**DOCKER**

Q: How to Install Docker in centos?

* Setup the Docker Ce repository

y**um-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo**

**or**

sudo yum-config-manager \

--add-repo \

https://download.docker.com/linux/centos/docker-ce.repo

* Install Docker Package.

**Yum –y install docker-ce**

* Start docker daemon

**systemctl start docker/service start docker (RHEL)**

* Check the Docker status

**systemctl status docker**

Q: How to install docker in ubuntu?

* **apt update**
* **apt install docker.io**

Q: How to create container in docker?

Syntax: **docker run –it <image> <command>**

Exa**: docker run –it ubuntu /bin/bash**

**i:** Used for enabling interactive mode

**t:** Used for connecting(attaching) the container to current terminal.

\*\* **By default. Once we exit from container, container will stop**

Q: How to assign (tag) a name to a container while creating for easily identification?

Syntax: **docker run --name <name of container> -it <image> <command>**

Exa: **docker run --name ubuntu-test –it ubuntu /bin/bash**

Q: How to start the container?

Syntax: **docker start <containerID/container name>**

Exa: **docker start ubuntu-test**

Q: How to attach a container to terminal?

Syntax: **docker attach <containerID/container name>**

Exa: **docker attach ubuntu-test**

Q: How to create a daemonized container?

\*\* **Daemonized container means, create a container and start the container and run it without**

**attaching to terminal.**

Syntax: **docker run --name <name of container> -d <image> <command>**

Exa: **docker run --name damcont –d ubuntu /bin/bash**

**\*\*<name of container>= the name you wants to provide for the container (it can be anything)**

Q: How to fetch container log (How to see/monitor the logs (including live logs)of the container)?

**\*\* As we have created container as daemonized mode, If we wants to see, what’s going on in the container we can see the logs using below command**.

Syntax: **docker logs –f <containerId/name>**

Exa: **docker logs –f damcont**

Q: How to list the running containers?

Ans: **docker ps**

Q: How to list all available containers (running & stopped)?

Ans: **docker ps –a**

Q: How to list only running containers **ID**?

Ans: **docker ps –q**

Q: How to stop the running container?

Syntax: **docker stop <containerId/name>**

Exa: **docker stop damcont**

Q: How to list the running process in the container?

Syntax: **docker top <containerId/name>**

Exa: **docker top damcont**

Q: How to list the all assigned properties of the container?

Syntax**: docker inspect <containerId/name>**

Exa: **docker inspect damcont**

Q: How to see the particular assigned properties?

Syntax**: docker inspect --format <properties> <containerId/name>**

Exa: **docker inspect --format ‘{{.NetworkSettings.IPAddress}}’ damcont**

**\*\* It will show only IP address**

Exa: **docker inspect --format ‘{{.state.Running}}’ damcont**

**\*\* It will show the running status (True/False) of container**.

Q: How to run a command or install an application in a running container without attaching the container?

Syntax: **docker exec <containerId/name> <command>**

Exa: **docker exec damcont hostname**

Exa: docker exec ubuntucont apt-get install –y apache2 (to install apache without attaching the container)

* **docker exec –it <containerId/name> /bin/bash**

**Basically used to attached a daemonized container for a particular exec(executable) command and we can run any exec command. If we will exit , it will stop the particular process here it is exec /bin/bash. It won’t exit(stop) from container**.

Q: How to display the live stream of container(s) resource (CPU, RAM, Disk, IO) usage?

Syntax: **docker stats <containerId/name> <containerId/name>….**

Exa: **docker stats damcont (for single container)**

Exa: **docker stats damcont eca3ce6cf389 (for multiple container)**

Q: How to see the all live events of the docker(not container)?

Ans: **docker events**

Q: How to see the particular event of the docker?

Exa: **docker events --filter event=attach**  (for single event)

**docker events --filter event=attach --filter event=die --filter event=start** (for multiple events)  
Q: How to see all events from a particular time?

Exa: **docker events --since ‘1h**’ (from last one hour)

**docker events --since ‘30m’** (from last 30 mnt)

Q: How to see all specific events from a specific time period?

Exa: **docker events --filter event=attach --since ‘1h’**

**By default all ports of a container are private. It means it’s not accessible.**

**To access any application (web application) like Jenkins, Artifactory, Nginx, etc.. Which is installed in the container, we need’s to make the port public. That means we need’s to redirect that port to host machine’s port. It is also called as port mapping, port exposing, etc..**

Q: How to expose/map/redirect a port pf a container while creating the container?

Syntax: **docker run --name <container name> -d –p <host port>:<container port> <image>**

Exa: **docker run --name Nginx –d –p 80:80 nginx:latest**

\*\* **docker run –p <port> : it will make the specified port of container as public**

**\*\* docker run –p : it will make all ports of container as public**

**Data Volumes**

\* It is a directory in container. It is basically used to share the data in between the containers

\* It will be still accessible even if container is stopped (not running)

\* It can be shared and reused in between the containers

\* A container doesn’t have to running to share its volume

\* Change to a volume made directly

\* Changes to a volume will not be included when you update an image

\* Volume persist until no containers use them

\* Data volume can accessible from multiple containers

\* All container can share the data between each other

Q: How to create a container with data volume?

Syntax: **docker run --name <container name> -it –v <path> <image>**

Exa: **docker run –name volcont –it –v /usr/sharedvol ubuntu**

**<path> : path of the directory which you wants to share**

Q: How to map the existing data volume while creating the container?

Syntax: **docker run --name <container name> –it --volumes-from <data volume containerId/name> -- privileged=true <image>**

Exa: **docker run --name test –it --volumes-from volcont --privileged=true ubuntu**

Q: how to share the volume in between the host machine and container while creating the container?

Syntax: **docker run --name <container name> -it –v <host machine directory path>:<container directory path> <image>**

Exa: **docker run --name volshare –it –v /usr/sharevolume:/usr/sharevolume ubuntu**

Q: How to rename or assign a name to a container?

Syntax: **docker rename <containerId/name> <new name>**

Exa: **docker rename volshare sharevol**

**docker rename 8e2f97fd15f3 testcont**

**\*\* We ca rename the container name not container ID. But we can assign a name to a container ID. That means container ID is not changeable**

Q: How to list all available images in your docker?

Ans: **docker images**

Q: How to search a specified image for open source community?

Syntax: **docker search <image name>**

Exa**: docker search Jenkins**

Q: How to download image form opensource?

Syntax: **docker pull <image name>**

Exa: **docker pull mattgruter/Artifactory**

Q: How to create image form your container or using your container?

Syntax: **docker commit <containerId/name> <new image name>**

Exa: **docker commit nginx**

Exa: **docker commit Nginx my1stimage**

**\*\*<new image name> is optional, you can provide you customized image name at the time of creation or you can give after the creation using the docker tag command**.

Q: How to assign a name to you image?

Syntax: **docker tag <imageId> <name>**

Exa: **docker tag b7c257eed6e9 my2ndimage**

\*\* Creating an container using customized image

Exa: docker run –name test1 my1stimage

**Building docker image using docker file**

Create a test dir & a Dockerfile

* mkdir docker
* cd docker && vim Dockerfile

**docker file text**

FROM ubuntu

RUN apt-get install vim

**Build the image:**

Syntax: **docker build –t <image name> .**

**-t: used for tagging a name to the image**

**. : it represent the Dockerfile in same in present working directory(PWD). Means it will use the file which has name as Dockerfile in PWD.**

Exa: **docker build –t dflimage** **.**

**Note(imp): In this case, It will create a container by using the information provided in the Dockerfile, Once container created it will create an image using the container. Once image creation completed it will delete the container. This container also called as intermediate/temporary container.**

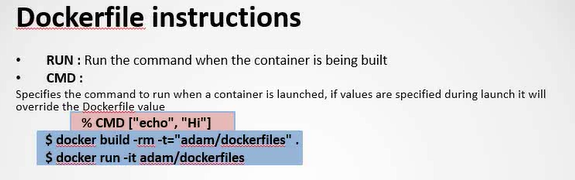
Q: How to see the history(all the layers of composing the image) of image creation.

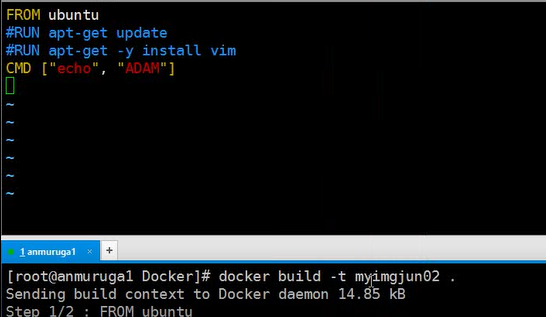
Syntax: **docker history <image name>**

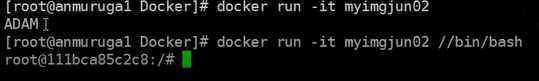
Exa: **docker history dflimage**

\*\* Creating an container using customized image

Exa: docker run –name test1 dflimage

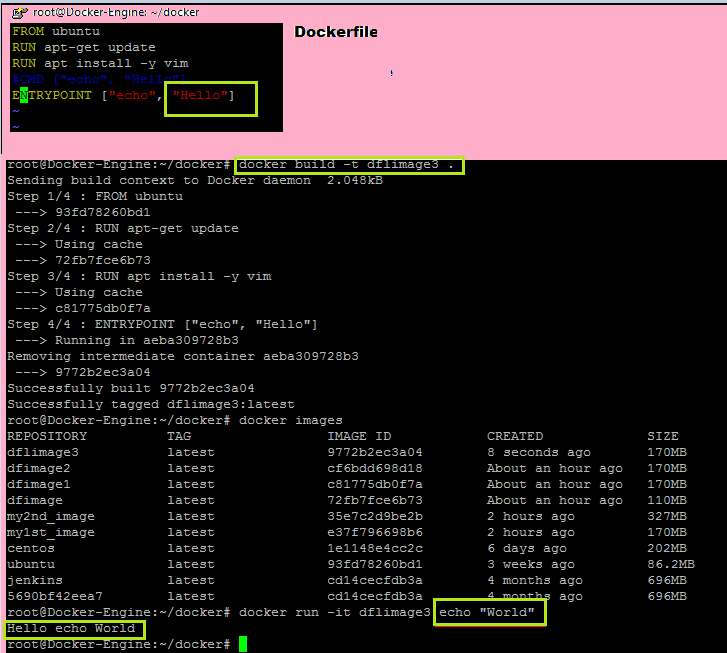


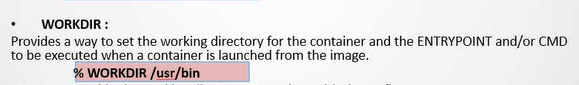




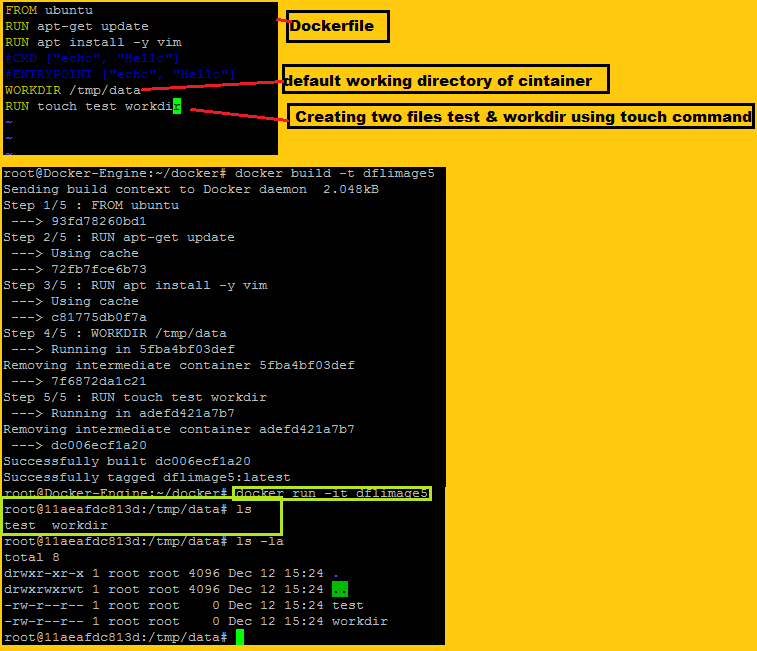
* **If we will pass the 1st command while creating the container, then the CMD(command ) provided in Dockerfile wont execute. It will be replaced by 1st command provided while creating.**



* **ENTRYPOINT is the same as CMD for the 1st command , but in CMD if we will pass any command at the time of container creation it will replace the docker file CMD, but in ENTRYPOINT it won’t replace it will append.**
* 



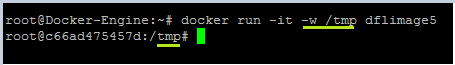
* **WORKDIR basically used to specify the default working directory of the container.**





;

Exa:



Docker rmi <imagename> : to delete the docker image

docker rmi $(docker images -q)  : to delete all docker images.

Docker stop $(docker ps –a –q): to stop all docker container

Docker rm $(docker ps –a –q) : to delete all docker container.

[What are docker engine and docker compose?](https://www.besanttechnologies.com/devops-interview-questions-and-answers)

Docker engine contacts the docker daemon inside the machine and creates the runtime environment and process for any container, docker composes links several containers to form as a stack used in creating application stacks like a LAMP, WAMP, XAMP.

LAMP(Linux. Apache, MYSQL, PHP): is collection of open source applications, which is basically used for creating web application and websites.

WAMP : Windows Apache MYSQL PHP

XAMPP : Cross-Platform (X), Apache (A), MariaDB (M) (formerly MYSQL), PHP (P) and Perl (P)

#### [Q48. How to Version control Docker images?](https://www.besanttechnologies.com/devops-interview-questions-and-answers)

Docker images can be version controlled using Tags , where you can assign tag to any image using docker tag <image-id> command. And if you are pushing any docker hub registry without tagging the default tag would be assigned which is latest , even if a image with the latest is present , it demotes that image without tag and reassign that to the latest push image.