K L UNIVERSITY

A Project Based Lab Report

On

Automating CI/CD Pipeline for a Web Application Deployment Using Jenkins, Docker, and AWS EC2

SUBMITTED BY:

I.D NUMBER	NAME
2200031608	CH. BALAJI

UNDER THE ESTEEMED GUIDANCE OF

G. ANJANEYULU

Asst.Professor

KL UNIVERSITY
Green fields, Vaddeswaram – 522 502
Guntur Dt., AP, India.

ABSTRACT

This project implements a Continuous Integration and Continuous Deployment (CI/CD) pipeline to automate the deployment of a Java-based web application. The pipeline uses Jenkins, Docker, and AWS EC2 to streamline the build, deployment, and hosting processes. By integrating Jenkins for automation, Docker for containerization, and EC2 for hosting, the project achieves a scalable and consistent deployment workflow. The pipeline pulls the source code from a Git repository, builds the application using Maven, creates a Docker container from a custom Dockerfile, and deploys it on an EC2 instance. This project demonstrates the importance of automation in modern software development by minimizing manual intervention and ensuring rapid, reliable updates to production.

INTRODUCTION

In modern software development, the need for faster and more reliable delivery of applications has made DevOps practices crucial. This project focuses on creating an automated CI/CD pipeline for deploying a web application using Jenkins, Docker, and AWS EC2. The primary objective is to eliminate the manual steps in deployment, ensuring a smooth transition from code changes to production-ready software.

Jenkins serves as the automation server to orchestrate the CI/CD process, while Docker ensures consistent and portable runtime environments. The application is hosted on an AWS EC2 instance, making it accessible via a public IP. The project workflow involves pulling the latest source code, building it with Maven, packaging it as a WAR file, and deploying it within a Docker container running a Tomcat server. This implementation not only enhances productivity but also demonstrates the scalability and reliability of automated deployment pipelines.

PROJECT STATEMENT

This project focuses on building a Continuous Integration and Continuous Deployment (CI/CD) pipeline to automate the deployment of a Java-based web application. The pipeline integrates Jenkins, Docker, and AWS EC2 to ensure an efficient and consistent delivery process. The workflow involves pulling the source code from a Git repository, building the application using Maven, creating a Docker container with the application, and deploying it on an EC2 instance. The pipeline eliminates manual intervention, accelerates deployment cycles, and ensures reproducibility by leveraging infrastructure as code (Dockerfile) and automation tools.

TOOLS AND TECHNOLOGIES USED

1. Jenkins

- Continuous Integration/Continuous Deployment (CI/CD) server for automating build and deployment processes.

2. Docker

- Containerization platform for packaging the application with all dependencies into portable containers.

3. AWS EC2

- Cloud-based virtual server used as the Docker host to deploy and run the application.

4. Maven

- Build automation tool for Java projects to manage dependencies, compile source code, and package it as a deployable WAR file.

5. Git

- Version control system to manage and track changes to the application's source code.

6. Tomcat

- Application server within the Docker container to run the Java-based web application.

7. Linux (Amazon Linux/Ubuntu)

- Operating system on the EC2 instance for hosting Docker and running related commands.

8. SSH

- Secure protocol used for remote management and transferring files to the Docker host.

9. Web Browser

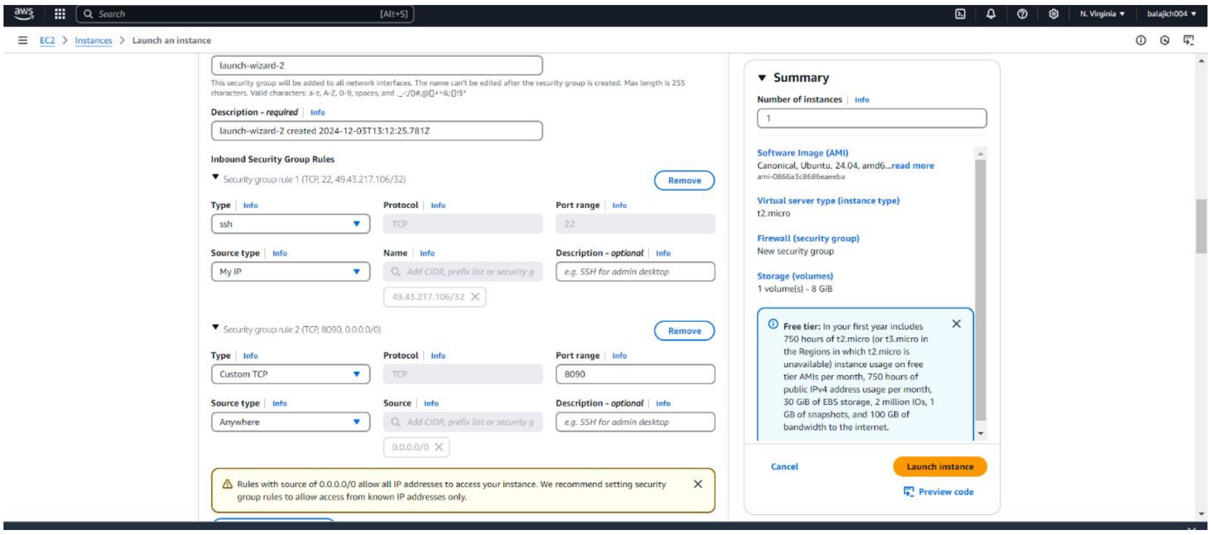
- For accessing Jenkins and testing the deployed web application via the public IP of the EC2 instance.

10. Java

- Programming language used to develop the web application.

PROJECT STEPS AND RESULT SCREENSHOTS

Step-1: Launching an EC2 Instance for Docker Host



EC2 > Instances > Launch an instance

launch-wizard-2

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-:/()#,@[]+=&;{}!$*

Description - *required*: launch-wizard-2 created 2024-12-03T13:12:25.781Z

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 49.43.217.106/32)

Type: ssh | Protocol: TCP | Port range: 22

Source type: My IP | Name: 49.43.217.106/32 | Description - *optional*: e.g. SSH for admin desktop

Security group rule 2 (TCP, 8090, 0.0.0.0/0)

Type: Custom TCP | Protocol: TCP | Port range: 8090

Source type: Anywhere | Source: 0.0.0.0/0 | Description - *optional*: e.g. SSH for admin desktop

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more ami-0866a3c8686eaeeba

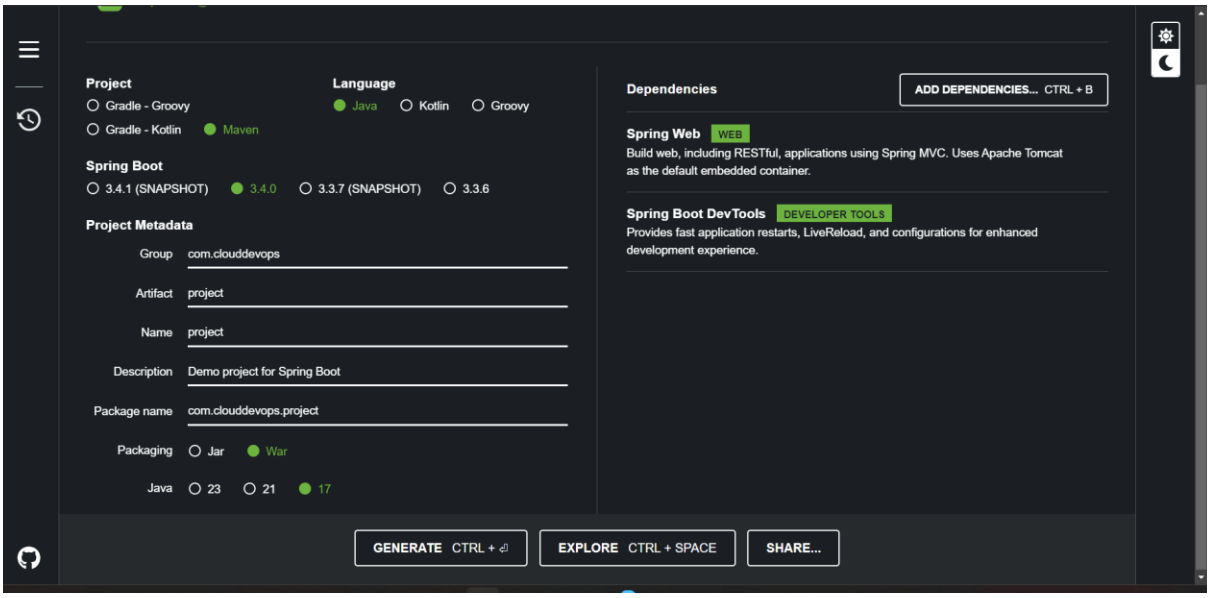
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Step-2 : Setting up a spring based git repository



Project: Gradle - Groovy, Gradle - Kotlin, [✓ Maven]

Language: [✓ Java], Kotlin, Groovy

Spring Boot: 3.4.1 (SNAPSHOT), [✓ 3.4.0], 3.3.7 (SNAPSHOT), 3.3.6

Project Metadata

Group: com.clouddevops

Artifact: project

Name: project

Description: Demo project for Spring Boot

Package name: com.clouddevops.project

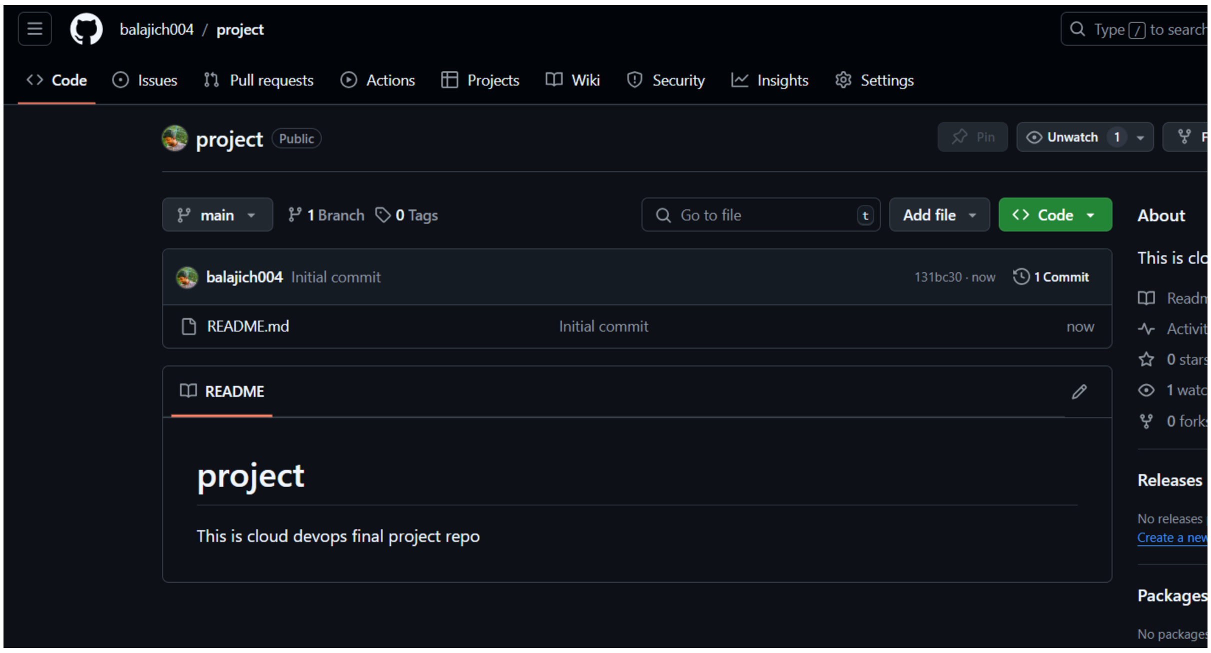
Packaging: Jar, [✓ War]

Java: 23, 21, [✓ 17]

Dependencies

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Boot DevTools DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.


balajich004 / project

project Public

main · 1 Branch · 0 Tags

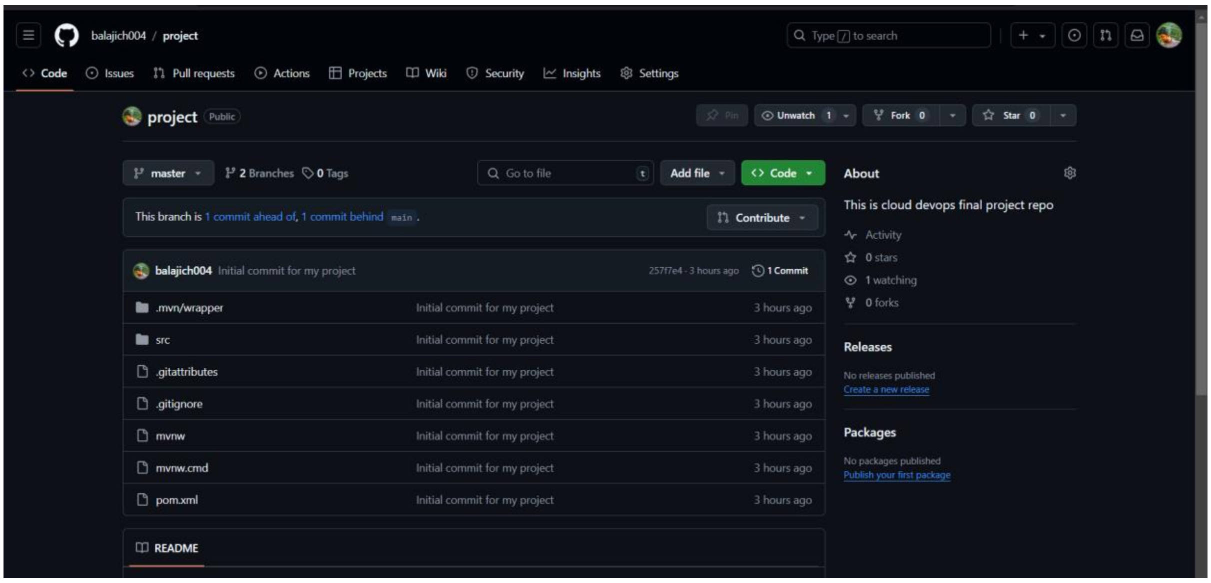
balajich004 Initial commit · 131bc30 · now · 1 Commit

README.md	Initial commit	now

README

project

This is cloud devops final project repo


balajich004 / project

project Public

master · 2 Branches · 0 Tags

This branch is 1 commit ahead of, 1 commit behind `main`.

balajich004 Initial commit for my project · 257f7e4 · 3 hours ago · 1 Commit

.mvn/wrapper	Initial commit for my project	3 hours ago
src	Initial commit for my project	3 hours ago
.gitattributes	Initial commit for my project	3 hours ago
.gitignore	Initial commit for my project	3 hours ago
mvnw	Initial commit for my project	3 hours ago
mvnw.cmd	Initial commit for my project	3 hours ago
pom.xml	Initial commit for my project	3 hours ago

About

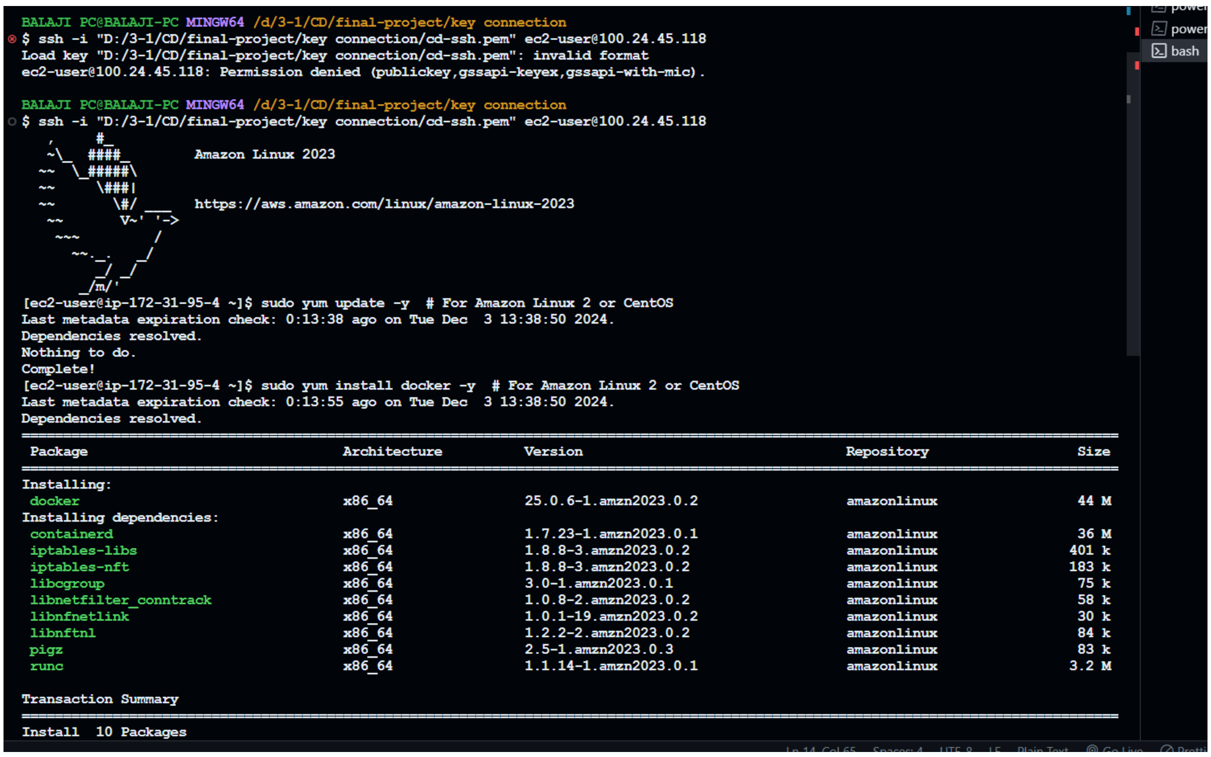
This is cloud devops final project repo

Activity · 0 stars · 1 watching · 0 forks

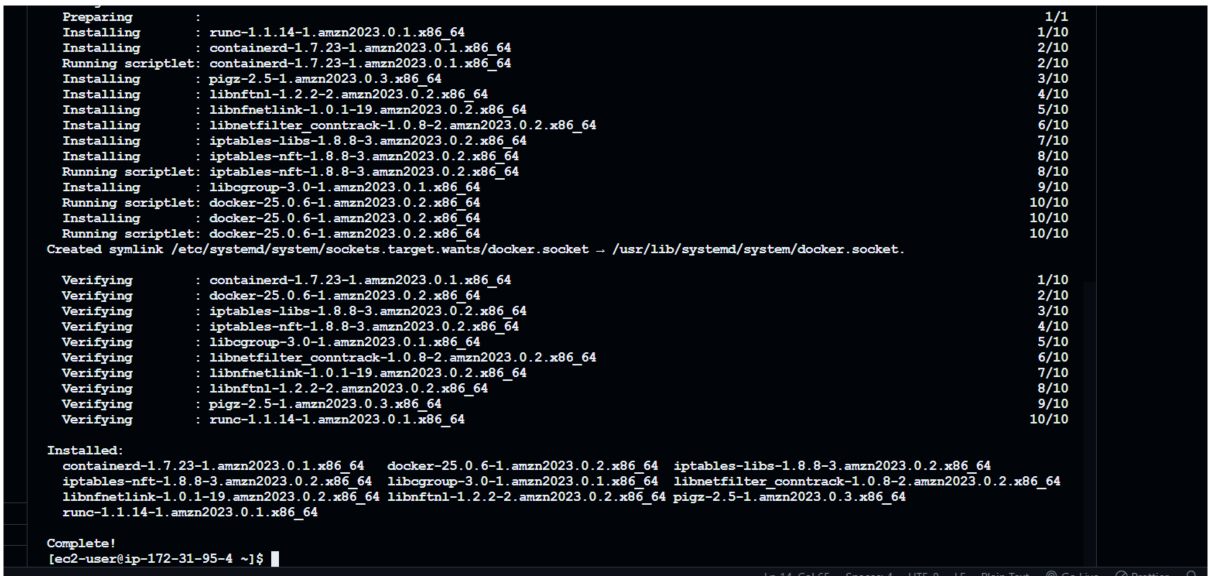
Releases: No releases published

Packages: No packages published

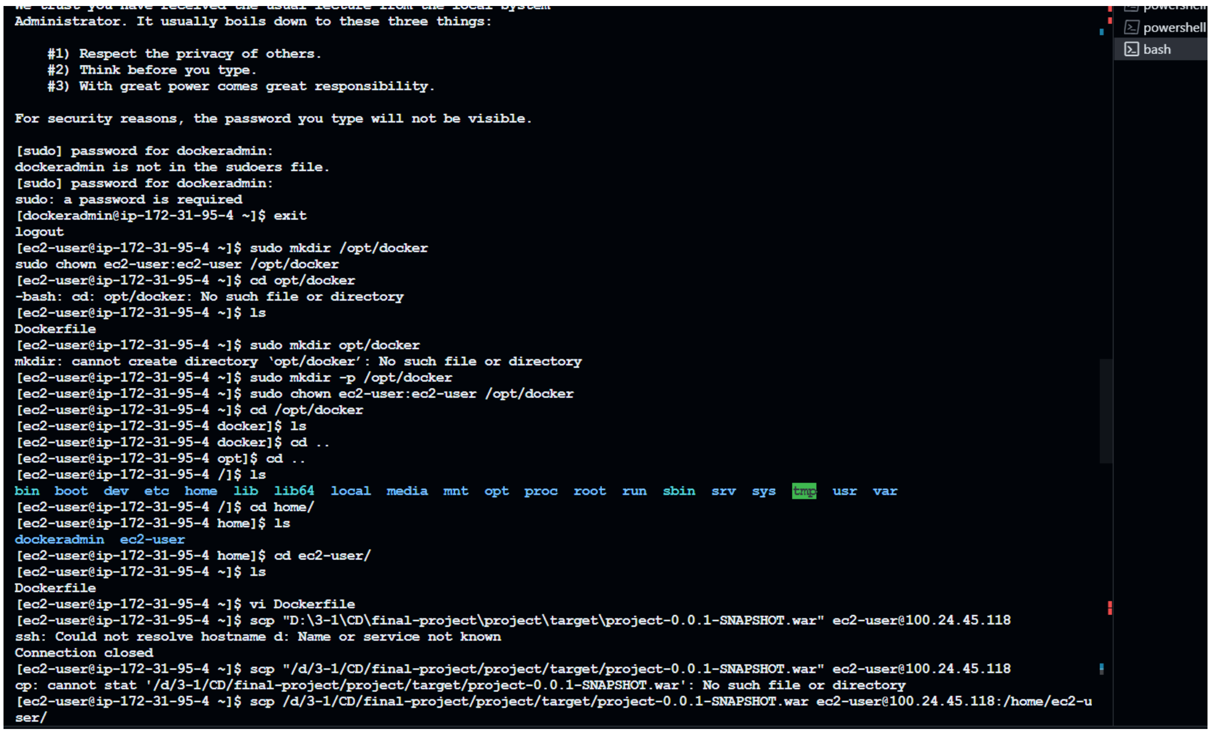
Step-3 : Install Docker on EC2



```
BALAJI PC@BALAJI-PC MINGW64 /d/3-1/CD/final-project/key connection
$ ssh -i "D:/3-1/CD/final-project/key connection/cd-ssh.pem" ec2-user@100.24.45.118
Load key "D:/3-1/CD/final-project/key connection/cd-ssh.pem": invalid format
ec2-user@100.24.45.118: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

BALAJI PC@BALAJI-PC MINGW64 /d/3-1/CD/final-project/key connection
$ ssh -i "D:/3-1/CD/final-project/key connection/cd-ssh.pem" ec2-user@100.24.45.118
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-95-4 ~]$ sudo yum update -y  # For Amazon Linux 2 or CentOS
Last metadata expiration check: 0:13:38 ago on Tue Dec  3 13:38:50 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-95-4 ~]$ sudo yum install docker -y  # For Amazon Linux 2 or CentOS
Last metadata expiration check: 0:13:55 ago on Tue Dec  3 13:38:50 2024.
Dependencies resolved.
=====================================================================================================================================
 Package                       Architecture          Version                             Repository                          Size
=====================================================================================================================================
Installing:
 docker                        x86_64                25.0.6-1.amzn2023.0.2               amazonlinux                         44 M
Installing dependencies:
 containerd                    x86_64                1.7.23-1.amzn2023.0.1               amazonlinux                         36 M
 iptables-libs                 x86_64                1.8.8-3.amzn2023.0.2                amazonlinux                        401 k
 iptables-nft                  x86_64                1.8.8-3.amzn2023.0.2                amazonlinux                        183 k
 libcgroup                     x86_64                3.0-1.amzn2023.0.1                  amazonlinux                         75 k
 libnetfilter_conntrack        x86_64                1.0.8-2.amzn2023.0.2                amazonlinux                         58 k
 libnfnetlink                  x86_64                1.0.1-19.amzn2023.0.2               amazonlinux                         30 k
 libnftnl                      x86_64                1.2.2-2.amzn2023.0.2                amazonlinux                         84 k
 pigz                          x86_64                2.5-1.amzn2023.0.3                  amazonlinux                         83 k
 runc                          x86_64                1.1.14-1.amzn2023.0.1               amazonlinux                        3.2 M

Transaction Summary
=====================================================================================================================================
Install  10 Packages
```

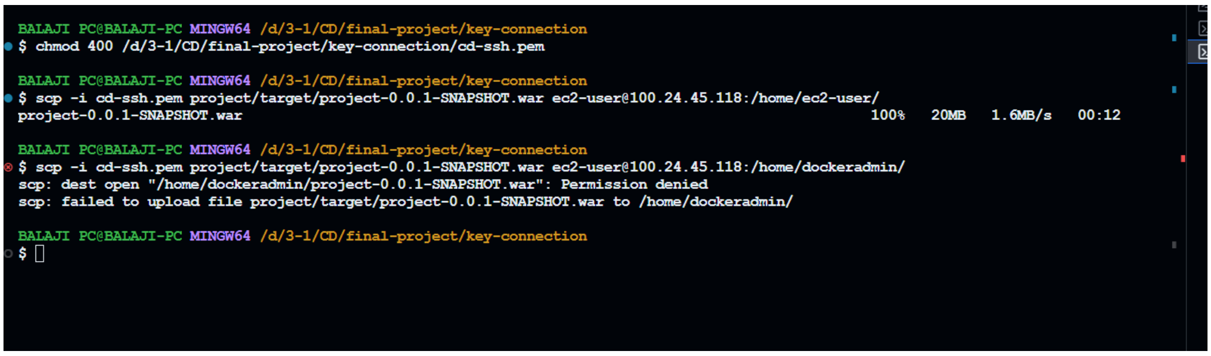


```
  Preparing        :                                                                        1/1
  Installing       : runc-1.1.14-1.amzn2023.0.1.x86_64                                     1/10
  Installing       : containerd-1.7.23-1.amzn2023.0.1.x86_64                               2/10
  Running scriptlet: containerd-1.7.23-1.amzn2023.0.1.x86_64                               2/10
  Installing       : pigz-2.5-1.amzn2023.0.3.x86_64                                        3/10
  Installing       : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                  4/10
  Installing       : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                             5/10
  Installing       : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                    6/10
  Installing       : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                             7/10
  Installing       : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                              8/10
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                              8/10
  Installing       : libcgroup-3.0-1.amzn2023.0.1.x86_64                                   9/10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64                                  10/10
  Installing       : docker-25.0.6-1.amzn2023.0.2.x86_64                                  10/10
  Running scriptlet: docker-25.0.6-1.amzn2023.0.2.x86_64                                  10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Verifying        : containerd-1.7.23-1.amzn2023.0.1.x86_64                               1/10
  Verifying        : docker-25.0.6-1.amzn2023.0.2.x86_64                                   2/10
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                             3/10
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                              4/10
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                                   5/10
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                    6/10
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                             7/10
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                  8/10
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                        9/10
  Verifying        : runc-1.1.14-1.amzn2023.0.1.x86_64                                    10/10

Installed:
  containerd-1.7.23-1.amzn2023.0.1.x86_64   docker-25.0.6-1.amzn2023.0.2.x86_64   iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64  libcgroup-3.0-1.amzn2023.0.1.x86_64   libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64 libnftnl-1.2.2-2.amzn2023.0.2.x86_64 pigz-2.5-1.amzn2023.0.3.x86_64
  runc-1.1.14-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-95-4 ~]$
```

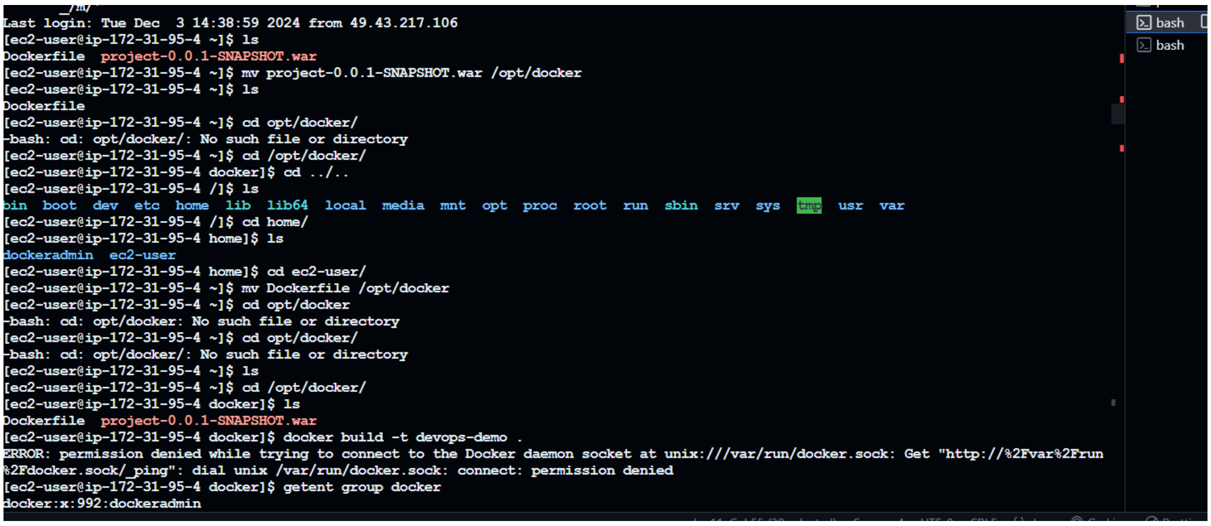


```
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

For security reasons, the password you type will not be visible.

[sudo] password for dockeradmin:
dockeradmin is not in the sudoers file.
[sudo] password for dockeradmin:
sudo: a password is required
[dockeradmin@ip-172-31-95-4 ~]$ exit
logout
[ec2-user@ip-172-31-95-4 ~]$ sudo mkdir /opt/docker
sudo chown ec2-user:ec2-user /opt/docker
[ec2-user@ip-172-31-95-4 ~]$ cd opt/docker
-bash: cd: opt/docker: No such file or directory
[ec2-user@ip-172-31-95-4 ~]$ ls
Dockerfile
[ec2-user@ip-172-31-95-4 ~]$ sudo mkdir opt/docker
mkdir: cannot create directory ‘opt/docker’: No such file or directory
[ec2-user@ip-172-31-95-4 ~]$ sudo mkdir -p /opt/docker
[ec2-user@ip-172-31-95-4 ~]$ sudo chown ec2-user:ec2-user /opt/docker
[ec2-user@ip-172-31-95-4 ~]$ cd /opt/docker
[ec2-user@ip-172-31-95-4 docker]$ ls
[ec2-user@ip-172-31-95-4 docker]$ cd ..
[ec2-user@ip-172-31-95-4 opt]$ cd ..
[ec2-user@ip-172-31-95-4 /]$ ls
bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[ec2-user@ip-172-31-95-4 /]$ cd home/
[ec2-user@ip-172-31-95-4 home]$ ls
dockeradmin  ec2-user
[ec2-user@ip-172-31-95-4 home]$ cd ec2-user/
[ec2-user@ip-172-31-95-4 ~]$ ls
Dockerfile
[ec2-user@ip-172-31-95-4 ~]$ vi Dockerfile
[ec2-user@ip-172-31-95-4 ~]$ scp "D:\3-1\CD\final-project\project\target\project-0.0.1-SNAPSHOT.war" ec2-user@100.24.45.118
ssh: Could not resolve hostname d: Name or service not known
Connection closed
[ec2-user@ip-172-31-95-4 ~]$ scp "/d/3-1/CD/final-project/project/target/project-0.0.1-SNAPSHOT.war" ec2-user@100.24.45.118
cp: cannot stat '/d/3-1/CD/final-project/project/target/project-0.0.1-SNAPSHOT.war': No such file or directory
[ec2-user@ip-172-31-95-4 ~]$ scp /d/3-1/CD/final-project/project/target/project-0.0.1-SNAPSHOT.war ec2-user@100.24.45.118:/home/ec2-u
ser/
```

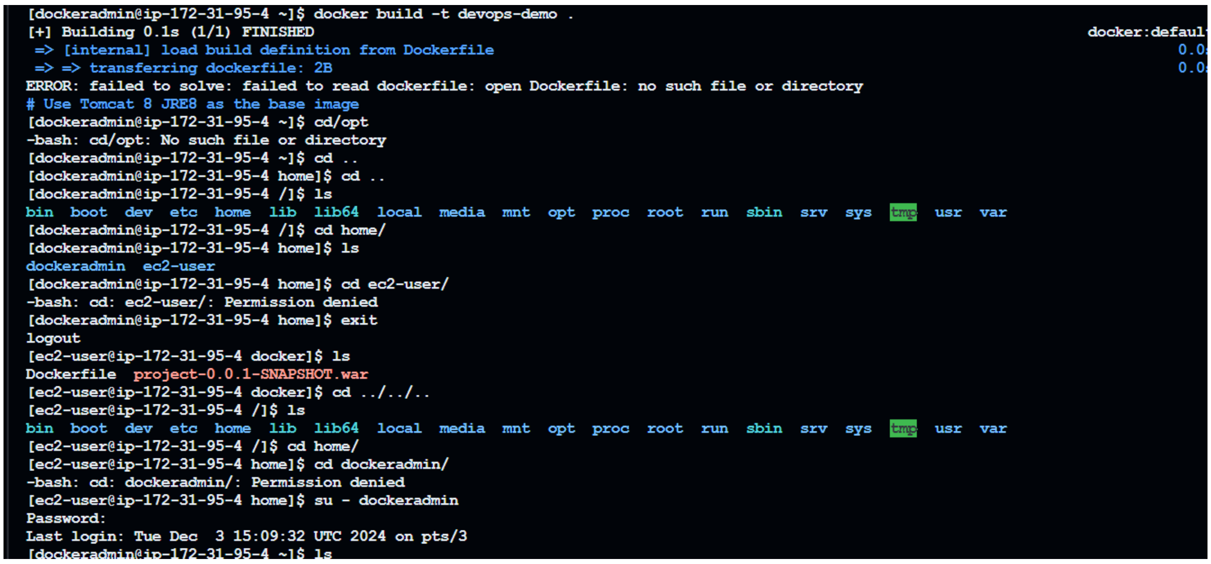
Step-4 : Creating Directory and Writing Dockerfile and deploying war file



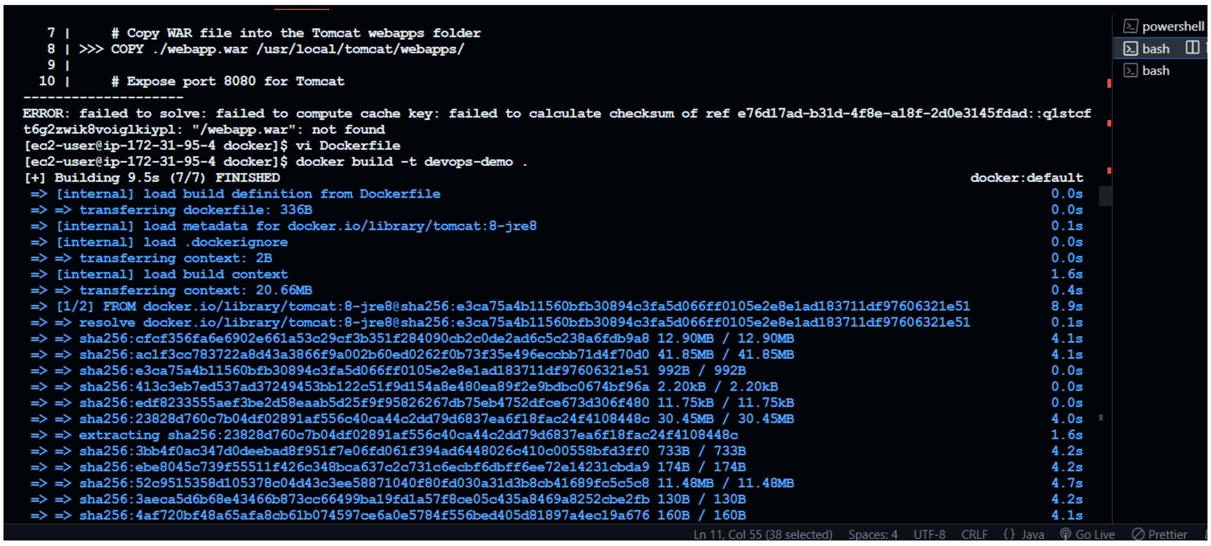
```
BALAJI PC@BALAJI-PC MINGW64 /d/3-1/CD/final-project/key-connection
$ chmod 400 /d/3-1/CD/final-project/key-connection/cd-ssh.pem

BALAJI PC@BALAJI-PC MINGW64 /d/3-1/CD/final-project/key-connection
$ scp -i cd-ssh.pem project/target/project-0.0.1-SNAPSHOT.war ec2-user@100.24.45.118:/home/ec2-user/
project-0.0.1-SNAPSHOT.war                                                              100%   20MB   1.6MB/s   00:12

BALAJI PC@BALAJI-PC MINGW64 /d/3-1/CD/final-project/key-connection
$ scp -i cd-ssh.pem project/target/project-0.0.1-SNAPSHOT.war ec2-user@100.24.45.118:/home/dockeradmin/
scp: dest open "/home/dockeradmin/project-0.0.1-SNAPSHOT.war": Permission denied
scp: failed to upload file project/target/project-0.0.1-SNAPSHOT.war to /home/dockeradmin/

BALAJI PC@BALAJI-PC MINGW64 /d/3-1/CD/final-project/key-connection
$ 
```



```
Last login: Tue Dec  3 14:38:59 2024 from 49.43.217.106
[ec2-user@ip-172-31-95-4 ~]$ ls
Dockerfile  project-0.0.1-SNAPSHOT.war
[ec2-user@ip-172-31-95-4 ~]$ mv project-0.0.1-SNAPSHOT.war /opt/docker
[ec2-user@ip-172-31-95-4 ~]$ ls
Dockerfile
[ec2-user@ip-172-31-95-4 ~]$ cd opt/docker/
-bash: cd: opt/docker/: No such file or directory
[ec2-user@ip-172-31-95-4 ~]$ cd /opt/docker/
[ec2-user@ip-172-31-95-4 docker]$ cd ../..
[ec2-user@ip-172-31-95-4 /]$ ls
bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[ec2-user@ip-172-31-95-4 /]$ cd home/
[ec2-user@ip-172-31-95-4 home]$ ls
dockeradmin  ec2-user
[ec2-user@ip-172-31-95-4 home]$ cd ec2-user/
[ec2-user@ip-172-31-95-4 ~]$ mv Dockerfile /opt/docker
[ec2-user@ip-172-31-95-4 ~]$ cd opt/docker
-bash: cd: opt/docker: No such file or directory
[ec2-user@ip-172-31-95-4 ~]$ cd opt/docker/
-bash: cd: opt/docker/: No such file or directory
[ec2-user@ip-172-31-95-4 ~]$ ls
[ec2-user@ip-172-31-95-4 ~]$ cd /opt/docker/
[ec2-user@ip-172-31-95-4 docker]$ ls
Dockerfile  project-0.0.1-SNAPSHOT.war
[ec2-user@ip-172-31-95-4 docker]$ docker build -t devops-demo .
ERROR: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun
%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-95-4 docker]$ getent group docker
docker:x:992:dockeradmin
```



```
[dockeradmin@ip-172-31-95-4 ~]$ docker build -t devops-demo .
[+] Building 0.1s (1/1) FINISHED                                                             docker:defaul
 => [internal] load build definition from Dockerfile                                                  0.0
 => => transferring dockerfile: 2B                                                                    0.0
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
# Use Tomcat 8 JRE8 as the base image
[dockeradmin@ip-172-31-95-4 ~]$ cd/opt
-bash: cd/opt: No such file or directory
[dockeradmin@ip-172-31-95-4 ~]$ cd ..
[dockeradmin@ip-172-31-95-4 home]$ cd ..
[dockeradmin@ip-172-31-95-4 /]$ ls
bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[dockeradmin@ip-172-31-95-4 /]$ cd home/
[dockeradmin@ip-172-31-95-4 home]$ ls
dockeradmin  ec2-user
[dockeradmin@ip-172-31-95-4 home]$ cd ec2-user/
-bash: cd: ec2-user/: Permission denied
[dockeradmin@ip-172-31-95-4 home]$ exit
logout
[ec2-user@ip-172-31-95-4 docker]$ ls
Dockerfile  project-0.0.1-SNAPSHOT.war
[ec2-user@ip-172-31-95-4 docker]$ cd ../../..
[ec2-user@ip-172-31-95-4 /]$ ls
bin  boot  dev  etc  home  lib  lib64  local  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[ec2-user@ip-172-31-95-4 /]$ cd home/
[ec2-user@ip-172-31-95-4 home]$ cd dockeradmin/
-bash: cd: dockeradmin/: Permission denied
[ec2-user@ip-172-31-95-4 home]$ su - dockeradmin
Password:
Last login: Tue Dec  3 15:09:32 UTC 2024 on pts/3
[dockeradmin@ip-172-31-95-4 ~]$ ls
```



```
   7 |     # Copy WAR file into the Tomcat webapps folder
   8 | >>> COPY ./webapp.war /usr/local/tomcat/webapps/
   9 |
  10 |     # Expose port 8080 for Tomcat
--------------------
ERROR: failed to solve: failed to compute cache key: failed to calculate checksum of ref e76d17ad-b31d-4f8e-a18f-2d0e3145fdad::q1stcf
t6g2zwik8voiglkiypl: "/webapp.war": not found
[ec2-user@ip-172-31-95-4 docker]$ vi Dockerfile
[ec2-user@ip-172-31-95-4 docker]$ docker build -t devops-demo .
[+] Building 9.5s (7/7) FINISHED                                                                      docker:default
 => [internal] load build definition from Dockerfile                                                           0.0s
 => => transferring dockerfile: 336B                                                                           0.0s
 => [internal] load metadata for docker.io/library/tomcat:8-jre8                                               0.1s
 => [internal] load .dockerignore                                                                              0.0s
 => => transferring context: 2B                                                                                0.0s
 => [internal] load build context                                                                              1.6s
 => => transferring context: 20.66MB                                                                           0.4s
 => [1/2] FROM docker.io/library/tomcat:8-jre8@sha256:e3ca75a4b11560bfb30894c3fa5d066ff0105e2e8e1ad183711df97606321e51   8.9s
 => => resolve docker.io/library/tomcat:8-jre8@sha256:e3ca75a4b11560bfb30894c3fa5d066ff0105e2e8e1ad183711df97606321e51   0.1s
 => => sha256:cfcf356fa6e6902e661a53c29cf3b351f284090cb2c0de2ad6c5c238a6fdb9a8 12.90MB / 12.90MB               4.1s
 => => sha256:ac1f3cc783722a8d43a3866f9a002b60ed0262f0b73f35e496eccbb71d4f70d0 41.85MB / 41.85MB               4.1s
 => => sha256:e3ca75a4b11560bfb30894c3fa5d066ff0105e2e8e1ad183711df97606321e51 992B / 992B                     0.0s
 => => sha256:413c3eb7ed537ad37249453bb122c51f9d154a8e480ea89f2e9bdbc0674bf96a 2.20kB / 2.20kB                 0.0s
 => => sha256:edf8233555aef3be2d58eaab5d25f9f95826267db75eb4752dfce673d306f480 11.75kB / 11.75kB               0.0s
 => => sha256:23828d760c7b04df02891af556c40ca44c2dd79d6837ea6f18fac24f4108448c 30.45MB / 30.45MB               4.0s
 => => extracting sha256:23828d760c7b04df02891af556c40ca44c2dd79d6837ea6f18fac24f4108448c                      1.6s
 => => sha256:3bb4f0ac347d0deebad8f951f7e06fd061f394ad6448026c410c00558bfd3ff0 733B / 733B                     4.2s
 => => sha256:ebe8045c739f55511f426c348bca637c2c731c6ecbf6dbff6ee72e14231cbda9 174B / 174B                     4.2s
 => => sha256:52c9515358d105378c04d43c3ee58871040f80fd030a31d3b8cb41689fc5c5c8 11.48MB / 11.48MB               4.7s
 => => sha256:3aeca5d6b68e43466b873cc66499ba19fd1a57f8ce05c435a8469a8252cbe2fb 130B / 130B                     4.2s
 => => sha256:4af720bf48a65afa8cb61b074597ce6a0e5784f556bed405d81897a4ec19a676 160B / 160B                     4.1s
```

powershell
bash
bash
Ln 11, Col 55 (38 selected)   Spaces: 4   UTF-8   CRLF   {} Java   Go Live   Prettier

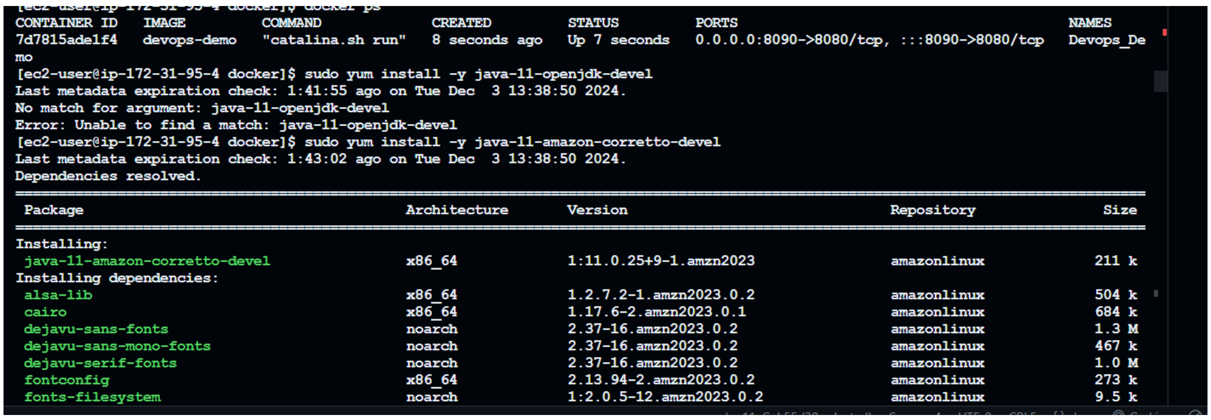
Docker file content

# Pull base image FROM tomcat:8-jre8

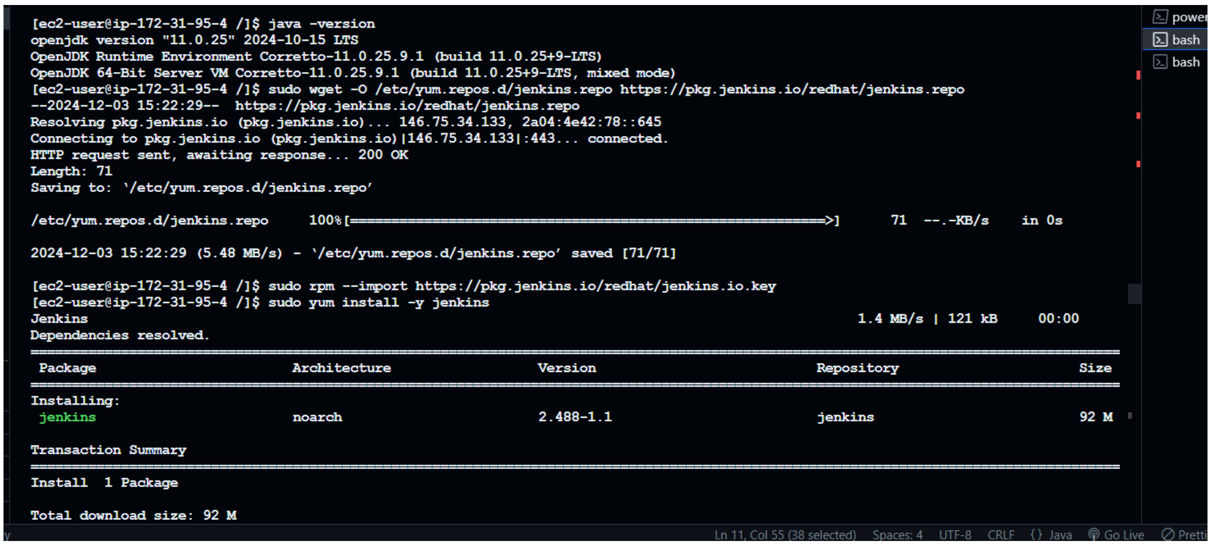
# Maintainer MAINTAINER "your_username"

 # Copy WAR file onto container COPY ./webapp.war /usr/local/tomcat/webapps

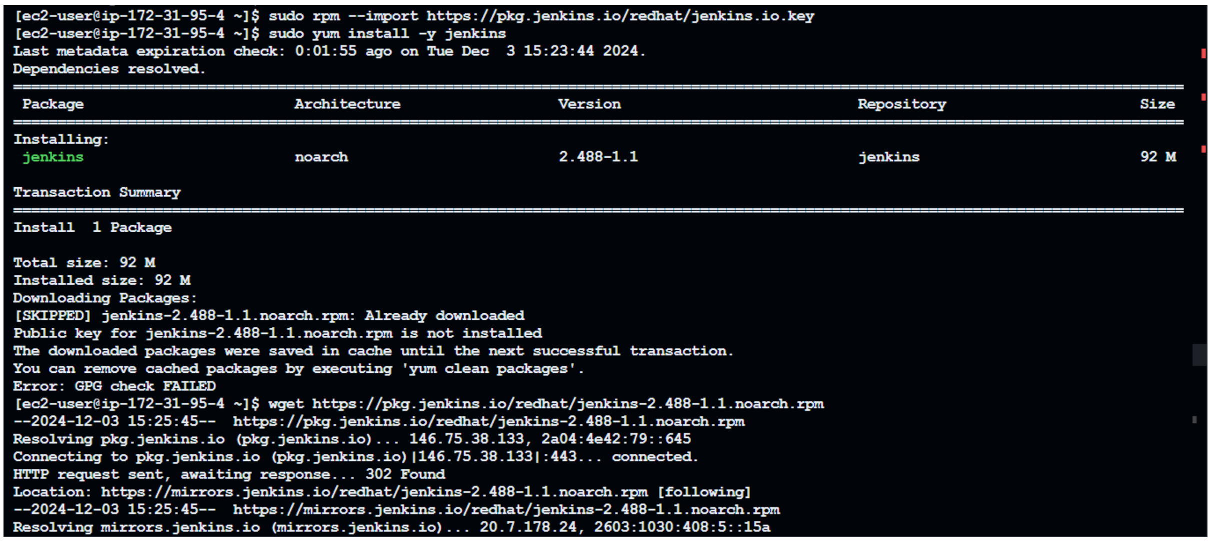
Step-5 : Install and Configure Jenkins



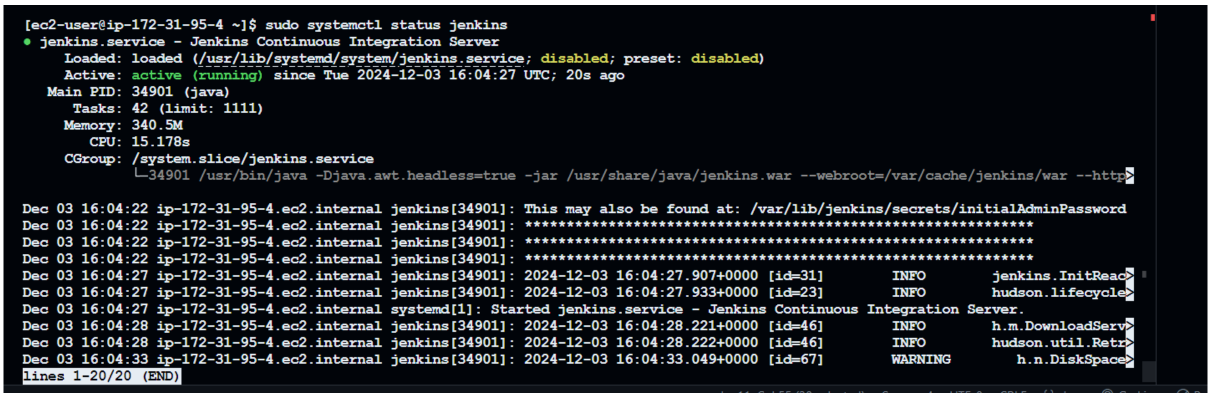
```
[ec2-user@ip-172-31-95-4 docker]$ docker ps
CONTAINER ID   IMAGE         COMMAND             CREATED         STATUS         PORTS                                       NAMES
7d7815ade1f4   devops-demo   "catalina.sh run"   8 seconds ago   Up 7 seconds   0.0.0.0:8090->8080/tcp, :::8090->8080/tcp   Devops_De
mo
[ec2-user@ip-172-31-95-4 docker]$ sudo yum install -y java-11-openjdk-devel
Last metadata expiration check: 1:41:55 ago on Tue Dec  3 13:38:50 2024.
No match for argument: java-11-openjdk-devel
Error: Unable to find a match: java-11-openjdk-devel
[ec2-user@ip-172-31-95-4 docker]$ sudo yum install -y java-11-amazon-corretto-devel
Last metadata expiration check: 1:43:02 ago on Tue Dec  3 13:38:50 2024.
Dependencies resolved.
=====================================================================================================================================
 Package                                    Architecture         Version                                 Repository            Size
=====================================================================================================================================
Installing:
 java-11-amazon-corretto-devel              x86_64               1:11.0.25+9-1.amzn2023                  amazonlinux          211 k
Installing dependencies:
 alsa-lib                                   x86_64               1.2.7.2-1.amzn2023.0.2                  amazonlinux          504 k
 cairo                                      x86_64               1.17.6-2.amzn2023.0.1                   amazonlinux          684 k
 dejavu-sans-fonts                          noarch               2.37-16.amzn2023.0.2                    amazonlinux          1.3 M
 dejavu-sans-mono-fonts                     noarch               2.37-16.amzn2023.0.2                    amazonlinux          467 k
 dejavu-serif-fonts                         noarch               2.37-16.amzn2023.0.2                    amazonlinux          1.0 M
 fontconfig                                 x86_64               2.13.94-2.amzn2023.0.2                  amazonlinux          273 k
 fonts-filesystem                           noarch               1:2.0.5-12.amzn2023.0.2                 amazonlinux          9.5 k
```



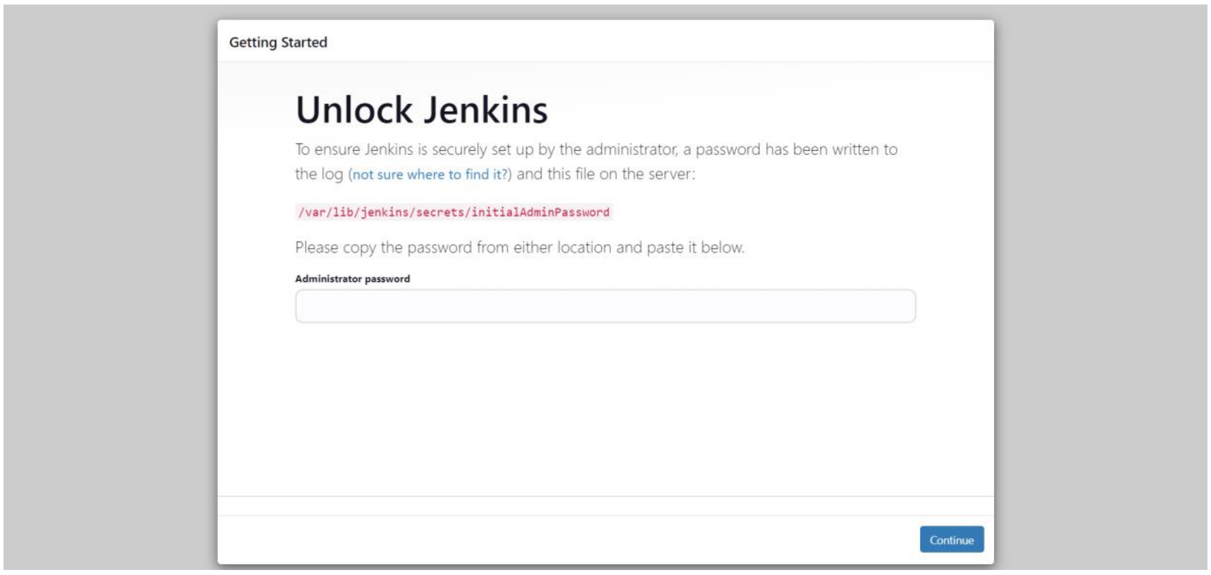
```
[ec2-user@ip-172-31-95-4 /]$ java -version
openjdk version "11.0.25" 2024-10-15 LTS
OpenJDK Runtime Environment Corretto-11.0.25.9.1 (build 11.0.25+9-LTS)
OpenJDK 64-Bit Server VM Corretto-11.0.25.9.1 (build 11.0.25+9-LTS, mixed mode)
[ec2-user@ip-172-31-95-4 /]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat/jenkins.repo
--2024-12-03 15:22:29--  https://pkg.jenkins.io/redhat/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 71
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.repo     100%[===================================================>]      71  --.-KB/s    in 0s

2024-12-03 15:22:29 (5.48 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [71/71]

[ec2-user@ip-172-31-95-4 /]$ sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key
[ec2-user@ip-172-31-95-4 /]$ sudo yum install -y jenkins
Jenkins                                                                                         1.4 MB/s | 121 kB     00:00
Dependencies resolved.
===================================================================================================================================
 Package                         Architecture                    Version                           Repository                  Size
===================================================================================================================================
Installing:
 jenkins                         noarch                          2.488-1.1                         jenkins                     92 M

Transaction Summary
===================================================================================================================================
Install  1 Package

Total download size: 92 M
```



```
[ec2-user@ip-172-31-95-4 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key
[ec2-user@ip-172-31-95-4 ~]$ sudo yum install -y jenkins
Last metadata expiration check: 0:01:55 ago on Tue Dec  3 15:23:44 2024.
Dependencies resolved.
==================================================================================================================================
 Package                       Architecture                  Version                            Repository                   Size
==================================================================================================================================
Installing:
 jenkins                       noarch                        2.488-1.1                          jenkins                      92 M

Transaction Summary
==================================================================================================================================
Install  1 Package

Total size: 92 M
Installed size: 92 M
Downloading Packages:
[SKIPPED] jenkins-2.488-1.1.noarch.rpm: Already downloaded
Public key for jenkins-2.488-1.1.noarch.rpm is not installed
The downloaded packages were saved in cache until the next successful transaction.
You can remove cached packages by executing 'yum clean packages'.
Error: GPG check FAILED
[ec2-user@ip-172-31-95-4 ~]$ wget https://pkg.jenkins.io/redhat/jenkins-2.488-1.1.noarch.rpm
--2024-12-03 15:25:45--  https://pkg.jenkins.io/redhat/jenkins-2.488-1.1.noarch.rpm
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.38.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.38.133|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://mirrors.jenkins.io/redhat/jenkins-2.488-1.1.noarch.rpm [following]
--2024-12-03 15:25:45--  https://mirrors.jenkins.io/redhat/jenkins-2.488-1.1.noarch.rpm
Resolving mirrors.jenkins.io (mirrors.jenkins.io)... 20.7.178.24, 2603:1030:408:5::15a
```



```
[ec2-user@ip-172-31-95-4 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; preset: disabled)
     Active: active (running) since Tue 2024-12-03 16:04:27 UTC; 20s ago
   Main PID: 34901 (java)
      Tasks: 42 (limit: 1111)
     Memory: 340.5M
        CPU: 15.178s
     CGroup: /system.slice/jenkins.service
             └─34901 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --http>

Dec 03 16:04:22 ip-172-31-95-4.ec2.internal jenkins[34901]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Dec 03 16:04:22 ip-172-31-95-4.ec2.internal jenkins[34901]: *************************************************************
Dec 03 16:04:22 ip-172-31-95-4.ec2.internal jenkins[34901]: *************************************************************
Dec 03 16:04:22 ip-172-31-95-4.ec2.internal jenkins[34901]: *************************************************************
Dec 03 16:04:27 ip-172-31-95-4.ec2.internal jenkins[34901]: 2024-12-03 16:04:27.907+0000 [id=31]        INFO        jenkins.InitReac>
Dec 03 16:04:27 ip-172-31-95-4.ec2.internal jenkins[34901]: 2024-12-03 16:04:27.933+0000 [id=23]        INFO        hudson.lifecycle>
Dec 03 16:04:27 ip-172-31-95-4.ec2.internal systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Dec 03 16:04:28 ip-172-31-95-4.ec2.internal jenkins[34901]: 2024-12-03 16:04:28.221+0000 [id=46]        INFO        h.m.DownloadServ>
Dec 03 16:04:28 ip-172-31-95-4.ec2.internal jenkins[34901]: 2024-12-03 16:04:28.222+0000 [id=46]        INFO        hudson.util.Retr>
Dec 03 16:04:33 ip-172-31-95-4.ec2.internal jenkins[34901]: 2024-12-03 16:04:33.049+0000 [id=67]        WARNING        h.n.DiskSpace>
lines 1-20/20 (END)
```



Getting Started

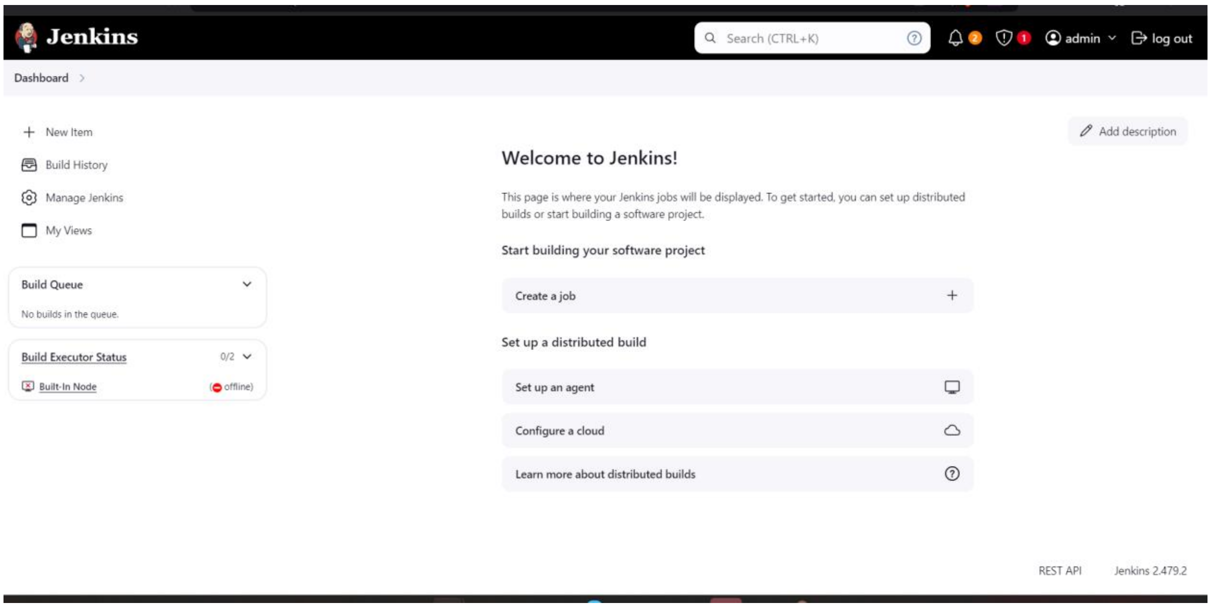
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password



Build Queue

No builds in the queue.

Build Executor Status 0/2

Built-In Node (offline)

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

REST API   Jenkins 2.479.2

CONCLUSION

This DevOps project successfully implemented continuous integration and continuous deployment (CI/CD) practices to automate the build, test, and deployment pipelines. By leveraging tools such as Jenkins, Docker, and AWS, the project streamlined development processes, reducing manual intervention and improving code quality through automated testing and integration.

Challenges were encountered in configuring and integrating various systems, particularly in setting up Jenkins on AWS EC2, managing Java versions, and ensuring seamless communication between different microservices. These issues were addressed by troubleshooting and reconfiguring the environment, leading to improved system stability and performance.

The implementation of automated deployment ensured quicker and more reliable delivery of applications to production, facilitating faster iterations and reducing the time-to-market for new features and updates. The project demonstrated the effectiveness of DevOps practices in improving collaboration between development and operations teams and in ensuring a more efficient, scalable, and robust software delivery process.

Overall, this DevOps implementation has provided a solid foundation for future improvements and scaling of the project, contributing to increased efficiency, productivity, and collaboration within the development lifecycle.