

DEVOPS CI-CD PIPELINE PROJECT

PREREQUISITES

- ❖ Download & Install GIT.
- ❖ Download, Install & Configure Java.
- ❖ Download Jenkins. War file.
- ❖ Download & Install Tomcat.
- ❖ Create Jenkins Home directory.
- ❖ Jenkins home path.
- ❖ Restart PC.
- ❖ Jenkins war file into Tomcat.
- ❖ Run Jenkins.
- ❖ Download & Configure ANT.
- ❖ Restart PC.
- ❖ Now process of project - jobs creation, testing, quality checking & deployment starts.

PLUGINS

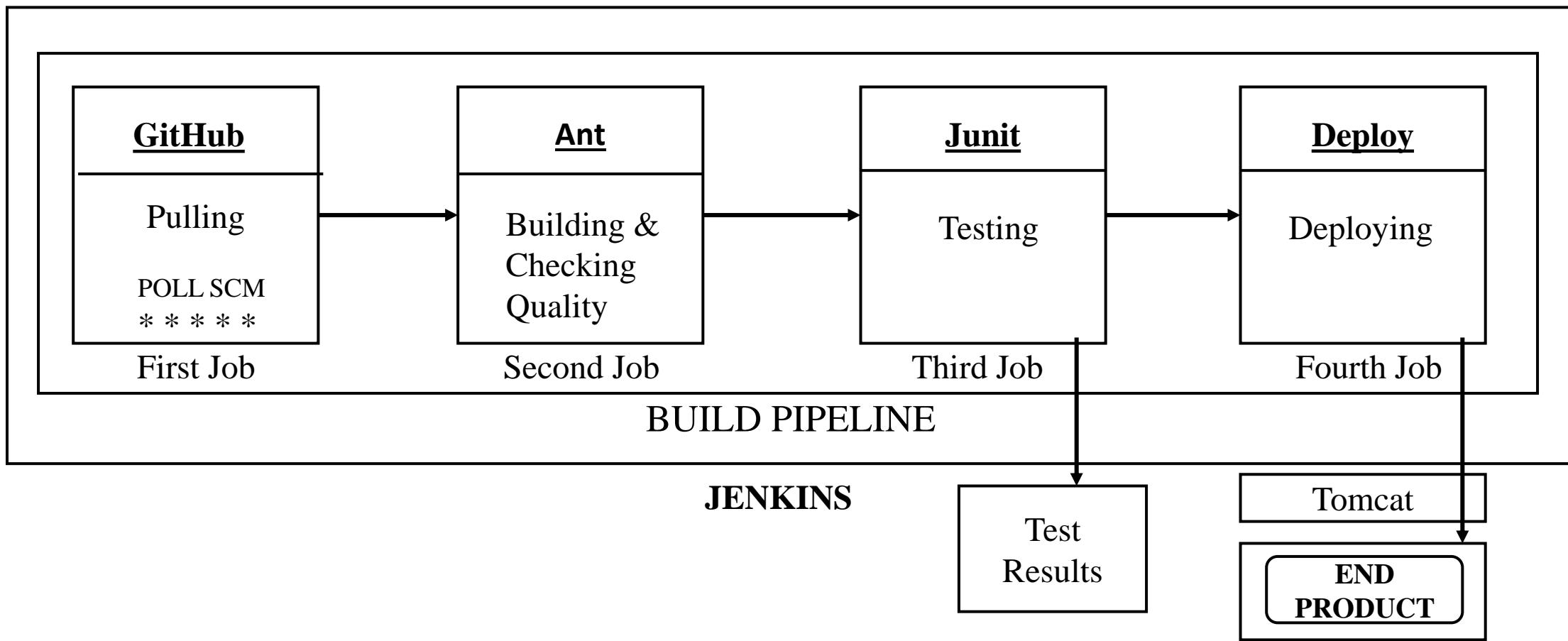
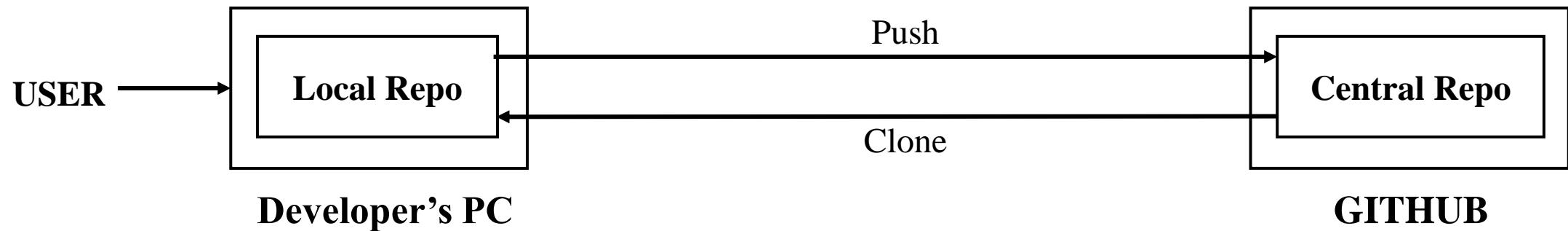
GITHUB.

ANT.

JUNIT.

DEPLOY.

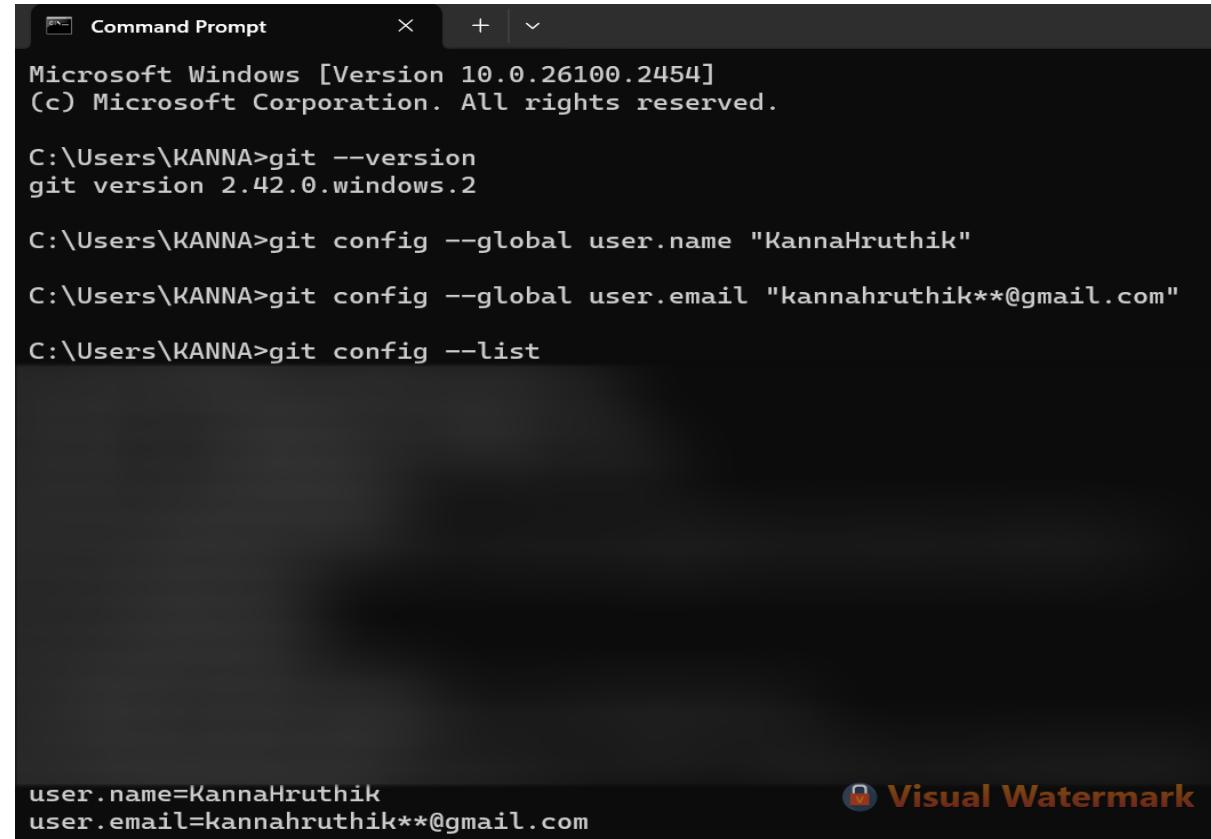
PIPELINE.



PROCESS

STEP-1 DOWNLOADING & INSTALLATION OF GIT :

- Download **GIT** from any browser - Link - <https://git-scm.com/downloads> for windows as per your requirements.
- After downloading, just run it for installation by selecting the requirements, to install GIT in your PC.
- After completion of installation, to verify open “Command Prompt (cmd)” in your pc & run the command - **git –version** .
- Setup the **Username & Email id** for Git, so run – **git config --global user.name “any required username” & git config --global user.email “any required email id”** . To verify run - **git config –list** .



```
Command Prompt
Microsoft Windows [Version 10.0.26100.2454]
(c) Microsoft Corporation. All rights reserved.

C:\Users\KANNA>git --version
git version 2.42.0.windows.2

C:\Users\KANNA>git config --global user.name "KannaHruthik"
C:\Users\KANNA>git config --global user.email "kannahruthik**@gmail.com"
C:\Users\KANNA>git config --list

user.name=KannaHruthik
user.email=kannahruthik**@gmail.com
```

Visual Watermark

STEP -2 DOWNLOADING, INSTALLATION & CONFIGURING OF JAVA :

- Download **JAVA** from any browser – Link - <https://www.oracle.com/in/java/technologies/downloads/#java21> for windows as per your requirements. So I downloaded the JAVA version-21 of x64 Installer in my PC.
- After downloading the Java, to install select the required options it as per your requirements.
- To verify whether the java is installed or not run – **java –version** in cmd prompt.
- Now we need to setup the “**environment variable**” for the JAVA. To setup search for the Jdk folder of java & open it. And copy the path of jdk-21 folder.

C:\Program Files\Java\jdk-21

- Now search for the “**environment variables**” in your windows pc & click on it. A popup window appears, in it click on “**environmental variables**”. Again a new popup window appears , in it click on “**new**” for the “**user varaibles for KANNA**” – now give the “**Variable name as JAVA_HOME & Variable path as C:\Program Files\Java\jdk-21** (the path that you had copied).
- And the same process for the “**below System Variables also Variable name as JAVA_HOME & Variable path as C:\Program Files\Java\jdk-21** (the path that you had copied).
- Now again search for the java jdk folder & copy the path of the bin folder of java jdk.

C:\Program Files\Java\jdk-21\bin

- Now again search for the “**environment variables**” in your windows pc & click on it. A popup window appears, in it click on “**environment variables**”. Now in the **System Variables** there will be the option “**path**” click on in it a popup window appears, then click on the option “**new**” & paste the path of the bin folder of java jdk.
- For Git, by default it chooses the all above mention process.
- To verify, whether the evnvironment variable is correctly set or not to confirm, open the cmd prompt & run – **echo %JAVA_HOME%** so it must show the java jdk folder path.

Environment Variables

X

User variables for KANNA

Variable	Value
JAVA_HOME	C:\Program Files\Java\jdk-21

New...Edit...Delete

System variables

Variable	Value
JAVA_HOME	C:\Program Files\Java\jdk-21

New...Edit...Delete

Edit environment variable

X

New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...

C:\Program Files\Java\jdk-21\bin

Microsoft Windows [Version 10.0.26100.2454]
(c) Microsoft Corporation. All rights reserved.

```
C:\Users\KANNA>echo %JAVA_HOME%
C:\Program Files\Java\jdk-21
```

 Visual Watermark

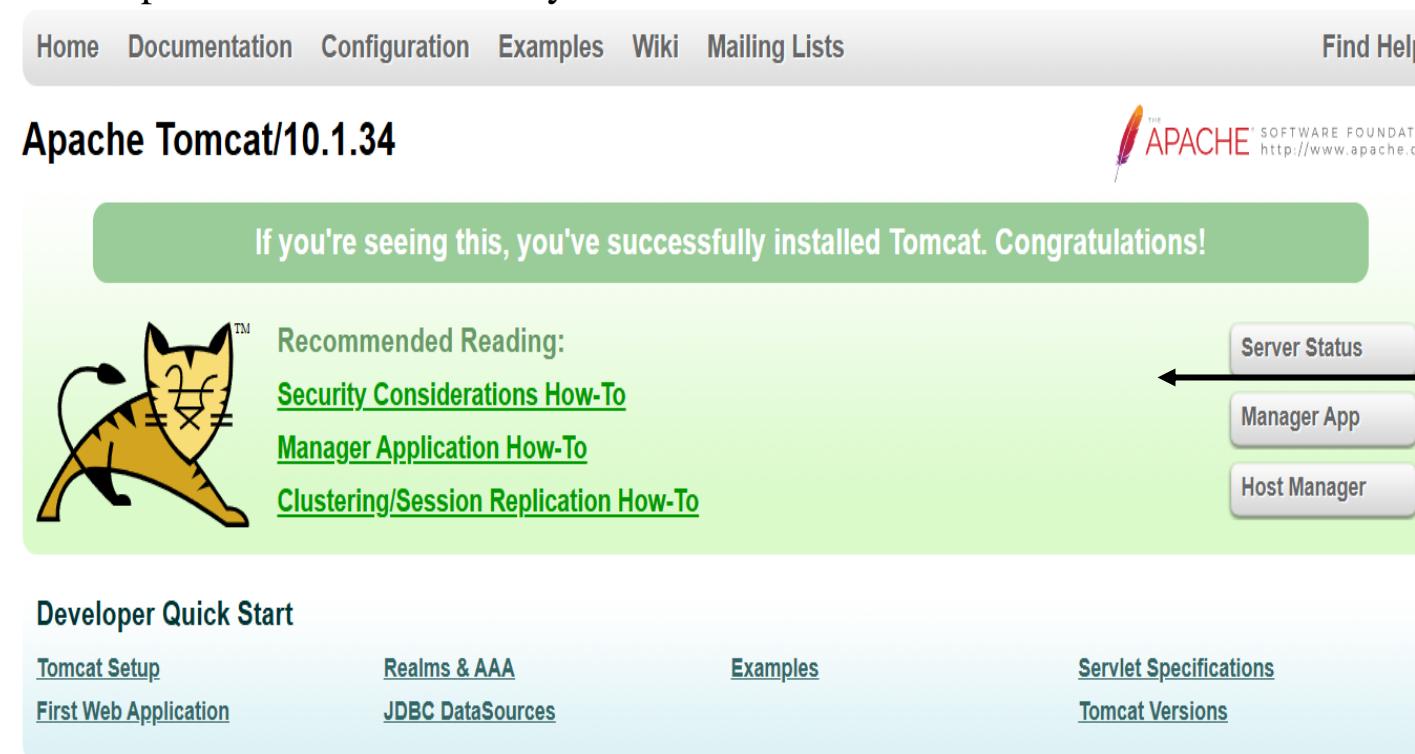
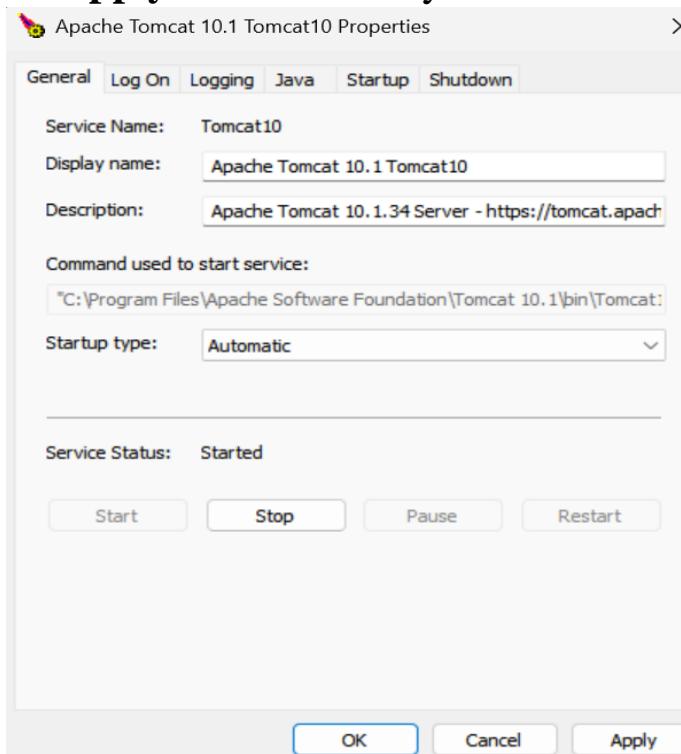
Cancel

 Visual Watermark

Cancel

STEP-3 DOWNLOADING & INSTALLATION OF TOMCAT :

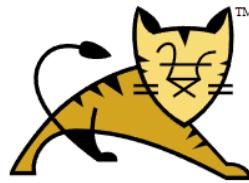
- Download **TOMCAT** from any browser – Link - <https://tomcat.apache.org/> for windows as per your requirements. So, I downloaded the tomcat 10 version - <https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.34/bin/apache-tomcat-10.1.34.exe> as per my PC requirements.
- After downloading, just install it by giving required requirement access like creation of username & password & give the installed Java JDK path & click on to install it, at last unselect the readme option & select to run the Apache tomcat option to finish.
- Now search for tomcat in your PC , click on it to open & now in **General** → **Startup type** → **Select Automatic & Click on apply**. So that when you switch on the PC & open the tomcat it directly runs tomcat.



After accessing of Tomcat via your browser, You will get this type of interface

- To access tomcat, go to any browser, run - <https://localhost:8080/> .
- To install web applications in tomcat, select the “**Manager App**” option in tomcat server & give the username & password of tomcat that you had created while installing the tomcat.

- After clicking on “Manager App” we are required to give the credentials of tomcat username & password. Then we will get this interface.



Tomcat Web Application Manager

Message: OK					
Manager					
List Applications		HTML Manager Help		Manager Help	
				Server Status	
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	<i>None specified</i>	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ <input type="text" value="30"/> minutes
/docs	<i>None specified</i>	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ <input type="text" value="30"/> minutes
/manager	<i>None specified</i>	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ <input type="text" value="30"/> minutes

- So basically the tomcat server runs on the “ Port – 8080”.
- NOTE :** By default both the tomcat & Jenkins runs on the port-8080, to avoid that conflict we will change the tomcat accessing port to our required port number, so that we can avoid the conflict of accessing the both tomcat & Jenkins.
- So, the changing of port process will be shown in further process...

- **STEP -4 DOWNLOADING JENKINS :**

- Basically we can download the Jenkins in two ways -

1. By Jenkins.exe
2. By Jenkins. War

- **Jenkins by .exe** – we normally download Jenkins in windows by – Link - <https://www.jenkins.io/download/thank-you-downloading-windows-installer-stable>

- And we install it based on our requirements like checking the port – 8080, Creation user and password, Accessing of the secret initial password to access the Jenkins for the first time, at last we install our required plugins. & now we are good to go and use the Jenkins.

- By default the Jenkins can be accessed by the “ **Port – 8080**”.

- By doing this process we get conflict between the Tomcat port and Jenkins port. To avoid it we will use a different method.

- **Jenkins by .War** - So now, I had downloaded the Jenkins via .war file in windows by – Link - <https://get.jenkins.io/war-stable/2.479.2/jenkins.war>

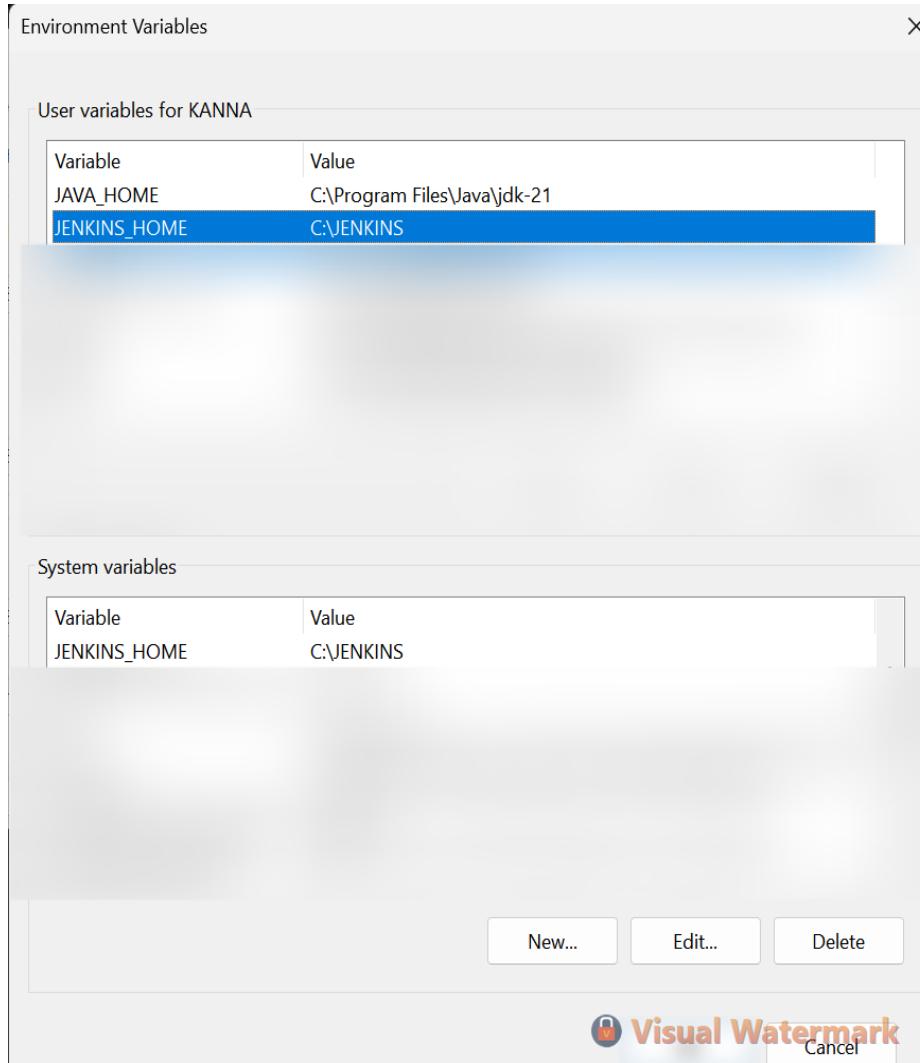
- Here after downloading the Jenkins. War file, we can't install it directly because our machines cannot able to understand the .war files format. So here we use the tomcat server (Which is used to run & execute the .war files, web applications etc).

- By default when we download & install Jenkins in the PC, it automatically chooses its own path. To avoid it we are using this method.

- By downloading in this way via .war file, I need to change the default port of Tomcat & I should give my own path for Jenkins & create a new directory to Jenkins that it can store it's resources in it to setup the customised “**environment variable**”.

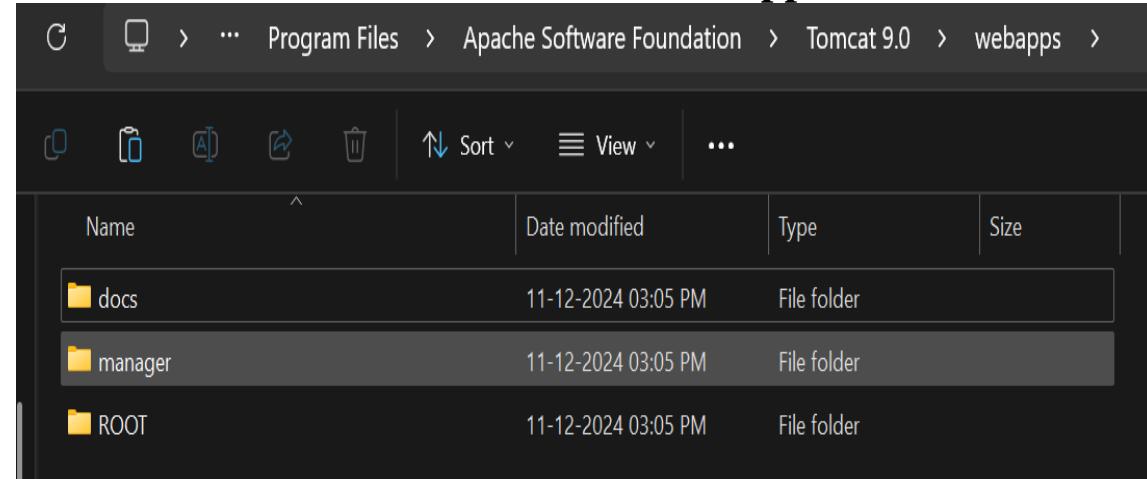
- **Create a new folder/directory in any location (preferred in C-drive) → Enter into the folder & copy the path of the folder to create environment variable → Now search for the “environment variables” in your windows pc & click on it. A popup window appears, in it click on “environmental variables”. Again a new popup window appears , in it click on “new” for the “user variables for KANNA” – now give the “Variable name as JENKINS_HOME & Variable path as C:\JENKINS (the path that you had copied). → PTO...**

- And the same process for the “ **below System Variables also Variable name as JENKINS_HOME & Variable path as C:\JENKINS** (the path that you had copied).
- To verify, whether the environment variable is correctly set or not, open the cmd prompt & run – **echo %JENKINS_HOME%** so it must show the jenkins folder path.



- Before running the Jenkins .War file in tomcat server, we need to restart the PC, so that the OS recognises the Java, Tomcat & Jenkins home directory.

- Now we need to access the tomcat server via this link - <https://localhost:8080> . And click on “manager app” option to login into the tomcat server with your credentials.
- Now the entire tomcat server will be in your local host machine, to see it go to this path in your local host - **C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps** .

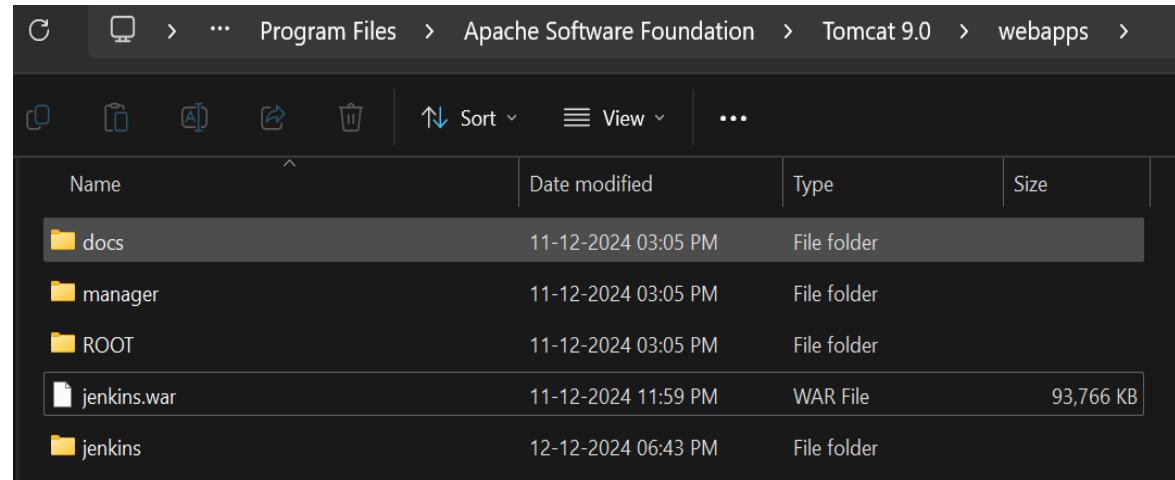


Tomcat Web Application Manager

Message: OK

Manager

List Applications		HTML Manager Help	Manager Help	Server Status	
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>
/jenkins	None specified	Jenkins v2.479.2	true	0	Start Stop Reload Undeploy <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy <input type="button" value="Expire sessions with idle ≥ 30 minutes"/>



Now copy & paste the Jenkins .war file in folder of webapps
C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps inside webapps we need to paste the Jenkins .war file, so it automatically creates Jenkins folder in webapps folder. Once when you hit refresh in your tomcat server web page it automatically displays the Jenkins.

- Now to access Jenkins, in tomcat server – Right click on the Jenkins to open it in the new tab to access Jenkins.

Getting Started

Unlock Jenkins

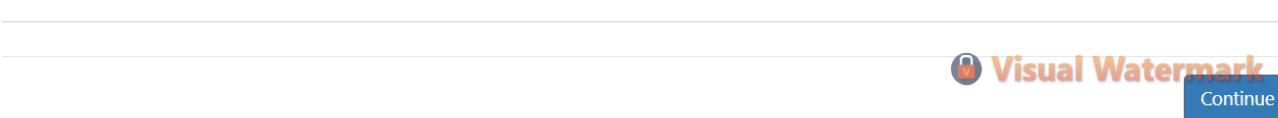
To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

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

Administrator password



Here the Jenkins secret password path will be displayed.



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

- Now paste the secrets initial password to access the plugins page.
- In this both options select your required option. And the further process goes on, by creating the first user details, then next it displays Jenkins accessible link - <http://localhost:8080/jenkins/>. Then now we are good go to start using Jenkins.

- The interface of Jenkins looks like this when you login...
- **NOTE : When I downloaded the tomcat version 9, there is a compatibility issue with Jenkins and tomcat, so I downloaded the tomcat version 10. It started working fine.**

The screenshot shows the Jenkins dashboard. At the top, there is a navigation bar with the Jenkins logo, a search bar containing "Search (CTRL+K)", a help icon, a notifications icon with a red dot (showing 1 notification), the user name "admin", and a "log out" button. Below the navigation bar, the page title is "Dashboard >". On the left side, there is a sidebar with links: "New Item", "Build History", "Manage Jenkins", and "My Views". Under "Manage Jenkins", there is a "Build Queue" section which says "No builds in the queue." and a "Build Executor Status" section which says "0/2". In the center, the main content area has a heading "Welcome to Jenkins!". Below it, a message says "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." There is a "Start building your software project" button. To the right of this, there is a "Create a job" button with a plus sign. Further down, there is a "Set up a distributed build" section with three items: "Set up an agent" (with a monitor icon), "Configure a cloud" (with a cloud icon), and "Learn more about distributed builds" (with a question mark icon).

Jenkins

Search (CTRL+K)

! 1 admin log out

Dashboard >

+ New Item

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

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 ?

• **STEP -4 DOWNLOADING & CONFIGURATION OF ANT :**

- Download **ANT** from any browser – Link - <https://ant.apache.org/bindownload.cgi> and in it I downloaded <https://dlcdn.apache.org//ant/binaries/apache-ant-1.10.15-bin.zip>
- After completion of download, go to your downloads folder and right click on the Apache-ant folder , then next click on extract all, then after give your required path to extract it.
- Now we need to setup the “**environment variable**” for the APACHE-ANT. To setup search for the apache-ant-1.10.15 folder & open it. And copy the path of apache-ant-1.10.15 folder.

C:\apache-ant-1.10.15

- Now search for the “**environment variables**” in your windows pc & click on it. A popup window appears, in it click on “**environmental variables**”. Again a new popup window appears , in it click on “**new**” for the “**user varaibles for KANNA**” – now give the “**Variable name as ANT_HOME & Variable path as C:\apache-ant-1.10.15** (the path that you had copied).
- And the same process for the “ **below System Variables also Variable name as ANT_HOME & Variable path as C:\apache-ant-1.10.15** (the path that you had copied).
- Now again search for the Apache-ant folder & copy the path of the bin folder of Apache-ant.

C:\apache-ant-1.10.15\bin

- Now again search for the “**environment variables**” in your windows pc & click on it. A popup window appears, in it click on “**environment variables**”. Now in the **System Variables** there will be the option “**path**” click on it a popup window appears, then click on the option “**new**” & paste the path of the bin folder of Apache-ant.
- To verify, open cmd prompt & run **ant –version** .
- To verify, whether the evnironment variable is correctly set or not to confirm, open the cmd prompt & run – **echo %ANT_HOME%** so it must show the Apache-ant folder path.

Environment Variables

X Edit environment variable X

User variables for KANNA

Variable	Value
ANT_HOME	C:\apache-ant-1.10.15
JAVA_HOME	C:\Program Files\Java\jdk-21
JENKINS_HOME	C:\JENKINS

New

Edit

Browse...

Delete

Move Up

Move Down

Edit text...

System variables

Variable	Value
ANT_HOME	C:\apache-ant-1.10.15
JAVA_HOME	C:\Program Files\Java\jdk-21
JENKINS_HOME	C:\JENKINS

Path

New...

Edit...

Delete

C:\Program Files\Java\jdk-21\bin
C:\apache-ant-1.10.15\bin

Command Prompt

Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. All rights reserved.

C:\Users\KANNA>ant -version
Apache Ant(TM) version 1.10.15 compiled on August 25 2024

C:\Users\KANNA>echo %ANT_HOME%
C:\apache-ant-1.10.15

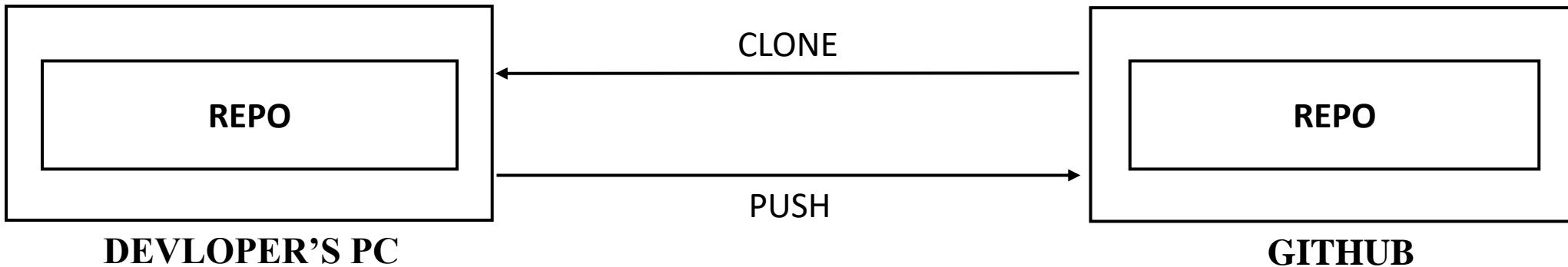
Now we need to restart the PC again, So that the Jenkins inside the tomcat & Ant gets recognized by the O.S.

- **STEP -5 PROJECT STARTS :**

- Now open the tomcat server & login into the server via your credentials through the manager app to access the Jenkins or if you want to access the Jenkins directly via link <http://localhost:8080/jenkins/>

- **PLUGINS INSTALLATION :**

- To install the required plugins **Click on manage Jenkins** → **Click on manage plugins** → **Click on available** → **Search for the required plugins** → **& select GITHUB, ANT, JUNIT REALTIME TEST REPORTER, DEPLOY TO CONTAINER, BUILD PIPELINE** → **Click on install without restart** → **Click on back to top page.**



- First, the developer clones the GitHub repo & then he/she add or commits any changes to code then they add, commit, & pushes to GitHub repo.
- Now we need to fork the GitHub repo of the sample code to our GitHub account. I forked this <https://github.com/SaiDevOpsFaculty/SampleWebApp>
- Now after clone the forked GitHub repo by copying its GitHub https URL of repo into our local machine. To do it, open the cmd & run
git clone <https://github.com/hruthikkanna/SampleWebApp.git> (GitHub repo HTTPS url).
- After cloning, to verify run **dir** so that you need to see the GitHub folder named – “sample web app folder”.

forked from ajaynsrinivas/SampleWebApp

Notifications

Fork 828

Star 5

Code Pull requests 10 Actions Projects Wiki Security Insights

master 1 Branch 0 Tags

Go to file

Code

This branch is 81 commits ahead of ajaynsrinivas/SampleWebApp:master .

#14

SaiDevOpsFaculty modified

cce806f - 3 weeks ago

93 Commits

WebContent modified 3 weeks ago

build/classes modified 6 years ago

src/example done change 6 years ago

TestCalculator_JUnitResult.txt.txt done test changes 6 years ago

About

No description, website, or topics provided.

Activity

5 stars

2 watching

828 forks

Report repository

Releases

No releases published

hruthikkanna / SampleWebApp

Code Pull requests Actions Projects Wiki Security Insights Settings

SampleWebApp Public

forked from SaiDevOpsFaculty/SampleWebApp

Pin Watch 0

Fork 0

Star 0

master 1 Branch 0 Tags

Go to file

Add file

Code

Local

Codespaces

Clone

HTTPS SSH GitHub CLI

Clone using the web URL.

Open with GitHub Desktop

Download ZIP

About

No description, website, or topics provided.

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Visual Watermark

Packages

- I Used this GitHub folder to fork the code into my GitHub account.

- After forking the Git repo, now clone that Repo into your local machine by using the HTTPS url.

Command Prompt

```
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. All rights reserved.

C:\Users\KANNA>git clone https://github.com/
Cloning into 'SampleWebApp'...
remote: Enumerating objects: 244, done.
remote: Counting objects: 100% (84/84), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 244 (delta 68), reused 67 (delta 63), pack-reused 160 (from 1)
Receiving objects: 100% (244/244), 16.83 MiB | 7.27 MiB/s, done.
Resolving deltas: 100% (78/78), done.

C:\Users\KANNA>dir
Volume in drive C is OS
Volume Serial Number is 0E6F-5B99

Directory of C:\Users\KANNA

13-12-2024 12.19 PM <DIR> SampleWebApp

Visual Watermark
```

- After cloning the Git repo folder into our local machine, now we need to create four jobs in Jenkins.

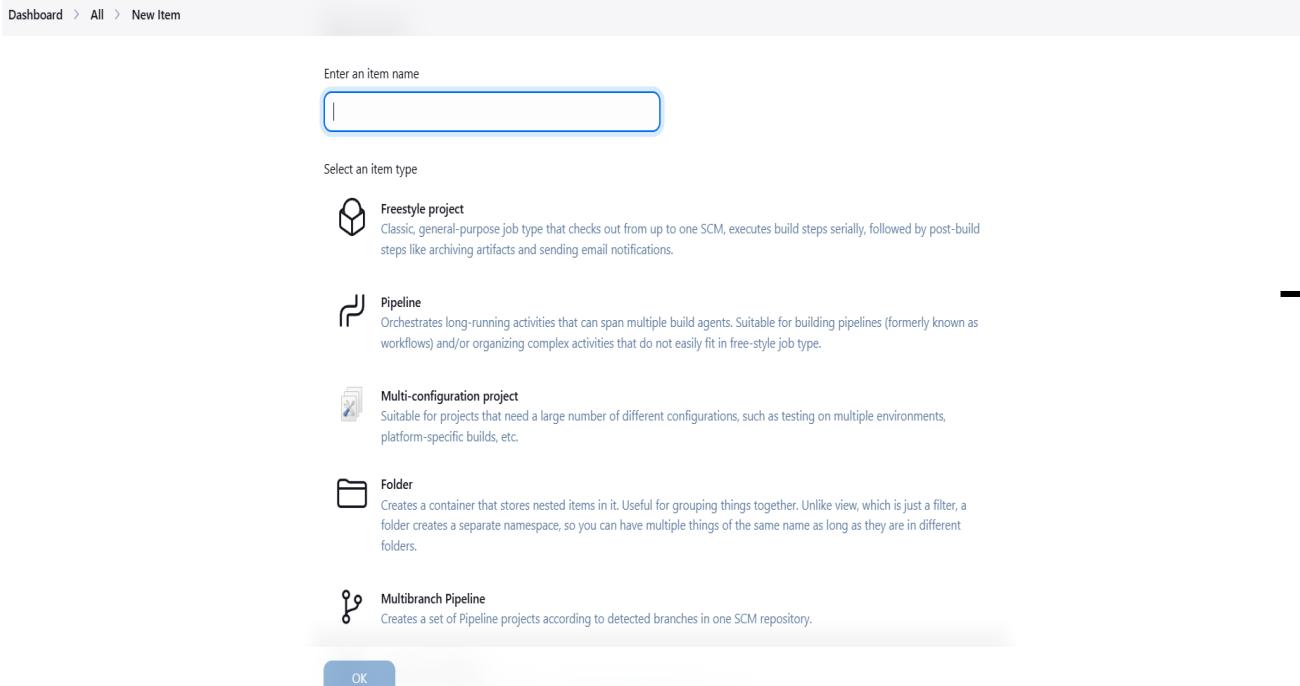
- **JOB – 1**



This job just pulls the code from the GitHub repo.

- **Process of Job creation :-**

- Go to the tomcat server & Login into the tomcat server via your credentials, & now again open the Jenkins & login into it via your credentials → Click on New item → Give any required name & select **free style project** → click on ok
Then if you want to give your custom workspace path (Then in **General** → Click on **advanced** & select “ **Use custom workspace** ” & give required path Ex - \${JENKINS_HOME}/workspace/Sample Web App) → In **Source Code Management** select “ **GIT** ” & paste the **HTTPS url of GitHub repo** → Scroll down, click on apply & save.



→ This is the interface, you will see while Creating the job process...

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Retry Count ?

Block build when upstream project is building ?

Block build when downstream project is building ?

Use custom workspace ?

Directory

`$(JENKINS_HOME)/workspace/SampleWebApp`

Display Name ?

Keep the build logs of dependencies ?

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Source Code Management

None

Git ?

Repositories ?

Repository URL ?

`https://github.com/`

Please enter Git repository.

Credentials ?

- none -

+ Add

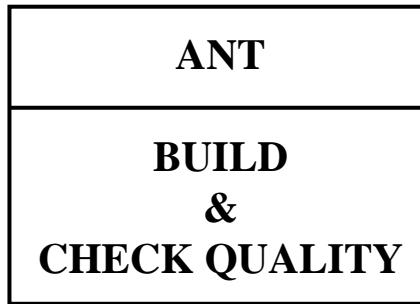
Advanced...

 Visual Watermark

Save Apply

Custom workspace path.
&
Process of pulling the code from GitHub repo.

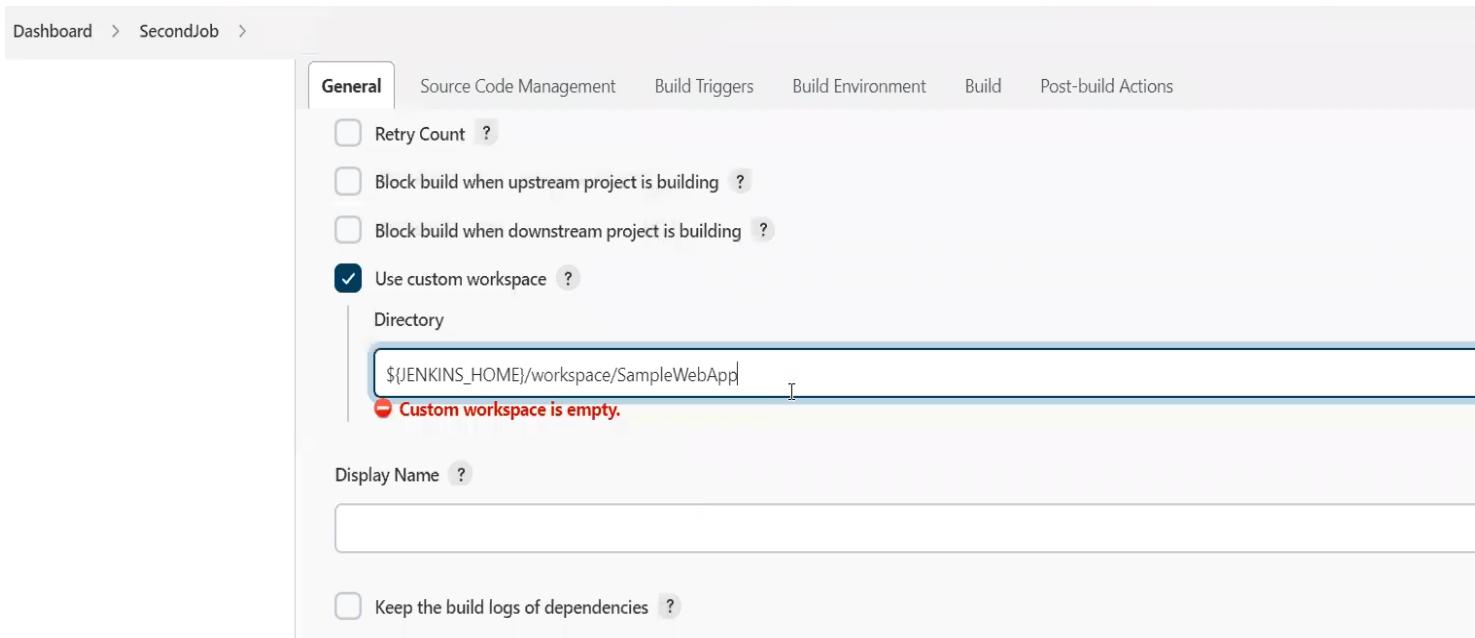
- **JOB – 2**



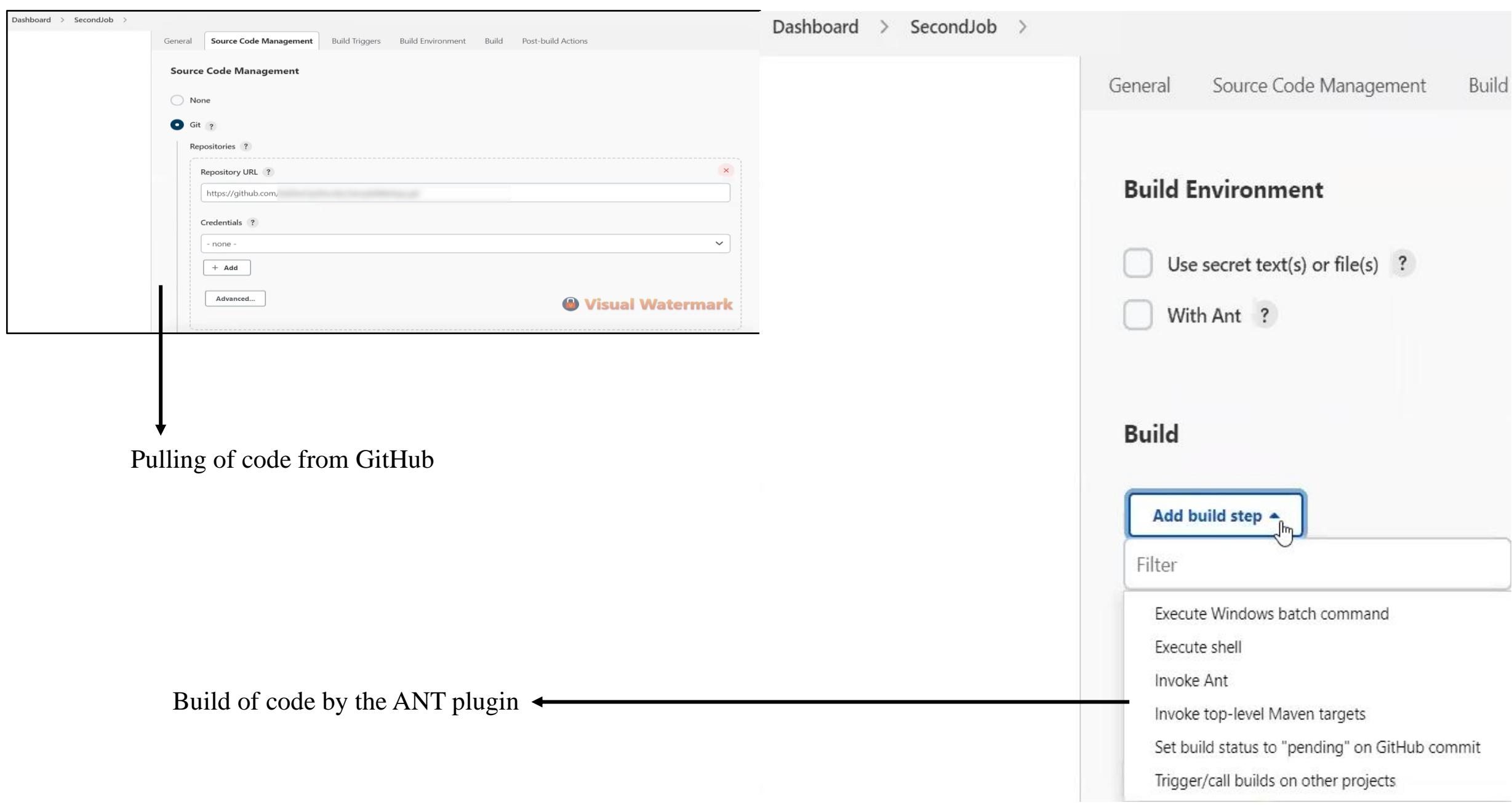
This job also pulls the code from the GitHub repo to build the job & to check its quality.

- **Process of Job creation :-**

- Click on New item → Give any required name & select **free style project** → click on ok. Then if you want to give your custom workspace path (Then in **General** → Click on **advanced** & select “ **Use custom workspace** ” & give required path Ex - \${JENKINS_HOME}/workspace/Sample Web App) → In **Source Code Management** select “ **GIT** ” & paste the **HTTPS url of GitHub repo** → Scroll down, In **Build** – Click on “ **ADD BUILD STEP** ” & Select “ **INVOKE ANT** ” & In **Targets** section give “ **checkstyle** ” → click on apply & save.



→ Custom workspace path.



General

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

 Use secret text(s) or file(s) ? With Ant ?**Build****Invoke Ant** ?

Targets ?

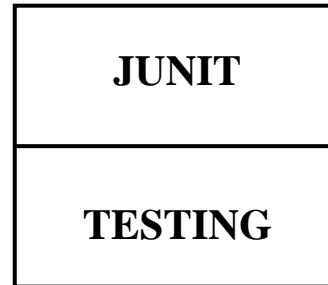
checkstyle I

Advanced...

Add build step ▾

Post-build Actions**Save****Apply**Checking quality of code by using the **checkstyle** target

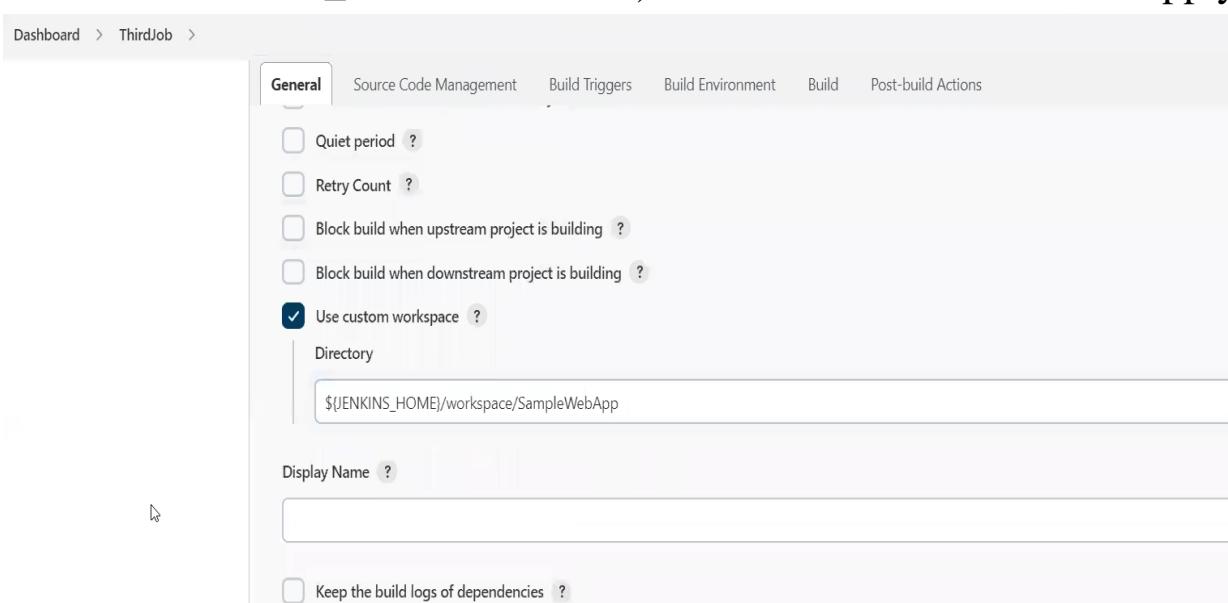
- **JOB – 3**



This job also pulls the code from the GitHub repo to build the job & to check its quality, then after to Test the code.

- **Process of Job creation :-**

- Click on New item → Give any required name & select **free style project** → click on ok. Then if you want to give your custom workspace path (Then in **General** → Click on **advanced** & select “ **Use custom workspace** ” & give required path Ex - \${JENKINS_HOME}/workspace/Sample Web App) → In **Source Code Management** select “ **GIT** ” & **paste the HTTPS url of GitHub repo** → Scroll down, In **Build** – Click on “ **ADD BUILD STEP** ” & Select “ **INVOKE ANT** ” & In **Targets** section give “ **junit** ” → In **Post Build actions** – Click on “ **ADD PORT BUILD ACTIONS** ” & select “ **PUBLISH JUNIT TEST RESULT REPORT OPTION** ” & In **Test results xml’s** give “ **any required .xml file** (Ex- **Testcalculator_JunitResult.xml**) → Now Click on apply & save.



→ Custom workspace path.

Dashboard > ThirdJob >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Keep the build logs of dependencies ?

Source Code Management

None

Git ?

Repositories ?

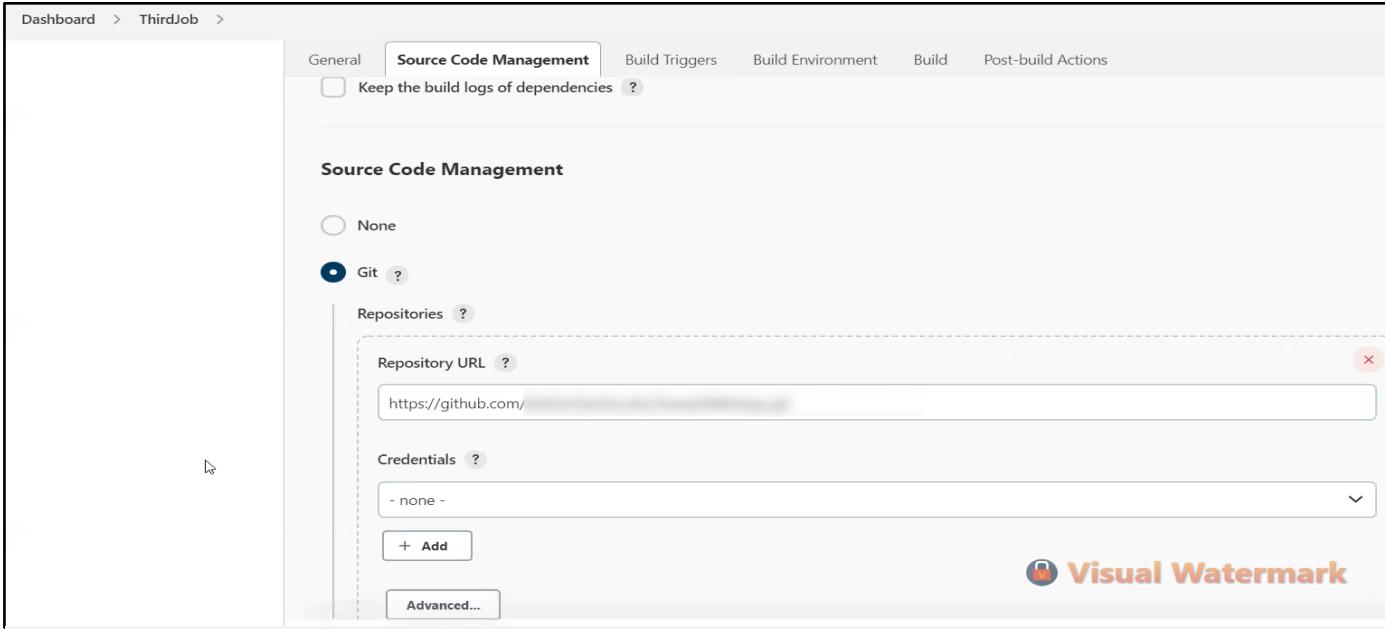
Repository URL ?

Credentials ?

+ Add

Advanced...

 Visual Watermark



Pulling of code from GitHub.

Dashboard > ThirdJob >

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

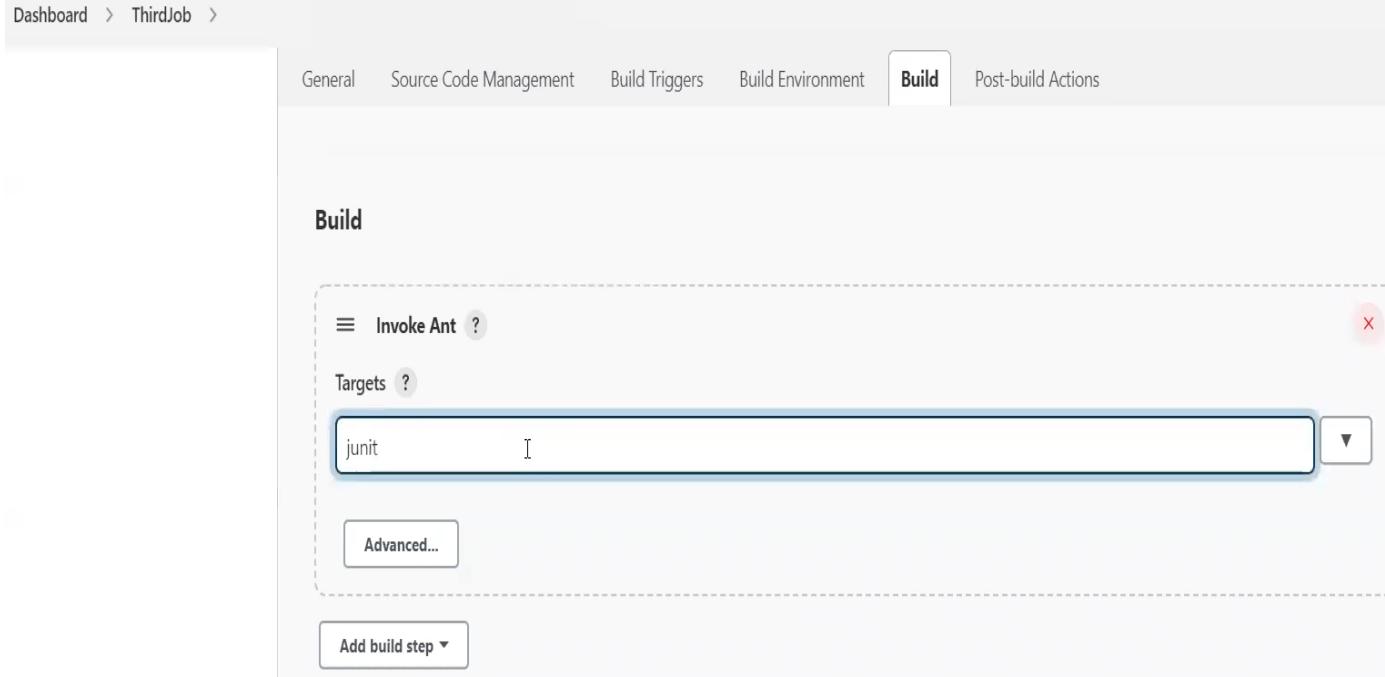
Build

☰ Invoke Ant ?

Targets ?

Advanced...

Add build step ▾



Junit testing of code.

General Source Code Management Build
Build

☰ **Invoke Ant** ?

Filter

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Publish JUnit test result report** 
- Record fingerprints of files to track usage
- Git Publisher
- Build other projects (manual step)
- Deploy war/ear to a container
- E-mail Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Trigger parameterized build on other projects

Add post-build action ▾

To get the test result report

General Source Code Management Build Triggers Build Environment Build **Post-build Actions**

☰ **Publish JUnit test result report** ?

Test report XMLs

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*xml'. Basedir of the fileset is the workspace root.

TestCalculator_JUnitResult.xml

Retain long standard output/error

Health report amplification factor ?

1.0

1% failing tests scores as 99% health. 5% failing tests scores as 95% health

Allow empty results ?

Do not fail the build on empty test results

Skip publishing checks ?

If unchecked, then issues will be published to SCM provider platforms

Save

Apply

Publishes the Junit test result report in .xml file

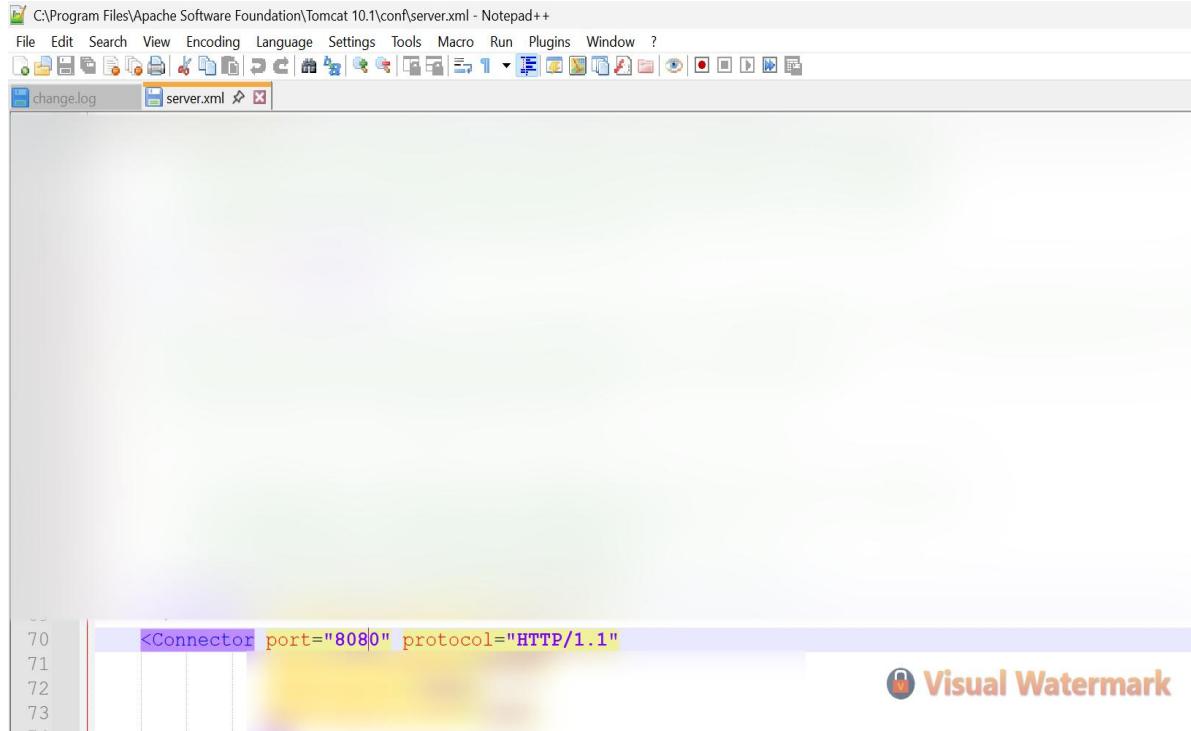
- **Changing the Tomcat port :-** (Not Mandatory)

- By default, the Tomcat runs on the port-8080, But we can change its port to our required port number to avoid conflicts between the Tomcat port & Jenkins port.

- **PROCESS OF PORT CHANGING :-**

- So open the C-drive —————> Program Files —————> Apache Software Foundation folder —————> Tomcat 10 —————> Conf

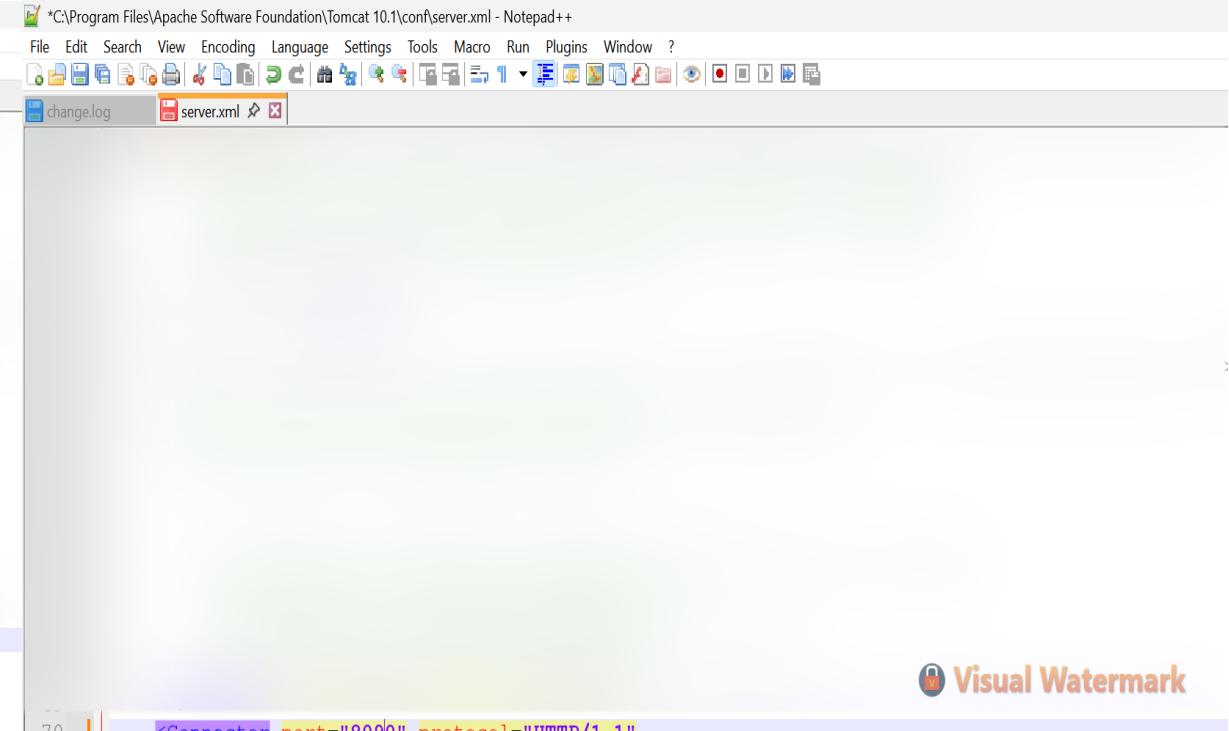
—————> Now Open the “ Server ” xml file via any application ——————> Now in the **Server file** in between there will be **Connector port = “8080”** (By default port) , **Change 8080** to your required port number **Ex- 8090** ——————> & Save it.



```
C:\Program Files\Apache Software Foundation\Tomcat 10.1\conf\server.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log server.xml

<Connector port="8080" protocol="HTTP/1.1"
```

Before changing of the Tomcat Port



```
*C:\Program Files\Apache Software Foundation\Tomcat 10.1\conf\server.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log server.xml

<Connector port="8090" protocol="HTTP/1.1"
```

After Changing of the Tomcat Port

- After saving it, then also the tomcat server runs on the port 8080 only, After restarting the tomcat server it changes its default port 8080 to our mentioned port 8090. The restart process of tomcat is pending....

The screenshot shows a web browser window with the URL "localhost:8080" in the address bar. The page content is the Apache Tomcat 10.1.34 welcome page. At the top, there is a navigation bar with links to Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists, along with a "Find Help" button. To the right of the navigation bar is the Apache Software Foundation logo. The main heading is "Apache Tomcat/10.1.34". Below the heading, a green banner displays the message "If you're seeing this, you've successfully installed Tomcat. Congratulations!". To the left of this message is a cartoon illustration of a yellow cat. To the right of the message are three links under the heading "Recommended Reading": "Security Considerations How-To", "Manager Application How-To", and "Clustering/Session Replication How-To". On the far right, there are three buttons labeled "Server Status", "Manager App", and "Host Manager". At the bottom of the page, there is a "Developer Quick Start" section with links to "Tomcat Setup", "First Web Application", "Realms & AAA", and "JDBC DataSources". There are also links for "Examples", "Servlet Specifications", and "Tomcat Versions". A "Visual Watermark" logo is visible in the bottom right corner.

← → C ⓘ localhost:8080

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/10.1.34

If you're seeing this, you've successfully installed Tomcat. Congratulations!

TM

Recommended Reading:

[Security Considerations How-To](#)

[Manager Application How-To](#)

[Clustering/Session Replication How-To](#)

Server Status

Manager App

Host Manager

Developer Quick Start

[Tomcat Setup](#)

[First Web Application](#)

[Realms & AAA](#)

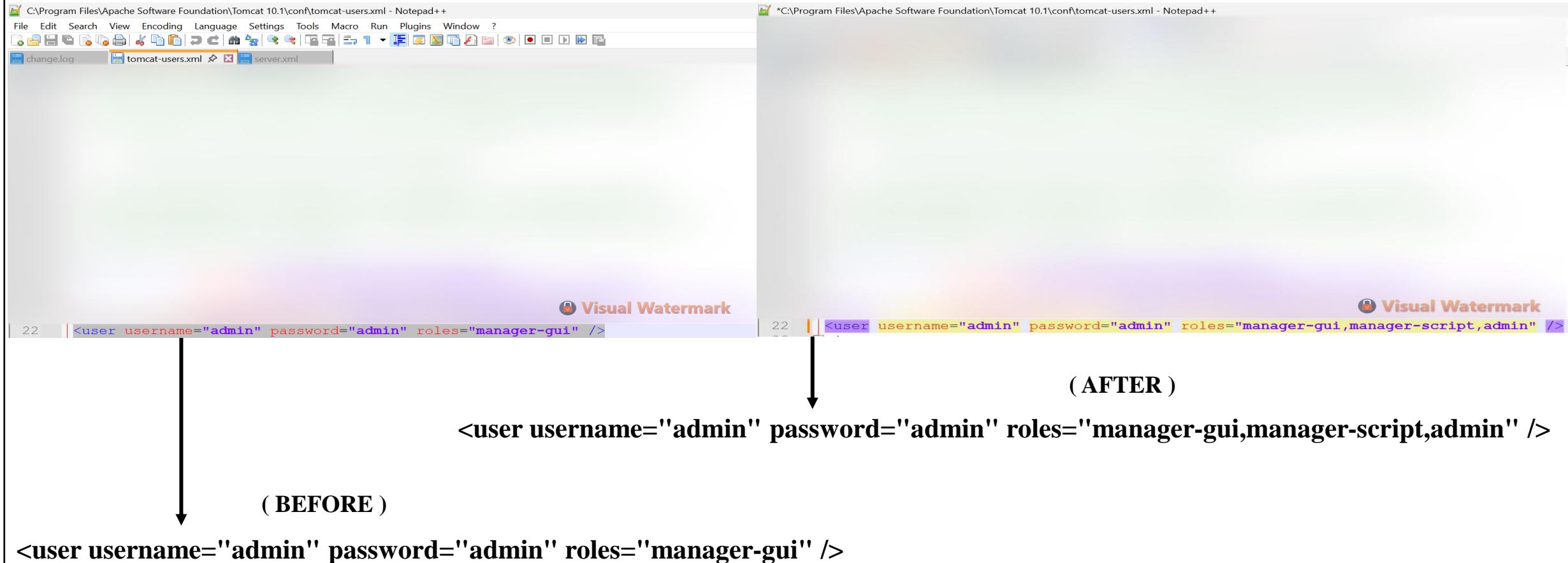
[JDBC DataSources](#)

[Examples](#)

 [Servlet Specifications](#)

[Tomcat Versions](#)

- So, we need to open the ports of tomcat, So that Jenkins can deploy its .war file into the tomcat server.
- To do it open the C-drive —————> Program Files —————> Apache Software Foundation folder —————> Tomcat 10
————> Conf —————> Open the “Tomcat-users” xml file via any application —————> In that file <user username = “admin” line will be at last of that line beside of manager-gui Add ,manager-script,admin —————> & Save it.



The image shows two side-by-side screenshots of the Notepad++ text editor. Both windows have the title 'C:\Program Files\Apache Software Foundation\Tomcat 10.1\conf\tomcat-users.xml - Notepad++'. The left window is labeled '(BEFORE)' and shows the XML code:

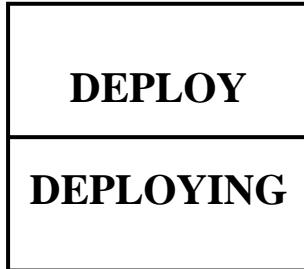
```
<user username="admin" password="admin" roles="manager-gui" />
```

The right window is labeled '(AFTER)' and shows the XML code with an additional role added:

```
<user username="admin" password="admin" roles="manager-gui,manager-script,admin" />
```

A vertical double-headed arrow on the left points down to the word 'roles' in the 'BEFORE' code. A vertical double-headed arrow on the right points down to the word 'roles' in the 'AFTER' code. A horizontal double-headed arrow points from the 'BEFORE' code to the 'AFTER' code, indicating the change made.

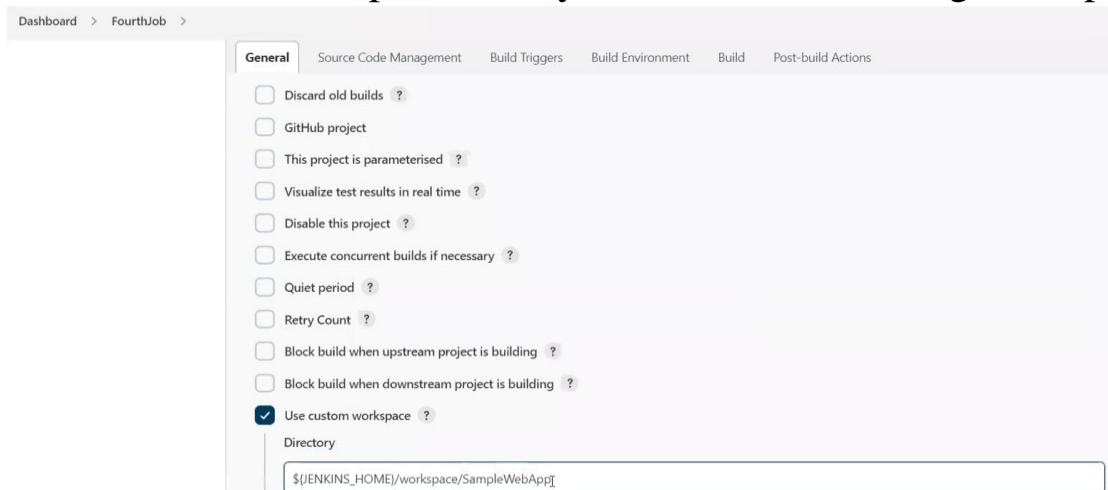
- **JOB – 4**



This job also pulls the code from the GitHub repo to build the job & to check its quality, then after to Test the code & It deploys.

- **Process of Job creation :-**

- Click on New item → Give any required name & select **free style project** → click on ok. Then if you want to give your custom workspace path (Then in **General** → Click on **advanced** & select “ **Use custom workspace** ” & give required path Ex - \${JENKINS_HOME}/workspace/Sample Web App) → In **Source Code Management** select “ **GIT** ” & **paste the HTTPS url of GitHub repo** → Scroll down, In **Build** – Click on “ **ADD BUILD STEP** ” & Select “ **INVOKE ANT** ” & In **Targets** section give “ **war** ” → In **Port Build actions** – Click on “ **ADD PORT BUILD ACTIONS** ” & select “ **DEPLOY WAR/EAR TO A CONTAINER OPTION** ” Now in **Deploy to container** section give ****/*.war** & In **context** give “ **any required name for the end product (ex – Samplewebapp)** ” & In **Containers**, Click on add containers → Select **Tomcat Version** (ex-**Tomcat version 10 Version**) → In **Tomcat Credentials**, Click on add & give “ **Tomcat Username & Password** ” & Now in **Credentials** dropdown Select **Tomcat Details**, → In **Tomcat URL** give the <http://localhost:8090/> (When we restart the Tomcat server runs on this port directly because we had changed the port.) → Click on apply & save.



→ Custom workspace path

Dashboard > FourthJob >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

None

Git ?

Repositories ?

Repository URL ?

Visual Watermark X

https://github.com/

This screenshot shows the 'Source Code Management' tab for a Jenkins job named 'FourthJob'. It has a 'General' tab and a 'Source Code Management' tab, which is currently selected. Under 'Source Code Management', there are two options: 'None' and 'Git'. The 'Git' option is selected, indicated by a blue dot next to it. Below the 'Git' section, there is a 'Repositories' section with a 'Repository URL' input field containing 'https://github.com/'. A watermark for 'Visual Watermark' is visible in the background.

Pulling of code from GitHub

Dashboard > FourthJob >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Build Environment

Use secret text(s) or file(s) ?

With Ant ?

Build

Add build step ▾

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant**
- Invoke top-level Maven targets
- Set build status to "pending" on GitHub commit
- Trigger/call builds on other projects

This screenshot shows the 'Build Environment' and 'Build' tabs for the same Jenkins job. The 'Build Environment' tab is active. Under 'Build Environment', there are two checkboxes: 'Use secret text(s) or file(s)' and 'With Ant'. The 'With Ant' checkbox is checked. The 'Build' tab is also visible below, showing a list of build steps. The 'Invoke Ant' step is highlighted with a blue selection bar and a mouse cursor icon over it. Other steps listed include 'Execute Windows batch command', 'Execute shell', 'Invoke top-level Maven targets', 'Set build status to "pending" on GitHub commit', and 'Trigger/call builds on other projects'.

Invoking ANT

Dashboard > FourthJob >

General Source Code Management Build Triggers **Build Environment** Build Post-build Actions

With Ant ?

Build

☰ Invoke Ant ?

Targets ?

war

Advanced...

Dashboard > FourthJob >

General Source Code Management Build

Build Environment

Post-build Actions

☰ Deploy war/ear to a container

WAR/EAR files ?

**/*.war

Context path ?

samplewebapp

Containers

Add Container ▾

Deploy on failure

Add post-build action ▾

(1) Targets – That the end product will be in The war format.

(2) Deploying the war file into the container

(3) The end product will be displayed in samplewebapp.war file

Dashboard > FourthJob >

General Source Code Management Build

Build Environment

Post-build Actions

☰ Invoke Ant ?

Filter

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Publish JUnit test result report
- Record fingerprints of files to track usage
- Git Publisher
- Build other projects (manual step)
- Deploy war/ear to a container**
- E-mail Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Trigger parameterized build on other projects

Add post-build action ▾

General Source Code Management Build Triggers Build Environment Build Post-build Actions

— Deploy war/car to a container

WAR/EAR files ?

**/*.war

Context path ?

samplewebapp

Containers

Add Container ▾

Filter

GlassFish 2.x

GlassFish 3.x

GlassFish 4.x

JBoss AS 3.x

JBoss AS 4.x

JBoss AS 5.x

JBoss AS 6.x

JBoss AS 7.x

Tomcat 4.x Remote

Tomcat 5.x Remote

Tomcat 6.x Remote

→ Selecting the Tomcat version in container.

Credentials

- none -

+ Add



Jenkins

TOUCHONE

Advanced...

→ Giving the credentials of Tomcat after this page.

Username ?
admin

Treat username as secret ?

Password ?

ID ?

Description ?

Add Cancel

Tomcat Credentials

Credentials

- none -
- none -
- admin/*****

After giving the tomcat credentials,
next, selection of that credentials.

Tomcat URL ?

Advanced...

Add Container ▾

Save **Apply**

Tomcat URL after changing of its port from 8080 to 8090

 Jenkins

Dashboard >

+ New Item **All** +

People Build History Manage Jenkins My Views New View

Build Queue No builds in the queue.

Build Executor Status 1 Idle 2 Idle

Icon: S M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

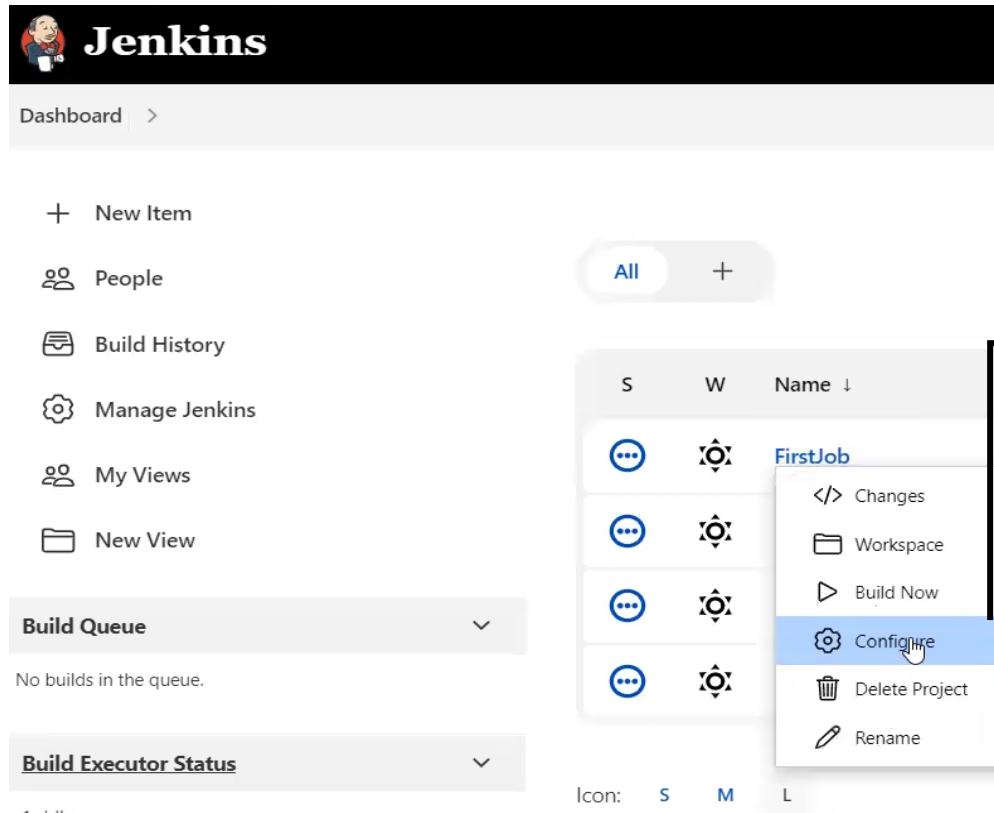
S	W	Name ↓	Last Success	Last Failure	Last Duration
...	...	FirstJob	N/A	N/A	N/A
...	...	FourthJob	N/A	N/A	N/A
...	...	SecondJob	N/A	N/A	N/A
...	...	ThirdJob	N/A	N/A	N/A

→ Four Jobs before build.

- We need to automate all the 4- jobs via using the Downstream (or) Upstream, So we use the **Downstream**, because it is easy to connect the jobs.

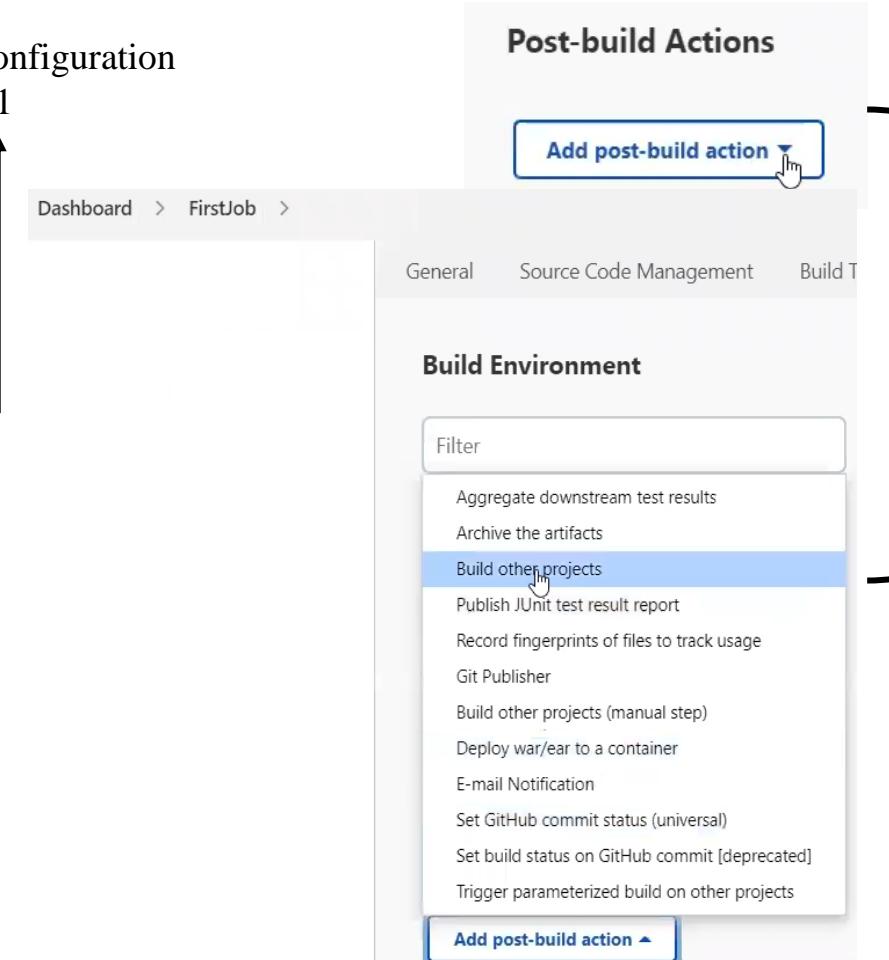
• PROCESS JOB-1 :-

Click on **job-1** —————> Click on **Configure** —————> Scroll down **Post build actions** —————> Select “ **Build Other Projects & give job-2 name & Select “ Trigger only if build is stable** ”
 Click on **Apply & Save**.



The Jenkins dashboard shows the 'FirstJob' project in the list of available items. A context menu is open over the 'FirstJob' entry, with the 'Configure' option highlighted.

(1) Configuration of job-1



The 'Post-build Actions' configuration screen for 'FirstJob'. The 'Build Environment' section is visible, showing various actions. The 'Build other projects' action is selected and highlighted with a blue background. The 'Add post-build action' button is located at the bottom left of the list.

(2) Selection of Build other projects In the Post Build Actions section.

Add build step ▾**Post-build Actions****Build other projects ?**

Projects to build

SecondJob

 Trigger only if build is stable Trigger even if the build is unstable Trigger even if the build fails**Add post-build action ▾**

(3) Mention/Selection of Second job name.

• PROCESS JOB-2 :-

- Click on **job-2** → Click on **Configure** → Scroll down **Post build actions** →
Select “ **Build Other Projects** & give **job-3 name** & Select “ **Trigger only if build is stable**”
Click on **Apply & Save**.

(1) Configuration of job-2

The Jenkins dashboard shows a list of jobs: FirstJob, FourthJob, SecondJob, and ThirdJob. A context menu is open over the SecondJob entry, with the 'Configure' option highlighted.

(2) Selection of Build other projects In the Post Build Actions section.

In the 'Post-build Actions' configuration for the SecondJob, the 'Build other projects' action is selected. The 'Projects to build' field contains 'ThirdJob'. The 'Trigger only if build is stable' radio button is selected.

(3) Mention/Selection of Third job name.

The 'Projects to build' field in the 'Build other projects' configuration contains the value 'ThirdJob'.

- **PROCESS JOB-3 :-**

Click on job-3 → Click on **Configure** → Scroll down **Post build actions** → Select “Build Other Projects & give job-4 name & Select “Trigger only if build is stable”
Click on **Apply & Save.**

Jenkins

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All +

S	W	Name ↓
...	...	FirstJob
...	...	FourthJob
...	...	SecondJob
...	...	ThirdJob

Icon: S M

- </> Changes
- Workspace
- ▷ Build Now
- Configure** (highlighted)
- >Delete Project
- Rename

(1) Configuration of job-3

Dashboard > ThirdJob >

Post-build Actions

Add post-build action

General Source Code Management Build

Do not fail the build on empty test results

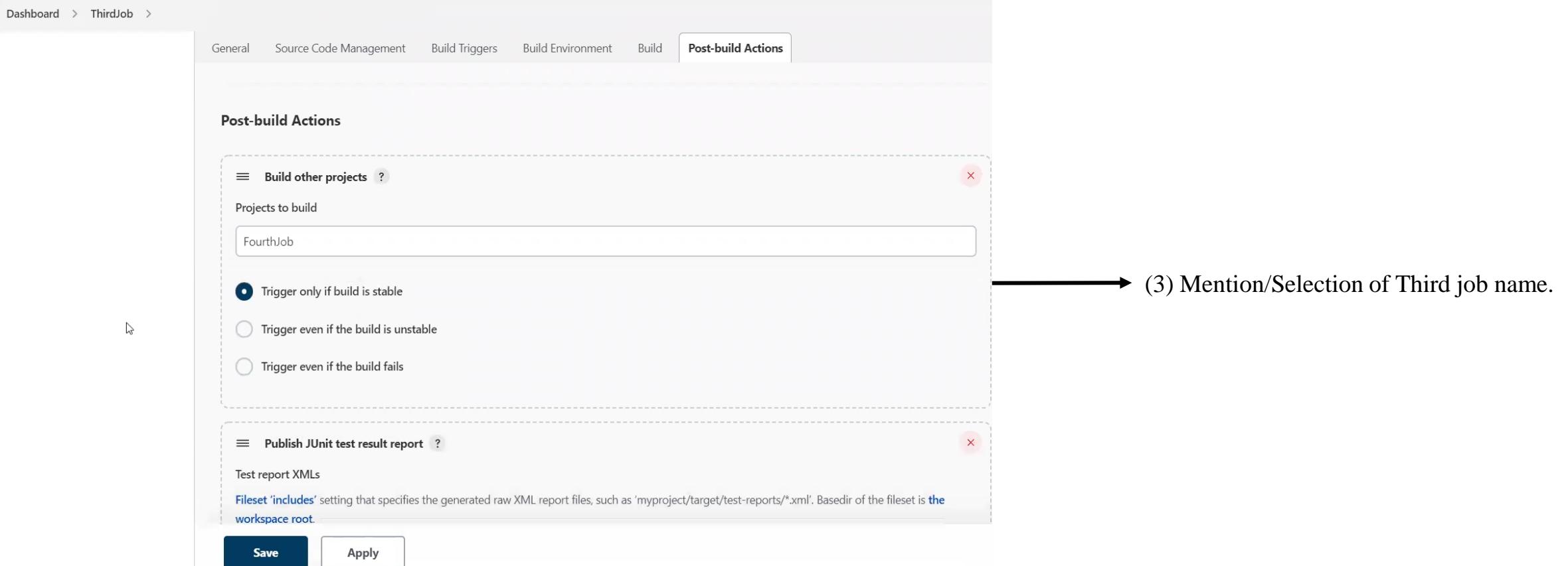
Skip publishing checks ?

Filter

- Aggregate downstream test results
- Archive the artifacts
- Build other projects** (selected)
- Publish JUnit test result report
- Record fingerprints of files to track usage
- Git Publisher
- Build other projects (manual step)
- Deploy war/ear to a container
- E-mail Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Trigger parameterized build on other projects

Add post-build action ▲

(2) Selection of Build other projects In the Post Build Actions section.



BUILDING A JENKINS PIPELINE:-

- In Jenkins dashboard, Click on +, to add a view → Give any required name & Select “ **Build Pipeline view** ”
→ & Click on Create → Now Scroll down in **Up Stream/Down Stream** config → In **Select Initial Job**
→ Select “ **First Job** ” → & Click on apply & save.

Jenkins

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All

Add description

S	W	Name	Last Success	Last Failure	Last Duration
		FirstJob	N/A	N/A	N/A
		FourthJob	N/A	N/A	N/A
		SecondJob	N/A	N/A	N/A
		ThirdJob	N/A	N/A	N/A

Icon: S M L

Icon legend Atom feed for all builds Atom feed for failures Atom feed for just latest builds

New view

Name

DevOpsCI-CDPipeline

Type

Build Pipeline View

Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

List View

Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

My View

This view automatically displays all the jobs that the current user has an access to.

Configure

Search

?

2

admin

log out

Dashboard > DevOpsCI-CDPipeline >

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream config

Select Initial Job ?

FirstJob

Trigger Options

Build Cards

Standard build card

Use the default build cards

Restrict triggers to most recent successful builds ?

Yes

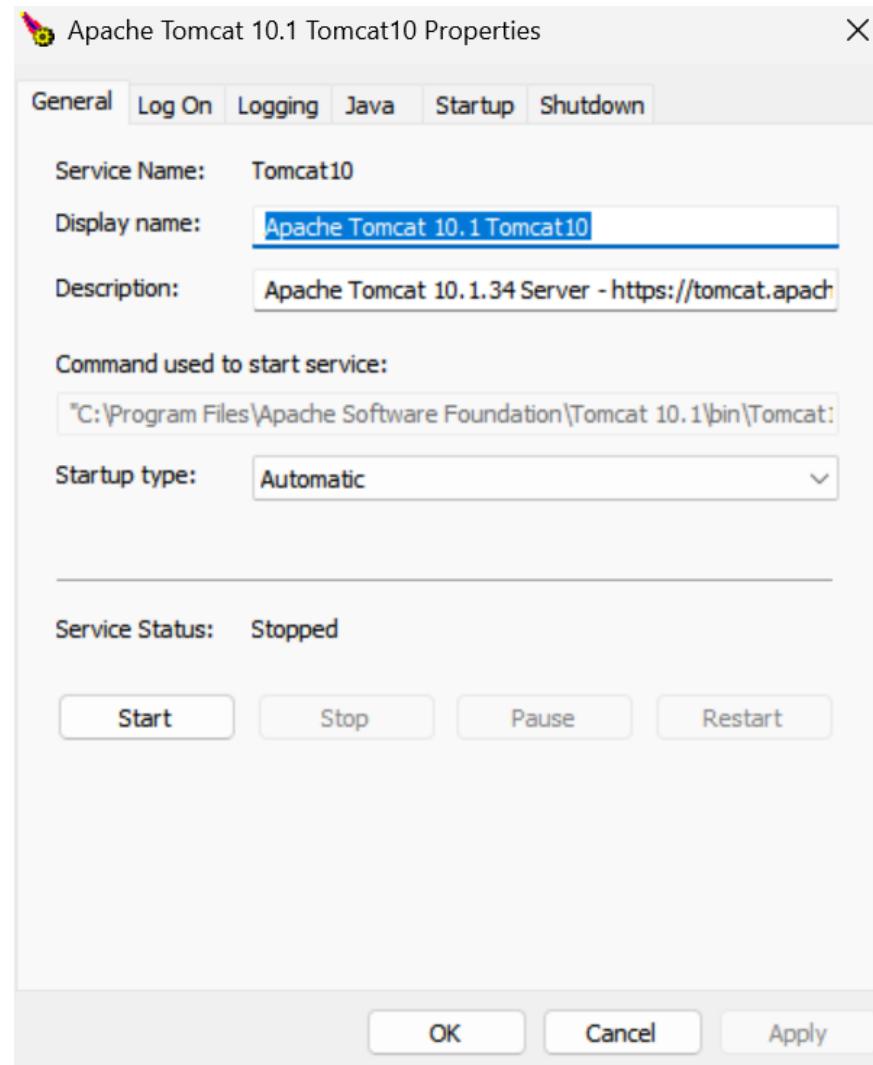
OK Apply

(1) Creation of pipeline view

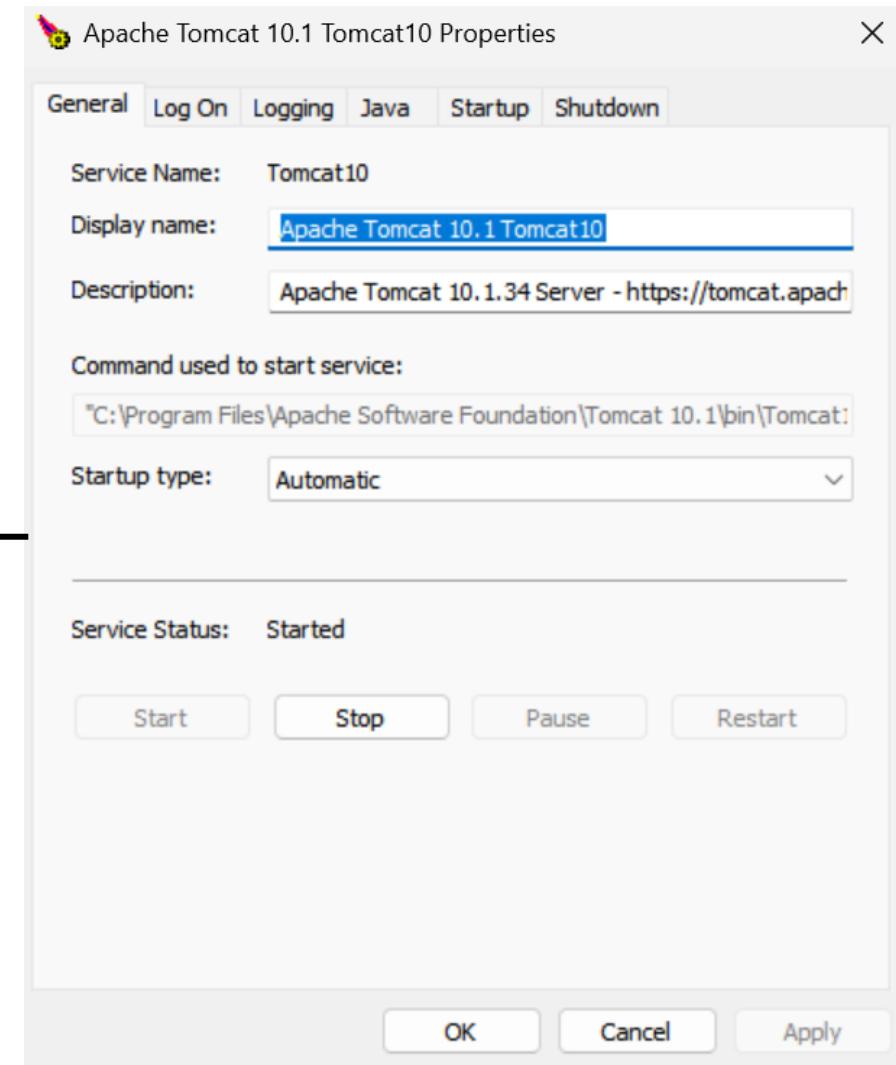
(2) Selection of Build pipeline view.
(This option gets when we installed the build pipeline plugin)

(3) Selection of Initial job in Downstream.

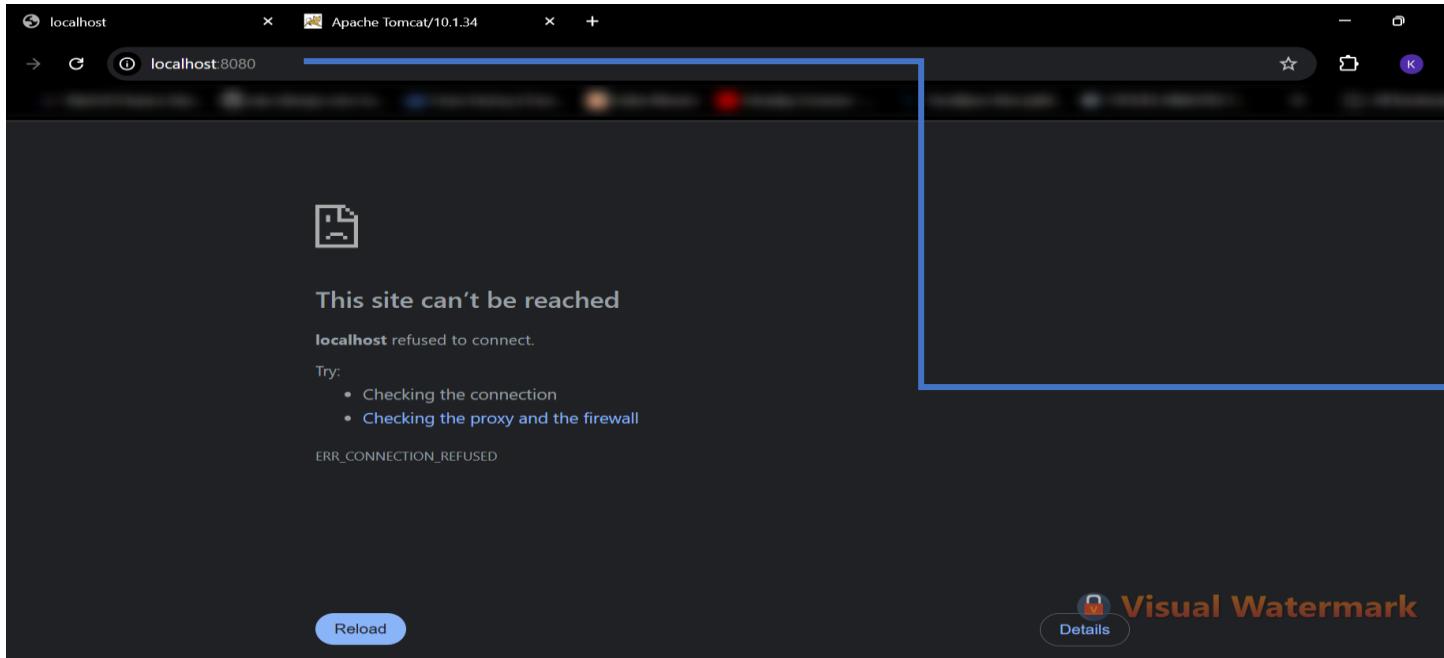
- Now separately we can see the Jenkins build pipeline view.
- Restart of Tomcat**, in Windows **search for tomcat** & double click on it → In General, Click on **Stop** & again Click on **Start**.
- From now onwards the Tomcat runs on **port 8090**.



Before



After



Now tomcat is not been accessed by the port 8080

A screenshot of a web browser window titled "localhost" and "Apache Tomcat/10.1.34". The address bar shows "localhost:8090". The main content area displays the Apache Tomcat 10.1.34 homepage. The page features a banner stating "If you're seeing this, you've successfully installed Tomcat. Congratulations!" with a cartoon cat icon. It includes sections for "Developer Quick Start", "Managing Tomcat", "Documentation", and "Getting Help". The "Developer Quick Start" section has links to "Tomcat Setup", "First Web Application", "Realms & AAA", "JDBC DataSources", "Examples", and "Servlet Specifications". The "Documentation" section has links to "Tomcat 10.1 Documentation", "Tomcat 10.1 Configuration", "Tomcat Wiki", and "Tomcat 10.1 JavaDocs". The "Getting Help" section has links to "FAQ and Mailing Lists" and "tomcat-announce", "tomcat-users", "tomcat-dev", and "tomcat-javadoc". A red "Visual Watermark" logo is visible at the bottom right of the page.

From now onwards the Tomcat will be accessed
by the port 8090

- To build the jobs periodically we use **POLLSCM**.

click on **First job** ————— Click on **Configure** ————— In **Build Triggers** select “**poll scm**” & give *** * * * ***———— Click on apply & save.

- Now all jobs get executed & build one by one automatically & the application gets created in the tomcat server.

The image shows two screenshots of the Jenkins interface. The left screenshot is the Jenkins dashboard, displaying a list of items including 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'New View'. A 'Build Queue' section indicates 'No builds in the queue.' The right screenshot shows the configuration of a specific job named 'FirstJob'. The 'Build Triggers' tab is selected, showing various trigger options: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling', and 'Poll SCM'. The 'Poll SCM' option is checked, and the schedule field contains the value '* * * * *'. Below the configuration are 'Save' and 'Apply' buttons. Arrows point from the 'Configure' option in the job list to the configuration screen, and another arrow points from the 'Poll SCM' setting to the text 'POLLSCM - * * * * *'. The overall title of the slide is 'Jenkins CI Pipeline'.

Dashboard >

+ New Item

People

All DevOpsCI-CDPipeline +

S W Name ↓

FirstJob

- </> Changes
- Workspace
- Build Now
- Configure**
- Delete Project
- Rename

General Source Code Management **Build Triggers** Build Environment Build Post-build Actions

Build Triggers

Trigger builds remotely (e.g., from scripts) ?

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Schedule ?

*****I

No schedules so will only run due to SCM changes if triggered by a post-commit hook

Ignore post-commit hooks ?

Save Apply

Configuration of first job.

POLLSCM - * * * * *

- After giving the POLLSCM, it takes some 1-2 mins of time to execute the builds of all the 4 jobs.

Tomcat Web Application Manager

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy
					Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy
					Expire sessions with idle ≥ 30 minutes
/jenkins	None specified	Jenkins v2.346.2	true	0	Start Stop Reload Undeploy
					Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy
					Expire sessions with idle ≥ 30 minutes

Dashboard > DevOpsCI-CDPipeline >

Build Pipeline

→ The build pipeline view before the build started.



This view has no jobs associated with it. You can either add some existing jobs to this view or create a new job in this view.

Tomcat server before the build

Build Pipeline

Trigger a Pipeline Pipeline History Configure Add Step Delete Manage

Pipeline	#1 FirstJob	#1 SecondJob	#1 ThirdJob	#1 FourthJob
#1	19-Dec-2024 4:15:02 PM 12 sec	19-Dec-2024 4:15:22 PM 7.5 sec	19-Dec-2024 4:15:36 PM 6.2 sec	19-Dec-2024 4:15:53 PM 10 sec

Tomcat Web Application Manager

Message: OK - Started application at context path [/jenkins]

Manager

[List Applications](#)[HTML Manager Help](#)[Manager Help](#)[Server Status](#)

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/jenkins	None specified	Jenkins v2.346.2	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/samplewebapp	None specified	Sample Web Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

After the successful build of all 4 jobs

While Jobs are in executing phase we will see different colours

- Yellow – Job is in executing phase.
- Green – Job executed.
- Blue – Job is going to be executed in next.

Tomcat server after the build Completed.

- To verify the Tomcat folder in your local host, go to **C:\Program Files\Apache Software Foundation\Tomcat 10.1\webapps**.
 - To access the samplewebapp folder, open it in the new tab of any browser from the tomcat server. It will look like this.
-

Game of Thrones



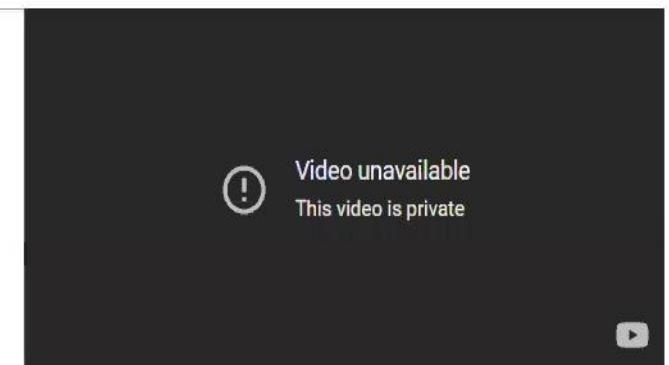
Game of Thrones is an Indian fantasy drama television series created by David Benioff and D. B. Weiss. It is an adaptation of a Song of Ice and Fire, George R. R. Martin's series of fantasy novels, the first of which is 'A Game of Thrones'!!!

General Plot:

A story of greed, lust, thirst for power, ruthlessness, barbarity - GOT is a story of seven kingdoms vying and sparring over absolute power and control to the Iron Throne.

Warring kingdoms:

- Kingdom of the North!
- Kingdom of the Mountain and the Vale!
- Kingdom of the Isles and Rivers
- Kingdom of the Rock
- Kingdom of the Stormlands
- Kingdom of the Reach
- Principality of Dorne



- If you want to do any changes in code or any file then open

Samplewebapp folder ————— **Webcontent** ————— **Open Index file with any software** ————— **Do any required changes & save it.**

- After doing the changes, Now we need to **Add, Commit, Push** to the Github. So that the newly updated changes makes the job triggers in the Jenkins.

```
C:\Users\KANNA\SampleWebApp>git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   WebContent/index.html

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\KANNA\SampleWebApp>git add .

C:\Users\KANNA\SampleWebApp>git commit -m "modified"
[master 225217e] modified
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\KANNA\SampleWebApp>git push origin master
info: please complete authentication in your browser...
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 349 bytes | 349.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/hruthikkanna/SampleWebApp.git
  cce806f..225217e  master -> master
```

→ Pushing the modified updates to the GitHub repo.

- So now it automatically triggers the jobs & creates the new end war file. So that new modifications will be visible to us.

The screenshot shows the Jenkins Build Pipeline interface. At the top, there's a navigation bar with a Jenkins logo, search bar, and user info (admin). Below it, the main title is "Build Pipeline". There are buttons for "Trigger a Pipeline", "Pipeline History", "Configure", "Add Step", "Delete", and "Manage". The pipeline itself consists of four green rectangular boxes representing jobs. Each job box has a small grey header with "#1" and a larger green body. The first job is labeled "#1 FirstJob" with a timestamp of "19-Dec-2024 4:15:02 PM" and a duration of "12 sec". The second job is "#1 SecondJob" with a timestamp of "19-Dec-2024 4:15:22 PM" and a duration of "7.5 sec". The third job is "#1 ThirdJob" with a timestamp of "19-Dec-2024 4:15:36 PM" and a duration of "6.2 sec". The fourth job is "#1 FourthJob" with a timestamp of "19-Dec-2024 4:15:53 PM" and a duration of "10 sec". Each job box contains a "console" button and a "re-run" button.

The new build gets created & the time will be updated

- If you want to give any E-mail notifications regarding the console output of any job, the process is ...

Click on the required job —————> **Click on Configure** —————> **At last in Post Build Actions, Select Email Notifications & give Required E-mail** —————> **Click on apply & save.**

... PTO

All DevOpsCI-CDPipeline +

S	W	Name ↓
		FirstJob
		FourthJob
		SecondJob

Icon: S M

Configuration

Post-build Actions

Filter

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Publish JUnit test result report
- Record fingerprints of files to track usage
- Git Publisher
- Build other projects (manual step)
- Deploy war/ear to a container
- E-mail Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Trigger parameterized build on other projects

Add post-build action ▾

Save Apply

Give required email ID

Game of Thrones



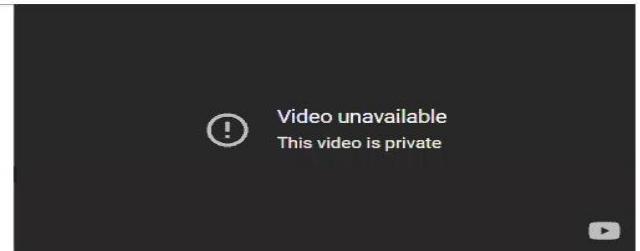
Game of Thrones is an Indian fantasy drama television series created by David Benioff and D. B. Weiss. It is an adaptation of a Song of Ice and Fire, George R. R. Martin's series of fantasy novels, the first of which is 'A Game of Thrones!!!'

General Plot:

A story of greed, lust, thirst for power, ruthlessness, barbarity - GOT is a story of seven kingdoms vying and sparring over absolute power and control to the Iron Throne.

Warring kingdoms:

- Kingdom of the North!
- Kingdom of the Mountain and the Vale!
- Kingdom of the Isles and Rivers
- Kingdom of the Rock
- Kingdom of the Stormlands
- Kingdom of the Reach
- Principality of Dorne



- This is the completion of the **DevOps CI-CD Pipeline Project**.
- If any queries feel free to contact me via

LINKEDLN - <https://www.linkedin.com/in/kanna-hruthik-a06a9b2aa/>

THANK YOU

GITHUB - <https://github.com/hruthikkanna/DevOps-CI-CD-PipeLine-Project>

KANNA HRUTHIK