Lesson 09 Demo 03

Understanding Cluster and Node Logs

Objective: To understand the procedure of inspecting and troubleshooting control-plane components like the API server, controller manager, etcd, and kubelet service in worker nodes

Tools required: kubeadm, kubectl, kubelet, and containerd

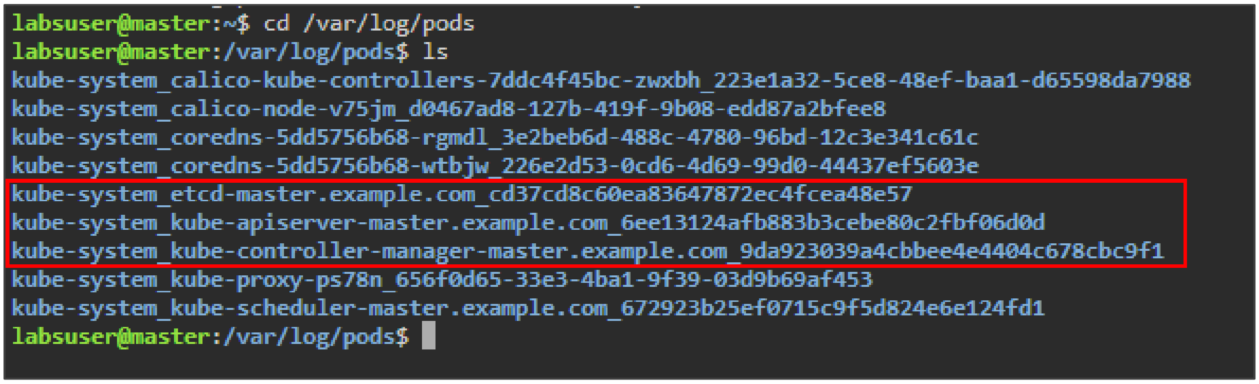
Prerequisites: A Kubernetes cluster should already be set up (refer to the steps provided in Lesson 02, Demo 01 for guidance).

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

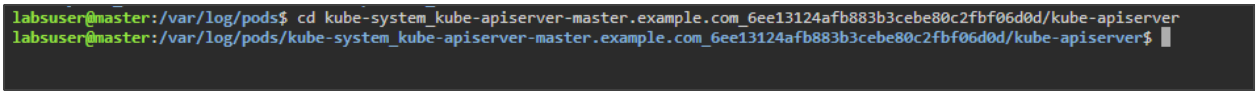
1.1 In the master node, navigate into the **log/pods** folder and list its components using the following commands:
cd /var/log/pods
ls



```
labsuser@master:~$ cd /var/log/pods
labsuser@master:/var/log/pods$ ls
kube-system_calico-kube-controllers-7ddc4f45bc-zwxbh_223e1a32-5ce8-48ef-baa1-d65598da7988
kube-system_calico-node-v75jm_d0467ad8-127b-419f-9b08-edd87a2bfee8
kube-system_coredns-5dd5756b68-rgmdl_3e2beb6d-488c-4780-96bd-12c3e341c61c
kube-system_coredns-5dd5756b68-wtbjw_226e2d53-0cd6-4d69-99d0-44437ef5603e
kube-system_etcd-master.example.com_cd37cd8c60ea83647872ec4fcea48e57
kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d
kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1
kube-system_kube-proxy-ps78n_656f0d65-33e3-4ba1-9f39-03d9b69af453
kube-system_kube-scheduler-master.example.com_672923b25ef0715c9f5d824e6e124fd1
labsuser@master:/var/log/pods$ 
```

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

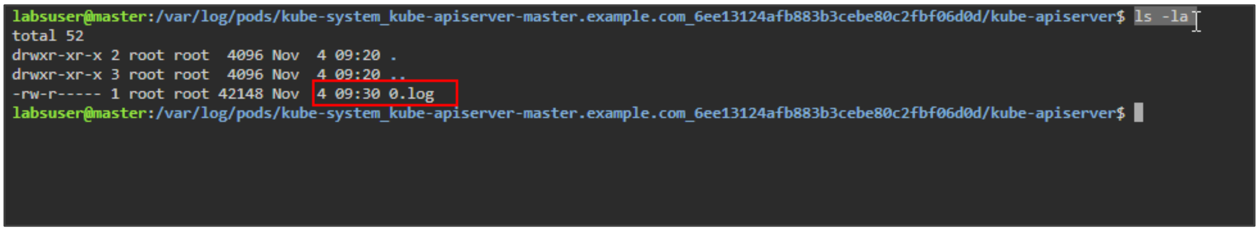


```
labsuser@master:/var/log/pods$ cd kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver$
```

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

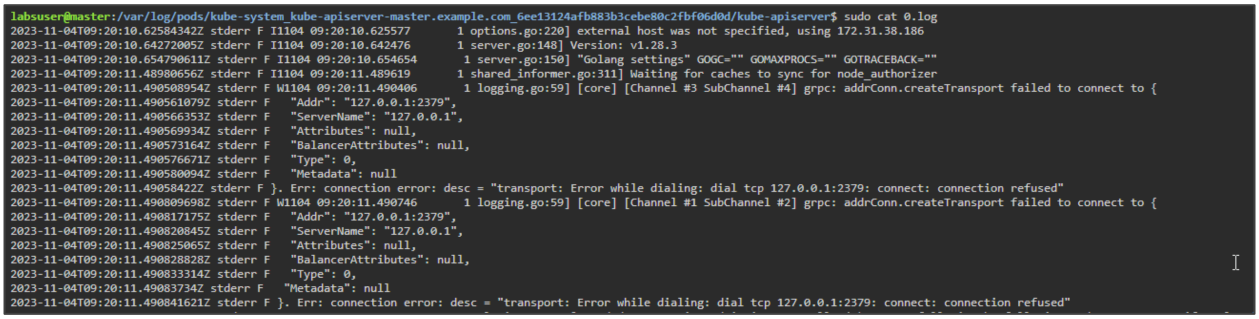


```
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver$ ls -la
total 52
drwxr-xr-x 2 root root  4096 Nov  4 09:20 .
drwxr-xr-x 3 root root  4096 Nov  4 09:20 ..
-rw-r----- 1 root root 42148 Nov  4 09:30 0.log
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee131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

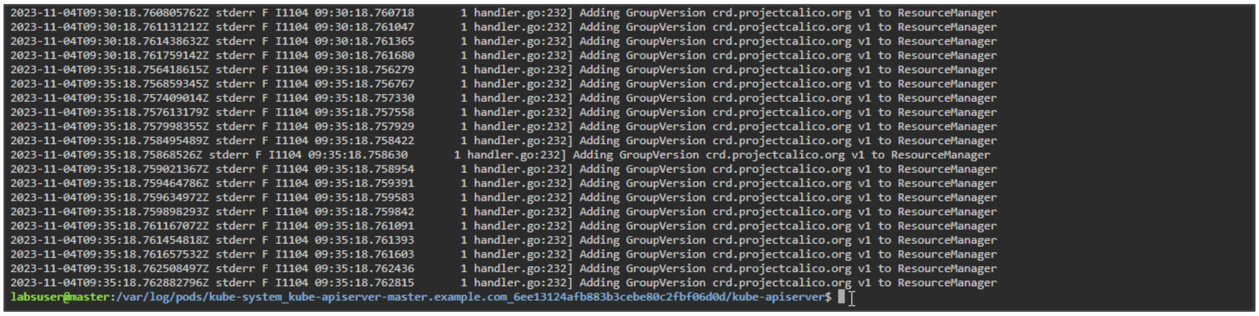
The file **0.log** is the latest log file.

1.4 View the logs using the following command:

sudo cat 0.log



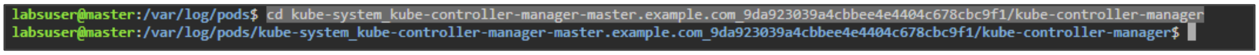
```
labsuser@master:/var/log/pods/kube-system_kube-apiserver-master.example.com_6ee13124afb883b3cebe80c2fbf06d0d/kube-apiserver$ sudo cat 0.log
2023-11-04T09:20:10.62584342Z stderr F I1104 09:20:10.625577       1 options.go:220] external host was not specified, using 172.31.38.186
2023-11-04T09:20:10.64272005Z 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=""
2023-11-04T09:20:11.48980656Z 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
2023-11-04T09:20:11.49058422Z 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,
2023-11-04T09:20:11.49083734Z 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$
```

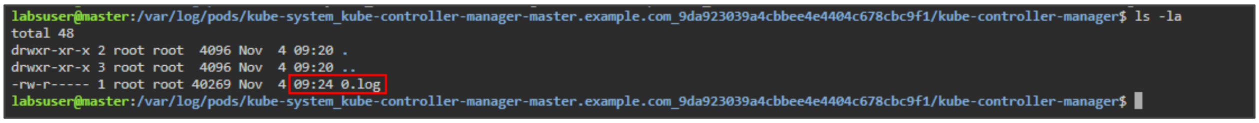
Step 2: View the controller manager logs

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$
```

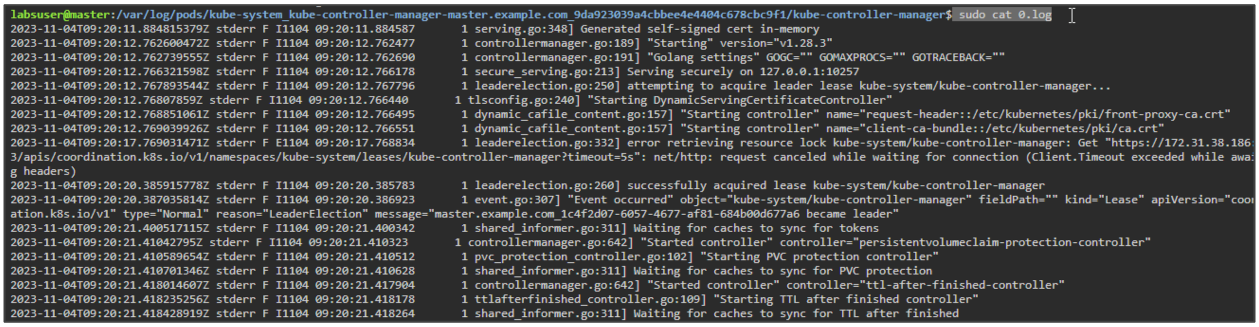
2.2 List the latest log file using the following command:
ls -la



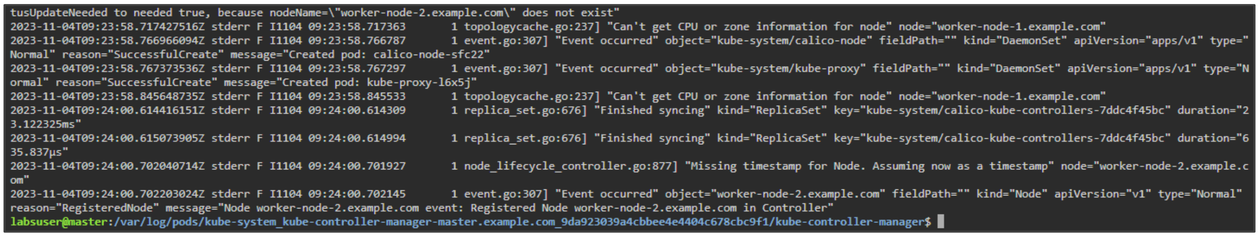
```
labsuser@master:/var/log/pods/kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1/kube-controller-manager$ ls -la
total 48
drwxr-xr-x 2 root root  4096 Nov  4 09:20 .
drwxr-xr-x 3 root root  4096 Nov  4 09:20 ..
-rw-r----- 1 root root 40269 Nov  4 09:24 0.log
labsuser@master:/var/log/pods/kube-system_kube-controller-manager-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...
2023-11-04T09:20:12.76807859Z 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 awa
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
2023-11-04T09:20:21.41042795Z 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
```

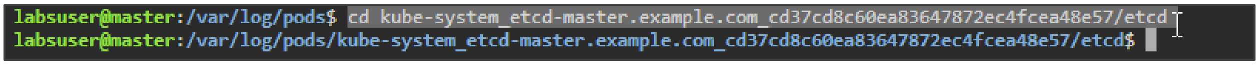


```
tusUpdateNeeded to needed true, because nodeName=\"worker-node-2.example.com\" does not exist"
2023-11-04T09:23:58.717427516Z stderr F I1104 09:23:58.717363       1 topologycache.go:237] "Can't get CPU or zone information for node" node="worker-node-1.example.com"
2023-11-04T09:23:58.766966094Z stderr F I1104 09:23:58.766787       1 event.go:307] "Event occurred" object="kube-system/calico-node" fieldPath="" kind="DaemonSet" apiVersion="apps/v1" type="
Normal" reason="SuccessfulCreate" message="Created pod: calico-node-sfc22"
2023-11-04T09:23:58.767373536Z stderr F I1104 09:23:58.767297       1 event.go:307] "Event occurred" object="kube-system/kube-proxy" fieldPath="" kind="DaemonSet" apiVersion="apps/v1" type="N
ormal" reason="SuccessfulCreate" message="Created pod: kube-proxy-l6x5j"
2023-11-04T09:23:58.845648735Z stderr F I1104 09:23:58.845533       1 topologycache.go:237] "Can't get CPU or zone information for node" node="worker-node-1.example.com"
2023-11-04T09:24:00.614416151Z stderr F I1104 09:24:00.614309       1 replica_set.go:676] "Finished syncing" kind="ReplicaSet" key="kube-system/calico-kube-controllers-7ddc4f45bc" duration="2
3.122325ms"
2023-11-04T09:24:00.615073905Z stderr F I1104 09:24:00.614994       1 replica_set.go:676] "Finished syncing" kind="ReplicaSet" key="kube-system/calico-kube-controllers-7ddc4f45bc" duration="6
35.837µs"
2023-11-04T09:24:00.702040714Z stderr F I1104 09:24:00.701927       1 node_lifecycle_controller.go:877] "Missing timestamp for Node. Assuming now as a timestamp" node="worker-node-2.example.c
om"
2023-11-04T09:24:00.702203024Z stderr F I1104 09:24:00.702145       1 event.go:307] "Event occurred" object="worker-node-2.example.com" fieldPath="" kind="Node" apiVersion="v1" type="Normal"
reason="RegisteredNode" message="Node worker-node-2.example.com event: Registered Node worker-node-2.example.com in Controller"
labsuser@master:/var/log/pods/kube-system_kube-controller-manager-master.example.com_9da923039a4cbbee4e4404c678cbc9f1/kube-controller-manager$
```

Step 3: View the etcd logs

3.1 Navigate into the etcd log sub-directory using the following command:

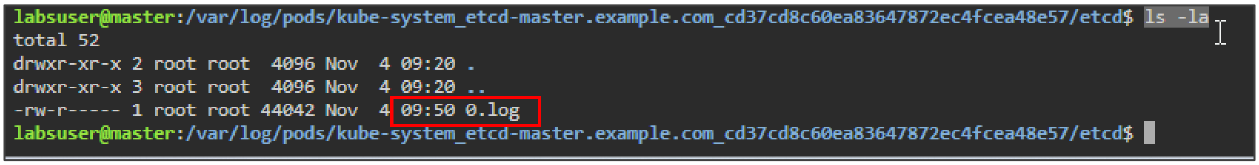
cd kube-system_etcd-master.example.com_cd37cd8c60ea83647872ec4fcea48e57/etcd



```
labsuser@master:/var/log/pods$ cd kube-system_etcd-master.example.com_cd37cd8c60ea83647872ec4fcea48e57/etcd
labsuser@master:/var/log/pods/kube-system_etcd-master.example.com_cd37cd8c60ea83647872ec4fcea48e57/etcd$
```

3.2 List the latest log file using the following command:

ls -la

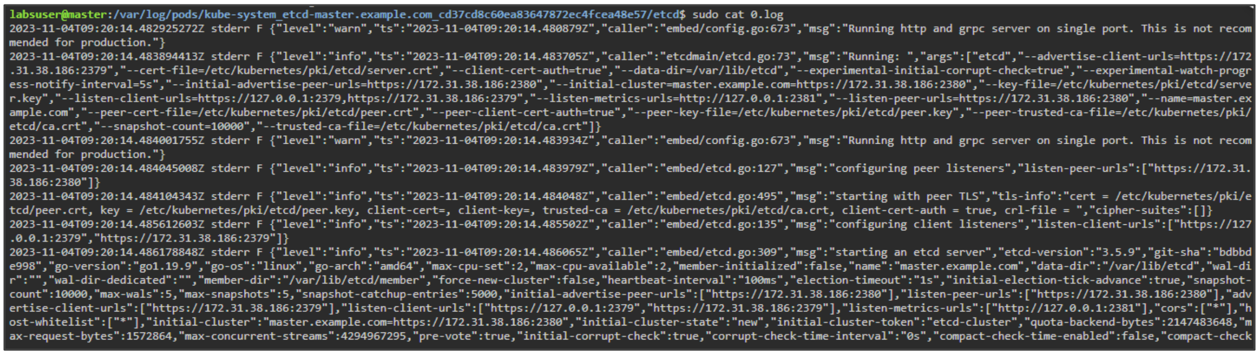


```
labsuser@master:/var/log/pods/kube-system_etcd-master.example.com_cd37cd8c60ea83647872ec4fcea48e57/etcd$ ls -la
total 52
drwxr-xr-x 2 root root  4096 Nov  4 09:20 .
drwxr-xr-x 3 root root  4096 Nov  4 09:20 ..
-rw-r----- 1 root root 44042 Nov  4 09:50 0.log
labsuser@master:/var/log/pods/kube-system_etcd-master.example.com_cd37cd8c60ea83647872ec4fcea48e57/etcd$
```

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

3.3 View the logs using the following command:

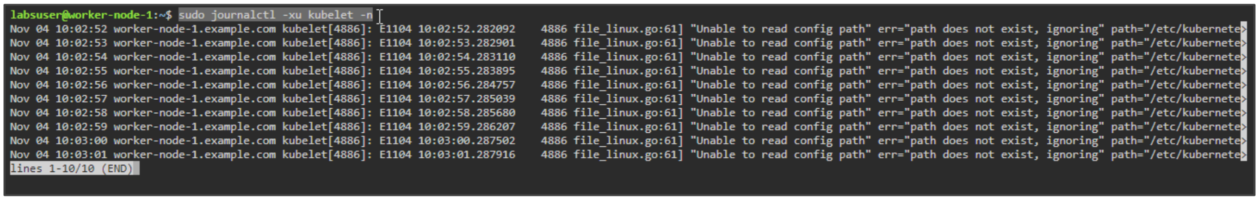
sudo cat 0.log



```
labsuser@master:/var/log/pods/kube-system_etcd-master.example.com_cd37cd8c60ea83647872ec4fcea48e57/etcd$ sudo cat 0.log
2023-11-04T09:20:14.482925272Z stderr F {"level":"warn","ts":"2023-11-04T09:20:14.480879Z","caller":"embed/config.go:673","msg":"Running http and grpc server on single port. This is not recom
mended for production."}
2023-11-04T09:20:14.483894413Z stderr F {"level":"info","ts":"2023-11-04T09:20:14.483705Z","caller":"etcdmain/etcd.go:73","msg":"Running: ","args":["etcd","--advertise-client-urls=https://172
.31.38.186:2379","--cert-file=/etc/kubernetes/pki/etcd/server.crt","--client-cert-auth=true","--data-dir=/var/lib/etcd","--experimental-initial-corrupt-check=true","--experimental-watch-progr
ess-notify-interval=5s","--initial-advertise-peer-urls=https://172.31.38.186:2380","--initial-cluster=master.example.com=https://172.31.38.186:2380","--key-file=/etc/kubernetes/pki/etcd/serve
r.key","--listen-client-urls=https://127.0.0.1:2379,https://172.31.38.186:2379","--listen-metrics-urls=http://127.0.0.1:2381","--listen-peer-urls=https://172.31.38.186:2380","--name=master.ex
ample.com","--peer-cert-file=/etc/kubernetes/pki/etcd/peer.crt","--peer-client-cert-auth=true","--peer-key-file=/etc/kubernetes/pki/etcd/peer.key","--peer-trusted-ca-file=/etc/kubernetes/pki/
etcd/ca.crt","--snapshot-count=10000","--trusted-ca-file=/etc/kubernetes/pki/etcd/ca.crt"]}
2023-11-04T09:20:14.484001755Z stderr F {"level":"warn","ts":"2023-11-04T09:20:14.483934Z","caller":"embed/config.go:673","msg":"Running http and grpc server on single port. This is not recom
mended for production."}
2023-11-04T09:20:14.484045008Z stderr F {"level":"info","ts":"2023-11-04T09:20:14.483979Z","caller":"embed/etcd.go:127","msg":"configuring peer listeners","listen-peer-urls":["https://172.31.
38.186:2380"]}
2023-11-04T09:20:14.484104343Z stderr F {"level":"info","ts":"2023-11-04T09:20:14.484048Z","caller":"embed/etcd.go:495","msg":"starting with peer TLS","tls-info":"cert = /etc/kubernetes/pki/e
tcd/peer.crt, key = /etc/kubernetes/pki/etcd/peer.key, client-cert=, client-key=, trusted-ca = /etc/kubernetes/pki/etcd/ca.crt, client-cert-auth = true, crl-file = ","cipher-suites":[]}
2023-11-04T09:20:14.485612603Z stderr F {"level":"info","ts":"2023-11-04T09:20:14.485502Z","caller":"embed/etcd.go:135","msg":"configuring client listeners","listen-client-urls":["https://127
.0.0.1:2379","https://172.31.38.186:2379"]}
2023-11-04T09:20:14.486178848Z stderr F {"level":"info","ts":"2023-11-04T09:20:14.486065Z","caller":"embed/etcd.go:309","msg":"starting an etcd server","etcd-version":"3.5.9","git-sha":"bdbbd
e998","go-version":"go1.19.9","go-os":"linux","go-arch":"amd64","max-cpu-set":2,"max-cpu-available":2,"member-initialized":false,"name":"master.example.com","data-dir":"/var/lib/etcd","wal-di
r":"","wal-dir-dedicated":"","member-dir":"/var/lib/etcd/member","force-new-cluster":false,"heartbeat-interval":"100ms","election-timeout":"1s","initial-election-tick-advance":true,"snapshot-
count":10000,"max-wals":5,"max-snapshots":5,"snapshot-catchup-entries":5000,"initial-advertise-peer-urls":["https://172.31.38.186:2380"],"listen-peer-urls":["https://172.31.38.186:2380"],"adv
ertise-client-urls":["https://172.31.38.186:2379"],"listen-client-urls":["https://127.0.0.1:2379","https://172.31.38.186:2379"],"listen-metrics-urls":["http://127.0.0.1:2381"],"cors":["*"],"h
ost-whitelist":["*"],"initial-cluster":"master.example.com=https://172.31.38.186:2380","initial-cluster-state":"new","initial-cluster-token":"etcd-cluster","quota-backend-bytes":2147483648,"m
ax-request-bytes":1572864,"max-concurrent-streams":4294967295,"pre-vote":true,"initial-corrupt-check":true,"corrupt-check-time-interval":"0s","compact-check-time-enabled":false,"compact-check
```

Step 4: View the worker node logs

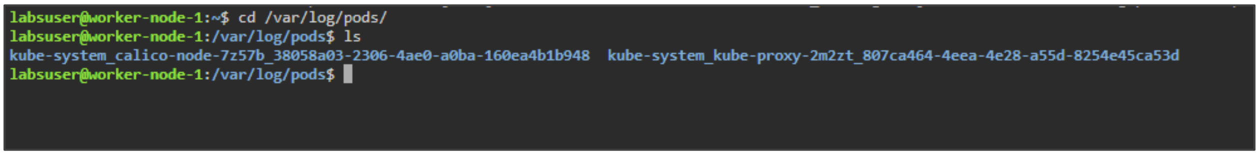
4.1 In the **worker node-1**, view the kubelet service logs using the following command:
sudo journalctl -xu kubelet -n



```
labsuser@worker-node-1:~$ sudo journalctl -xu kubelet -n
Nov 04 10:02:52 worker-node-1.example.com kubelet[4886]: E1104 10:02:52.282092    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:02:53 worker-node-1.example.com kubelet[4886]: E1104 10:02:53.282901    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:02:54 worker-node-1.example.com kubelet[4886]: E1104 10:02:54.283110    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:02:55 worker-node-1.example.com kubelet[4886]: E1104 10:02:55.283895    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:02:56 worker-node-1.example.com kubelet[4886]: E1104 10:02:56.284757    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:02:57 worker-node-1.example.com kubelet[4886]: E1104 10:02:57.285039    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:02:58 worker-node-1.example.com kubelet[4886]: E1104 10:02:58.285680    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:02:59 worker-node-1.example.com kubelet[4886]: E1104 10:02:59.286207    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:03:00 worker-node-1.example.com kubelet[4886]: E1104 10:03:00.287502    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
Nov 04 10:03:01 worker-node-1.example.com kubelet[4886]: E1104 10:03:01.287916    4886 file_linux.go:61] "Unable to read config path" err="path does not exist, ignoring" path="/etc/kubernete>
lines 1-10/10 (END)
```

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



```
labsuser@worker-node-1:~$ cd /var/log/pods/
labsuser@worker-node-1:/var/log/pods$ ls
kube-system_calico-node-7z57b_38058a03-2306-4ae0-a0ba-160ea4b1b948  kube-system_kube-proxy-2m2zt_807ca464-4eea-4e28-a55d-8254e45ca53d
labsuser@worker-node-1:/var/log/pods$
```

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 will successfully be able to inspect and troubleshoot the control-plane components in the worker nodes.