**Setting Up Jenkins Pipeline**

This section will guide you to**:**

* Build a Docker Jenkins pipeline to implement CI/CD workflow.

This lab has three subsections, namely**:**

4.6.1 Installing Docker plugin and configuring Docker cloud

4.6.2 Configuring Jenkins job

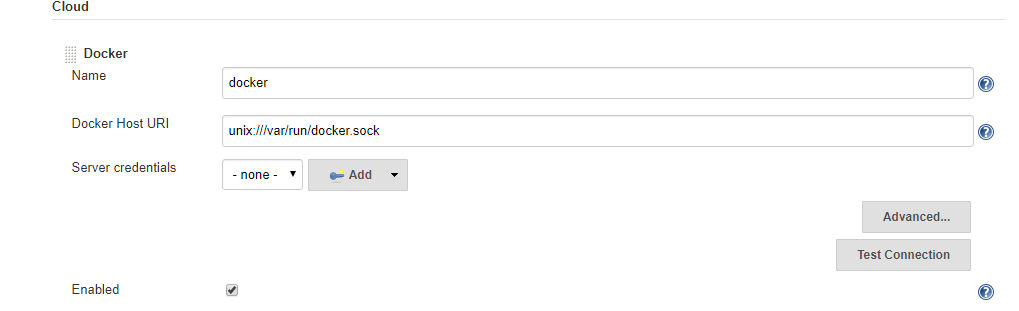
4.6.3 Pushing the code to GitHub repositories

* Docker is already installed in your lab. ()

**Step 4.6.1:** Installing Docker plugin and configuring Docker cloud

* Add Docker cloud by accessing Manage Jenkins 🡪 Configure system. Then, you have to add Docker cloud details as shown below:

Docker Host URI: unix:///var/run/docker.sock



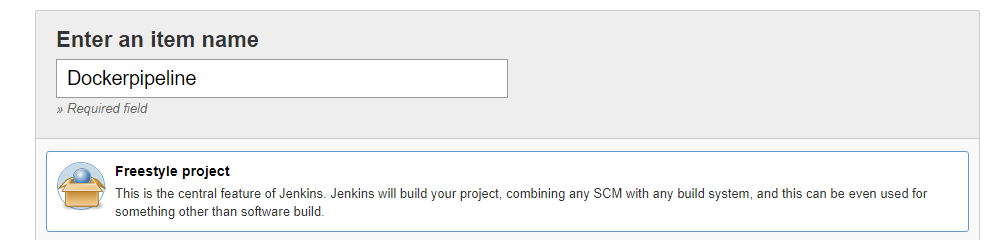
* Configure the Docker cloud to give complete access to docker.sock file so that Jenkins will be able to connect to Docker process

chmod 777 /var/run/docker.sock



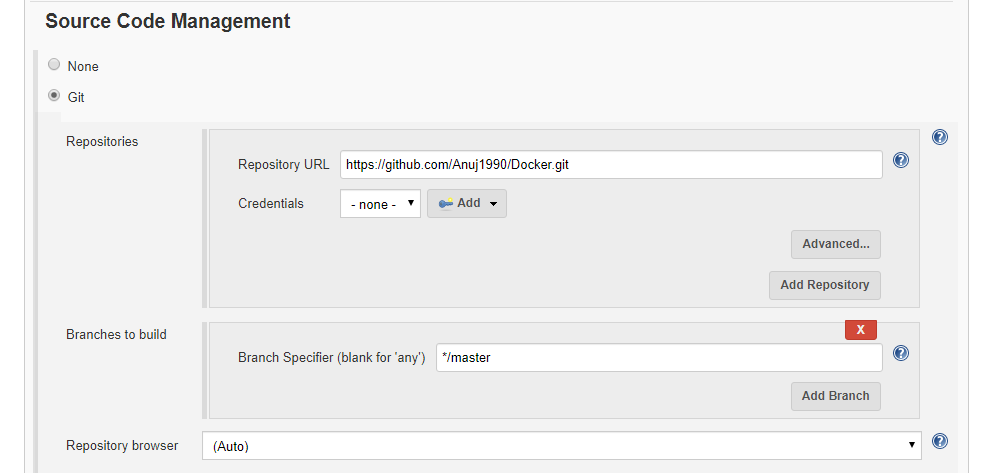
**Step 4.6.2:** Configuring Jenkins job

* Create a new Jenkins pipeline job for supporting CI/CD workflow

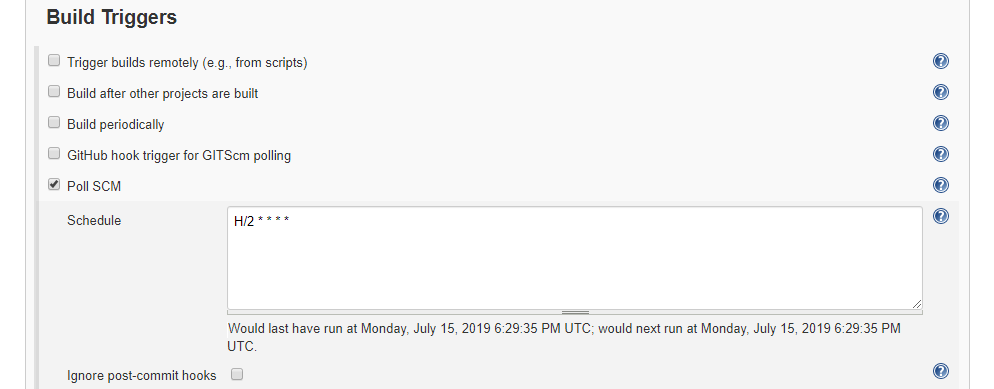


* Configure Git repository so that we can have Dockerfile to build Docker container and push it to Docker Hub

https://github.com/Anuj1990/Docker.git



* Configure build triggers to enable Poll SCM feature so that once any push is detected



* Configure Docker build option to configure build configurations. Some of the configurations are mentioned below:

Directory for Dockerfile . (Represents current location)

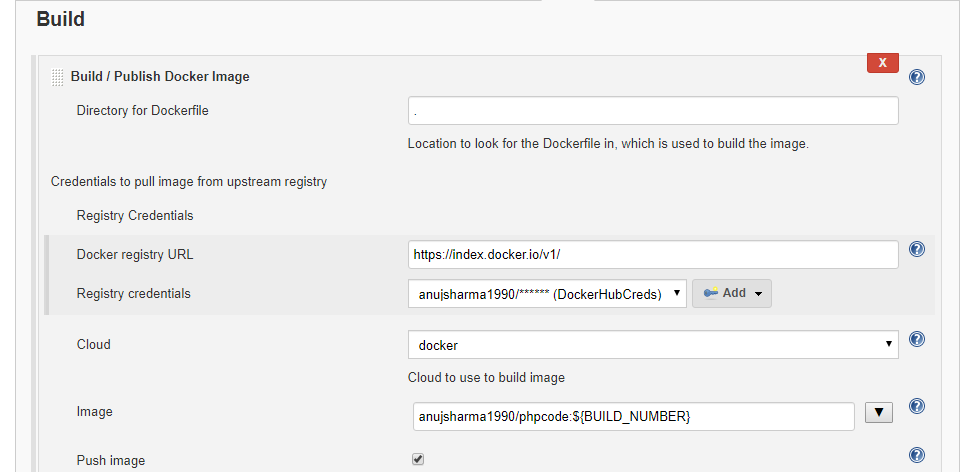
Docker Registry URL https://index.docker.io/v1/

Docker credentials Docker hub username password

Cloud Select Docker Cloud created in dropdown

Image anujsharma1990/phpcode:${BUILD\_NUMBER}

Registry Credentials Docker hub username password

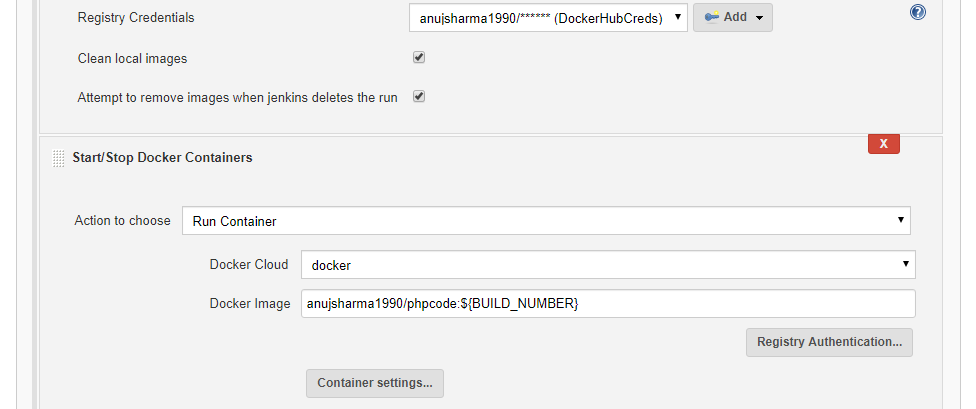


**Please Note:** We have configured how our custom Docker image will be pushed to Docker Hub. Now, we have to configure the Docker container deployment.

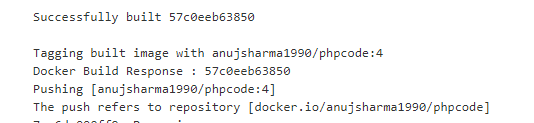
* Follow the steps mentioned below to configure how Docker containers will be deployed

Docker cloud Select from dropdown Docker cloud details

Docker image anujsharma1990/phpcode:${BUILD\_NUMBER}

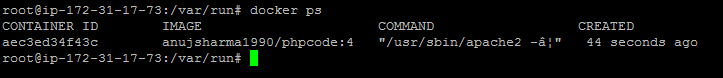


* Once job configuration is done, save the configuration and proceed with triggering build in order to build custom container and deploy the container





* Once the build is successful, validate the Docker container deployment on Docker host which will help us to implement complete CI/CD workflow for Docker container



**Step 4.6.3:** Pushing the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**