**Name:** Gopavaram Mohan Kumar

**Batch:** April 15th, 2022

**Tutor:** Raham Shaik

**Project Name: Automation Deployment Through Jenkins**

**WAR FILE DEPLOYMENT IN TOMCAT:**

1. Installed Jenkins and connected slave to Jenkins master to ssh connection.
2. Installed prerequisites in master and slave nodes.
3. Logged into the Jenkins and created a Build Executor node(Slave) on Jenkins. By following the steps Manage Jenkins -> Manage node -> New Node -> Name -> Permanent Agent -> OK
4. The Build Executor is online.

Graphical user interface, application

Description automatically generated

1. Logged into Slave and install the all the prerequisites ( Git, Maven, Java).
2. Logged into Jenkins and created a New Job GitHub Repos url <https://github.com/gmk1995/mywebapp.git> and build the job. The build is successful.

Text

Description automatically generated with low confidence

1. Fetch the Tomcat repos url to the Jenkins slave server to deploy the web app on the tomcat web server.
2. Untar the Tomcat repos.

Text

Description automatically generated

1. Moved to Tomcat Directory and give read, write, and execute permissions to Webapp Directory.
2. Moved to Directory apache-tomcat-9.0.64/webapps/manager/META-INF and opened the content.xml file and deleted the two lines (value and allow tag lines).
3. Moved back to Directory apache-tomcat-9.0.64/conf and open tomcatuser.xml, Added tomcat server username, password, and roles to user.Text

   Description automatically generated
4. Moved to Directory apache-tomcat-9.0.64/bin and run the command (./startup.sh) to start the tomcat services in the Slaves Node.



1. Taken the Slave Node Public IP address and assessed the Tomcat server with port number 8080.

Graphical user interface, website

Description automatically generated

1. Clicked on Manager App and used the Tomcat user login details which I have added in the tomcatuser.xml file.

Graphical user interface, application

Description automatically generated

1. Successfully Logged into the Tom Web Application Manager.

Graphical user interface, application

Description automatically generated

1. Configured Plugins (Deploy to Container) in Jenkins to install the web app in the tomcat web server.
2. Configured the Previous build job with post-build actions -> Deploy war/ear to a container and added tomcat user details.
3. Build the Job (webapp\_deployment). The Build is successful.

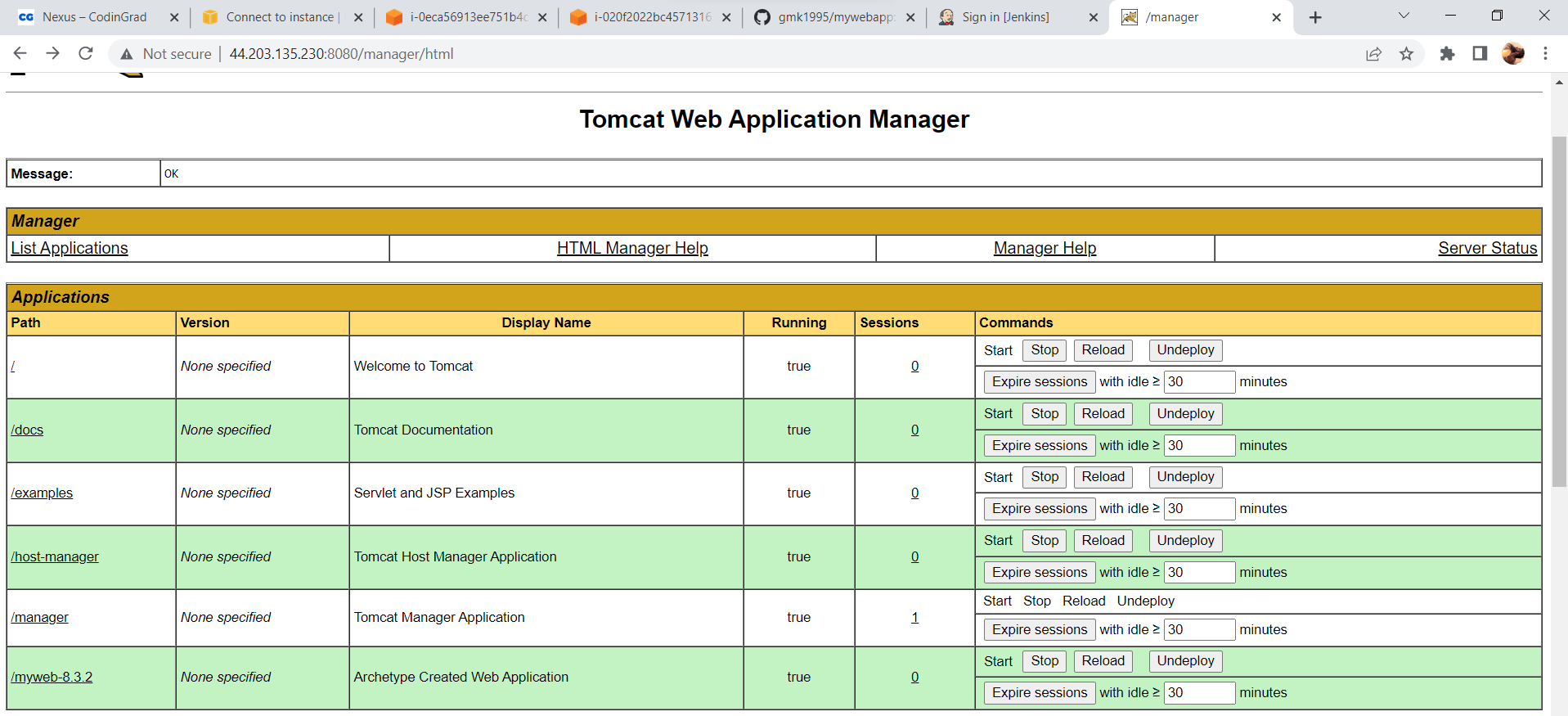
Graphical user interface, text, application, email

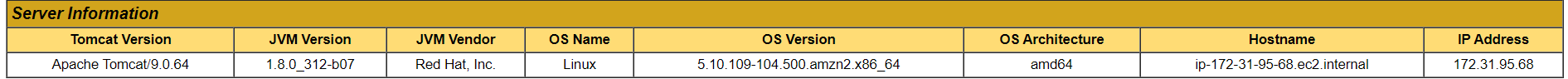
Description automatically generated

Graphical user interface, text, application

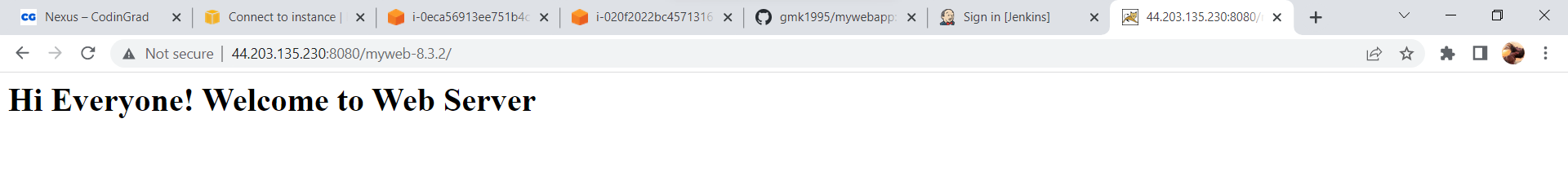
Description automatically generated

1. Went to Tomcat Webserver and refreshed the page the Webapp installation was successfully installed.





1. Clicked on the myweb-8.3.2. Web app accessed successfully.



1. Configured the webapp\_depolyment job with poll scm by adding (\* \* \* \* \*).
2. Logged into GitHub and went to src/main/webapp/index.jsp and edited the Html code.

Graphical user interface, text, application

Description automatically generated

1. New Build is automatically triggered and the Build is successful.

Graphical user interface, text, application

Description automatically generated

1. Refreshed the tomcat server and clicked on myweb-8.3.2. Web app to access the webapp.

