tksahu.devops@gmail.com

COMPLETE DEVOPS CI/CD PIPELINE PROJECT USING GIT, GITHUB, TOMCAT, JENKINS, MAVEN, AND JAVA ON AN UBUNTU SERVER WITH JENKINS INSTALLED ON TOMCAT 10

READ MORE

tksahu.devops@gmail.com

PROJECT DESCRIPTION:

PROJECT DESCRIPTION:

This project aims to demonstrate a basic CI/CD pipeline for a Java web application using Git and GitHub for version control, Jenkins for continuous integration, Tomcat 10 as the application server, and Maven for build automation. Jenkins will be deployed on a Tomcat 10 server running on Ubuntu, and we'll automate the build and deployment process from source code in GitHub to a running Java web application on Tomcat.

tksahu.devops@gmail.com

TOOLS AND TECHNOLOGIES USED:

- GIT: VERSION CONTROL SYSTEM TO TRACK CHANGES IN SOURCE CODE.
- GITHUB: REMOTE REPOSITORY FOR THE GIT PROJECT.
- MAVEN: BUILD AUTOMATION TOOL TO MANAGE PROJECT DEPENDENCIES AND COMPILE CODE.
- JENKINS: CI/CD TOOL TO AUTOMATE BUILDS, TESTING, AND DEPLOYMENT.
- TOMCAT 10: WEB SERVER TO HOST THE JAVA APPLICATION.
- UBUNTU: OPERATING SYSTEM TO HOST JENKINS AND TOMCAT.
- JAVA: PROGRAMMING LANGUAGE FOR THE APPLICATION.

tksahu.devops@gmail.com

Step 1: Launch an EC2 Instance

1. Sign in to your AWS Management Console.
2. Launch an EC2 instance (Ubuntu 20.04 or later is recommended):
   - Choose an instance type (t2.micro should be enough for testing).
   - Configure security group to allow inbound traffic on ports 8080 (Tomcat) and 22 (SSH).

1. Step 2: Connect to Your EC2 Instance
2. Connect to your EC2 instance via SSH:
3.
4. bash
5. ssh -i <your-key-pair>.pem ubuntu@<your-ec2-public-ip>
   -

tksahu.devops@gmail.com

Step#1 : Update System Packages

It's always a good idea to update the package lists for upgrades and new installations. Open your terminal, and enter the following command:

UBUNTU@TK.DEVOPS:~$ SUDO APT UPDATE && SUDO APT UPGRADE -Y

```
it:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
it:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
it:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
it:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
ll packages are up to date.
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
alculating upgrade... Done
 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

tksahu.devops@gmail.com

Step#2 : Install OpenJDK- Java
As we know the key requirement to install Tomcat is Java, thus first we set up an open-source Java Development kit on Ubuntu 22.04 LTS using the terminal.

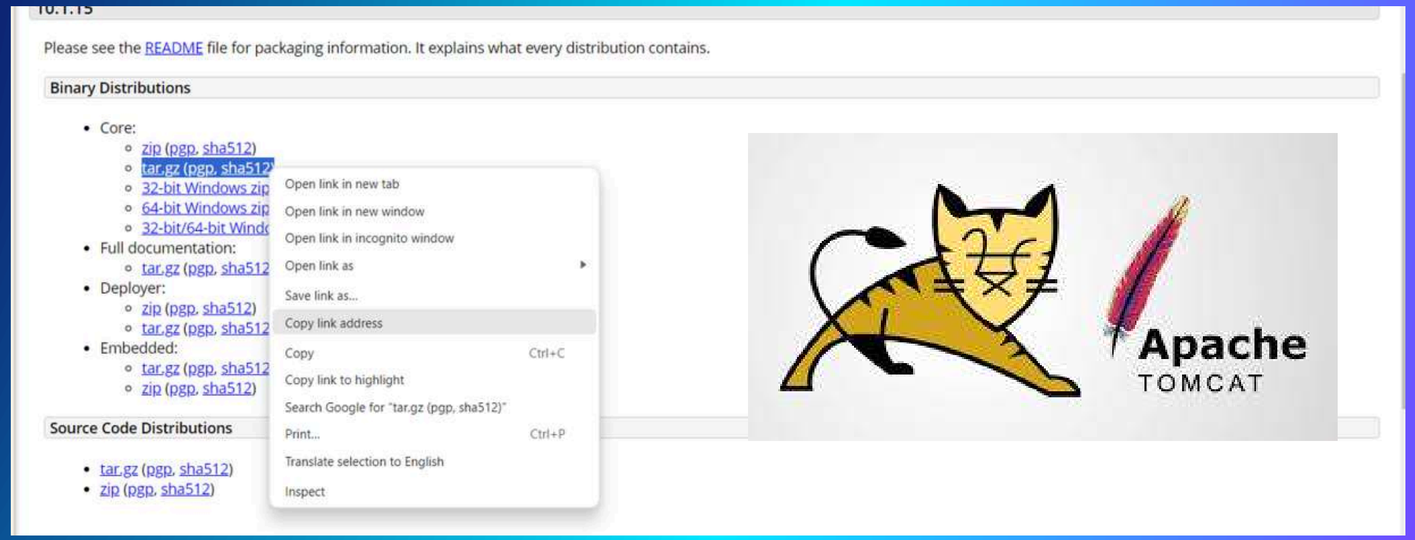
UBUNTU@TK.DEVOPS:~$ SUDO APT INSTALL DEFAULT-JDK

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
default-jdk is already the newest version (2:1.11-72build2).
0 upgraded, 0 newly installed, 0 to remove and 133 not upgraded.
ubuntu@RushiInfotech:~$ java -version
openjdk version "11.0.20.1" 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
```

tksahu.devops@gmail.com

Step#3 : Download Apache Tomcat

You can get the latest version directly from the official webpage of Tomcat. If you are using a GUI system then can download it by simply clicking on the Zip or Tar file. However, those who are accessing a remote server with CLI via SSH can use the wget. Simply, right-click on the Tart.gz file and copy the link address. After that type wget on your terminal and paste the link.wget paste-link



UBUNTU@TK.DEVOPS:~$ SUDO WGET HTTPS://DLCDN.APACHE.ORG/TOMCAT/TOMCAT-10/V10.1.15/BIN/APACHE-TOMCAT-10.1.15.TAR.GZ

```
--2023-10-17 10:28:10--  https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.15/bin/apache-tomcat-10.1.15.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12408080 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-10.1.15.tar.gz'

apache-tomcat-10.1.15.tar.gz   100%[===================================================>]  11.83M  --.-KB/s    in 0.09s
```

tksahu.devops@gmail.com

Step#4 : Install Apache Tomcat on Ubuntu 22.04

Extract the downloaded file to /opt directory so that we won't delete it accidentally.

UBUNTU@TK.DEVOPS:~$ SUDO MKDIR -P /OPT/TOMCAT

```
~$ sudo mkdir -p /opt/tomcat
~$ cd /opt
/opt$ ls

/opt$
```

UBUNTU@TK.DEVOPS:~$ SUDO TAR XZVF APACHE-TOMCAT-*TAR.GZ -C /OPT/TOMCAT --STRIP-COMPONENTS=1

```
apache-tomcat-10.1.15/conf/
apache-tomcat-10.1.15/conf/catalina.policy
apache-tomcat-10.1.15/conf/catalina.properties
apache-tomcat-10.1.15/conf/context.xml
apache-tomcat-10.1.15/conf/jaspic-providers.xml
apache-tomcat-10.1.15/conf/jaspic-providers.xsd
apache-tomcat-10.1.15/conf/logging.properties
apache-tomcat-10.1.15/conf/server.xml
apache-tomcat-10.1.15/conf/tomcat-users.xml
apache-tomcat-10.1.15/conf/tomcat-users.xsd
apache-tomcat-10.1.15/conf/web.xml
```

tksahu.devops@gmail.com

3. Create a dedicated user

To ensure the security of the system while testing various web applications, let's create a non-root user that has only access to the created /opt/tomcat folder.

Tubuntu@TK.devops:~$ sudo groupadd tomcat

ubuntu@TK.devops:~$ sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

```
:/$ cat /etc/passwd | grep tomcat
tomcat:x:1001:1001::/opt/tomcat:/bin/false
```

Step#5 : Assigning Tomcat user permissions

Well, we already have set up the files that require using this open-source web application server, now let's assign the permission of the folder to the user we have created above for it.

ubuntu@TK.devops:~$ sudo chown -R tomcat: /opt/tomcat

ubuntu@TK.devops:~$ sudo sh -c 'chmod +x /opt/tomcat/bin/*.sh'

```
/$ sudo chown -R tomcat: /opt/tomcat
/$ sudo sh -c 'chmod +x /opt/tomcat/bin/*.sh'
/$
```

tksahu.devops@gmail.com

Step#6 : Create a Systemd service file

By default, we won't have a Systemd unit file for Tomcat to run it in the background and to easily stop, start and enable its services. Thus, we create one, so that we could manage it without any issues.

Create Systemd unit file

ubuntu@TK.devops:~$ sudo nano /etc/systemd/system/tomcat.service

```
  GNU nano 6.2                          /etc/systemd/system/tomcat.service *
[Unit]
Description=Tomcat webs servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat
RestartSec=10
Restart=always
Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64"
Environment="JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom"

Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

To save the press Ctrl+X, type –Y, and hit the Enter Key.

tksahu.devops@gmail.com

Note: In the above-given code for creating a systemd file, we have to mention the path of Java. However, the given one in the above code is the default path, still, to confirm the same you can run the below command:

ubuntu@TK.devops:~$ sudo update-java-alternatives -l

```
/$ sudo update-java-alternatives -l
java-1.11.0-openjdk-amd64      1111      /usr/lib/jvm/java-1.11.0-openjdk-amd64
```

Step#7 : Enable and start Tomcat service on Ubuntu 22.04

Finally, we have plugged in all the necessary things to start the Tomcat service in the background on Ubuntu 22.04 LTS Jammy. Let's enable and run the same.

ubuntu@TK.devops:~$ sudo systemctl daemon-reload
ubuntu@TK.devops:~$ sudo systemctl start tomcat
ubuntu@TK.devops:~$ sudo systemctl enable tomcat

```
:/$ sudo systemctl daemon-reload
:/$  sudo systemctl start tomcat
:/$ sudo systemctl enable tomcat
Created symlink /etc/systemd/system/multi-user.target.wants/tomcat.service → /etc/systemd/system/tomcat.service.
```

tksahu.devops@gmail.com

To confirm everything is working normally, check the status of service

ubuntu@TK.devops:~$ sudo systemctl status tomcat --no-pager -l

```
● tomcat.service - Tomcat webs servlet container
     Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-10-17 10:44:46 UTC; 43s ago
   Main PID: 4183 (java)
      Tasks: 29 (limit: 1141)
     Memory: 133.2M
        CPU: 4.067s
     CGroup: /system.slice/tomcat.service
             └─4183 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/loggin
g.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -Djava.security.egd
=file:/dev/./urandom -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.
apache.catalina.security.SecurityListener.UMASK=0027 --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java
.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-op
ens=java.rmi/sun.rmi.transport=ALL-UNNAMED -Xms512M -Xmx1024M -server -XX:+UseParallelGC -classpath /opt/tomcat/bin/bootstr
ap.jar:/opt/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/opt/tomcat -Dcatalina.home=/opt/tomcat -Djava.io.tmpdir=/opt/tomcat
/temp org.apache.catalina.startup.Bootstrap start

Oct 17 10:44:46 RushiInfotech systemd[1]: Starting Tomcat webs servlet container...
Oct 17 10:44:46 RushiInfotech startup.sh[4176]: Tomcat started.
Oct 17 10:44:46 RushiInfotech systemd[1]: Started Tomcat webs servlet container.
ubuntu@RushiInfotech:/$
```

tksahu.devops@gmail.com

Step#8 : Add Roles and Admin username and password

This step is important, without performing it we will get an error: “403 Access Denied on Tomcat 10/9/8 error” as we click on “Server Status“, “Manager App” and “Host Manager” links on the Apache Tomcat Web interface.

Edit user configuration file

```
ubuntu@TK.devops:~$ sudo nano /opt/tomcat/conf/tomcat-users.xml
```

```
<role rolename="admin"/>
<role rolename="admin-gui"/>
<role rolename="manager"/>
<role rolename="manager-gui"/>

<user username="TK.devops" password="TK123" roles="admin,admin-gui,manager,manager-gui"/>
```

Save the file and exit- Ctrl+X, type- Y, and hit the Enter key.

tksahu.devops@gmail.com

Step#9 : Enable Tomcat and Host Manager Remote access

By default, you won't be able to access your installed Tomcat Manager sections (web interface) outside the local system. For that, we have to enable it by editing individually the context.xml file available for Tomcat Manager and Host Manager.

For Tomcat Manager's remote access:

Edit the Context file

<user username="TK.devops" password="TK123" roles="admin,admin-gui,manager,manager-gui"/>

ubuntu@TK.devops:~$ sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml

In the file, scroll and go to the end and comment out the following block of text-

<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />

Just add <!-- at the beginning and --> in the end, after that, this will look something like this-

<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->

```
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
                   sameSiteCookies="strict" />
<!--  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalina\.filters\.
</Context>
```

Save the file and exit- Ctrl+X, type- Y, and hit the Enter key.

tksahu.devops@gmail.com

For Host manager remote access:

ubuntu@TK.devops:~$ sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml

Just like above, also add <!-- at the beginning and --> at the end of the text given below in the file, after that, this will look like something this-

```
<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
                    allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
```

```
-->
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
                   sameSiteCookies="strict" />
<!--  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalina\.filters\.>
</Context>
```

Save the file and exit- Ctrl+X, type- Y, and hit the Enter key.

Restart Tomcat service-
ubuntu@TK.devops:~$ sudo systemctl restart tomcat

tksahu.devops@gmail.com

Step#10 : Open port 8080 in Ubuntu 22.04 Firewall

The service to access the web interface via browser is available on port 8080 and to access the same remotely using any other system, we have to allow its outgoing connection in the firewall.

ubuntu@TK.devops:~$ sudo ufw allow 8080

```
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
ubuntu@RushiInfotech:/$ sudo ufw allow 8080
Rule added
Rule added (v6)
ubuntu@RushiInfotech:/$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
8080                       ALLOW       Anywhere
8080 (v6)                  ALLOW       Anywhere (v6)
```

tksahu.devops@gmail.com

Step#11 : Access the Tomcat Web interface

To access the tomcat default page we need to change the security settings of the instance.

Change the Inbound rules in Security of the instance and simply add a rule :

1.Select Type info – Custom TCP

2.Port range –8080

3.Select Source info – Anywhere IPv4

Amazon EC2

EC2 > Security Groups > sg-018d809a9c7e93a57 - launch-wizard-115 > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0e4fefa18d08e250d	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-02fed80d00b1122ed	Custom TCP	TCP	8080	Custom	0.0.0.0/0		Delete

Add rule

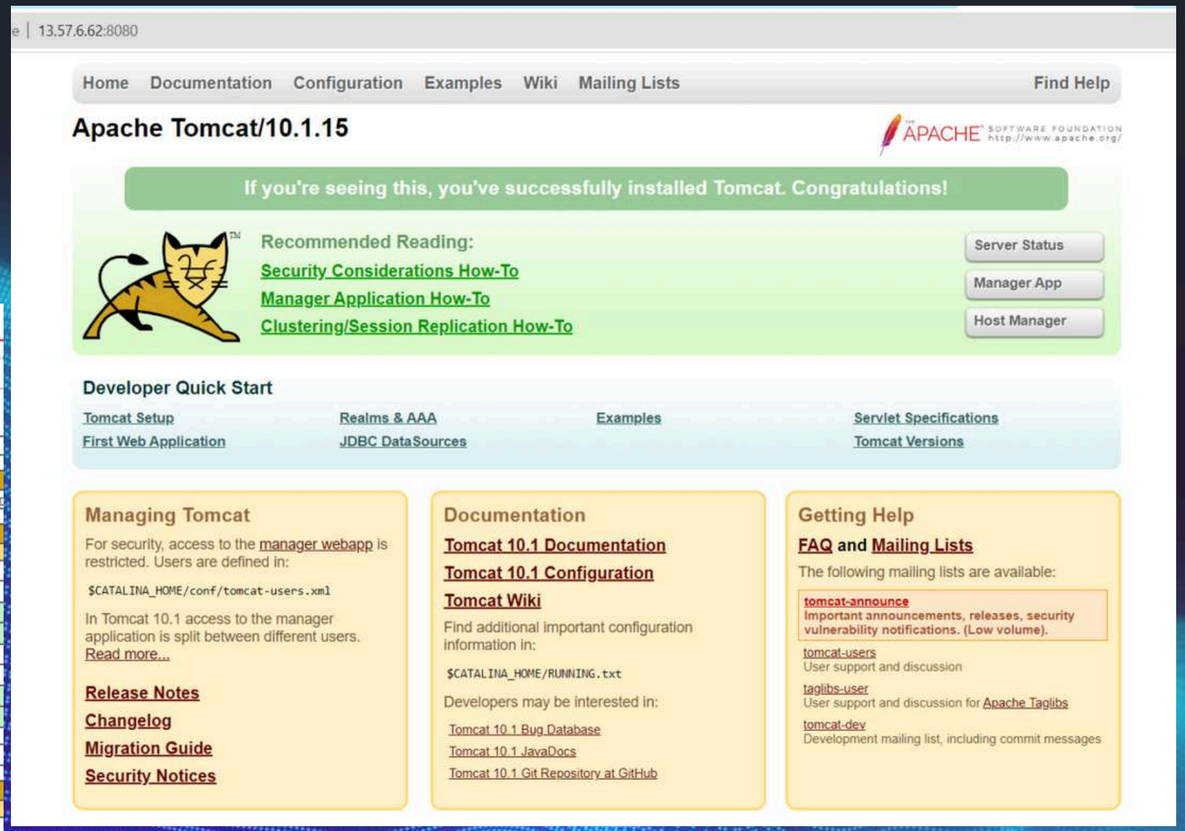
And then Open any browser on the local or remote system and point it to the IP address or domain of the server where you have installed the Apache Tomcat.

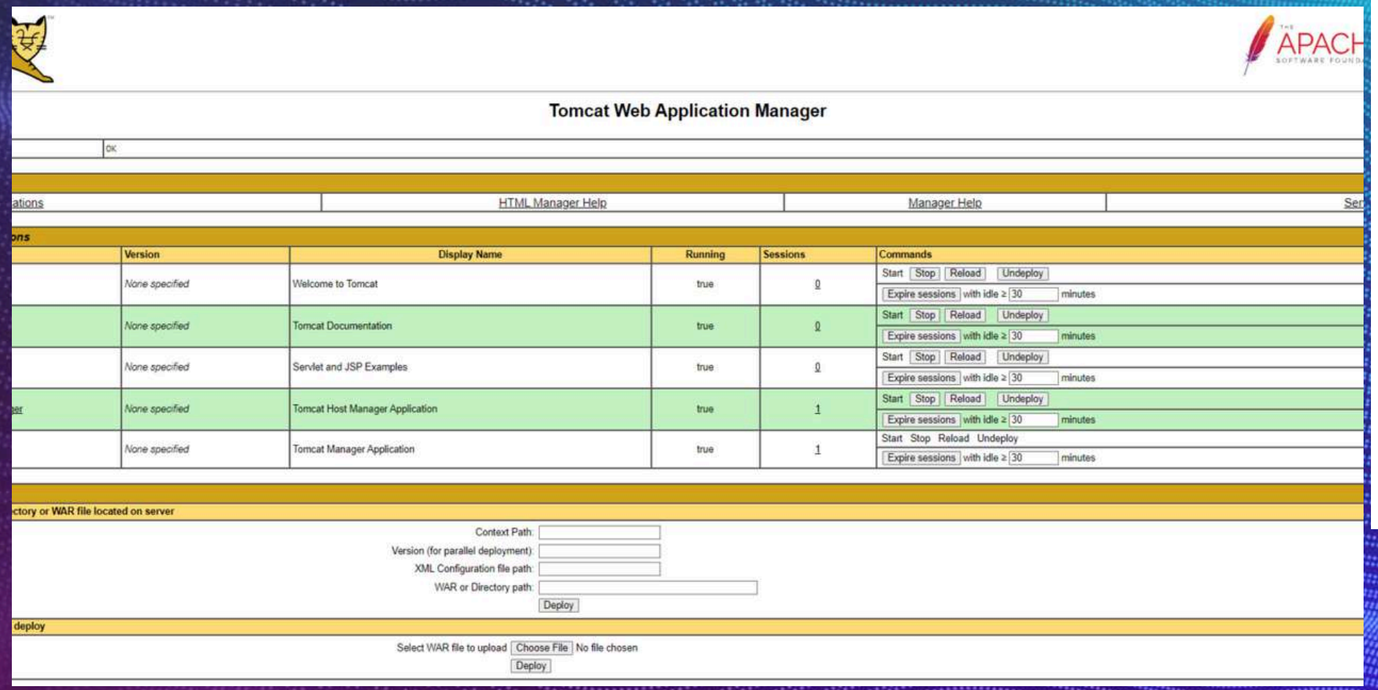
tksahu.devops@gmail.com

For example:

Open any browser on the local or remote system and point it to the IP address or domain of the server where you have installed the Apache Tomcat.

http://server-ip-addres:8080
or
http://yourdomain.com:8080





For example:

Open any browser on the local or remote system and point it to the IP address or domain of the server where you have installed the Apache Tomcat.

http://server-ip-addres:8080
or
http://yourdomain.com:8080

tksahu.devops@gmail.com

Download jenkins stable version war file

find the latest version of jenkins at War Jenkins Packages , Download the jenkins war file using wget command

```
$ cd /tmp
$ wget https://get.jenkins.io/war-stable/2.346.1/jenkins.war
```

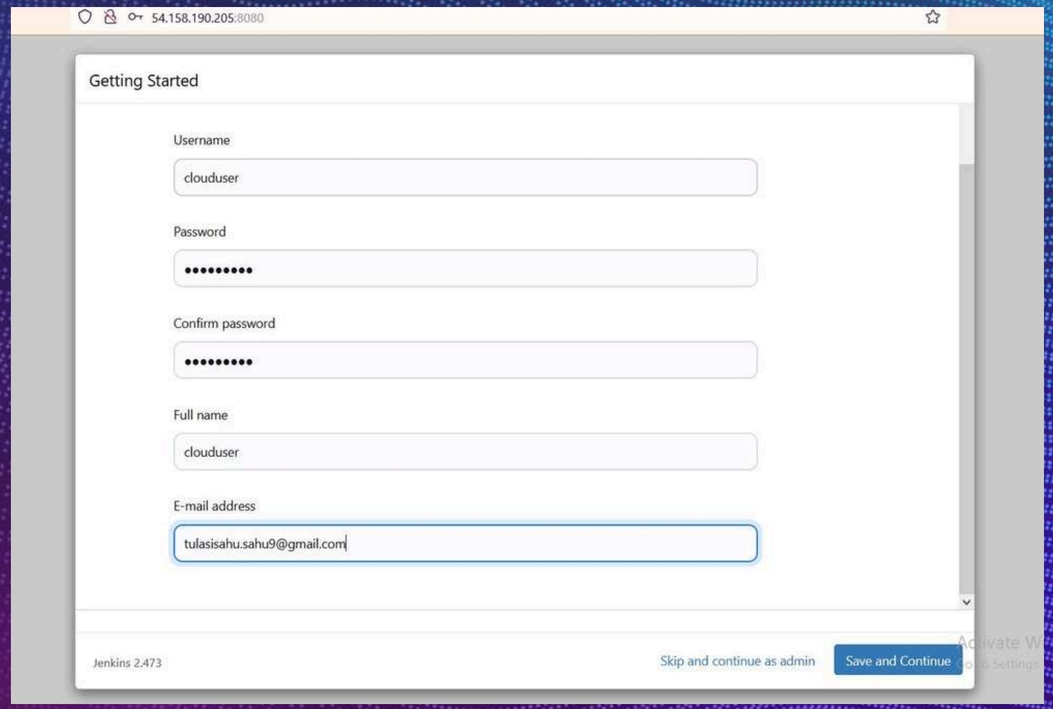
Deploy download jenkins war file into tomcat webapps folder

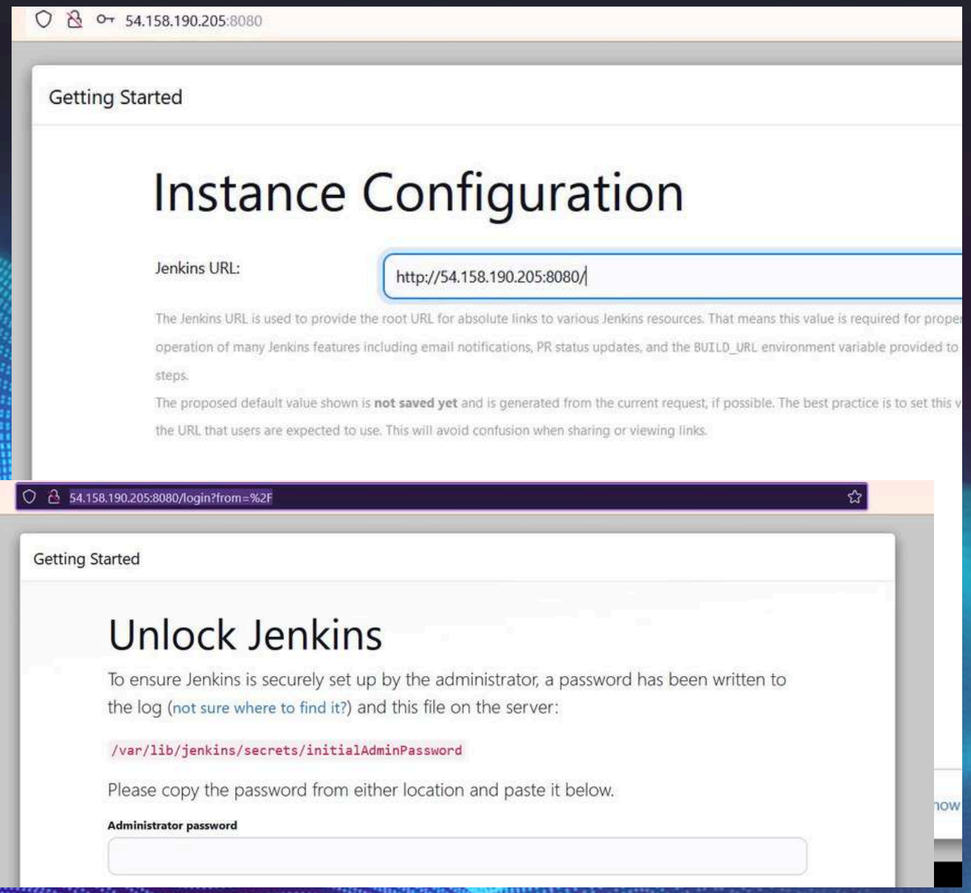
```
$ sudo cp jenkins.war /opt/tomcat/webapps/
```

Restart tomcat service

```
$ sudo systemctl restart tomcat
```

Hit the tomcat url in your web browser http://ip-address:8080/jenkins





tksahu.devops@gmail.com

MAVEN INSTALLATION

```
ubuntu@ip-172-31-40-136:/$ cd /tmp

ubuntu@ip-172-31-40-136:/tmp$ sudo wget https://dlcdn.apache.org/maven/
maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz -p /tmp
--2024-10-10 18:23:07--  https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/
apache-maven-3.9.9-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9102945 (8.7M) [application/x-gzip]
Saving to: ‘dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-
bin.tar.gz’

dlcdn.apache.org/ma 100%[===================>]  8.68M  13.6MB/s    in 0.6s

  2024-10-10 18:23:08 (13.6 MB/s) - ‘dlcdn.apache.org/maven/maven-3/3.9.9/binaries/
                apache-maven-3.9.9-bin.tar.gz’ saved [9102945/9102945]
```

tksahu.devops@gmail.com


```
root@ip-172-31-40-136:/tmp# cd /tmp
root@ip-172-31-40-136:/tmp# ls -l /tmp
total 8924
-rw-r--r-- 1 root   root   9102945 Aug 17 18:44 apache-maven-3.9.9-bin.tar.gz
drwxr-xr-x 3 root   root      4096 Oct 10 18:23 dlcdn.apache.org
drwxr-x--- 2 ubuntu ubuntu    4096 Oct 10 18:07 hsperfdata_ubuntu
drwx------ 2 root   root      4096 Oct 10 17:22 snap-private-tmp
drwx------ 3 root   root      4096 Oct 10 17:22 systemd-
private-49ec61875ef54f9797d67c3aa4be5381-ModemManager.service-oTDngv
drwx------ 3 root   root      4096 Oct 10 17:22 systemd-
private-49ec61875ef54f9797d67c3aa4be5381-chrony.service-SYIhMc
```

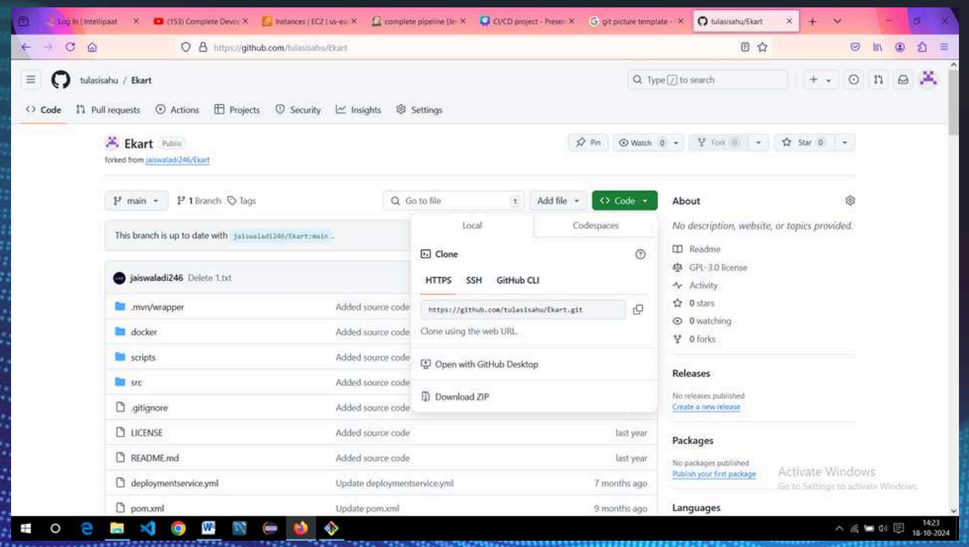
```
root@ip-172-31-40-136:/tmp# mvn -version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: /opt/maven
Java version: 11.0.24, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1016-aws", arch: "amd64", family: "unix"
root@ip-172-31-40-136:/tmp#
```

tksahu.devops@gmail.com

this will project part of git

```
ubuntu@ip-172-31-40-136:~$ cd gitworkspace
ubuntu@ip-172-31-40-136:~/gitworkspace$ git clone
```

https://github.com/tulasisahu/Ekart.git




```
Cloning into ' Ekart '...
remote: Enumerating objects: 208, done.

remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 208 (delta 53), reused 53 (delta 53), pack-reused 142 (from 1)
Receiving objects: 100% (208/208), 16.83 MiB | 27.75 MiB/s, done.
Resolving deltas: 100% (63/63), done.
ubuntu@ip-172-31-40-136:~/gitworkspace$ dir
Ekart
ubuntu@ip-172-31-40-136:~/gitworkspace$
```

tksahu.devops@gmail.com

To install Jenkins plugins, follow these steps:

Through Jenkins UI:

1. Go to Jenkins Dashboard > Manage Jenkins > Manage Plugins.
2. Under the Available tab, search for the plugin you want

FOR THIS PROJECT REQUIRED PLUGINS ARE :

(Maven, GitHub, warning next generation,junit,deploy to container).

3 . Select the plugin and click Install without restart or Download now and install after restart.

Using Jenkins CLI:

Download Jenkins CLI using:

wget http://<JENKINS_URL>/jnlpJars/jenkins-cli.jar

Install plugins via CLI: -

java -jar jenkins-cli.jar -s http://<JENKINS_URL>/ install-plugin <plugin-name>

tksahu.devops@gmail.com

Create a Jenkins Job

After installing the plugins, follow these steps to create a new job in Jenkins:

Step-by-Step Job Creation Process:

1. Login to Jenkins:
   - Access your Jenkins dashboard at http://<your-jenkins-server>:8080.
2. Create a New Job:
   - On the dashboard, click on "New Item".
   - Choose "Freestyle project" or "Pipeline project" depending on your need.
   - Name the project and click OK.

A

Job Configuration:

3. Job Configuration: Source Code Management (GitHub)

- Source Code Management:
  a. Under the Source Code Management section, select Git.
  b. In the Repository URL, paste the GitHub repository URL where your Maven project is located (e.g., https://github.com/user/repo.git).
  c. Add GitHub credentials (if needed).
  d. Optionally, set up branch (e.g., */main or */develop).

tksahu.devops@gmail.com

4. Job Configuration: Build Triggers (Optional)
   - Under Build Triggers, set up a trigger if you want the job to run automatically (e.g., after a GitHub push, scheduled time, etc.):
     - For GitHub triggers, select GitHub hook trigger for GITScm polling (requires webhook setup on GitHub).
     - For scheduled jobs, use Build periodically and define a cron job (e.g., H/5 * * * * to build every 5 minutes)
     -

5. Job Configuration: Build Steps (Maven, Warnings, JUnit)
   - Maven Build: A
     a. In the Build section, click Add build step.
     b. Select Invoke top-level Maven targets.
     c. Specify the Goals for Maven. Common goals are:
        - Clean, Install, Test: clean install will compile the project, run tests, and package it.
     d. Select the Maven Version installed in Jenkins.
   - Warnings Next Generation:
     e. Add another build step for analyzing code warnings.
     f. Select Record compiler warnings and static analysis results.
     g. Choose the desired report parsers (e.g., Maven, JUnit, etc.) to collect warnings.
     h. Define thresholds for build success/failure based on the number of warnings.
   - JUnit Tests:
     i. After Maven builds, add Publish JUnit test result report in the Post-build Actions section.
     j. Enter the path where the JUnit report will be stored (e.g., **/target/surefire-reports/*.xml for Maven projects).

tksahu.devops@gmail.com

6. Job Configuration: Deploy to Container (Docker/Kubernetes)

Deploying to Docker:

In the Post-build Actions, select Deploy to container or add a Build Step to run a Docker container.

If using Docker, add a shell script or a Docker plugin step to build and run the Docker container:

```
docker build -t your-app-image:${BUILD_ID} .
docker run -d -p 8080:8080 your-app-image:${BUILD_ID}
```

Deploying to Kubernetes:

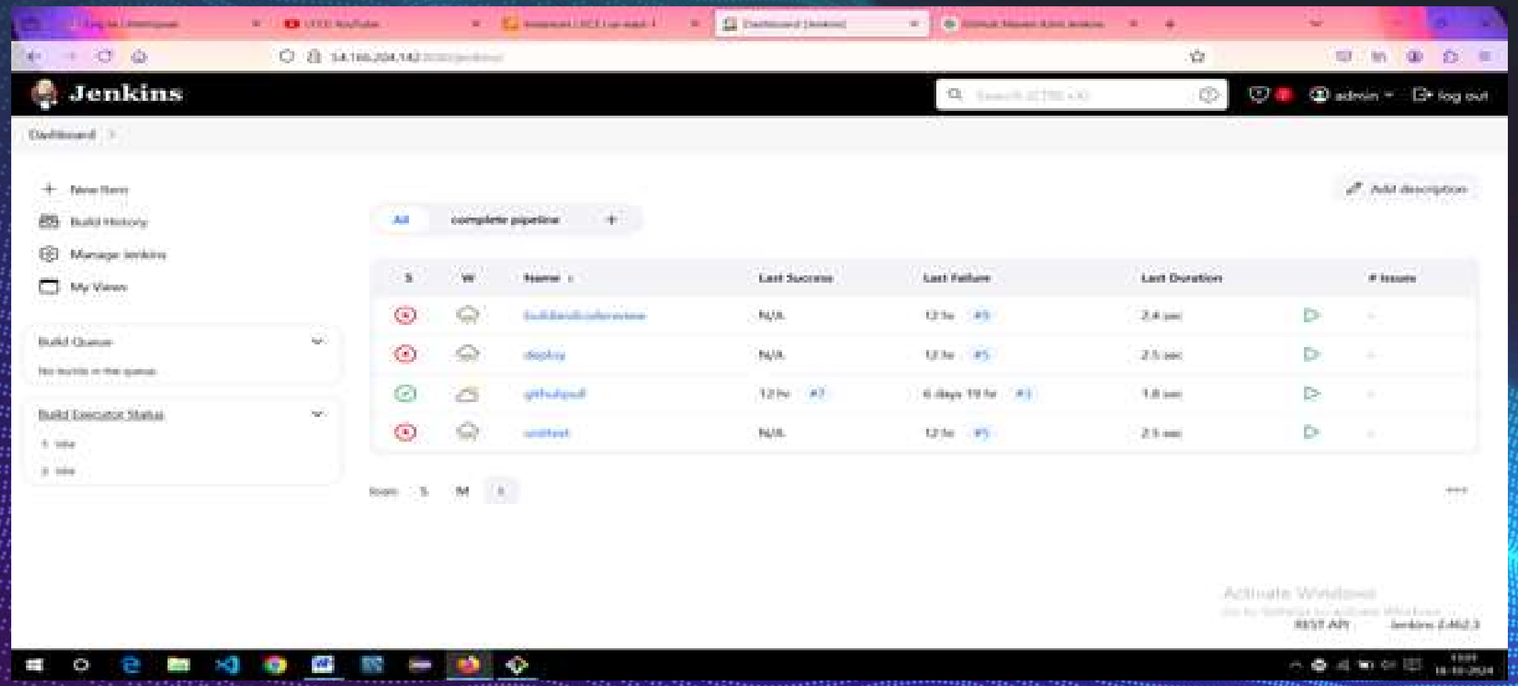
You can apply a Kubernetes manifest in a Post-build step to deploy to a Kubernetes cluster. This may involve using the kubectl command.

7. Post-Build Actions
- Add Post-build Actions to monitor and report the build status.
- Examples include:
  - Email Notification: To notify when the build is successful or fails.
  - Deploy to container: If deployment to a Docker container or Kubernetes is part of your pipeline.

8. Save and Build
- Once you've configured the build steps, click Save.
- Now, trigger the build manually by clicking Build Now or wait for an automatic trigger (if configured).
  a.

tksahu.devops@gmail.com

Based on this project all jobs are configured .
For reference please follow the pdf .



tksahu.devops@gmail.com

Before do the pipeline install “build pipeline” plugins then create pipeline job

What is Poll SCM in Jenkins?

Poll SCM in Jenkins is a build trigger that periodically checks the source code repository for any changes. If any changes are detected (e.g., new commits or updated files in the repository), Jenkins triggers a new build automatically. Unlike webhooks that push changes to Jenkins from the source (e.g., GitHub), Poll SCM allows Jenkins to "poll" or query the repository at regular intervals.

This is useful when:

Webhooks are not available or not configured in the repository.
You want Jenkins to decide when to check for updates based on a schedule.

How to Configure Poll SCM in Jenkins

Follow these steps to set up Poll SCM for a Jenkins job:
1. Access Your Jenkins Job

Go to your Jenkins dashboard.
Click on the job you want to configure for Poll SCM.
In the job dashboard, click "Configure" to modify its settings.

tksahu.devops@gmail.com

2. Enable Poll SCM in Build Triggers

Scroll down to the Build Triggers section.
Check the option "Poll SCM".

3. Set the Polling Schedule

Once you enable Poll SCM, you will see a field to input the polling schedule.
The schedule is written in cron syntax. For example:
H/5 * * * * — Poll the SCM every 5 minutes.
H H * * 1-5 — Poll at a random minute during the first hour of every day, Monday to Friday.
H/10 * * * * — Poll every 10 minutes.
MINUTE: (0-59)
HOUR: (0-23)
DOM (Day of the month): (1-31)
MONTH: (1-12)
DOW (Day of the week): (0-7, where 0 and 7 are Sunday)

Example schedules:

-

H/15 * * * *: Poll the repository every 15 minutes.
H 2 * * 1-5: Poll once daily at a random minute of the second hour from Monday to Friday.
H 22 * * *: Poll once daily at 10:00 PM.

Note: The H is used to hash the time to spread the polling load evenly across multiple jobs in Jenkins to avoid simultaneous polling.
4. Configure the SCM (Source Code Management)

Ensure you've configured your SCM section (e.g., Git or SVN).
For Git, add the Repository URL of your project (e.g., GitHub, GitLab).

5. Save the Configuration

After setting up the polling schedule, scroll down and click "Save".

tksahu.devops@gmail.com

Based on this project pipeline job is configured .
For reference please follow the pdf .