

## (July 2023 – DevOps) Minikube fails to start under Linux- *It seems the kubelet isn't running or is healthy*

Minikube is a lightweight Kubernetes implementation that creates a VM on your local machine and deploys a simple cluster containing only one node. Minikube is available for Linux, macOS, and Windows systems and is predominantly used while learning to work with Kubernetes; among other uses.

I have a minikube installed on a Linux environment and ran into an issue. The troubleshooting process and resolution is documented below.

### Server Information<sup>i</sup>

- AWS Ubuntu medium for Master 2 vCPU + 4gb ram
- Linux ip-172-31-27-156 4.4.0-1128-aws #142-Ubuntu SMP Fri Apr 16 12:42:33 UTC 2021 x86\_64 x86\_64 GNU/Linux
- Docker version 18.09.7, build 2d0083d

### Issue

During kubeadm init

```
[ kubelet -check ] Initial timeout of 40 s passed .
[ kubelet - check ] It seems like the kubelet isn't running or healthy .
[ kubelet - check ] The HTTP call equal to 'curl -sSL http://localhost:10248/healthz' failed
[ kubelet - check ] It seems like the kubelet isn't running or healthy .
[ kubelet - check ] The HTTP call equal to 'curl -sSL http://localhost:10248/healthz' failed
[ kubelet - check ] It seems like the kubelet isn't running or healthy .
[ kubelet - check ] The HTTP call equal to 'curl -sSL http://localhost:10248/healthz' failed
[ kubelet - check ] It seems like the kubelet isn't running or healthy .
[ kubelet - check ] The HTTP call equal to 'curl -sSL http://localhost:10248/healthz' failed
- check ] It seems like the kubelet isn't running or healthy .
[ kubelet - check ] The HTTP call equal to 'curl -sSL http://localhost:10248/healthz' failed
```

Looking at the above problem, it seems that the kubelet failed to start. Execute the following command to see details of the issue:

```
tail /var/log/syslog
```

Shown:

```
Jan 27 15:45:24 ip-172-31-27-156 kubelet[10517]: E0127 15:45:24.337666 10517 server.go:302] "Failed to run kubelet" err="failed to run Kubelet: misconfiguration: kubelet cgroup driver: \"systemd\" is different from docker cgroup driver: \"cgroupfs\""
Jan 27 15:45:24 ip-172-31-27-156 systemd[1]: kubelet.service: Main process exited, code=exited, status=1/FAILURE
Jan 27 15:45:24 ip-172-31-27-156 systemd[1]: kubelet.service: Unit entered failed state.
Jan 27 15:45:24 ip-172-31-27-156 systemd[1]: kubelet.service: Failed with result 'exit-code'.
```

The above log shows that the cgroup driver of kubelet is cgroupfs, and the cgroup driver of docker is systemd. The inconsistency between the two causes kubelet to fail to start.

### Solve the problem

1. I have tried to modify the cgroup driver of kubelet (file location: `/etc/systemd/system/kubelet.service.d/10-kubeadm.conf`), but it will be overwritten every time minikube is started, so I can only give up this method and instead modify the cgroup driver settings of docker.
2. Open the file `/usr/lib/systemd/system/docker.service`, as shown below.
3. Change the **systemd** in the red box to **cgroupfs**:

1. Note for Ubuntu path<sup>ii</sup>

docker.service - Docker Application Container Engine

Loaded: loaded (`/lib/systemd/system/docker.service`; enabled; vendor preset: enabled)

Active: active (running) since Thu 2022-01-27 15:21:12 UTC; 2h 22min ago

```
[Service]
Type=notify
NotifyAccess=main
EnvironmentFile=-/run/containers/registries.conf
EnvironmentFile=-/etc/sysconfig/docker
EnvironmentFile=-/etc/sysconfig/docker-storage
EnvironmentFile=-/etc/sysconfig/docker-network
Environment=GOTRACEBACK=crash
Environment=DOCKER_HTTP_HOST_COMPAT=1
Environment=PATH=/usr/libexec/docker:/usr/bin:/usr/sbin
ExecStart=/usr/bin/dockerd-current \
    --add-runtime docker-runc=/usr/libexec/docker/docker-runc-current \
    --default-runtime=docker-runc \
    --exec-opt native.cgroupdriver=systemd \
    --userland-proxy-path=/usr/libexec/docker/docker-proxy-current \
    --init-path=/usr/libexec/docker/docker-init-current \
    --seccomp-profile=/etc/docker/seccomp.json \
    $OPTIONS \
    $DOCKER_STORAGE_OPTIONS \
    $DOCKER_NETWORK_OPTIONS \
    $ADD_REGISTRY \
    $BLOCK_REGISTRY \
    $INSECURE_REGISTRY \
    $REGISTRIES
ExecReload=/bin/kill -s HUP $MAINPID
LimitNOFILE=1048576
LimitNPROC=1048576
LimitCORE=infinity
```

1. Reload the configuration information and restart the service:

```
systemctl daemon-reload && systemctl restart docker
```

After completing the above modifications, restart minikube and it will be successful, as follows:

```
[ root@minikube ~ ] # minikube delete
* Uninstalling Kubernetes v1 . 14.3 using kubeadm ...
* Deleting "minikube" from none ...
* The "minikube" cluster has been deleted .
[ root@minikube ~ ] # minikube start - -vm - driver = none
* minikube v1.1.1 on linux ( amd64 ) * Creating none _ VM_ ( CPUs =
2 , Memory = 2048 MB , Disk = 20000 MB ) ... * Configuring environment for Kubernetes v1 .14.
- minikube / _ _
storage - provisioner_v1.8.1 : stat / root / .minikube / cache / images / gcr.io / k8s - min
or directory * Pulling images ... * Launching Kubernetes ... * Configuring _ local host env

! The ' none ' driver provides limited isolation and may reduce system security and reliability
! For more information , see :
- https://github.com/kubernetes/minikube/blob/master/docs/vmdriver-none.md_ _ _ _ _ _ _ _ _ _

! kubectl and minikube configuration will be stored in / root
! To use kubectl or minikube commands as your own user , you may
! need to relocate them . For example , to overwrite your own settings :
- sudo mv /root/.kube/root/.minikube$HOME -sudo chown -R $ USER $ HOME / .kube $ HOME / .min
```

I hope this article can provide some references when you encounter similar problems.

## Note

For those who do not want to make changes in **docker.service** (or do not have permission) alternate method is to change the driver in **kublet.service.d** to match the value in Docker. Ubuntu docker distribution by default uses the native control group driver **cgroupfs** while kublet assumes **systemd**. As an alternate you may change the kublet config in **(/etc/systemd/system/kubelet.service.d**.

Suggested change:

```
Environment="KUBELET_KUBECONFIG_ARGS=--bootstrap-
kubecfg=/etc/kubernetes/bootstrap-kubelet.conf --
kubecfg=/etc/kubernetes/kubelet.conf --cgroup-driver=cgroupfs"
```

---

<sup>i</sup> Server is no longer available and was part of a training lab.

<sup>ii</sup> Docs: <https://docs.docker.com>