

Table of contents

- Table of contents
 - Docker Installation
 - Install Docker on EC2 (Manually)
 - Install Docker on EC2 (via EC2 User data)
 - Docker Commands
 - Starting and Stopping Containers

Docker Installation

Install Docker on EC2 (Manually)

- Create a simple EC2 instance with Amazon Linux 2 and execute these commands after you SSH into it.

```
sudo yum update -y
sudo amazon-linux-extras install docker
docker -v
```

- Above commands will install Docker Server, Start the docker service.
- Start the docker service and verify the status of Docker Daemon

```
sudo systemctl start docker
sudo systemctl status docker
```

```
[ec2-user@ip-172-31-1-197 ~]$ sudo systemctl status docker
* docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2021-07-17 06:39:46 UTC; 3h 36min ago
     Docs: https://docs.docker.com
    Process: 7628 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
    Process: 7618 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
   Main PID: 7634 (dockerd)
      Tasks: 9
     Memory: 205.7M
    CGroup: /system.slice/docker.service
            └─7634 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nfile=1024:4096

Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.075372236Z" level=info msg="ClientConn sw...=grpc
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.125677984Z" level=info msg="Loading conta...art."
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.311067572Z" level=info msg="Default bridg...ress"
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.369856229Z" level=info msg="Loading conta...one."
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.386963306Z" level=info msg="Docker daemon...10.4
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.387468167Z" level=info msg="Daemon has co...tion"
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal systemd[1]: Started Docker Application Container Engine.
Jul 17 06:39:46 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T06:39:46.415505904Z" level=info msg="API listen on...sock"
Jul 17 07:02:27 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T07:02:27.940196427Z" level=info msg="ignoring even...lete"
Jul 17 10:16:17 ip-172-31-1-197.ec2.internal dockerd[7634]: time="2021-07-17T10:16:17.625481480Z" level=info msg="ignoring even...lete"
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-1-197 ~]$
```

- After installation of docker service, a Linux Group **docker** is created.
- Add **ec2-user** to the docker group to execute the docker commands in linux with **ec2-user**

```
docker info
cat /etc/group | grep docker
sudo usermod -a -G docker ec2-user
cat /etc/group | grep docker
docker info
```

- Logout from the EC2 instance and log back in or restart the ssh session and execute `docker info`

Install Docker on EC2 (via EC2 User data)

- Below script can be passed via EC2 User Data while creating the EC2 Instance in Step **Configure Instance Details**

```
#!/bin/sh
yum update -y
amazon-linux-extras install docker
service docker start
usermod -a -G docker ec2-user
chkconfig docker on
```

▼ Advanced Details

User data ⓘ

☒ As text ☐ As file ☐ Input is already base64 encoded

```
#!/bin/sh
yum update -y
amazon-linux-extras install docker
service docker start
usermod -a -G docker ec2-user
chkconfig docker on
```

User data script might take some time to execute and make Docker CLI ready to use.

Docker Commands

```
docker info
docker images
docker pull ubuntu:18.04
docker images
docker pull amazonlinux
# starts a container, allocates a pseudo-TTY connected to the container's stdin,
and creates an interactive bash shell in the container.
docker run -i -t ubuntu /bin/bash
docker run -i -t ubuntu:18.04 /bin/bash
docker run -i -t ubuntu:20.04 /bin/bash
docker run -i -t amazonlinux:latest /bin/bash

docker images
```

In the first `docker run` command above the default tag used is `latest`.

- Search for Docker Image in Docker Hub Registry

```
docker search nginx
```

- Connect to docker interactively

```
docker run -i -t ubuntu:20.04 /bin/bash
docker run --name=myUbuntuContainer -i -t ubuntu:20.04 /bin/bash
```

- The above command will run container with image specified, if image is not present locally it will download it using `docker pull`
- Run below commands inside the docker container bash shell

```
hostname
id
echo $HOME
pwd
```

- Detach docker from interactive container, we can detach it from our container by using the `Ctrl + P` and `Ctrl + Q` escape sequence. This escape sequence will detach the TTY from the container and land us in the Docker host prompt \$, however the container will continue to run.

Starting and Stopping Containers

- Start a container

```
docker start [CONTAINER]
```

- Stop a running container

```
#Stop the container - docker stop <CONTAINER_ID> OR docker stop <CONTAINER_NAME>
sudo docker ps -a
sudo docker stop ContainerID
sudo docker ps -a
```

- Stop a running container and start it up again

```
docker restart [CONTAINER]
```

- Attach local standard input, output, and error streams to a running container

```
docker attach [CONTAINER]
```

- Run below commands in EC2 linux shell

```
sudo docker ps
sudo docker ps --no-trunc
```

- Below is the information for the details related to **docker ps** command
 - **CONTAINER ID**: This shows the container ID associated with the container.
 - **IMAGE**: This shows the image from which the Docker container has been created.
 - **COMMAND**: This shows you the command executed during the container launch.
 - **CREATED**: Time when the container was created.
 - **STATUS**: Current status of the container.
 - **PORTS**: This tells you if any port has been assigned to the container.
 - **NAMES**: The Docker engine auto-generates a random container name by concatenating an adjective and a noun. Either the container ID or its name can be used to take further action on the container. The container name can be manually configured by using the **--name** option in the docker run subcommand.
- Connect back to container prompt, replace below **CONTAINER_ID** with actual container id value also **CONTAINER_NAME** with actual container name.

```
sudo docker attach CONTAINER_ID
OR
sudo docker attach CONTAINER_NAME
```

- Track the changes inside the containers, connect into shell of the docker and create some directories and files inside the container.

```
sudo docker run -i -t ubuntu:18.04 /bin/bash
cd /home
ls -al
touch file{1..5}
touch {f1.txt,f2.txt,f3.txt}
ls -al
```

- Use docker diff command to check the changes made inside the container

```
sudo docker ps
sudo docker diff ContainerID
```