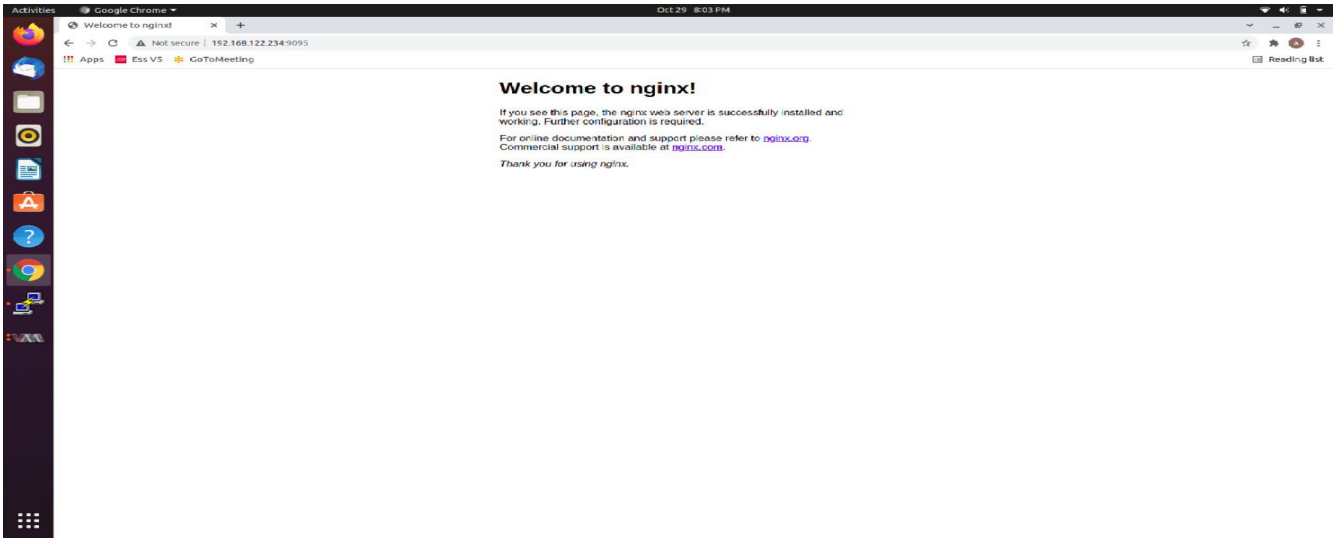
Assignment docker

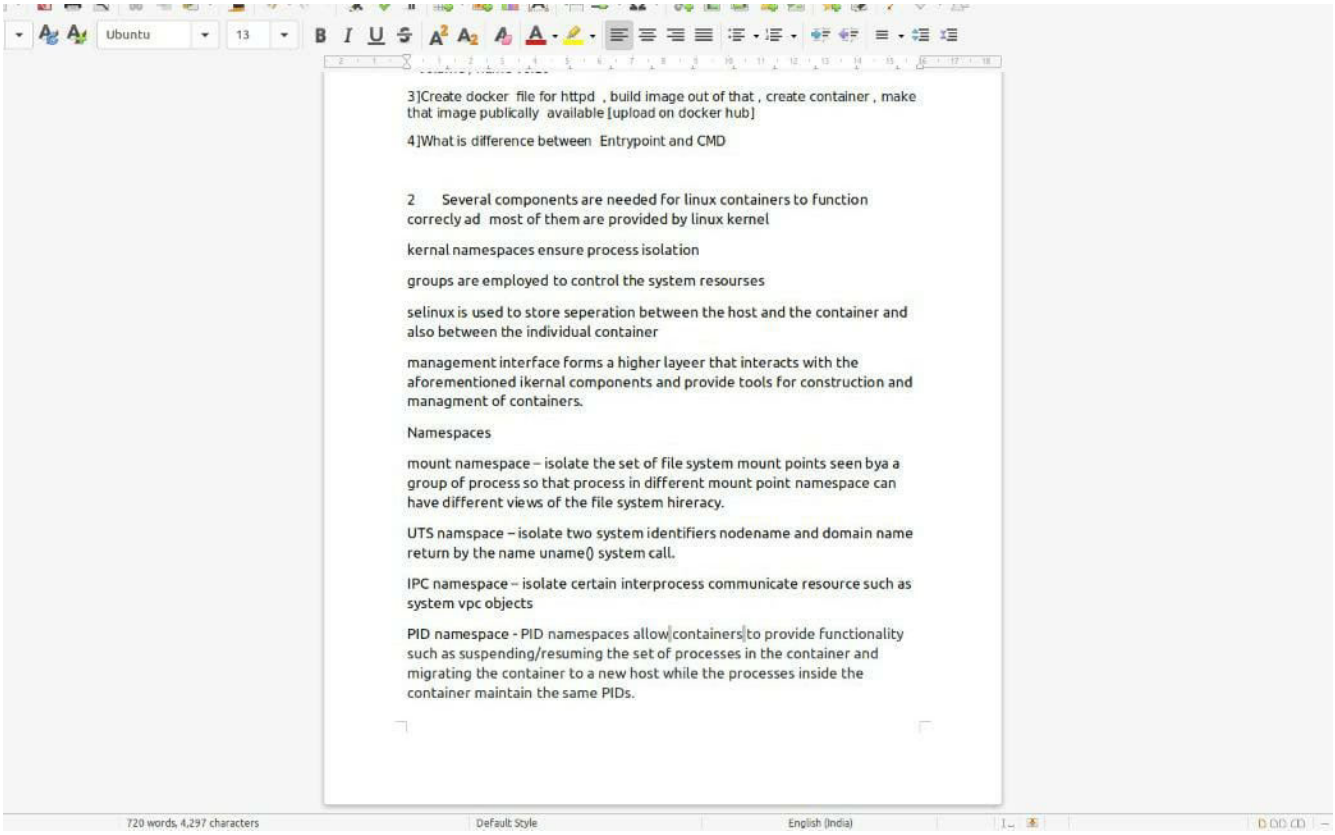
1.



For Nginx using port 9095:80

For httpd using port 9096:80

2.



3]Create docker file for httpd , build image out of that , create container , make that image publically available [upload on docker hub]

4]What is difference between Entrypoint and CMD

2 Several components are needed for linux containers to function correcly ad most of them are provided by linux kernel

kernal namespaces ensure process isolation

groups are employed to control the system resourses

selinux is used to store seperation between the host and the container and also between the individual container

management interface forms a higher layeer that interacts with the aforementioned ikernal components and provide tools for construction and managment of containers.

Namespaces

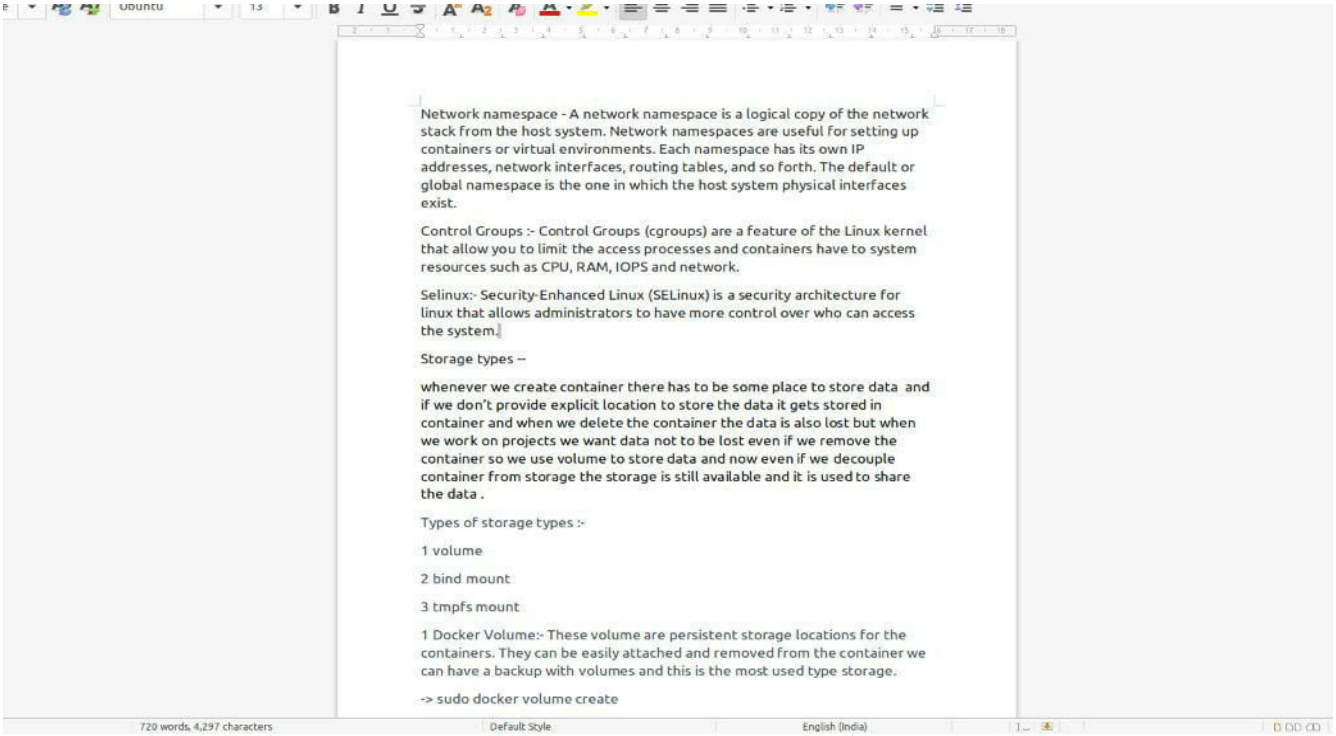
mount namespace – isolate the set of file system mount points seen bya a group of process so that process in different mount point namespace can have different views of the file system hireracy.

UTS namspace – isolate two system identifiers nodename and domain name return by the name uname() system call.

IPC namespace – isolate certain interprocess communicate resource such as system vpc objects

PID namespace - PID namespaces allow containers to provide functionality such as suspending/resuming the set of processes in the container and migrating the container to a new host while the processes inside the container maintain the same PIDs.

720 words, 4,297 characters Default Style English (India)



Network namespace - A network namespace is a logical copy of the network stack from the host system. Network namespaces are useful for setting up containers or virtual environments. Each namespace has its own IP addresses, network interfaces, routing tables, and so forth. The default or global namespace is the one in which the host system physical interfaces exist.

Control Groups :- Control Groups (cgroups) are a feature of the Linux kernel that allow you to limit the access processes and containers have to system resources such as CPU, RAM, IOPS and network.

Selinux:- Security-Enhanced Linux (SELinux) is a security architecture for linux that allows administrators to have more control over who can access the system.

Storage types --

whenever we create container there has to be some place to store data and if we don't provide explicit location to store the data it gets stored in container and when we delete the container the data is also lost but when we work on projects we want data not to be lost even if we remove the container so we use volume to store data and now even if we decouple container from storage the storage is still available and it is used to share the data .

Types of storage types :-

1 volume

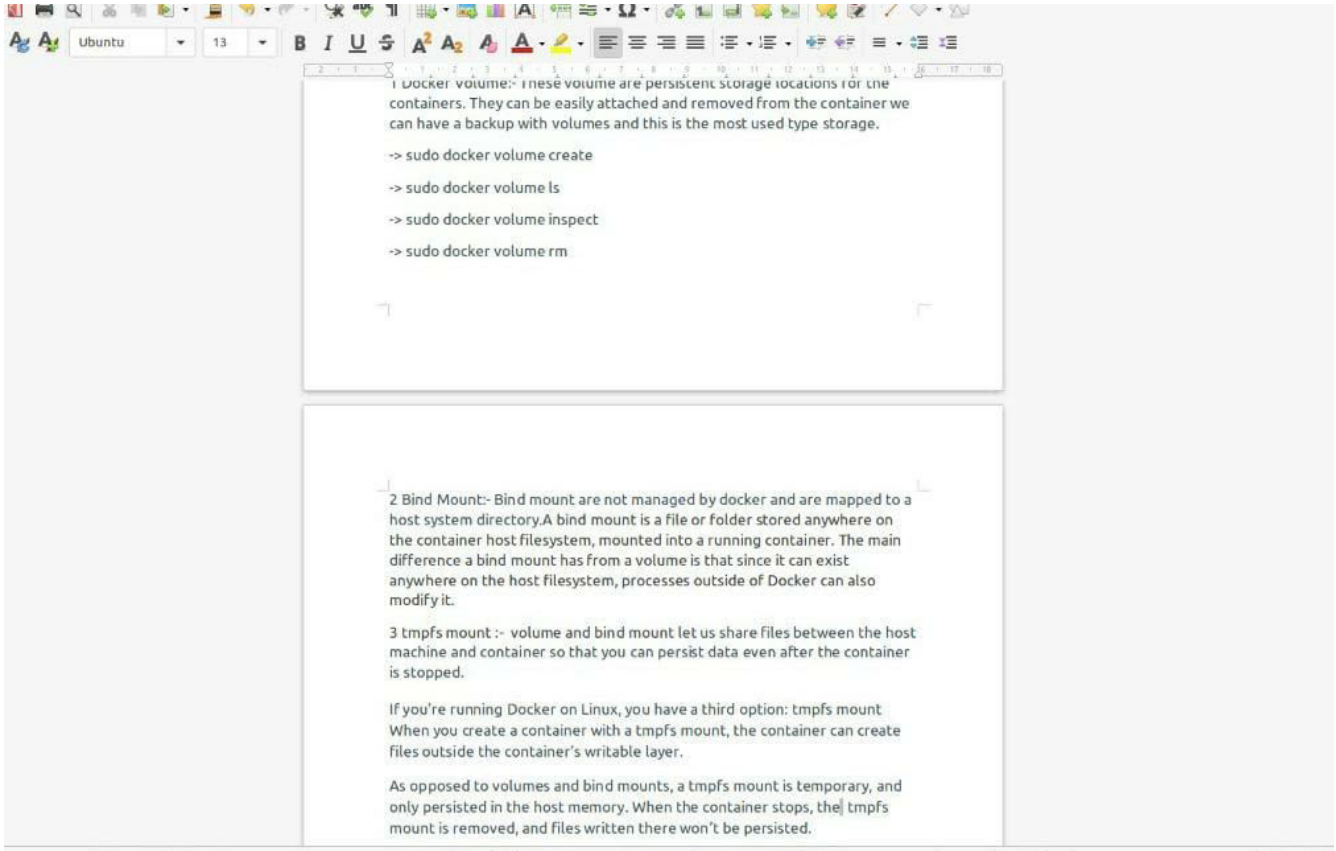
2 bind mount

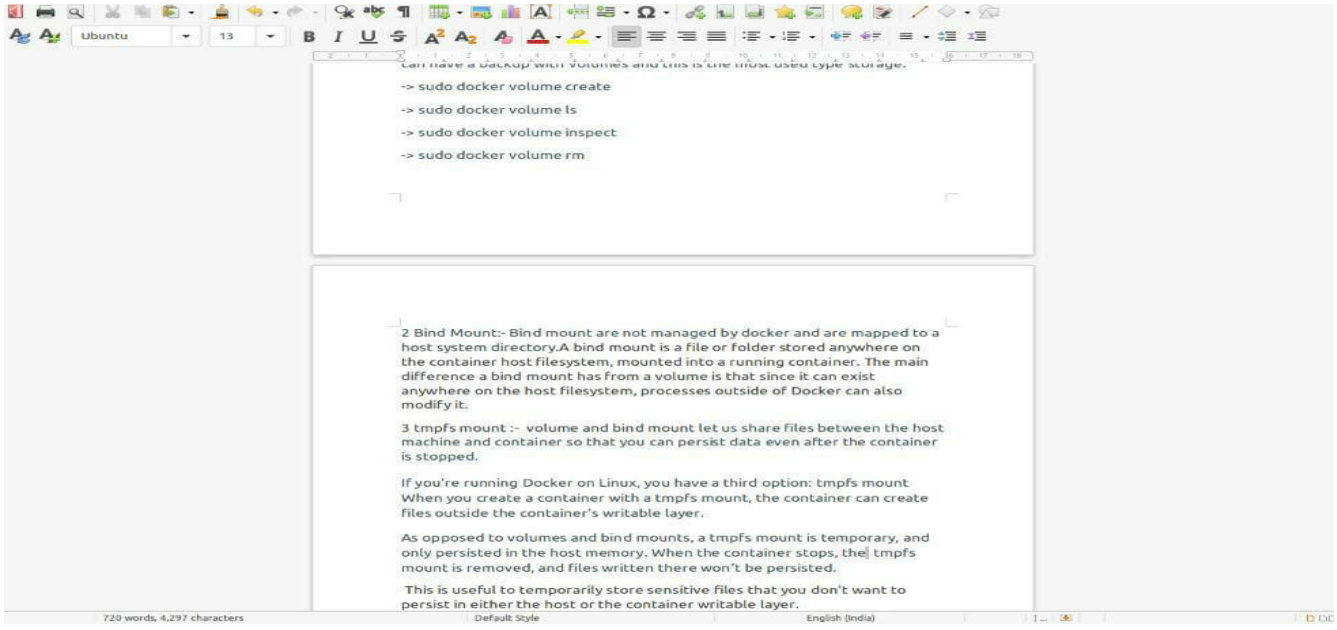
3 tmpfs mount

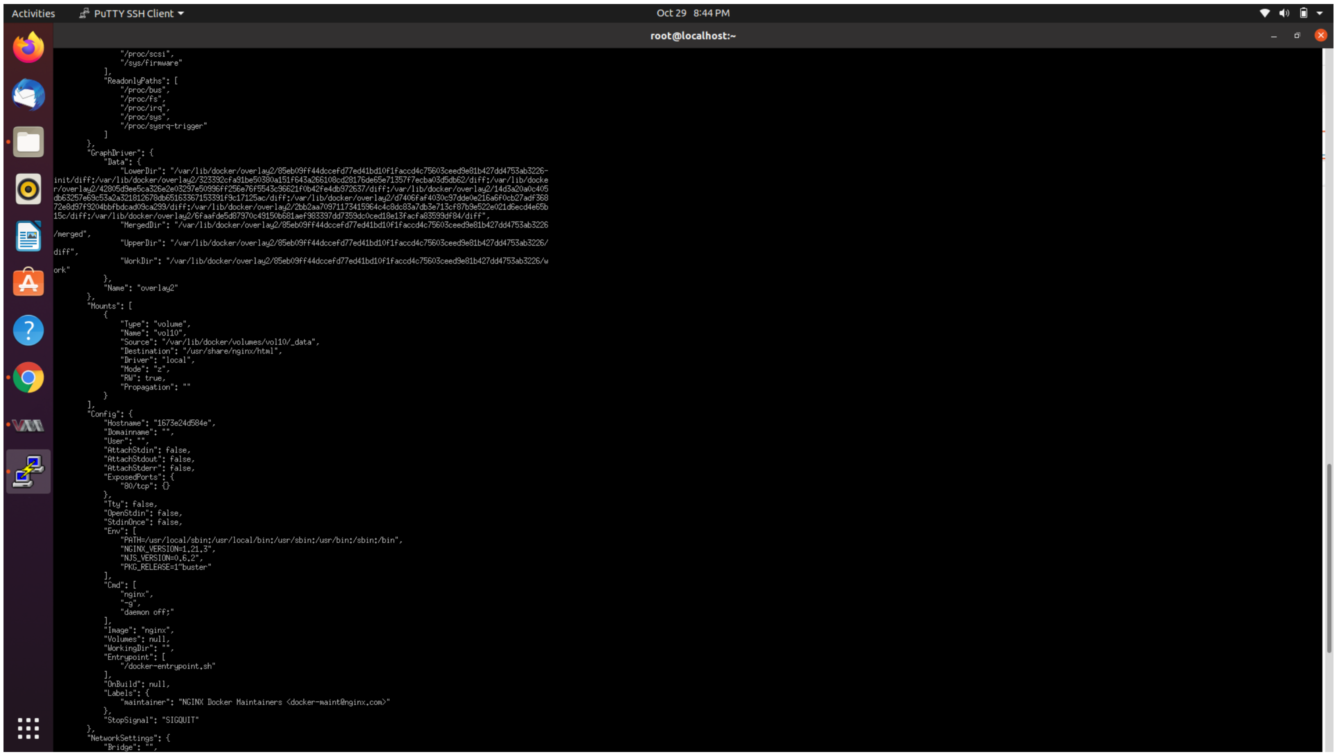
1 Docker Volume:- These volume are persistent storage locations for the containers. They can be easily attached and removed from the container we can have a backup with volumes and this is the most used type storage.

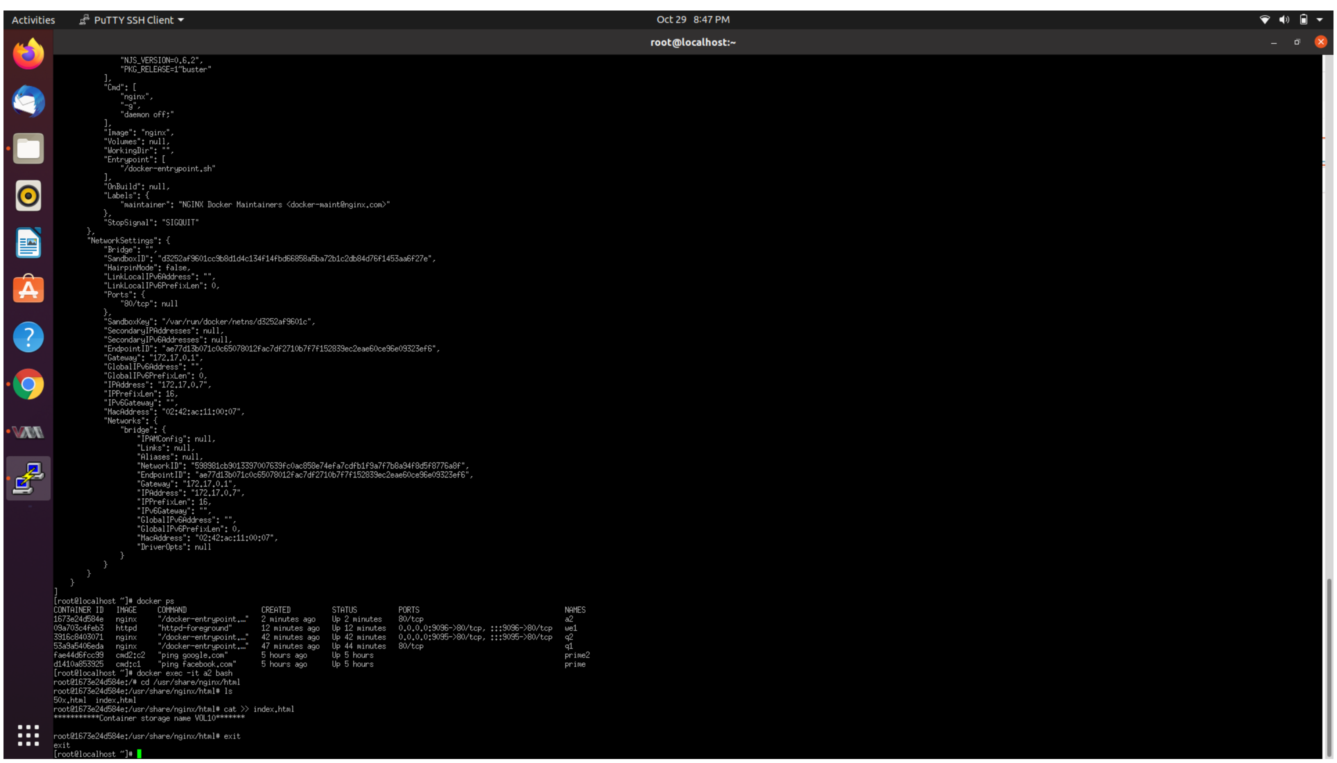
-> sudo docker volume create

720 words, 4,297 characters Default Style English (India)

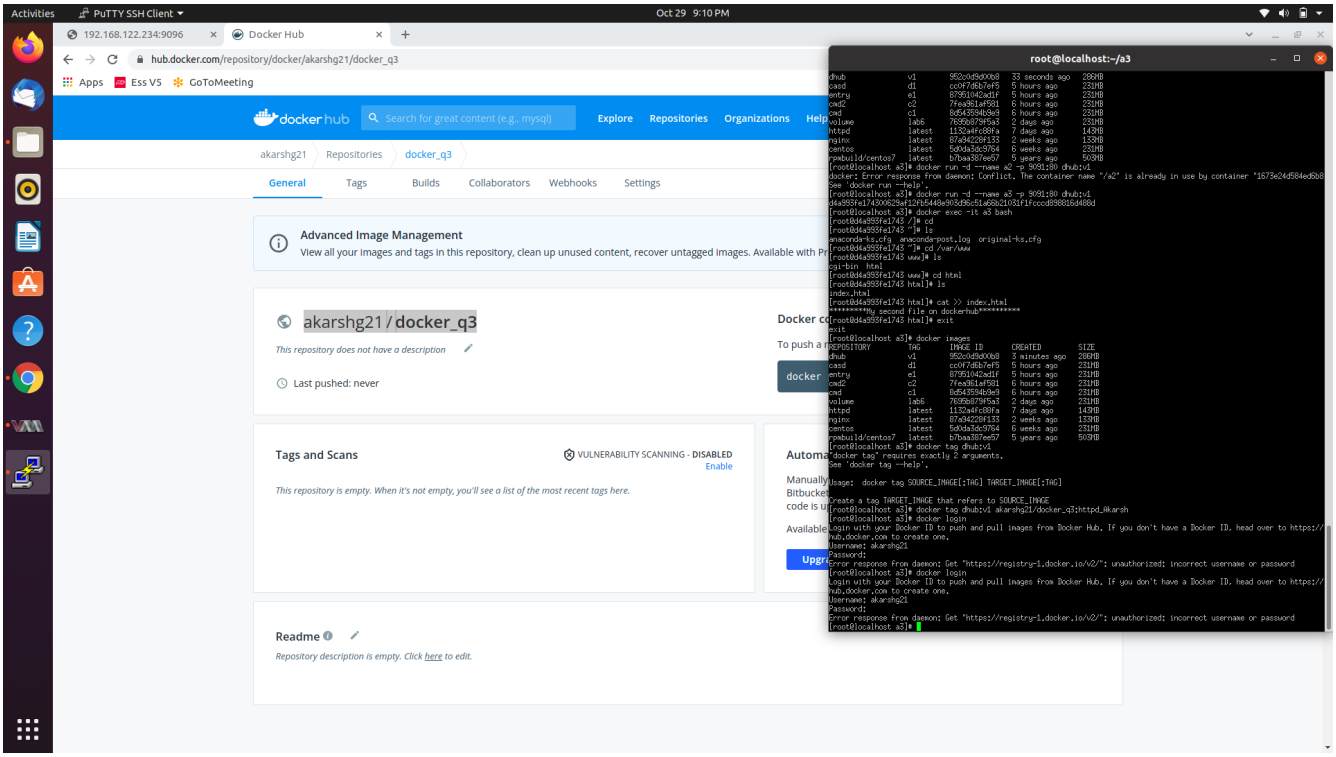

Ubuntu
13
1 Docker Volume:- These volume are persistent storage locations for the containers. They can be easily attached and removed from the container we can have a backup with volumes and this is the most used type storage.
-> sudo docker volume create
-> sudo docker volume ls
-> sudo docker volume inspect
-> sudo docker volume rm
2 Bind Mount:- Bind mount are not managed by docker and are mapped to a host system directory.A bind mount is a file or folder stored anywhere on the container host filesystem, mounted into a running container. The main difference a bind mount has from a volume is that since it can exist anywhere on the host filesystem, processes outside of Docker can also modify it.
3 tmpfs mount :- volume and bind mount let us share files between the host machine and container so that you can persist data even after the container is stopped.
If you're running Docker on Linux, you have a third option: tmpfs mount
When you create a container with a tmpfs mount, the container can create files outside the container's writable layer.
As opposed to volumes and bind mounts, a tmpfs mount is temporary, and only persisted in the host memory. When the container stops, the tmpfs mount is removed, and files written there won't be persisted.


Ubuntu
13
-> sudo docker volume create
-> sudo docker volume ls
-> sudo docker volume inspect
-> sudo docker volume rm
2 Bind Mount:- Bind mount are not managed by docker and are mapped to a host system directory.A bind mount is a file or folder stored anywhere on the container host filesystem, mounted into a running container. The main difference a bind mount has from a volume is that since it can exist anywhere on the host filesystem, processes outside of Docker can also modify it.
3 tmpfs mount :- volume and bind mount let us share files between the host machine and container so that you can persist data even after the container is stopped.
If you're running Docker on Linux, you have a third option: tmpfs mount
When you create a container with a tmpfs mount, the container can create files outside the container's writable layer.
As opposed to volumes and bind mounts, a tmpfs mount is temporary, and only persisted in the host memory. When the container stops, the tmpfs mount is removed, and files written there won't be persisted.
This is useful to temporarily store sensitive files that you don't want to persist in either the host or the container writable layer.
720 words, 4,297 characters
Default Style
English (India)

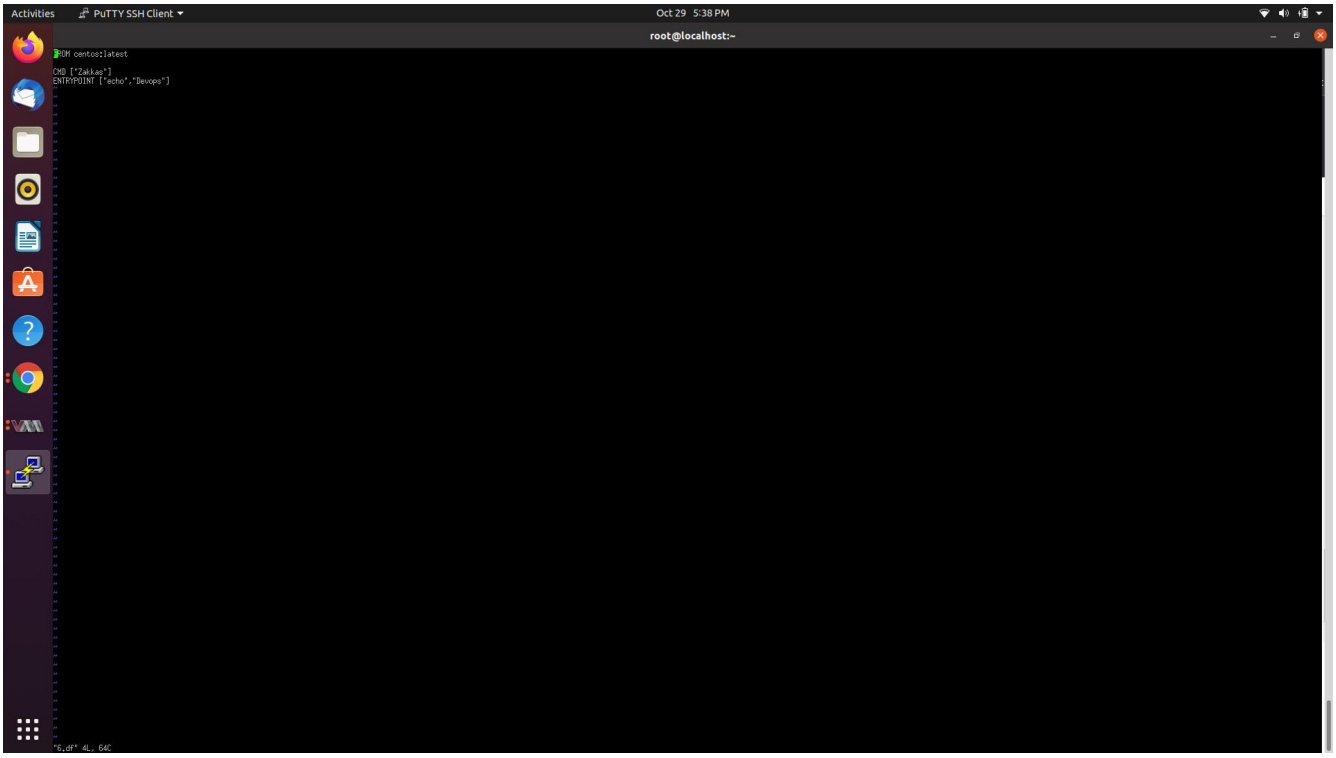

Activities
PuTTY SSH Client
Oct 29 8:44 PM
root@localhost:~
"/proc/scsi",
"/sys/firmware"
],
"ReadonlyPaths": [
"/proc/bus",
"/proc/fs",
"/proc/irq",
"/proc/sys",
"/proc/sysrq-trigger"
]
},
"GraphDriver": {
"Data": {
"LowerDir": "/var/lib/docker/overlay2/85eb09ff44dccefd77ed41bd10f1faccd4c75603ceed9e81b427dd4753ab3226-init/diff:/var/lib/docker/overlay2/323392cfa91be50380a151f643a266108cd28176de65e71357f7ecba03d5db62/diff:/var/lib/docker/overlay2/42805d9ee5ca326e2e03297e50996ff256e76f5543c96621f0b42fe4db972637/diff:/var/lib/docker/overlay2/14d3a20a0c405db63257e69c53a2a321812678db6516336715339 1f9c17125ac/diff:/var/lib/docker/overlay2/d7406faf4030c97dde0e216a6f0cb27adf36872e8d97f9204bbfbdcad09ca299/diff:/var/lib/docker/overlay2/2bb2aa70971173415964c4c8dc83a7db3e713cf87b9e522e021d6ecd4e65b15c/diff:/var/lib/docker/overlay2/6faafde5d87970c49150b681aef983397dd7359dc0ced18e13facfa83599df84/diff",
"MergedDir": "/var/lib/docker/overlay2/85eb09ff44dccefd77ed41bd10f1faccd4c75603ceed9e81b427dd4753ab3226/merged",
"UpperDir": "/var/lib/docker/overlay2/85eb09ff44dccefd77ed41bd10f1faccd4c75603ceed9e81b427dd4753ab3226/diff",
"WorkDir": "/var/lib/docker/overlay2/85eb09ff44dccefd77ed41bd10f1faccd4c75603ceed9e81b427dd4753ab3226/work"
},
"Name": "overlay2"
},
"Mounts": [
{
"Type": "volume",
"Name": "vol10",
"Source": "/var/lib/docker/volumes/vol10/_data",
"Destination": "/usr/share/nginx/html",
"Driver": "local",
"Mode": "z",
"RW": true,
"Propagation": ""
}
],
"Config": {
"Hostname": "1673e24d584e",
"Domainname": "",
"User": "",
"AttachStdin": false,
"AttachStdout": false,
"AttachStderr": false,
"ExposedPorts": {
"80/tcp": {}
},
"Tty": false,
"OpenStdin": false,
"StdinOnce": false,
"Env": [
"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
"NGINX_VERSION=1.21.3",
"NJS_VERSION=0.6.2",
"PKG_RELEASE=1~buster"
],
"Cmd": [
"nginx",
"-g",
"daemon off;"
],
"Image": "nginx",
"Volumes": null,
"WorkingDir": "",
"Entrypoint": [
"/docker-entrypoint.sh"
],
"OnBuild": null,
"Labels": {
"maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"
},
"StopSignal": "SIGQUIT"
},
"NetworkSettings": {
"Bridge": "",


Activities
PuTTY SSH Client
Oct 29 8:47 PM
root@localhost:~
"NJS_VERSION=0.6.2",
"PKG_RELEASE=1~buster"
],
"Cmd": [
"nginx",
"-g",
"daemon off;"
],
"Image": "nginx",
"Volumes": null,
"WorkingDir": "",
"Entrypoint": [
"/docker-entrypoint.sh"
],
"OnBuild": null,
"Labels": {
"maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"
},
"StopSignal": "SIGQUIT"
},
"NetworkSettings": {
"Bridge": "",
"SandboxID": "d3252af9601cc9b8d1d4c134f14fbd66858a5ba72b1c2db84d76f1453aa6f27e",
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"Ports": {
"80/tcp": null
},
"SandboxKey": "/var/run/docker/netns/d3252af9601c",
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "ae77d13b071c0c65078012fac7df2710b7f7f152839ec2eae60ce96e09323ef6",
"Gateway": "172.17.0.1",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.7",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"MacAddress": "02:42:ac:11:00:07",
"Networks": {
"bridge": {
"IPAMConfig": null,
"Links": null,
"Aliases": null,
"NetworkID": "598981cb9013397007639fc0ac858e74efa7cdfb1f9a7f7b8a94f8d5f8776a8f",
"EndpointID": "ae77d13b071c0c65078012fac7df2710b7f7f152839ec2eae60ce96e09323ef6",
"Gateway": "172.17.0.1",
"IPAddress": "172.17.0.7",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"MacAddress": "02:42:ac:11:00:07",
"DriverOpts": null
}
}
}
}
]
[root@localhost ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
1673e24d584e nginx "/docker-entrypoint.…" 2 minutes ago Up 2 minutes 80/tcp a2
09a703c4feb3 httpd "httpd-foreground" 12 minutes ago Up 12 minutes 0.0.0.0:9096->80/tcp, :::9096->80/tcp we1
3916c8403071 nginx "/docker-entrypoint.…" 42 minutes ago Up 42 minutes 0.0.0.0:9095->80/tcp, :::9095->80/tcp q2
53a9a5406eda nginx "/docker-entrypoint.…" 47 minutes ago Up 44 minutes 80/tcp q1
fae44d6fcc99 cmd2:c2 "ping google.com" 5 hours ago Up 5 hours prime2
d1410a853925 cmd:c1 "ping facebook.com" 5 hours ago Up 5 hours prime
[root@localhost ~]# docker exec -it a2 bash
root@1673e24d584e:/# cd /usr/share/nginx/html
root@1673e24d584e:/usr/share/nginx/html# ls
50x.html index.html
root@1673e24d584e:/usr/share/nginx/html# cat >> index.html
***********Container storage name VOL10*******
root@1673e24d584e:/usr/share/nginx/html# exit
exit
[root@localhost ~]#

3.


Activities
PuTTY SSH Client
Oct 29 9:10 PM
192.168.122.234:9096
Docker Hub
hub.docker.com/repository/docker/akarshg21/docker_q3
Apps
Ess VS
GoToMeeting
docker hub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
akarshg21
Repositories
docker_q3
General
Tags
Builds
Collaborators
Webhooks
Settings
Advanced Image Management
View all your images and tags in this repository, clean up unused content, recover untagged images. Available with
akarshg21 / docker_q3
This repository does not have a description
Last pushed: never
Tags and Scans
VULNERABILITY SCANNING - DISABLED
Enable
This repository is empty. When it's not empty, you'll see a list of the most recent tags here.
Readme
Repository description is empty. Click here to edit.
root@localhost:~/a3

4.


Activities
PuTTY SSH Client
Oct 29 5:38 PM
root@localhost:~
FROM centos:latest
CMD ["Zakkas"]
ENTRYPOINT ["echo","Devops"]
"G.df" 4L, 64C

```
13  vi file.df
14  vi 6.df
15  docker build . -f 6.df -t casd:d1
16  docker run -d --name prime9 casd:d1
17  docker ps -a
18  docker logs prime9
19  cat 6.df
20  docker run -d --name prime10 casd:d1 good students
21  docker logs prime10
```

```
  831  history
[root@localhost ~]# docker logs prime10
Devops good students
[root@localhost ~]#
```