eb75b1dccac465a7b8
sha256:4873874c08efc72e9729683a83ffbb7502ee729e9a5ac097723806ea7fa13517
sha256:e6f1816883972d4be47bd48879a08919b96afcd344132622e4d444987919323c
root@master:/home/labsuser#
```

2.4 List and view active containers in the runtime environment using the following command:
crictl ps

```
sha256:7a5d9d67a13f6ae031989bc2969ec55b06437725f397e6eb75b1dccac465a7b8
sha256:4873874c08efc72e9729683a83ffbb7502ee729e9a5ac097723806ea7fa13517
sha256:e6f1816883972d4be47bd48879a08919b96afcd344132622e4d444987919323c
root@master:/home/labsuser# crictl ps
CONTAINER           IMAGE               CREATED             STATE               NAME                      ATTEMPT             POD ID              POD
1060570672113       7a5d9d67a13f6       20 minutes ago      Running             kube-scheduler            94                  10043edd89017       kube-scheduler-master.ex
ample.com
9daa8e9b86423       1919f2787fa70       36 minutes ago      Running             calico-kube-controllers   16                  e6835e48b8944       calico-kube-controllers-
7ddc4f45bc-chwjv
aa80d4c6d4a1c       ead0a4a53df89       36 minutes ago      Running             coredns                   16                  41545e9c6f33e       coredns-5dd5756b68-s7jkb
0f2c9b6f9c7d9       ead0a4a53df89       36 minutes ago      Running             coredns                   16                  524fc0b0b57d4       coredns-5dd5756b68-6lb9j
7250b7f3a6872       8065b798a4d67       36 minutes ago      Running             calico-node               16                  969018e988d77       calico-node-q7vnv
dda1c35231c29       c120fed2beb84       36 minutes ago      Running             kube-proxy                16                  a6d80f95ed284       kube-proxy-bvgrj
dc1cb3694fea3       73deb9a3f7025       37 minutes ago      Running             etcd                      16                  914a106327462       etcd-master.example.com
eaae36e6869a1       cdcab12b2dd16       37 minutes ago      Running             kube-apiserver            16                  38cdd83b3539a       kube-apiserver-master.ex
ample.com
bbe307c60e4e2       55f13c92defb1       37 minutes ago      Running             kube-controller-manager   61                  147a21d2f804a       kube-controller-manager-
master.example.com
root@master:/home/labsuser#
```

Note: Copy any container ID from the list of containers, as shown in the above screenshot

2.5 Retrieve all the active and inactive containers in the runtime environment using the following command:

crictl ps -a

```
root@master:/home/labsuser# crictl ps -a
CONTAINER           IMAGE               CREATED             STATE               NAME                ATTEMPT             POD ID              POD
cfccfaf716504       4c5693dacb42b       About a minute ago  Exited              lab-scheduler       764                 a9fab6d223aba       lab-scheduler-master.ex
ample.com
1060570672113       7a5d9d67a13f6       22 minutes ago      Running             kube-scheduler      94                  10043edd89017       kube-scheduler-master.e
xample.com
3f9cb60d634bb       7a5d9d67a13f6       27 minutes ago      Exited              kube-scheduler      93                  10043edd89017       kube-scheduler-master.e
xample.com
9daa8e9b86423       1919f2787fa70       37 minutes ago      Running             calico-kube-controllers   16                  e6835e48b8944       calico-kube-controllers
-7ddc4f45bc-chwjv
aa80d4c6d4a1c       ead0a4a53df89       37 minutes ago      Running             coredns             16                  41545e9c6f33e       coredns-5dd5756b68-s7jk
b
0f2c9b6f9c7d9       ead0a4a53df89       37 minutes ago      Running             coredns             16                  524fc0b0b57d4       coredns-5dd5756b68-6lb9
j
7250b7f3a6872       8065b798a4d67       37 minutes ago      Running             calico-node         16                  969018e988d77       calico-node-q7vnv
53470b477c02b       8065b798a4d67       37 minutes ago      Exited              mount-bpffs         0                   969018e988d77       calico-node-q7vnv
f05b353140905       9dee260ef7f59       38 minutes ago      Exited              install-cni         0                   969018e988d77       calico-node-q7vnv
dda1c35231c29       c120fed2beb84       38 minutes ago      Running             kube-proxy          16                  a6d80f95ed284       kube-proxy-bvgrj
dab071938ebd2       9dee260ef7f59       38 minutes ago      Exited              upgrade-ipam        1                   969018e988d77       calico-node-q7vnv
```

2.6 Access and view container logs for a specific container using the following command:

crictl logs <container-ID>

```
root@master:/home/labsuser# crictl logs 1060570672113
I1027 03:54:23.127045       1 serving.go:348] Generated self-signed cert in-memory
I1027 03:54:23.904608       1 server.go:154] "Starting Kubernetes Scheduler" version="v1.28.2"
I1027 03:54:23.904813       1 server.go:156] "Golang settings" GOGC="" GOMAXPROCS="" GOTRACEBACK=""
I1027 03:54:23.920363       1 requestheader_controller.go:169] Starting RequestHeaderAuthRequestController
I1027 03:54:23.920732       1 secure_serving.go:210] Serving securely on 127.0.0.1:10259
I1027 03:54:23.920933       1 shared_informer.go:311] Waiting for caches to sync for RequestHeaderAuthRequestController
I1027 03:54:23.921683       1 tlsconfig.go:240] "Starting DynamicServingCertificateController"
I1027 03:54:23.922062       1 configmap_cafile_content.go:202] "Starting controller" name="client-ca::kube-system::extension-apiserver-authentication::client-ca-file"
I1027 03:54:23.922122       1 shared_informer.go:311] Waiting for caches to sync for client-ca::kube-system::extension-apiserver-authentication::client-ca-file
I1027 03:54:23.922311       1 configmap_cafile_content.go:202] "Starting controller" name="client-ca::kube-system::extension-apiserver-authentication::requestheader-clien
t-ca-file"
I1027 03:54:23.922360       1 shared_informer.go:311] Waiting for caches to sync for client-ca::kube-system::extension-apiserver-authentication::requestheader-client-ca-f
ile
I1027 03:54:24.022038       1 leaderelection.go:250] attempting to acquire leader lease kube-system/kube-scheduler...
I1027 03:54:24.022489       1 shared_informer.go:318] Caches are synced for client-ca::kube-system::extension-apiserver-authentication::requestheader-client-ca-file
I1027 03:54:24.022601       1 shared_informer.go:318] Caches are synced for RequestHeaderAuthRequestController
I1027 03:54:24.022773       1 shared_informer.go:318] Caches are synced for client-ca::kube-system::extension-apiserver-authentication::client-ca-file
I1027 03:54:39.770168       1 leaderelection.go:260] successfully acquired lease kube-system/kube-scheduler
```

Note: Replace the <**container-ID**> with the ID of the container that you copied in the step 1.5

2.7 Retrieve the latest log entry for a specific container using the following command:

crictl logs --tail=1 <container-ID>

```
I1027 03:54:24.022773       1 shared_informer.go:318] Caches are synced for client-ca::kube-system::extension-apiserver-authentication::client-ca-file
I1027 03:54:39.770168       1 leaderelection.go:260] successfully acquired lease kube-system/kube-scheduler
root@master:/home/labsuser# crictl logs --tail=1 1060570672113
I1027 03:54:39.770168       1 leaderelection.go:260] successfully acquired lease kube-system/kube-scheduler
root@master:/home/labsuser#
```

Note: Replace the <**container-ID**> with the ID of the container that you copied in the step 1.5

By following these steps, you have successfully executed crictl commands for performing container runtime operations. This can be used to retrieve information on pods, container images, and logs, demonstrating effective management of container operations.