**Pushed images into docker hub**

**Step-1**

* **Go to google sign in aws console**
* Sign in already existing root user
* Click on ec2 service
* Click on instances
* Start the docker instance
* Connect
* Click on ssh client
* Go to file explorer
* Click on downloads
* Open .pem key in downloads folder (docker instance key)
* Right click
* click on Connect git bash here
* Copy third and example commands and paste here.
* Success fully connected linux server.

**Step-2**

* **Execute some commands**
* Execute sudo su
* Install docker (yum install –y docker) if already docker installed check the version of docker.
* Docker –version
* Ifconfig –a
* To start the docker service (service docker start)
* Docker info (it will give information of docker)
* Go to docker user (cd /var/lib/docker)
* Ls
* Execute docker images or docker image ls –it shows all available images means created images.
* Docker ps –a (it shows list of containers)

**Step –3**

* **Docker hub**
* First, we will create repository in docker hub.
* Sign in docker hub account
* Click on create repository
* Give name (environment name Dev,QA,Prod)
* Click on create
* Repository will be created.

**Step –4**

* **Python image:**
* Docker pull python (it downloads information about python)
* Docker images or docker image –ls (it shows image created or not) (image tag,image id, image size.)
* Docker inspect <image id> (it shows information about python image)
* docker run -itd --name sribhavya -p 3000:80 <image id> (create container with port)
* Docker ps (it shows container created or not and CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAME it shows)
* Docker login (give dockerhub username and password)
* docker tag python:latest bhavya503/dev:jenkins2(to create tag)
* docker push bhavya503/dev:jenkins2.
* **Nginx image:**
* docker pull nginx
* docker images
* docker inspect <image id>
* docker run -itd --name bhavya -p 40:80 <image id>
* docker ps –a
* docker login
* docker tag nginx:latest bhavya503/dev:jenkins
* docker push bhavya503/dev:jenkins
* **Mysql image**:
* docker pull mysql
* docker images
* docker inspect <image id>
* docker run -itd --name mysql1 -p 3306:80 <image id>
* docker ps –a
* docker login
* docker tag mysql:latest bhavya503/dev:jenkins4
* docker push bhavya503/dev:jenkins4
* The push refers to repository [docker.io/bhavya503/dev]
* 55e9644f21c3: Layer already exists
* 7d22e2347c12: Layer already exists
* f6d5815f290e: Layer already exists
* 791f0a07985c: Layer already exists
* cabea05c000e: Layer already exists
* c68632c455ae: Layer already exists
* 5f1ee22ffb5e: Layer already exists
* jenkins: digest: sha256:8ff5f063bf8cc99e473d552e4cc34fe8d37f10563d258eef9e2aa71b6455ab06 size: 1778
* **Ruby image:**
* docker pull ruby
* docker image ls
* docker inspect <image id>
* docker run -itd --name sribhavya -p 3000:80 <image id>
* docker ps -a
* docker login
* docker tag ruby:latest bhavya503/dev:jenkins3
* docker push bhavya503/dev:jenkins3
* **Gradle image:**
* docker pull gradle
* docker images
* docker inspect <image id>
* docker run -itd --name sri 70:80 <image id>
* docker ps -a
* docker login
* docker tag gradle:latest bhavya503/dev:jenkins4
* docker push bhavya503/dev:jenkins4.
* **In docker hub i created QA and PROD environments.**
* Sign in docker hub account
* Click on create repository
* Give name (environment name QA,Prod)
* Click on create
* Repository will be created.
* **Httpd server image:**
* docker pull httpd
* docker images
* docker inspect <image id>
* docker run -itd --name sri 80:80 <image id>
* docker ps -a
* docker login
* docker tag httpd:latest bhavya503/dev:jenkins1
* docker push bhavya503/dev:jenkins1.
* Remove containers and create new containers:
* First stop the container (docker stop <container name or container id> then after remove the container by using below command.
* Docker rm <container name>
* Remove images
* Docker rmi <image name>
* Docker rmi <image id>.
* After completion of container creation, it was automatically exited.
* To remove the exited state using below command
* docker run --name mysql -d -e MYSQL\_ROOT\_PASSWORD=password mysql:latest.