Lab 9. Installing Docker on local VM

Objectives

- Install Docker on Oracle Linux 7
- Verify Docker installation
- Start and enable Docker Engine

Prerequisites

- 64-bit Oracle Linux 7

Sequence 1. Installing Docker

There are two methods for installing Docker on OL 7.

- One method involves installing it on an existing installation of the operating system.
- The other involves spinning up a server with a tool called Docker Machine that auto-installs Docker on it.

In this Lab, you'll learn how to install and use it on an existing installation of Oracle Linux 7.

1. Add these to Existing Repo file /etc/yum.repos.d/oracle-linux-ol7.repo

```
# vi /etc/yum.repos.d/oracle-linux-ol7.repo
[ol7_developer]
name=Oracle Linux $releasever Development Packages ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/developer/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol7_developer_EPEL]
name=Oracle Linux $releasever Development Packages ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/developer_EPEL/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

2. The Docker installation packages are available in the official developer repository as you configured in ***Lab 6***. Login as root user and follow the steps

```
# yum install docker-engine -y
```

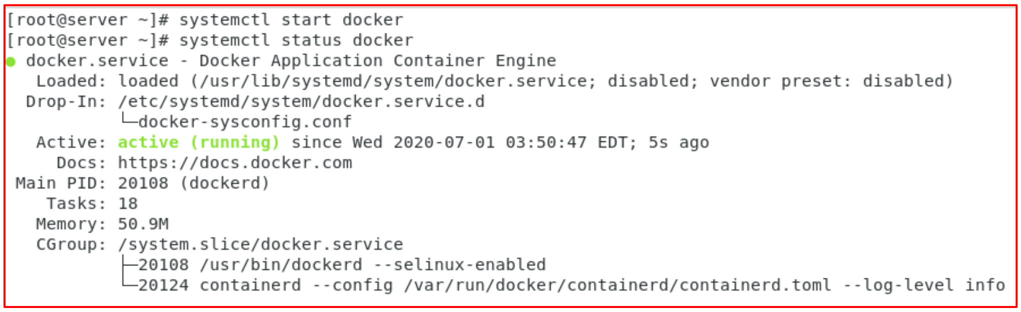
3. After installation is complete, start the Docker daemon and, make sure it starts at every server reboot.

```
# systemctl enable --now docker
```

4. Verify that it’s running:

```
# systemctl status docker
```

5. The output should be similar to the following, showing that the service is active and running:



```
[root@server ~]# systemctl start docker
[root@server ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/docker.service.d
           └─docker-sysconfig.conf
   Active: active (running) since Wed 2020-07-01 03:50:47 EDT; 5s ago
     Docs: https://docs.docker.com
 Main PID: 20108 (dockerd)
    Tasks: 18
   Memory: 50.9M
   CGroup: /system.slice/docker.service
           ├─20108 /usr/bin/dockerd --selinux-enabled
           └─20124 containerd --config /var/run/docker/containerd/containerd.toml --log-level info
```

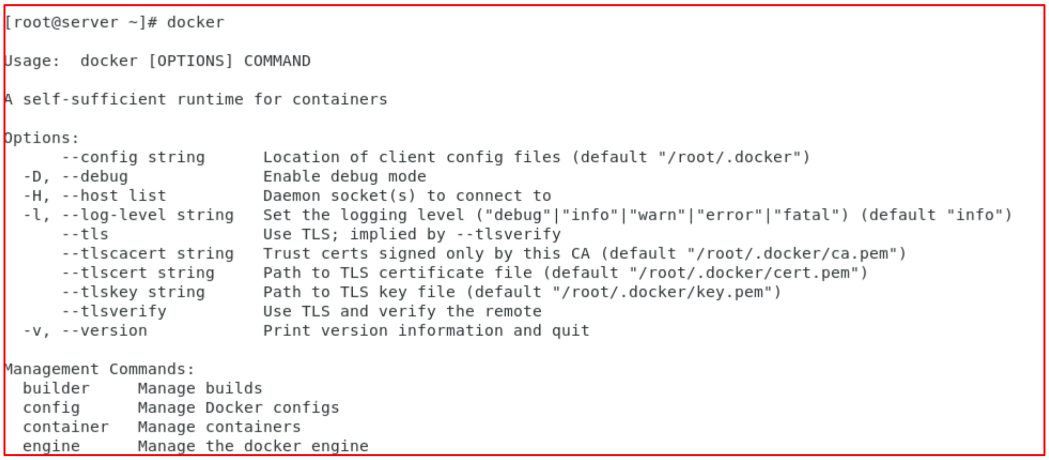
Sequence 2. Using the Docker Command

Using docker consists of passing it a chain of options and subcommands followed by arguments. The syntax is:

```
docker [option] [command] [arguments]
```

1. To view all available subcommands, type:

```
# docker
```



```
[root@server ~]# docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default "/root/.docker")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level ("debug"|"info"|"warn"|"error"|"fatal") (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default "/root/.docker/ca.pem")
      --tlscert string     Path to TLS certificate file (default "/root/.docker/cert.pem")
      --tlskey string      Path to TLS key file (default "/root/.docker/key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Management Commands:
  builder     Manage builds
  config      Manage Docker configs
  container   Manage containers
  engine      Manage the docker engine
```

2. To view the switches available to a specific command, type:

```
# docker docker-subcommand --help
```

3. To view system-wide information, use:

```
# docker info
```