.

Lesson 07 Demo 03

Inspecting Cluster and Node Logs

Objective: To inspect the control-plane components like the API server, controller manager, etcd, and kubelet service in worker nodes for monitoring and troubleshooting

Tools required: kubeadm, kubectl, kubelet, and containerd

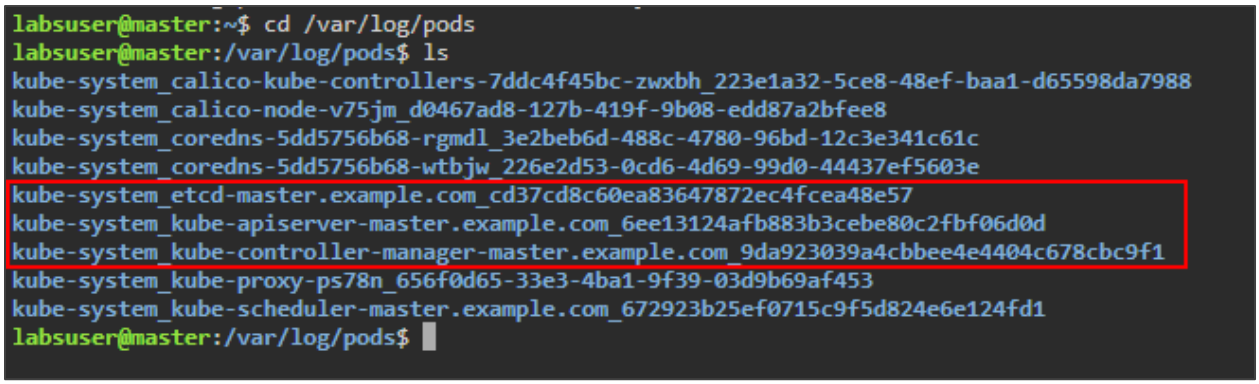
Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

Steps to be followed:

1. View the control-plane component logs
2. View the controller manager logs
3. View the etcd logs
4. View the worker node logs

Step 1: View the control-plane component logs

1.1 In the master node, navigate into the **log/pods** folder and list its components using the following commands:
cd /var/log/pods
ls



.

1.2 Navigate into the API server log sub-directory using the following command:
cd kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver

```
labsuser@master:/var/log/pods$ cd kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver$
```

> **Note:** The alphanumeric number after the API server component name will vary between servers.

1.3 List the latest log file using the following command:
ls -la

```
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver$ ls -la
total 52
drwxr-xr-x 2 root root  4096 Nov  4 09:20 .
drwxr-xr-x 3 root root  4096 Nov  4 09:20 ..
-rw-r----- 1 root root 42148 Nov  4 09:30 0.log
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver$
```

The file **0.log** is the latest log file.

1.4 View the logs using the following command:
sudo cat 0.log

```
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver$ sudo cat 0.log
2023-11-04T09:20:10.625843422Z stderr F I1104 09:20:10.625577       1 options.go:220] external host was not specified, using 172.31.38.186
2023-11-04T09:20:10.642720052Z stderr F I1104 09:20:10.642476       1 server.go:148] Version: v1.28.3
2023-11-04T09:20:10.654790611Z stderr F I1104 09:20:10.654654       1 server.go:150] "Golang settings" GOGC="" GOMAXPROCS="" GOTRACEBACK=""
2023-11-04T09:20:11.489806562Z stderr F I1104 09:20:11.489619       1 shared_informer.go:311] Waiting for caches to sync for node_authorizer
2023-11-04T09:20:11.490508954Z stderr F W1104 09:20:11.490406       1 logging.go:59] [core] [Channel #3 SubChannel #4] grpc: addrConn.createTransport failed to connect to {
2023-11-04T09:20:11.490561079Z stderr F   "Addr": "127.0.0.1:2379",
2023-11-04T09:20:11.490566353Z stderr F   "ServerName": "127.0.0.1",
2023-11-04T09:20:11.490569934Z stderr F   "Attributes": null,
2023-11-04T09:20:11.490573164Z stderr F   "BalancerAttributes": null,
2023-11-04T09:20:11.490576671Z stderr F   "Type": 0,
2023-11-04T09:20:11.490580094Z stderr F   "Metadata": null
2023-11-04T09:20:11.490584222Z stderr F }. Err: connection error: desc = "transport: Error while dialing: dial tcp 127.0.0.1:2379: connect: connection refused"
2023-11-04T09:20:11.490809698Z stderr F W1104 09:20:11.490746       1 logging.go:59] [core] [Channel #1 SubChannel #2] grpc: addrConn.createTransport failed to connect to {
2023-11-04T09:20:11.490817175Z stderr F   "Addr": "127.0.0.1:2379",
2023-11-04T09:20:11.490820845Z stderr F   "ServerName": "127.0.0.1",
2023-11-04T09:20:11.490825065Z stderr F   "Attributes": null,
2023-11-04T09:20:11.490828828Z stderr F   "BalancerAttributes": null,
2023-11-04T09:20:11.490833314Z stderr F   "Type": 0,
2023-11-04T09:20:11.490837342Z stderr F   "Metadata": null
2023-11-04T09:20:11.490841621Z stderr F }. Err: connection error: desc = "transport: Error while dialing: dial tcp 127.0.0.1:2379: connect: connection refused"
```

.

```
2023-11-04T09:30:18.760805762Z stderr F I1104 09:30:18.760718       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:30:18.761131212Z stderr F I1104 09:30:18.761047       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:30:18.761438632Z stderr F I1104 09:30:18.761365       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:30:18.761759142Z stderr F I1104 09:30:18.761680       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.756418615Z stderr F I1104 09:35:18.756279       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.756859345Z stderr F I1104 09:35:18.756767       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.757409014Z stderr F I1104 09:35:18.757330       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.757613179Z stderr F I1104 09:35:18.757558       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.757998355Z stderr F I1104 09:35:18.757929       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.758495489Z stderr F I1104 09:35:18.758422       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.75868526Z stderr F I1104 09:35:18.758630       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.759021367Z stderr F I1104 09:35:18.758954       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.759464786Z stderr F I1104 09:35:18.759391       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.759634972Z stderr F I1104 09:35:18.759583       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.759898293Z stderr F I1104 09:35:18.759842       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.761167072Z stderr F I1104 09:35:18.761091       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.761454818Z stderr F I1104 09:35:18.761393       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.761657532Z stderr F I1104 09:35:18.761603       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.762508497Z stderr F I1104 09:35:18.762436       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
2023-11-04T09:35:18.762882796Z stderr F I1104 09:35:18.762815       1 handler.go:232] Adding GroupVersion crd.projectcalico.org v1 to ResourceManager
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver$
```

Step 2: View the controller manager logs

2.1 Navigate into the controller manager log sub-directory using the following command:
cd kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1/kube-controller-manager

```
labsuser@master:/var/log/pods$ cd kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1/kube-controller-manager
labsuser@master:/var/log/pods/kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1/kube-controller-manager$
```

2.2 List the latest log file using the following command:
ls -la

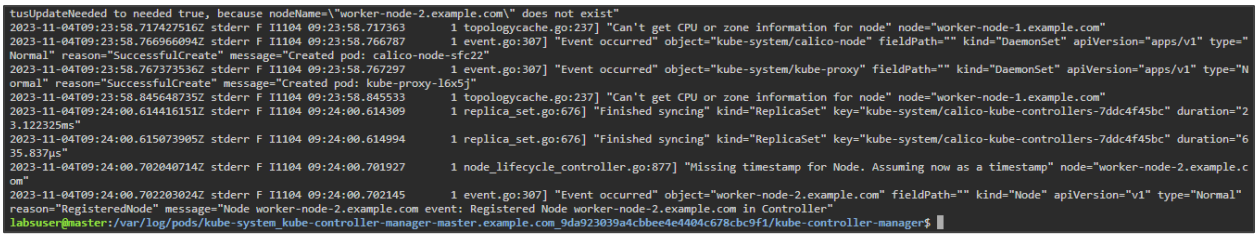
```
labsuser@master:/var/log/pods/kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1/kube-controller-manager$ ls -la
total 48
drwxr-xr-x 2 root root  4096 Nov  4 09:20 .
drwxr-xr-x 3 root root  4096 Nov  4 09:20 ..
-rw-r----- 1 root root 40269 Nov  4 09:24 0.log
labsuser@master:/var/log/pods/kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1/kube-controller-manager$
```

The file **0.log** is the latest log file.

2.3 View the logs using the following command:
sudo cat 0.log

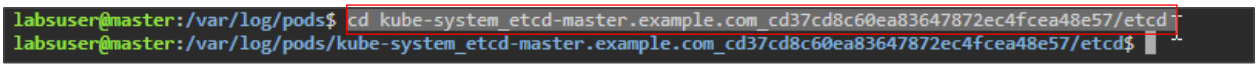
```
labsuser@master:/var/log/pods/kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1/kube-controller-manager$ sudo cat 0.log
2023-11-04T09:20:11.884815379Z stderr F I1104 09:20:11.884587       1 serving.go:348] Generated self-signed cert in-memory
2023-11-04T09:20:12.762600472Z stderr F I1104 09:20:12.762477       1 controllermanager.go:189] "Starting" version="v1.28.3"
2023-11-04T09:20:12.762739555Z stderr F I1104 09:20:12.762690       1 controllermanager.go:191] "Golang settings" GOGC="" GOMAXPROCS="" GOTRACEBACK=""
2023-11-04T09:20:12.766321598Z stderr F I1104 09:20:12.766178       1 secure_serving.go:213] Serving securely on 127.0.0.1:10257
2023-11-04T09:20:12.767893544Z stderr F I1104 09:20:12.767796       1 leaderelection.go:250] attempting to acquire leader lease kube-system/kube-controller-manager...
2023-11-04T09:20:12.768078759Z stderr F I1104 09:20:12.766440       1 tlsconfig.go:240] "Starting DynamicServingCertificateController"
2023-11-04T09:20:12.768851061Z stderr F I1104 09:20:12.766495       1 dynamic_cafile_content.go:157] "Starting controller" name="request-header::/etc/kubernetes/pki/front-proxy-ca.crt"
2023-11-04T09:20:12.769039926Z stderr F I1104 09:20:12.766551       1 dynamic_cafile_content.go:157] "Starting controller" name="client-ca-bundle::/etc/kubernetes/pki/ca.crt"
2023-11-04T09:20:17.769031471Z stderr F E1104 09:20:17.768834       1 leaderelection.go:332] error retrieving resource lock kube-system/kube-controller-manager: Get "https://172.31.38.186
3/apis/coordination.k8s.io/v1/namespaces/kube-system/leases/kube-controller-manager?timeout=5s": net/http: request canceled while waiting for connection (Client.Timeout exceeded while awai
g headers)
2023-11-04T09:20:20.385915778Z stderr F I1104 09:20:20.385783       1 leaderelection.go:260] successfully acquired lease kube-system/kube-controller-manager
2023-11-04T09:20:20.387035814Z stderr F I1104 09:20:20.386923       1 event.go:307] "Event occurred" object="kube-system/kube-controller-manager" fieldPath="" kind="Lease" apiVersion="coo
ation.k8s.io/v1" type="Normal" reason="LeaderElection" message="master.example.com_1c4f2d07-6057-4677-af81-684b00d677a6 became leader"
2023-11-04T09:20:21.400517115Z stderr F I1104 09:20:21.400342       1 shared_informer.go:311] Waiting for caches to sync for tokens
2023-11-04T09:20:21.410427795Z stderr F I1104 09:20:21.410323       1 controllermanager.go:642] "Started controller" controller="persistentvolumeclaim-protection-controller"
2023-11-04T09:20:21.410589654Z stderr F I1104 09:20:21.410512       1 pvc_protection_controller.go:102] "Starting PVC protection controller"
2023-11-04T09:20:21.410701346Z stderr F I1104 09:20:21.410628       1 shared_informer.go:311] Waiting for caches to sync for PVC protection
2023-11-04T09:20:21.418014607Z stderr F I1104 09:20:21.417904       1 controllermanager.go:642] "Started controller" controller="ttl-after-finished-controller"
2023-11-04T09:20:21.418235256Z stderr F I1104 09:20:21.418178       1 ttlafterfinished_controller.go:109] "Starting TTL after finished controller"
2023-11-04T09:20:21.418428919Z stderr F I1104 09:20:21.418264       1 shared_informer.go:311] Waiting for caches to sync for TTL after finished
```

.

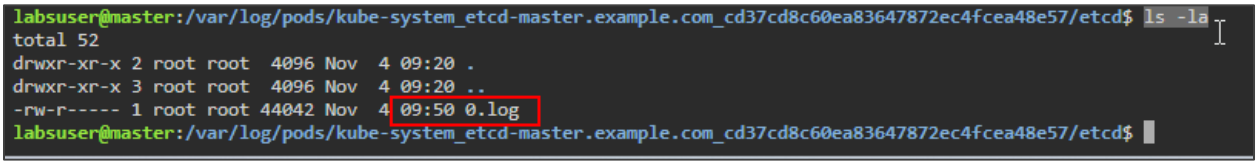


Step 3: View the etcd logs

3.1 Navigate into the etcd log sub-directory using the following command:
cd kube-system_etcd-master.example.com_cd37cd8c60ea83647872ec4fcea48e57/etcd

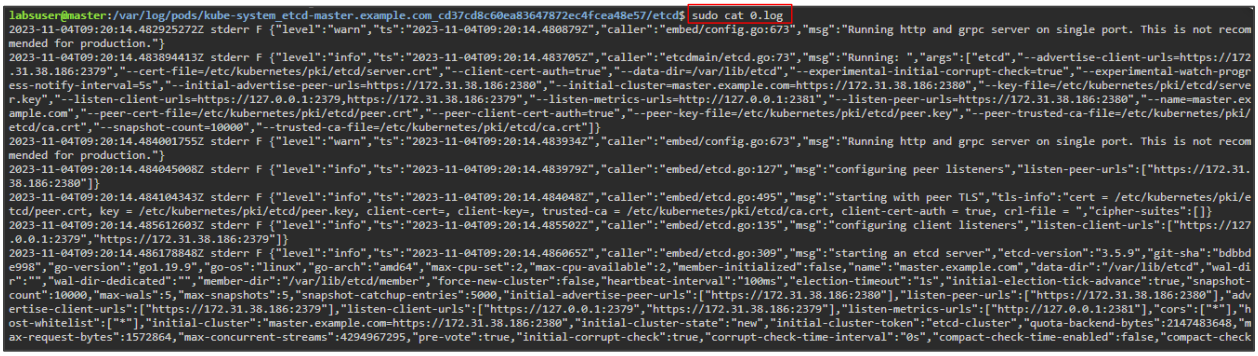


3.2 List the latest log file using the following command:
ls -la



The file **0.log** is the latest log file.

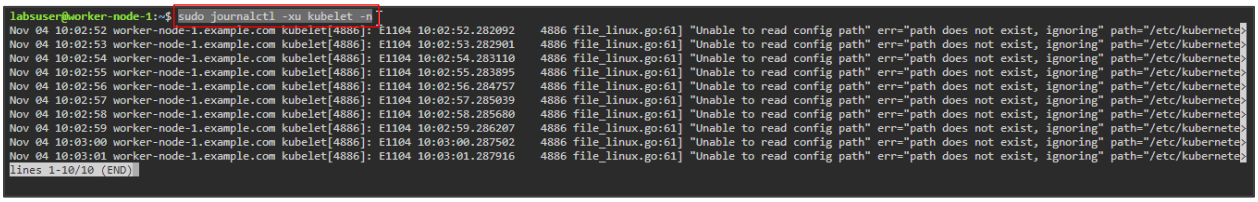
3.3 View the logs using the following command:
sudo cat 0.log



.

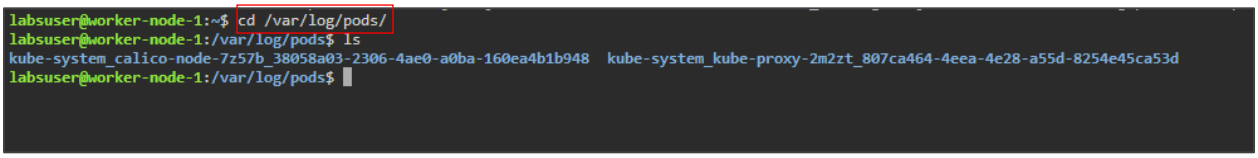
Step 4: View the worker node logs

4.1 In the **worker node-1**, view the kubelet service logs using the following command:
sudo journalctl -xu kubelet -n



Press **q** on the keyboard to exit from the above command.

4.2 View pod logs in the worker node using the following commands:
cd /var/log/pods/
ls



The container name will be listed as folders. The logs of any pod within the container can be viewed by navigating inside the folder.

By following these steps, you have successfully inspected the control-plane components like the API server, controller manager, etcd, and kubelet service in worker nodes for monitoring and troubleshooting.