

Tested By Kalaivani Ravishankar

Vani212008@gmail.com

<https://github.com/vanishankar/Newrepo.git>

PG DO - DevOps Capstone Project

| TOPICS | PAGE NO |
|---|---------|
| Introduction | 2 |
| Application | 2 |
| Tools used | 2 |
| Steps to create ec2-instance | 2 |
| MobaXterm | 5 |
| Install Git on Docker_server and Jenkins server | 5 |
| GitHub Repo | 6 |
| Installing Docker | 6 |
| Creating user | 6 |
| Add a user to docker group to manage docker | 6 |
| SSHKEY | 7 |
| configured Jenkins server | 7 |
| Accessing Jenkins | 8 |
| Configure java and maven path | 8 |
| Create a Docker file | 11 |
| Setting Up Environment in jenkins | 12 |
| Plugins | 12 |
| CREATING A JOB | 13 |
| Console output | 17 |
| Test result | 19 |
| Access from browser | 20 |
| Docker hub | 21 |
| Concepts used | 21 |
| Docker Troubleshooting | 21 |
| Deleted and pruned images | 21 |
| Conclusion | 22 |

Create a CI/CD Pipeline to convert the legacy development process to a DevOps process.

Introduction:

In this project I will apply the skills and knowledge which were developed throughout Post Graduate Program in DevOps

- Working in AWS
- Docker
- Git, GitHub,
- Jenkins,
- Tomcat Apache and
- Maven

Application:

The application I used sample java application. I used ec2 instance to create Docker server and Jenkins server

Tools used:

1. EC2
2. Jenkins
3. Docker
4. Git

Steps to create ec2-instance:

Step 1: Login to the aws console: -

I use this link <https://console.aws.amazon.com/ec2/> to login to aws console account.

Step 2: Choose Launch Instance

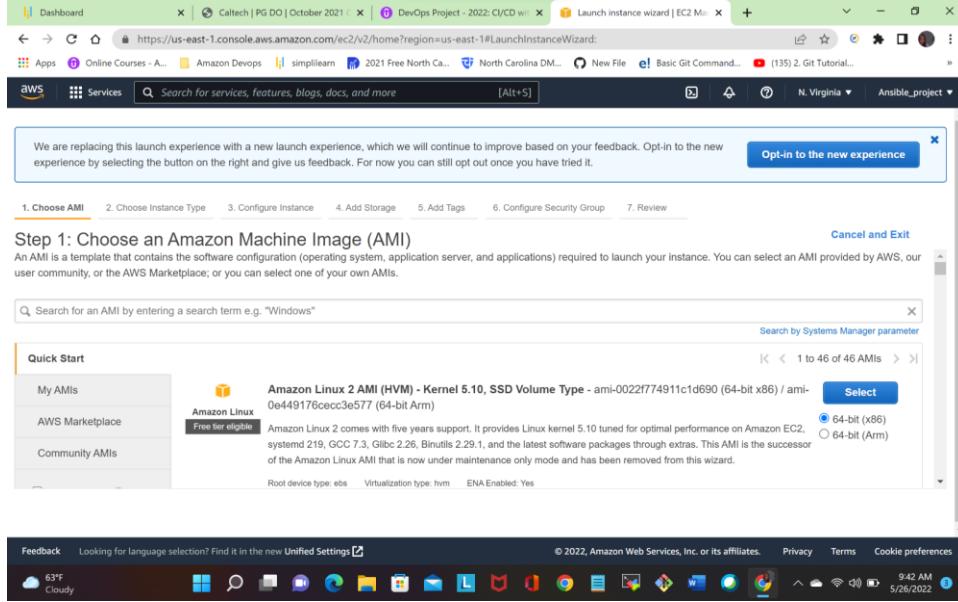
First of all, click on 'Launch Instance' button shows in the below picture for launch/create new ec2 instance in aws:

The screenshot shows the AWS EC2 Instances page. The sidebar on the left has 'Instances' selected. The main area displays a table of instances with the following data:

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|----------------|---------------------|----------------|---------------|--------------|--------------|
| jenkins | i-006493c979cecc41 | Stopped | t2.micro | - | No alarms |
| dockerservice | i-0b3834b0a833102f6 | Stopped | t2.micro | - | No alarms |
| Tomcatserver | i-0beb71a3abe067d0f | Stopped | t2.micro | - | No alarms |
| Docker_server | i-0a3a1d05b413712a1 | Stopped | t2.micro | - | No alarms |
| jenkins_server | i-01ddc0489cf68d06 | Stopped | t2.micro | - | No alarms |

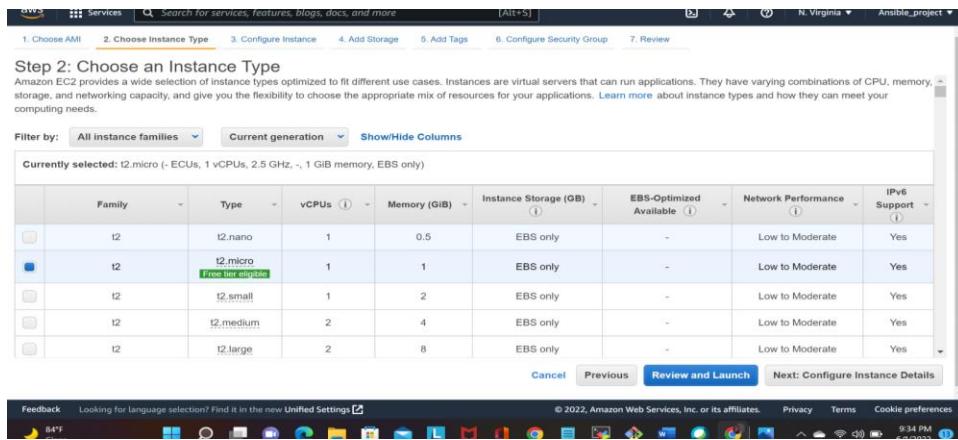
In this step, choose AMI (I chose Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0022f774911c1d690 (64-bit x86) / ami-0e449176cecc3e577 (64-bit Arm))

Shown in the below fig

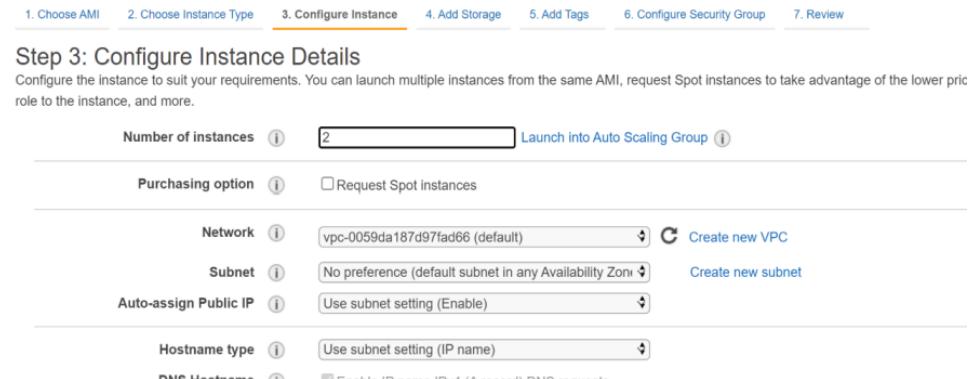


Step 4: Choose EC2 Instance Types

In this step, choose ec2 instance type shown in below picture:

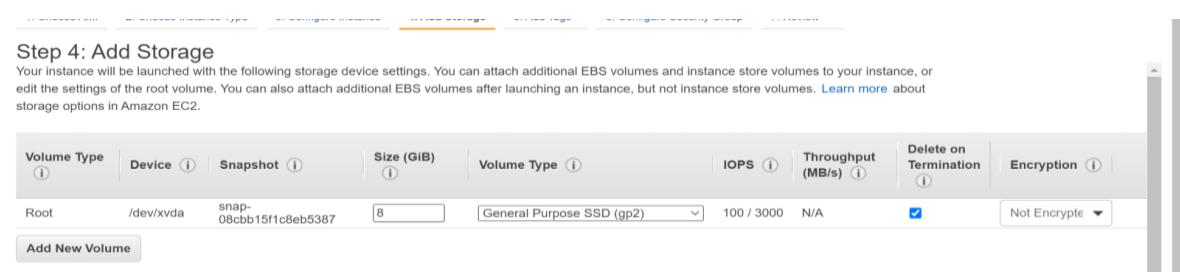


Step 5: Configure Instance Detail:



Step 6: Add Storage of Ec2 Instance

In this step, I chose storage. By default, ec2 t2-micro provide 8gb ssd. Shown in the below picture:



Step 7: Tag Instance of Ec2 Instance

In this step, I added tag of instance with a key-value pair

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

| Key | (128 characters maximum) | Value | (256 characters maximum) | Instances | Volumes | Network Interfaces |
|------|--------------------------|-------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Name | Tomcat_deploymentserver | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Add another tag (Up to 50 tags maximum)

Step 8: Configure Security Groups

In this next step of configuring Security Groups

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security group name: capstoneproject

Description: capstoneproject

| Type | Protocol | Port Range | Source | Description |
|--------------|----------|------------|------------------|----------------------------|
| SSH | TCP | 22 | Custom 0.0.0.0/0 | e.g. SSH for Admin Desktop |
| Custom TCP F | TCP | 8080-9000 | Custom 0.0.0.0/0 | e.g. SSH for Admin Desktop |

Add Rule

Warning
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.

Cancel Previous Review and Launch

I created security group called capstone project and Defining protocols which I want enabled on my instance

Step 9: Once, the firewall rules are set- Review and launch

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, capstoneproject, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0022f774911c1d690

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is no longer supported.

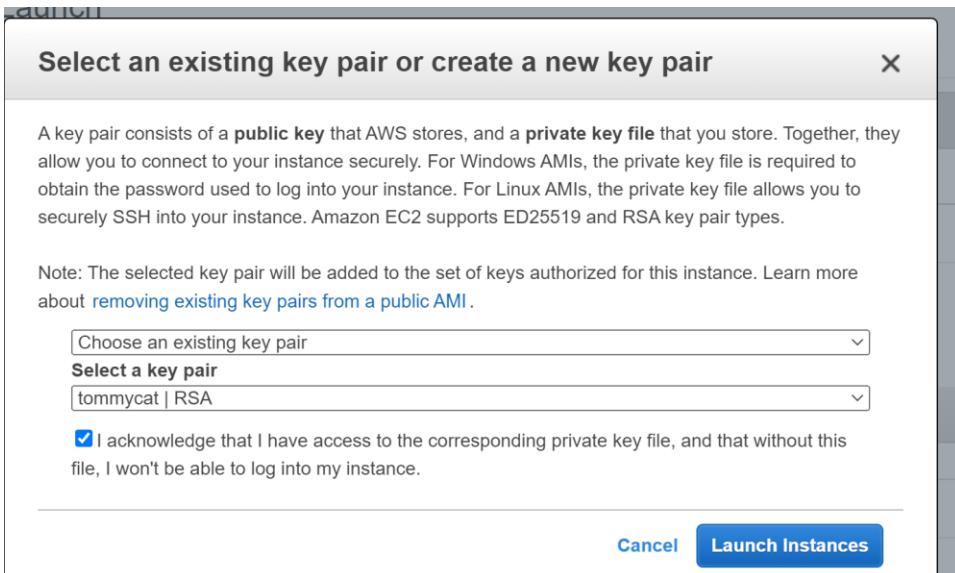
Instance Type

| Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|------|-------|--------------|-----------------------|-------------------------|---------------------|
| t2.micro | - | 1 | 1 | EBS only | - | Low to Moderate |

Edit Instance type

Cancel Previous Launch

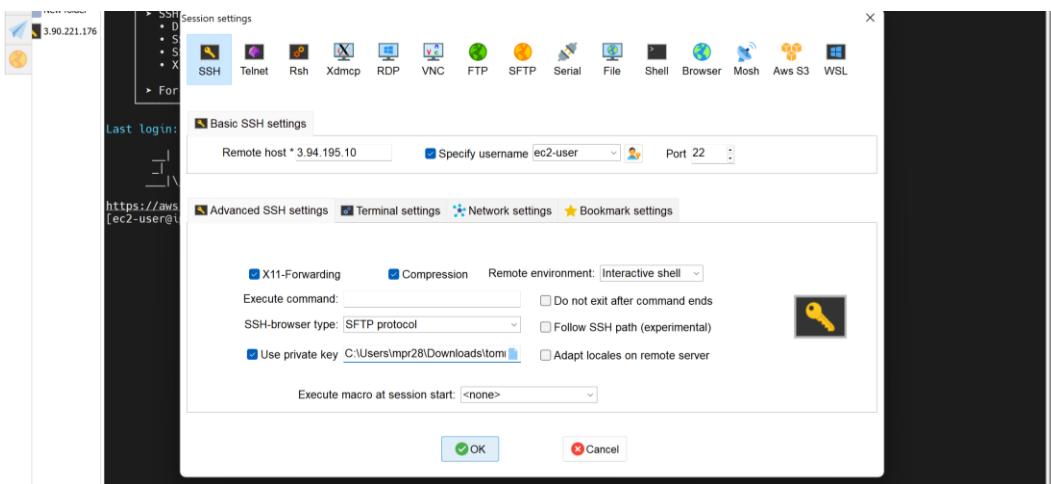
Step 10: Create Key-Pair for Instance Access: Creating a keypair for ec2 instance, (I used existing keypair) create new key pair and add the name of this key. Then download and save it in your secured folder. Shown in below fig.



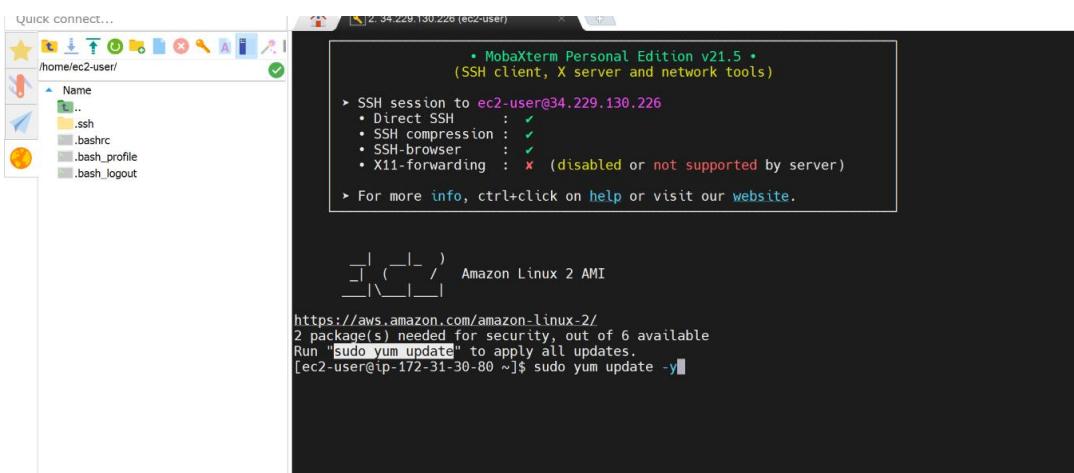
Use pem file in MobaXterm to connect ec2 machine in AWS: -

I used MobaXterm to connect ec2 machine. I downloaded mobaxterm .go to the session settings to create a new session, fill up the remote host and username. And click advance setting and select use private key, select your private key.

It will help to connect to ec2-instance. As shown in the below fig



After clicking ok it taking into page shown in below fig



I created two ec2 instance machine (Jenkins and Docker server)

Install Git on Docker_server and Jenkins server

Using following command

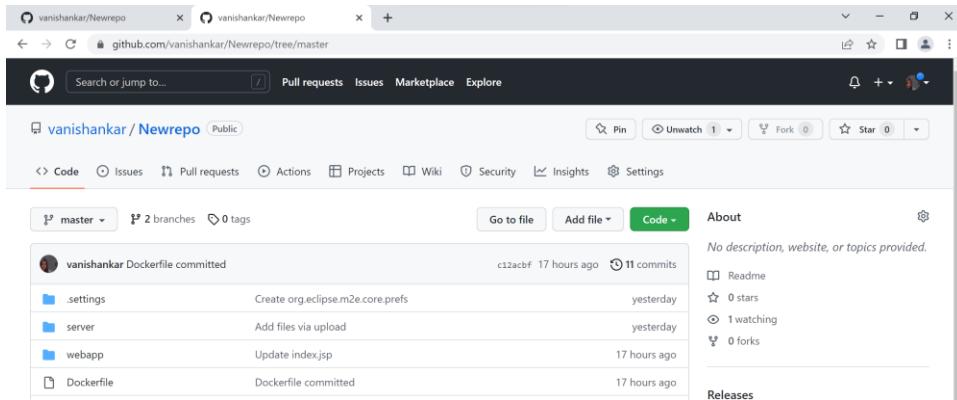
[yum install git -y](#)

Complete!

```
[root@ip-172-31-24-109 ~]# git --version
git version 2.32.0
```

GitHub Repo

- Create a GitHub project and check in the code of Aetna project. Also, we will be checking in the Docker file.
- The link to the public GitHub repo is <https://github.com/vanishankar/Newrepo.git>



Installing Docker:

Command to install docker

Yum install docker.

The following pic shows docker version

```
git version 2.32.0
[root@ip-172-31-24-109 ~]# docker --version
Docker version 20.10.13, build a224086
[root@ip-172-31-24-109 ~]#
```

Command to start docker and check the status:

```
service docker start
service docker status
```

Creating user:

I created user called admin in both the server using following commands

```
useradd admin
```

passwd admin (Shown in below fig)

```
Docker version 20.10.13, build a224086
[root@ip-172-31-24-109 ~]# useradd admin
[root@ip-172-31-24-109 ~]# passwd admin
Changing password for user admin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-24-109 ~]#
```

```
[root@ip-172-31-19-44 ~]# useradd admin
[root@ip-172-31-19-44 ~]# passwd admin
Changing password for user admin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-19-44 ~]#
```

Add a user to docker group to manage docker: -

I added a user in docker group using following command

```
usermod -aG docker admin
```

Adding Users to Sudoers File:

Next step is to add admin user into sudoers file using following commands

sudo visudo

Shown in the following fig

```
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)        ALL  
admin   ALL=(ALL)        NOPASSWD: ALL  
#  
## Allows members of the 'sys' group to run networking, software,
```

The place where you see root ALL=(ALL:ALL) ALL, that's where we'll be amending our user name. And next use following command to configure sshd_config file to enable password less authentication, show in the following

Enable Password Authentication in below File Path

vi /etc/ssh/sshd_config. uncomment the password authentication and change no to yes shown in following fig

```
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication yes  
#PermitEmptyPasswords no  
PasswordAuthentication yes
```

Reload Service with the command

service sshd reload

```
[root@ip-172-31-24-109 ~]# visudo  
[root@ip-172-31-24-109 ~]# vi /etc/ssh/sshd_config  
[root@ip-172-31-24-109 ~]# service sshd restart  
Redirecting to /bin/systemctl restart sshd.service  
[root@ip-172-31-24-109 ~]#
```

And I followed same steps in Jenkins server to add user and enable password less authentication.

Next steps are to create a connection between docker machine and Jenkins by downloading sshkey ,using following command .login as an admin using following command

sudo su – admin

And generate sshkey using command ssh-keygen

After generating I copied the public key to Jenkins server using following command ssh-copy-id
admin@172-31-24-109 (ssh-copy-id user@<target-server> - Authorized_keys)

```
[admin@ip-172-31-24-109 ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/admin/.ssh/id_rsa):  
Created directory '/home/admin/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/admin/.ssh/id_rsa.  
Your public key has been saved in /home/admin/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:QYd4e2c97yYY+537WfzgAP2ei9c/Xo9yJJLrYpXhq0 admin@ip-172-31-24-109.ec2.internal  
The key's randomart image is:  
+---[RSA 2048]---+  
| ..0... |  
| ..0... |  
| ..0 |  
| 0 o |  
| S * . |  
| + 0.... |  
| +0oB + |  
| .E.+0 =X|  
| ...o++=*0|  
+---+  
[admin@ip-172-31-24-109 ~]$
```

And next step is to configured Jenkins server to install Jenkins, Maven, java

JAVA

<http://openjdk.java.net/install/>

Install Java in Linux

yum install java-1.8.0-openjdk-devel

Java is installed in usr/lib directory

Set Java Home Path in Bash profile

JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64

PATH=\$PATH:\$HOME/bin:\$JAVA_HOME:

Install maven

Download maven packages <https://maven.apache.org/download.cgi> onto Jenkins server. In this case, I am using /opt/maven as my installation directory

```
Last login: Sat Jun 4 19:50:20 UTC 2022 on pts/0
[root@ip-172-31-19-44 ~]# mvn -v
Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)
Maven home: /opt/apache-maven-3.8.5
Java version: 1.8.0_312, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.10.112-108.499.amzn2.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-19-44 ~]#
```

I used wget command to download maven (wget is used to download files from an HTTP/HTTPS or FTP server)

wget <https://dlcdn.apache.org/maven/maven-3/3.8.5/binaries/apache-maven-3.8.5-bin.tar.gz>

Next step is to extract the archive using following command

tar xzvf apache-maven-3.8.5-bin.tar.gz

Maven home path in bash profile

JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64

MAVEN_HOME=/opt/apache-maven-3.8.5

M2=/opt/apache-maven-3.8.5/bin

PATH=\$PATH:\$HOME/bin:\$JAVA_HOME:\$MAVEN_HOME:\$M2

The following fig shows maven and java path in .bash_profile

```
# User specific environment and startup program

JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64

MAVEN_HOME=/opt/apache-maven-3.8.5
M2=/opt/apache-maven-3.8.5/bin
PATH=$PATH:$HOME/bin:$JAVA_HOME:$MAVEN_HOME:$M2
```

Install Jenkins:-

Install Jenkins using the rpm or by setting up the Repo

I downloaded the latest Version of Jenkins form <https://pkg.jenkins.io/redhat-stable/> and install

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
amazon-linux-extras install epel
```

After downloading Start the Jenkins Services using following command

Service jenkins start

(Shown in following Fig.)

```
Starting jenkins (via systemctl):
[root@ip-172-31-19-44 ~]# service jenkins status
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
  Active: active (running) since Sun 2022-06-05 16:51:07 UTC; 54s ago
    Main PID: 9133 (java)
       CGroup: /system.slice/jenkins.service
              └─9133 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Jun 05 16:51:03 ip-172-31-19-44.ec2.internal jenkins[9133]: 2022-06-05 16:51:03.903+0000 [id=28]      INFO      jenkins.InitReacto...
Jun 05 16:51:05 ip-172-31-19-44.ec2.internal jenkins[9133]: 2022-06-05 16:51:05.579+0000 [id=28]      INFO      jenkins.InitReacto...
Jun 05 16:51:06 ip-172-31-19-44.ec2.internal jenkins[9133]: 2022-06-05 16:51:06.933+0000 [id=27]      INFO      jenkins.InitReacto...
```

Accessing Jenkins

By default, jenkins runs at port 8080, You can access jenkins at

<http://YOUR-SERVER-PUBLIC-IP:8080>

Configure Jenkins:

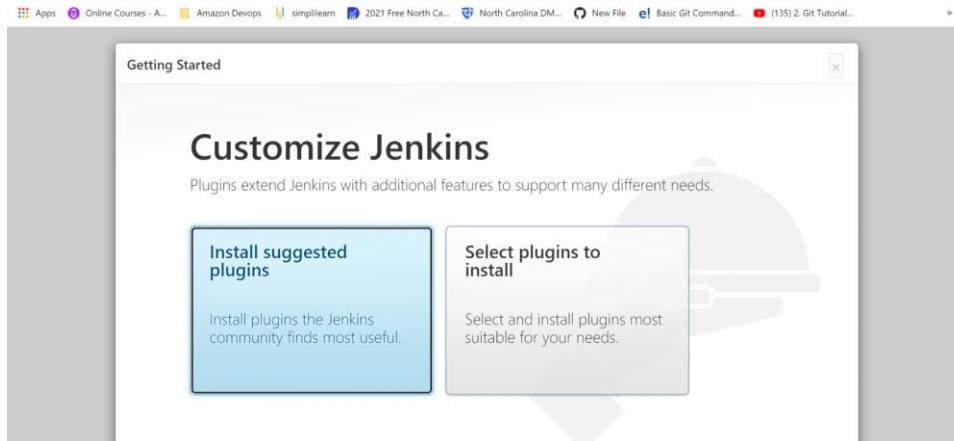
Grab the default password

Password Location:/var/lib/jenkins/secrets/initialAdminPassword, paste it

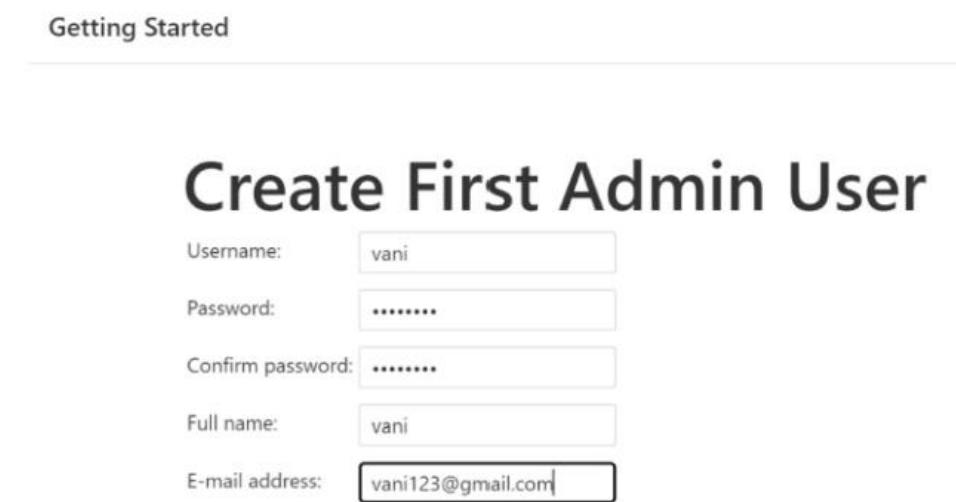
It is shown in following fig



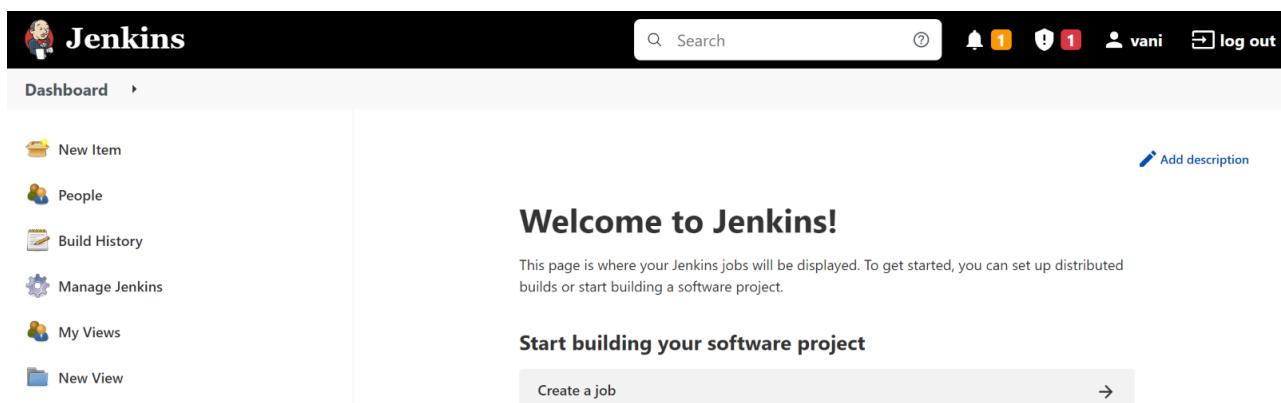
Choose suggested plugins. Shown in below fig



And enter username, password email address shown in below fig



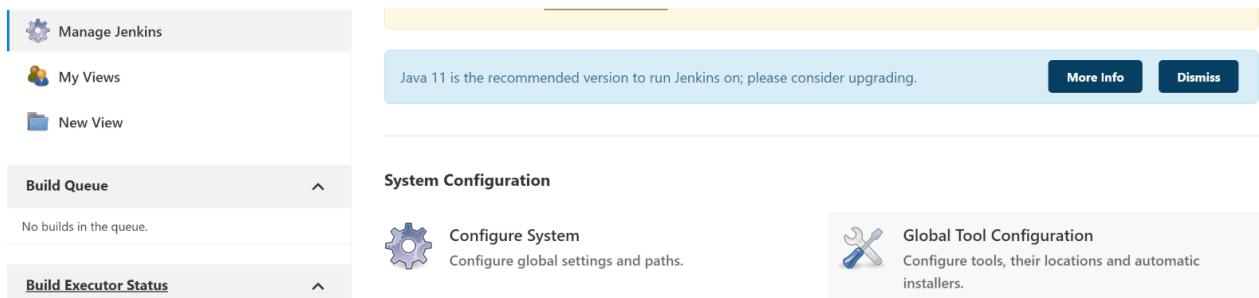
Click save and continue.it taken you into **jenkins dashboard**



The screenshot shows the Jenkins dashboard. At the top, there's a navigation bar with icons for search, help, notifications (1), and user 'vani'. Below the bar, a sidebar on the left lists options: New Item, People, Build History, Manage Jenkins, My Views, and New View. The main content area features a 'Welcome to Jenkins!' message with a sub-section 'Start building your software project' and a 'Create a job' button. A note at the bottom says 'Java 11 is the recommended version to run Jenkins on; please consider upgrading.' with 'More Info' and 'Dismiss' buttons.

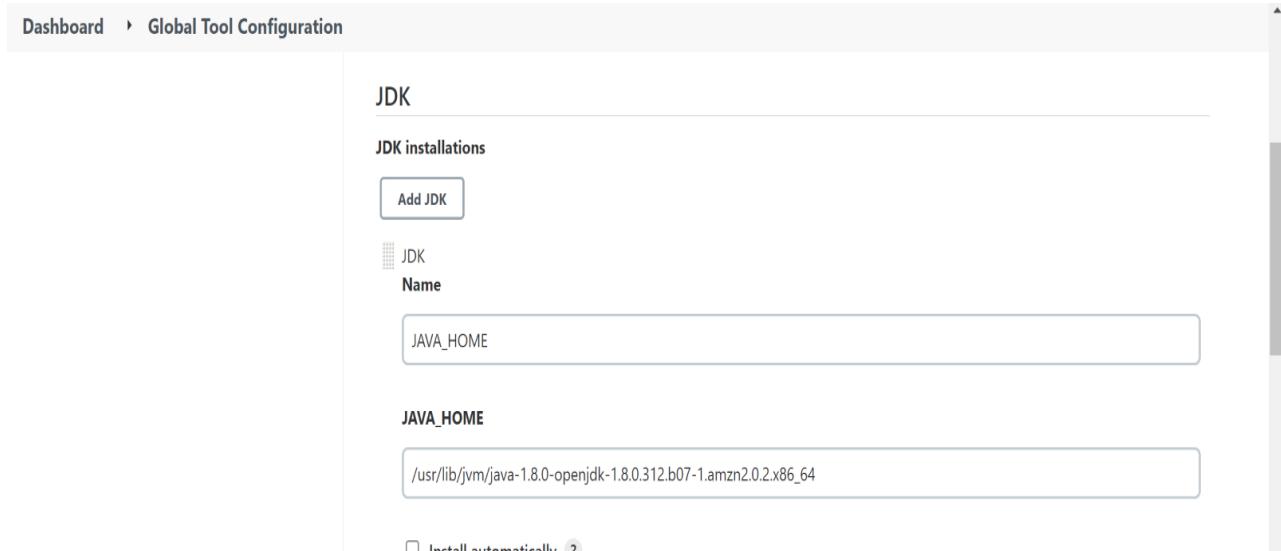
Configure java and maven path

Manage Jenkins > Global Tool Configuration > JDK



This screenshot shows the 'System Configuration' section of the Global Tool Configuration. It includes a 'Configure System' link for global settings and a 'Global Tool Configuration' link for tools like Java and Maven. A note at the top says 'Java 11 is the recommended version to run Jenkins on; please consider upgrading.' with 'More Info' and 'Dismiss' buttons.

Click add JDK and give JDK name as JAVA_HOME and path of JDK



This screenshot shows the 'JDK installations' configuration page. It has a 'Name' field set to 'JAVA_HOME' and a 'Path' field set to '/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64'. There's also an unchecked checkbox for 'Install automatically'.

Next configure maven path, click maven give name for maven as MAVEN_HOME And path of maven

Maven

Maven installations

Add Maven



Maven

Name

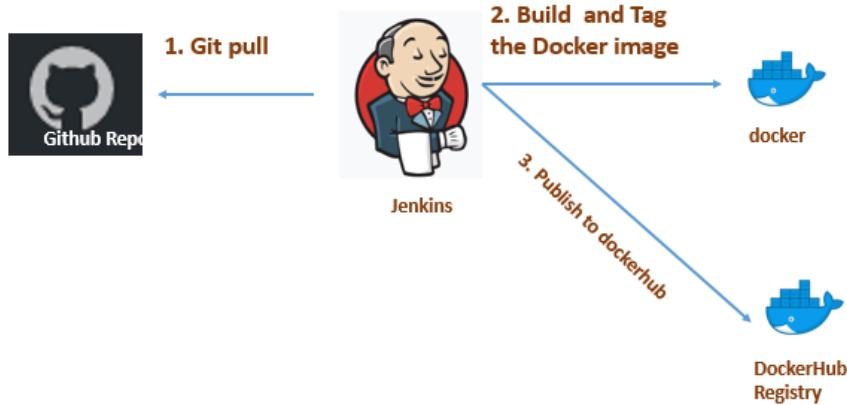
MAVEN_HOME

MAVEN_HOME

/opt/apache-maven-3.8.5

After finishing all configure setting we are going to push the docker image into docker hub using jenkins.

Publish Docker image to Docker Hub using Jenkins: -



For creating docker image we need to write a **Dockerfile**. Go to the docker server, login as dockeradmin user using following command

```
sudo su – dockeradmin
```

Create a Docker file using vi editor

The Dockerfile added to the project will use a prebuilt tomcat image. Over this image, it will add the war file that was built inside the container to the tomcat webapps directory so that microservice api can be accessed through any web browser.

Dockerfile:

```
vi Dockerfile
```

```
FROM tomcat:latest
```

```
COPY ./webapp.war /usr/local/tomcat/webapps
```

```
RUN cp -r /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
```

```
FROM tomcat:latest
COPY ./webapp.war /usr/local/tomcat/webapps
RUN cp -r /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
~
~
~
~
~
~
```

After docker file is created, I pushed into GitHub

| | | | | | |
|--|-------------|----------------------------------|---------|--------------|------------|
| | vanishankar | Dockerfile committed | c12acbf | 17 hours ago | 11 commits |
| | .settings | Create org.eclipse.m2e.coreprefs | | yesterday | |
| | server | Add files via upload | | yesterday | |
| | webapp | Update index.jsp | | 17 hours ago | |
| | Dockerfile | Dockerfile committed | | 17 hours ago | |

LOGIN TO DOCKERHUB

Next **login to dockerhub** using following command

```
docker login,
```

I Gave username and password

Shown in fig

```

tomcat      latest   c795915cb678  7 days ago  680MB
[admin@ip-172-31-24-109 ~]$ docker login
Login with your Docker ID to push and pull images from Docker Hub.
one.
Username: vanishankar
Password:
WARNING! Your password will be stored unencrypted in /home/admin/.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#creden
>Login Succeeded
[admin@ip-172-31-24-109 ~]$ 

```

After I moved to browser (browse <http://YOUR-SERVER-PUBLIC-IP:8080>) And configure jenkins

Setting Up Environment in jenkins

Install Maven Integration plugin and publish over ssh

This plugin is required to used integrate maven (maven integration) and publish over ssh is used to Send build artifacts over SSH.

Click on Manage Jenkins -> Manage Plugins-> search for maven integration and publish over ssh -> Install it without restart. Shown in below fig

The screenshot shows the Jenkins dashboard with the 'Manage Jenkins' link highlighted in the sidebar. Below it, the 'Plugin Manager' section is open, showing the 'Available' tab. Two specific plugins are listed:

- Maven Integration**: Version 3.18, Build Tools, Last updated 2 months 26 days ago. Description: This plug-in provides a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).
- Publish Over SSH**: Version 1.24, Artifact Uploaders, Build Tools, Last updated 3 months 3 days ago. Description: Send build artifacts over SSH.

At the bottom of the plugin list, there are buttons for 'Install without restart' and 'Download now and install after restart'. A note says 'Update information obtained: 15 min ago'.

Click on Manage Jenkins ->system configuration-> ssh server and configure all the details shown in following fig

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

Java 11 is the recommended version to run Jenkins on; please consider upgrading.

System Configuration

Configure System
Configure global settings and paths.

SSH Server
Name ?
Tomcat_deployserver

Hostname ?
172.31.24.109

Username ?
admin

Remote Directory ?

Use password authentication, or use a different key ?
Passphrase / Password ?
.....

Save **Apply**

Once I finished all configuration. Next, I created a new job for pushing docker image using docker hub.

CREATING A NEW JOB

Click on New Item.

New Item

People

Build History

Project Relationship

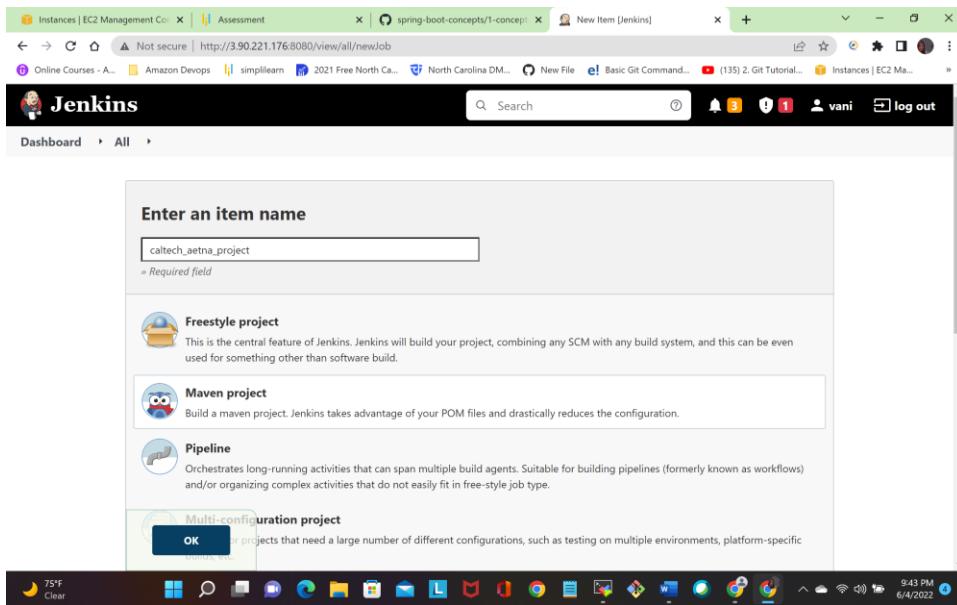
Check File Fingerprint

Manage Jenkins

All +

| S | W | Name | Last Success | Last Failure | Last Duration |
|-------|------|--|--------------|--------------|---------------|
| Green | Blue | demopipeline | 3 days 10 hr | #1 N/A | 6.8 sec |
| Green | Blue | deploy Java Application to Tomcat Apache | 1 day 4 hr | #1 N/A | 17 sec |

It will navigate you to the following screen:



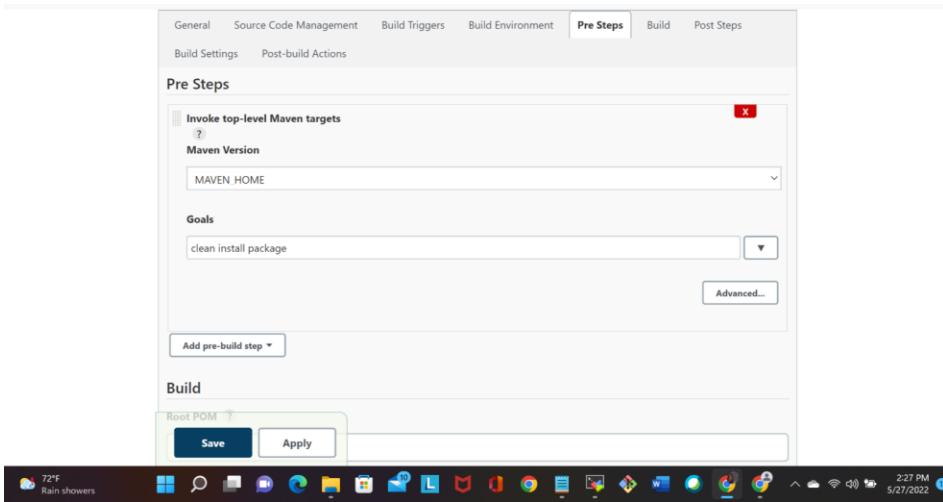
Click on Maven Project and I gave a name for project(caltech_aetna_project).

Once done, click on Ok button.

Navigate down to general Section and give description about project, next navigate to source code management give git hub repo URL GitHub:<https://github.com/vanishankar/Newrepo.git>, shown in below fig

Next I navigate to pre steps select top level maven goals shown in foll fig

and I gave maven goals clean, install, package



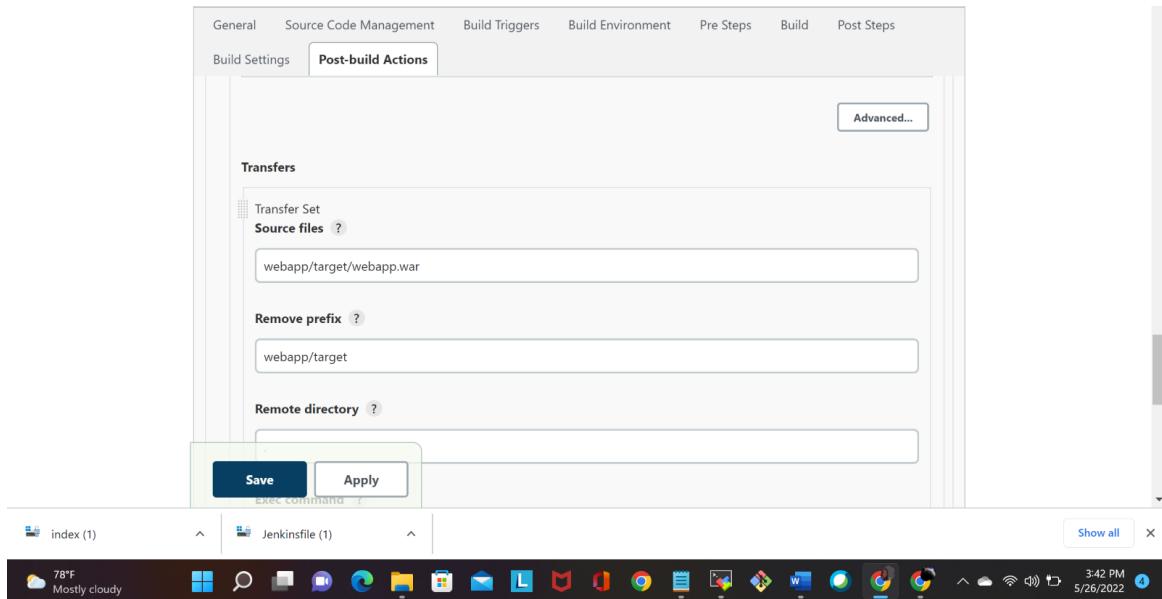
Next, I navigate to post steps I click send build artifacts over ssh(to build artifacts to docker server)

The screenshot shows a dropdown menu of post-build actions. The 'Send build artifacts over SSH' option is highlighted with a blue background. Other options include: Aggregate downstream test results, Archive the artifacts, Build other projects, Deploy artifacts to Maven repository, Record fingerprints of files to track usage, Stop Docker Containers, Git Publisher, Editable Email Notification, Set GitHub commit status (universal), Set build status on GitHub commit [deprecated], and Delete workspace when build is done. At the bottom of the list is an 'Add post-build action ▾' button.

And give the source files path, choose the remote directory to store the war file

POST-BUILD-ACTIONS:

The screenshot shows the configuration for the 'Send build artifacts over SSH' action. Under the 'SSH Publishers' heading, there is a 'SSH Server' section. The 'Name' field is set to 'Tomcat_deployserver'. Below this, there is a checkbox for 'Verbose output in console' which is checked. There are also three unchecked checkboxes for 'Credentials', 'Retry', and 'Label'. At the bottom of the section is a 'Transfers' button.



and click exec command to automate docker file and give the following command

docker build -t .

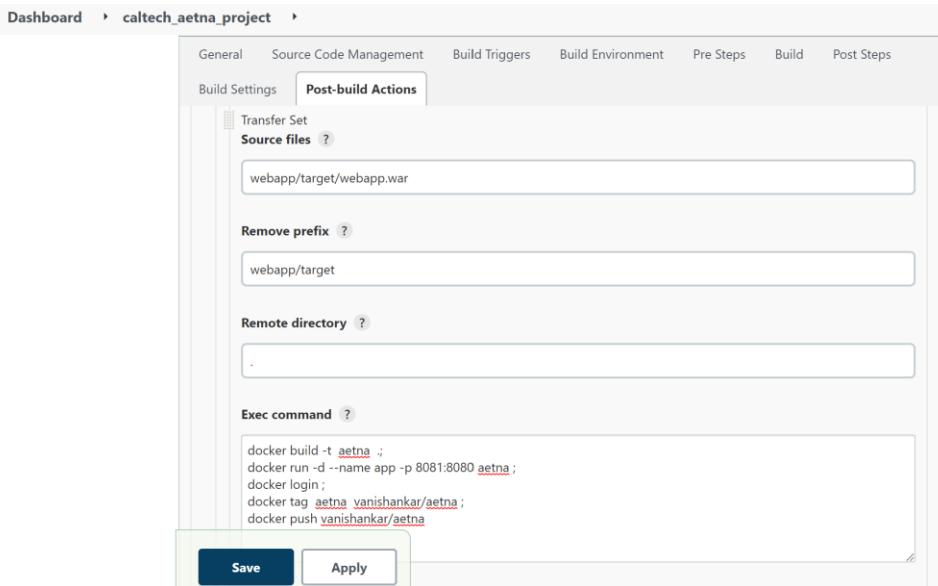
docker run -d -- name app -p 8081:8080 aetna

docker login

docker tag aetna vanishankar/aetna

docker push vanishankar/aetna

This docker command will help to build image, run the container, login to docker hub and pushes the image what we created



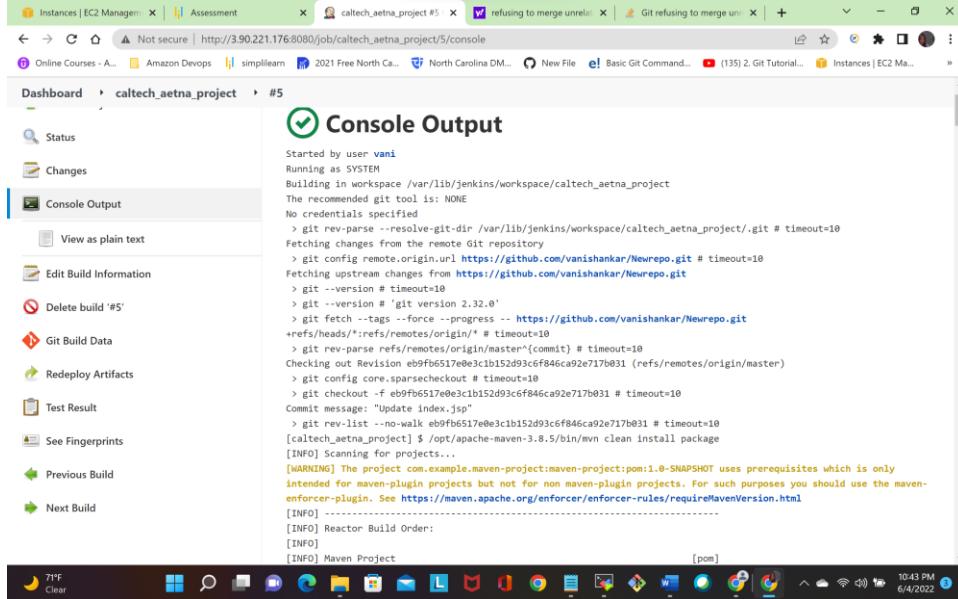
Click save and move to dashboard click Build now

| Build | Result | Time |
|-------|--------|------------------|
| #1 | Passed | 6 min 58 sec ago |
| #1 | Passed | 6 min 58 sec ago |
| #1 | Passed | 6 min 58 sec ago |
| #1 | Passed | 6 min 58 sec ago |

Finally, our caltech_aetna_project job is success

Console output:

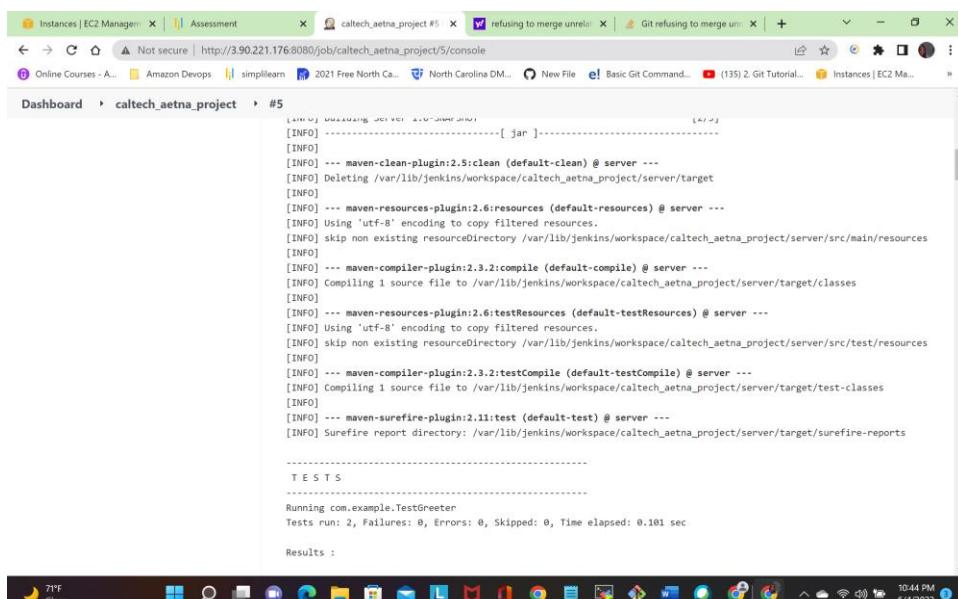
The foll fig shows output of our job



Dashboard > caltech_aetna_project > #5

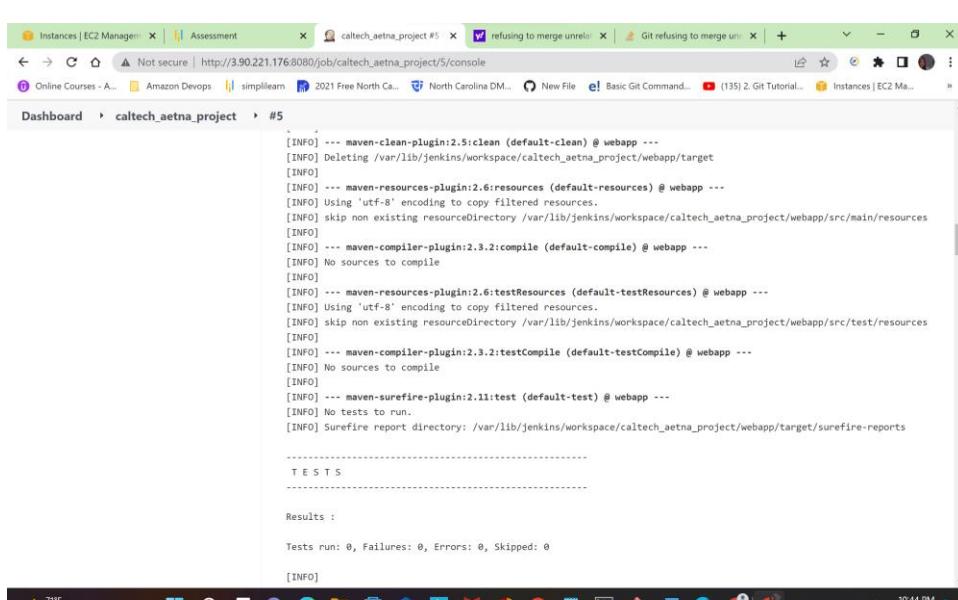
Console Output

```
Started by user vani
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/caltech_aetna_project
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/caltech_aetna_project/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/vanishankar/Newrepo.git # timeout=10
Fetching upstream changes from https://github.com/vanishankar/Newrepo.git
> git -version # timeout=10
> git -version # 'git' version 2.32.0'
> git fetch -tags --force --progress -- https://github.com/vanishankar/Newrepo.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision eb9fb6517e0e3c1b152d93c6f846ca92e717b031 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f eb9fb6517e0e3c1b152d93c6f846ca92e717b031 # timeout=10
Commit message: "Update index.jsp"
> git rev-list --no-walk eb9fb6517e0e3c1b152d93c6f846ca92e717b031 # timeout=10
[caltech_aetna_project] $ /opt/apache-maven-3.8.5/bin/mvn clean install package
[INFO] Scanning for projects...
[WARNING] The project com.example.maven-project:maven-project:pom:1.0-SNAPSHOT uses prerequisites which is only intended for maven-plugin projects but not for non maven-plugin projects. For such purposes you should use the maven-enforcer-plugin. See https://maven.apache.org/enforcer/enforcer-rules/requireMavenVersion.html
[INFO] -----
[INFO] Reactor Build Order:
[INFO] [INFO] Maven Project
[INFO] [pom]
```



Dashboard > caltech_aetna_project > #5

```
[INFO] -----
[INFO] [jar]
[INFO] -----
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ server ---
[INFO] Deleting /var/lib/jenkins/workspace/caltech_aetna_project/server/target
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ server ---
[INFO] Using 'utf-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/caltech_aetna_project/server/src/main/resources
[INFO] -----
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ server ---
[INFO] Compiling 1 source file to /var/lib/jenkins/workspace/caltech_aetna_project/server/target/classes
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ server ---
[INFO] Using 'utf-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/caltech_aetna_project/server/src/test/resources
[INFO] -----
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ server ---
[INFO] Compiling 1 source file to /var/lib/jenkins/workspace/caltech_aetna_project/server/target/test-classes
[INFO] -----
[INFO] --- maven-surefire-plugin:2.11:test (default-test) @ server ---
[INFO] Surefire report directory: /var/lib/jenkins/workspace/caltech_aetna_project/server/target/surefire-reports
-----
T E S T S
-----
Running com.example.TestGreeter
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.101 sec
Results :
```



Dashboard > caltech_aetna_project > #5

```
[INFO] -----
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ webapp ---
[INFO] Deleting /var/lib/jenkins/workspace/caltech_aetna_project/webapp/target
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ webapp ---
[INFO] Using 'utf-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/caltech_aetna_project/webapp/src/main/resources
[INFO] -----
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ webapp ---
[INFO] No sources to compile
[INFO] -----
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ webapp ---
[INFO] Using 'utf-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/caltech_aetna_project/webapp/src/test/resources
[INFO] -----
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ webapp ---
[INFO] No sources to compile
[INFO] -----
[INFO] --- maven-surefire-plugin:2.11:test (default-test) @ webapp ---
[INFO] Surefire report directory: /var/lib/jenkins/workspace/caltech_aetna_project/webapp/target/surefire-reports
-----
T E S T S
-----
Results :
```

```
[INFO] Instances | EC2 Manager | Assessment | caltech_aetna_project #5 | refusing to merge unrelated histories | Git refusing to merge unrelated histories | + - < > X
← → C ⌂ ⌂ Not secure | http://3.90.221.176:8080/job/caltech_aetna_project/5/console
Online Courses - A... Amazon Devops simplilearn 2021 Free North Ca... North Carolina DM... New File Basic Git Command... (135) 2. Git Tutorial... Instances | EC2 Ma...
Dashboard > caltech_aetna_project > #5
[INFO] Assembling webapp [webapp] in [/var/lib/jenkins/workspace/caltech_aetna_project/webapp/target/webapp]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/caltech_aetna_project/webapp/src/main/webapp]
[INFO] Webapp assembled in [7 secs]
[INFO] Building war: [/var/lib/jenkins/workspace/caltech_aetna_project/webapp/target/webapp.war]
[INFO] WEB-INF/web.xml already added, skipping
[INFO] -----
[INFO] Reactor Summary for Maven Project 1.0-SNAPSHOT:
[INFO]
[INFO] Maven Project ..... SUCCESS [ 0.784 s]
[INFO] Server ..... SUCCESS [ 3.881 s]
[INFO] Webapp ..... SUCCESS [ 0.822 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.869 s
[INFO] Finished at: 2022-06-05T02:05:31Z
[INFO] -----
[INFO] -----
[INFO] Parsing POMs
Established TCP socket on 35741
[caltech_aetna_project] $ /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64/bin/java -cp
/var/lib/jenkins/plugins/maven-plugin/WEB-INF/lib/maven35-agent-1.13.jar:/opt/apache-maven-3.8.5/boot/plexus-classworlds-2.6.0.jar:/opt/apache-maven-3.8.5/conf/logging Jenkins.maven3.agent.Maven3Main /opt/apache-maven-3.8.5
/var/lib/jenkins/WC/jenkins/war/WEB-INF/lib/remoting-4.13.jar /var/lib/jenkins/plugins/maven-plugin/WEB-INF/lib/maven35-interceptor-1.13.jar /var/lib/jenkins/plugins/maven-plugin/WEB-INF/lib/maven3-interceptor-commons-1.13.jar 35741
<==>JENKINS REMOTING CAPACITY==>channel started
Executing Maven: -B -f /var/lib/jenkins/workspace/caltech_aetna_project/pom.xml clean install package
[INFO] Scanning for projects...
[WARNING] The project com.example.maven-project:maven-project:pom:1.0-SNAPSHOT uses prerequisites which is only
```

```
71°F Clear 10:45 PM 6/4/2022
Instances | EC2 Manager | Assessment | caltech_aetna_project #5 | refusing to merge unrelated histories | Git refusing to merge unrelated histories | + - < > X
← → C ⌂ ⌂ Not secure | http://3.90.221.176:8080/job/caltech_aetna_project/5/console
Online Courses - A... Amazon Devops simplilearn 2021 Free North Ca... North Carolina DM... New File Basic Git Command... (135) 2. Git Tutorial... Instances | EC2 Ma...
Dashboard > caltech_aetna_project > #5
[INFO] --- maven-surefire-plugin:2.11:test (default-test) @ server ---
[INFO] Surefire report directory: /var/lib/jenkins/workspace/caltech_aetna_project/server/target/surefire-reports
-----
T E S T S
-----
Running com.example.TestGreeter
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.115 sec
Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[JENKINS] Recording test results
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ server ---
[INFO] Building jar: /var/lib/jenkins/workspace/caltech_aetna_project/server/target/server.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ server ---
[INFO] Installing /var/lib/jenkins/workspace/caltech_aetna_project/server/target/server.jar to
/var/lib/jenkins/.m2/repository/com/example/maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[INFO] Installing /var/lib/jenkins/workspace/caltech_aetna_project/server/pom.xml to
/var/lib/jenkins/.m2/repository/com/example/maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ server ---
[INFO] Using 'utf-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/caltech_aetna_project/server/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ server ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
71°F Clear 10:45 PM 6/4/2022
```

```
[INFO] No tests to run.
[INFO] Surefire report directory: /var/lib/jenkins/workspace/caltech_aetna_project/webapp/target/surefire-reports
-----
T E S T S
-----
Results :

Tests run: 0, Failures: 0, Errors: 0, Skipped: 0

[JENKINS] Recording test results
[INFO]
[INFO] --- maven-war-plugin:2.2:war (default-war) @ webapp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [webapp] in [/var/lib/jenkins/workspace/caltech_aetna_project/webapp/target/webapp]
```

```
← → C ⌂ ⌂ Not secure | http://3.90.221.176:8080/job/caltech_aetna_project/5/console
Online Courses - A... Amazon Devops simplilearn 2021 Free North Ca... North Carolina DM... New File Basic Git Command... (135) 2. Git Tutorial... Instances | EC2 Ma...
Dashboard > caltech_aetna_project > #5
[jun 4 2022 11:00:00] [INFO] [Docker] Using Docker daemon: docker: default
docker run -d --name apple -p 8082:8080 aetna ;
docker login ;
docker tag aetna vanishankar/aetna ;
docker push vanishankar/aetna ...
SSH: EXEC: connected
Sending build context to Docker daemon 11.78kB

Step 1/3 : FROM tomcat:latest
--> c795915cb678
Step 2/3 : COPY ./webapp.war /usr/local/tomcat/webapps
--> bb5d49590e08
Step 3/3 : RUN cp -r /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
--> Running in 88fc7a2e437a
Removing intermediate container 88fc7a2e437a
--> 8571bd16da67
Successfully built 8571bd16da67
Successfully tagged aetna:latest
b0d82a3ac62a3e1f10e518d1abc0711a06673ad37994e147794bad0add2921d
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/admin/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

Instances | EC2 Manager | Assessment | caltech_aetna_project #5 | refusing to merge unrelated histories | Git refusing to merge unrelated histories | + | - | X

Online Courses - A... Amazon Devops simplilearn 2021 Free North Ca... North Carolina DM... New File Basic Git Command... (135) 2. Git Tutorial... Instances | EC2 Ma...

Dashboard > caltech_aetna_project > #5

```

    YOUR_PASSWORD WILL BE STORED UNENCRYPTED IN: /home/vanishankar/.config/git/config
    Configure a credential helper to remove this warning. See
    https://docs.docker.com/engine/reference/commandline/Login/#credentials-store

    Login Succeeded
    Using default tag: latest
    The push refers to repository [docker.io/vanishankar/aetna]
    f217e703f1a8: Preparing
    3ae4f9a9b9f0: Preparing
    9cd8d0fd1f4: Preparing
    879928deaa571: Preparing
    f0bbc1d0b6f7: Preparing
    e5ce437a3a3d: Preparing
    d744b7303b0d: Preparing
    817e710a8d04: Preparing
    ee509edee976: Preparing
    9177197c67d0: Preparing
    7dbadaf2b9b0d8: Preparing
    e7597c345c2e: Preparing
    e5ce437a3a3d: Waiting
    d744b7303b0d: Waiting
    817e710a8d04: Waiting
    ee509edee976: Waiting
    9177197c67d0: Waiting
    7dbadaf2b9b0d8: Waiting
    e7597c345c2e: Waiting
    f0bbc1d0b6f7: Mounted from library/tomcat
    9cd8d0fd1f4: Mounted from library/tomcat
    879928deaa571: Mounted from library/tomcat
    e5ce437a3a3d: Mounted from library/tomcat
    d744b7303b0d: Mounted from library/tomcat
    817e710a8d04: Mounted from library/tomcat
    ee509edee976: Mounted from library/tomcat
    9177197c67d0: Mounted from library/tomcat
    3ae4f9a9b9f0: Pushed
    7dbadaf2b9b0d8: Mounted from library/tomcat
    e7597c345c2e: Mounted from library/tomcat
    f217e703f1a8: Pushed
    latest: digest: sha256:a2cf4116d127afdd757dcfb7753792861d44e06d9b7a5c19229c6054ff751775 size: 2841
    SSH: EXEC: completed after 5,407 ms
    SSH: Disconnecting configuration [Tomcat_deployserver] ...
    SSH: Transferred 1 file(s)
    Finished: SUCCESS
  
```

71°F Clear REST API Jenkins 2.332.3

Test result:

Instances | EC2 Manager | Assessment | caltech_aetna_project/commit/9 | refusing to merge unrelated histories | Git refusing to merge unrelated histories | + | - | X

Online Courses - A... Amazon Devops simplilearn 2021 Free North Ca... North Carolina DM... New File Basic Git Command... (135) 2. Git Tutorial... Instances | EC2 Ma...

Jenkins

Dashboard > caltech_aetna_project > Server > #9 > Test Results

Test Result

0 failures (±0)

2 tests (±0)
Took 0.13 sec.

Add description

All Tests

| Package | Duration | Fail (diff) | Skip (diff) | Pass (diff) | Total (diff) |
|-------------|----------|-------------|-------------|-------------|--------------|
| com.example | 24 ms | 0 | 0 | 2 | 2 |

See Fingerprints

Next, we can access from browser with docker_server ip address with port (we gave 8081)
ACCESSING FROM BROWSER

FINAL OUTPUT

Now login to docker hub with id our image aetna is pushed to **Docker hub**

Now login to docker hub with id and clock continue button.it will navigate into Docker hub repo home page

DOCKERHUB REPOSITORY

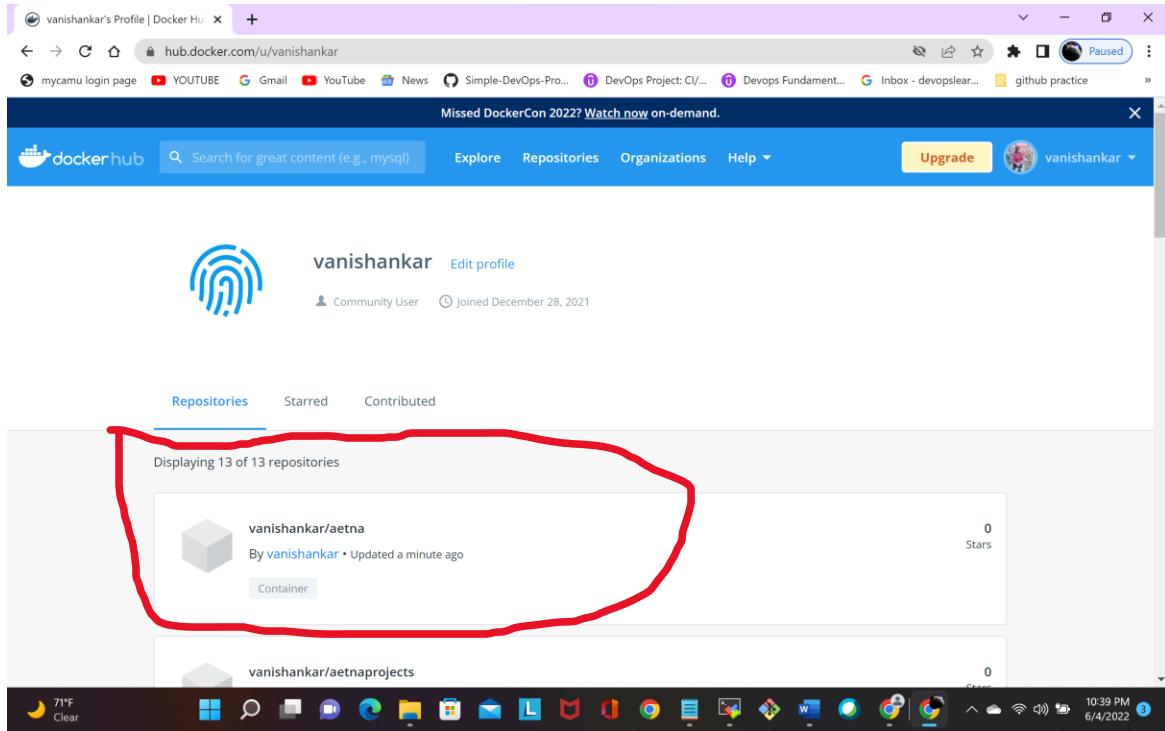


image aetna is pushed to **Docker hub**. Successfully pushed our docker image into docker hub using jenkins.

Concepts used in the project is containerization(docker) it helps Faster delivery of your applications, Deploy and scale more easily Get higher density and run more workloads

Docker Troubleshooting

If you see the below error, it means docker service is not running on the machine. Start the docker service and try the build again

docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?

If you see the below error “Permission Denied”

```
+ docker run hello-world
docker: Got permission denied while trying to connect to the Docker daemon socket
at unix:///var/run/docker.sock: Post
http://%2Fvar%2Frun%2Fdocker.sock/v1.40/containers/create: dial unix
/var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
```

Then add jenkins user to the docker group and restart the jenkins service as shown below:

```
sudo usermod -aG docker admin
```

```
sudo service jenkins stop
```

```
sudo service jenkins start
```

Finally, I deleted the all images and container running in the Docker server using following command

To check or list out what are the images are present in our server I used following command

```
docker images
```

To list out all container

```
docker ps -a or docker container ls
```

To list out only running container:

```
docker ps
```

Delete docker images

```
docker rmi -f image id
```

Delete docker container

```
docker rm -f container id
```

Stop docker service

```
docker stop container id
```

Delete unused image

```
docker image prune
```

Delete unused container

```
docker container prune
```

Check the status

```
Service docker status
```

Following fig shows deleted, pruned images and container

```
[admin@ip-172-31-24-109 ~]$ docker images
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
aetna              latest     ae9b3865fad9  About a minute ago  684MB
vanishankar/aetna  latest     ae9b3865fad9  About a minute ago  684MB
vanishankar/aetna  <none>    8571bd16da67  6 minutes ago   684MB
vanishankar/aetna  <none>    2987b7a0dcdb8  9 minutes ago   684MB
tomcat             latest     c795915cb678  7 days ago    680MB

[admin@ip-172-31-24-109 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
eac1c881befd        aetna              "catalina.sh run"   2 seconds ago     Up Less than a second   0.0.0.0:8083->8080/tcp, ::::8083->8080/tcp   applepi
b0d82a3ac962        8571bd16da67        "catalina.sh run"   6 minutes ago     Up 6 minutes        0.0.0.0:8082->8080/tcp, ::::8082->8080/tcp   apple
ce4ee2fffc3a        2987b7a0dcdb8      "catalina.sh run"   9 minutes ago     Up 9 minutes        0.0.0.0:8081->8080/tcp, ::::8081->8080/tcp   app

[admin@ip-172-31-24-109 ~]$ docker stop eac1c881befd b0d82a3ac962
eac1c881befd
b0d82a3ac962
[admin@ip-172-31-24-109 ~]$ docker rm -f eac1c881befd b0d82a3ac962
eac1c881befd
b0d82a3ac962
[admin@ip-172-31-24-109 ~]$ docker rmi -f ae9b3865fad9 ae9b3865fad9 8571bd16da67 2987b7a0dcdb8 c795915cb678
Untagged: vanishankar/aetna@sha256:ba42625e3f3e90e5964a2610aefb017bcd7fb504d6c81eb06408803ec5291
Deleted: sha256:ae9b3865fad94919a20b3574163e0619d75c5974e3978b50fe36331fd7314f
Deleted: sha256:6a2256a62a33e43bb3b71402a1fa36d62245ff0366e764a38543d8f7ec6cd
Deleted: sha256:b5b12151187086366166cdcf3db01cfefb983db164c31033d39431f1diaae
Deleted: sha256:368366240c73b12896a9762a7fa60edca616529e234ed47476f153d959327678
Untagged: vanishankar/aetna@sha256:e2cf4116d127afdd757dcfb753792861d44e06d9b7a5c19229c6054ff751775
Deleted: sha256:8571bd16da673c4e193d0b804247eb7727f9e62fc1a1624f75587c522774592
Deleted: sha256:e0f84b872c4779cbca07df6e126623bf100917443ce9b3e01be2b3ad2a4671
Deleted: sha256:b5d49590e0879a15aac6df9c3b298dd19b4a0e5d5a/cze0c81502714e9afbe
Deleted: sha256:8b:c979d56979ed93963288ed1a846ad7fe22559166bb8f8e7e0cb7c51c137778

[admin@ip-172-31-24-109 ~]$ docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
[admin@ip-172-31-24-109 ~]$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
[admin@ip-172-31-24-109 ~]$ docker system prune
WARNING! This will remove:
 - all stopped containers
 - all networks not used by at least one container
 - all dangling images
 - all dangling build cache

Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
[admin@ip-172-31-24-109 ~]$
```

Conclusion

Advantages and USPs: Docker The main reason is that Jenkins work really well with Docker. Without Docker you need to install additional tools and add different agents to Jenkins. With Docker, there is no need to install additional tools, you just use images of these tools. Jenkins will download them from internet for you (Docker Hub). Jenkins is a popular continuous integration and delivery (CI/CD) tool. Jenkins is open source and has a great developer community that maintains its existing software and builds plugins to increase its functionality. Though with excellent community support, while using Jenkins, you may suffer from issues like configuration files clashes which you can resolve by running Jenkins in a container technology like Docker.

