DOCKER VOLUMES

* When we create a Container then Volume will be created. Volume is simply a directory inside our container.
* First, we have to declare the directory Volume and then share Volume. Even if we stop the container still, we can access the volume.
* You can declare directory as a volume only while creating container. We can’t create volume from existing container.
* You can share one volume across many number of Containers. Volume will not be included when you update an image.
* If Container-1 volume is shared to Container-2 the changes made by Container-2 will be also available in the Container-1.

**You can map Volume in two ways:**

1. Container < ------ > Container

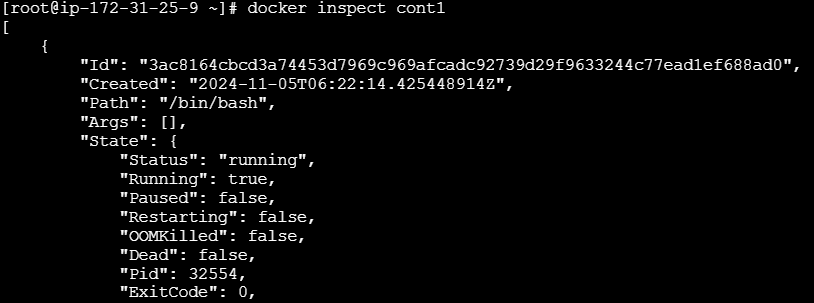
2. Host < ------- > Container

**USES OF VOLUMES:**

* Decoupling Container from storage.
* Share Volume among different Containers.
* Attach Volume to Containers.
* On deleting Container Volume will not be deleted.

Step1: Lunch instance amazon linux server and connect.

* sudo su –
* yum install docker –y
* systemctl start docker
* docker run –it --name cont1 ubuntu
* ll
* mkdir docker1 **( create the docker)**
* docker inspect cont1 **( we can see the all information continer)**

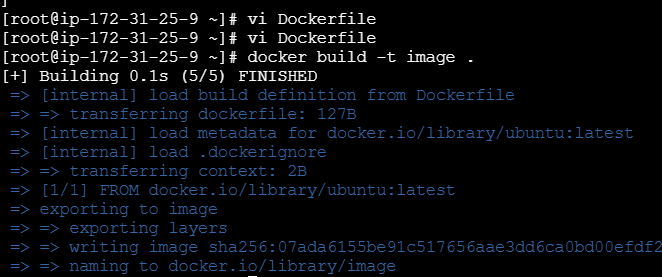


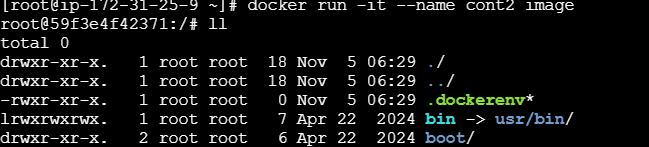
**Step2: create the Docker file:**

* vi Dockerfile

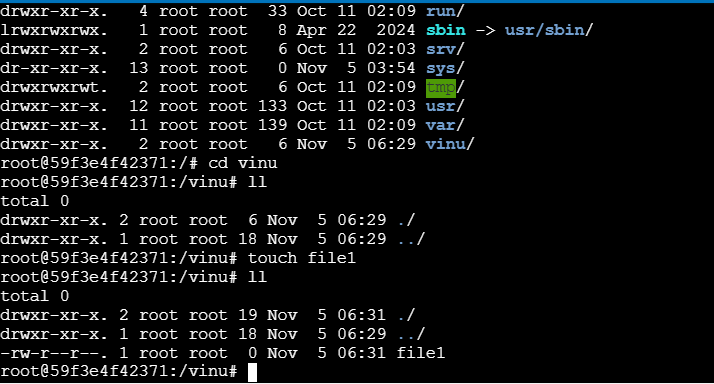
C:\Users\ADMIN\Pictures\Screenshots\Screenshot (181).png

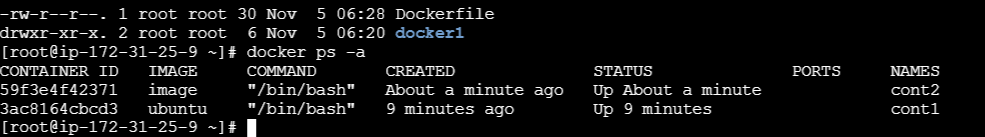
* docker build -t image .
* docker run –it - -name cont2 image



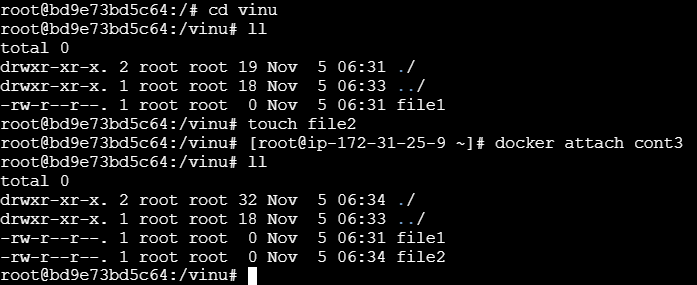


* ll
* cd vinu **( here go to inside the file)**
* ll
* touch file1
* ll
* ctrl p q (exit )
* ll
* docker ps –a
* docker run –it –name cont 3 –privileged=true –volumes-from cont2 image
* ll





* **cd vinu**
* **ll**
* **touch file2**

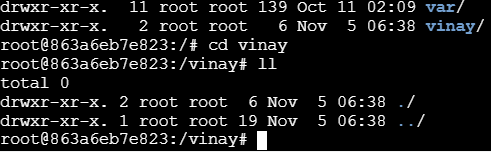


* ctrl p q
* docker attach cont3
* ll
* cd
* ll
* cd /
* ll
* rmdir vinu **( divice or resource is busy)**
* mkdir dir2
* rmdir dir2 **(delete directory)**
* ll
* ctrl p q
* **docker run –it –name cont4 –v/vinay Ubuntu**
* **ll**

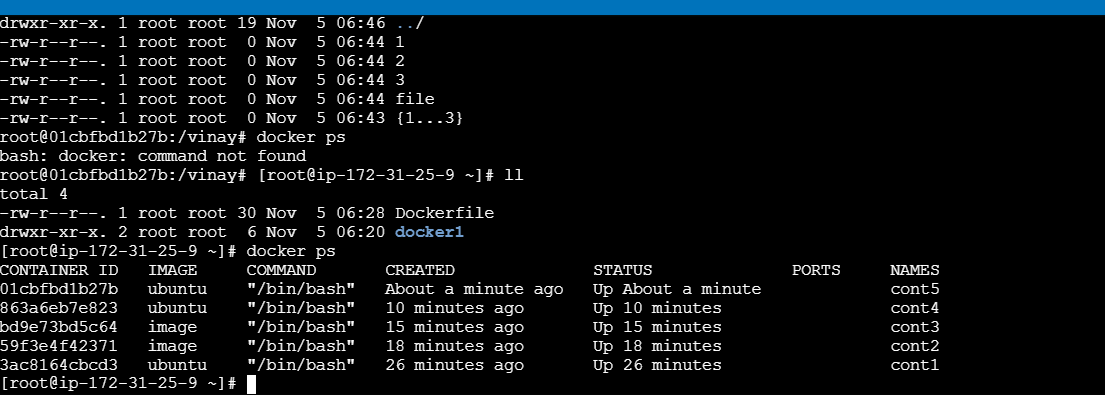
**Note: while creation of image we don’t have volume.**

**Here going to inside of the volume:**

* cd vinay
* ll

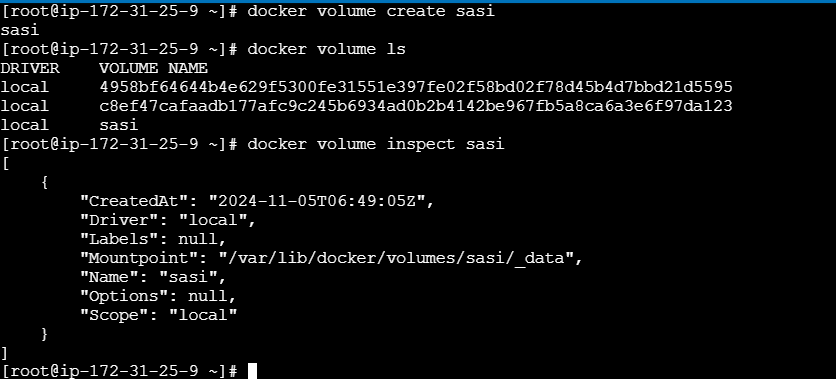


* touch file {1…3}
* ll
* rm –rf file{1..3}
* touch file{1..3}
* ll
* cd
* ctrl p q
* docker ps
* dpcker run –it --name cont5 –volumes-from cont4 –privileged=true Ubuntu
* ll
* cd vinay
* ll
* ctrl p q
* ll
* docker ps

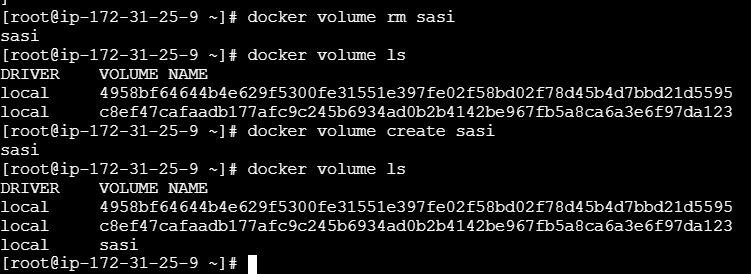


**Here create the volume:**

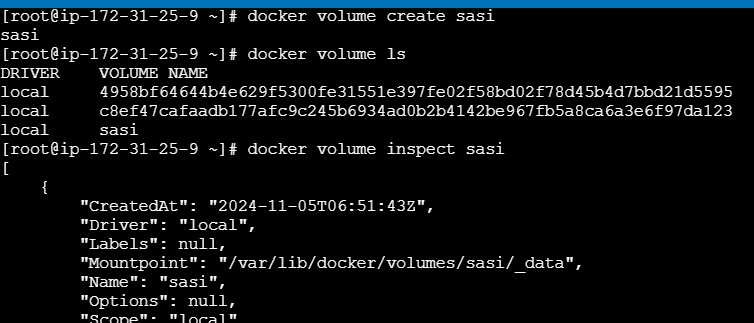
* docker volume create sasi
* docker volume ls **( see the all volumes data)**
* docker volume inspect sasi **( when it has created using mountpoint we can able to attach)**

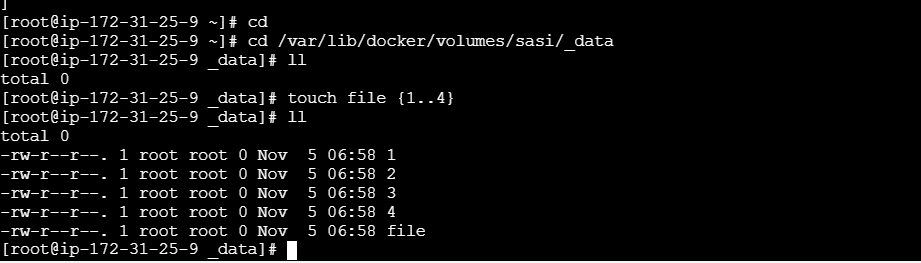


* docker volume rm **s**asi **( remove volume)**
* docker volumes ls **(see the volume list)**
* **after remove volume and again create the volume**
* docker cretate sasi
* docker volume ls



* docker volume inspect sasi
* cd  **( copy the mountpoint go to inside the data)**
* ll
* touch file {1..4}
* ll
* cd





* docker run –it –name cont6 –mount source=sasi,destination=/sasiM Ubuntu
* ll
* cd var
* ll
* cd lib
* ll
* cd
* ll
* cd **/ ( volume will be created)**
* ll
* cd sasiM/ **( go to the inside volume)**
* ll
* rm –rf file{1..4} **( remove the file)**
* **ctrl p q**
* ll
* docker volume ls **( here you will see the continer data)**
* docker volume inspect sasi
* cd ( copy “mountup’)
* ll

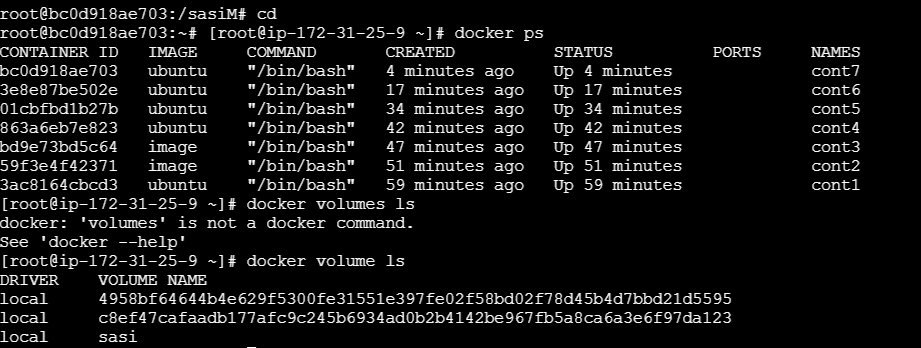
****

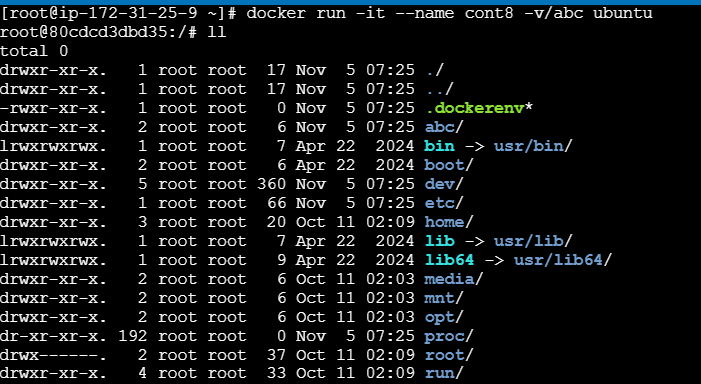
**Here sharing the volume:**

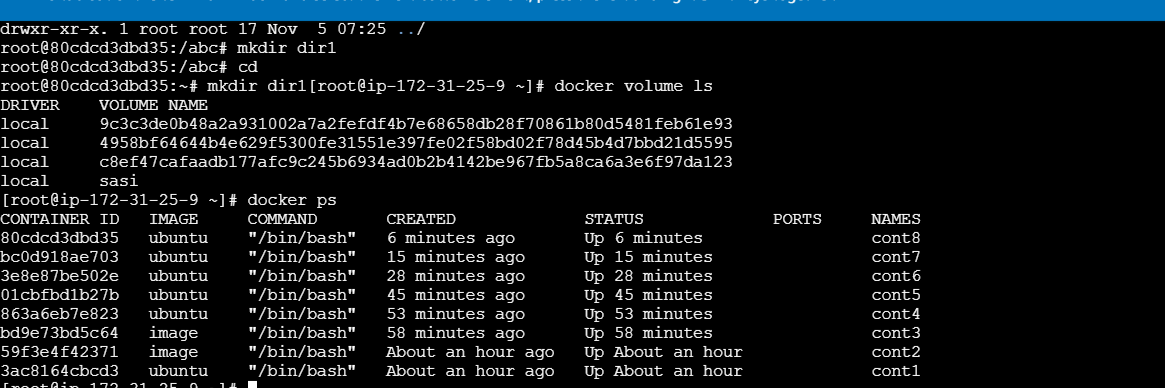
* **docker run –it –name cont 7 –volumes-from cont6 –privileged=true Ubuntu**
* **ll**
* **cd sasiM/**
* **ll**

**Note : volume will not delete directly and continar will be delete.**

* **Cd**
* **Ctrl p q**
* **Docker ps**







Docker networking allows Docker containers, services, and components to connect with each other and with external components. It also provides isolation for Docker containers.

A Docker bridge network is a software bridge that allows containers on the same Docker daemon host to communicate with each other, while keeping them isolated from containers on other bridge networks. It is also default bridge.

In Docker, "none" refers to a network driver that isolates a container from any external network.

A Docker host is a physical or virtual server that runs Docker containers and workloads. It's a platform that provides an environment for executing applications, and includes the Docker daemon, containers, images, networks, and storage

