|  |
| --- |
|  |
| Dockerizing Jenkins Pipeline |
| Technical document |

|  |
| --- |
| Aryashree Pritikrishna  11/10/2020 |

In this technical document you will find the steps for the continuous integration and delivery by Dockerizing Jenkins Pipeline. We will cover below topics in this document.

* GitHub – To store the application code and track its revisions
* Git – To connect and push files from local system to GitHub
* Docker – To build the application in a Docker container and push it to Docker Hub
* Docker Hub – To store the Docker image
* Linux (Ubuntu) – As a base operating system to start and execute the project
* Jenkins – To automate the deployment process during continuous integration

**STEP 1**

Setup a code workspace and the Github repository.

1. Open the code editor.
2. Create a directory “simplilearn-devops-certification”.
3. Run “git init” to initialize repository in “simplilearn-devops-certification” directory.
4. Create a repository “simplilearn-devops-certification” in the github.com.
5. Create a file name “README.md” add details of the project in the file.
6. Execute the below steps and commit the code to github repository.
7. git remote add origin https://github.com/aryashreep/simplilearn-devops-certification.git
8. git add .
9. git commit –m “ Initial Commit”
10. git push origin master

**STEP 2**

Setup a Jenkin Server and a docker machine.

Before the Jenkins install make sure Java 8 JDK should installed.

1. Download Jenkin.war from <http://mirrors.jenkins.io/war-stable/latest/jenkins.war>
2. Execute this on a terminal of the machine using   
   “java -jar D:\Softwares\jenkins.war”
3. The previous command will run the Jenkin server at localhost:8080 port and it can be accessed in the browser using <http://localhost:8080/>
4. Select “Install Selected packages” and wait for installation to finish.
5. Create a new user after installation is finished.
6. Install the “Docker Pipeline” plugin so it will help to initialize the Docker commands.
7. Install a docker on the same machine where Jenkin is installed.

**STEP 3**

Setup Jenkinsfile in the repository

1. Create a “Jenkinsfile” and “Dockerfile”, commit to the repo.
2. Add following content in the “Jenkinsfile” file.

pipeline {

environment {

registry = "aryashreep/simplilearn-devops-certification"

registryCredential = 'docker-hub-credentials'

dockerImage = ''

}

agent any

stages {

stage('Cloning our Git') {

steps {

git 'https://github.com/aryashreep/simplilearn-devops-certification.git'

}

}

stage('Building our image') {

steps {

script {

dockerImage = docker.build registry + ":$BUILD\_NUMBER"

}

}

}

stage('Deploy our image') {

steps {

script {

docker.withRegistry( '', registryCredential ) {

dockerImage.push()

}

}

}

}

stage('Cleaning up') {

steps {

sh "docker images"

sh "docker rmi --force $registry:$BUILD\_NUMBER"

}

}

}

}

node {

stage('Execute Image'){

def customImage = docker.build("aryashreep/simplilearn-devops-certification:${env.BUILD\_NUMBER}")

customImage.inside {

sh 'echo This is the code executing inside the container.'

}

}

}

1. There are four stages in the Jenkins.
   1. Building Image
   2. Deploying the image in the dockerhub repository
   3. Removing the Image from Jenkin node.
   4. Executing Image from dockerhub.

**STEP 4**

Register and open <https://hub.docker.com/> with your own login.

1. Create a file ‘Dockerfile’ in the project created in STEP1
2. Add following content in the file and commit to the repo.

# Dockerfile

FROM busybox

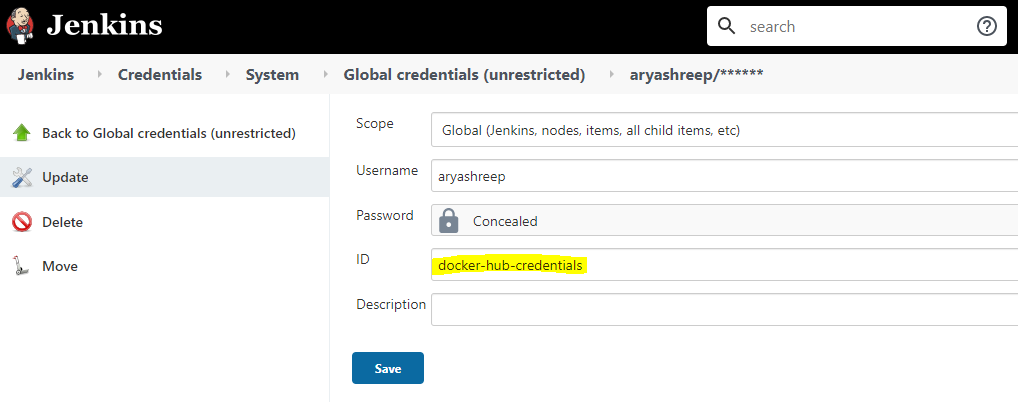
CMD echo "Hello world! This is my first Docker image."

**STEP 5**

Add the Docker hub credentials in Jenkins as it required to push the image to Docker hub. Here are the below steps to add the global credentials in Jenkins.

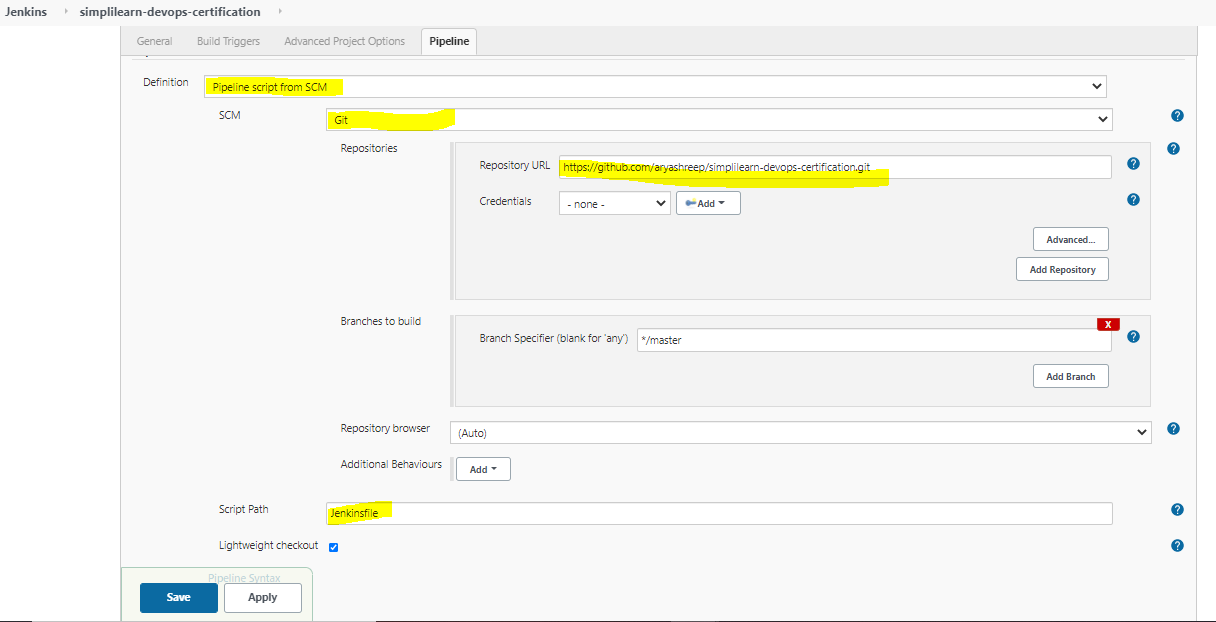
To add new global credentials to your Jenkins instance:

1. If required, ensure you are logged in to Jenkins (as a user with the **Credentials > Create** permission).
2. From the Jenkins home page (i.e. the Dashboard of the Jenkins classic UI), click **Credentials > System** on the left.
3. Under **System**, click the **Global credentials (unrestricted)** link to access this default domain.
4. Click **Add Credentials** on the left.  
   **Note:** If there are no credentials in this default domain, you could also click the **add some credentials** link (which is the same as clicking the **Add Credentials** link).
5. From the **Kind** field, choose the [type of credentials](https://www.jenkins.io/doc/book/using/using-credentials/#types-of-credentials) to add.
6. From the **Scope** field, choose either:
   * **Global** - if the credential/s to be added is/are for a Pipeline project/item. Choosing this option applies the scope of the credential/s to the Pipeline project/item "object" and all its descendent objects.
   * **System** - if the credential/s to be added is/are for the Jenkins instance itself to interact with system administration functions, such as email authentication, agent connection, etc. Choosing this option applies the scope of the credential/s to a single object only.
7. Add the credentials themselves into the appropriate fields for your chosen credential type:
   * **Username and password** - specify the credential’s **Username** and **Password** in their respective fields.
8. In the **ID** field, specify a meaningful credential ID value - for example, docker-hub-credentials. You can use upper- or lower-case letters for the credential ID, as well as any valid separator character. However, for the benefit of all users on your Jenkins instance, it is best to use a single and consistent convention for specifying credential IDs.  
   **Note:** This field is optional. If you do not specify its value, Jenkins assigns a globally unique ID (GUID) value for the credential ID. Bear in mind that once a credential ID is set, it can no longer be changed.
9. Specify an optional **Description** for the credential/s.
10. Click **OK** to save the credentials.

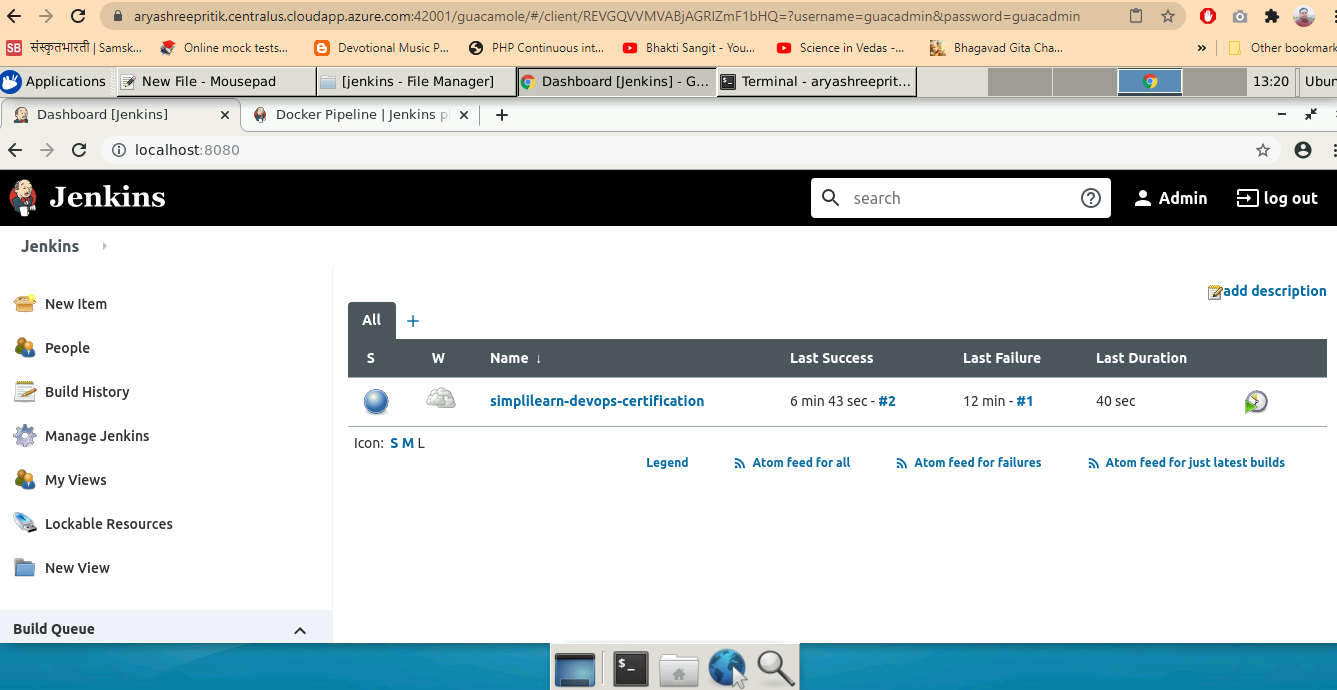


**STEP 6**

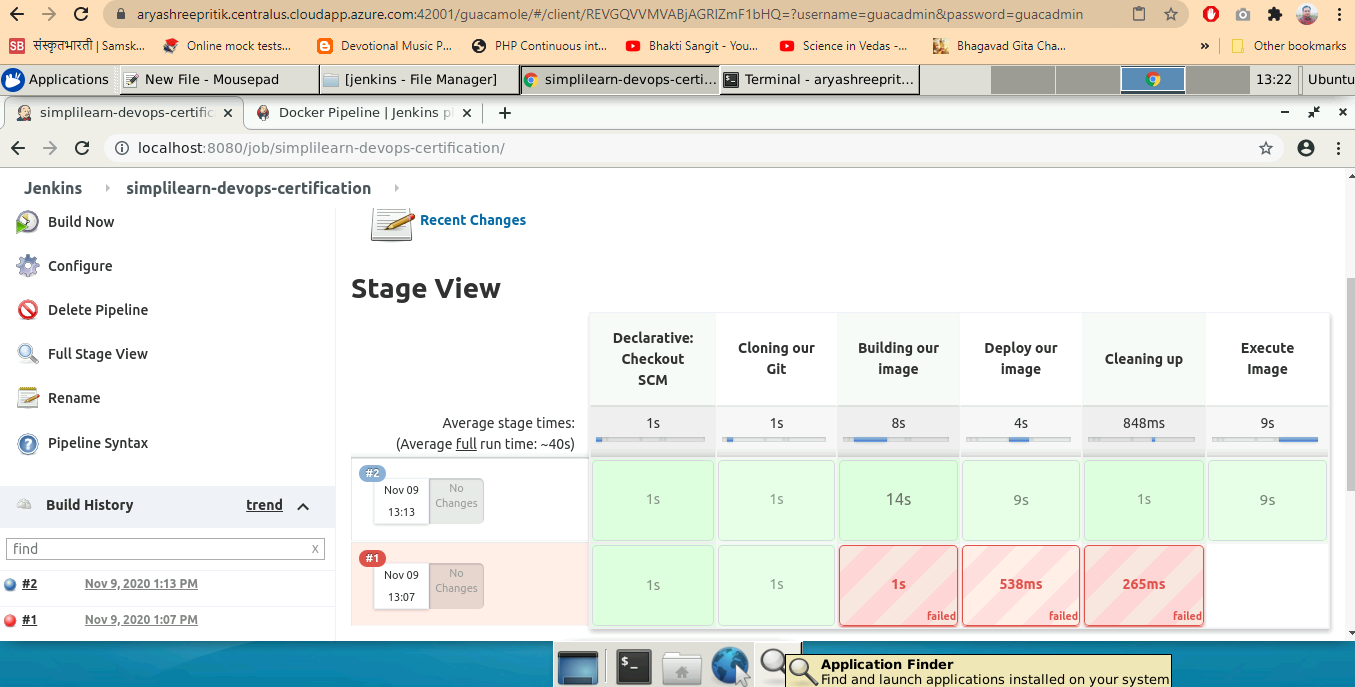
Enable the pipeline in the Jenkins. Here are the steps to create a Pipeline project.

1. Click on the “New Item” on Jenkins dashboard and select the “Pipeline”.
2. Click on the “Pipeline” tab and select the definition “Pipeline script from SCM” and SCM will be “Git”. Please find the below screen-shot for your reference.  
   

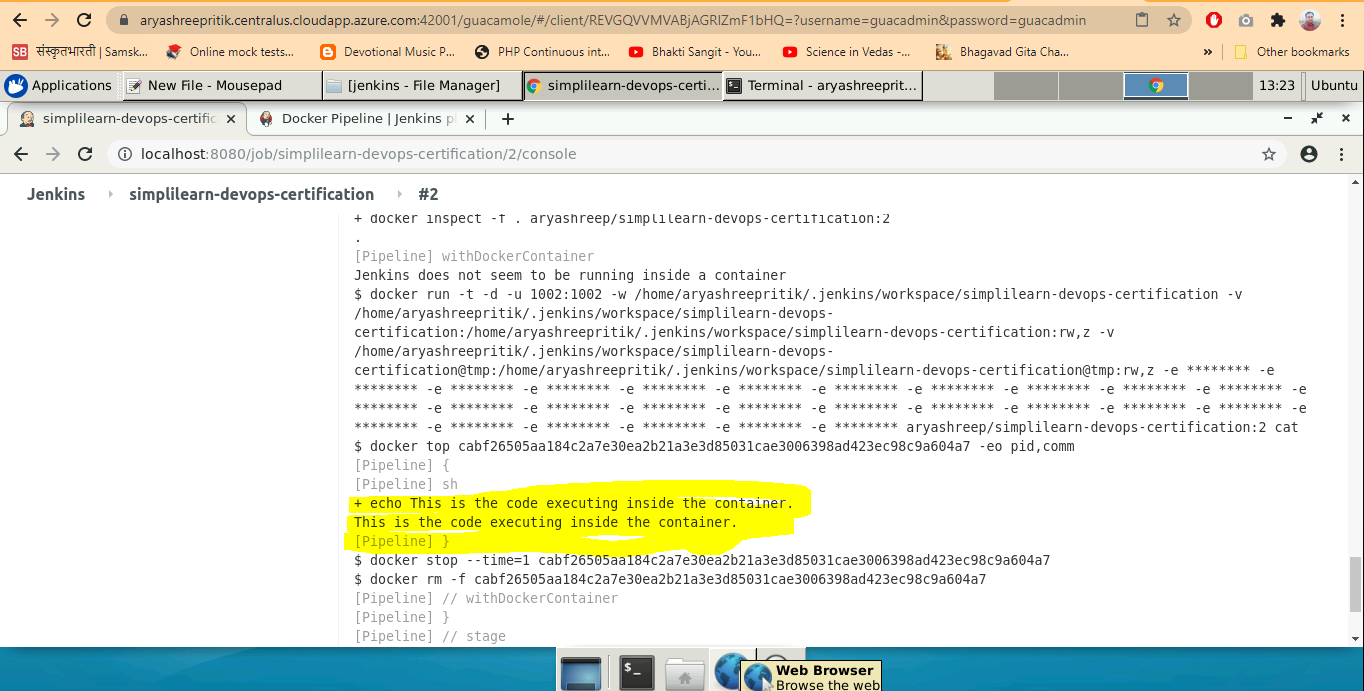
**Jenkin CI pipeline for building and pushing the image to docker hub**



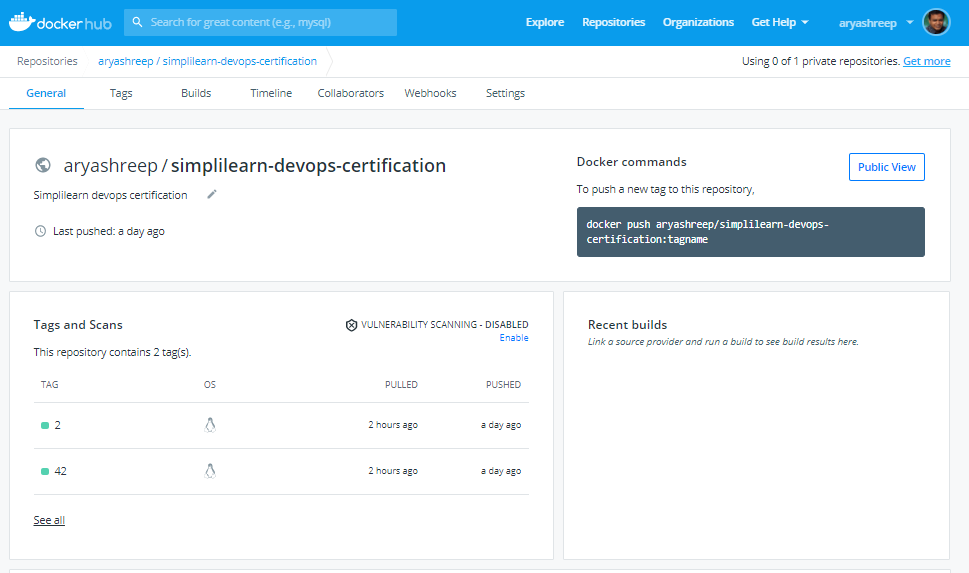
**CI pipeline at the time of execution**



**CI pipeline executing the command “echo This is the code executing inside the container.” Inside the container.**

****

**Docker Hub**

****