**Set Up a Docker Jenkins Master:**

1. sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
2. sudo yum clean all
3. sudo yum install docker-ce-17.03.0.ce-1.el7.centos
4. sudo systemctl start docker
5. sudo docker pull jenkins:2.19.4
6. sudo mkdir /var/jenkins\_home
7. sudo docker run -d -u root -p 8080:8080 -p 50000:50000 -v /var/jenkins\_home:/var/jenkins\_home jenkins:2.19.4

**Forking Github repo :**

<https://github.com/vishnugonela/content-jenkins-freestyle.git>

**Docker login:**

1. sudo docker exec -it <ContainerID> bash
2. cd ~
3. ssh-keygen
4. cat .ssh/id\_rsa.pub

Return to your fork of the Jenkins Freestyle project on GitHub and press **Settings**, then **Deploy keys**. Press **Add deploy key**. Copy your public key into the text box. Give the key a title. Check *Allow write access*. **Add key**.

Press **Integrations & services** on the left menu. Under **Services, select Add service**, then find *Jenkins (Git plugin)*. For the Jenkins URL input the server's IP address, including the port 8080, then press Add Service.

Return to the Jenkins **Dashboard** and click on **New Item**. Give it an item name of *Lab Freestyle*, then select **Freestyle project**. Click OK.

Check the **GitHub project checkbox**, and input the GitHub URL for your fork of the project. Under **Source Code Management**, select **Git**, then input the repository's clone with SSH link, found under the green **Clone or download** button on the repository's page. Press **Add, Jenkins** besides the **Credentials** option. For Kind select **SSH Username with private key**. Set the **Username to *root*,** then set **Private Key** to From the **Jenkins master ~/.ssh**. Press Add. Now select the key from the credentials dropdown.

For Repository Browser, select **githubweb** and set the URL to the main project URL. Save the changes.

**Adding Build Steps and a Trigger:**

Under the project page, select Configure from the left menu. Scroll down to the Build Triggers section and check Poll SCM. We need to do this to get the GitHub plugin to work. Leave the schedule box empty.

Find the Build section and under the dropdown menu select Execute shell. We want to set some environment variable interpolation.

Input:

echo "${JOB\_NAME} [${BUILD\_NUMBER}]"

Add a second *Execute shell* with the following to compile the Java code:

echo "Building Java Code"

javac -d . src/\*.java echo Main-Class: Rectangulator > MANIFEST.MF

jar -cmf MANIFEST.MF rectangle.jar \*.class

Add another Execute shell to execute the rectangle file:

java -jar rectangle.jar 2 3

And click **Save**.

**From your Workstation:**

1. git clone <https://github.com/USERNAME/contentjenkins-freestyle.git>
2. cd content-jenkins-freestyle/
3. echo "My rectangle Java code" >> README.md
4. git commit -am "Added subtitle to the README.md file"
5. git push origin master

Check now console output from jenkins browser.