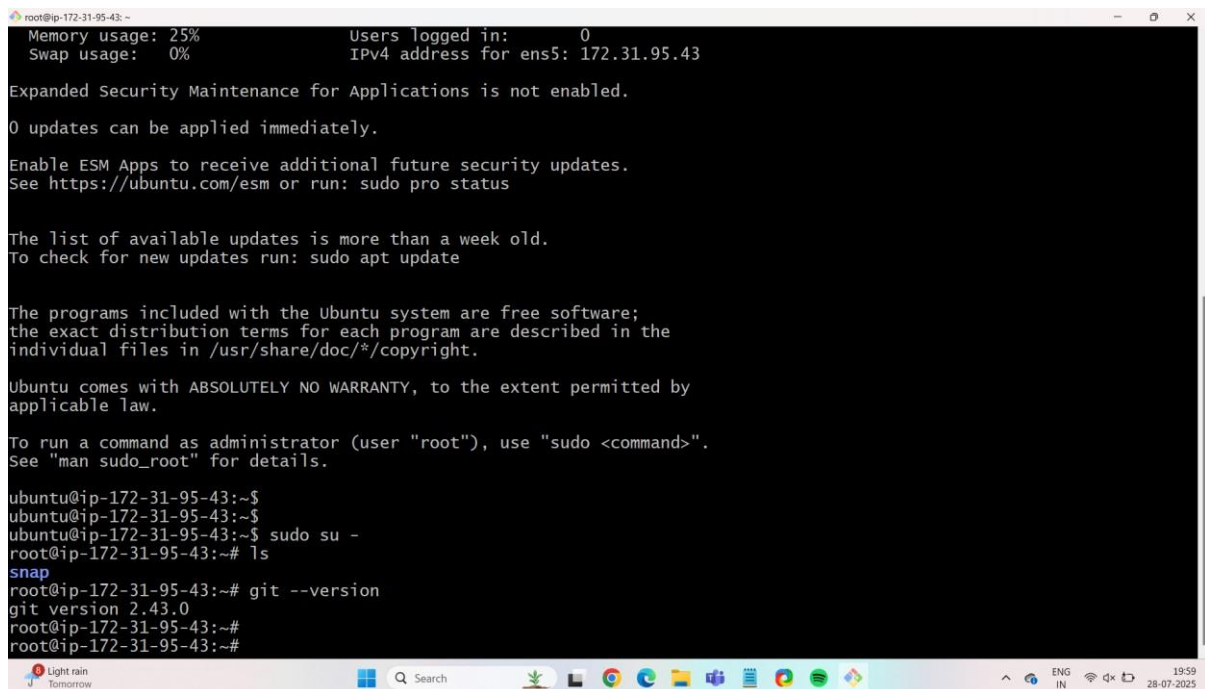**Name : Laxmi Swami**

Topic : Docker Volume &sharing volume & Create customized image.

Install docker in ubuntu OS. curl -fsSL

https://get.docker.com -o install-docker.sh sh install-

docker.sh docker version :28.3.2 git version :2.43.0

- create one directory: mkdir aws
- Create docker volume
- Docker run –name cont1 -v /aws -it imageid • Check that "/aws" is mounted in docker.

1. Create files & directories in container inside /data.

```
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-95-43:~$
ubuntu@ip-172-31-95-43:~$
ubuntu@ip-172-31-95-43:~$ sudo su -
root@ip-172-31-95-43:~# ls
snap
root@ip-172-31-95-43:~# git --version
git version 2.43.0
root@ip-172-31-95-43:~#
root@ip-172-31-95-43:~# apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1023 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1282 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [260 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [163 kB]
```

```
root@ip-172-31-95-43:~#
root@ip-172-31-95-43:~#  curl -fsSL https://get.docker.com -o install-docker.sh
root@ip-172-31-95-43:~# sudo sh install-docker.sh
# Executing docker install script, commit: 8555c7c554f6c7e4b2e67dd644b4dc46297330c2
+ sh -c apt-get -qq update >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install ca-certificates curl >/dev/null
+ sh -c install -m 0755 -d /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" -o /etc/apt/keyrings/docker.asc
+ sh -c chmod a+r /etc/apt/keyrings/docker.asc
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu noble s
table" > /etc/apt/sources.list.d/docker.list
+ sh -c apt-get -qq update >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install docker-ce docker-ce-cli containerd.io docker-compose-plu
gin docker-ce-rootless-extras docker-buildx-plugin docker-model-plugin >/dev/null
Scanning processes...
Scanning linux images...
+ sh -c docker version
Client: Docker Engine - Community
 Version:           28.3.2
 API version:       1.51
 Go version:        go1.24.5
 Git commit:        578ccf6
 Built:             Wed Jul  9 16:13:45 2025
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          28.3.2
  API version:      1.51 (minimum version 1.24)
  Go version:       go1.24.5
  Git commit:       e77ff99
  Built:            Wed Jul  9 16:13:45 2025
  OS/Arch:          linux/amd64
```

```
root@ip-172-31-95-43:~# docker images
REPOSITORY    TAG      IMAGE ID    CREATED     SIZE
root@ip-172-31-95-43:~#
root@ip-172-31-95-43:~# docker ps
CONTAINER ID   IMAGE      COMMAND    CREATED    STATUS    PORTS     NAMES
root@ip-172-31-95-43:~#
root@ip-172-31-95-43:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
32f112e3802c: Pull complete
Digest: sha256:a08e551cb33850e4740772b38217fc1796a66da2506d312abe51acda354ff061
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
root@ip-172-31-95-43:~#
```

In that aws directory create some files:

- Check the mount location for con1 container use following command:.
- Docker inspect con1

```
root@3e9488bb9888:/aws# ls
abc.txt  file.txt
root@3e9488bb9888:/aws# echo "hello" > lmn.txt
root@3e9488bb9888:/aws# cat lmn.txt
hello
root@3e9488bb9888:/aws# cd ..
root@3e9488bb9888:/# exit
exit
root@ip-172-31-95-43:~#
root@ip-172-31-95-43:~#
root@ip-172-31-95-43:~# docker ps
CONTAINER ID   IMAGE       COMMAND     CREATED    STATUS     PORTS      NAMES
root@ip-172-31-95-43:~# docker ps -a
CONTAINER ID   IMAGE           COMMAND       CREATED        STATUS                  PORTS      NAMES
3e9488bb9888   65ae7a6f3544    "/bin/bash"   2 minutes ago  Exited (0) 25 seconds ago          mydata
root@ip-172-31-95-43:~# docker inspect 3e9488bb9888
[
    {
        "Id": "3e9488bb98885e25b1f6ff5bc33182c705ca243a4a8bbd49423b19577e711879",
        "Created": "2025-07-28T14:34:02.348705237Z",
        "Path": "/bin/bash",
        "Args": [],
        "State": {
            "Status": "exited",
            "Running": false,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 0,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2025-07-28T14:34:02.441955158Z",
            "FinishedAt": "2025-07-28T14:36:26.964353051Z"
```

```
        "GraphDriver": {
            "Data": {
                "ID": "3e9488bb98885e25b1f6ff5bc33182c705ca243a4a8bbd49423b19577e711879",
                "LowerDir": "/var/lib/docker/overlay2/ba5efcf008fef802ac363791a133710cf22022a75e5f9d7416a7596a5b2168a5
-init/diff:/var/lib/docker/overlay2/9c5eb4107ee8197313e818af16a34713742f31abc1265b8d52becabab85e1334/diff",
                "MergedDir": "/var/lib/docker/overlay2/ba5efcf008fef802ac363791a133710cf22022a75e5f9d7416a7596a5b2168a
5/merged",
                "UpperDir": "/var/lib/docker/overlay2/ba5efcf008fef802ac363791a133710cf22022a75e5f9d7416a7596a5b2168a5
/diff",
                "WorkDir": "/var/lib/docker/overlay2/ba5efcf008fef802ac363791a133710cf22022a75e5f9d7416a7596a5b2168a5/
work"
            },
            "Name": "overlay2"
        },
        "Mounts": [
            {
                "Type": "volume",
                "Name": "bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023",
                "Source": "/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_d
ata",
                "Destination": "/aws",
                "Driver": "local",
                "Mode": "",
                "RW": true,
                "Propagation": ""
            }
        ],
        "Config": {
            "Hostname": "3e9488bb9888",
            "Domainname": "",
            "User": "",
            "AttachStdin": true,
            "AttachStdout": true,
            "AttachStderr": true,
```

- If your container is exit or stop then use following command for start container & copy source:
- Go to the source location and check the files & dir created in container.

```
              "IPv6Gateway": "",
              "MacAddress": "",
              "Networks": {
                  "bridge": {
                      "IPAMConfig": null,
                      "Links": null,
                      "Aliases": null,
                      "MacAddress": "",
                      "DriverOpts": null,
                      "GwPriority": 0,
                      "NetworkID": "e06234e1f14eacca37569d62315bfbd7bd47331ad858c216e6558396fb0ded7b",
                      "EndpointID": "",
                      "Gateway": "",
                      "IPAddress": "",
                      "IPPrefixLen": 0,
                      "IPv6Gateway": "",
                      "GlobalIPv6Address": "",
                      "GlobalIPv6PrefixLen": 0,
                      "DNSNames": null
                  }
              }
          }
      }
]
```

```
root@ip-172-31-95-43:~# cd /var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_d
ata
root@ip-172-31-95-43:/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_data# l
s
abc.txt  file.txt  lmn.txt
root@ip-172-31-95-43:/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_data#
root@ip-172-31-95-43:/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_data#
root@ip-172-31-95-43:/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_data# c
d
root@ip-172-31-95-43:~#
```

```
root@ip-172-31-95-43:~# docker ps -a
CONTAINER ID   IMAGE         COMMAND       CREATED         STATUS                   PORTS     NAMES
3e9488bb9888   65ae7a6f3544  "/bin/bash"   4 minutes ago   Exited (0) 2 minutes ago           mydata
root@ip-172-31-95-43:~#
root@ip-172-31-95-43:~# docker rm 3e9488bb9888
3e9488bb9888
root@ip-172-31-95-43:~#
root@ip-172-31-95-43:~# cd /var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_d
ata
root@ip-172-31-95-43:/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_data#
root@ip-172-31-95-43:/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_data#
root@ip-172-31-95-43:/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_data# l
s
abc.txt  file.txt  lmn.txt
root@ip-172-31-95-43:/var/lib/docker/volumes/bf70f2fcc4f85fcd6273c5f1845c331f91c5b367179cefb37bfe34a6216b3023/_data#
```

- Volume sharing by con2 or con3
- Create new container with volume sharing from con1.
- Docker run –name con2 –volumes-from con1 -it ubuntu
- Docker run –name con3 –volumes-from con1 -it ubuntu
- Check whether the files of con1 are visible.
- Create new files from cont2.

- Now use docker attach con1 command for run container:
- In con3 we will create one file con3.txt this data will visible on con1 and con2 also

```
root@07b6cf7e50e6: /aws                                                                      —  □  ×

root@ip-172-31-95-141:/# docker -v
Docker version 28.3.2, build 578ccf6
root@ip-172-31-95-141:/# docker attach con3
root@45911a2ffe57:/# ls
aws  bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@45911a2ffe57:/# cd aws
root@45911a2ffe57:/aws# ls
abc.txt  cds.txt  file.java  file.txt  file1.txt
root@45911a2ffe57:/aws# touch con3.txt
root@45911a2ffe57:/aws#
root@45911a2ffe57:/aws# cd ..
root@45911a2ffe57:/# read escape sequence
root@ip-172-31-95-141:/# ls
bin   dev  home  lib32  libx32      media  opt   root  sbin  srv  tmp  var
boot  etc  lib   lib64  lost+found  mnt    proc  run   snap  sys  usr
root@ip-172-31-95-141:/#
root@ip-172-31-95-141:/# docker ps
CONTAINER ID   IMAGE    COMMAND       CREATED         STATUS         PORTS     NAMES
45911a2ffe57   ubuntu   "/bin/bash"   8 minutes ago   Up 8 minutes             con3
9d759933837d   ubuntu   "/bin/bash"   14 minutes ago  Up 14 minutes            con2
07b6cf7e50e6   ubuntu   "/bin/bash"   18 minutes ago  Up 18 minutes            co1
root@ip-172-31-95-141:/# docker attach con2
root@9d759933837d:/# ls
aws  bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@9d759933837d:/# cd aws
root@9d759933837d:/aws# ls
abc.txt  cds.txt  con3.txt  file.java  file.txt  file1.txt
root@9d759933837d:/aws# cd ..
root@9d759933837d:/# read escape sequence
root@ip-172-31-95-141:/# ls
bin   dev  home  lib32  libx32      media  opt   root  sbin  srv  tmp  var
boot  etc  lib   lib64  lost+found  mnt    proc  run   snap  sys  usr
root@ip-172-31-95-141:/# docker attach co1
root@07b6cf7e50e6:/# ls
aws  bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@07b6cf7e50e6:/# cd aws
root@07b6cf7e50e6:/aws# ls
abc.txt  cds.txt  con3.txt  file.java  file.txt  file1.txt
root@07b6cf7e50e6:/aws# |
```