

DevSecOps Training

DAY – 1

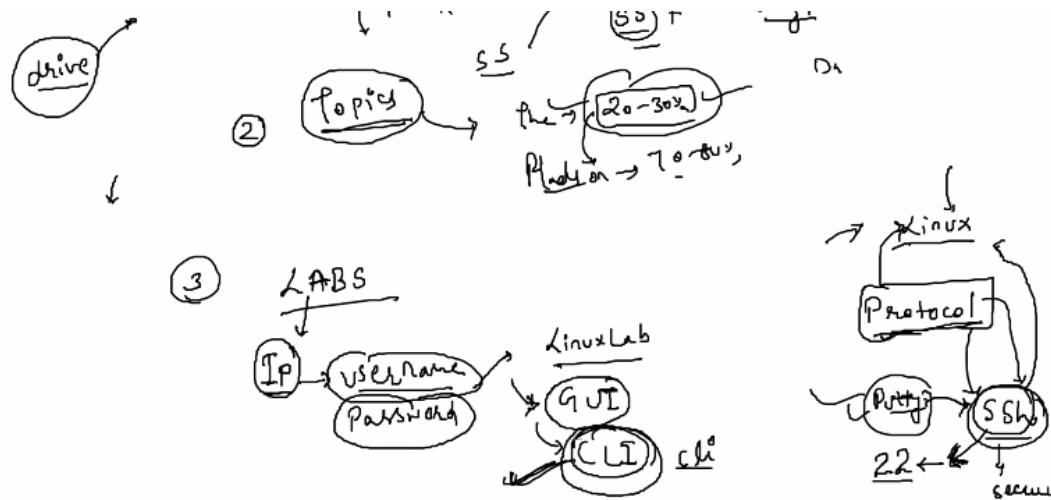
Trainer: Mayank Modi – Red Hat Certified Architect.

Agenda:

Idea on how to deploy the software

How to secure the data while deploying

Relevant materials, screenshot of the material will be shared by the Trainer



Note the IP address, username and Password:

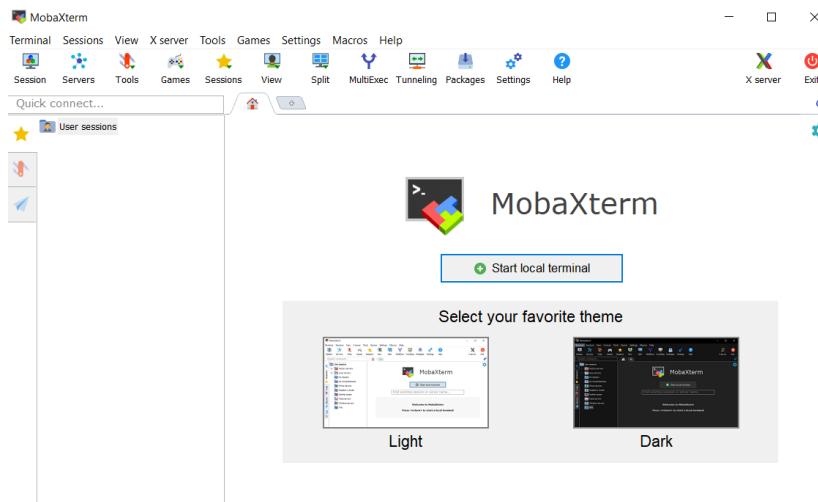
DevSecOps_Collins_25-04-2022

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

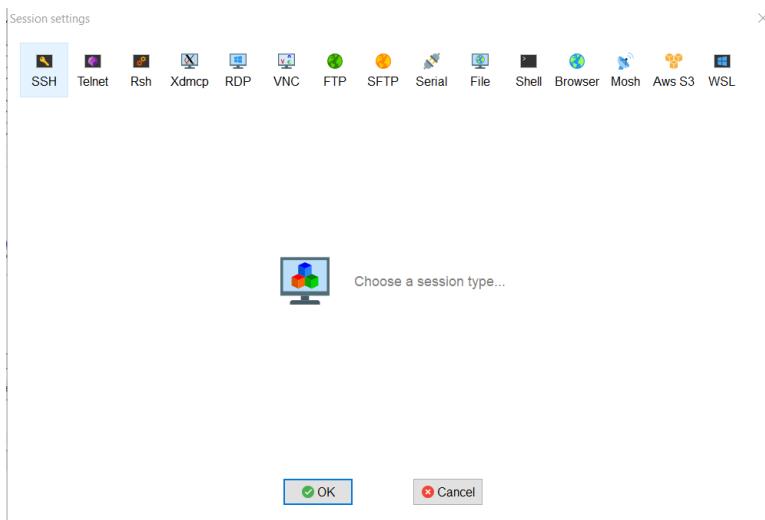
E25 - fx | vasihnavi

A	B	C	D	E	F
S.No	Ip Address- Day1	User Name	Password	Participants Name	
1	Trainer	54.85.90.121	root	redhat	Trainner
2	1 3.87.7.124	root	redhat	alpana	
3	2 3.82.120.189	root	redhat	baljeet	
4	3 3.87.48.63	root	redhat	Devraj	
5	4 54.165.233.74	root	redhat	ganesha	
6	5 54.152.136.178	root	redhat	kaustubh	
7	6 52.90.7.33	root	redhat	manish	
8	7 18.204.19.9	root	redhat	manoj	
9	8 18.233.8.215	root	redhat	naveen	
10	9 54.175.99.81	root	redhat	neha	
11	10 44.204.49.76	root	redhat	pravallika	
12	11 54.159.134.45	root	redhat	ramanand sai	
13	12 18.212.10.157	root	redhat	revanth	
14	13 54.159.200.197	root	redhat	rohit	
15	14 54.210.68.104	root	redhat	rudra	
16	15 44.201.228.108	root	redhat	sahitya	
17	16 3.86.25.98	root	redhat	sampat	
18	17 54.174.206.53	root	redhat	sangamesh	
19	18 54.226.211.129	root	redhat	sashi	
20	19 44.204.195.66	root	redhat	shashidhar	
21	20 3.93.238.83	root	redhat	shreeta	
22	21 54.166.35.154	root	redhat	sudheer	
23	22 54.211.90.94	root	redhat	udey	
24	23 54.173.38.100	root	redhat	vasihnavi	
25					
26					
27					
28					

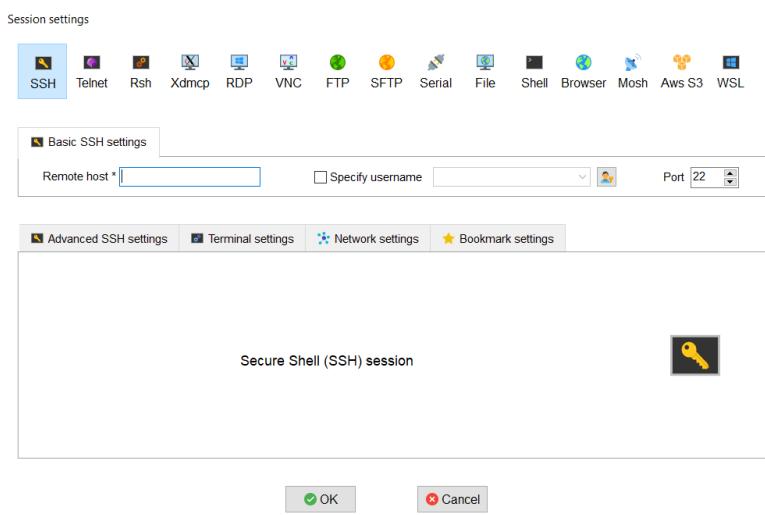
Open MobaXterm:



Select Session:



Click on SSH:

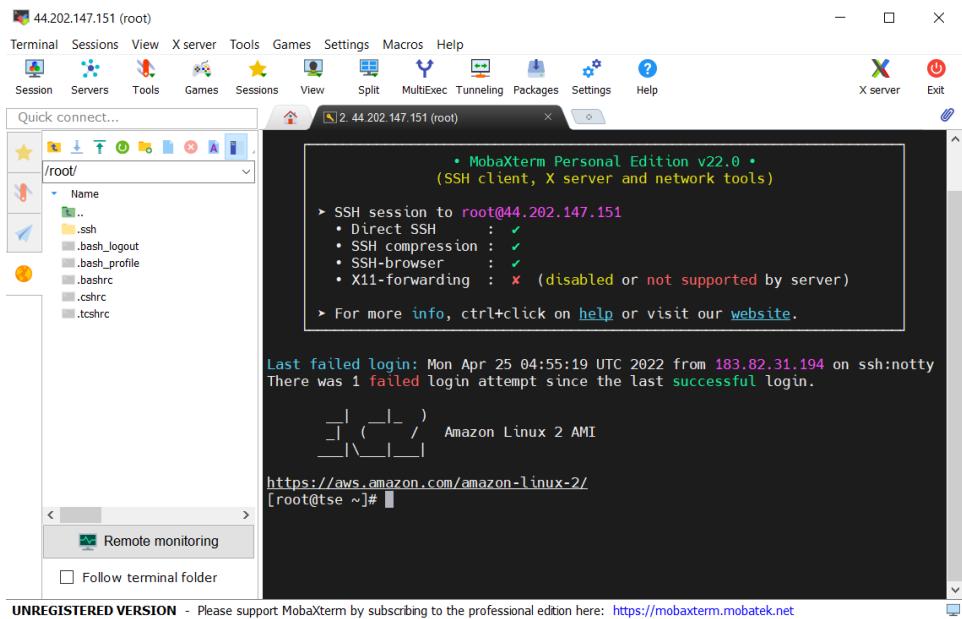


Details of the machine:

IP Address: 44.202.147.151

Username: root

Password: redhat



Optional: If you want to change the hostname.

```
[root@tse ~]# ls
[root@tse ~]# hostname
tse
[root@tse ~]# hostnamectl set-hostname Vishali
[root@tse ~]# bash
[root@vishali ~]# hostname
vishali
[root@vishali ~]#
```

DevOps + DevSecOps:

Example:



We should have a PLAN to do this - Where and how to code it?

We have phases to do it – Software Development Life Cycle – SDLC

We need code to perform this. One is called monolithic application and microservice application.

Difference between Monolithic Architecture and Microservices Architecture:

Sr. No.	Key	Monolithic architecture	Microservices architecture
1	Basic	Monolithic architecture is built as one large system and is usually one code-base	Microservices architecture is built as small independent module based on business functionality
2	Scale	It is not easy to scale based on demand	It is easy to scale based on demand.
3	Database	It has shared database	Each project and module has their own database
4	Deployment	Large code base makes IDE slow and build time gets increase.	Each project is independent and small in size. So overall build and development time gets decrease.
5	Tightly Coupled and Loosely coupled	It extremely difficult to change technology or language or framework because everything is tightly coupled and depend on each other	Easy to change technology or framework because every module and project is independent

Examples of Monolithic Applications:

Testing code is of two types – Manual and Automated Testing

Developer will try to upload the code into GitHub and operation buy will download it from the GitHub

Github/BitBucket/GitLab - Version Control System (VCS) and Source Control Management (SCM)



Deploy it – Operator should know from where to take the code. If there are some updates V1 -> V2. Github will reflects it.

Traditional Approach:

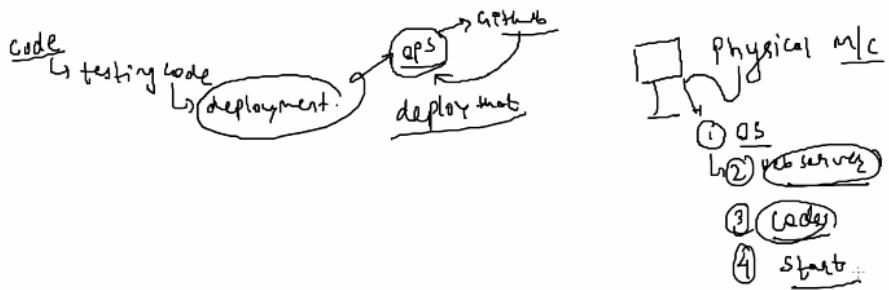
We need one physical machine and install operating system, web server, put the codes and start the applications.

Disadvantage:

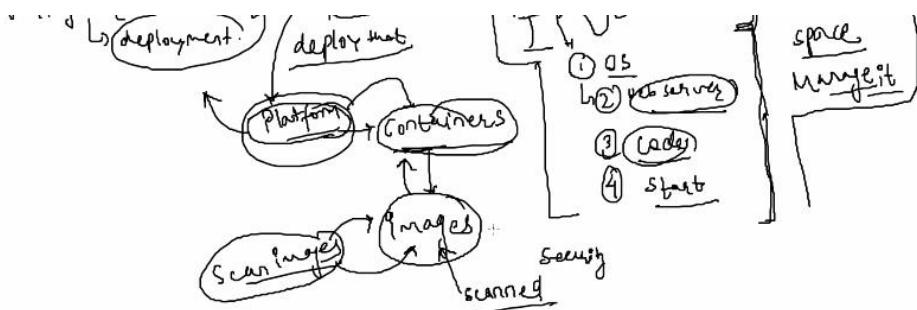
Consume more time

Consume more space

Someone has to manage it



Now, we introduce a platform where we will be deploying our application is by using containers. Containers use images to do it. Images are scanned for security purposes. After scanning the images only, we will make the containers. If container is running, then your application is also running.

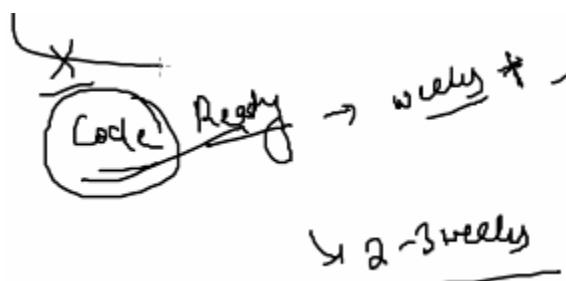


Steps:

Code -> Test -> GitHub -> Platform -> Image -> Image Scanning -> Containers -> Access the Application

In this the developers, operations and troubleshoot guys will be coming. All these works can be done by a single person. That single person should know about the develop, operations and problem solving. This guy is called DevOps Engineer.

Approximate time to do all these processes: 1 month. If the coding part is removed, then it will take 2-3 weeks.



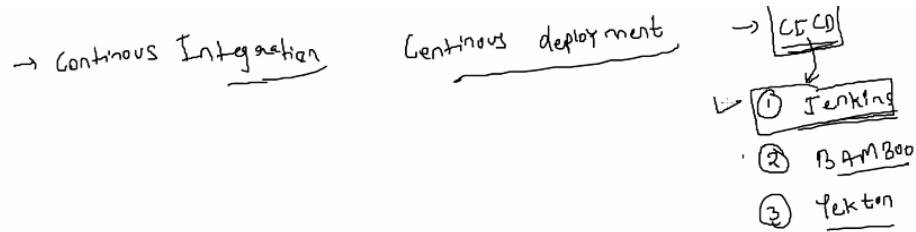
It changes the code, whenever there is a need for an update ----> Continuous Integration

It deploys the code, whenever there is an update -----> Continuous Deployment

Here, we are creating a CI/CD Pipeline. This overall process is called CICD.

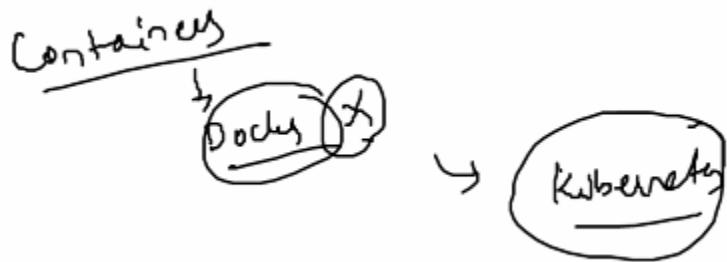
There are multiple tools which helps in CICD:

This is with security.



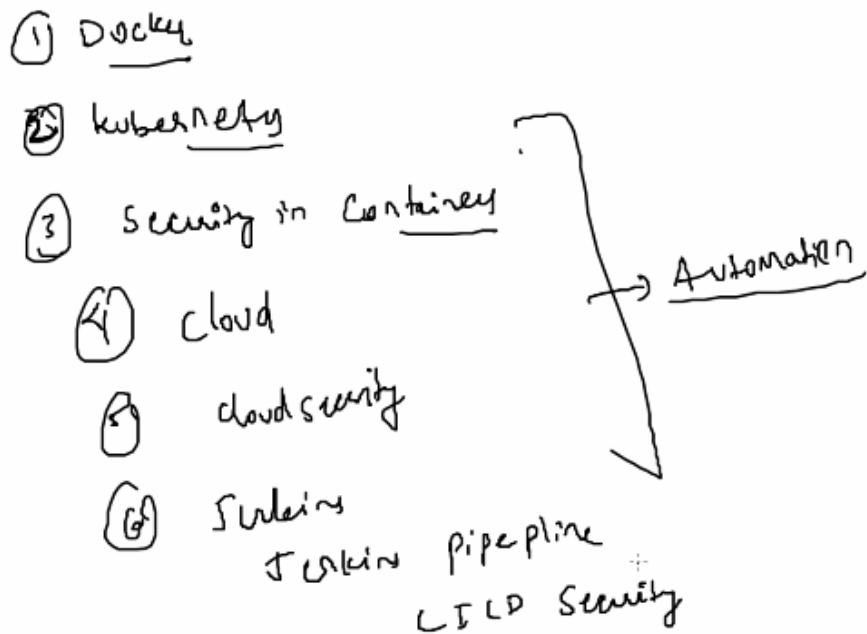
We need to create pipelines in Jenkins to perform CICD.

Containers -> Docker (not sufficient) -> Move to Kubernetes part.



Tools which are going to be covered in this training:

1. Docker
2. Kubernetes
3. Security in Containers
4. Cloud
5. Cloud Security
6. Jenkins, Jenkins Pipeline and CICD Security

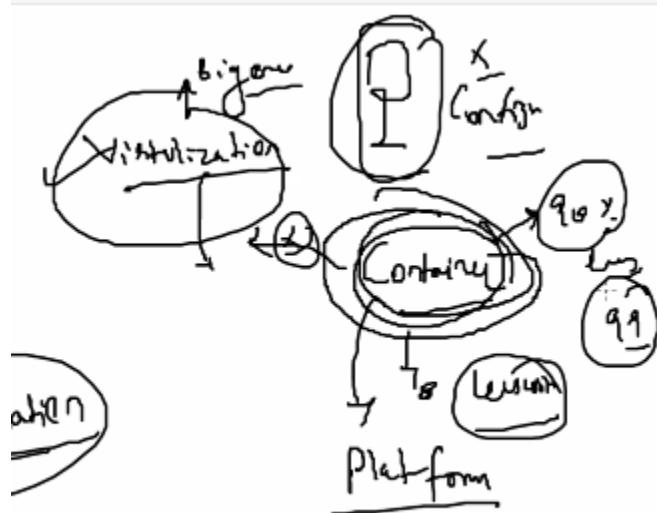


Where will your application run? Containers

In earlier stages, we need to buy physical servers and deploy our application. Now we will use containers to deploy your application with less cost.

Jenkins is responsible to be the updated code connected to operations team.

In virtualization, we can create multiple operating systems. Containers will take 99% less.

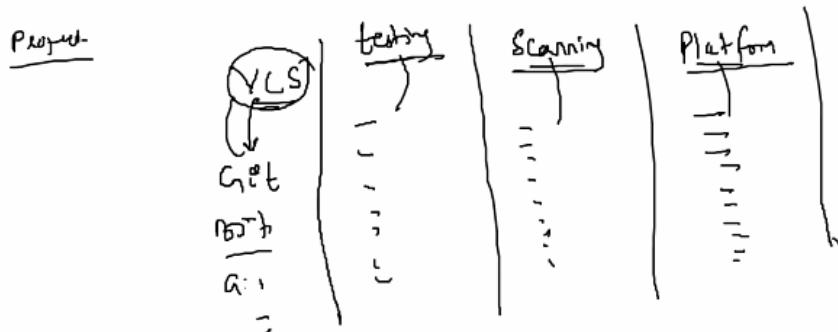


Question:

In exploring the opportunity in the DevOps, do we need to have prerequisite to go for the defined approach?

Security, efficient part needs to be taken care. For each phase, we have multiple tools.

Project:



If you know the objective and know any one tool, then you can implement the same knowledge with other tools. Major part is to understand the basics of it.

If you take the phases of Cloud Server Provider – AWS, GCP and Azure. AWS is the most used along with AI/ML in Cloud.

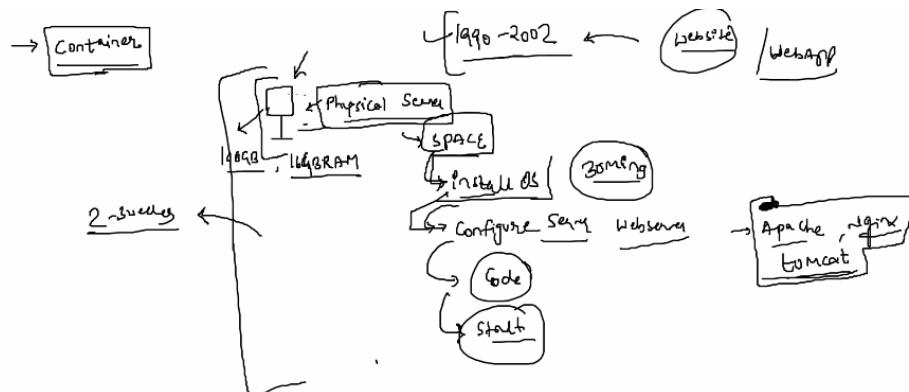
If you talk about automation or configuration management tools – Ansible

For containerization, the best is – Docker

For Orchestration purposes, the best one is – Kubernetes

For CICD Pipeline, the best one is Jenkins.

Container:



In 1992 – 2002, We need to buy a physical server with 100GB / 164 GB RAM

How much time will it take to install an operating system? 25 – 30 mins

After installation, we have to configure OS with the Web server – Apache, Nginx and Tomcat.

Then code and start.

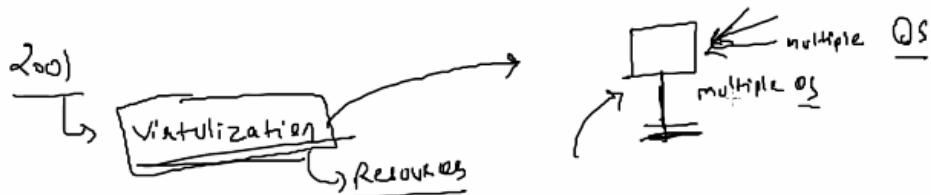
How much time it will take to do this process? 2 – 3 weeks.

Shipping the physical server -> Install OS (Windows/Linux OS) - It will take minimum 30 mins for the setup -> Configurations -> Then coding -> application start

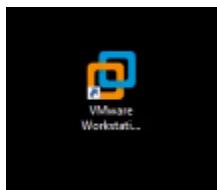
If the load increases (no of users trying to access the website) in the application, how they will manage it? Increase the machine, buy more physical servers.

How much time will it take for scalability? 11 – 13 days for a single website.

In 2001, Virtualization comes into picture. In one single machine, we can run multiple operating system.



Software - VMVirtualizationPro



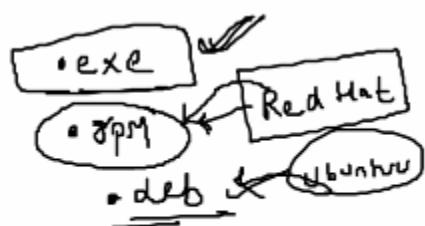
For virtual machine, we need to install a software.

Windows OS – Do you know what are the steps we need to follow for installing an OS in your machine?

.exe - executable file to run a app or software. This is for Windows

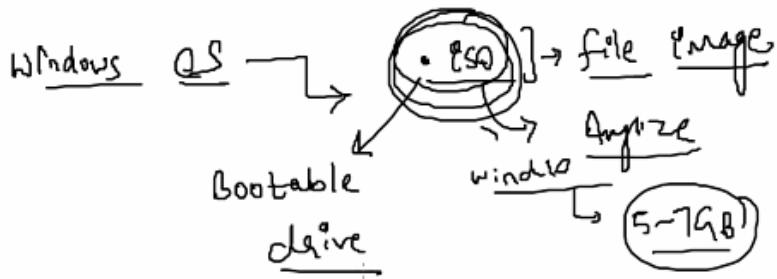
.rpm - This is for RedHat

.deb - This is for Linux and Ubuntu



To install the OS (not the software). Do you know which extension to be used?

.ios



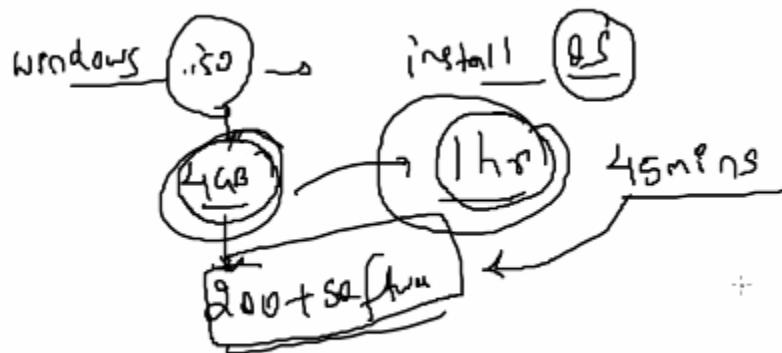
By the bootable device, we can install OS.

Pendrive – Install OS

Window – 5 – 7GB. **Why the size of any kind of ios file is bigger?**

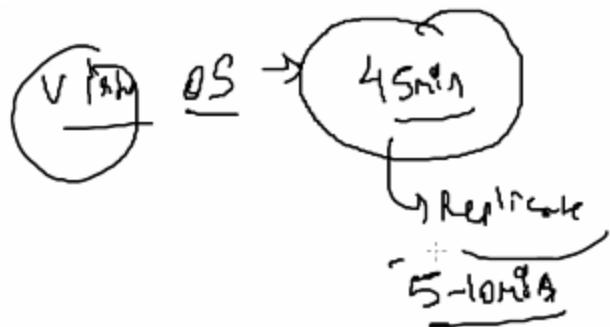
Because it has the libraries and dependency and packages of that OS.

You don't need any software to configure Apache web server. Why do you need to waste time to install a particular software?

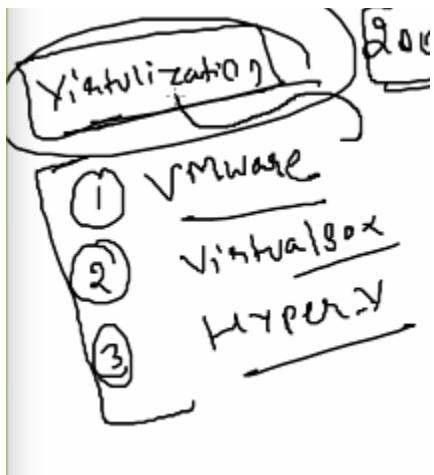


200+ Software is getting installed which we don't need. That is wasting our time (45 mins). We only need Apache web server.

In Virtualization, it will help to create multiple OS and VMs. The problem here is even it takes 5 –10 mins. Here too, we are installing 200+ software are getting installed. We can't wait even for 5 – 10 mins.



Tools used in the background to create VMs:



Containerization – 2013:

Solomon Hykes - who found the Docker

In 2019, Mirantis bought Docker.

How Container is efficient?

Major problem was Time in the earlier stages. In the Container part, we just need to have an image.

From where do we get these iso files to install OS?

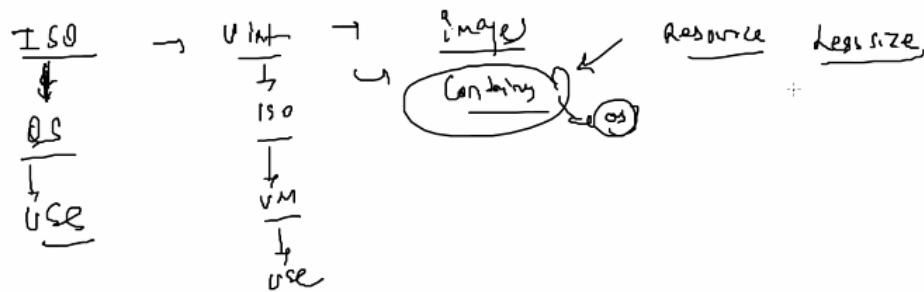
From Pirated website.

Linux is free

Windows OS – you can get it from the website. (eg: getinfopc.com)

In container part, you don't need any iso files. Instead of iso files, you need images. These images are called docker images. Inside this docker image, we don't have anything – no dependency and no software. It has got simple files - libraries and packages, to call them as new operating system. So, the size of the docker image is less.

Image will help us to create containers and use it. Containers is like operating system.



Pendrive – Make it Bootable.

First, install it in the Laptop1 – Folder test

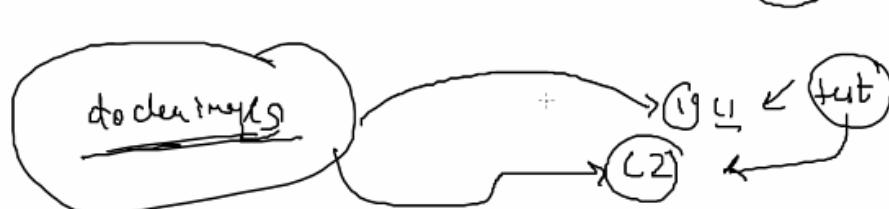
Now use the same pendrive in the Laptop 2 – Will it create the folder test? No, since it is only readable.



Docker image:

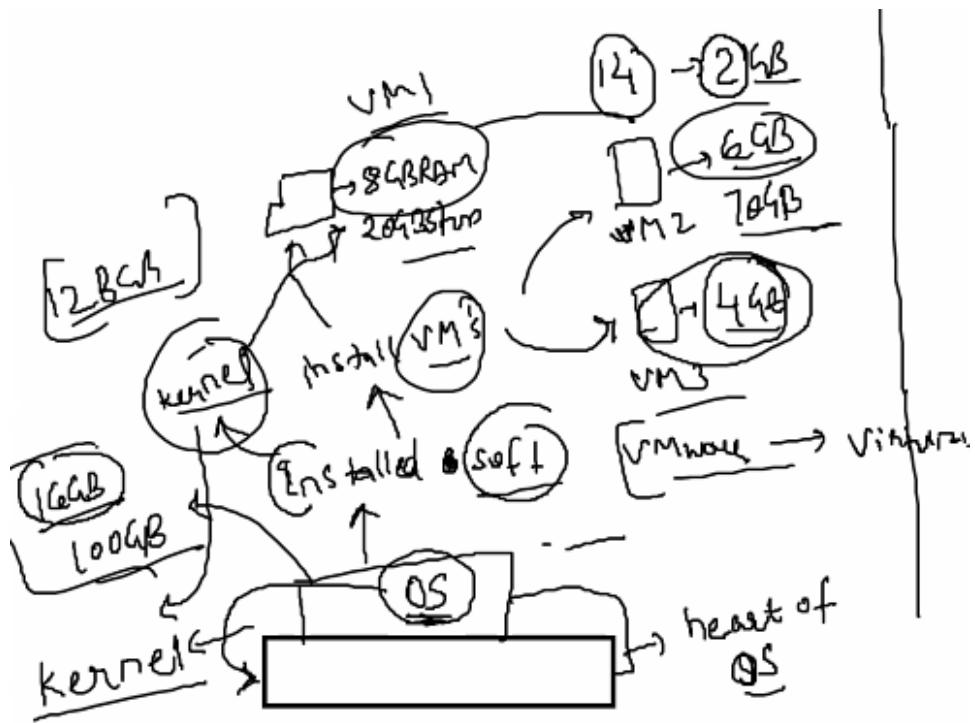
Similar to OS but the difference is it won't have those dependencies.

If I use the same image in another laptop, will I get the test folder in the C2? No



How will containerization help us to overcome the problem faced in virtualization?

Let's try to understand how the virtualization works!



Can we create a 4GB Ram (3rd VM)? We can create it but can't use it parallel. But it can't use more than 2GB.

$$8 + 6 = 14 \text{ GB}$$

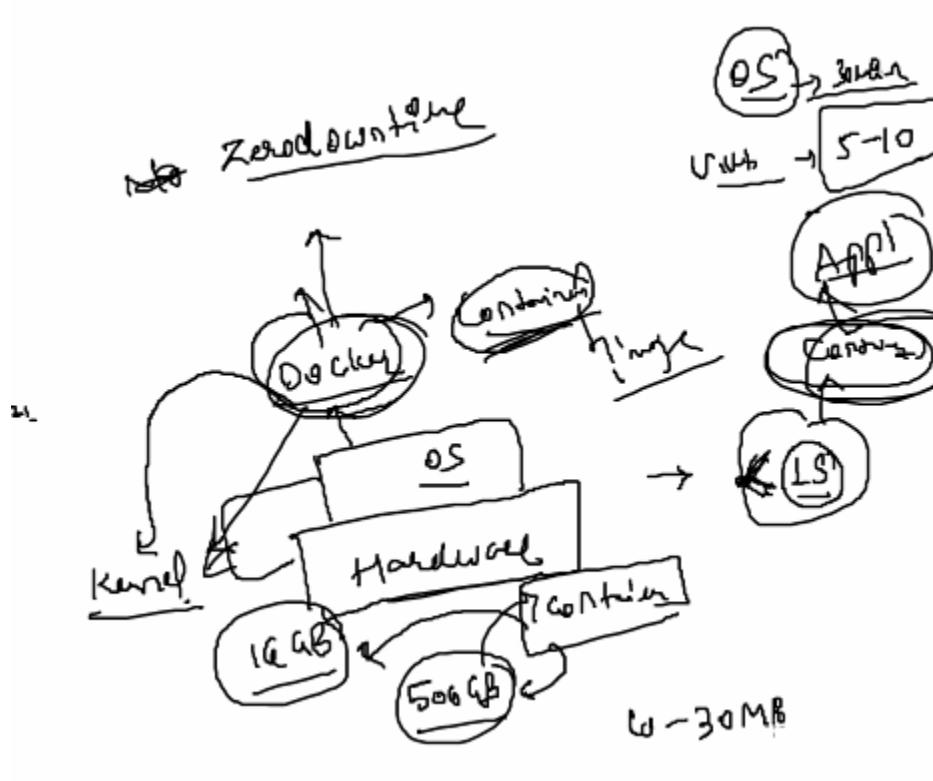
Total Kernel GB is 16. So only 2 is left out. Using this kernel only they are taking the resources.

Can we create 100 VMs? We can create and consume but not at the single time. Because it requires more resources.

Difference:

Area	Virtualization	Containerization
Isolation	Provides complete isolation from the host operating system and the other VMs	Typically provides lightweight isolation from the host and other containers, but doesn't provide as strong a security boundary as a VM
Operating System	Runs a complete operating system including the kernel, thus requiring more system resources such as CPU, memory, and storage	Runs the user-mode portion of an operating system, and can be tailored to contain just the needed services for your app using fewer system resources
Guest compatibility	Runs just about any operating system inside the virtual machine	Runs on the same operating system version as the host
Deployment	Deploy individual VMs by using Hypervisor software	Deploy individual containers by using Docker or deploy multiple containers by using an orchestrator such as Kubernetes
Persistent storage	Use a Virtual Hard Disk (VHD) for local storage for a single VM or a Server Message Block (SMB) file share for storage shared by multiple servers	Use local disks for local storage for a single node or SMB for storage shared by multiple nodes or servers
Load balancing	Virtual machine load balancing is done by running VMs in other servers in a failover cluster	An orchestrator can automatically start or stop containers on cluster nodes to manage changes in load and availability.
Networking	Uses virtual network adapters	Uses an isolated view of a virtual network adapter. Thus, provides a little less virtualization

Containerization:



Here, the image size is less and time to create the image is less.

What is the time taken to create a one single container from image? Less than a second. From this your application can easily run.

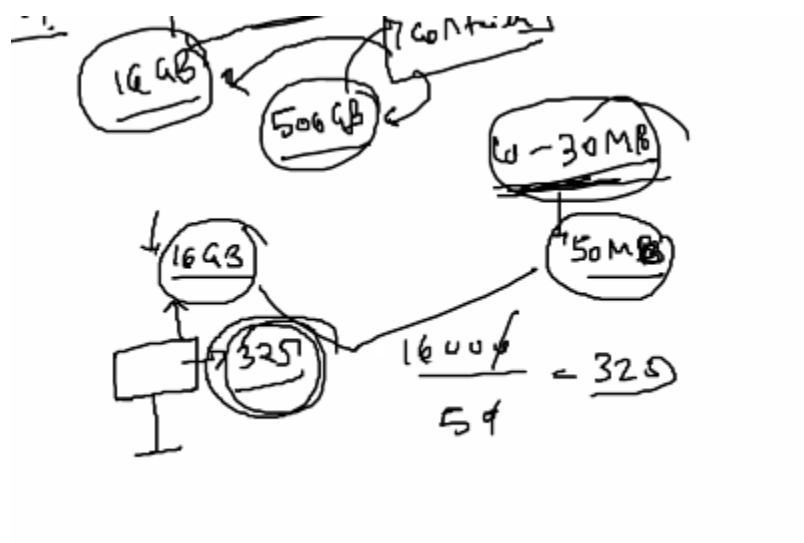
Zero down time - Zero downtime deployment is a deployment method where your website or application is never down or in an unstable state during the deployment process. To achieve this the web server doesn't start serving the changed code until the entire deployment process is complete.

In Docker, they won't create their own kernel. It depends on the base machine RAM (16GB) and Storage (500GB)

By default, the containers will take 10 – 30 MB. This is because only one process runs here. Then depending upon the load, it might increase.

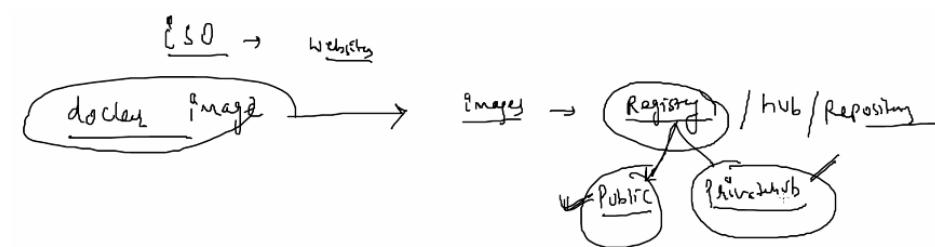
If you have 16GB RAM base machine and 50 MB – docker image, then how many containers we can create?

325 Container we can create. Container is equivalent to website. We can create 325 websites.



Application is only a CLI. Command like interface.

From where you can download docker images?

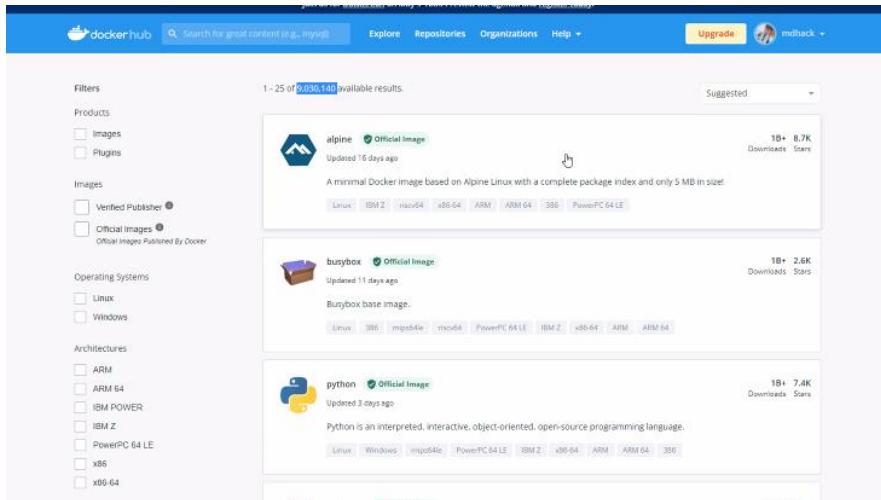


The location where you find the images are called registry /hub/ repository – 2 types -> Public and Private Hub

Private Hub: Internal to company or personnel

Public: Docker Hub

Docker Hub website:

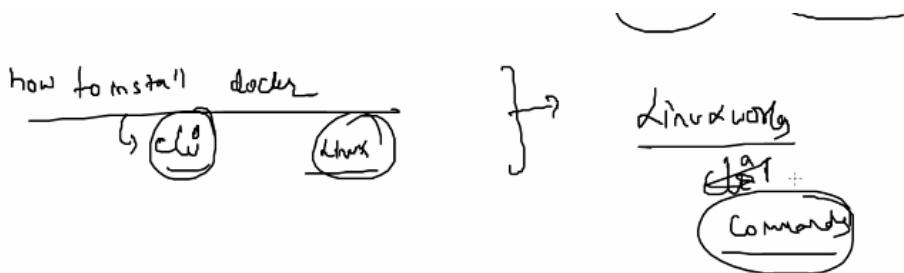


Next Phase we will learn about

How to install docker?

How to use CLI of Docker? (Linux)

Linux – Basic commands – create a file/folder.



Whatever you can do in the OS, you can do it in the containers. We can do whatever we want on the top of the container. The main thing is they only need less resources.

Question: Because of security reasons, we are not able to track the process/logs inside the container?

Start running the cyber security inside the container and then track the process inside the container.

Containers are working on different networks. We need to tell the security team which network we are using it. Then only we can track the process inside the container.

Which type of Linux, I am using – how to check that?

cat /etc/os-release

Linux – Basic commands:

```
https://aws.amazon.com/amazon-linux-2/
[root@vishali ~]# date
Mon Apr 25 06:58:18 UTC 2022
[root@vishali ~]# cal
    April 2022
Su Mo Tu We Th Fr Sa
                1  2
3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

[root@vishali ~]# pwd
/root
[root@vishali ~]# whoami
root
[root@vishali ~]# cat /etc/os-release
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"
HOME_URL="https://amazonlinux.com/"
[root@vishali ~]# █
```

The `pwd` Linux command **prints the current working directory path, starting from the root (`/`)**. Use the `pwd` command to find your way in the Linux file system structure maze or to pass the working directory in a Bash script

```
[root@vishali ~]# cd /
[root@vishali /]# pwd
/
[root@vishali /]# ls
bin  dev  home  lib64  media  opt  root  sbin  sshd_config  tmp  var
boot  etc  lib  local  mnt  proc  run  srv  sys  usr
[root@vishali /]# cd /root/
[root@vishali ~]# cd /
[root@vishali /]# ls
bin  dev  home  lib64  media  opt  root  sbin  sshd_config  tmp  var
boot  etc  lib  local  mnt  proc  run  srv  sys  usr
[root@vishali /]# █
```

What command is used to create an empty file? Touch

What command is used to create a folder? mkdir

```
[root@vishali ~]# cd /
[root@vishali /]# pwd
/
[root@vishali /]# ls
bin dev home lib64 media opt root sbin sshd_config tmp var
boot etc lib local mnt proc run srv sys
[root@vishali /]# cd /root/
[root@vishali ~]# cd /
[root@vishali /]# ls
bin dev home lib64 media opt root sbin sshd_config tmp var
boot etc lib local mnt proc run srv sys
[root@vishali /]# file
Usage: file [-bchikllNnprsvz0] [--apple] [--mime-encoding] [--mime-type]
            [-e testname] [-F separator] [-f namefile] [-m magicfiles] file ...
            file -C [-m magicfiles]
            file [--help]
[root@vishali /]# man file
[root@vishali /]#
[root@vishali /]# touch file1
[root@vishali /]# ls
bin dev file1 lib local mnt proc run srv sys usr
boot etc home lib64 media opt root sbin sshd_config tmp var
[root@vishali /]# cat file1
[root@vishali /]# mkdir vishali
[root@vishali /]# ls
bin dev file1 lib local mnt proc run srv sys usr vishali
boot etc home lib64 media opt root sbin sshd_config tmp var
[root@vishali /]#
```

How to know whether it is a file/folder?

```
[root@vishali /]# ls -l
total 20
lrwxrwxrwx 1 root root 7 Apr 19 16:15 bin -> usr/bin
dr-xr-xr-x 4 root root 4096 Apr 19 16:16 boot
drwxr-xr-x 15 root root 2900 Apr 25 04:52 dev
drwxr-xr-x 80 root root 8192 Apr 25 04:58 etc
-rw-r--r-- 1 root root 0 Apr 25 07:01 file1
drwxr-xr-x 3 root root 22 Apr 25 04:52 home
lrwxrwxrwx 1 root root 7 Apr 19 16:15 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Apr 19 16:15 lib64 -> usr/lib64
drwxr-xr-x 2 root root 6 Apr 19 16:14 local
drwxr-xr-x 2 root root 6 Apr 9 2019 media
drwxr-xr-x 2 root root 6 Apr 9 2019 mnt
drwxr-xr-x 2 root root 6 Apr 25 07:01 modi
drwxr-xr-x 4 root root 27 Apr 19 16:16 opt
dr-xr-xr-x 172 root root 0 Apr 25 04:52 proc
dr-xr-x--- 3 root root 103 Apr 25 04:52 root
drwxr-xr-x 29 root root 980 Apr 25 04:52 run
lrwxrwxrwx 1 root root 8 Apr 19 16:15 sbin -> usr/sbin
drwxr-xr-x 2 root root 6 Apr 9 2019 srv
-rw-r--r-- 1 root root 3958 Apr 25 04:52 sshd_config
dr-xr-xr-x 13 root root 0 Apr 25 04:52 sys
drwxrwxrwt 8 root root 172 Apr 25 06:57 tmp
drwxr-xr-x 13 root root 155 Apr 19 16:15 usr
drwxr-xr-x 19 root root 269 Apr 25 04:52 var
drwxr-xr-x 2 root root 6 Apr 25 07:01 vishali
[root@vishali /]#
```

If it starts with **d – Folder, l – simulink and - - file.**

To remove the empty folder: Use rmdir

```
[root@vishali /]# cd vishali
[root@vishali vishali]# pwd
/vishali
[root@vishali vishali]# ls
[root@vishali vishali]# touch file2
[root@vishali vishali]# pwd
/vishali
[root@vishali vishali]# ls
file2
[root@vishali vishali]# cd
[root@vishali ~]# pwd
/root
[root@vishali ~]# cd /
[root@vishali /]# ls
bin etc lib media opt run sshd_config usr
boot file1 lib64 mnt proc sbin sys var
dev home local modi root srv tmp vishali
[root@vishali /]# rmdir vishali
rmdir: failed to remove 'vishali': Directory not empty
[root@vishali /]#
```

If the folder is not empty, use **rm -rf <foldername>**

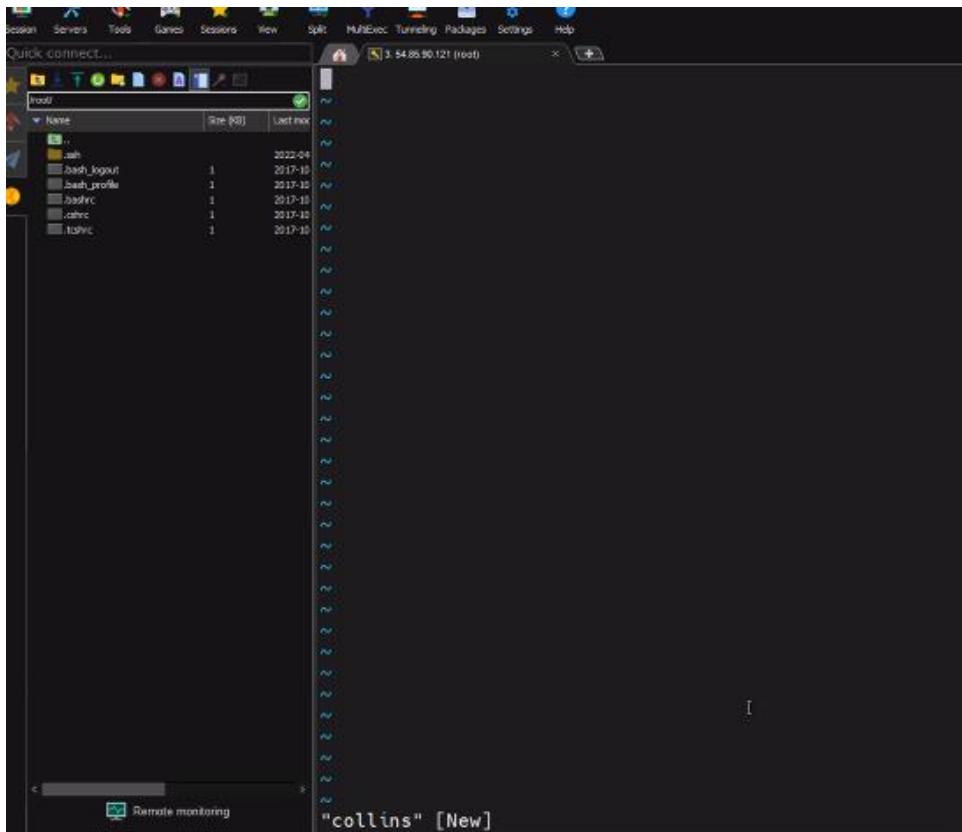
```
[root@vishali /]# rm -f file1
[root@vishali /]# rm -rf vishali
[root@vishali /]# ls
bin dev home lib64 media modi proc run srv sys usr
boot etc lib local mnt opt root sbin sshd_config tmp var
[root@vishali /]#
[root@trainer /]# cp sshd_config var/
[root@trainer /]# ls var/
account cache empty gopher lib lock mail opt run sshd_config yp
adm db games kerberos local log nis preserve spool tmp
[root@trainer /]# rm -f sshd_config
[root@trainer /]# ls
bin boot dev etc home lib lib64 local media mnt opt proc root run sbin srv sys tmp usr var
[root@trainer /]# cp /var/s
pool/ sshd_config
[root@trainer /]# cp /var/sshd_config .
[root@trainer /]# ls
bin dev home lib64 media opt root sbin sshd_config tmp var
boot etc lib local mnt proc run srv sys usr
[root@trainer /]#
```

To remove or move a file:

```
bin dev home lib lib64 local media mnt opt proc root run sbin srv sys tmp var
[root@trainer /]# mv sshd_config mayank
[root@trainer /]# ls
bin dev home lib64 mayank mnt proc run srv tmp var
boot etc lib local media opt root sbin sys usr
[root@trainer /]# mv mayank /var/
[root@trainer /]# ls
bin boot dev etc home lib lib64 local media mnt opt proc root run sbin srv sys tmp usr var
[root@trainer /]# cp /var/
account cache empty gopher lib lock mail nis preserve spool tmp
adm db games kerberos local log mayank opt run sshd_config yp
[root@trainer /]#
```

To create a file with “collins”:

```
[root@trainer ~]# vim collins
```



To insert mode into vim: Press "i":

Save or without save:

Save:

Press Escape to come out of Insert Mode. Type :wq to save the file. W – write and q is for quit.

Without Save:

Press Escape to come out of Insert Mode. Type :q! to not to save the file

```
[root@trainer ~]# vim collins
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]# ls
collins
[root@trainer ~]# cat collins
Hello From VIM Learning
[root@trainer ~]# vim collins
[root@trainer ~]# cat collins
Hello From VIM Learning
[root@trainer ~]# █
```

Google Drive shared from Trainer:

<https://drive.google.com/drive/folders/1iKkFYOxqVNsjOv0dj87mn4rETW2srU6L?usp=sharing>

Linux Commands:

```
1 ls
2 pwd
3 ls
4 cat /etc/os-release
5 date
6 cal
7 pwd
8 whoami
9 cd /
10 pwd
11 l
12 ls
13 pwd
14 cd /root/
15 pwd
16 cd /
17 ls
18 file
19 man file
20 touch file1
21 ls
22 cat file1
23 mkdir modi
24 ls
25 ls --color=none
26 ls -l
27 ls
28 cd modi
29 pwd
30 ls
```

```
31 touch fil2
32 pwd
33 ls
34 cd
35 pwd
36 cd /
37 ls
38 rmdir modi
39 rm -rf modi/
40 ls
41 rm -f file1
42 ls
43 pwd
44 ls
45 cp sshd_config var/
46 ls var/
47 rm -f sshd_config
48 ls
49 cp /var/sshd_config .
50 ls
51 mv sshd_config mayank
52 ls
53 mv mayank /var/
54 ls
55 ls /var/
56 cd
57 vim collins
58 ls
59 cat collins
60 vim collins
```

61 cat collins

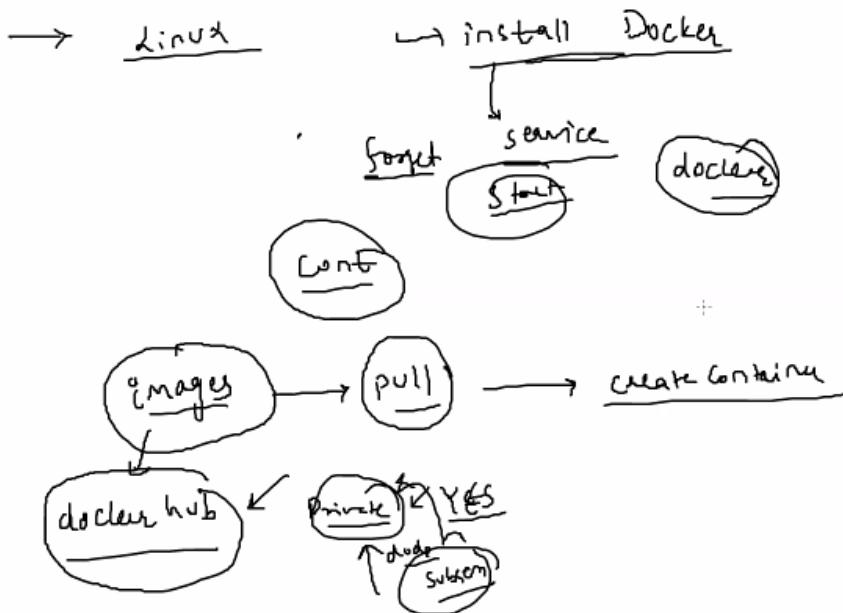
62 history

TASK1:

TRY TO PRACTICE ALL these commands – 15 mins

Docker:

How to install Docker in Linux?



To know whether docker is installed or not:

```
https://aws.amazon.com/amazon-linux-2/
[root@vishali ~]# docker
-bash: docker: command not found
[root@vishali ~]# rpm -q docker
package docker is not installed
[root@vishali ~]#
```

Press double tab to know commands starting with “do”:

```
[root@vishali ~]# do
do                      dockerd          done
docker                  docker-init      dosfsck
docker-containerd       docker-proxy     dosfslabel
docker-containerd-shim  docker-runc
docker-ctr              domainname
[root@vishali ~]# do
```

Docker commands starting with docker: (Press double tab)

```
[root@vishali ~]# docker          .
docker              docker-ctr          docker-proxy
docker-containerd   dockerd            docker-runc
docker-containerd-shim docker-init
```

Install the bas-completion:

```
[root@vishali ~]# yum install bash-completion -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package 1:bash-completion-2.1-6.amzn2.noarch already installed and latest version
Nothing to do
[root@vishali ~]#
```

To know the list of docker images available internally:

```
docker.io/library/hello-world:latest
[root@trainer ~]# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
hello-world     latest    feb5d9fea6a5  7 months ago  13.3kB
centos          latest    5d0da3dc9764  7 months ago  231MB
[root@trainer ~]#
```

Create a hello-world container:

```
[root@trainer ~]# docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

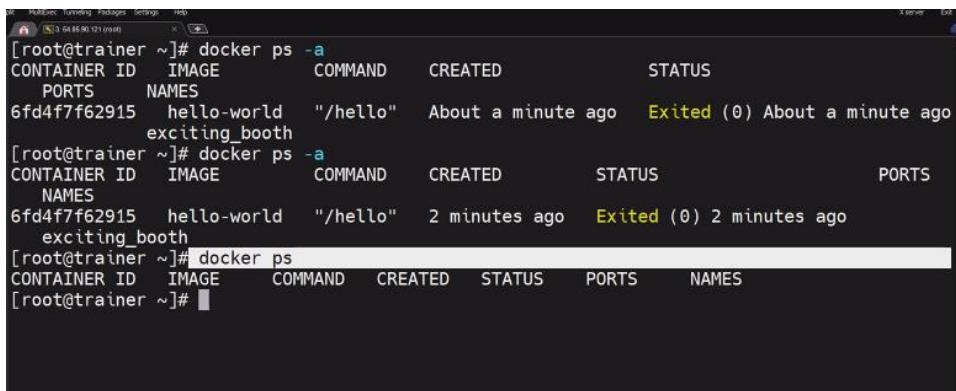
To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent
it
to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
[root@trainer ~]#
```

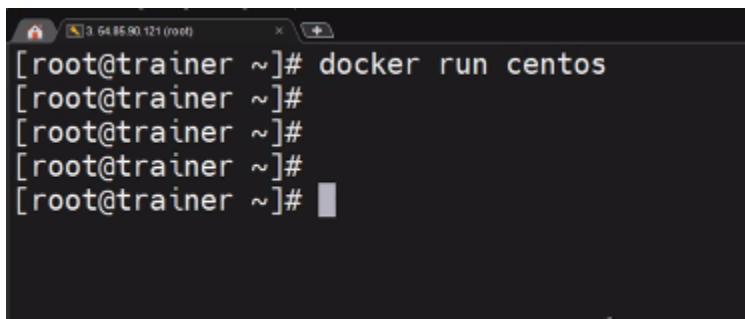
To know the list of containers:



```
[root@trainer ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
6fd4f7f62915 hello-world "/hello" About a minute ago Exited (0) About a minute ago
exciting_booth
[root@trainer ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PORTS NAMES
6fd4f7f62915 hello-world "/hello" 2 minutes ago Exited (0) 2 minutes ago
exciting_booth
[root@trainer ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@trainer ~]#
```

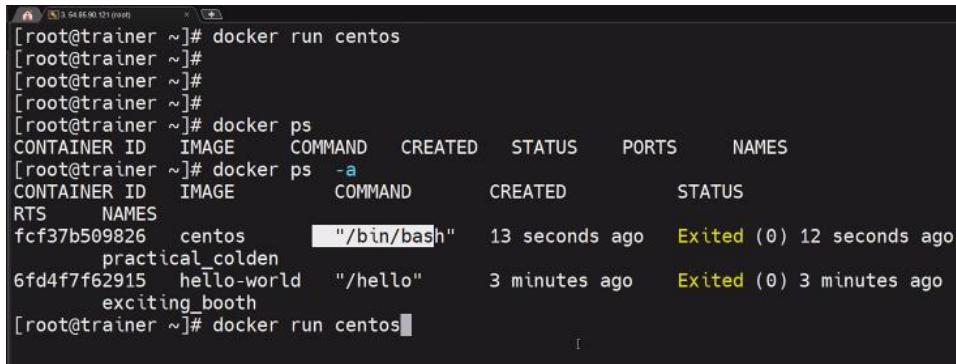
A container should have a process inside it to run all the time. If not, it will be exited.

Centos needs a terminal should work.



```
[root@trainer ~]# docker run centos
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]#
```

Create a centos container:



```
[root@trainer ~]# docker run centos
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@trainer ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
fcf37b509826 centos "/bin/bash" 13 seconds ago Exited (0) 12 seconds ago
practical_colden
6fd4f7f62915 hello-world "/hello" 3 minutes ago Exited (0) 3 minutes ago
exciting_booth
[root@trainer ~]# docker run centos
```

```
[root@trainer ~]# docker run centos
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@trainer ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
RTS NAMES
fcb37b509826 centos "/bin/bash" 13 seconds ago Exited (0) 12 seconds ago
practical_colden
6fd4f7f62915 hello-world "/hello" 3 minutes ago Exited (0) 3 minutes ago
exciting_booth
[root@trainer ~]# docker run --name mayank centos
[root@trainer ~]#
```

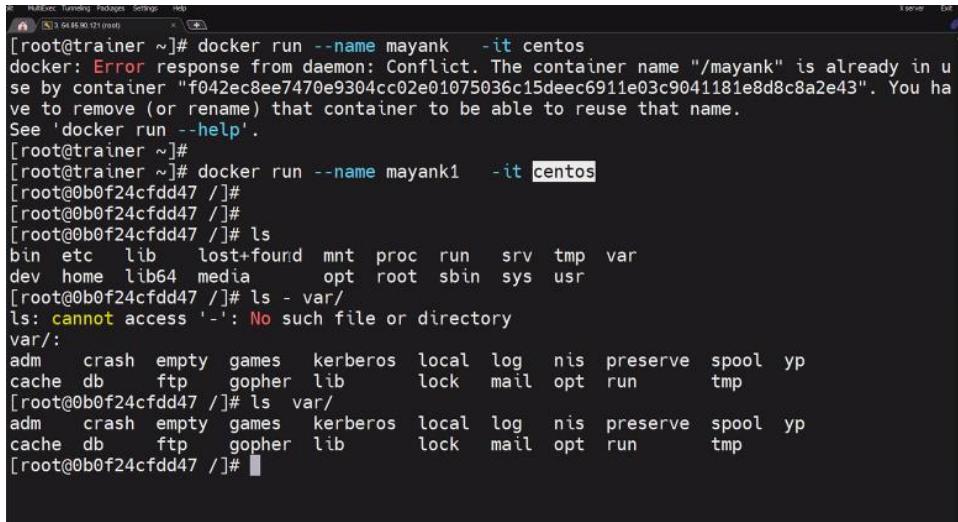
To provide name to the container:

```
exciting_booth
[root@trainer ~]# docker run --name mayank centos
[root@trainer ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@trainer ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
f042ec8ee747 centos "/bin/bash" 8 seconds ago Exited (0) 7 seconds ago
mayank
fcf37b509826 centos "/bin/bash" About a minute ago Exited (0) About a minute
ago
practical_colden
6fd4f7f62915 hello-world "/hello" 4 minutes ago Exited (0) 4 minutes ago
exciting_booth
[root@trainer ~]#
```

To create a container in the interactive mode:

```
[root@trainer ~]# docker run --name mayank -it centos
docker: Error response from daemon: Conflict. The container name "/mayank" is already in use by container "f042ec8ee7470e9304cc02e01075036c15deec6911e03c9041181e8d8c8a2e43". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@trainer ~]#
[root@trainer ~]# docker run --name mayank1 -it centos
[root@0b0f24cfdd47 ~]#
[root@0b0f24cfdd47 ~]#
[root@0b0f24cfdd47 ~]#
```

Container has its own ip, hostname:



```
[root@trainer ~]# docker run --name mayank -it centos
docker: Error response from daemon: Conflict. The container name "/mayank" is already in use by container "f042ec8ee7470e9304cc02e01075036c15deec6911e03c9041181e8d8c8a2e43". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@trainer ~]#
[root@trainer ~]# docker run --name mayank1 -it centos
[root@0b0f24cfdd47 ~]#
[root@0b0f24cfdd47 ~]# ls
bin etc lib lost+found mnt proc run srv tmp var
dev home lib64 media opt root sbin sys usr
[root@0b0f24cfdd47 ~]# ls - var/
ls: cannot access '-': No such file or directory
var/:
adm crash empty games kerberos local log nis preserve spool yp
cache db ftp gopher lib lock mail opt run tmp
[root@0b0f24cfdd47 ~]# ls var/
adm crash empty games kerberos local log nis preserve spool yp
cache db ftp gopher lib lock mail opt run tmp
[root@0b0f24cfdd47 ~]#
```

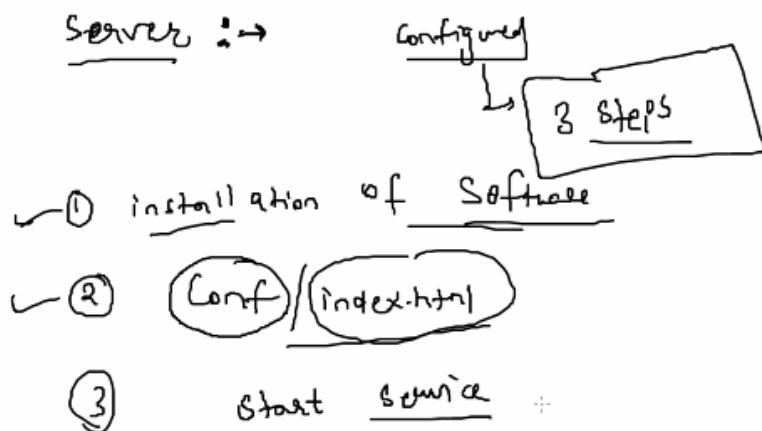
The container going to stop when we exit:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
0
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
10: eth0@if11: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP gro
ault
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
[root@0b0f24cfdd47 ~]# hostname
0b0f24cfdd47
[root@0b0f24cfdd47 ~]# exit
exit
[root@trainer ~]# docker ps -a
CONTAINER ID        IMAGE       COMMAND      CREATED          STATUS
PORTS      NAMES
0b0f24cfdd47        centos     "/bin/bash"   About a minute ago   Exited (0) 3 seconds
        mayank1
f042ec8ee747        centos     "/bin/bash"   2 minutes ago    Exited (0) 2 minutes
        mayank
fcf37b509826        centos     "/bin/bash"   4 minutes ago    Exited (0) 4 minutes
        practical_colden
6fd4f7f62015        hello-world "/hello"     7 minutes ago    Exited (0) 7 minutes
```

To create an Apache web server, how much time will it take to create? 7 – 8 secs

How to create a server?

We need 3 steps



How can you guys access the web server? If someone creates a server

Using IP Address – it should be public.

Install the apache server:

> yum install -y httpd

To check whether it is installed or not:

```
[root@trainer ~]# rpm -q httpd
httpd-2.4.52-1.amzn2.x86_64
[root@trainer ~]#
```

```
[root@trainer ~]# cd /var/www/html/
[root@trainer html]# pwd
/var/www/html
[root@trainer html]# vim index.html
[root@trainer html]#
```

A screenshot of a terminal window titled "3: 54.85.90.121 (root)". The window displays the content of the Apache web server's index page, which includes an

tag with the text "Hello From Apache Webserver In DevSecOps Training". Below the title, there is a large amount of repeated placeholder text (~). The terminal interface includes a menu bar with "Split", "MultiExec", "Tunneling", "Packages", "Settings", and "Help". At the bottom, there is a status bar showing system information like memory usage (0.22 GB / 1.92 GB), network speed (0.01Mbps), and disk activity (0.00 Mb/s). A watermark at the bottom left reads "describing to the professional edition here: https://inbaxterm.mubatik.net".

A screenshot of a terminal window titled "3: 54.85.90.121 (root)". The session shows the following commands and output:

```
[root@trainer ~]# cd /var/www/html/
[root@trainer html]# pwd
/var/www/html
[root@trainer html]# vim index.html
[root@trainer html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service)
  Active: inactive (dead)
    Docs: man:httpd.service(8)
[root@trainer html]# systemctl start httpd
[root@trainer html]#
```

Not secure | 54.85.90.121

Hello From Apache Webserver In DevSecOps Training

```
yum install -y httpd  
rpm -q httpd  
cd /var/www/html/  
pwd  
vim index.html  
systemctl status httpd  
systemctl start httpd  
history
```

Same thing we can do in the container also.

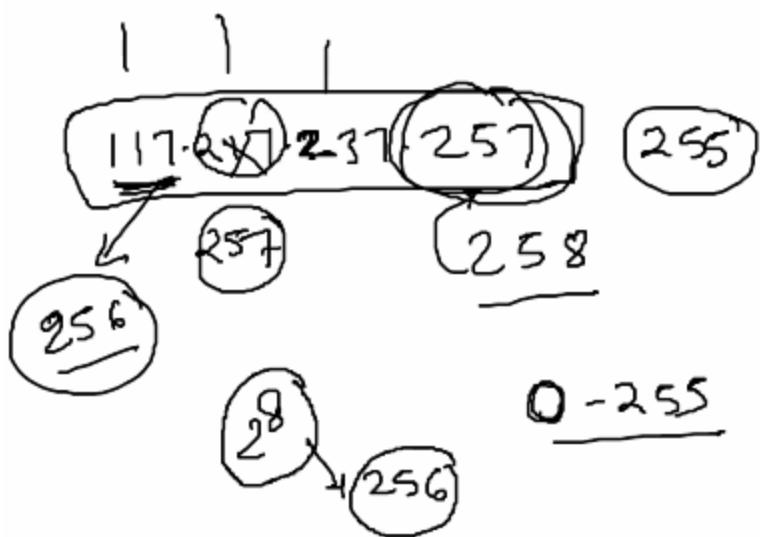
Use centos:

```
[root@trainer html]#  
[root@trainer html]# docker run --name webserver -it centos  
[root@trainer html]# docker run --name Webserver -it centos  
[root@3d3fd9b51510 /]#
```

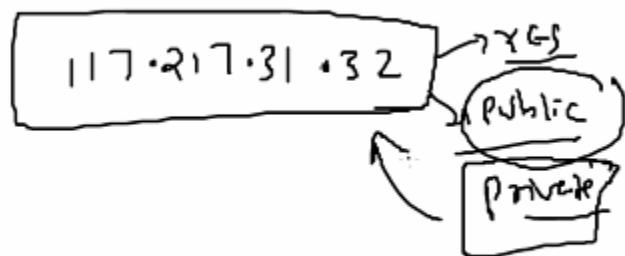
Now, How can you use the docker web server in other machines? IP address

```
[root@trainer html]# docker run --name webserver -it centos  
[root@3d3fd9b51510 /]#  
[root@3d3fd9b51510 /]#  
[root@3d3fd9b51510 /]# exit  
[root@trainer html]# docker run --name webserver1 -it centos  
  
[root@df5ab63ec788 /]#  
[root@df5ab63ec788 /]#  
[root@df5ab63ec788 /]#  
[root@df5ab63ec788 /]# ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 100  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
14: eth0@if15: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default  
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0  
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0  
        valid_lft forever preferred_lft forever  
[root@df5ab63ec788 /]#
```

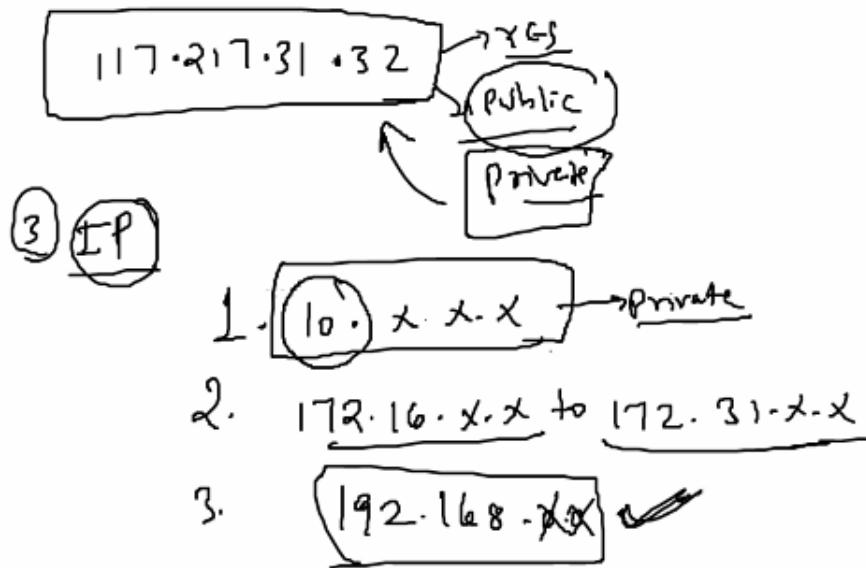
IP Address range:



How to identify whether the IP address is public or private?



There is a range for private address



Apart from this all are public ip address.

Docker IP address start with 172 -> which means it is private ip address.

```
[root@df5ab63ec788 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 100
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
14: eth0@if15: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
[root@df5ab63ec788 ~]#
```

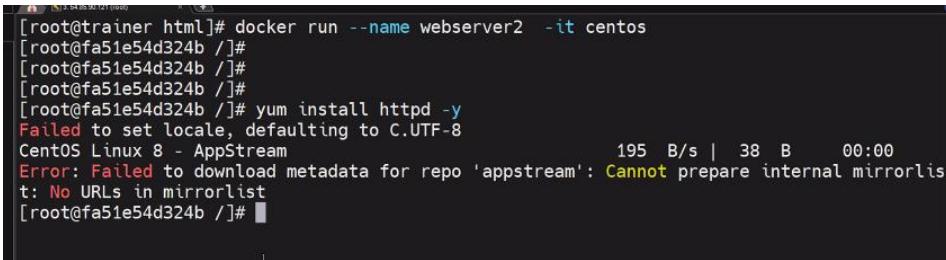
How to check whether the IP is working or not?

By using curl command

It shows the content of the website

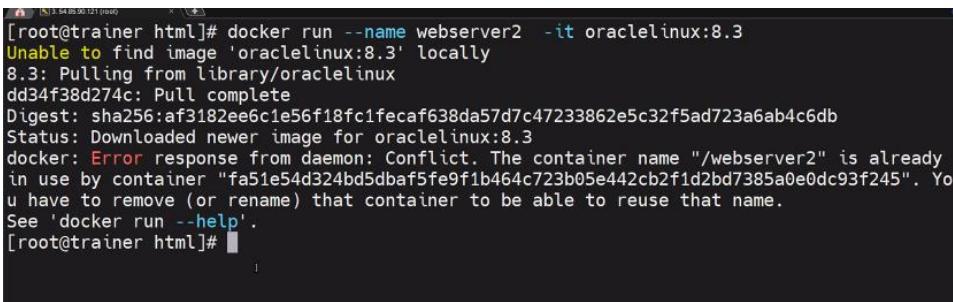
```
[root@trainer html]# curl 54.85.90.121
<h1>Hello From Apache Webserver In DevSecOps Training</h1>
[root@trainer html]# curl google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
[root@trainer html]# curl https://google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="https://www.google.com/">here</A>.
</BODY></HTML>
[root@trainer html]# curl https://google.co.in
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="https://www.google.co.in/">here</A>.
</BODY></HTML>
[root@trainer html]#
```

Centos – can't be used. Problem is going on for past 4 months



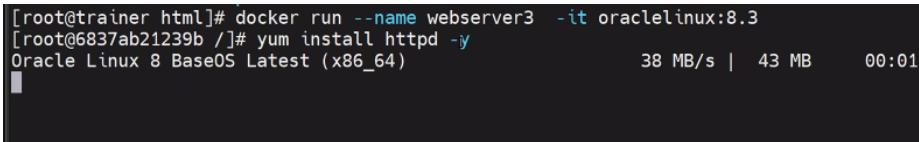
```
[root@trainer html]# docker run --name webserver2 -it centos
[root@fa51e54d324b /]#
[root@fa51e54d324b /]#
[root@fa51e54d324b /]#
[root@fa51e54d324b /]# yum install httpd -y
Failed to set locale, defaulting to C.UTF-8
CentOS Linux 8 - AppStream
Error: Failed to download metadata for repo 'appstream': Cannot prepare internal mirrorlist
t: No URLs in mirrorlist
[root@fa51e54d324b /]#
```

You can use oraclelinux:8.3 instead. No difference between both. Both are same.



```
[root@trainer html]# docker run --name webserver2 -it oraclelinux:8.3
Unable to find image 'oraclelinux:8.3' locally
8.3: Pulling from library/oraclelinux
dd34f38d274c: Pull complete
Digest: sha256:af3182ee6c1e56f18fc1fecaf638da57d7c47233862e5c32f5ad723a6ab4c6db
Status: Downloaded newer image for oraclelinux:8.3
docker: Error response from daemon: Conflict. The container name "/webserver2" is already in use by container "fa51e54d324bd5dbaf5fe9f1b464c723b05e442cb2f1d2bd7385a0e0dc93f245". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@trainer html]#
```

Problem above is the name of the container – already exist!



```
[root@trainer html]# docker run --name webserver3 -it oraclelinux:8.3
[root@6837ab21239b /]# yum install httpd -y
Oracle Linux 8 BaseOS Latest (x86_64)          38 MB/s | 43 MB      00:01
```

Configuration step:

```
[root@6837ab21239b /]# cd /var/www/html/
[root@6837ab21239b html]# vim
bash: vim: command not found
[root@6837ab21239b html]#
```

Echo command:

```
[root@6837ab21239b html]# echo MAYANK
MAYANK
[root@6837ab21239b html]# echo MAYANK modi
MAYANK modi
[root@6837ab21239b html]# echo MAYANK modi > index.html
```

Redirecting the content to index.html

Problem in docker is systemctl won't work.

```
[root@6837ab21239b html]# systemctl
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
[root@6837ab21239b html]#
```

Will container allow you to run the system-d process by default? NO! Since it's big in size.

You cannot connect to IP address

```
[root@6837ab21239b html]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 100
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 00:00:00:00:00:00 scope host lo
        valid_lft forever preferred_lft forever
18: eth0@if19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
[root@6837ab21239b html]# curl 172.17.0.2
bash: curl: command not found
[root@6837ab21239b html]# curl 172.17.0.2
curl: (7) Failed to connect to 172.17.0.2 port 80: Connection refused
[root@6837ab21239b html]#
```

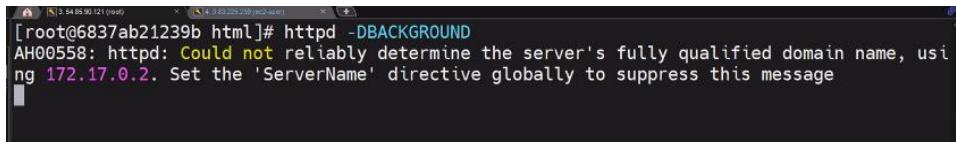
Since systemctl won't work. Use the below command.

Difference between foreground and background?

Foreground command:

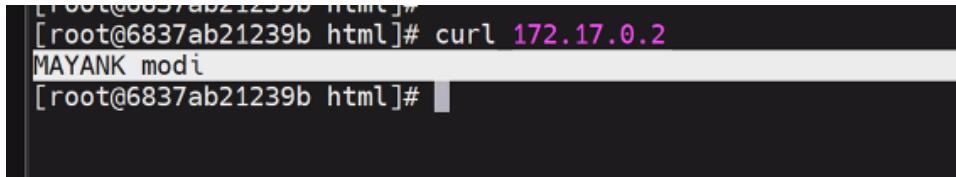
```
curl: (7) Failed to connect to 172.17.0.2 port 80: Connection refused
[root@6837ab21239b html]# httpd -DFOREGROUND
```

Using Dbackground you can!



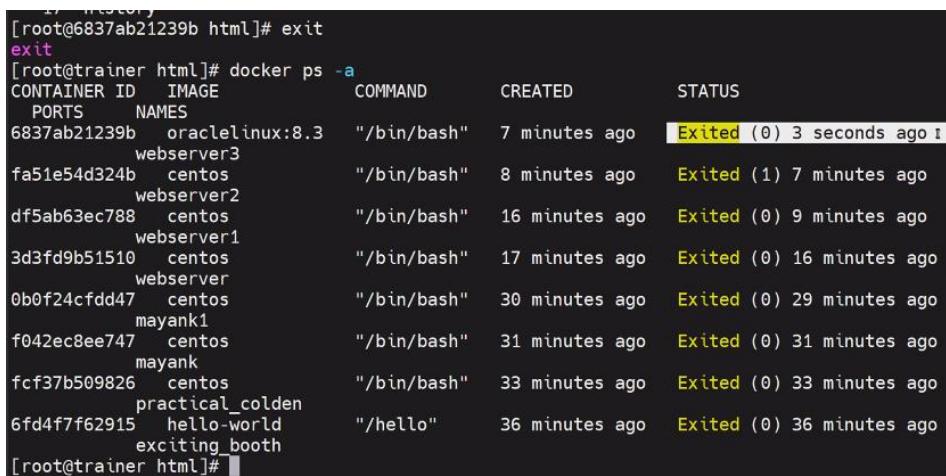
```
[root@6837ab21239b html]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
```

You can see the website is created and running inside container:



```
[root@6837ab21239b html]# curl 172.17.0.2
MAYANK modi
[root@6837ab21239b html]#
```

Exit command: Container will exit.



```
[root@6837ab21239b html]# exit
exit
[root@trainer html]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS      NAMES
6837ab21239b        oraclelinux:8.3   "/bin/bash"        7 minutes ago     Exited (0) 3 seconds ago
                                                webserver3
fa51e54d324b        centos             "/bin/bash"        8 minutes ago     Exited (1) 7 minutes ago
                                                webserver2
df5ab63ec788        centos             "/bin/bash"        16 minutes ago    Exited (0) 9 minutes ago
                                                webserver1
3d3fd9b51510        centos             "/bin/bash"        17 minutes ago    Exited (0) 16 minutes ago
                                                webserver
0b0f24cfdd47        centos             "/bin/bash"        30 minutes ago    Exited (0) 29 minutes ago
                                                mayank1
f042ec8ee747        centos             "/bin/bash"        31 minutes ago    Exited (0) 31 minutes ago
                                                mayank
fcf37b509826        centos             "/bin/bash"        33 minutes ago    Exited (0) 33 minutes ago
                                                practical_colden
6fd4f7f62915        hello-world        "/hello"           36 minutes ago    Exited (0) 36 minutes ago
                                                exciting_booth
[root@trainer html]#
```

DOCKER CONTAINERS

- 1 hostname
- 2 l
- 3 ls
- 4 cat /etc/os-release
- 5 s
- 6 free -m
- 7 <<X
- 8 ls
- 9 hostname
- 10 hostnamectl set-hostname Trainer

```
11 bash
Yum install docker -y
Systemctl start docker
curl https://raw.githubusercontent.com/docker/docker-ce/master/components/cli/contrib/completion/bash/docker -o /etc/bash_completion.d/docker.sh
bash
12 docker --help
13 docker images
14 docker search centos
15 docker images
16 docker pull centos
17 docker images
18 docker search hello-world
19 docker pull hello-world
20 docker images
21 docker container --help
22 docker run hello-world
23 docker images
24 docker ps -a
25 docker ps
26 docker ps -a
27 docker images
28 docker run centos
29 docker ps
30 docker ps -a
31 docker run --name mayank centos
32 docker ps
33 docker ps -a
34 docker run --name mayank -it centos
35 docker run --name mayank1 -it centos
```

```
36 docker ps -a
37 <<X
38 yum install -y httpd
39 rpm -q httpd
40 cd /var/www/html/
41 pwd
42 vim index.html
43 systemctl status httpd
44 systemctl start httpd
45 history
46 docker run --name webserver -it centos
47 docker run --name webserver1 -it centos
48 curl 54.85.90.121
curl 44.202.147.151
49 curl google.com
50 curl https://google.com
51 curl https://google.co.in
52*
53 docker run --name webserver2 -it centos
54 docker run --name webserver2 -it oraclelinux:8.3
55 docker run --name webserver3 -it oraclelinux:8.3
56 docker ps -a
57 history
```

Inside Container for configuring Apache

```
1 yum install httpd -y
2 cd /var/www/html/
3 vim
4 echo MAYANK
5 echo MAYANK modi
```

```
6 echo MAYANK modi > index.html
7 cat index.html
8 systemctl
9 ip
10 ip a
11 curl 172.17.0.2
12 curl 172.17.0.2
13 httpd -DFOREGROUND
14 httpd -DBACKGROUND
15 curl 172.17.0.2
16 HIS
17 history
```

TASKS: 2

1. Try to install docker software in ur own labs

```
[root@vishali ~]# yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package docker-20.10.7-5.amzn2.x86_64 already installed and latest version
Nothing to do
[root@vishali ~]#
```

2. Try to find out a command to download the docker image and download a image called httpd (— use every command that i have shown)

Bash-completion command: update the changes done to docker

```
[root@vishali html]# curl https://raw.githubusercontent.com/docker/docker-ce/master/components/cli/contrib/completion/bash/docker -o /etc/bash_completion.d/docker.sh
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent   Left  Speed
100  114k  100  114k    0     0  4156k      0  --:--:--  --:--:-- 4232k
[root@vishali html]# bash
```

```
yum install httpd -y
```

```
Installed:  
httpd.x86_64 0:2.4.52-1.amzn2  
  
Dependency Installed:  
apr.x86_64 0:1.7.0-9.amzn2  
apr-util.x86_64 0:1.6.1-5.amzn2.0.2  
apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2  
generic-logos-htpd.noarch 0:18.0.0-4.amzn2  
httpd-filesystem.noarch 0:2.4.52-1.amzn2  
httpd-tools.x86_64 0:2.4.52-1.amzn2  
mailcap.noarch 0:2.1.41-2.amzn2  
mod_http2.x86_64 0:1.15.19-1.amzn2.0.1  
  
Complete!  
[root@vishali ~]#
```

Q.3 Try to download images from docker hub

ubuntu

oraclelinux:8.3

Centos

Hello-world

mdhack/myserver

mdhack/myapache

```
[root@vishali html]# docker images  
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE  
ubuntu          latest   3f4714ee068a  3 days ago   77.8MB  
hello-world     latest   feb5d9fea6a5  7 months ago  13.3kB  
centos          latest   5d0da3dc9764  7 months ago  231MB  
oraclelinux      8.3     816d99f0bbe8  12 months ago 224MB  
mdhack/myapache  latest   ea36c5376ba4  20 months ago 166MB  
mdhack/myserver  latest   e8ce7504414a  20 months ago 350MB  
[root@vishali html]#
```

Q.4 create a container using every image and u should need to give a container name to every container.

docker run --name container1 ubuntu

docker run --name container2 oraclelinux:8.3

docker run --name container3 centos

docker run --name container4 hello-world

docker run --name container5 mdhack/myserver

docker run --name container6 mdhack/myapache

```
[root@vishali ~]# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS         PORTS          NAMES
b9fdfcd79dfa  mdhack/myapache " /bin/sh -c 'httpd -...'"   About a minute ago
p About a minute ago           80/tcp        container6
0c4a7204cc08  mdhack/myserver " /bin/sh -c 'httpd -...'"   2 minutes ago
xited (0) 2 minutes ago      container5
cb6d11eaf90c  hello-world    " /hello"                2 minutes ago
xited (0) 2 minutes ago      container4
766b155a5cf6  centos         " /bin/bash"              3 minutes ago
xited (0) 3 minutes ago      container3
287515c2c28d  oraclelinux:8.3 " /bin/bash"              3 minutes ago
xited (0) 3 minutes ago      container2
d20d02123f83  ubuntu          " bash"                  5 minutes ago
xited (0) 5 minutes ago      container1
[root@vishali ~]#
```

Q.5 Create and configure apache server into in ur own lab and share me the ip address so that i can also verify it. Try to create interesting index.html file content AND ADDITIONALLY CREATE 2 MORE HTML FILES WITH NAME TRAINER.HTML And training.html inside tat files are write about the training

```
yum install -y httpd
```

```
rpm -q httpd
```

```
cd /var/www/html/
```

```
pwd
```

```
vim index.html
```

```
[root@vishali html]# vim index.html
[root@vishali html]# vim trainer.html
[root@vishali html]# cat trainer.html
<h1> Training is going good </h1>
[root@vishali html]# cat index.html
<h1> Hello! This is the content inside the index.html file</h1>
```

```
systemctl status httpd
```

```
systemctl start httpd
```

```
[root@vishali ~]# cd /var/www/html/
[root@vishali html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor
t: disabled)
     Active: active (running) since Mon 2022-04-25 09:58:31 UTC; 17min ago
       Docs: man:httpd.service(8)
   Main PID: 2478 (httpd)
     Status: "Total requests: 1; Idle/Busy workers 100/0;Requests/sec: 0.000
bytes served/sec: 0 B/sec"
     Tasks: 47
    Memory: 9.7M
      CGroup: /system.slice/httpd.service
              ├─2478 /usr/sbin/httpd -DFOREGROUND
              ├─2479 /usr/sbin/httpd -DFOREGROUND
              ├─2480 /usr/sbin/httpd -DFOREGROUND
              ├─2481 /usr/sbin/httpd -DFOREGROUND
              ├─2482 /usr/sbin/httpd -DFOREGROUND
              └─2483 /usr/sbin/httpd -DFOREGROUND

Apr 25 09:58:31 vishali systemd[1]: Starting The Apache HTTP Server...
Apr 25 09:58:31 vishali httpd[2478]: AH00558: httpd: Could not reliably de
Apr 25 09:58:31 vishali systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@vishali html]# systemctl start httpd
[root@vishali html]#
```

Q.6 Create a container using oraclelinux:8.3 image and try to configure apache into it

Create the container:

```
[root@vishali ~]# docker run --name webserver3 -it oraclelinux:8.3
[root@2f7adf04887a /]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qdisc mq 0: lo
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
28: eth0@if29: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
group default
    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
[root@2f7adf04887a /]# ^C
[root@2f7adf04887a /]#
```

curl 44.202.147.151

```
[root@vishali html]# systemctl start httpd
[root@vishali html]# httpd -DFOREGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::10ba:16ff:fe63:93db%eth0. Set the 'ServerName' directive globally to suppress this message
httpd (pid 2478) already running
[root@vishali html]# curl 172.17.0.3/
curl: (7) Failed to connect to 172.17.0.3 port 80 after 0 ms: Connection refused
[root@vishali html]# curl 172.17.0.2/
<body bgcolor='blue'>
<h1>Welcome</h1>
<h2>Hello From MDhack/Mayank Server</h2>
<h3>Apache Server</h3>

[root@vishali html]# curl 172.17.0.1/
<h1> Hello! This is the content inside the index.html file</h1>
[root@vishali html]# ^C
[root@vishali html]#
```

```
[root@vishali html]# curl 172.17.0.1/
<h1> Hello! This is the content inside the index.html file</h1>
[root@vishali html]# ^C
[root@vishali html]# curl 44.202.147.151
<h1> Hello! This is the content inside the index.html file</h1>
[root@vishali html]#
```

<http://44.202.147.151/>



Hello! This is the content inside the index.html file

<http://44.202.147.151/trainer.html>



Training is going good

Now, we will see how to start the existing container.

I want to start the existing container

```
[root@trainer ~]# docker start webserver3
```

To list the container

More details about the container: inspect command

```
webserver3  
[root@trainer ~]# docker inspect webserver3
```

To get the ip address:

```
[root@trainer ~]# docker inspect webserver3 | grep -i ip
    "IpcMode": "private",
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
        "IPAMConfig": null,
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
[root@trainer ~]# █

```

0% 0.32 GB / 1.92 GB 0.01 MB/s 0.00 MB/s 6 hours root 0.13%

```
GlobalIPv6PrefixLen": 0,
[root@trainer ~]# docker inspect webserver3 | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.2",
    "IPAddress": "172.17.0.2",
[root@trainer ~]# █
```

Start the exiting the container:

```
"IP Address": "172.17.0.2",
[root@trainer ~]# curl 172.17.0.2
curl: (7) Failed to connect to 172.17.0.2 port 80 after 0 ms: Connection refused
[root@trainer ~]#
```

Failed to connect

Manually start the service

Attach command: Not useful right now - attach /bin/bash processes

```
trainer
[root@trainer ~]# docker attach webserver3
[root@6837ab21239b /]#
```

```
[root@trainer ~]# docker attach webserver3
[root@6837ab21239b /]#
[root@6837ab21239b /]#
[root@6837ab21239b /]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[root@6837ab21239b /]#
```

```
[root@trainer ~]# docker attach webserver3
[root@6837ab21239b /]#
[root@6837ab21239b /]#
[root@6837ab21239b /]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[root@6837ab21239b /]# curl 172.17.0.2
MAYANK modi
[root@6837ab21239b /]#
```

Execute command: Useful to run any command inside the container.

```
[root@trainer ~]# docker start webserver3
webserver3
[root@trainer ~]#
[root@trainer ~]#
[root@trainer ~]# docker ps
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS          PORTS
 NAMES
6837ab21239b        oraclelinux:8.3   "/bin/bash"    About an hour ago   Up 1 second
webserver3
[root@trainer ~]# docker exec -it webserver3 bash
```

```
[root@trainer ~]# docker exec -it webserver3 bash
[root@6837ab21239b ~]#
[root@6837ab21239b ~]#
[root@6837ab21239b ~]#
[root@6837ab21239b ~]# curl 172.17.0.2
curl: (7) Failed to connect to 172.17.0.2 port 80: Connection refused
[root@6837ab21239b ~]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain name,
using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[root@6837ab21239b ~]# curl 172.17.0.2
MAYANK modi
[root@6837ab21239b ~]#
```

Why is it not exiting?

We are running bin/bash command on top of the bash process

```
[root@trainer ~]# docker exec -it webserver3 bash
[root@6837ab21239b ~]#
[root@6837ab21239b ~]#
[root@6837ab21239b ~]#
[root@6837ab21239b ~]# curl 172.17.0.2
curl: (7) Failed to connect to 172.17.0.2 port 80: Connection refused
[root@6837ab21239b ~]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[root@6837ab21239b ~]# curl 172.17.0.2
MAYANK modi
[root@6837ab21239b ~]# exit
exit
[root@trainer ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAME
6837ab21239b oraclelinux:8.3 "/bin/bash" 2 hours ago Up 2 minutes webserver3
[root@trainer ~]# curl 172.17.0.2
MAYANK modi
[root@trainer ~]#
```

Manually stop the container:

```
[root@trainer ~]# docker stop webserver3
webserver3
[root@trainer ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6837ab21239b oraclelinux:8.3 "/bin/bash" 2 hours ago Exited (0) 1 second ago
webserver3
fa51e54d324b centos "/bin/bash" 2 hours ago Exited (1) 2 hours ago
webserver2
df5ab63ec788 centos "/bin/bash" 2 hours ago Exited (0) 2 hours ago
webserver1
3d3fd9b51510 centos "/bin/bash" 2 hours ago Exited (0) 2 hours ago
webserver
0b0f24cfdd47 centos "/bin/bash" 2 hours ago Exited (0) 2 hours ago
mayank1
f042ec8ee747 centos "/bin/bash" 2 hours ago Exited (0) 2 hours ago
mayank
fcf37b509826 centos "/bin/bash" 2 hours ago Exited (0) 2 hours ago
practical_colden
6fd4f7f62915 hello-world "/hello" 2 hours ago Exited (0) 2 hours ago
exciting_booth
[root@trainer ~]#
```

Rename the container:

```
[root@trainer ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6837ab21239b oraclelinux:8.3 "/bin/bash" 2 hours ago Up 1 second
webserver3
[root@trainer ~]# docker rename webserver3 collins
[root@trainer ~]#
```

```
59 docker ps -a
60 docker start webserver3
61 docker ps -a
62 docker ps
63 docker inspect webserver3
64 docker inspect webserver3 | grep -i ip
65 docker inspect webserver3 | grep IPAddress
66 curl 172.17.0.2
67 docker ps
68 pwd
69 hostname
70 docker attach webserver3
71 curl 172.17.0.2
72 docker ps -a
73 docker start webserver3
74 docker ps
75 docker exec -it webserver3 bash
76 docker ps
77 curl 172.17.0.2
78 docker ps -a
79 docker inspect webserver3
80 docker inspect webserver3 | grep -i ip
81 III
82 II
83 ls
```

```
84 history
85 docker ps | 
86 docker ps
87 docker stop webserver3
88 docker ps -a
89 history
90 docker start webserver3
91 docker ps
92 docker rename webserver3 collins
93 docker ps
94 history
```

TASK: 3

Q.1 START THE previous container and go inside into it and start the apache and access it by using curl command

```
[root@vishali ~]# docker exec -it webserver3 bash
[root@2f7adf04887a /]# curl 172.17.0.3
curl: (7) Failed to connect to 172.17.0.3 port 80: Connection refused
[root@2f7adf04887a /]# httpd -DBACKGROUND
bash: httpd: command not found
[root@2f7adf04887a /]# curl 172.17.0.3
curl: (7) Failed to connect to 172.17.0.3 port 80: Connection refused
[root@2f7adf04887a /]# yum install -y httpd
Oracle Linux 8 BaseOS Latest (x86_64)      57 MB/s | 43 MB   00:00
Oracle Linux 8 Application Stream (x86_64)  56 MB/s | 33 MB   00:00
Last metadata expiration check: 0:00:09 ago on Mon Apr 25 11:17:27 2022.
Dependencies resolved.
=====
Package          Arch    Version           Repository      Size
=====
Installing:
httpd           x86_64  2.4.37-43.0.3.module+el8.5.0+20624+5d3b49d0.3
                           ol8_appstream   1.4 M
Installing dependencies:
apr              x86_64  1.6.3-12.el8       ol8_appstream   129 k
=====

```

```

Complete!
[root@2f7adf04887a /]# cd /var/www/html/
[root@2f7adf04887a html]# vim index.html
bash: vim: command not found
[root@2f7adf04887a html]# vi index.html
[root@2f7adf04887a html]# systemctl status httpd
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
[root@2f7adf04887a html]# systemctl start httpd
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
[root@2f7adf04887a html]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain
name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress thi
s message
[root@2f7adf04887a html]# curl 172.17.0.3
<h1> This is vishali </h1>
[root@2f7adf04887a html]# EXIT
bash: EXIT: command not found
[root@2f7adf04887a html]# exit
exit
[root@vishali ~]# ^C
[root@vishali ~]# curl 172.17.0.3
<h1> This is vishali </h1>
[root@vishali ~]# █

```

MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Q.2 Try to rename any container

```

[root@vishali ~]# docker rename webserver3 collins
[root@vishali ~]# docker ps -a
CONTAINER ID   IMAGE      COMMAND          CREATED        STATUS
ATUS           oraclelinux:8.3 "/bin/bash"      58 minutes ago  Up
4 minutes      "collins"
b9fdcd79dfa   mdhack/myapache  "/bin/sh -c 'httpd -...'"  About an hour ago  Up
About an hour  80/tcp      "container6"
0c4a7204cc08   mdhack/myserver  "/bin/sh -c 'httpd -...'"  About an hour ago  Ex
ited (0) About an hour ago  "container5"
cb6d11eaf90c   hello-world    "/hello"
ited (0) About an hour ago  "container4"
766b155a5cf6   centos       "/bin/bash"
ited (0) About an hour ago  "container3"
287515c2c28d   oraclelinux:8.3 "/bin/bash"
ited (0) About an hour ago  "container2"
d20d02123f83   ubuntu        "bash"
ited (0) About an hour ago  "container1"
[root@vishali ~]# █

```

MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Q.3 You try to use both the option that I have shown you right now.

Attach and exec both

```

[root@vishali ~]# docker attach collins
[root@2f7adf04887a /]# curl 172.17.0.3
<h1> This is vishali </h1>
[root@2f7adf04887a /]# exit
exit
[root@vishali ~]# curl 172.17.0.3
curl: (7) Failed to connect to 172.17.0.3 port 80 af
t

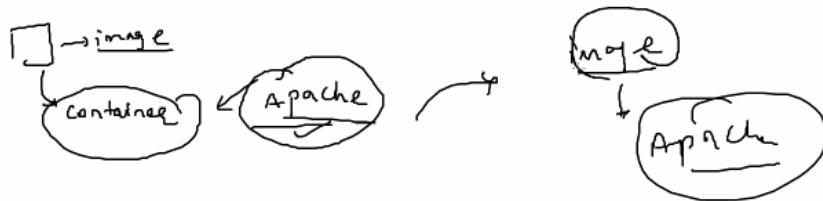
```

```
[root@vishali ~]# docker start collins
collins
[root@vishali ~]# docker exec -it collins bash
[root@2f7adf04887a /]# curl 172.17.0.3
curl: (7) Failed to connect to 172.17.0.3 port 80: Connection refused
[root@2f7adf04887a /]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
[root@2f7adf04887a /]# curl 172.17.0.3
<h1> This is vishali </h1>
[root@2f7adf04887a /]# EXIT
bash: EXIT: command not found
[root@2f7adf04887a /]# EXIT
bash: EXIT: command not found
[root@2f7adf04887a /]# exit
exit
[root@vishali ~]# curl 172.17.0.3
<h1> This is vishali </h1>
```

You have to go inside the container and start the Apache server to access the ip of the container outside the container.

We are configuring Apache server every time whenever we create the container.

We need images which have Apache server already configured



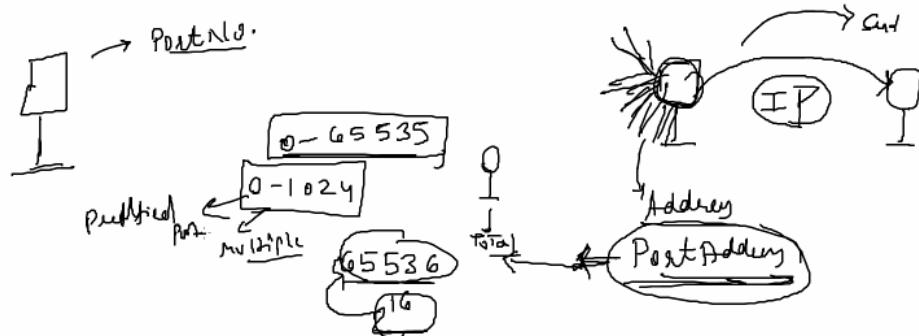
Port number:

What port number means?

A port number is **a way to identify a specific process to which an internet or other network message is to be forwarded when it arrives at a server**. All network-connected devices come equipped with standardized ports that have an assigned number.



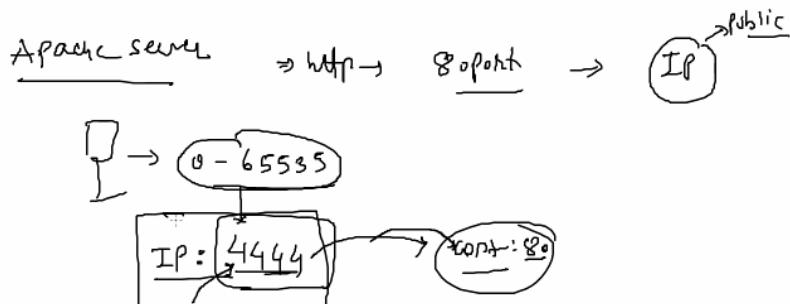
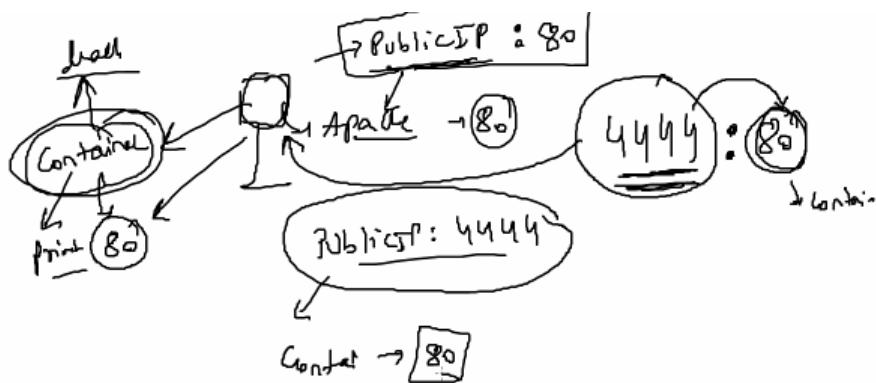
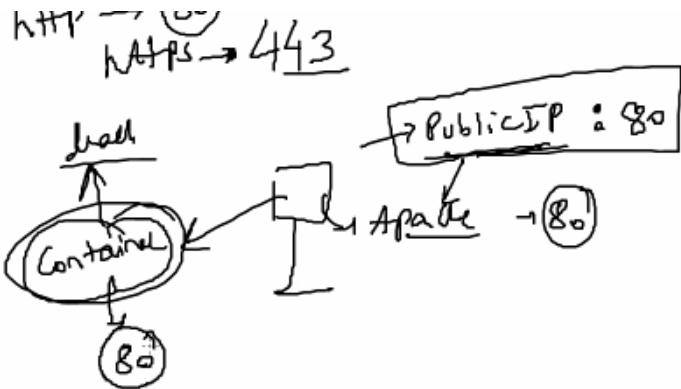
How many port number address we have for single machine?

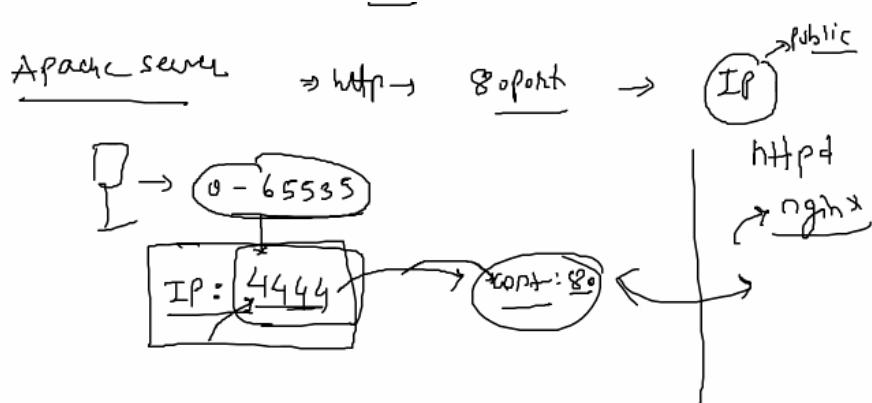


Predefined port numbers:

	Predefined port	Mw
22		
FTP → 21		
SMTP → 25		
DNS → 53		
HTTP → 80		
HTTPS → 443		

Default port number for http:





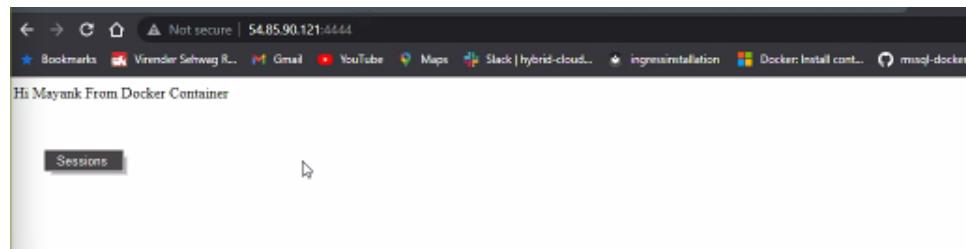
Port mapping: Done only when creating the container

```
[root@trainer ~]# docker run -it --name test1 -p 4444:80 oraclelinux:8.3
```

Nothing is running on the port

Install and configure apache server inside the container.

```
Starting logs httpd: Starting httpd...
Complete!
[root@a48f731ef3d6 /]# cd /var/www/
[root@a48f731ef3d6 www]# cd html/
[root@a48f731ef3d6 html]# s
bash: s: command not found
[root@a48f731ef3d6 html]# ls
[root@a48f731ef3d6 html]# echo Hi Mayank From Docker Container > index.html
[root@a48f731ef3d6 html]#
```



Start the service

```
# httpd -DBACKGROUND
```

We have exposed or mapped the port 4444 to 80 port number.

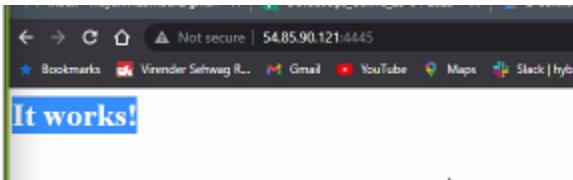
Some images are already configured with apache server

```
[root@trainer ~]# docker run -it --name test2 httpd
```

Example: nginx

```
[root@trainer ~]# docker run -it --name test2 -p 4445:80 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
1fe172e4850f: Pull complete
e2fa1fe9b1ec: Pull complete
60dd7398e74e: Pull complete
ea2ca81c6d4c: Pull complete
f646c69a26ec: Pull complete
Digest: sha256:e02a2ef36151905c790efb0a8472f690010150f062639bd8c0760e7b1e884c07
Status: Downloaded newer image for httpd:latest
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
[Mon Apr 25 11:45:55.625829 2022] [mpm_event:notice] [pid 1:tid 140663501892928] AH00489: Apache/2.4.53 (Unix) configured -- resuming normal operations
[Mon Apr 25 11:45:55.626236 2022] [core:notice] [pid 1:tid 140663501892928] AH00094: Command line: 'httpd -D FOREGROUND'
117.199.171.96 - - [25/Apr/2022:11:45:57 +0000] "GET / HTTP/1.1" 200 45
117.199.171.96 - - [25/Apr/2022:11:45:57 +0000] "GET /favicon.ico HTTP/1.1" 404 196
```

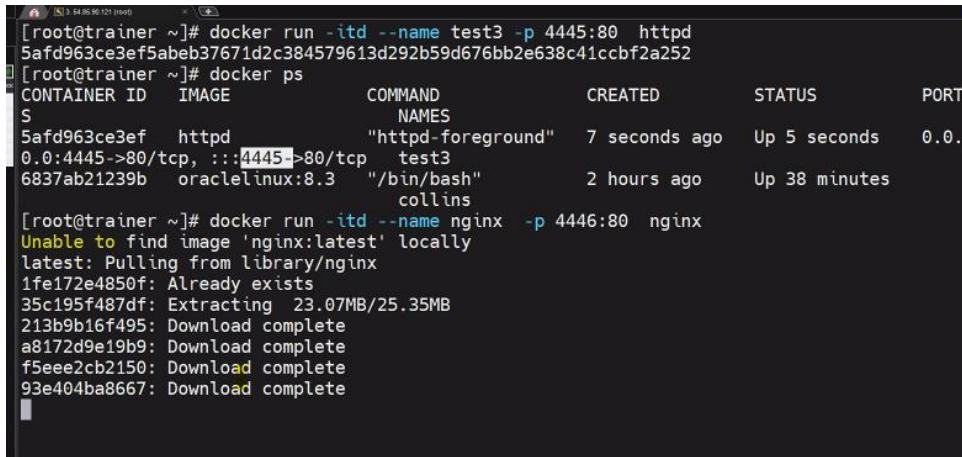
In front of you, it is running - foreground



If you don't want to run in front of you use itd

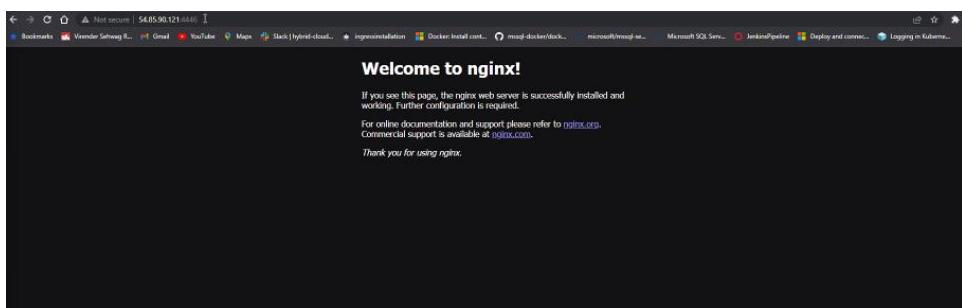
```
[root@trainer ~]# docker run -itd --name test2 -p 4445:80 httpd
```

D stands for detach



```
[root@trainer ~]# docker run -itd --name test3 -p 4445:80 httpd
5af963ce3ef5abeb37671d2c384579613d292b59d676bb2e638c41ccbf2a252
[root@trainer ~]# docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS
S              httpd     "httpd-foreground"  7 seconds ago Up 5 seconds  0.0.
0.0:4445->80/tcp, :::4445->80/tcp  test3
6837ab21239b  oraclelinux:8.3  "/bin/bash"    2 hours ago  Up 38 minutes
collins

[root@trainer ~]# docker run -itd --name nginx -p 4446:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
1fe172e4850f: Already exists
35c195f487df: Extracting 23.07MB/25.35MB
213b9b16f495: Download complete
a8172d9e19b9: Download complete
f5eee2cb2150: Download complete
93e404ba8667: Download complete
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.

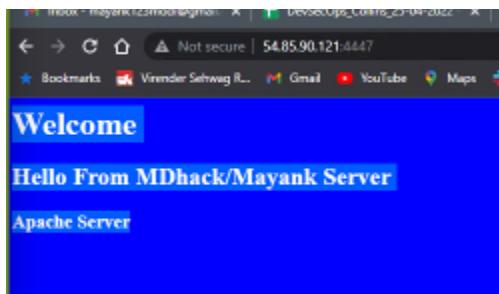
Commercial support is available at nginx.com.

Thank you for using nginx.

Create your own container:

Example 1:

```
[root@trainer ~]# docker run -itd --name my -p 4447:80 mdhack/myapache
Unable to find image 'mdhack/myapache:latest' locally
latest: Pulling from mdhack/myapache
bf5952930446: Pull complete
3d3fecf6569b: Pull complete
b5fc3125d912: Pull complete
679d69c01e90: Pull complete
76291586768e: Pull complete
8fb864d2a9c7: Pull complete
Digest: sha256:1c90388d0472b6774770c06489be3f74f8d38221a42f024f03c3caec34b87714
Status: Downloaded newer image for mdhack/myapache:latest
645a1ab87aeda139877059684a984cb3d901a495ccc9e257847e258bf79d8463
[root@trainer ~]#
```



Example 2:

```
[root@trainer ~]# docker run -itd --name my1 -p 4448:80 mdhack/myserver
Unable to find image 'mdhack/myserver:latest' locally
latest: Pulling from mdhack/myserver
74f0853ba93b: Extracting 66.29MB/72.25MB
7aa70b934c32: Download complete
2d68deff9aaf: Download complete
31ea4127808d: Download complete
```

Welcome to MDhack/Mayank web server for testing

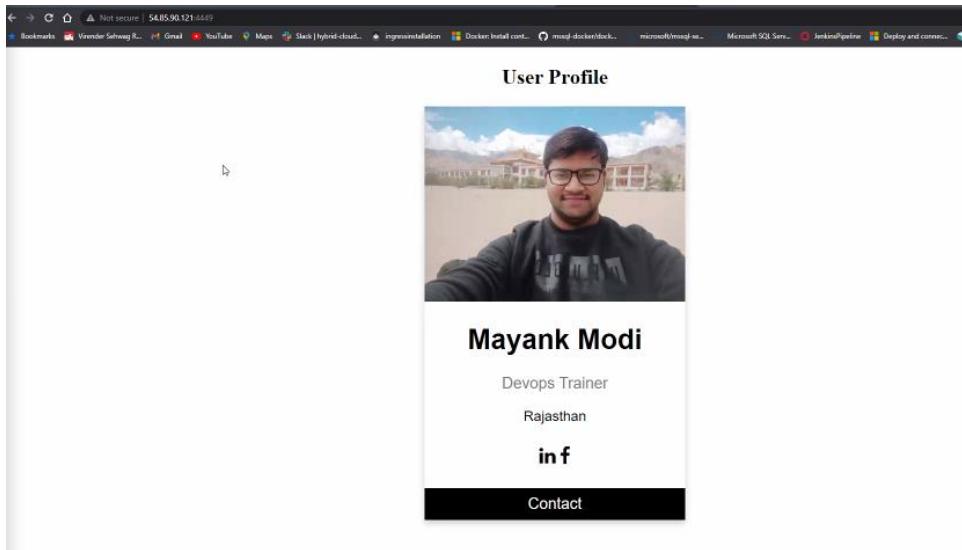
```
eth0: flags=4163 mtu 1500
      inet 172.17.0.6 netmask 255.255.0.0 broadcast 172.17.255.255
      ether 02:42:ac:11:00:06 txqueuelen 0 (Ethernet)
      RX packets 10 bytes 1175 (1.1 Kib)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 5 bytes 282 (282.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      loop txqueuelen 1000 (Local Loopback)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

You will Definately enjoy this training

Example 3:

```
[root@trainer ~]# docker run -itd --name my2 -p 4449:80 mdhack/mayanknginximage
Unable to find image 'mdhack/mayanknginximage:latest' locally
latest: Pulling from mdhack/mayanknginximage
5eb5b503b376: Downloading 4.194MB/31.37MB
1ae07ab881bd: Downloading 263.7kB/25.35MB
78091884b7be: Download complete
091c283c6a66: Download complete
55de5851019b: Waiting
b559bad762be: Waiting
b2d854f6f009: Waiting
```



http://54.85.90.121:4449/

http://54.85.90.121:4448

http://54.85.90.121:4447

http://54.85.90.121:4446

http://54.85.90.121:4445

TASK 4:

Q.1 create a container using oraclelinux:8.3 image and configure apache inside into it and come out from the container and then use curl command to verify whether it is working or not name should be task1.

docker run --name task1 -it oraclelinux:8.3

```
[root@vishali ~]# docker run --name task1 -it oraclelinux:8.3
[root@e449ba86c341 /]# yum install -y httpd
Oracle Linux 8 BaseOS Latest (x86_64)           25 MB/s | 43 MB    00:0
Oracle Linux 8 Application Stream (x86_64)       75 MB/s | 33 MB    00:0
Last metadata expiration check: 0:00:09 ago on Mon Apr 25 12:00:19 2022.
Dependencies resolved.
=====
 Package          Arch      Version           Repository
=====
 Installing:
 httpd           x86_64    2.4.37-43.0.3.moduleol8.5.0+20624+5d3b40d0_3
```

```
Complete!
[root@e449ba86c341 ~]# cd /var/www/html/
[root@e449ba86c341 html]# vi index.html
[root@e449ba86c341 html]# cat index.html
<h1> This is task1 container </h1>
[root@e449ba86c341 html]# cd /var/www/html/
[root@e449ba86c341 html]# systemctl status httpd
System has not been booted with systemd as init system (PID 1). Can't op
Failed to connect to bus: Host is down
[root@e449ba86c341 html]# systemctl start httpd
System has not been booted with systemd as init system (PID 1). Can't op
Failed to connect to bus: Host is down
```

```
[root@e449ba86c341 html]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN gro
t qdisc mq
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
46: eth0@if47: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
    group default
        link/ether 02:42:ac:11:00:04 brd ff:ff:ff:ff:ff:ff link-netnsid 0
        inet 172.17.0.4/16 brd 172.17.255.255 scope global eth0
            valid_lft forever preferred_lft forever
[root@e449ba86c341 html]# curl 127.0.0.1
curl: (7) Failed to connect to 127.0.0.1 port 80: Connection refused
[root@e449ba86c341 html]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualifi
    name, using 172.17.0.4. Set the 'ServerName' directive globally to sup
    s message
[root@e449ba86c341 html]# curl 127.0.0.1
<h1> This is task1 container </h1>
[root@e449ba86c341 html]# █
```

```
[root@vishali ~]# docker start task1
task1
[root@vishali ~]# docker exec -it task1 bash
[root@e449ba86c341 /]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group 0
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
48: eth0@if49: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
[root@e449ba86c341 /]# curl 172.17.0.2
curl: (7) Failed to connect to 172.17.0.2 port 80: Connection refused
[root@e449ba86c341 /]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified name, using 172.17.0.2. Set the 'ServerName' directive globally to support this message
[root@e449ba86c341 /]# curl 172.17.0.2
<h1> This is task1 container </h1>
[root@e449ba86c341 /]# EXIT
bash: EXIT: command not found
[root@e449ba86c341 /]# exit
exit
[root@vishali ~]# curl 172.17.0.2
<h1> This is task1 container </h1>
[root@vishali ~]#
```

MohaXterm hv subscribing to the professional edition here: <https://mohaxterm.mohatek.net>

Q.2 Create a container using same image as above and expose it on 3333 port no. and configure apache into it so u can verify from ur browser whether it is working or not name should be task2

```

[root@vishali ~]# docker run --name task2 -itd -p 3333:80 oraclelinux:8.3
5501a10168ca67f7589db212910a6fb4468cb14f1740fd90eb8a5b216ae48abb
[root@vishali ~]# docker exec -it task2 bash
[root@5501a10168ca /]# yum install -y httpd
=====
Package           Arch    Version          Repository      Size
=====
[root@5501a10168ca html]# cat index.html
<h1> This is task2 container </h1>
[root@5501a10168ca html]# systemctl status httpd
System has not been booted with systemd as init system (PID 1). Can't operate
Failed to connect to bus: Host is down
[root@5501a10168ca html]# systemctl start httpd
System has not been booted with systemd as init system (PID 1). Can't operate
Failed to connect to bus: Host is down
[root@5501a10168ca html]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qdisc mq 0: link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 00:00:00:00:00:00 scope host lo
        valid_lft forever preferred_lft forever
50: eth0@if51: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
group default
    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
[root@5501a10168ca html]# curl 172.17.0.3
curl: (7) Failed to connect to 172.17.0.3 port 80: Connection refused
[root@5501a10168ca html]# httpd -DBACKGROUND
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
[root@5501a10168ca html]# curl 172.17.0.3
<h1> This is task2 container </h1>
[root@5501a10168ca html]# exit
exit
[root@vishali ~]# curl 172.17.0.3
<h1> This is task2 container </h1>
[root@vishali ~]# █

```

Q.3 CREATE A CONtainer using httpd image and expose it on port no. 3334 name should be task3

```

[11:12:45 ~]# curl 172.17.0.3
[root@vishali ~]# docker run --name task3 -itd -p 3334:80 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
1fe172e4850f: Pull complete
e2fa1fe9b1ec: Pull complete
60dd7398e74e: Pull complete
ea2ca81c6d4c: Pull complete
f646c69a26ec: Pull complete
Digest: sha256:e02a2ef36151905c790efb0a8472f690010150f062639bd8c0760e7b1e884c
Status: Downloaded newer image for httpd:latest
8bd05642abbd59caef62db578a8fe75399126e29e5a81f820f0428ff1545340c
[root@vishali ~]# █

```

Q.4 CREATE A CONtainer using nginx image and expose it on port no. 3335 name should be task4

```
8bd05b42abbd59cae162db578a81e75399126e29e5a81182010428111545340c
[root@vishali ~]# docker run --name task4 -itd -p 3335:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
1fe172e4850f: Already exists
35c195f487df: Pull complete
213b9b16f495: Pull complete
a8172d9e19b9: Pull complete
f5eee2cb2150: Pull complete
93e404ba8667: Pull complete
Digest: sha256:859ab6768a6f26a79bc42b231664111317d095a4f04e4b6fe79ce37b3d199097
Status: Downloaded newer image for nginx:latest
1f325d33d7d115510ee40837d54eb01b55251bc4bfc67fbcf563be73715523f9
[root@vishali ~]#
```

Q.5 CREATE A CONTAINER using mdhack/myapache image and expose it on port no. 3336 name should be task5

```
[root@vishali ~]# docker run --name task5 -itd -p 3336:80 mdhack/myapache
67fdb9edb4fba2ef1712a47d54d1224f8b64ef6cc0f0b6a5cc95298e5486b03
```

Q.6 CREATE A CONTAINER using mdhack/myserver image and expose it on port no. 3338 name should be task6

```
[root@vishali ~]# docker run --name task6 -itd -p 3338:80 mdhack/myserver
d2e8c264ed5b48062f69c901f34b92619ef57b4b7addef6d3b1b9d638f32369f
```

Q.7 CREATE A CONTAINER using mdhack/mayanknginximage image and expose it on port no. 3337 name should be task7

```
[root@vishali ~]# docker run --name task7 -itd -p 3337:80 mdhack/mayanknginximage
Unable to find image 'mdhack/mayanknginximage:latest' locally
latest: Pulling from mdhack/mayanknginximage
5eb5b503b376: Pull complete
1ae07ab881bd: Pull complete
78091884b7be: Pull complete
091c283c6a66: Pull complete
55de5851019b: Pull complete
b559bad762be: Pull complete
b2d854f6f009: Pull complete
Digest: sha256:57e74ddb1e8c223646c4779d71118716381570efcd6a71d047c85e736b731b85
Status: Downloaded newer image for mdhack/mayanknginximage:latest
1dc2d41f8afe44d50706383b0fa296276c4bde386fc5ddb12548eaa644d358
[root@vishali ~]#
```

Output:

http://44.202.147.151:3333/
http://44.202.147.151:3334/
http://44.202.147.151:3335/
http://44.202.147.151:3336/
http://44.202.147.151:3338/
http://44.202.147.151:3337/

