Docker again

Logical Automation

>> Create a python script for delete unused/unattached ebs volumes from all the enabled region of AWS using iam role, delete the ebs volume older than 30 days in chatgpt

>> Create a github repository for deleting old ebs volumes ignoring python - **delete\_unsed\_ebs\_volume\_weekly**

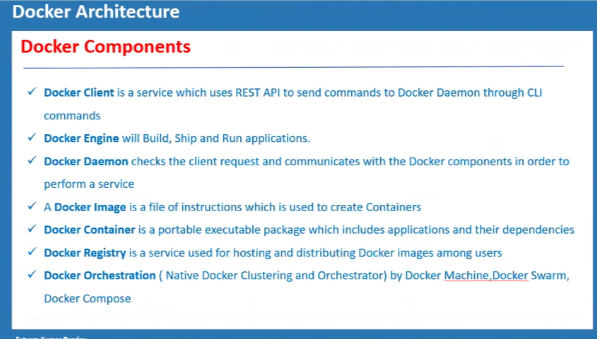
* Create new file - delete\_old\_unattached\_ebs.py
* Add script from chatgpt including pagination

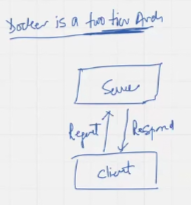
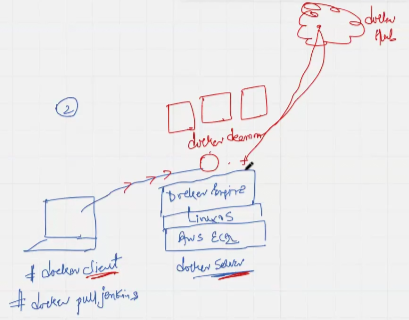
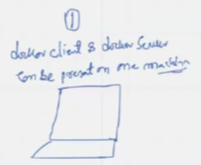
>> Create a new pipeline for same deleting ebs volume in jenkins

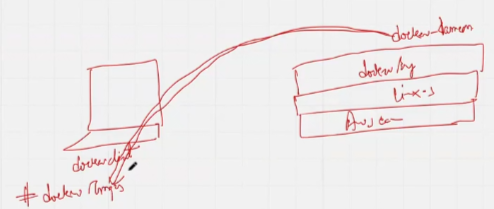
* Get me a jenkins file to run the above script from chatgpt
* (Use iam role delete-ebs-role and add a weekly cronjob) from chatgpt - Add final jenkinsfile to new file in github - Jenkinsfile
* Goto Jenkins >> Configure >> Pipeline >> Pipeline script from SCM >> SCM=git >> Repository URL=git repo url >> Script path=Jenkinsfile >> Build now ……displays errorfailure

Docker components



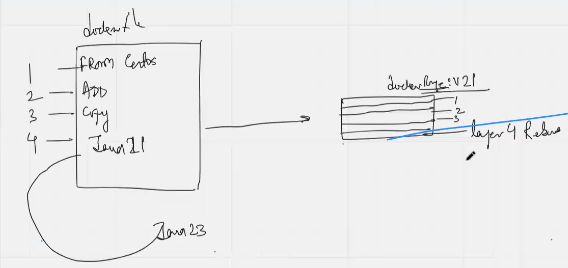


CDK v/s SDK in AWS

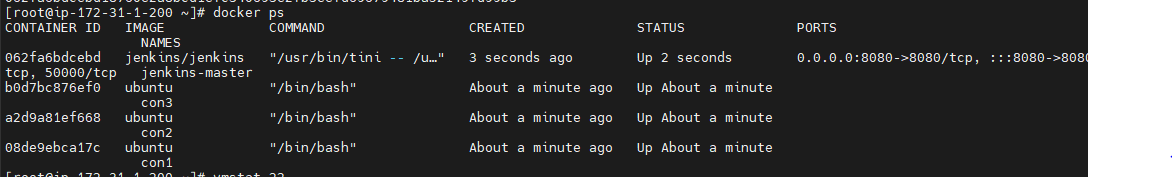
Goto AWS >> EC2 >> name=dockerhost >> keypair >> Launch instance >> Connect

* Sudo su -
* Yum install docker -y
* Usermod -aG docker ec2-user
* Service docker start
* Chkconfig docker on
* Docker –version
* **Docker version** ……displays client-server configuration including docker-server, containerd, runc, docker-init
* Docker ↵
* Docker pull mysql; docker pull jenkins/jenkins; docker pull ubuntu ……compress file downloaded, extracted, pulled and follow uncompression automatically
* Docker images ……3 images setup



Displays that a 4 layer dockerfile creates 4 layer dockerimage

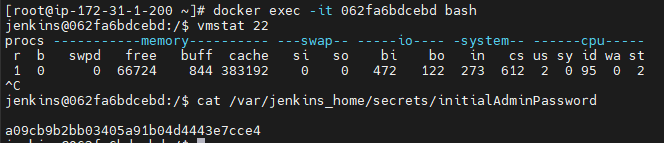
* docker container run [OPTIONS] IMAGE [COMMAND] [ARG...] ……display example
* Docker run -itd –name con1 ubuntu; Docker run -itd –name con2 ubuntu; Docker run -itd –name con3 ubuntu
* Docker ps; docker images ……ps shows containers, images show downloaded images



* Docker run -itd –name jenkins-master -p 8080:8080 jenkins/jenkins
* Docker exec -it <container\_id> bash ……login to jenkins docker container

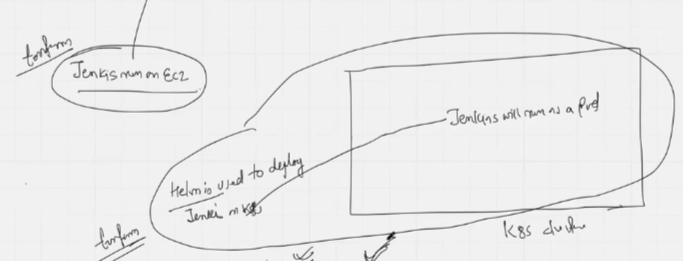
>> Goto browser >> http://public-ip:8080 >> initialAdminpassword

* Cat <password\_path>
* Hostname ……displays “062fa6bdcebd”
* Exit ……exits container



Examples of various IaC



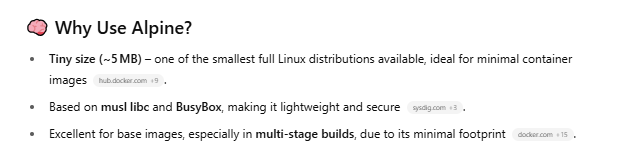
Different ways of deploying Jenkins at industry level:

Different ways of deploying Jenkins at industry level:

Create aws eks using terraform then install jenkins using helm and control the helm with terraform resource in chatgpt

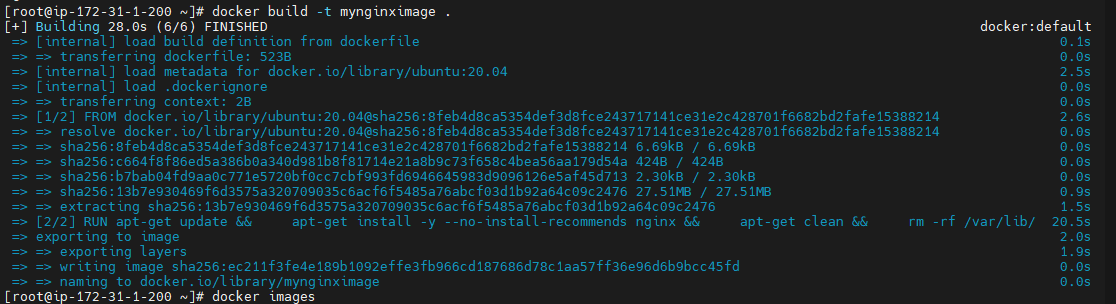
Dockerfile formatting

* Docker pull alpine

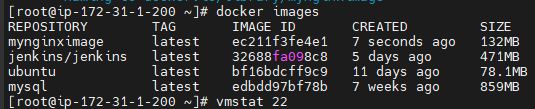


1. Goto Greentea >> Docker-7th-Feb-2025/dockerlab-24thmay

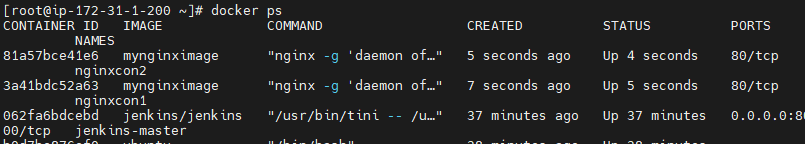
* Cat > dockerfile >> add data from point D
* Docker build -t mynginximage . ……builds layer by layer image from dockerfile



* Docker images ……displays mynginximage image



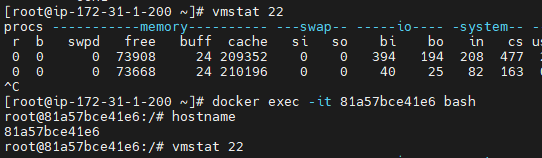
* Docker run -itd –name nginxcon1 mynginximage
* Docker run -itd –name nginxcon2 mynginximage
* Docker ps



* Docker exec -it <container\_id> bash ……login to con2
* Service nginx status ……display that nginx is running on container



* Docker build -t mynginximagepython .
* Docker images
* Docker run -itd –name nginx-py mynginximagepython
* Docker exec -it <container\_id> bash

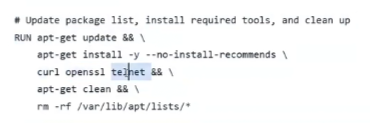


* Service nginx status



* Python –version

1. With curl, openssl & telnet



1. With python, jdk



* Docker build -t myimage-py-jdk .
* Docker run -itd –name 011-py-jdk myimage-py-jdk
* Docker exec -it <container\_id> bash

>> Goto Github >> More dockerfile examples are located in greentea/Docker-7Feb/10May

# Use the official Ubuntu base image

FROM ubuntu:20.04

# Set a label instead of using deprecated MAINTAINER

LABEL maintainer="demousr@gmail.com"

# Install Nginx and clean up

RUN apt-get update && \

apt-get install -y --no-install-recommends nginx && \

apt-get clean && \

rm -rf /var/lib/apt/lists/\*

# Expose the default Nginx port

EXPOSE 80

# Start Nginx in the foreground

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