**Building Docker Images and understanding Containers**

**RUN instruction**

* RUN is used to execute the commands during docker image build time. This will lead to creation of a new layer on top of existing layer
* for the official docs(https://docs.docker.com/engine/reference/builder/#run)
* In docker the syntax of RUN command

RUN <command>

* In docker commands can take two formats
  + shell format: You write the command as you execute on the shell
* RUN touch 1.txt
  + exec format: you write the commands in the form of ["executable", "args"]
* RUN ["touch", "1.txt" ]
* in the RUN command it is often a good practice to combine commands

RUN apt update

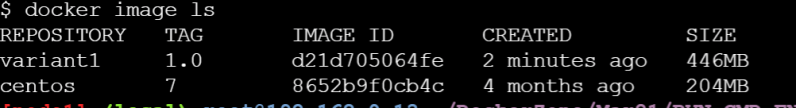
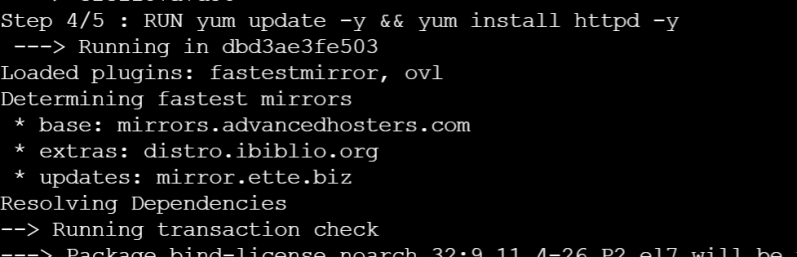
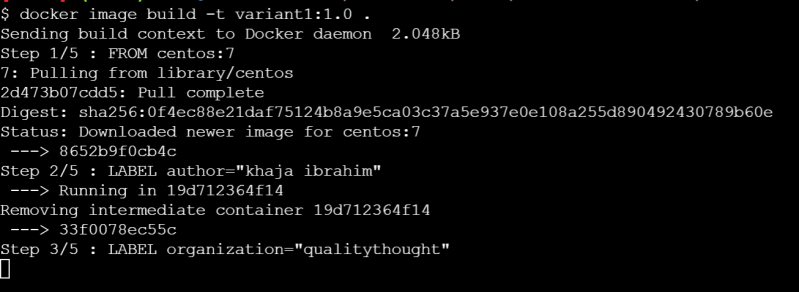
RUN apt-get install nginx -y

#good practice

RUN apt update && apt-get install nginx -y

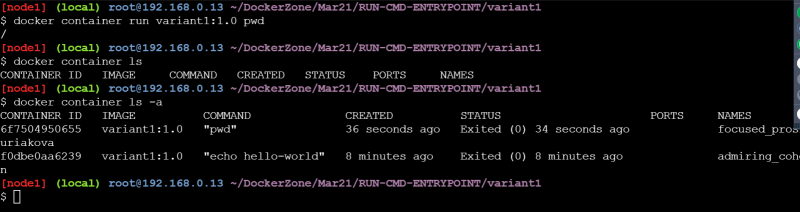
**CMD Instruction**

* CMD is used to provide the default initialization command that will be executed when a docker container is executed to create a container from Docker Image.
* CMD command also can be written in shell form and exec form
* Now lets take an simple example to install apache from the docker base image centos
  + for the first variant of the Dockerfile( Link: https://github.com/asquarezone/DockerZone/commit/97a18820545f71a4f963929c27d66bee29cddd80)
  + Now let’s try to build the docker image as we know the Docker Image is a template to create Docker containers
  + The docker image build process is initiated by Docker CLI docker image build and is executed by Docker daemon
  + To generate a Docker image Docker daemon needs to access the Dockerfile
* docker image build <context>
* docker image build -t <someimage>:<sometag> .
  + When we use . as a context all the files in the current directory will be send to docker daemon
  + We can skip sending some particular files to docker daemon by using a Dockerignore file
  + Now lets try to build docker image
* docker image build -t variant1:1.0 .



* Now lets try to create the container Preview
* Now lets try to see if our container is running 
* our container is not running, lets see the status of all the containers Preview
* Why did our container exit. Lets look at CMD
  + The container will be running as long as the command CMD is in running state. Once the command finishes execution docker daemon will assume the containers job is done and exits the container
* CMD should contain commands which will not exit till your application is alive.
* The commands which we have written in the CMD can be overriten

docker container run variant1:1.0 pwd



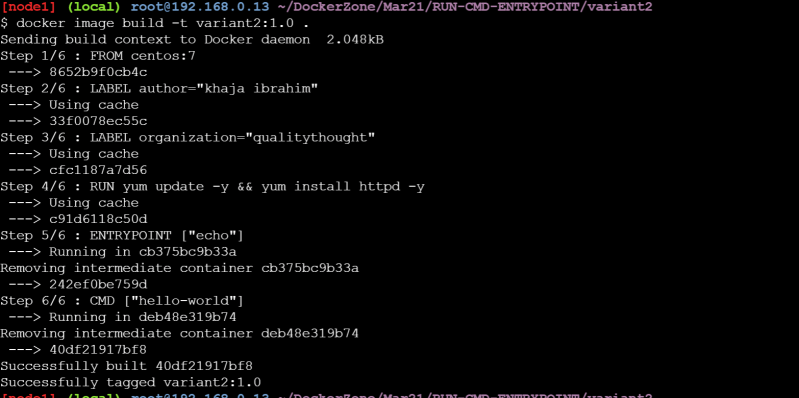
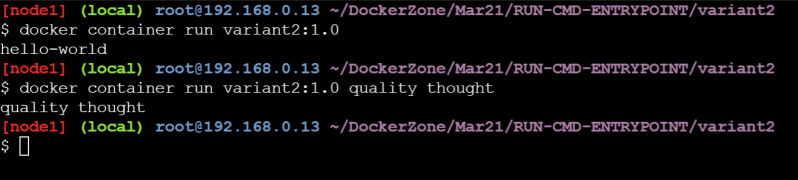
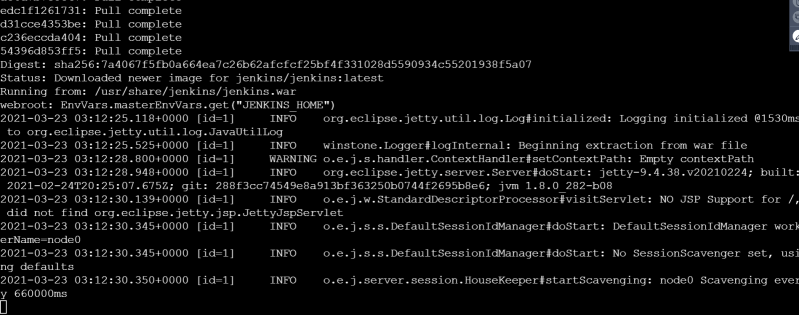
**ENTRYPOINT**

* ENTRYPOINT is also used to provide the default initialization command when the container is executed
* ENTRYPOINT command cannot be overwritten by users
* There are cases where Both ENTRYPOINT and CMD are defined in a Dockerfile

ENTRYPOINT ["echo", "quality"]

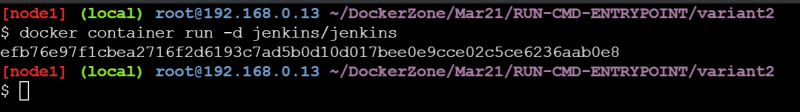
CMD ["thought"]

=> ["echo", "quality" "thought"]

* Lets build the variant2 image 
* Now lets run the container 
* When we run the container the container by default will be in attached mode i.e. it will redirect stdout to our terminal 
* In reality we would not want to run our containers in the foreground/attached mode rather we want to run containers in the background /detached mode

docker container run -d jenkins/jenkins

# This command will just output the container id

\*