**Team Name**: Bay Area Rockers

**Team Members**: Shawn Chumbar, Dhruval Shah, Sajal Agarwal

CMPE 272: HW#5 - Jenkins

**Jenkins Homework**

References:

1. <https://www.youtube.com/watch?v=ANd6Qzkwjus&ab_channel=Coralogix>
2. <https://www.youtube.com/watch?v=AYohbnOqox0&ab_channel=GeeksCodeBook>
3. <https://www.youtube.com/watch?v=PhxZamqYJws&ab_channel=JavaHomeCloud>
4. <https://coralogix.com/blog/how-to-install-and-configure-jenkins-on-the-mac-os/>

For this assignment, our goal was to integrate Jenkins with our CMPE-272 GitHub Repository. Please see below for the steps we took to complete this assignment:

1. Installed Jenkins onto my mac using Docker.
2. Completed the initial Jenkins Configuration.
3. Created a new Jenkins Job named “CMPE272 Jenkins HW”.
4. Added Job Description
5. Selected “Git” for the option labeled **Source Code Management**
6. Inputted the GitHub Repository URL for our project (NOTE: Since the GitHub Repository is currently private, we had to generate a Personal Access Token (PAT) from GitHub by going to:
7. **Profile > Settings > Developer Settings > Personal Access Tokens > Tokens (Classic) > Generate New Token > Generate new token (classic).**
8. After generating the token, we then copy-and-pasted the token.
9. Within Jenkins, we selected **Add > Jenkins** to add the credential to Jenkins.
   1. Insert GitHub username in “Username” field.
   2. Insert GitHub PAT into “Password” field.
   3. Insert **Jenkins\_PAT** into the “ID” field.
   4. Click **Add.**
10. Change the “Branch Specifier” field to **\*/main**
11. Under the “Build Triggers” section, select **GitHub hook trigger for GITScm polling**
12. Under “Build Environment”, select **Delete workspace before build starts**
13. Under “Build Steps”, add a build step to echo Hello World.
14. Click **Apply** and **Save**.
15. Click **Build Now** to start a new pipeline run.

After all the above steps are completed, we have successfully created a Jenkins job for our GitHub Repository. The reason I created a simple “Hello World” pipeline run is because Jenkins does not allow me to run windows commands to set up my project. I figured creating a “hello world” job would show that I am able to create a pipeline and am able to pull the associated GitHub Repo into Jenkins. All that is left for me to do is to figure out the steps required to build my project in MacOS (I usually use one of my classmate’s computers to test changes for the project).

Please see below for screenshots showcasing the various steps for this assignment.

1. Setting up Jenkins via MacOS CLI and Docker:

A screenshot of a computer

Description automatically generated

1. Jenkins Job Creation:

A screenshot of a computer

Description automatically generated

1. Jenkins Description:

A screenshot of a computer

Description automatically generated

1. Creating GitHub Personal Access Token (PAT)

A screenshot of a computer

Description automatically generated

1. Configuring Jenkins Job Source Code Management Options

A screenshot of a computer

Description automatically generated

1. Configuring Jenkins Job Build Triggers

A screenshot of a computer

Description automatically generated

1. Configuring Jenkins Job Build Steps

A screenshot of a computer

Description automatically generated

1. Configuring Jenkins Job Post-build actions

A screenshot of a computer

Description automatically generated