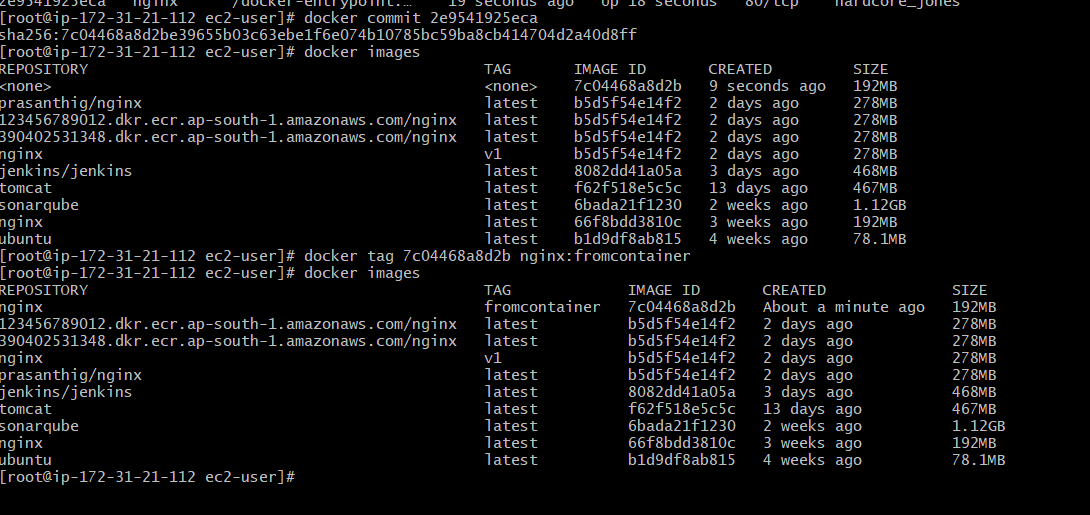
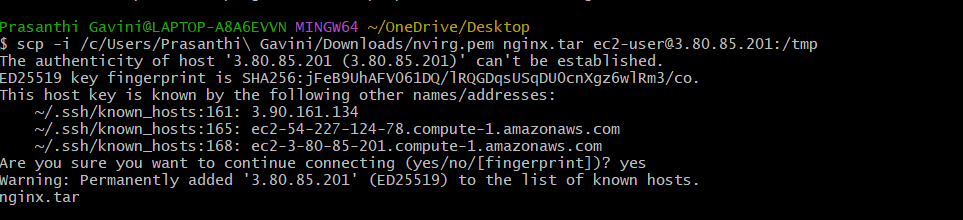
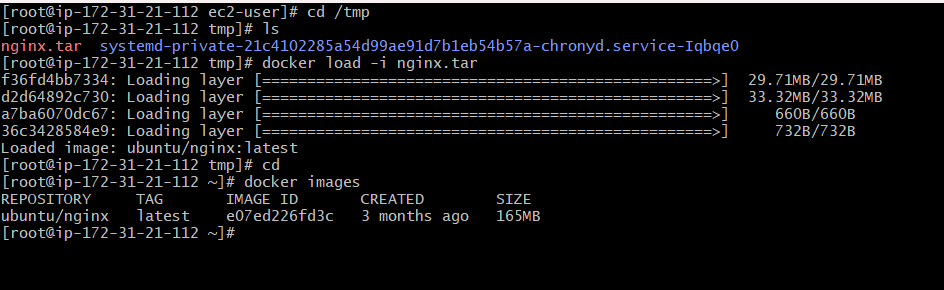
1) Create a image from running container.



2) Copy image from local machine to docker server and load the image.





3) Create Docker image using alpine and customize with tomcat.

* Pull Alpine image

docker pull alpine

* Run the container using alpine image

docker container run -itd alpine

* Go into the container

docker exec -it 869250cfc4de /bin/sh

* Install tomcat

Steps to install tomcat

apk update

apk add --no-cache openjdk11 curl

curl [https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.98/bin/apache-tomcat-9.0.98.tar.gz -o tomcat.tar](https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.98/bin/apache-tomcat-9.0.98.tar.gz%20-o%20tomcat.tar)

mkdir -p /usr/local/tomcat

tar -xvf tomcat.tar

mv tomcat /usr/local/tomcat

export CATALINA\_HOME=/usr/local/tomcat

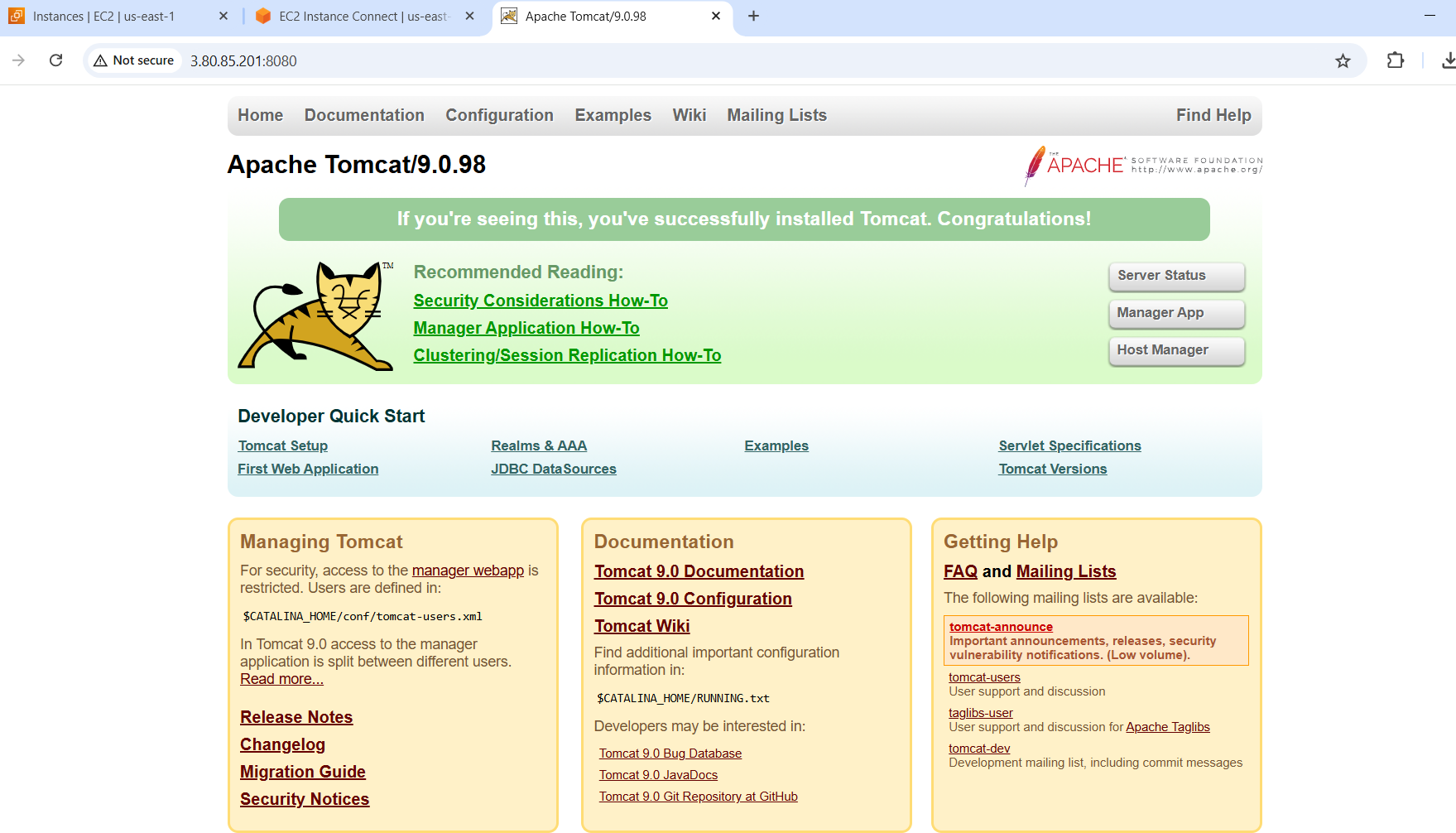
export PATH=$CATALINA\_HOME/bin:$PATH

* Create an image from the running container

docker commit 869250cfc4de alpine-tomcat

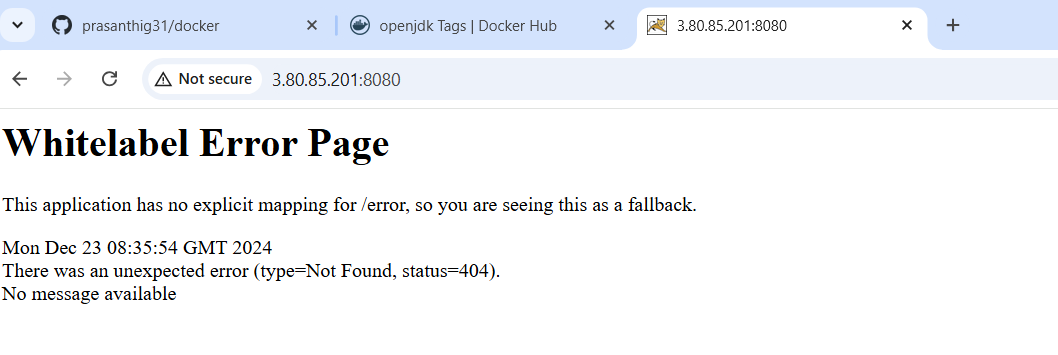
* Run the container

docker container run -itd -p 8080:8080 alpine-tomcat /usr/local/tomcat/bin/Catalina.sh run

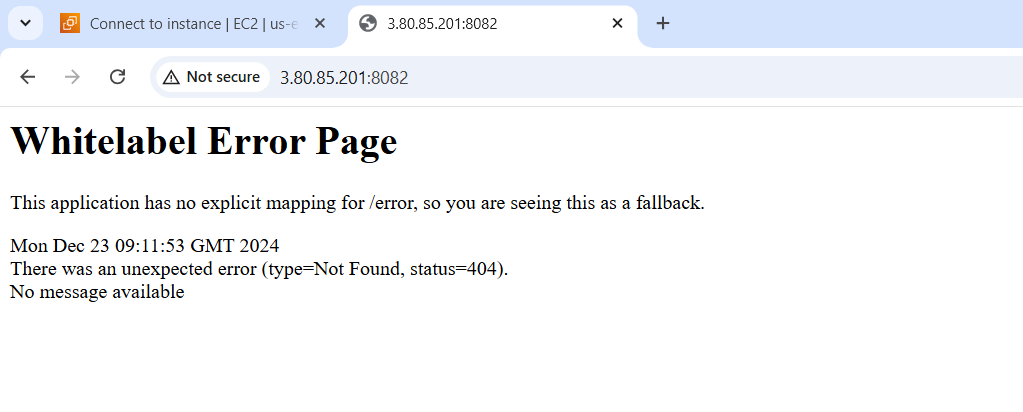


4) Create single stage and multi stage docker file using the below source code.  
   <https://github.com/betawins/multi-stage-example.git>

Single stage docker file

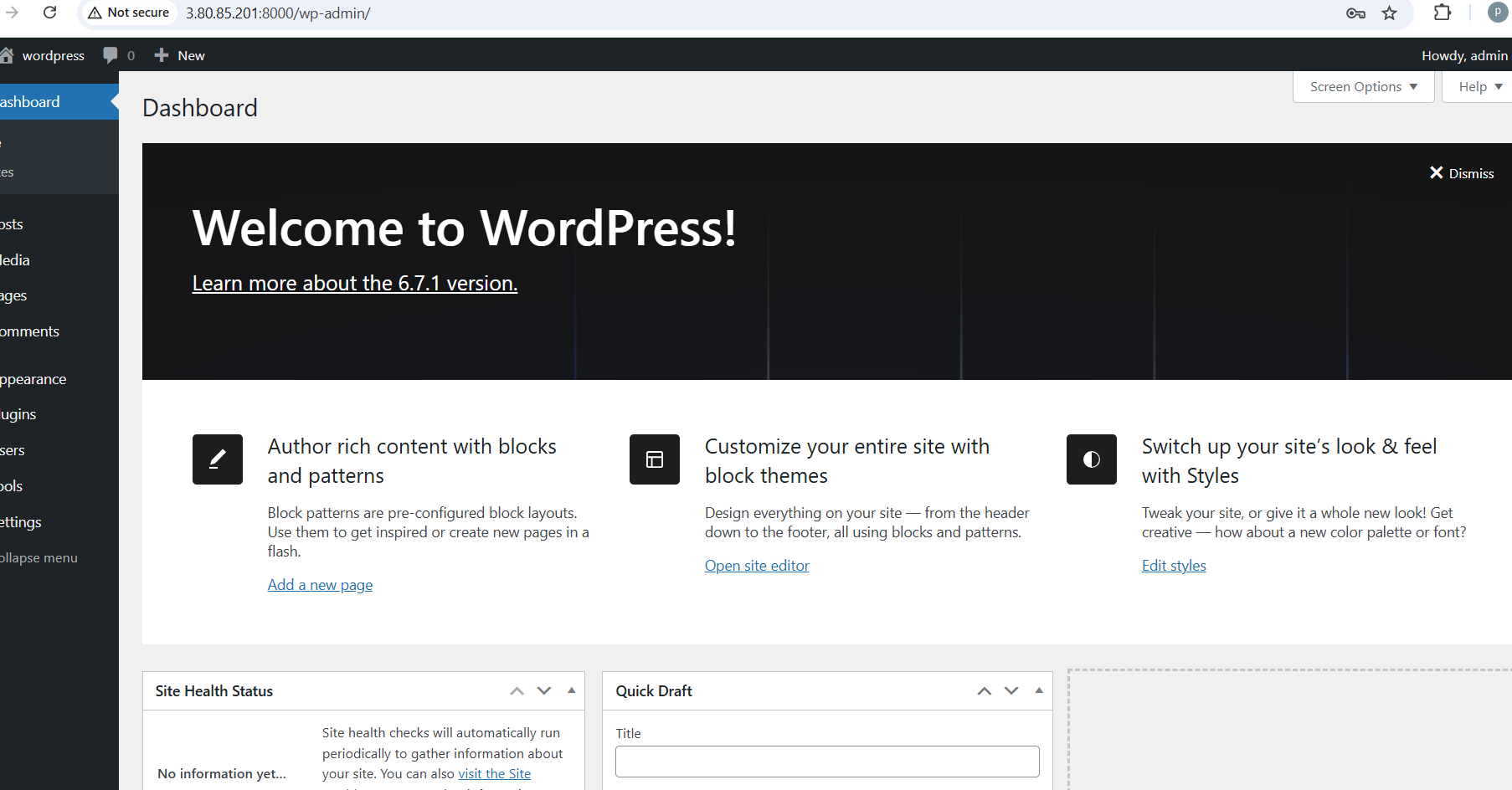


Multistage docker file

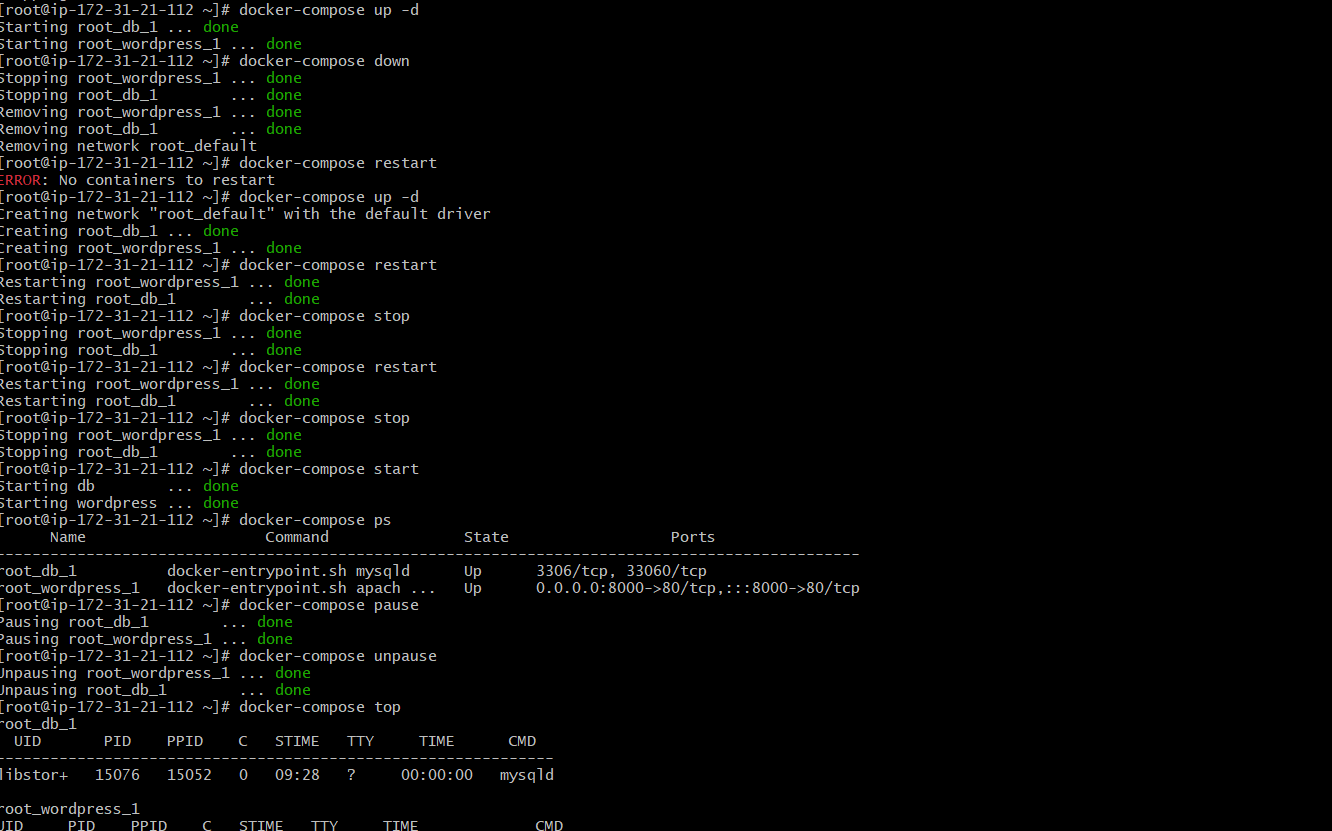


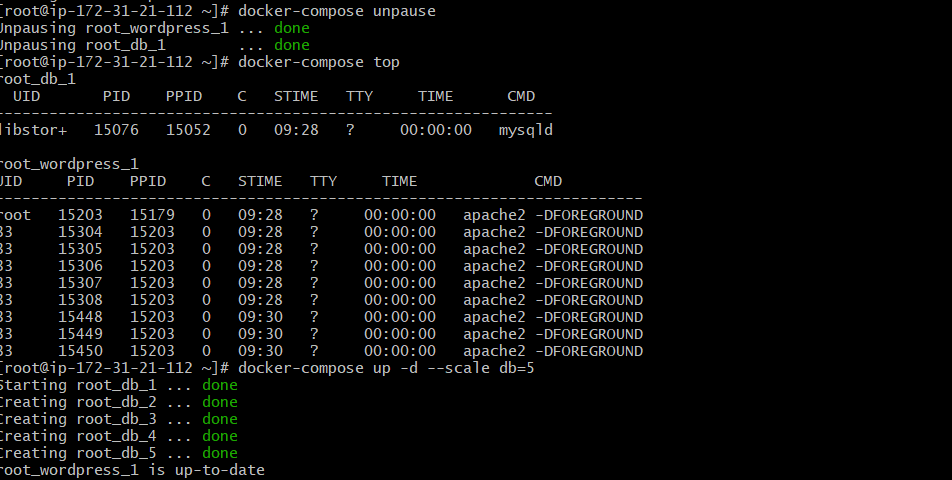
5) Install docker compose and execute sample applciation.

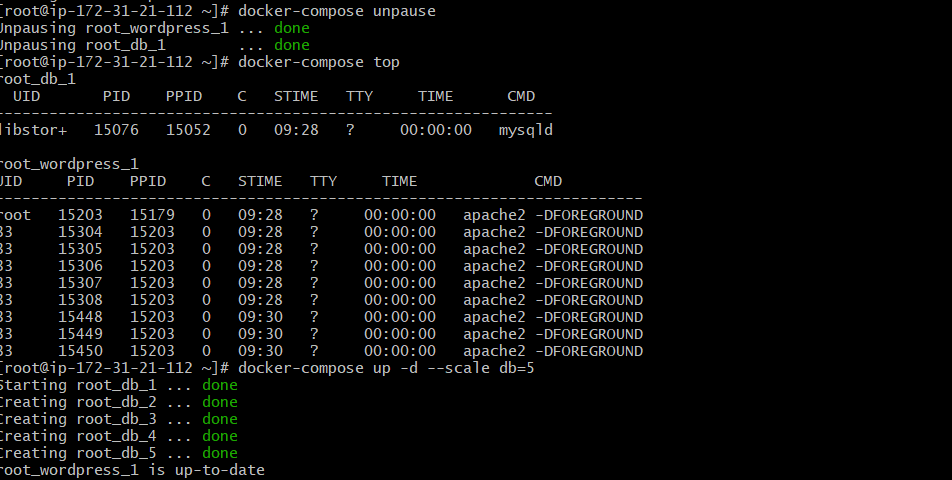
Docker-compose up



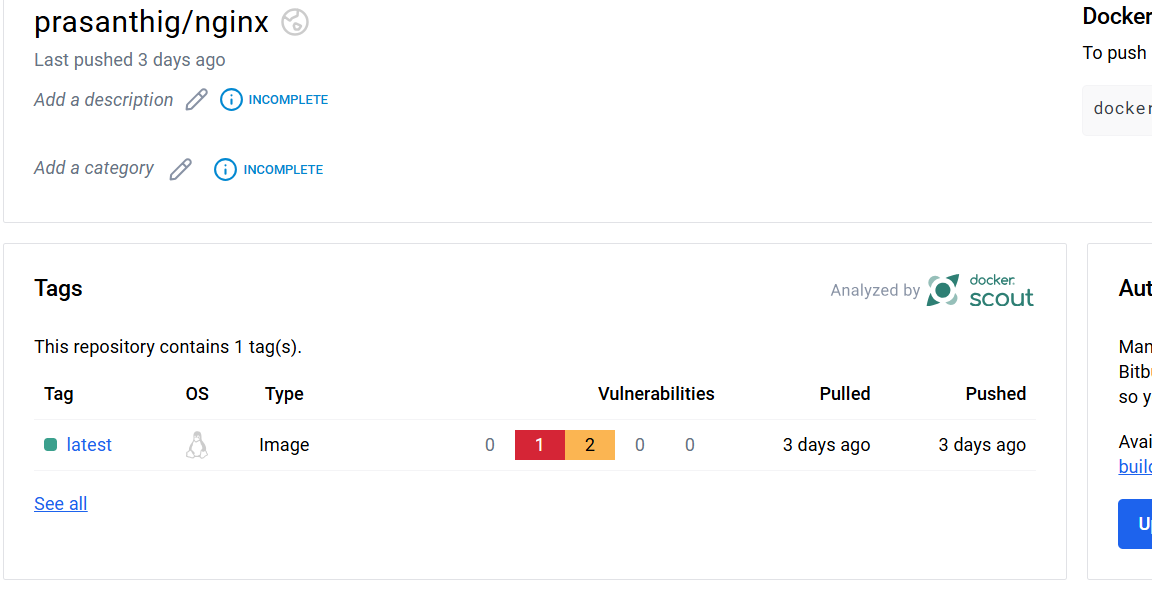
Docker-compose down

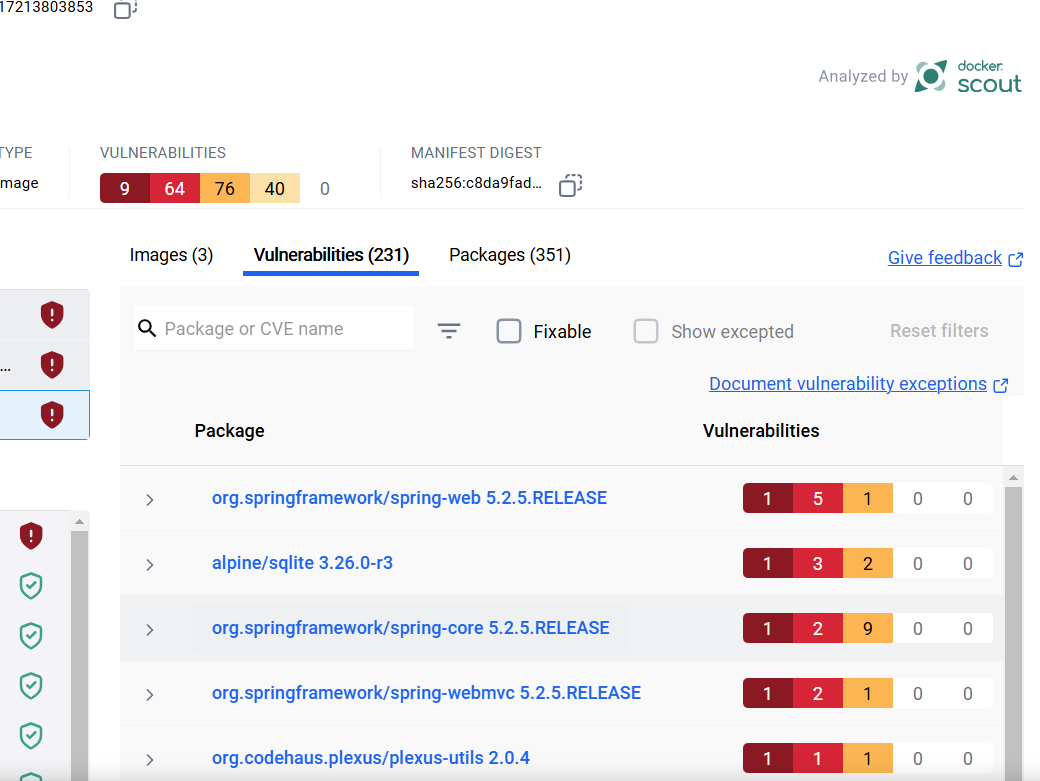






6) Implement solution to scan images when pushed to docker registry.



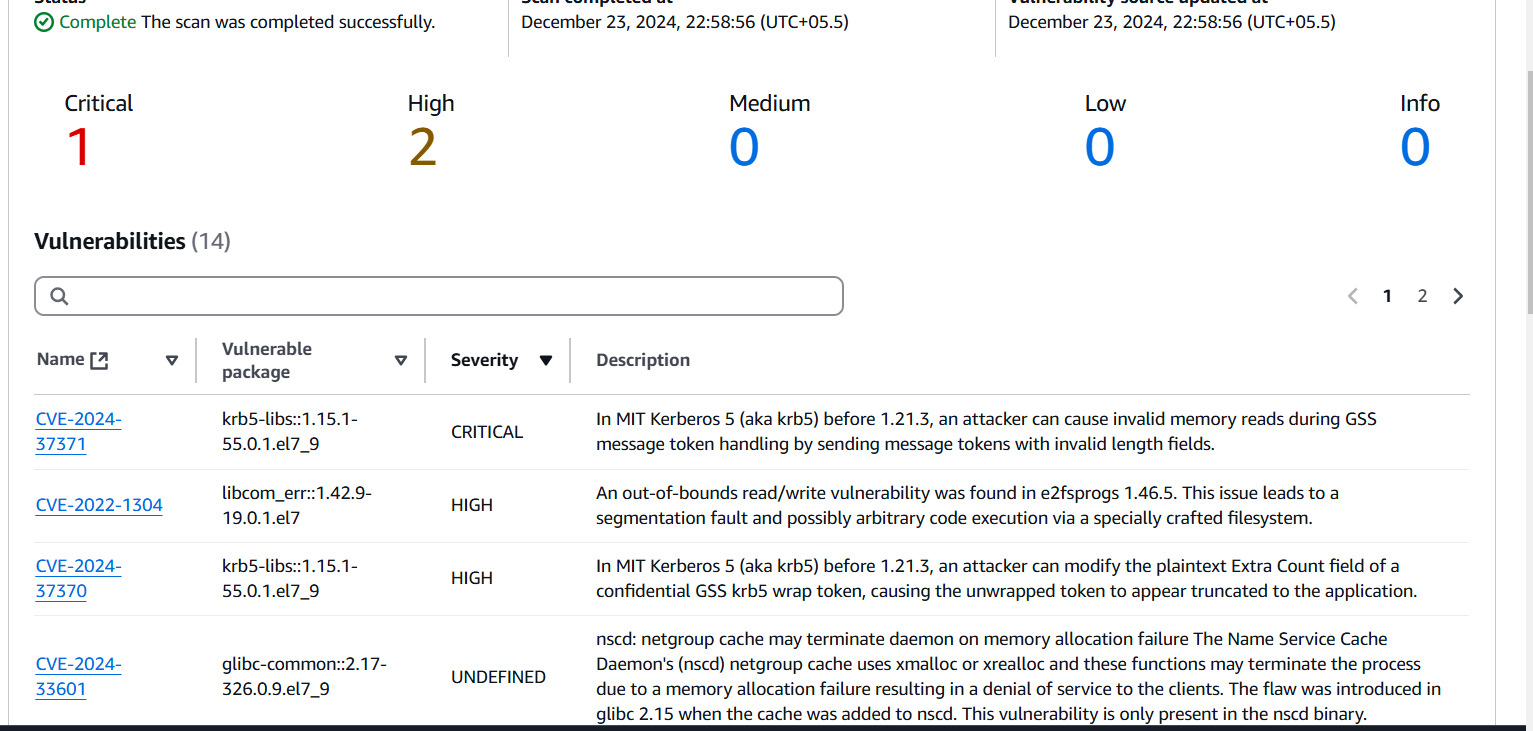


7) Implement solution to scan images when pushed to aws ecr.

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 390402531348.dkr.ecr.us-east-1.amazonaws.com

docker tag mysql:5.7 390402531348.dkr.ecr.us-east-1.amazonaws.com/mysql:latest

docker push 390402531348.dkr.ecr.us-east-1.amazonaws.com/mysql:latest



8) Create a jenkins pipeline to create a docker image and push the image to dockerhub.

