Rajashekhar Reddy---10-Sep-2024---**Docker-02**

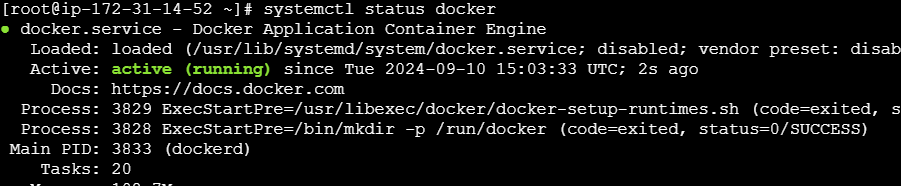
1. **Create a tomcat container on 8080 and deploy sample application in tomcat.**

\* First we need to check whether docker is running or not

systemctl status docker

\* If it is not in running state the run this docker

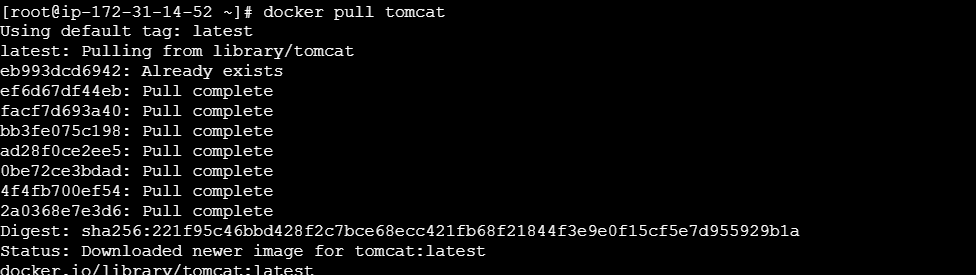
\* systemctl start docker



\* Then download the tomcat using docker

**docker pull tomcat**

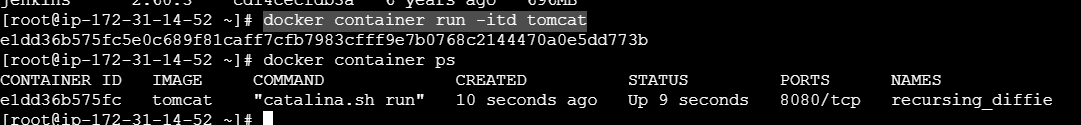
**\*** whch are already available in the machine



\* Now am using below command to create and run the tomcat container

**docker container run -itd tomcat**

\* To check these container is running and on which port it’s working **docker container ps**



\* Now I want to enter into the tomcat container using below command

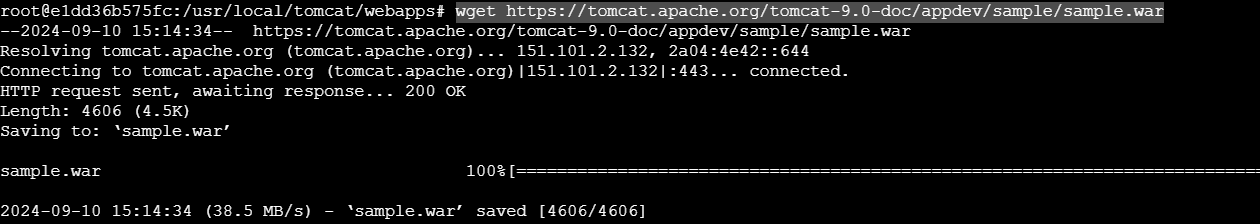
**docker exec -it e1dd36b575fc /bin/bash**



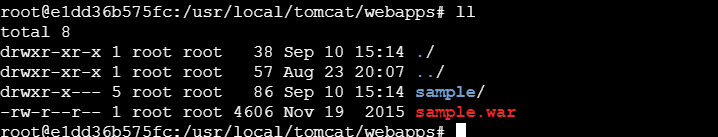
\* Then go to webapps for deploying the sample application

Download the sample war file using

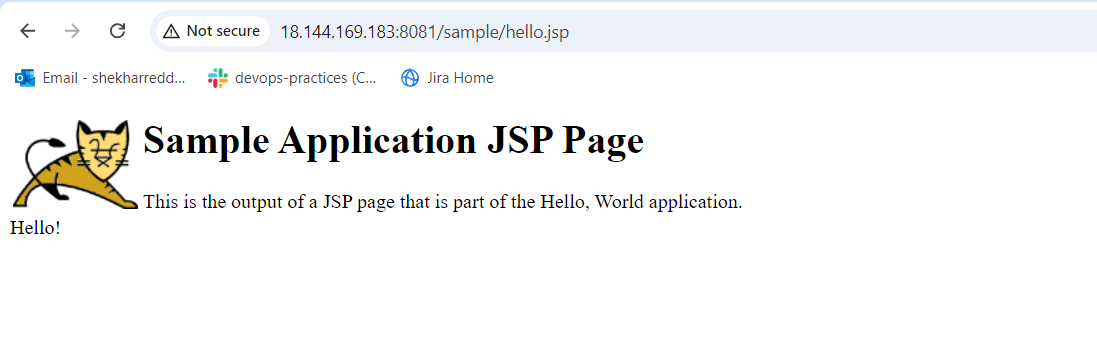
**wget https://tomcat.apache.org/tomcat-9.0-doc/appdev/sample/sample.war**



\* Then we can find the sample war file



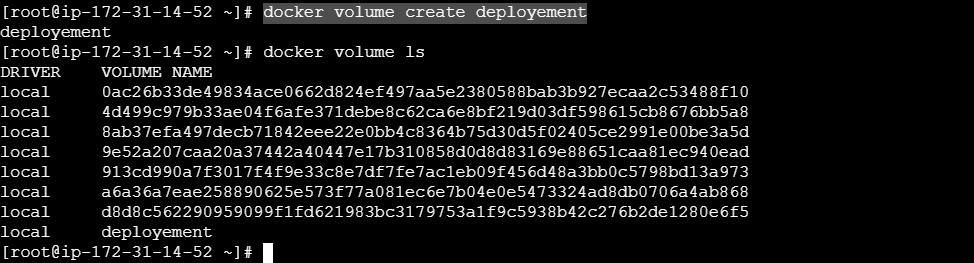
\* deploy the sample application using **18.144.169.183:8081/sample**



1. **Create volume and deploy tomcat container on port 8081.**

\* I created a new volume using docker

**docker volume create deployement**



\* to check the path of the volumes

\* Go to **cd /var/lib/docker/volumes/**

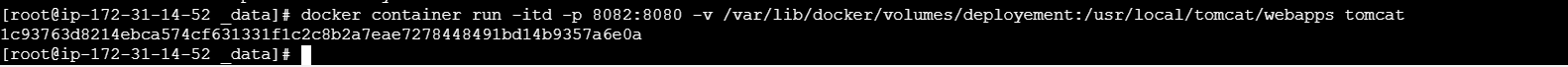
\* in this volumes we can fine the deployment volume, then go to \_data folder here we need to download the sample war file



\* Here in this folder downloaded the sample war file

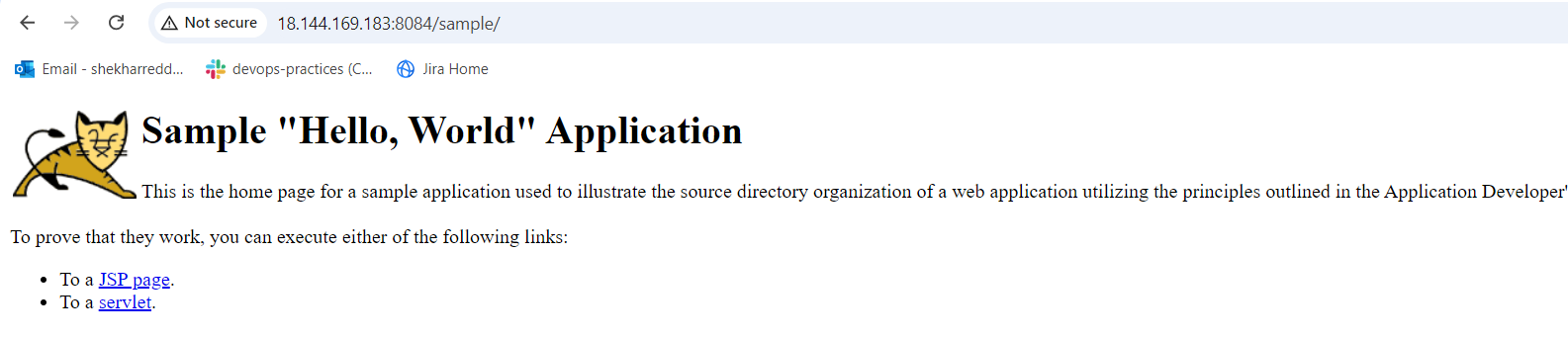
\* Using below command to mapping the deployment volume to the existing tomcat:latest

**docker container run -itd -p 8082:8080 -v /var/lib/docker/volumes/deployement:/usr/local/tomcat/webapps tomcat**



\*now we can see the files which are available in the existing volume in the present deployement volume

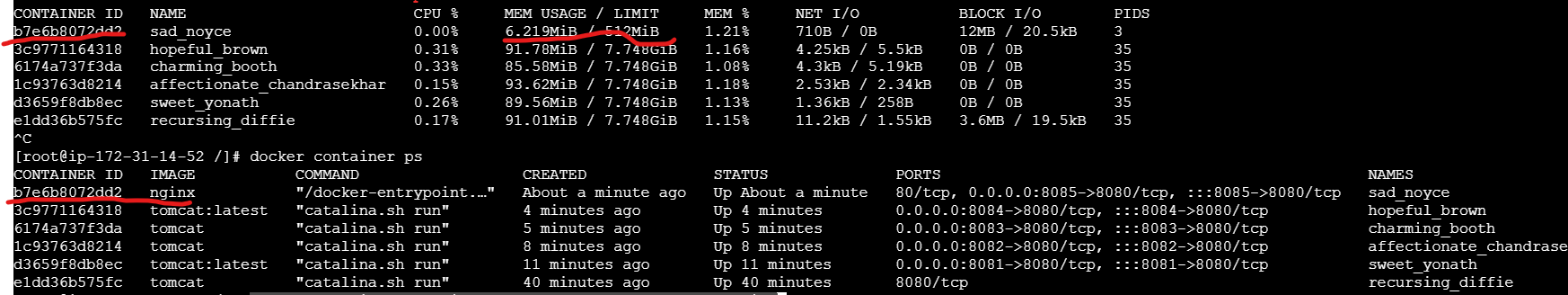
\* now deploy the tomcat application



1. **Limit the nginx container to 500 MB.**

\* Using below command to limit the nginx container

**docker container run -itd -p 8085:8080 --memory=512m nginx**



**4) Create a sample docker file using below instructions.**

**1) Base module as amazonlinux:latest**

**2) Maintainer you name**

**3) Install nginx**

**4) COPY one index.html file to image**

**5) EXpose on port 80**

**6) Command to start the nginx container**

\* First created a **Dockerfile** and adding above instructions

**# Use the latest Amazon Linux image as the base image**

**FROM amazonlinux:latest**

**# Label the image with the maintainer's name**

**MAINTAINER SHEKHAR**

**# Install Nginx and other necessary packages**

**RUN yum update -y && \**

**yum install -y nginx && \**

**yum clean all**

**# Copy the index.html file to the appropriate directory in the container**

**COPY index.html /usr/share/nginx/html/index.html**

**# Expose port 80 for Nginx**

**EXPOSE 80**

**# Command to start Nginx**

**CMD ["nginx", "-g", "daemon off;"]**

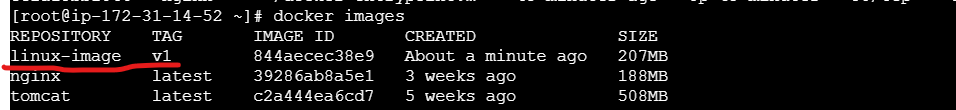
**\*** After adding the instructions build the image using below command

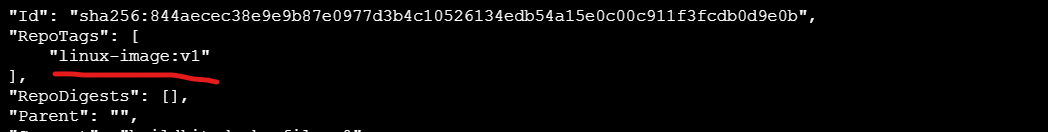
**docker build -t linux-image:v1 .**

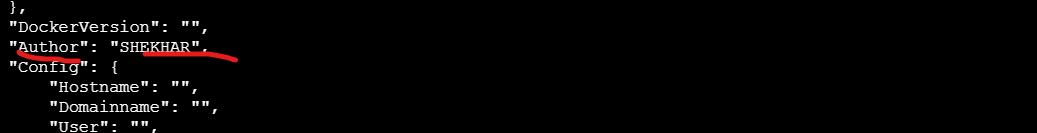
Then it will successfully executed

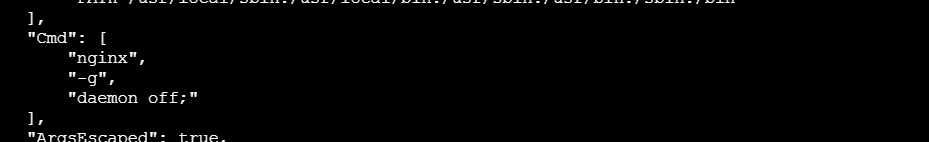


\* Then we can find the created image using **docker images**





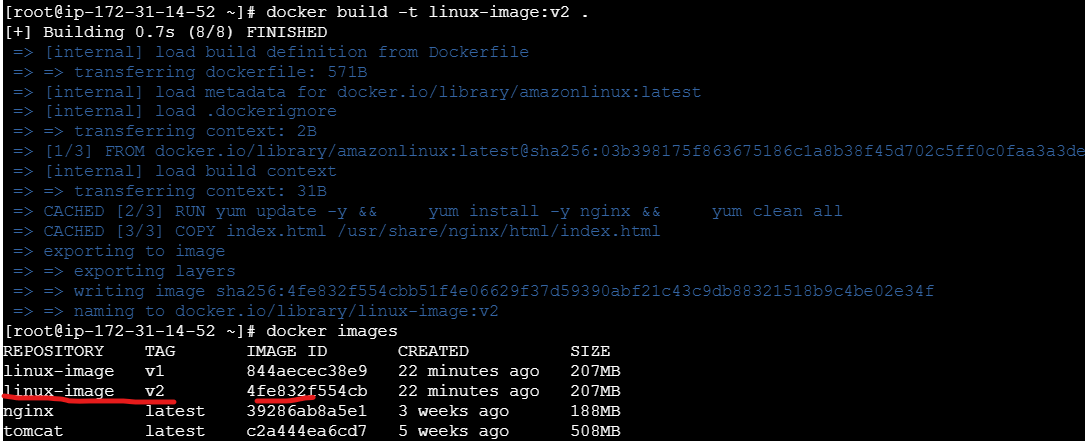




\* Then I tried to change the tag name and build the docker image then it will create another image with tag name

**docker build -t linux-image:v2 .**

**\*** It will create another image using cache , which is help us to increase the performance

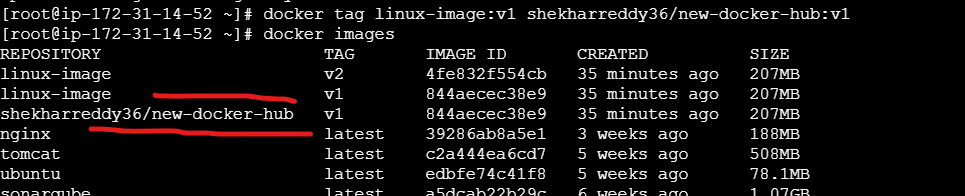


**5)Push image to dockerhub**

\* First I created a dockerhub account and create a repository

\* Then using the username and repository name created the copy of image

**docker tag linux-image:v1 shekharreddy36/new-docker-hub:v1**

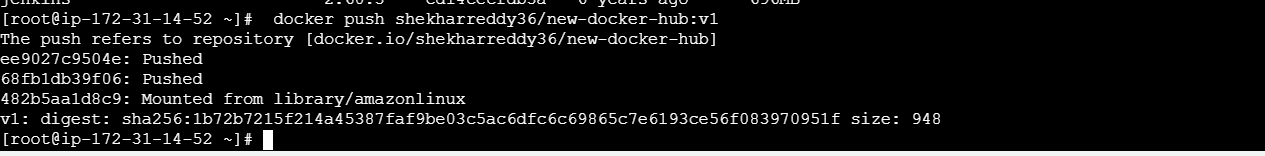


\* Then I want to push this image into the docker hub repository ,

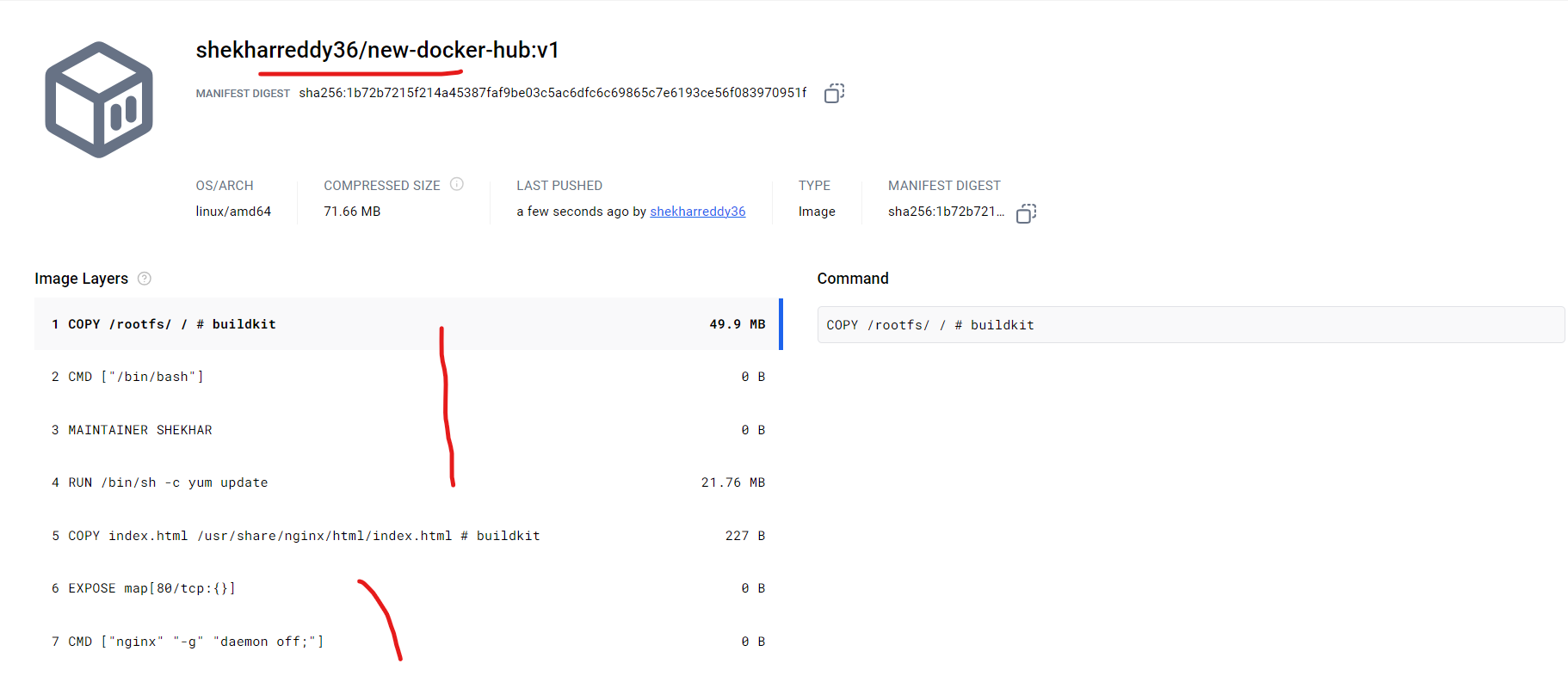
\* First I need to login with username and password

\* Then push the image using below command

**docker push shekharreddy36/new-docker-hub:v1**

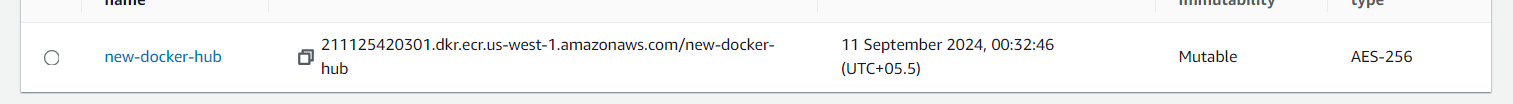


**\*** we can find the files in the docker hub repository which are available in this image



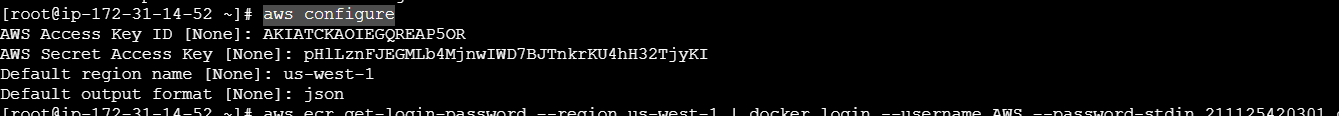
1. **push image to aws ecr**

\* First I createda new repository in ECR(Elastic containr registry)



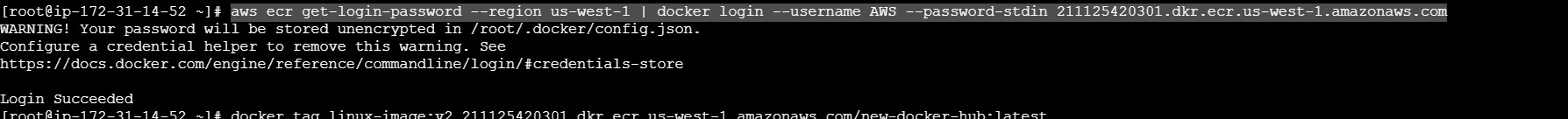
\* Then I have configure with global credentials for connecting via cli

**aws configure**



**\*** we need to authenticate Docker to ECR registry to be able to push images

**aws ecr get-login-password --region us-west-1 | docker login --username AWS --password-stdin 211125420301.dkr.ecr.us-west-1.amazonaws.com**



**\*** we need to tag Docker image with the ECR repository URL

**docker tag linux-image:v2 211125420301.dkr.ecr.us-west-1.amazonaws.com/[new-docker-hub](https://us-west-1.console.aws.amazon.com/ecr/repositories/private/211125420301/new-docker-hub?region=us-west-1):latest**



\*Now we can push the tagged Docker image to my ECR repository.

docker push 211125420301.dkr.ecr.us-west-1.amazonaws.com/new-docker-hub:latest



\* Now we can find the image in the ecr repository

