**TASKS ON DOCKER**

**TASKS:**

**1. Build docker image for Java application**

**2. Build docker image for Angular application**

**3. Build docker image for Android application**

**4. Push images to AWS ECR**

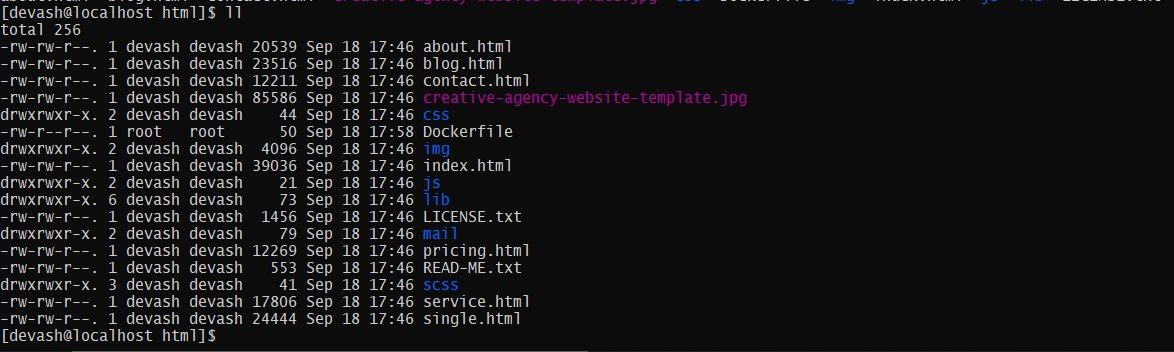
**5. Multi-stage dockerfile creation and building the image**

**6. Building the image using dockerfile and attaching volumes to the running containers (In Production Best Practices)**

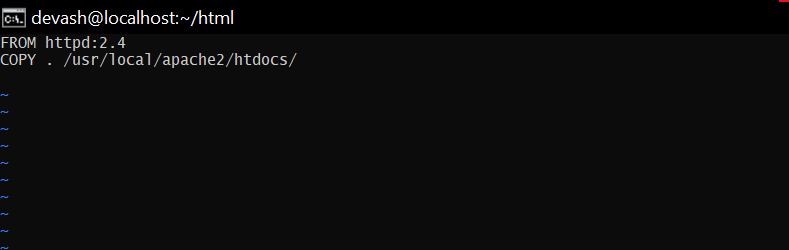
**7. Optimization of docker image( With the best scenario)**

**TASK 1: Build docker image for Java application**

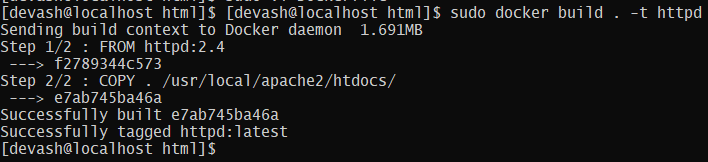
* **Downloaded a free HTML website template files**



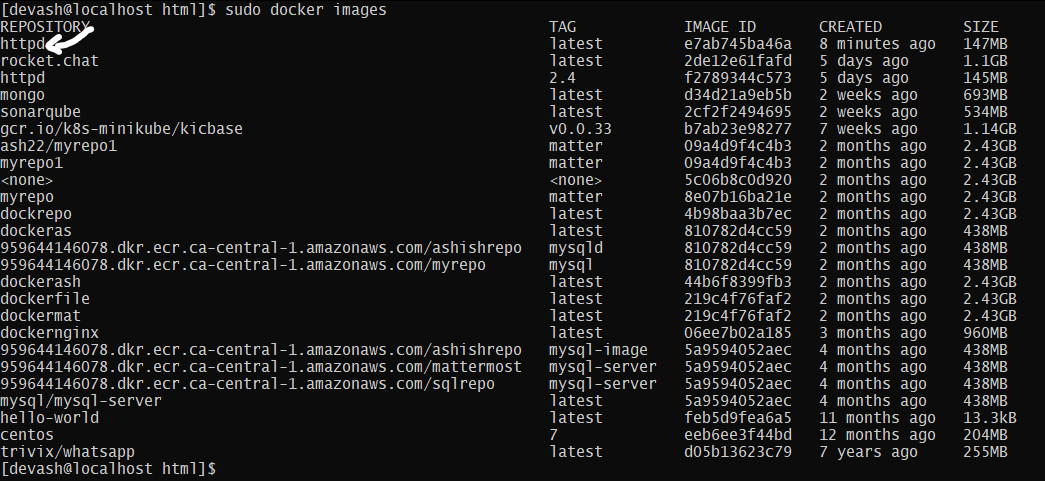
* **Created a Dockerfile for httpd**



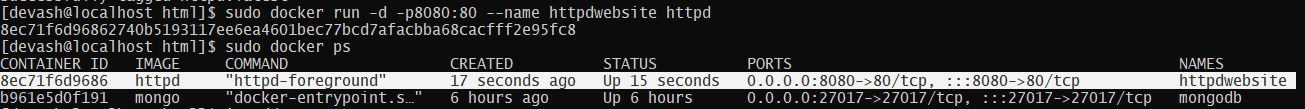
* **Built the image using docker build . -t “image name”**

****

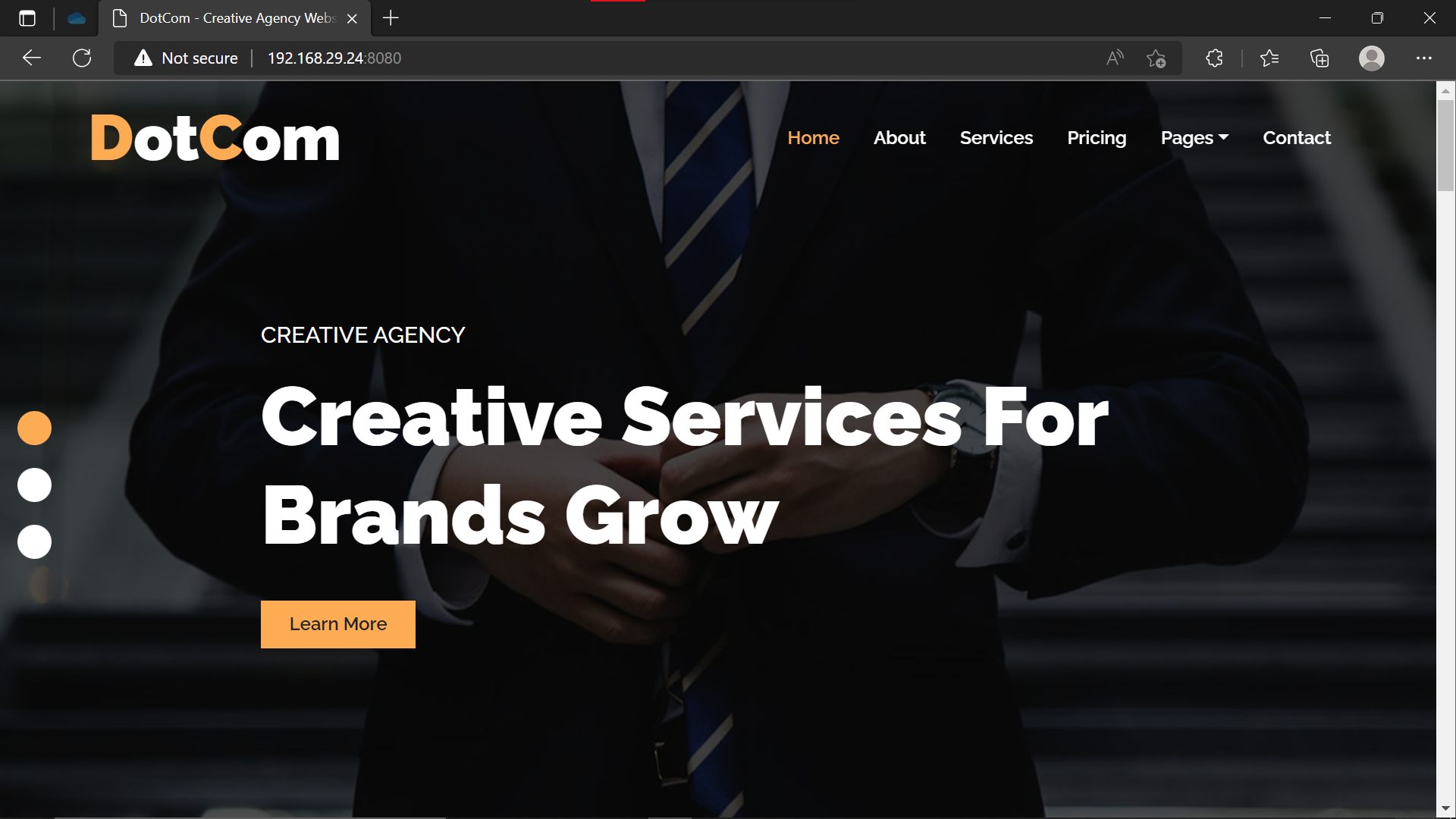
* **docker images to check the image is successfully built**



* **Created a container and tested the httpd using docker run**

****

* **Output Website**

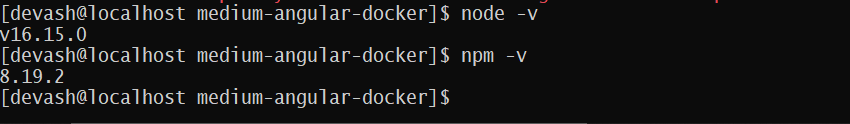
****

**TASK 2: Build docker image for Angular application**

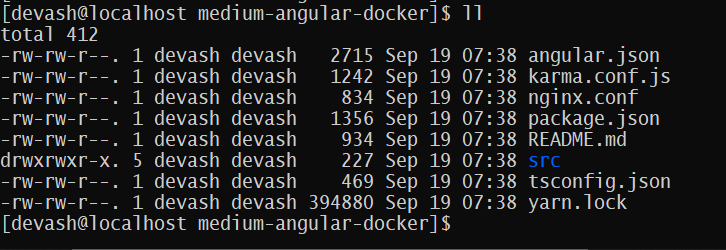
**Pre-req:**

**-Dependencies: yum install gcc-c++ make-git**

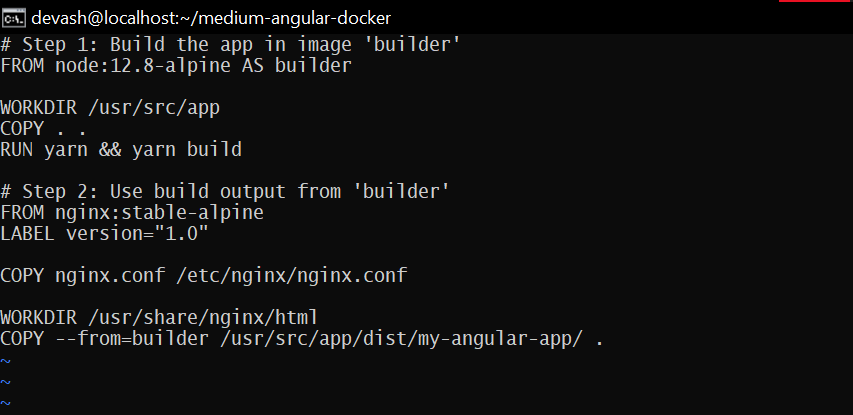
**-node , npm ,Angular CLI**

****

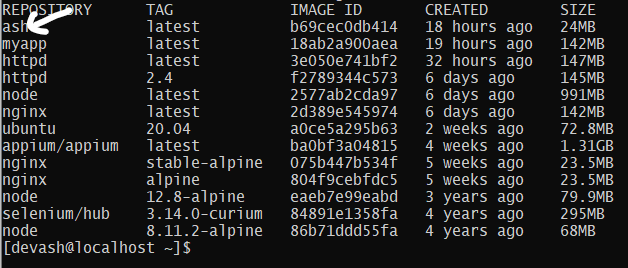
* **Cloned the code of example angular application from Github using git clone**

****

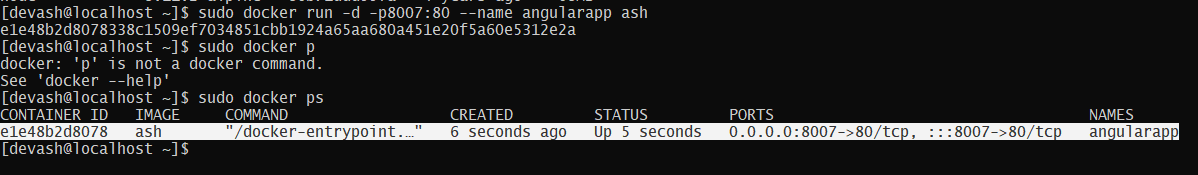
* **Built a Dockerfile**

****

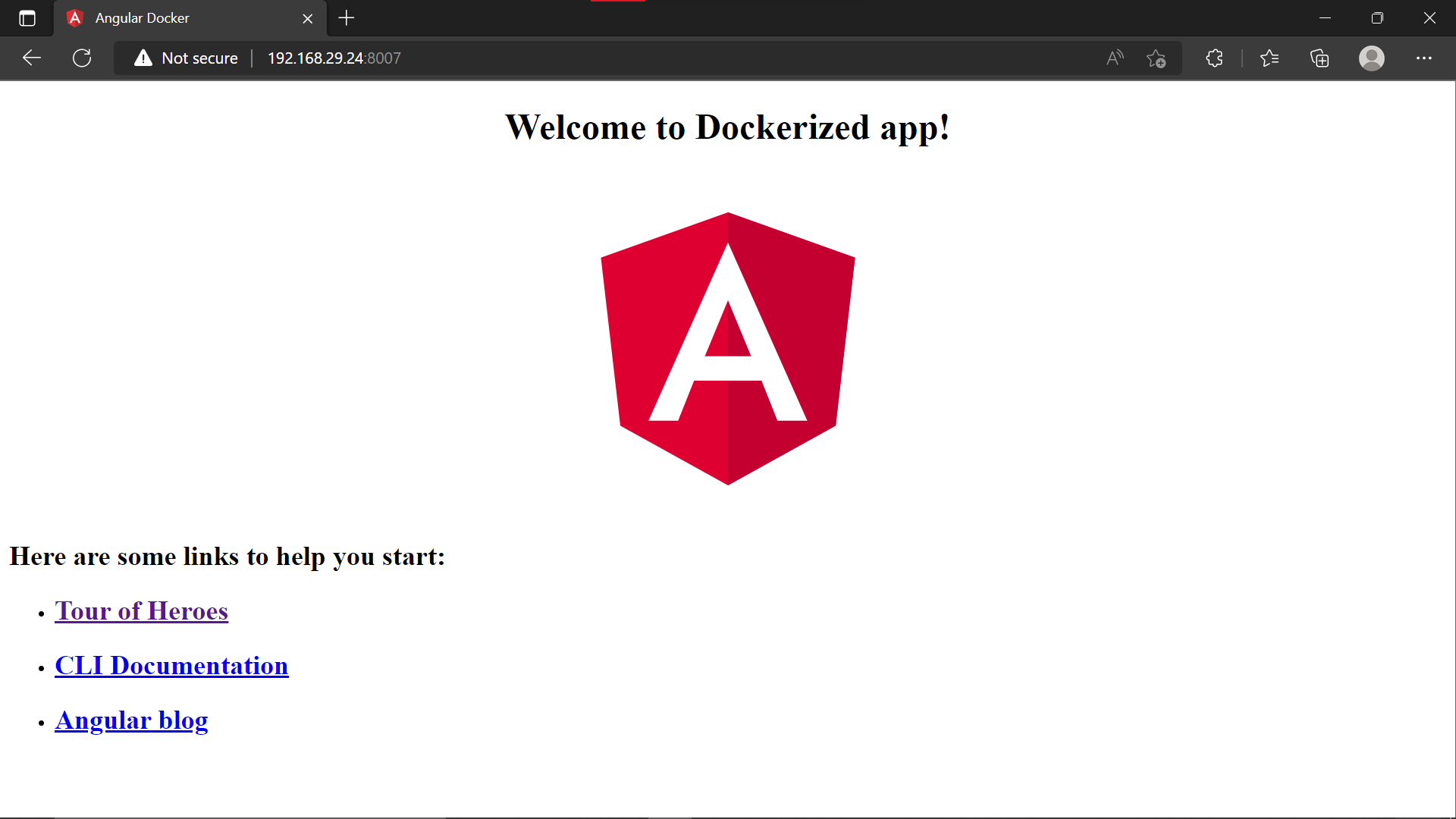
* **Built a docker image**

****

* **Started the container Using docker run**

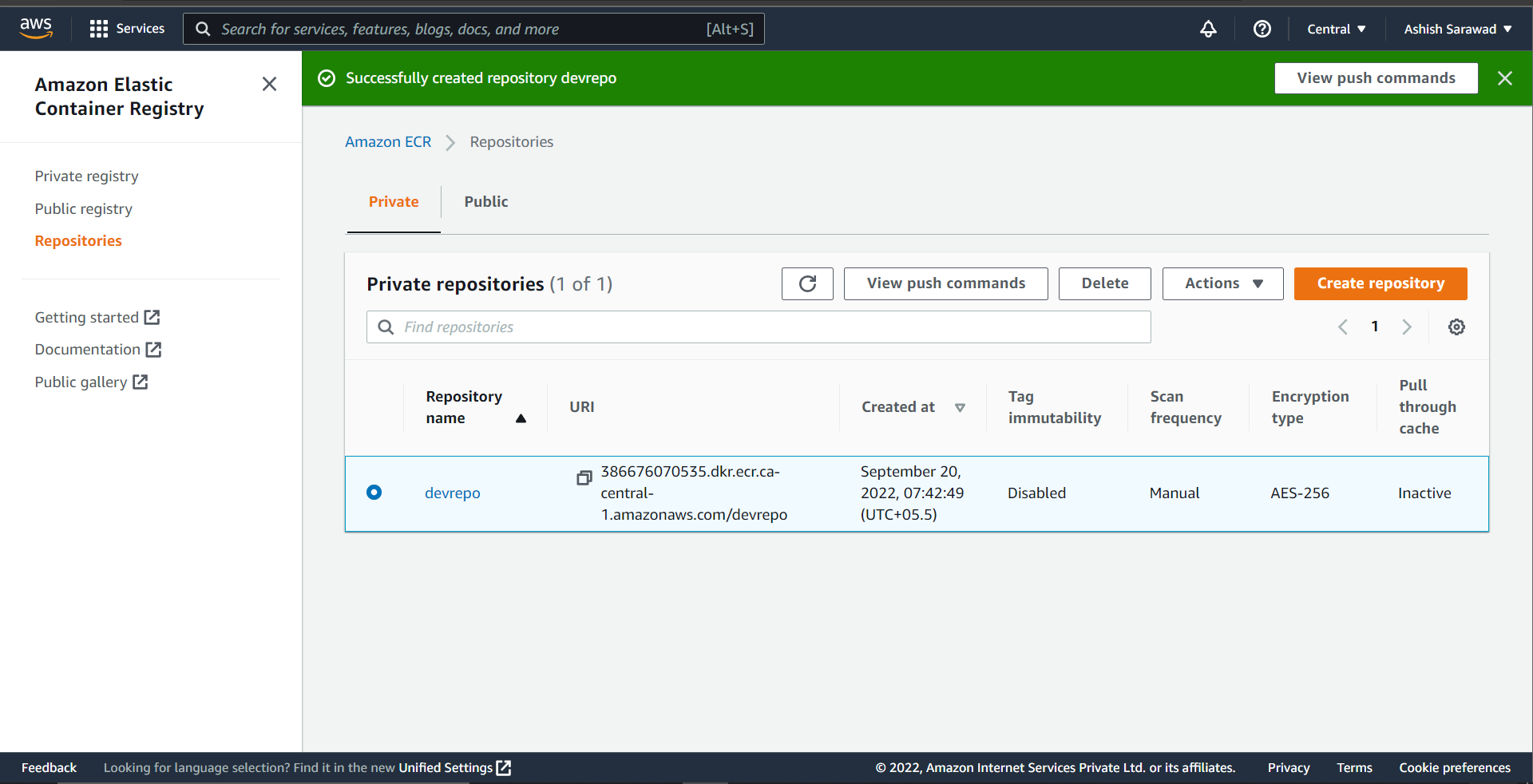
****

* **OUTPUT**

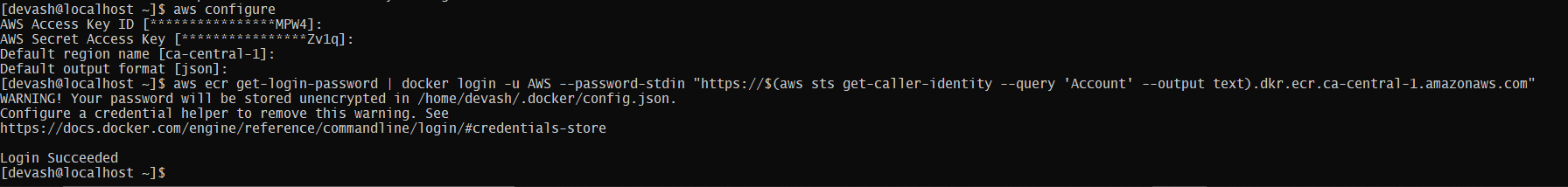
****

**TASK 4 : Push docker images it to AWS ECR**

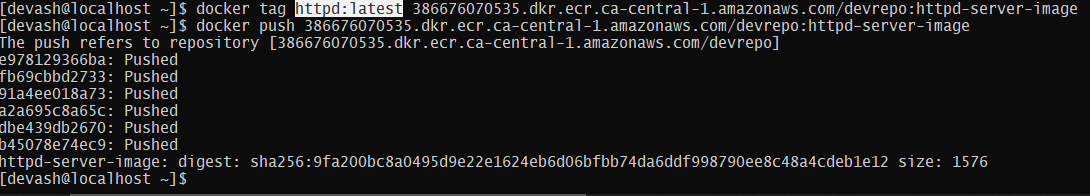
* **Created a repository in ECR “Devrepo”**

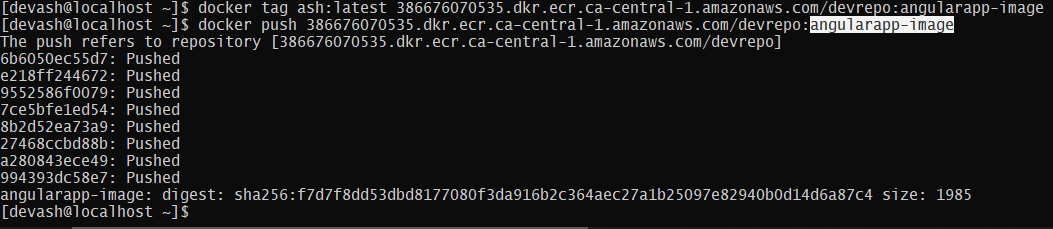
****

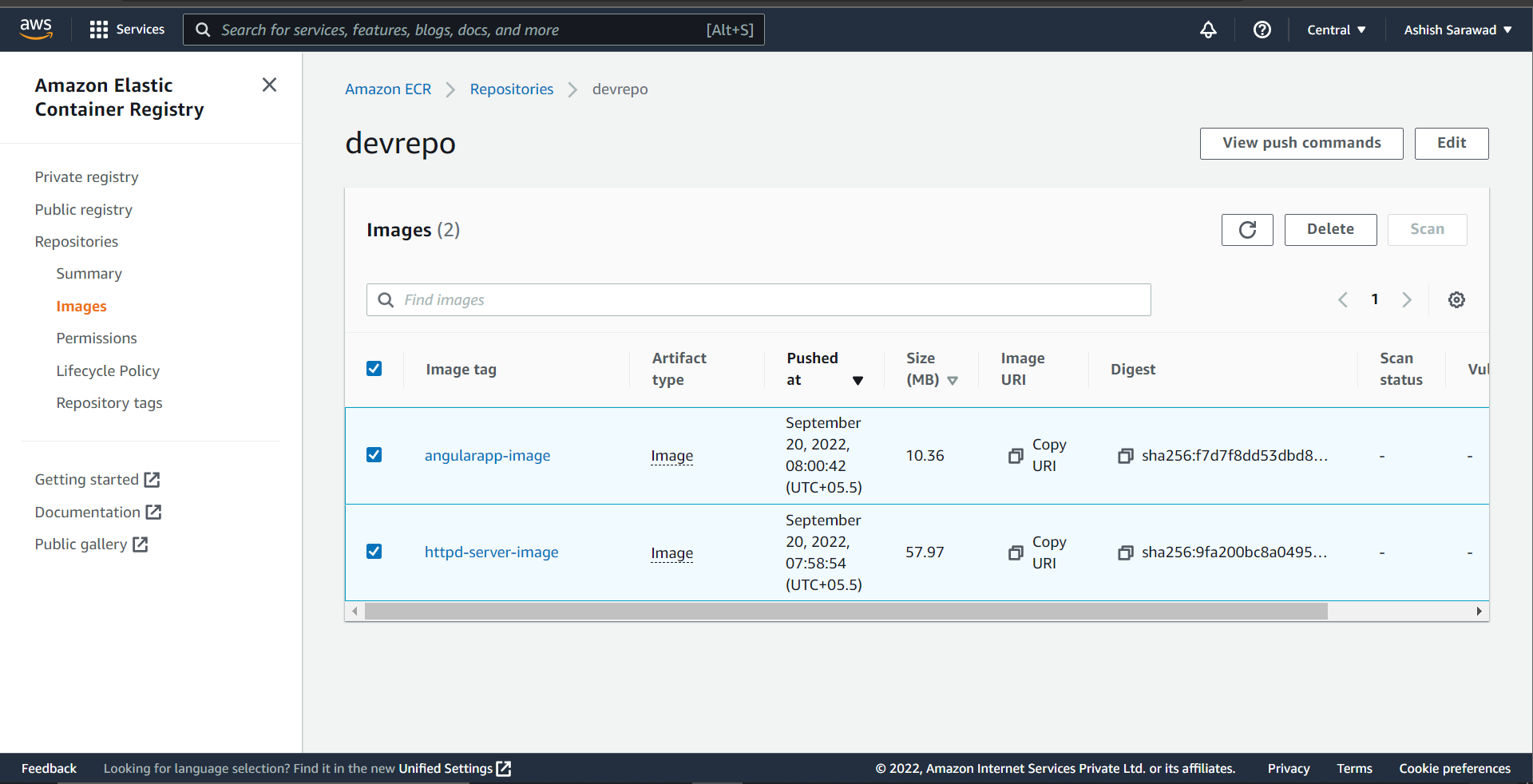
* **Configured AWS CLI and Logged in to ECR repo**

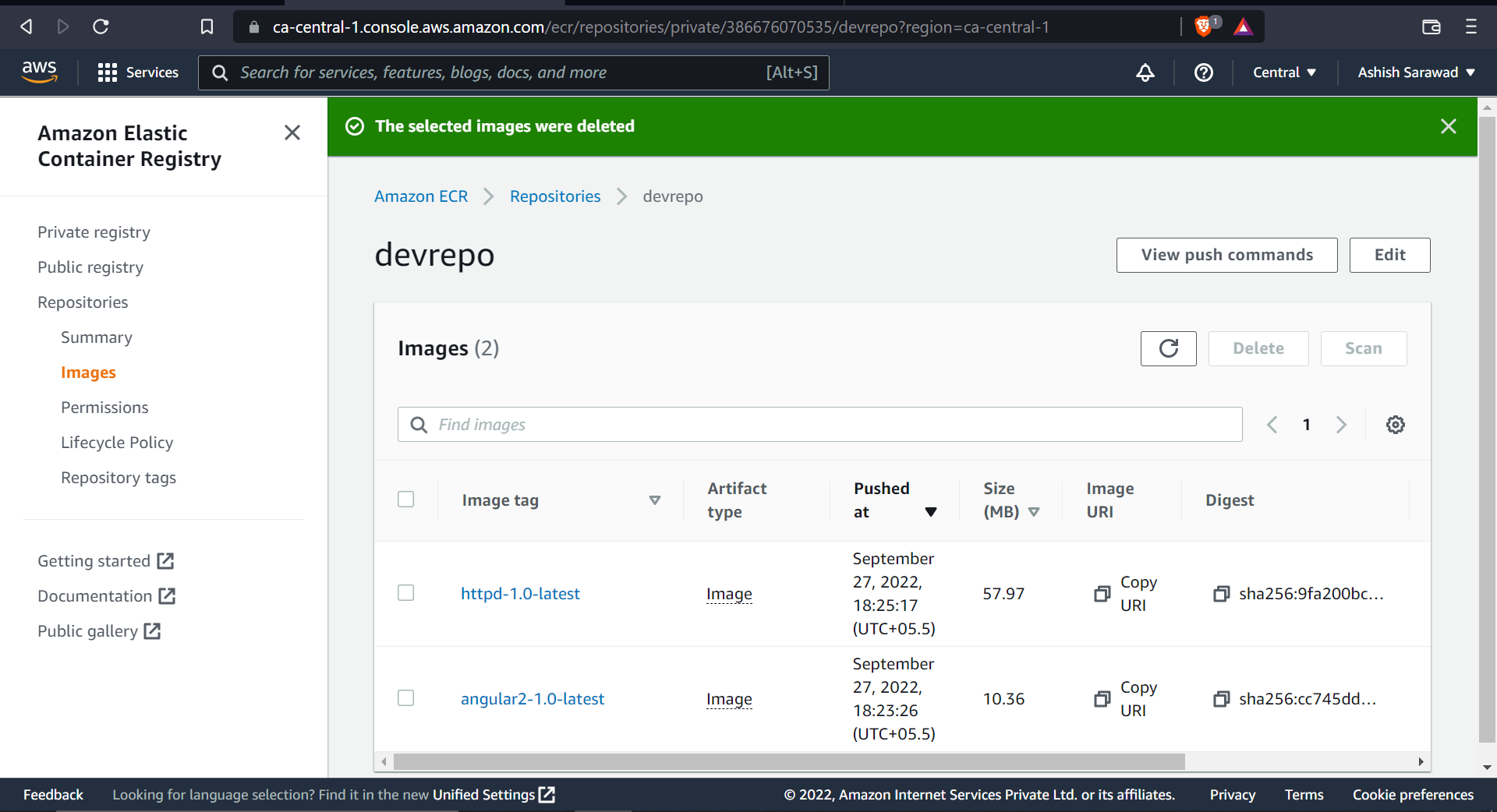
****

* **Tagged and Pushed the images to ECR Repo**

****

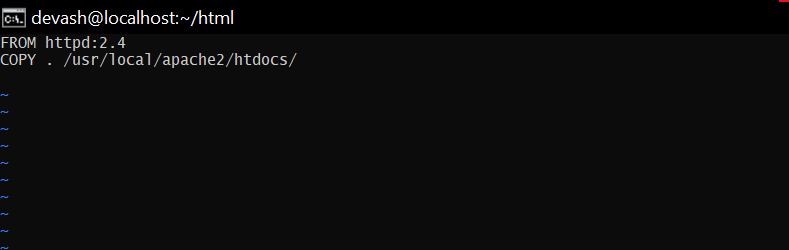
****

****

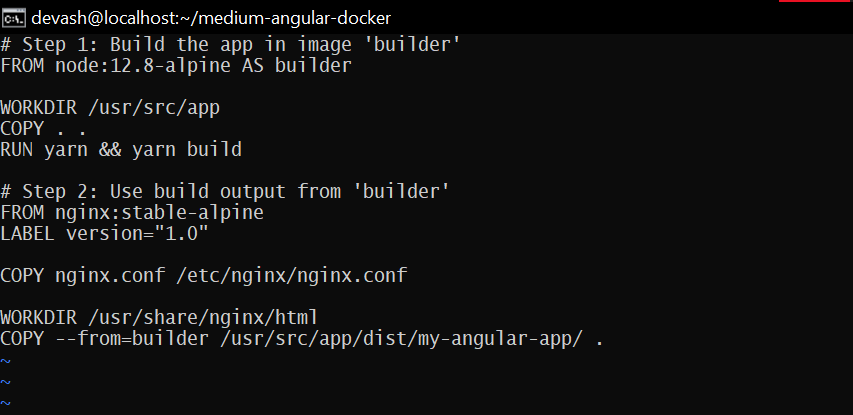
****

**TASK 5: Multi-stage dockerfile creation and building the image**

* **Java Application(httpd)**

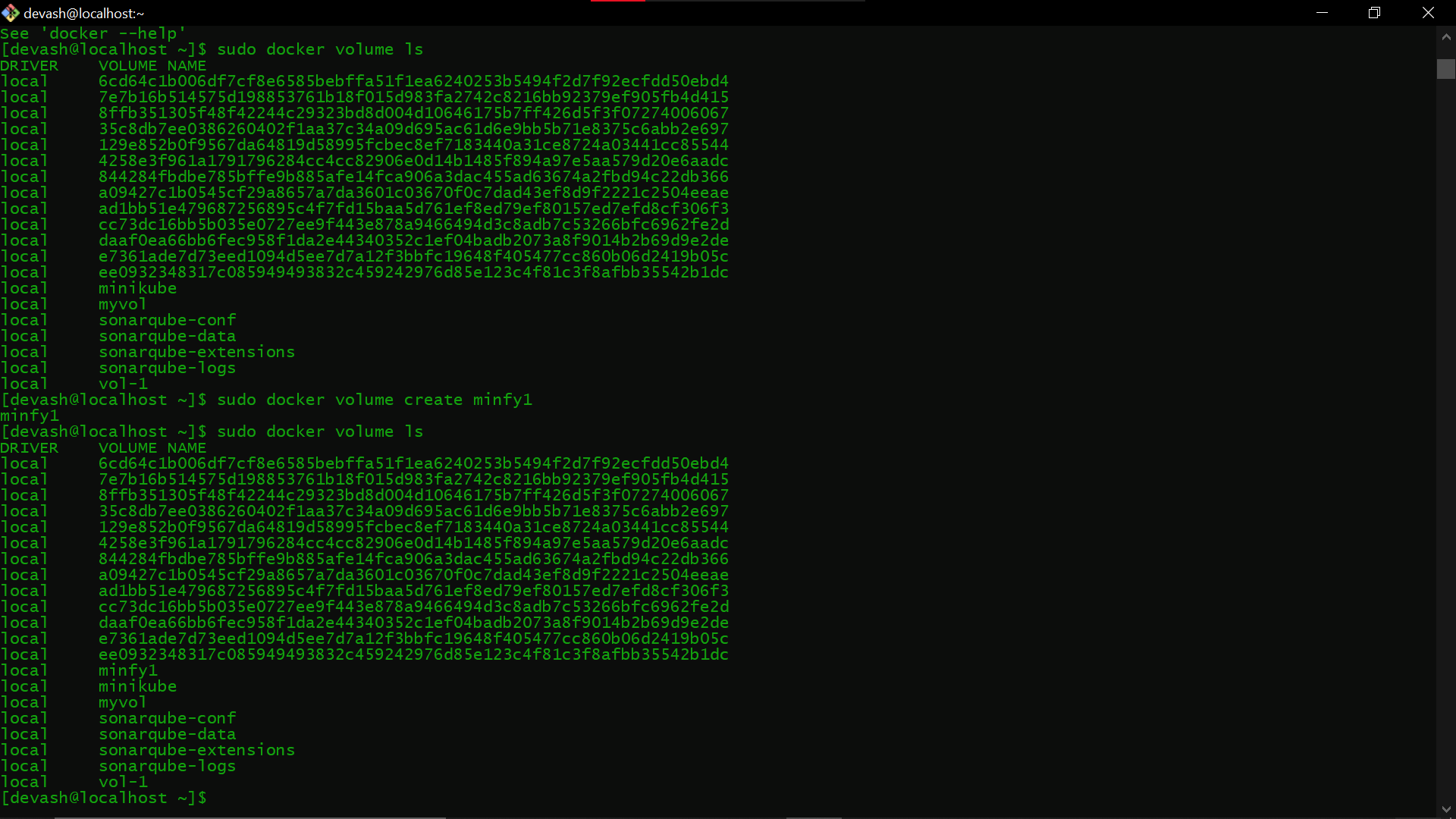


* **AngularApp**

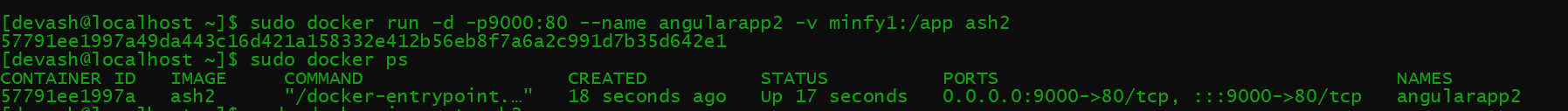
****

**TASK 6: Building the image using dockerfile and attaching volumes to the running containers (In Production Best Practices)**

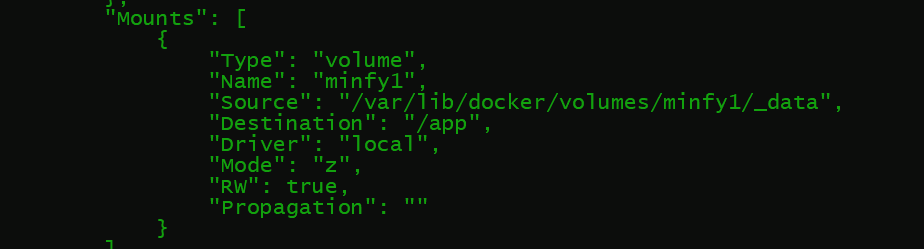
* **Create a docker volume using “docker volume create vol\_name”**

****

* **Run the docker container with -v flag to add the volume**

****

* **Use docker inspect to check whether the volume is mounted or not**

****

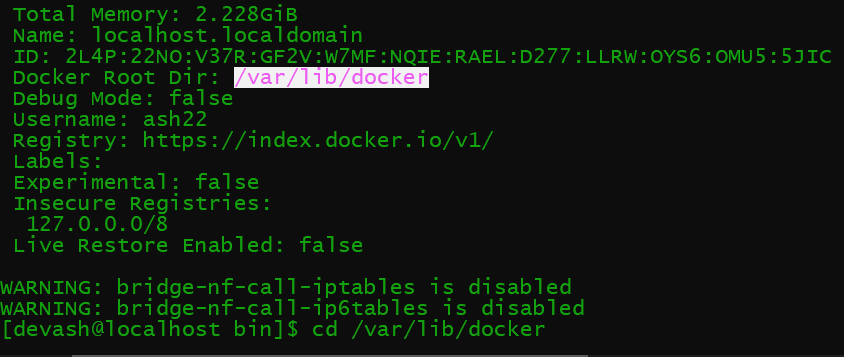
**TASK :7 Optimization of docker image( With the best scenario)**

**Best Practices for optimization of docker images**

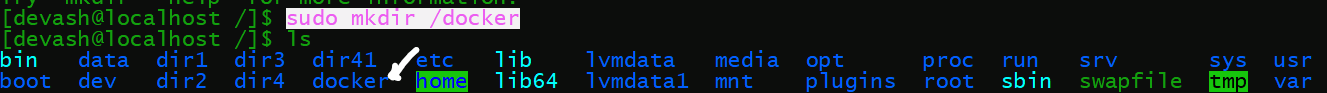
* **Using minimal base images like alpine**
* **Implementing multistage build In dockerfile**
* **Minimizing number of layers**
* **Writing a dockerfile such that whenever it is the images are built again and again from it , it uses more cached layers only**
* **Using the .dockerignore file**
* **Specifying non-Privileged user to make the container more secure**
* **Using ephemeral containers**

**TASK: 8 Change the default directory for the docker**

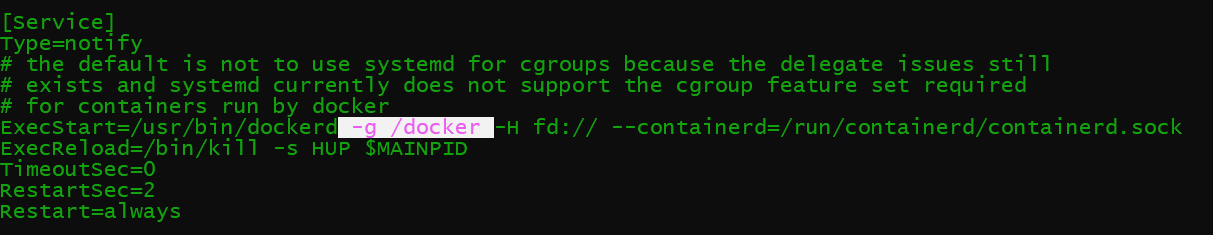
* **Use docker info to check the present root directory**

****

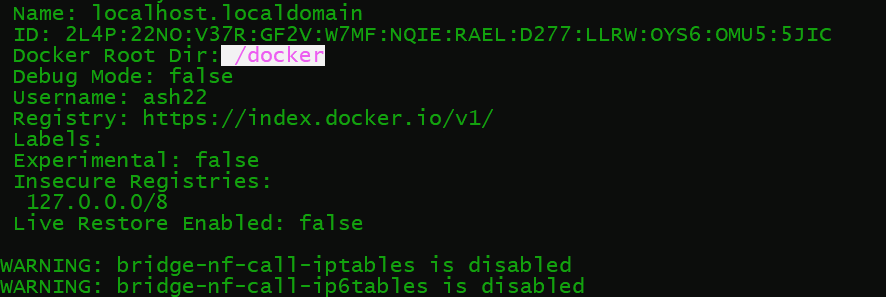
* **Created a directory to set as root directory of docker**

****

* **Make changes in docker.service file using vi /usr/lib/systemd/system/docker.service**
* **To add new directory, specify the path with -g in Execstart line**

****

* **Use docker info to check the changes**

****