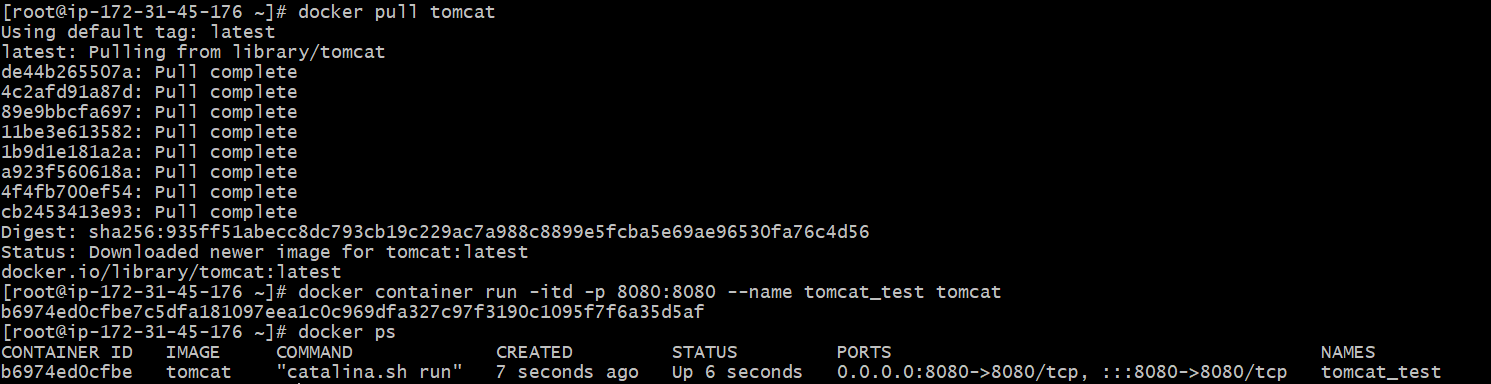
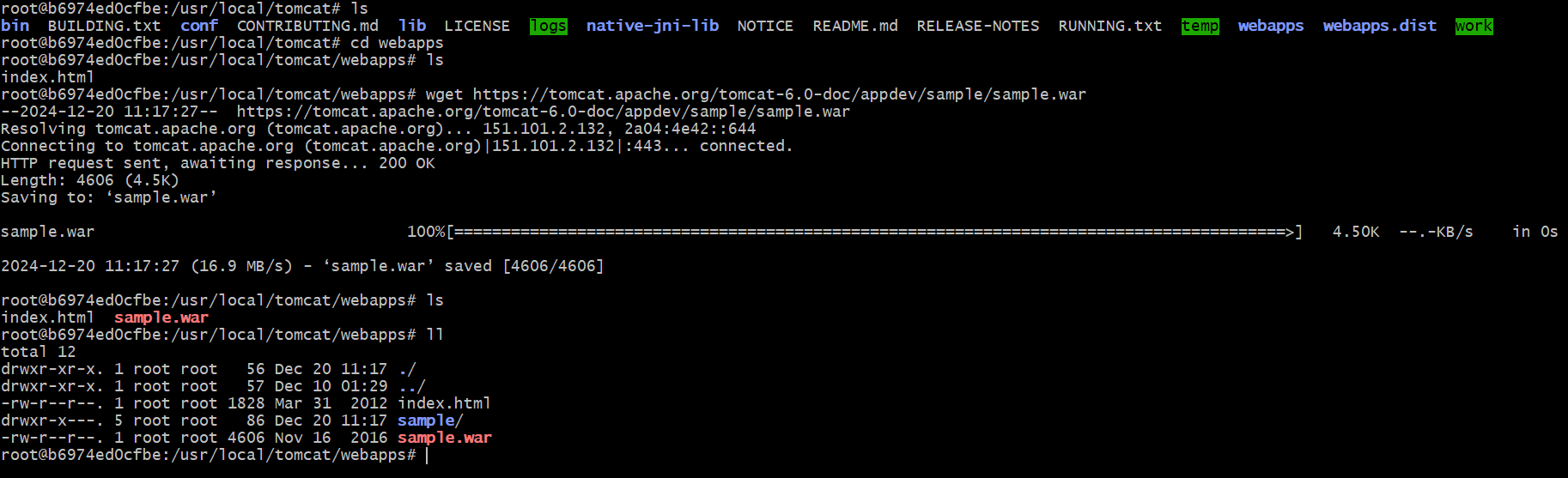
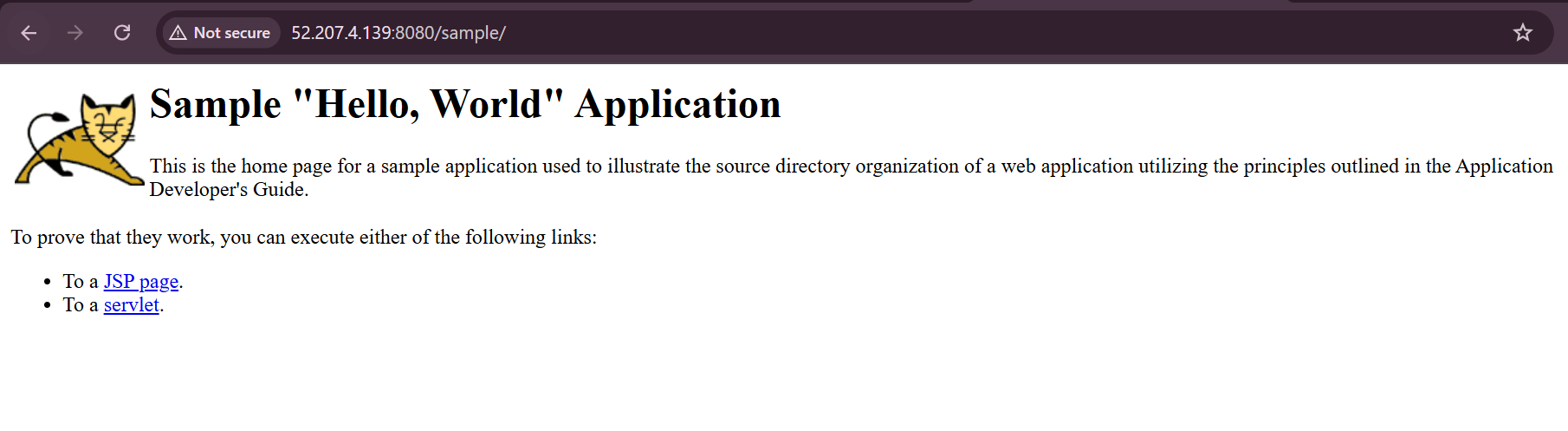
DOCKER[TASK-2]

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





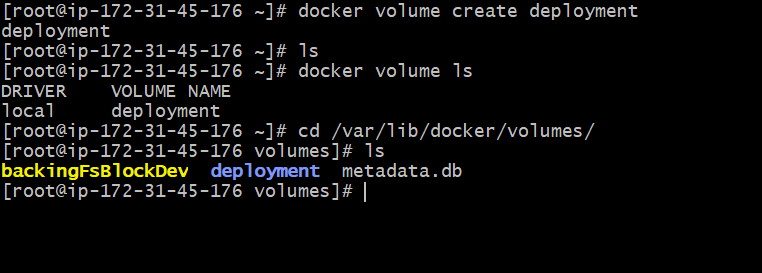


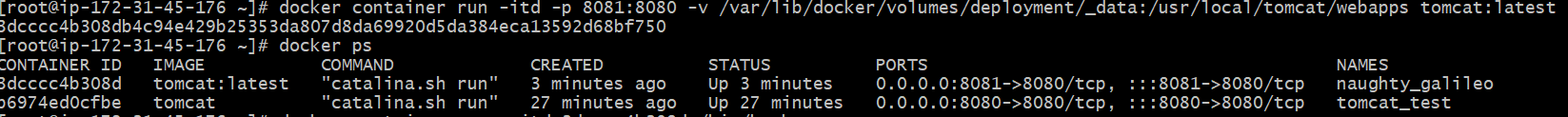


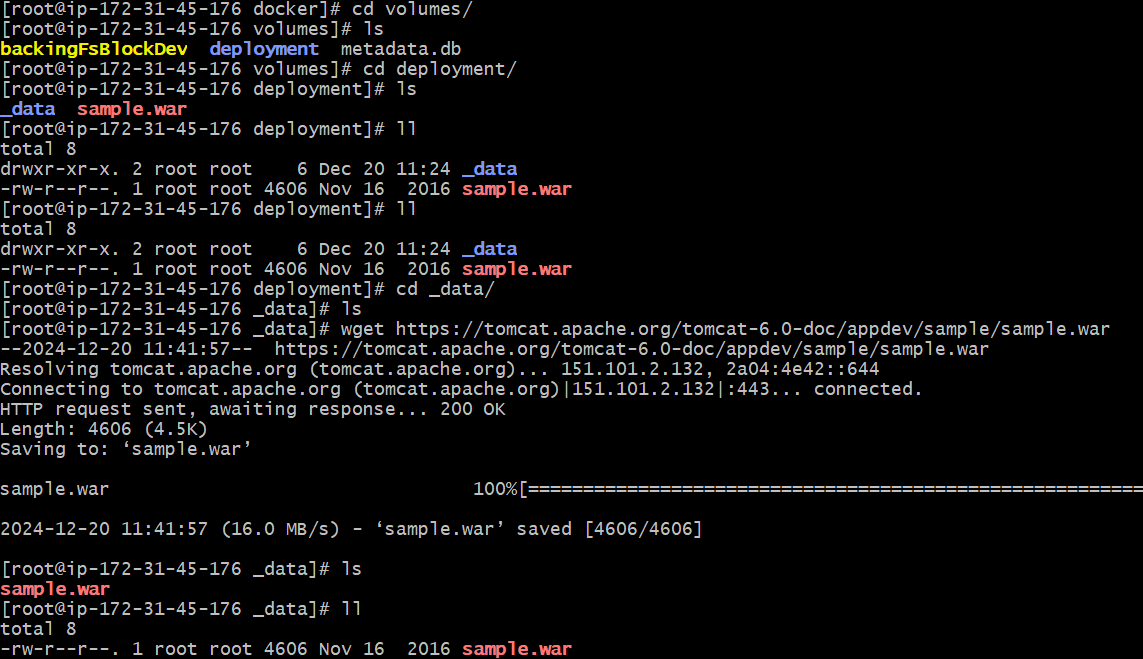
2) Create volume and deploy tomcat container on port 8081.

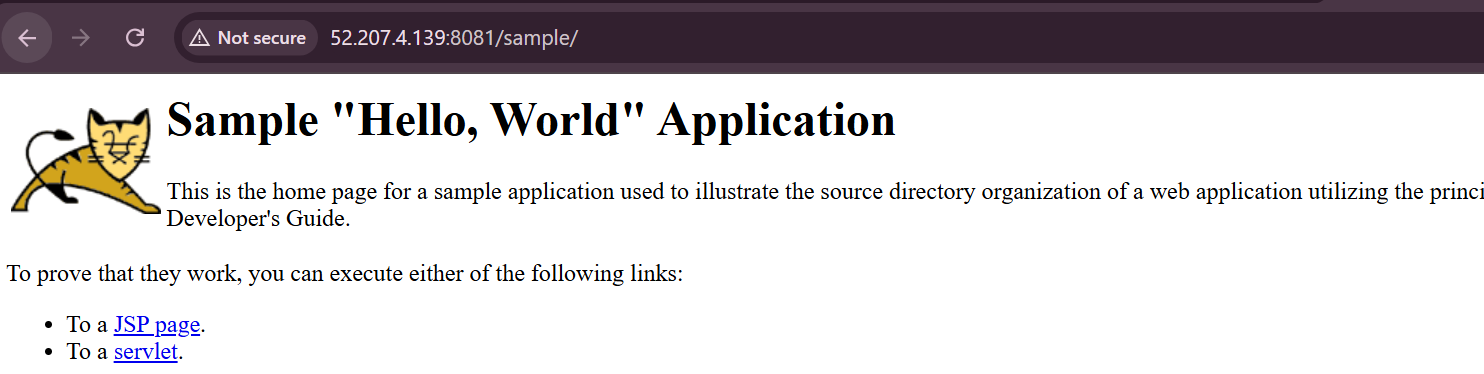
Created volume with a name “deployment”

Cmd: *docker volume create deployment*

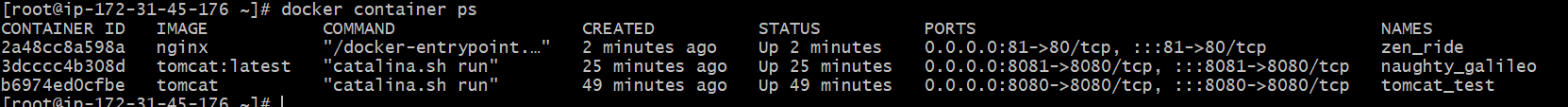




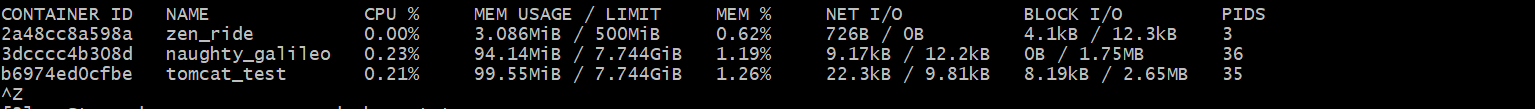




3) Limit the nginx container to 500 MB.

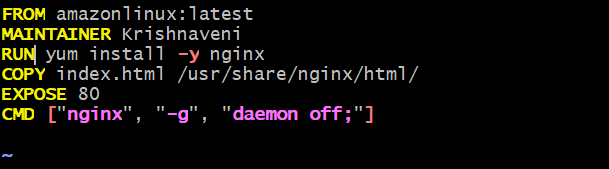


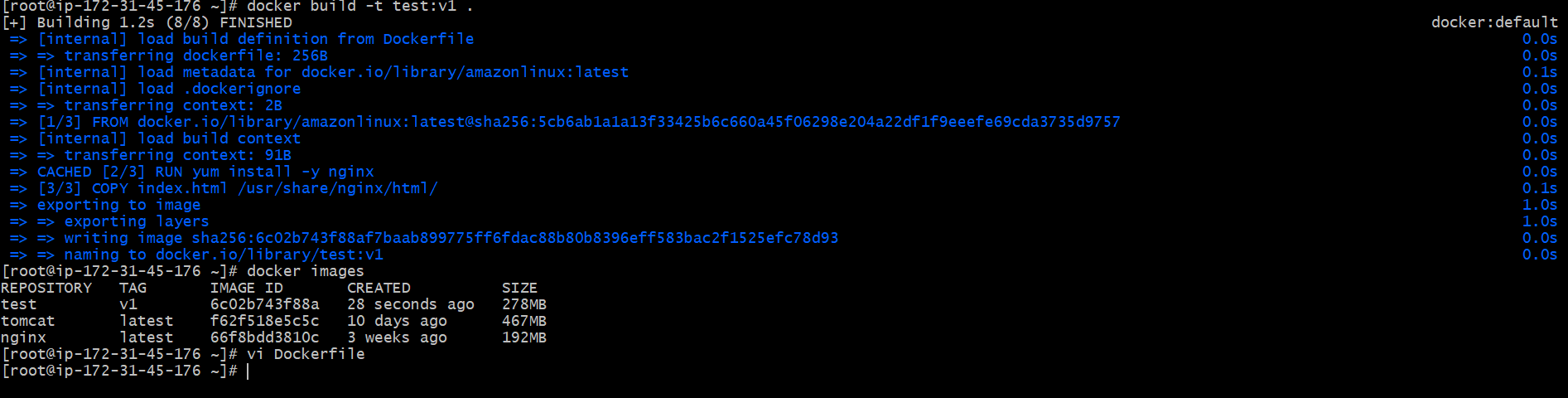
*Cmd:docker container run –itd –p 81:80 –memory=500mb nginx*



4) Create a sample docker file using below instructions.

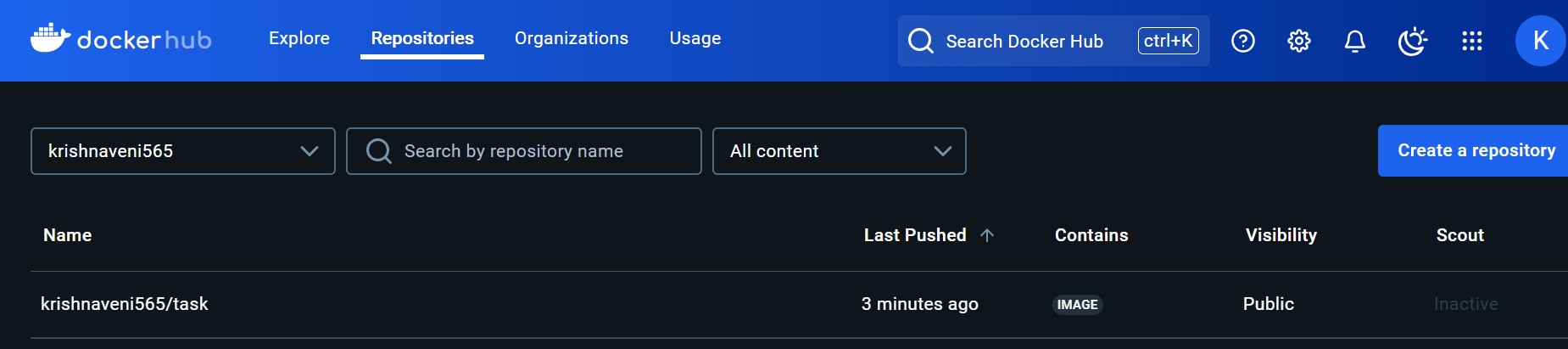
* Base module as amazonlinux:latest
* Maintainer you name
* Install nginx
* COPY one index.html file to image
* 5) Expose on port 80
* 6) Command to start the nginx container

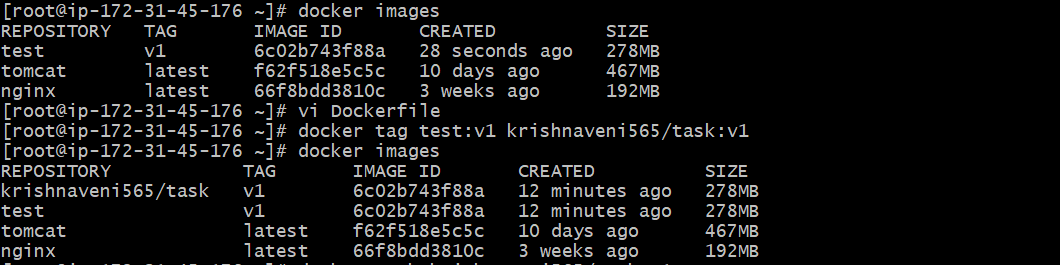


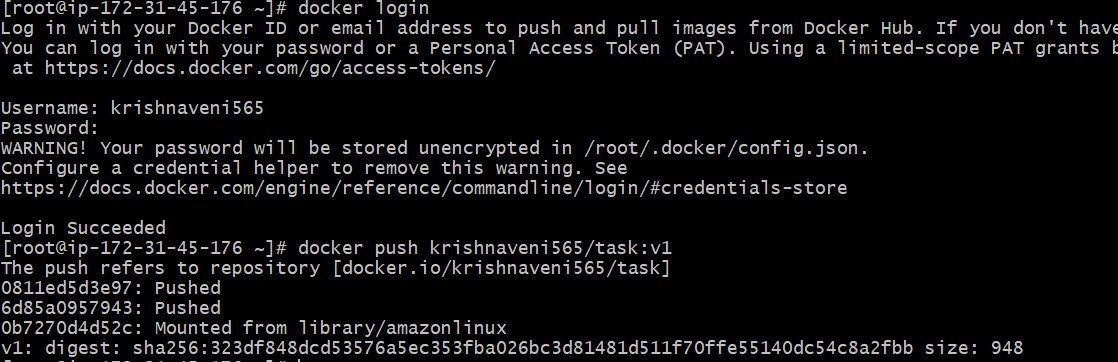


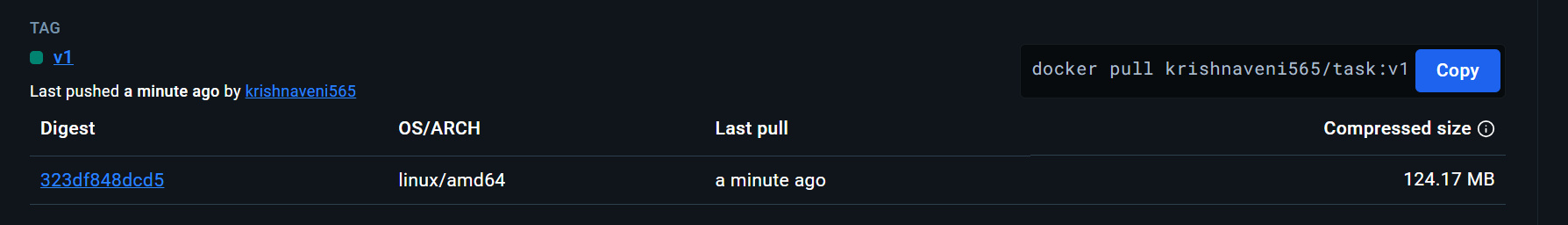
5) Push image to dockerhub

Created one repo in docker hub





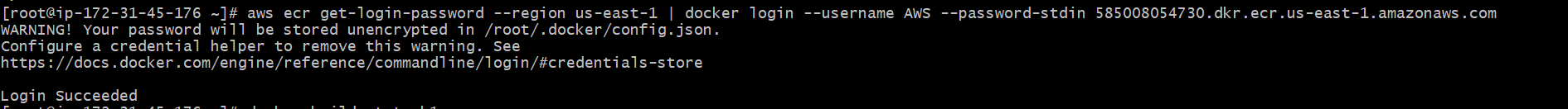




6) push image to aws ecr

Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

*aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 585008054730.dkr.ecr.us-east-1.amazonaws.com*



*docker build -t task1* .



After the build completes, tag your image so you can push the image to this repository:

*docker tag task1:latest 585008054730.dkr.ecr.us-east-1.amazonaws.com/task1:late*

Run the following command to push this image to your newly created AWS repository:

*docker push 585008054730.dkr.ecr.us-east-1.amazonaws.com/task1:latest*

