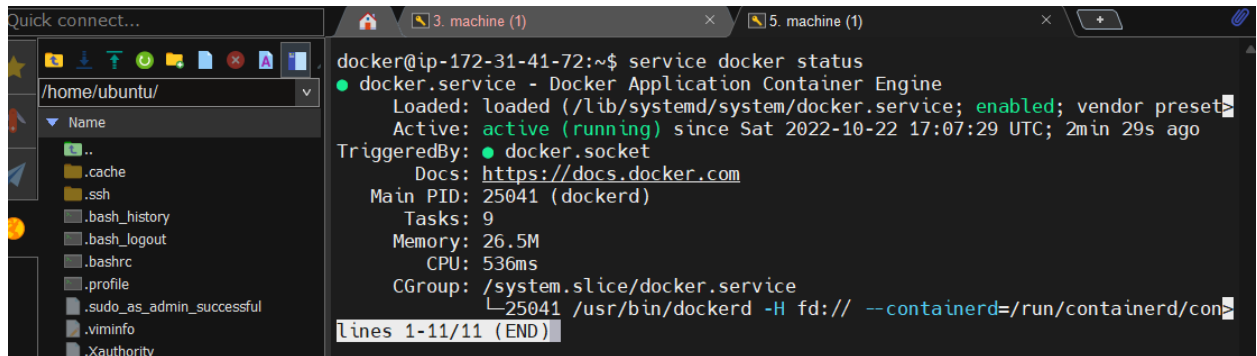


Docker commands

1. To check status = `$ service docker status`



```
docker@ip-172-31-41-72:~$ service docker status
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2022-10-22 17:07:29 UTC; 2min 29s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 25041 (dockerd)
      Tasks: 9
     Memory: 26.5M
        CPU: 536ms
    CGroup: /system.slice/docker.service
            └─25041 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

2. To check docker version = `$ docker version`

```
docker@ip-172-31-39-160:~$ docker version
Client: Docker Engine - Community
 Version: 20.10.21
  API version: 1.41
  Go version: go1.18.7
  Git commit: baeda1f
  Built: Tue Oct 25 18:01:58 2022
  OS/Arch: linux/amd64
  Context: default
  Experimental: true

Server: Docker Engine - Community
 Engine:
  Version: 20.10.21
  API version: 1.41 (minimum version 1.12)
  Go version: go1.18.7
  Git commit: 3056208
  Built: Tue Oct 25 17:59:49 2022
  OS/Arch: linux/amd64
  Experimental: false
 containerd:
  Version: 1.6.9
  GitCommit: 1c90a442489720eec95342e1789ee8a5e1b9536f
 runc:
  Version: 1.1.4
  GitCommit: v1.1.4-0-g5fd4c4d
 docker-init:
  Version: 0.19.0
  GitCommit: de40ad0
```

We can also use `docker --version` command but that will only give the version and not the complete information

```
docker@ip-172-31-39-160:~$ docker --version
Docker version 20.10.21, build baeda1f
```

3. If we need any kind of help if we get in docker we can use this command - `$ docker help`

```

docker@ip-172-31-39-160:~$ docker help

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                        "/home/docker/.docker")
  -c, --context string  Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -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
                        "/home/docker/.docker/ca.pem")
  --tlscert string      Path to TLS certificate file (default
                        "/home/docker/.docker/cert.pem")
  --tlskey string       Path to TLS key file (default
                        "/home/docker/.docker/key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  app*      Docker App (Docker Inc., v0.9.1-beta3)
  builder   Manage builds
  buildx*   Docker Buildx (Docker Inc., v0.9.1-docker)
  compose*  Docker Compose (Docker Inc., v2.12.2)
  config    Manage Docker configs
  container Manage containers
  context    Manage contexts
  image     Manage images
  manifest  Manage Docker image manifests and manifest lists
  network   Manage networks

```

4. We can directly login in to docker Hub by the following command - \$ docker login

```

docker@ip-172-31-39-160:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: aarushiii02
Password:
WARNING! Your password will be stored unencrypted in /home/docker/.docker/config
.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

```

5. We can find images in docker by using command: \$ docker images

```

docker@ip-172-31-39-160:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5  13 months ago  13.3kB
docker@ip-172-31-39-160:~$ █

```

6. We can download and run new images like MySQL: \$ docker run mysql

```

docker@ip-172-31-39-160:~$ docker run mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
50cbc88660a5: Pull complete
92ca853f7184: Pull complete
9a2047696230: Pull complete
fe3fea56f9fb: Pull complete
b058249d3104: Pull complete
9d5014a20163: Pull complete
906aa7388ee2: Pull complete
86b5e2150967: Pull complete
7c6b15dcdf4e: Pull complete
21de4337b977: Pull complete
35dab154f2ae: Pull complete
Digest: sha256:06314a7a220f6043436cfd72fd9c7f174fd58ef69fe4b788625fa53be4ab66aa
Status: Downloaded newer image for mysql:latest
d2022-10-26 04:40:06+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
2022-10-26 04:40:06+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2022-10-26 04:40:06+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
2022-10-26 04:40:06+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
You need to specify one of the following as an environment variable:
- MYSQL_ROOT_PASSWORD
- MYSQL_ALLOW_EMPTY_PASSWORD
- MYSQL_RANDOM_ROOT_PASSWORD
docker@ip-172-31-39-160:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mysql         latest   8fad08b3c84b   4 days ago    535MB
hello-world   latest   feb5d9fea6a5   13 months ago 13.3kB
docker@ip-172-31-39-160:~$

```

7. **To remove a docker image we can use:** `docker rmi -f (image id)`

for example here we are removing mysql image

```

docker@ip-172-31-39-160:~$ docker rmi -f 8fad08b3c84b
Untagged: mysql:latest
Untagged: mysql@sha256:06314a7a220f6043436cfd72fd9c7f174fd58ef69fe4b788625fa53be4ab66aa
Deleted: sha256:8fad08b3c84be3e9164f86153224ab616bf71ee2c79677154c2e5cd3179cccf
docker@ip-172-31-39-160:~$

```

8. **To run the container in a detached mode:** `$ docker run -d mysql`

```

docker@ip-172-31-39-160:~$ docker run -d mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
50cbc88660a5: Already exists
92ca853f7184: Already exists
9a2047696230: Already exists
fe3fea56f9fb: Already exists
b058249d3104: Already exists
9d5014a20163: Already exists
906aa7388ee2: Already exists
86b5e2150967: Already exists
7c6b15dcdf4e: Already exists
21de4337b977: Already exists
35dab154f2ae: Already exists
Digest: sha256:06314a7a220f6043436cfd72fd9c7f174fd58ef69fe4b788625fa53be4ab66aa
Status: Downloaded newer image for mysql:latest
f11ab2853c42eee2b27d5138efd85b1eb7ffde604b061316ea7e7ae280e376d9
docker@ip-172-31-39-160:~$ docker run -d mysql
7a3c0836f29c36bbd0c063620d5e7d74165a2a5e0215f40314a79d3517c7324f

```

We will get the id of the container as the output.

9. We see the docker container id and the images running on it by using: `$docker ps -a` or `$docker ps`

```
docker@ip-172-31-39-160:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
7a3c0836f29c   mysql         "docker-entrypoint.s... 6 minutes ago   Exited
(1) 6 minutes ago          silly_moser
f11ab2853c42   mysql         "docker-entrypoint.s... 6 minutes ago   Exited
(1) 6 minutes ago          fervent_kepler
956f52d52034   mysql         "docker-entrypoint.s... 44 minutes ago  Exited
(1) 44 minutes ago          competent_cohen
79e688444e44   hello-world   "/hello"                About an hour ago Exited
(0) About an hour ago          gracious_neumann
```

10. We can restart the container in case it crashes by using: `$ docker stop (container id)`

```
docker@ip-172-31-39-160:~$ docker stop 7a3c0836f29c
7a3c0836f29c
```

We can restart the container by using: `$ docker start (container id)`

```
docker@ip-172-31-39-160:~$ docker start 7a3c0836f29c
7a3c0836f29c
```