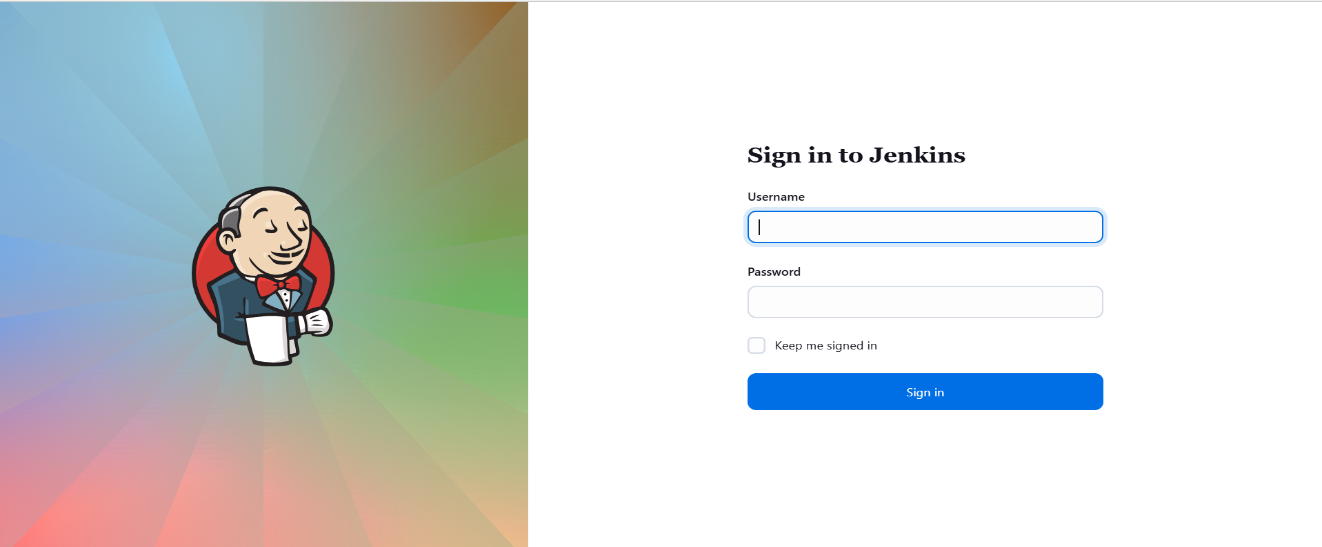
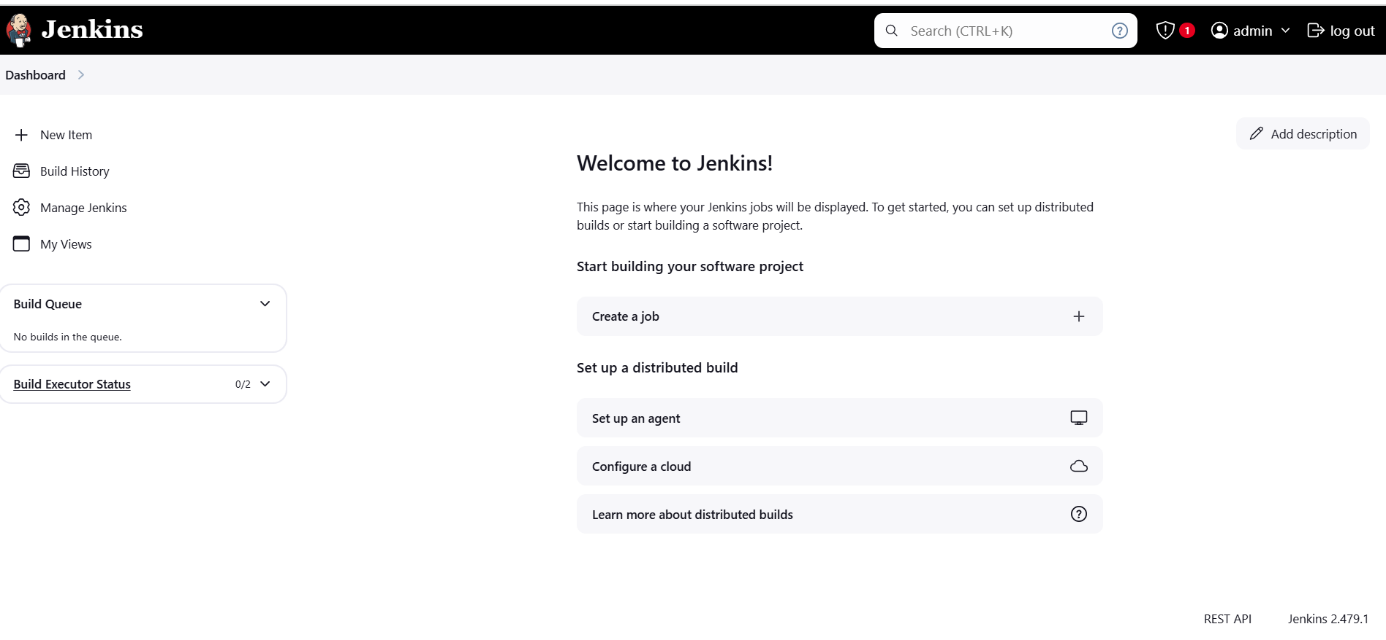
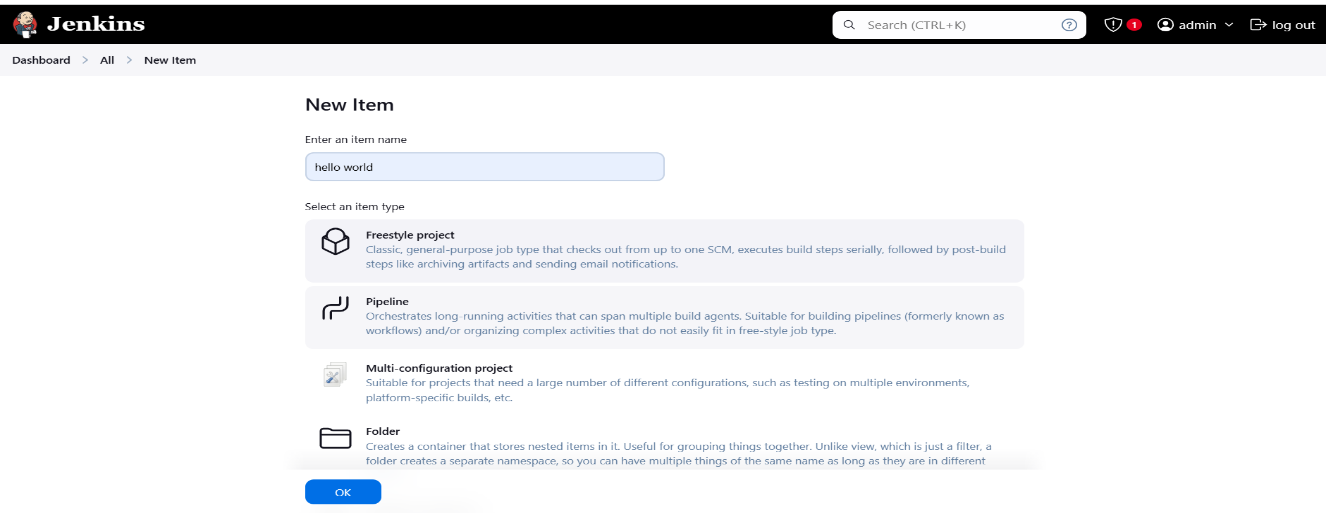
**EXPERIMENT-5:**

5.Demonstrate continuous integration and development using jenkins

* Login to aws
* Start the instance in which jenkins is installed
* In new tab type jenkins instance IP address:8080
* Login to jenkins



**BUILD PERIODICALLY**

* ****Add New item
* Configure the project
* In source code management select Git and give github repository url
* In build triggers select build periodically
* Schedule format

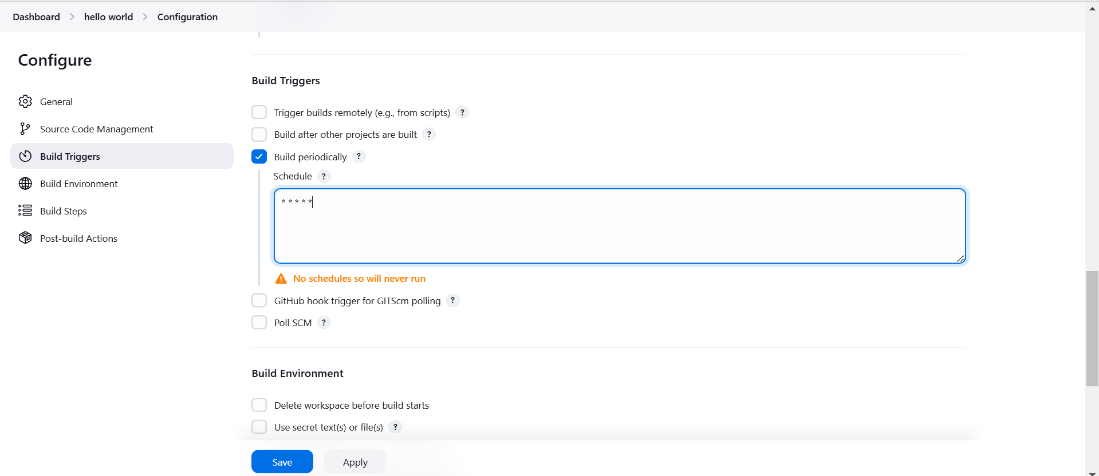
MINUTE HOUR DOM MONTH DOW

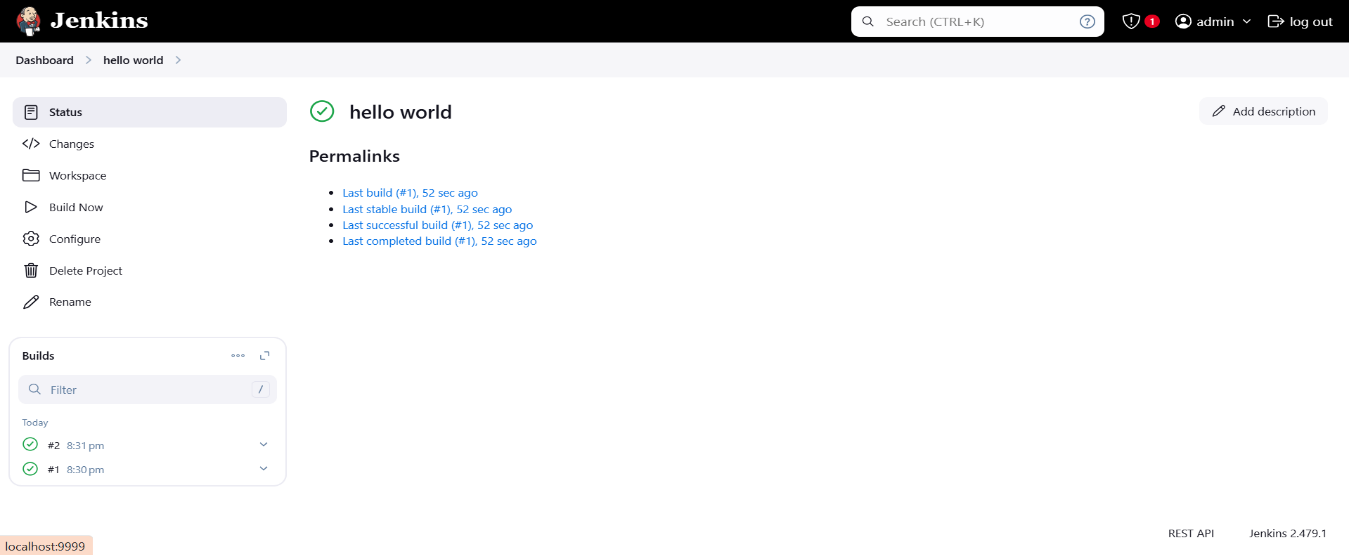
\* MINUTE:0 TO 59

\*HOUR:0 TO 23

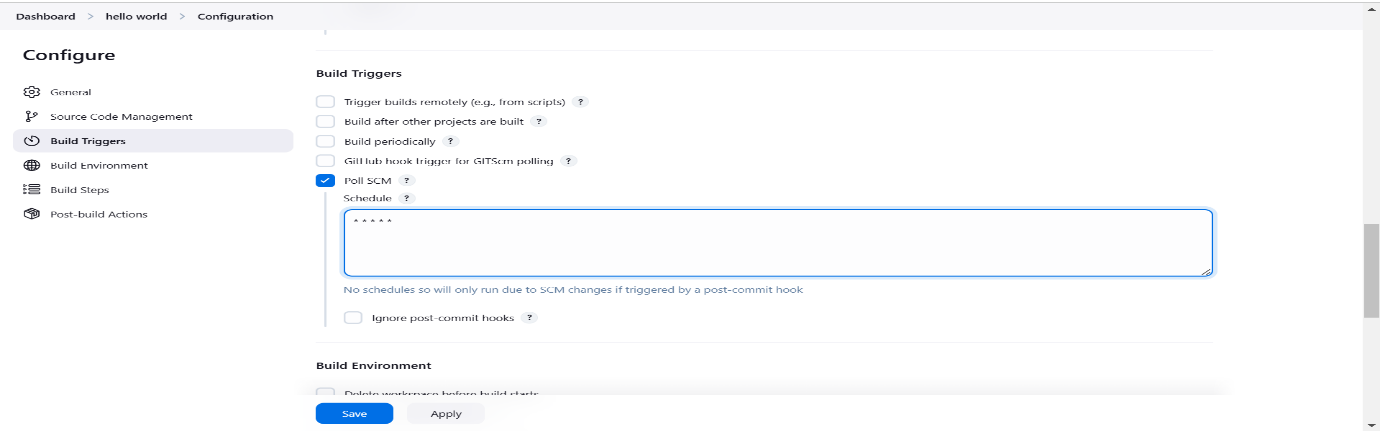
\*DOM(DAY OF MONTH):1 TO 31

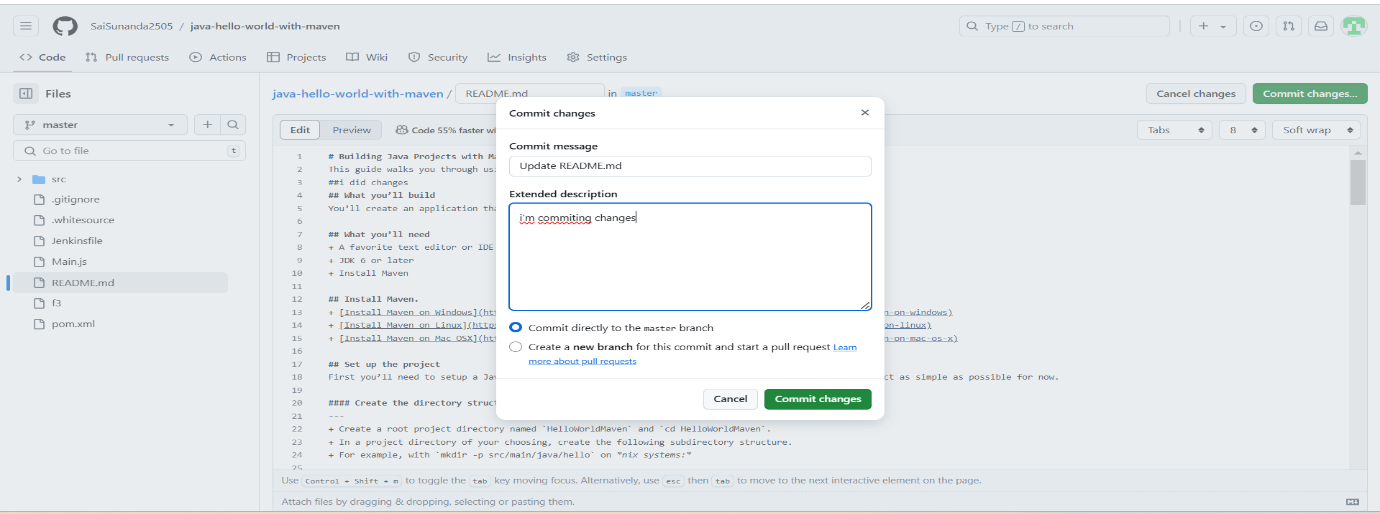
\*MONTH:1 TO 12

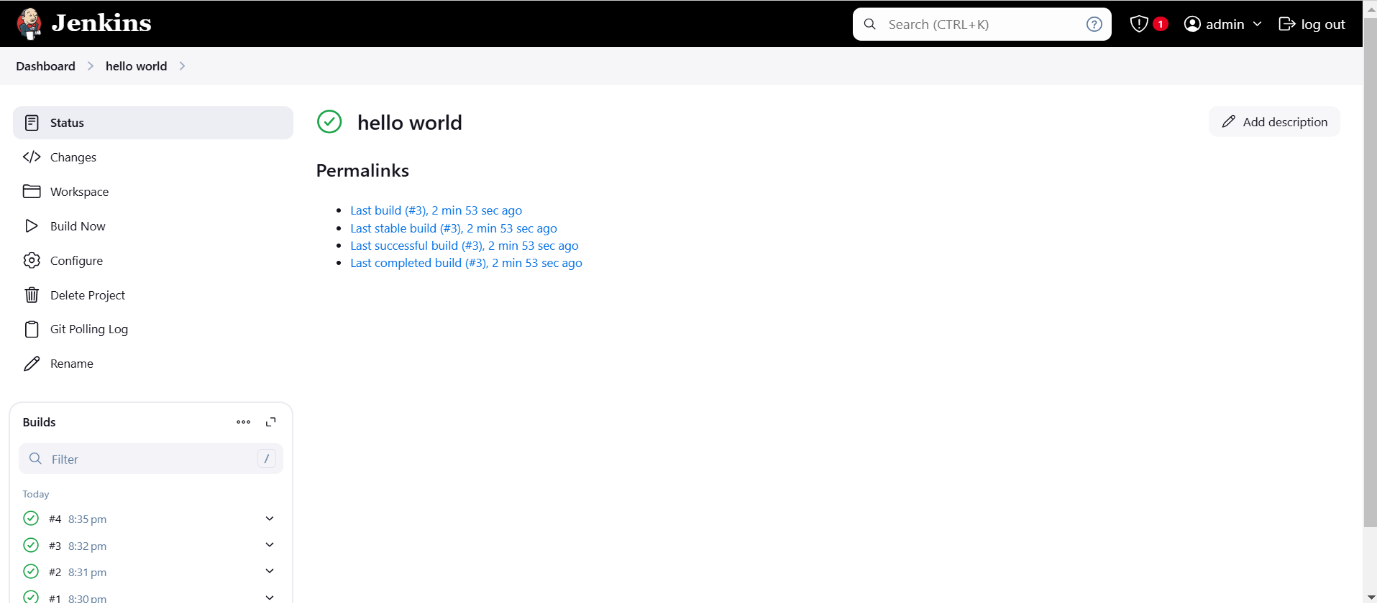
\*DOW(DAY OF WEEK):0 TO 7

* In build steps add build step as invoke top level maven target and goal as compile/package
* Build starts as per the schedule

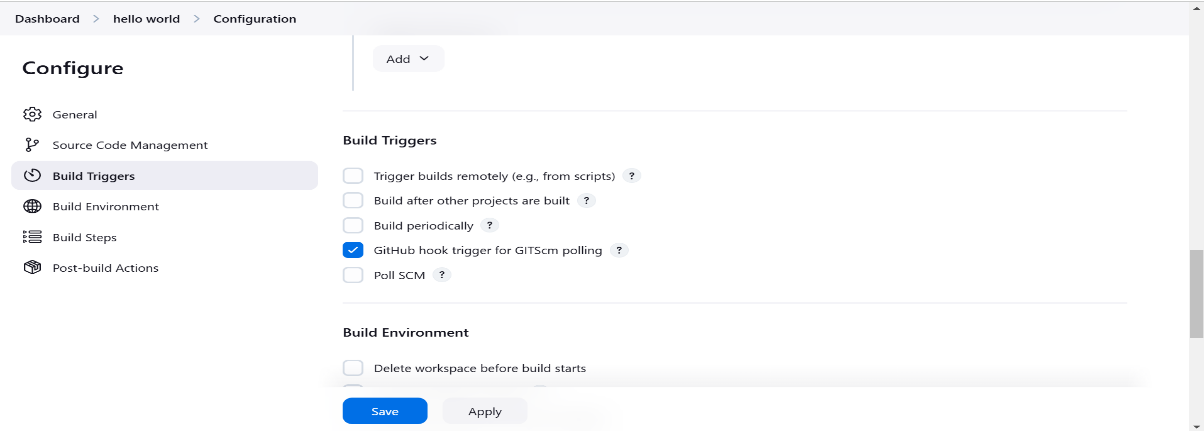
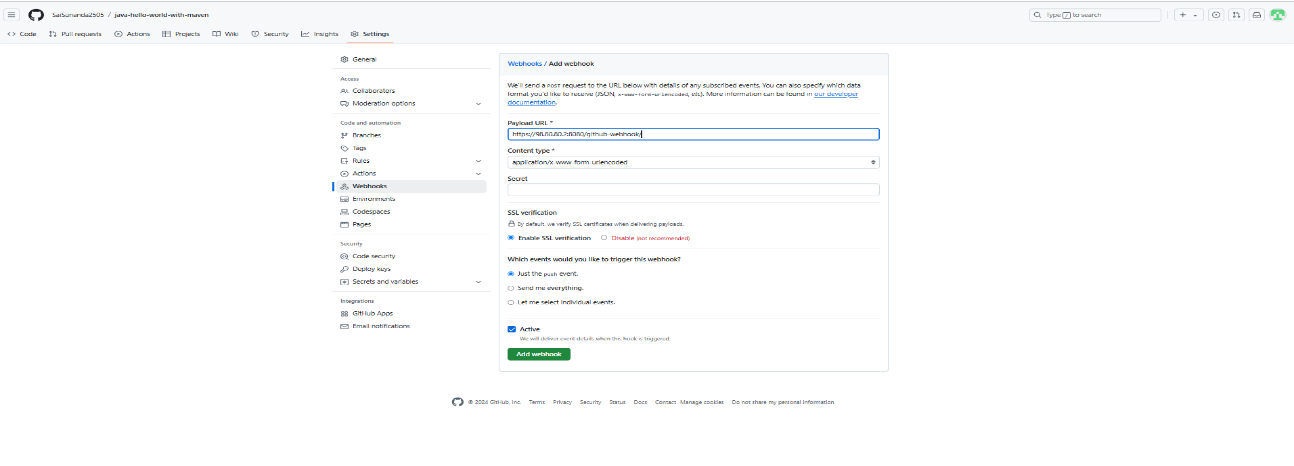
**POLL SCM :**

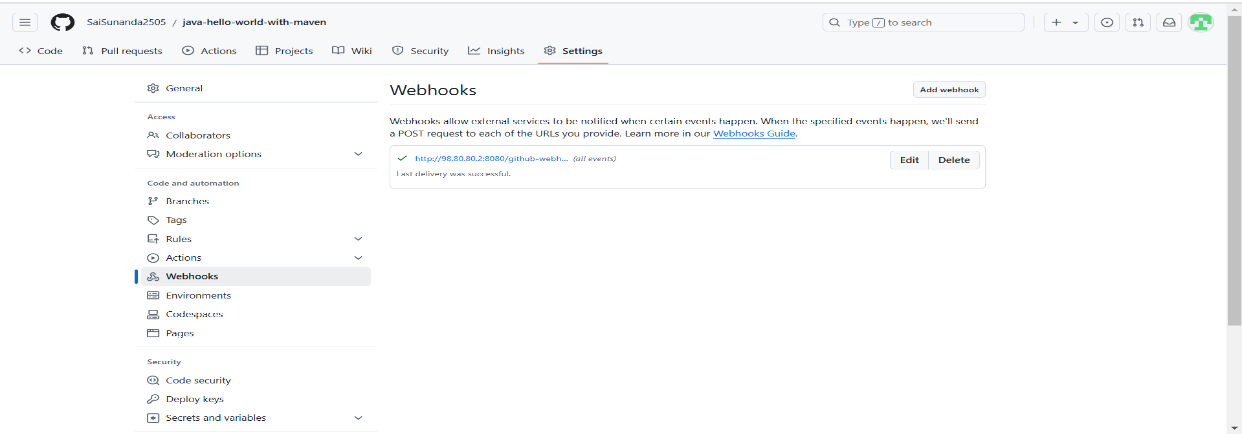
* In configuration select poll scm under build triggers
* Whenever there is a commit in github repository a new build is intiated

****

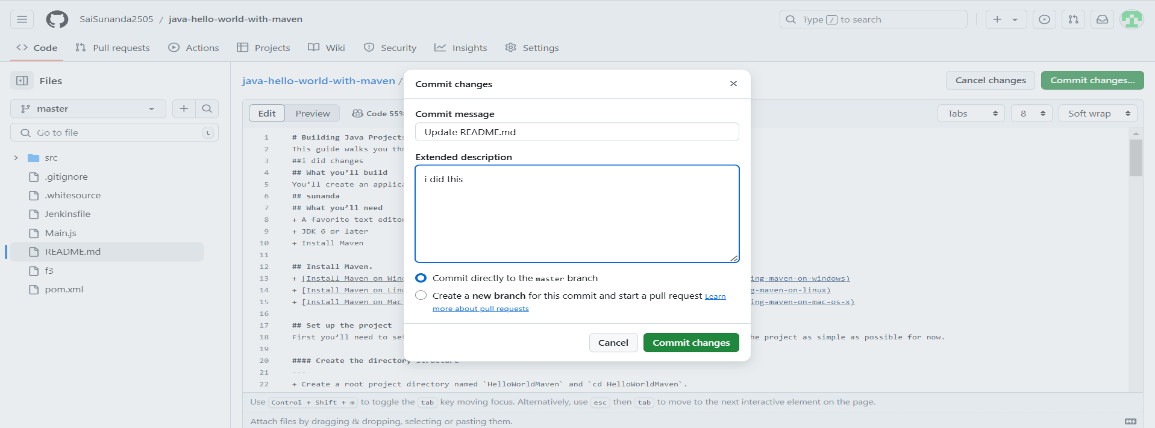
* We can observe a build in jenkins

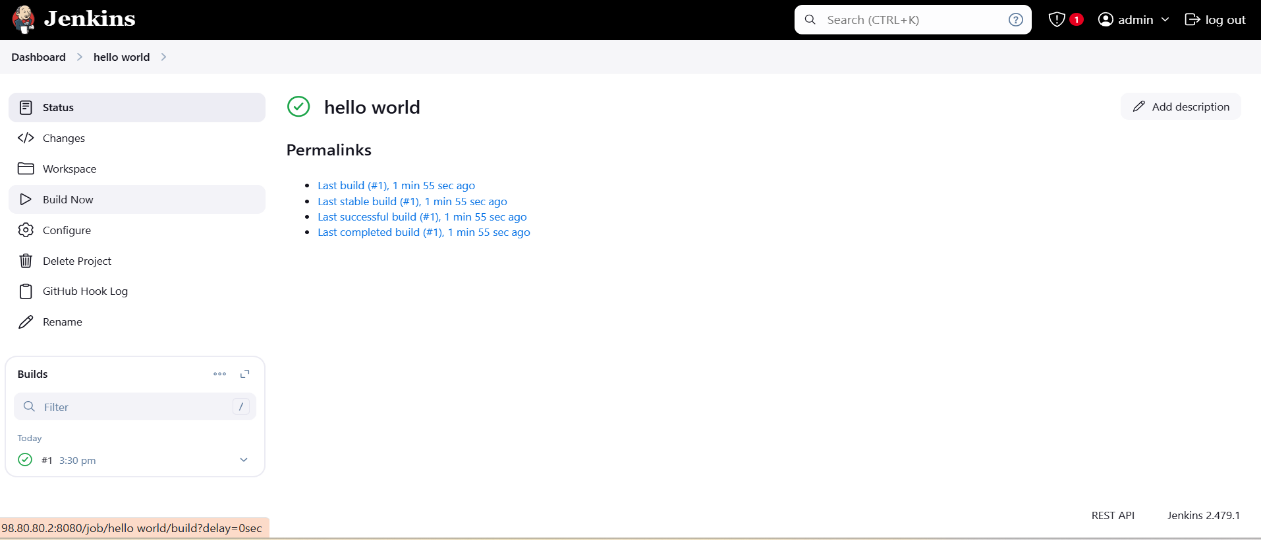
**WEBHOOK :**

* In configuration select GitHub hook trigger for GITScm polling under build trigger
* In github go to repository settings then webhook and add webhook
* In payload url give jenkins instance privae IP:8080
* Click on add webhook

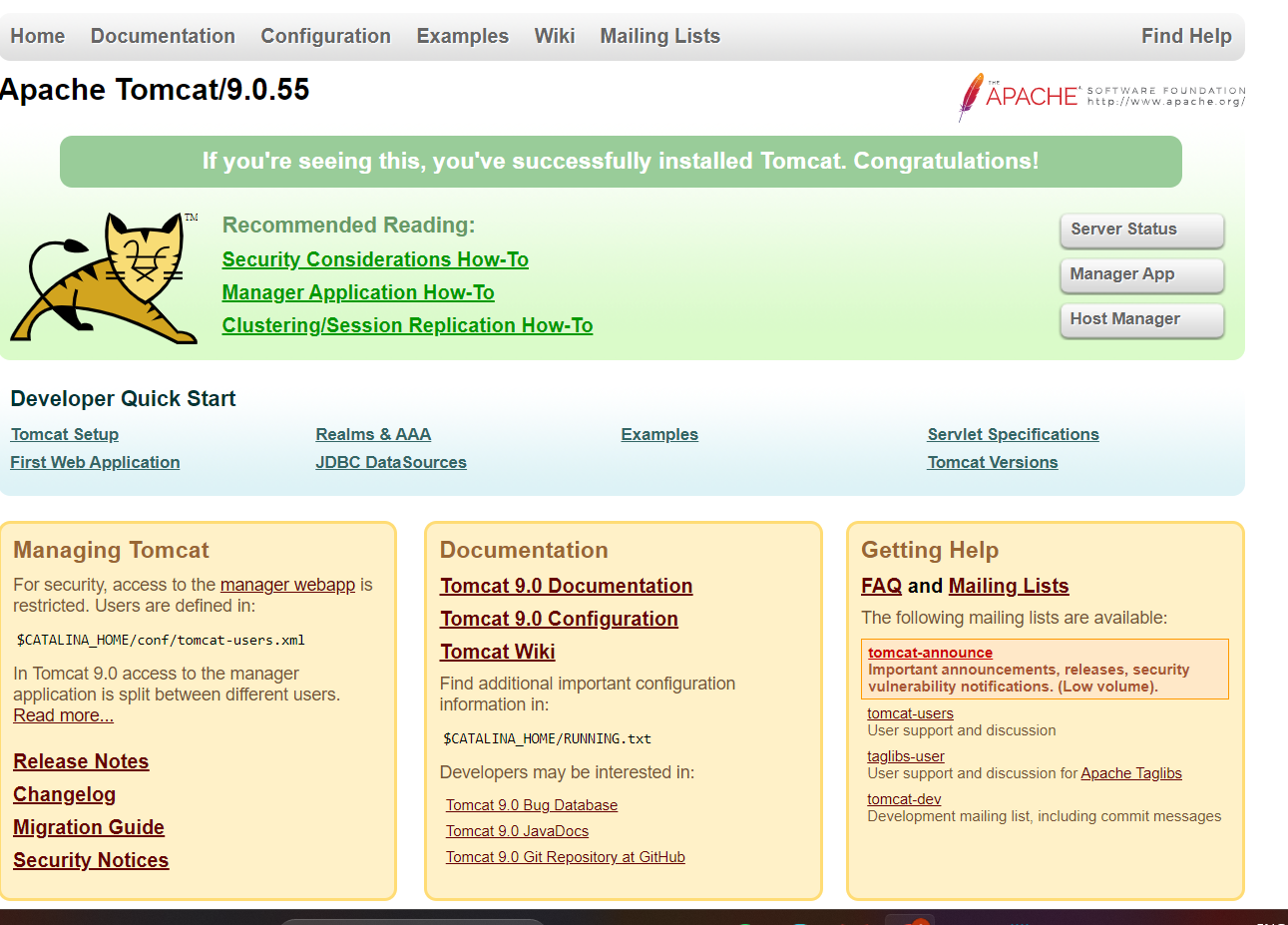
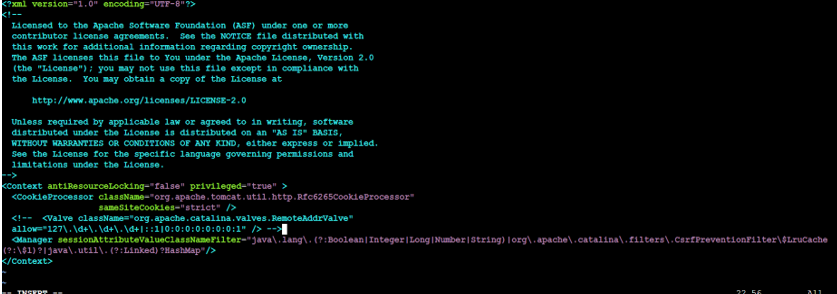
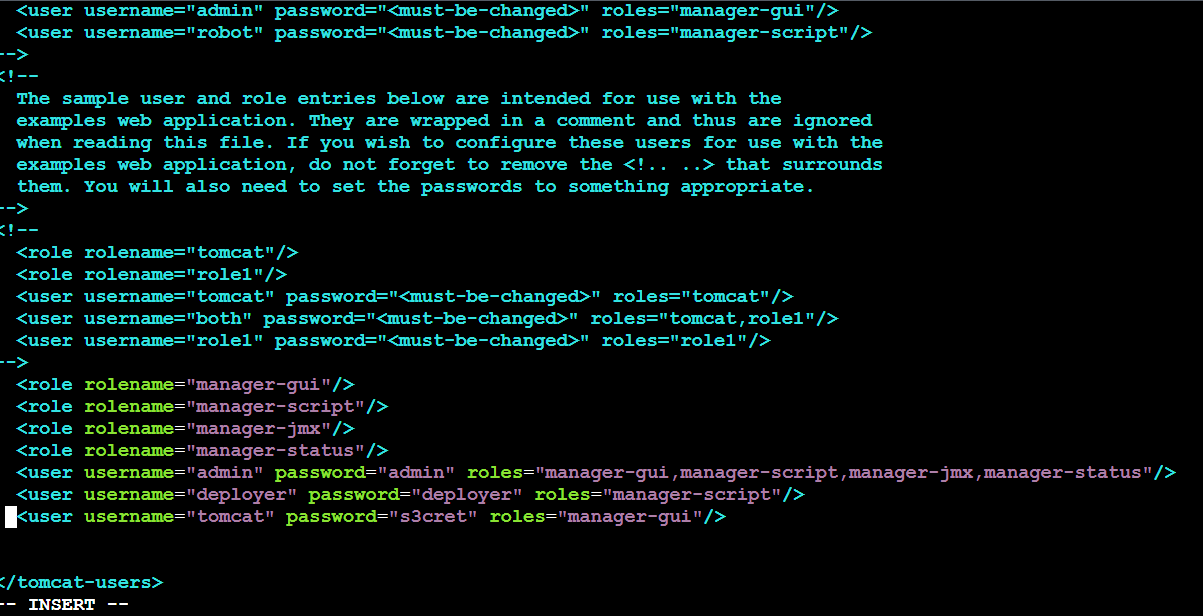
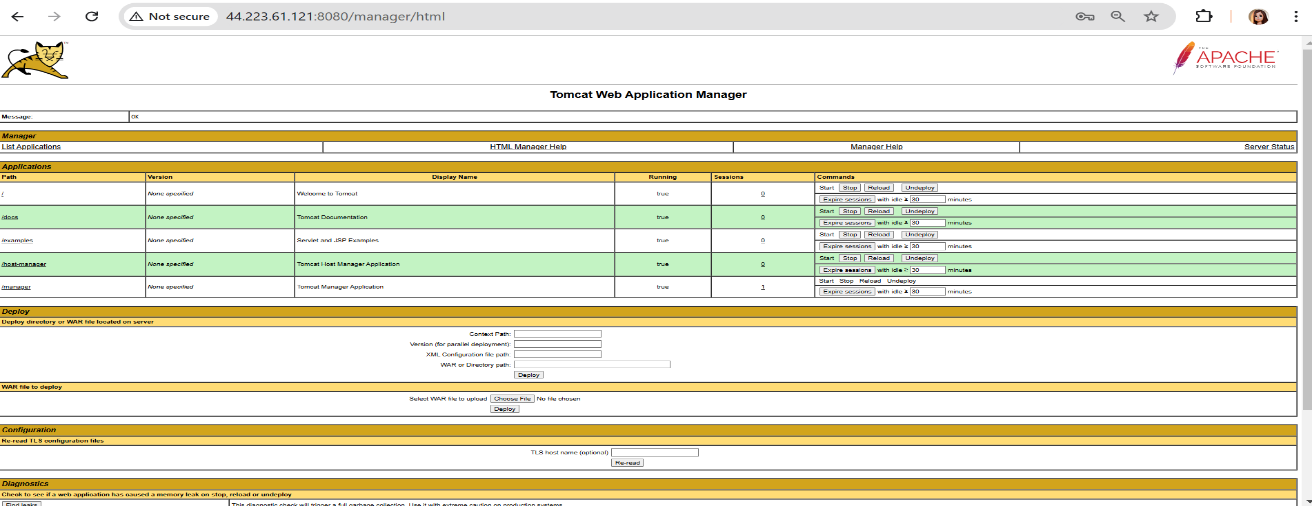


* Whenever there is a commit in github repository a new build is intiated

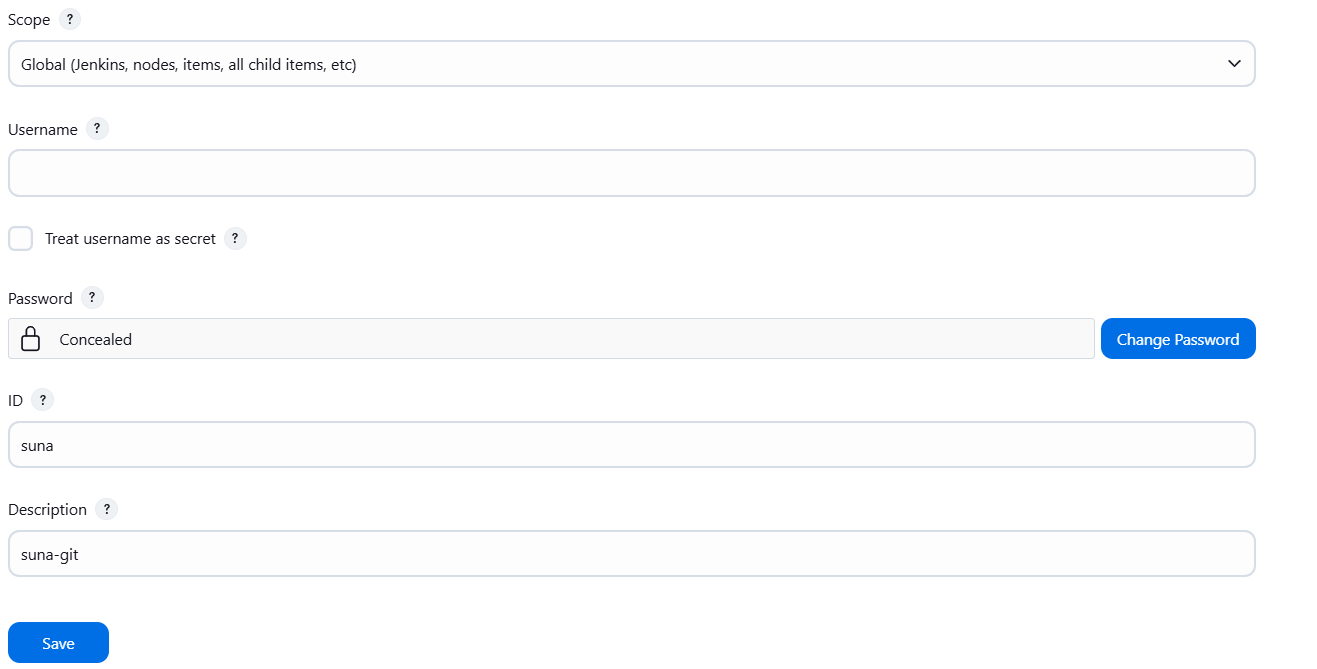


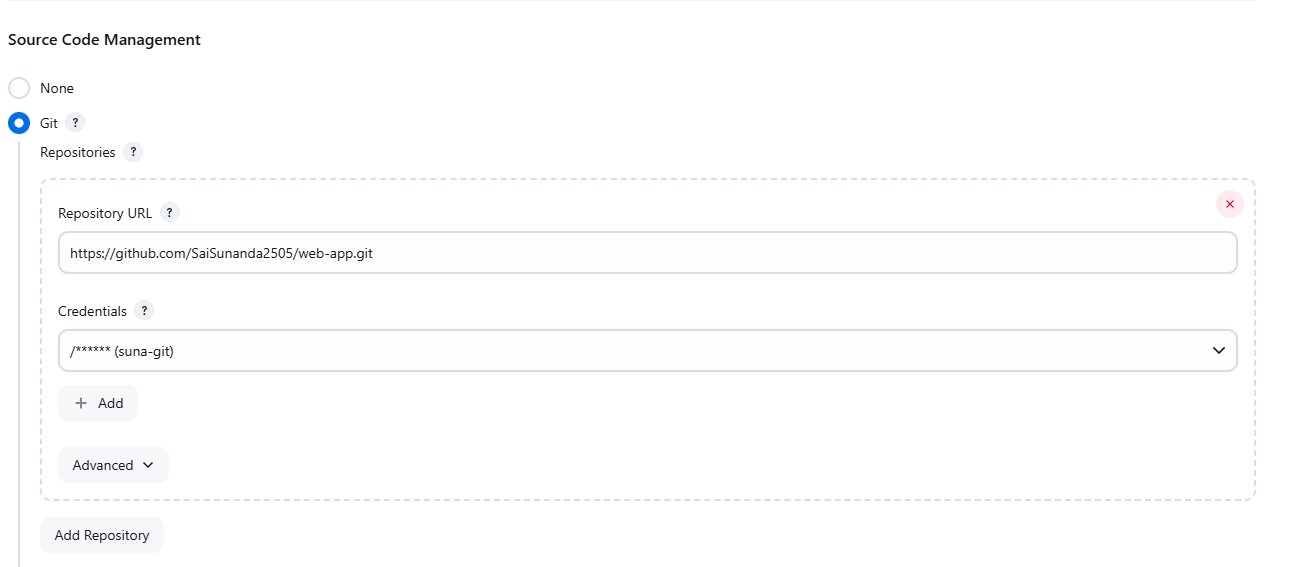
* We can observe a build in jenkins

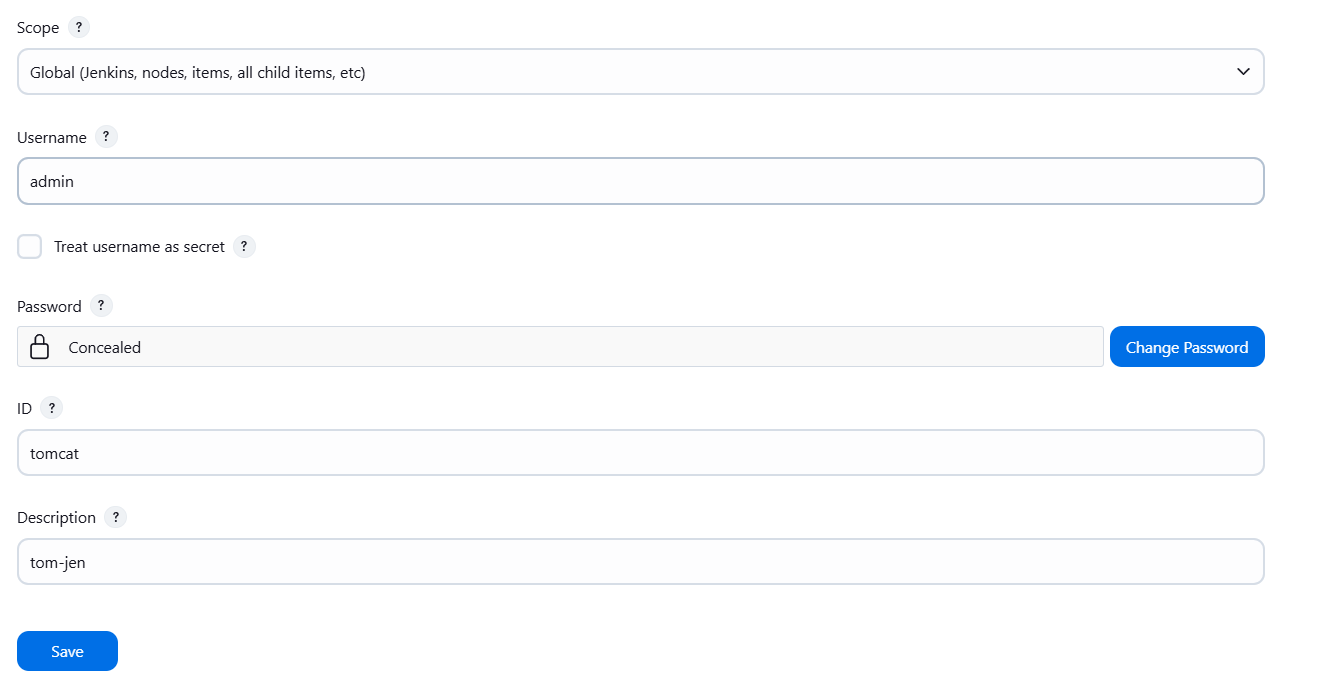
**TOMCAT** :

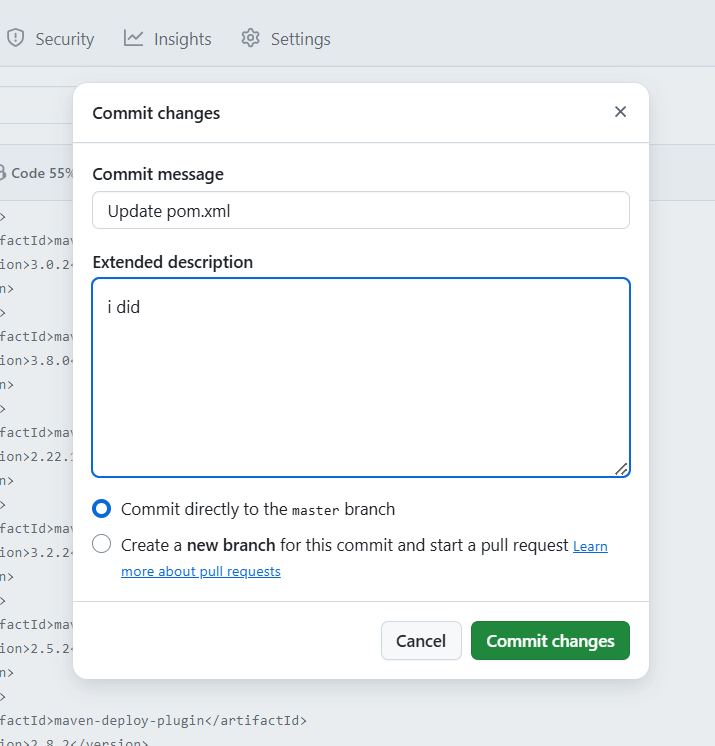
* Create a new instance in aws
* Start the instance
* In security add inbound rules with port 8080
* Open gitbash in pem file location
* Connect with the instance using ssh client link
* Follow the commands to install tomcat
* sudo apt update
* java and javac to intall java
* sudo su –
* hostname tomcat
* sudo su –
* cd /opt
* Copy the tar file address in this website <https://tomcat.apache.org/download-90.cgi>
* wget <tar file url>
* tar -xvzf <file name>
* mv <file name> tomcat
* cd tomcat
* cd bin
* ./startup.sh
* In browser tomcat instance public IP address:8080
* find / -name context.xml
* vi /opt/tomcat/webapps/manager/META-INF/context.xml
* Comment down the valve lines.Do the same in
* vi /opt/tomcat/webapps/host-manager/META-INF/context.xml
* ./shutdown.sh
* ./startup.sh
* To Configure tomcat go to tomcat folder
* cd conf
* vi tomcat-users.xml
* cd ../bin/
* ./shutdown.sh
* ./startup.sh
* Again refresh the browser now you are able to login to tomcat

**Deploy war file to tomcat**

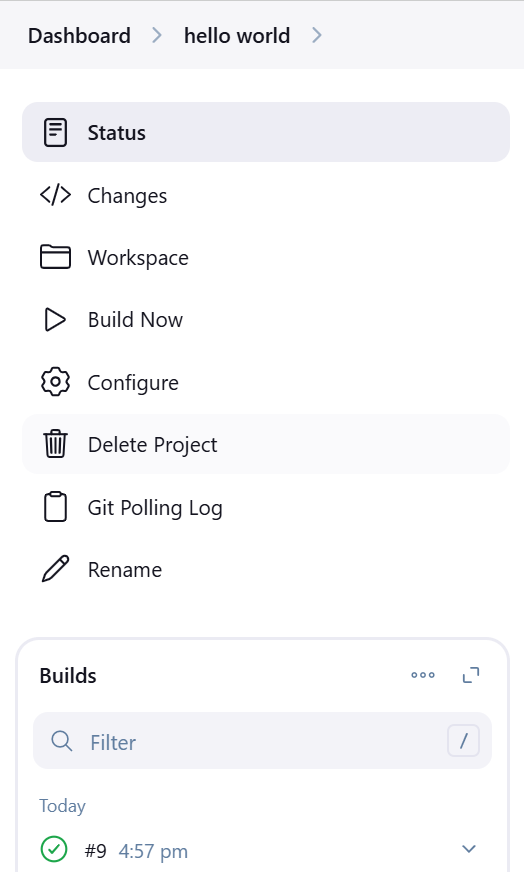
* Add a new item in jenkins
* Go to manage jenkins then plugins then available plugins
* Install copy artifact,Deploy to container plugins
* Go to configuration of item
* In source code management select git and give github repository url and add credentials
* Password is github token



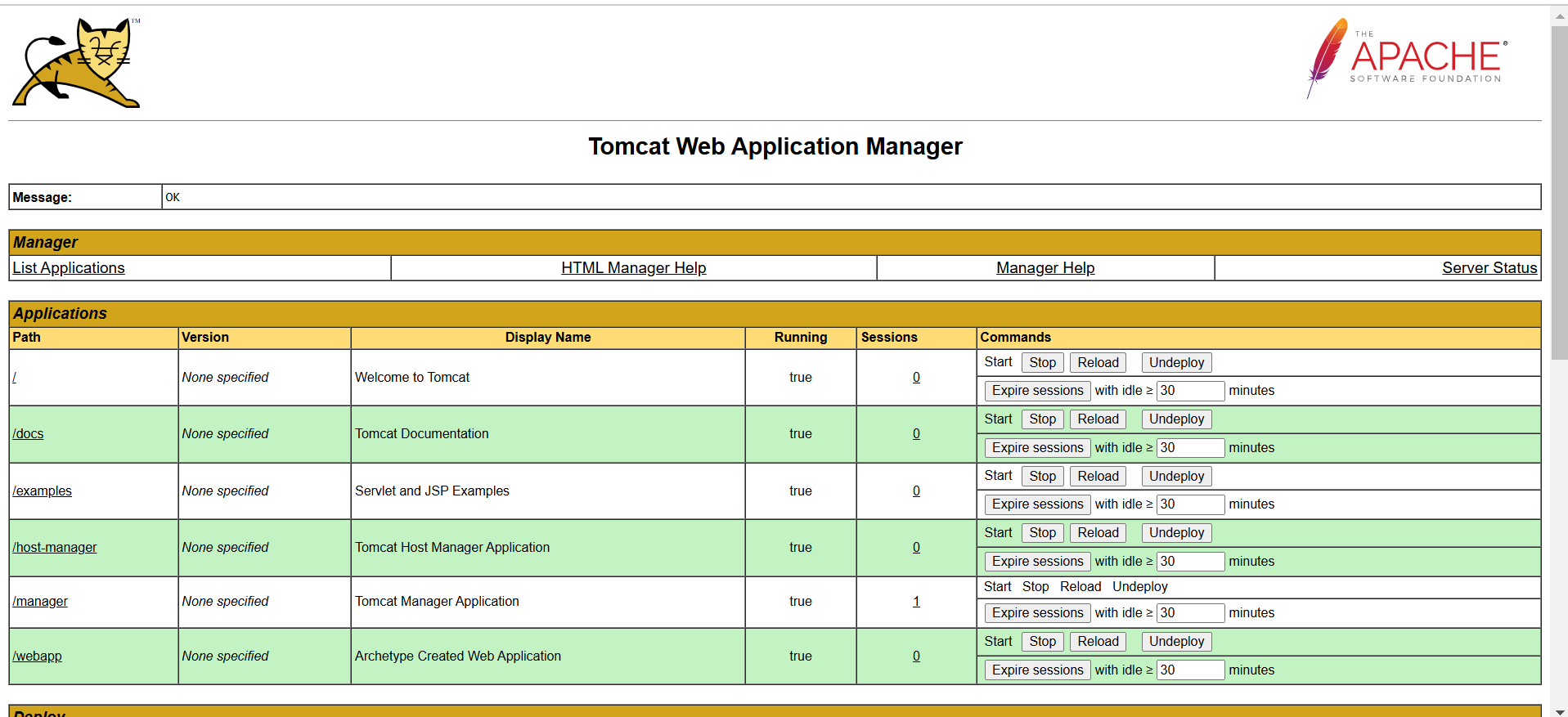
* In build triggers select poll scm
* In build steps select invoke top level maven targets and goal as package
* In post build actions select deploy war/ear to container
* Tomcat url is tomcat instance private Ip address:8080
* Add credentials as username and password of jenkins
* Save and apply
* Whenever there is a commit in github abuild is initiated and war file is deployed to tomcat



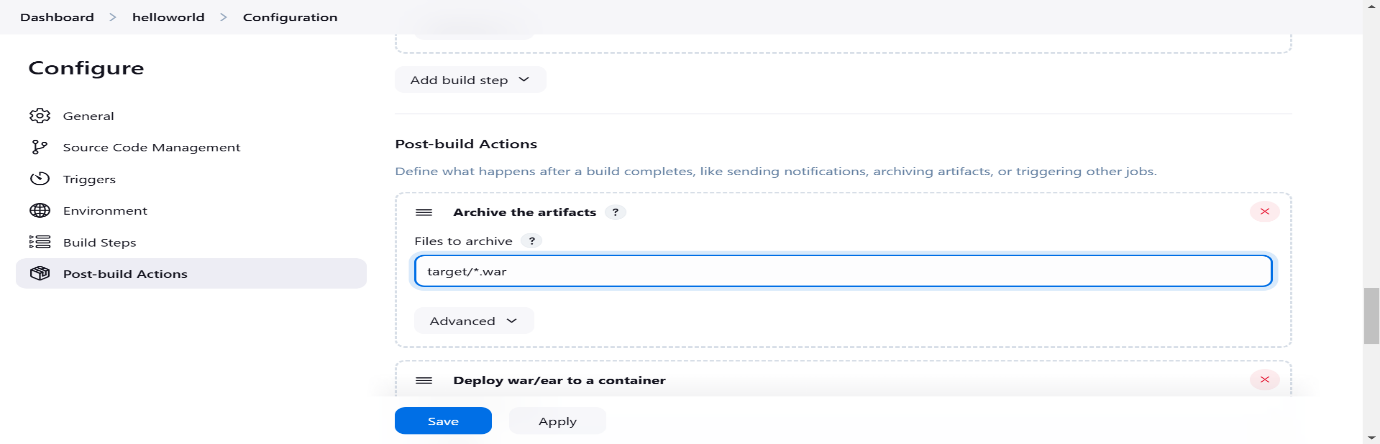
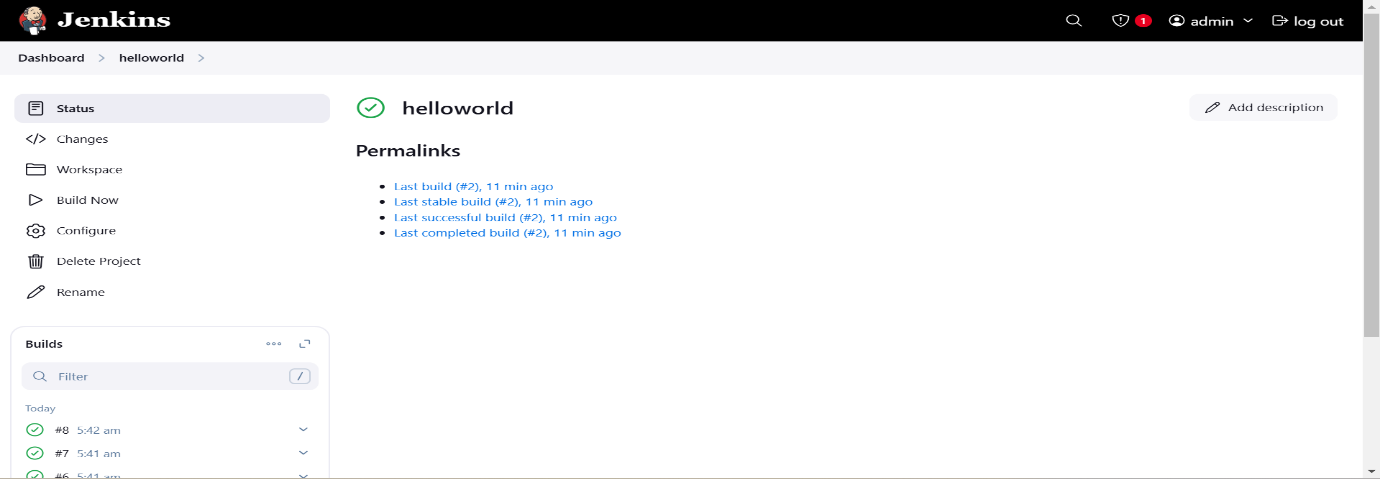
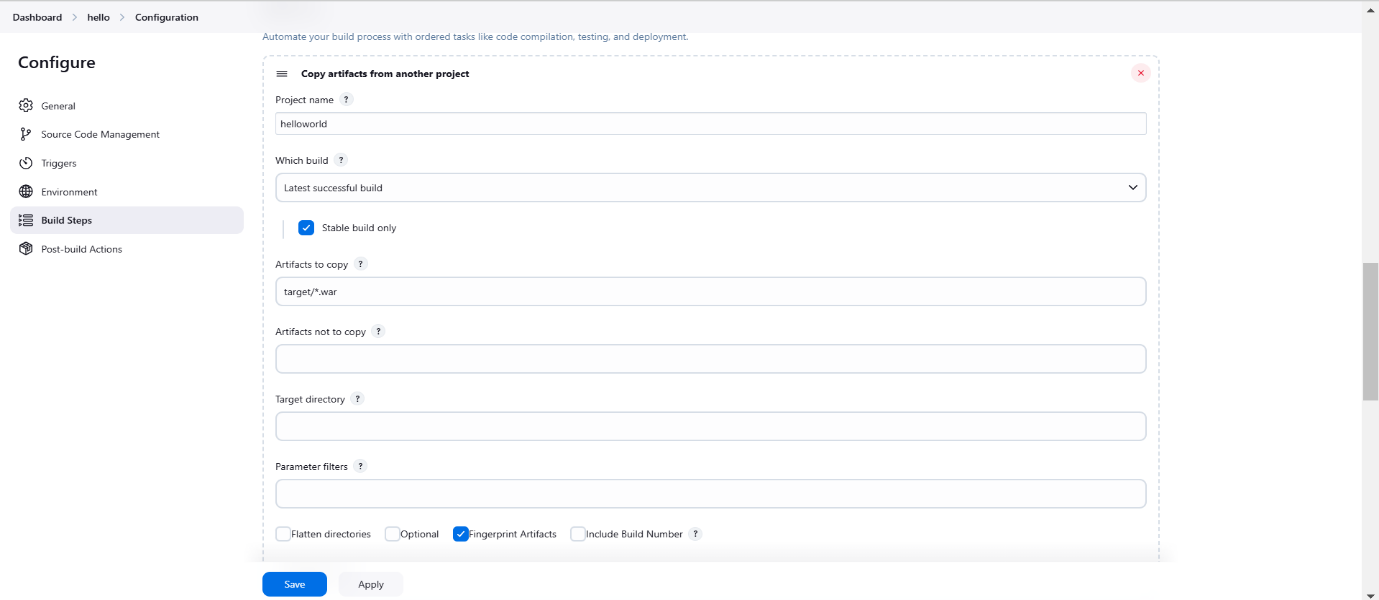
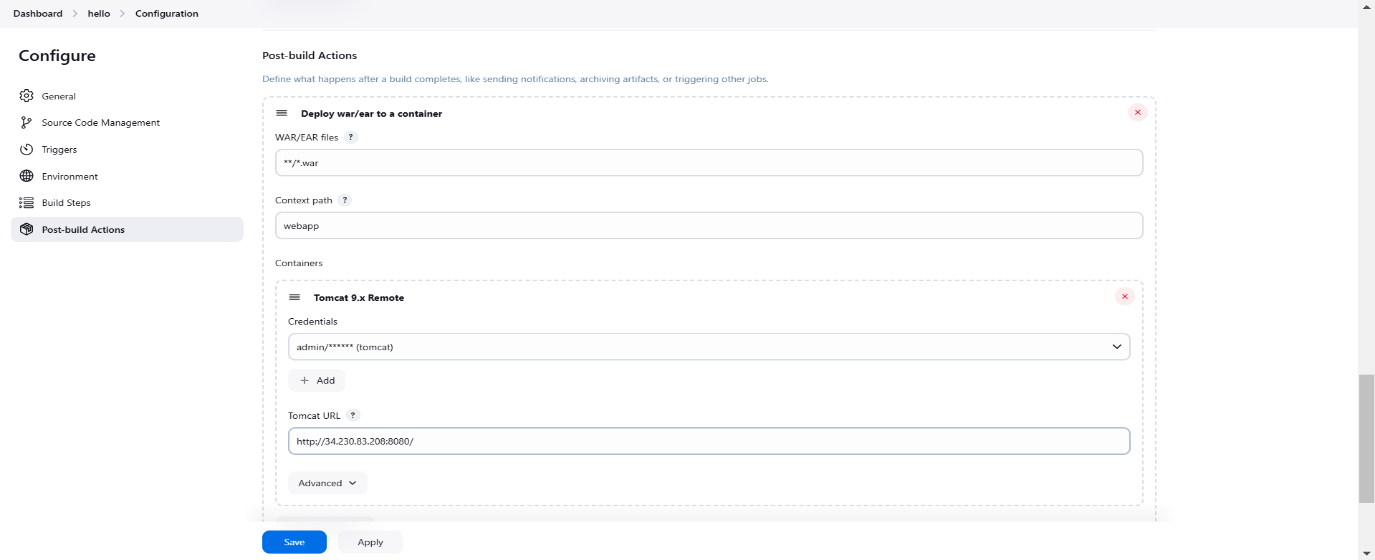
* We can observe a build in jenkins

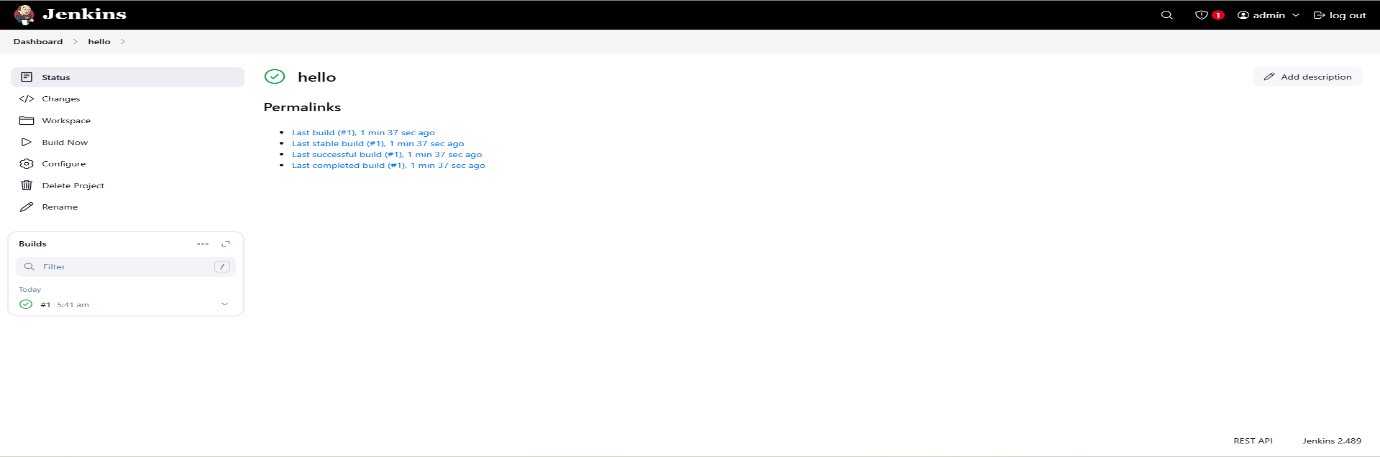


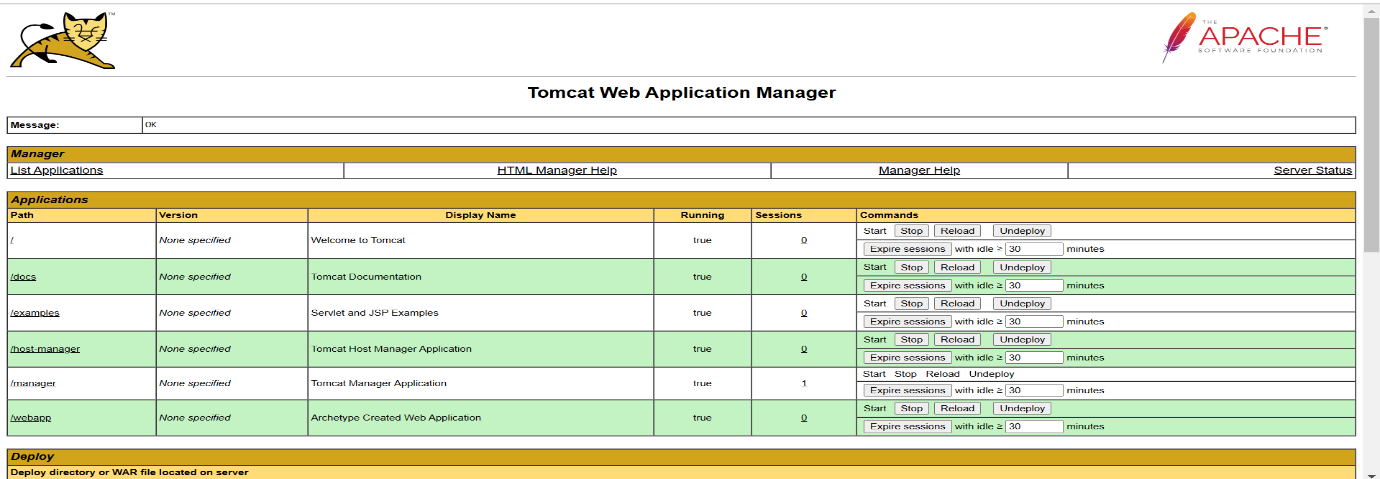
* Our project is deployed into tomcat
* Refresh tomcat
* Go to manager apps
* Login
* We can see webapp



**Deployment from one server to another server:**

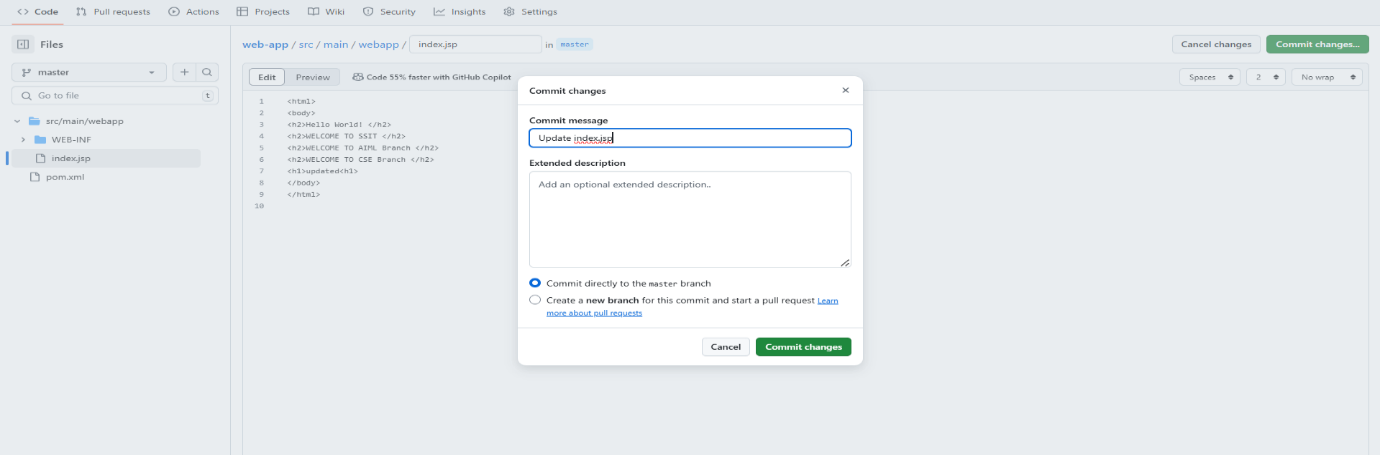
* In the post build actions of already deployed job(hello world) make the following changes
* Save and apply and execute a successful build
* Create new job(hello)
* Add only build step and post build actions in the configuration of this job as given below
* In the Tomcat url add the Public IP of the new tomcat server
* Save and apply and execute a successful build



* In new tab open the new tomcat server by using it’s public IP:port number.We can see webapp here



**Updating the changes**

* Commit changes of the job’s repository and execute latest successful build’s in both jobs
* Now we can see the changes in tomcat server’s webapp

