**Project Name: Demonstrate Continuous integration and Delivery by using Devops tools.**

DESCRIPTION

Demonstrate the continuous integration and delivery by Dockerizing Jenkins Pipeline.

**Problem Statement Scenario:**   
You are a DevOps consultant in AchiStar Technologies. The company decided to implement DevOps to develop and deliver their products. Since it is an Agile organization, it follows Scrum methodology to develop the projects incrementally. You are working with multiple DevOps Engineers to Dockerize the Jenkins Pipeline. During the sprint planning, you agreed to take the lead on this project and plan on the requirements, system configurations, and track the efficiency. The tasks you are responsible for:

* Availability of the application and its versions in the GitHub.
* Track their versions every time a code is committed to the repository.
* Build the application in Docker and host it in Docker Hub.
* Pull the Docker image and run it again.

The company goal is to deliver the product frequently to the production with high-end quality.

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

**Following requirements should be met:**  
    • Document the step-by-step process from the initial installation to the final stage.  
    • Track the versions of the code in the GitHub repository  
    • Availability of the application in the Docker Hub  
    • Track the build status of Jenkins for every increment of the project

**Demonstration:**

Step 1:

Install Base Operating system in server to deploy application and tools.

**Note: I will be using root credentials to execute all operations in this project**



**Step 2:**

Install pre-requisites tools Docker, Github and Jenkins:

**Git Installation:**

Step 1:

$yum install git

Step 2:

Once installed, you can verify it by placing below commands.

$git –version

To config username and mail, use below commands.

$git config –global user.name “yourname”

$git config –global user. Email “youremail”



**Docker Installation:**

Step 1:

$ yum update

$ yum install yum-utils device-mapper-persistent-data lvm2

Step 2:

Execute below command to enable docker repository in your system.

$ yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

Step 3:

Once the docker repository enabled install latest version of docker using below command:

$ yum install docker-ce

Step 4:

Once the docker installed enable it to start automatically during boot time.

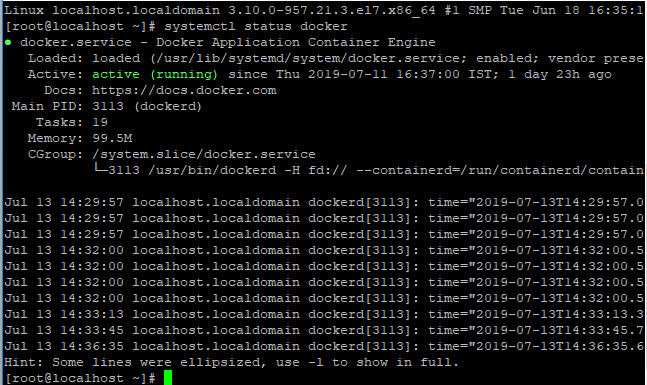
$ systemctl start docker

$ systemctl enable docker

Step 5:

To verify docker installed properly check with below command

$ systemctl status docker



**Jenkins Installation:**

Step 1:

Jenkins is java application and as part of pre-requisite install OpenJDK

$ yum install java-1.8.0-openjdk-devel

Step 2:

Enable Jenkins repository. To do that, import GPG key using curl command

curl --silent --location http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo | sudo tee /etc/yum.repos.d/jenkins.repo

**and add the repository to system using**

$ rpm --import <https://jenkins-ci.org/redhat/jenkins-ci.org.key>

Step 3:

Once repository enabled install latest version of Jenkins

$ yum install Jenkins

Step 4:

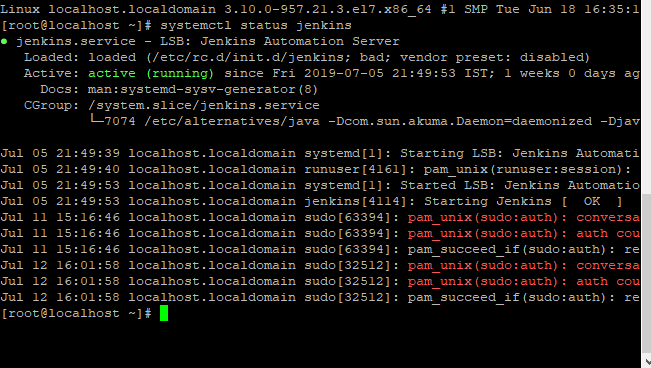
Post installation start Jenkins

$ systemctl start Jenkins

Step 5:

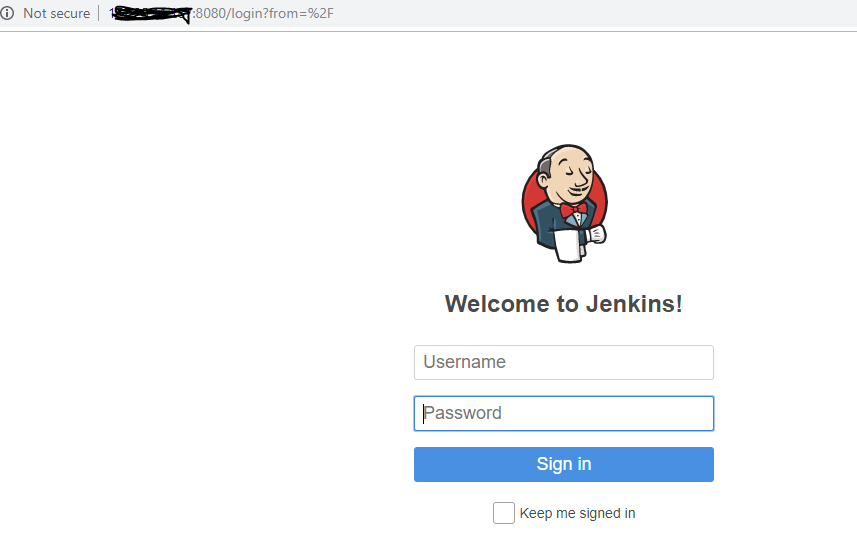
Check status by using below command

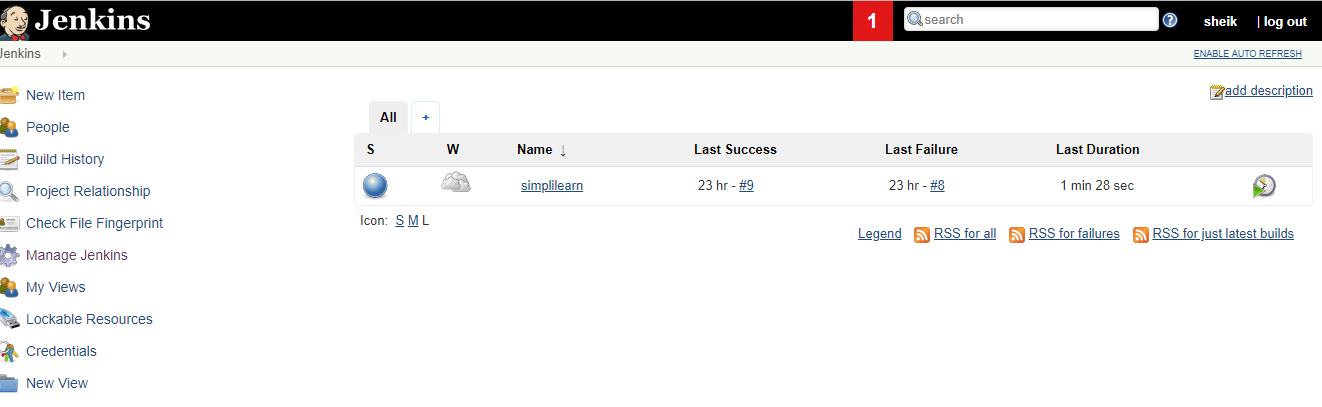
$systemctl status Jenkins



Step 6:

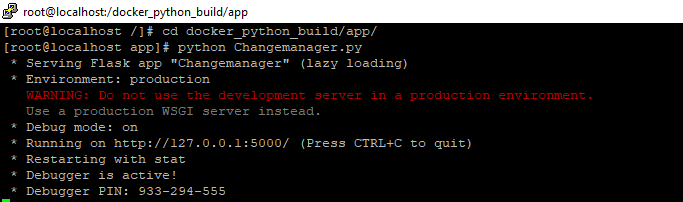
Browse http://your server IP:8080 to login in Jenkins.

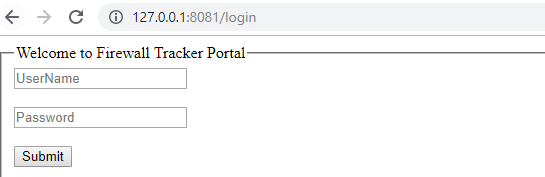




Step 3:

**Develop an application using python and check whether it is running**





Step 4 :

**Since it is running ,build an image using docker file.**

Below are the contents of docker file:

FROM ubuntu:latest

MAINTAINER sheik kalidh "sheik.kalidh@gmail.com"

RUN apt-get update -y

RUN apt-get install -y python-pip python-dev build-essential

COPY ./requirements.txt /requirements.txt

COPY ./app /app

RUN pip install -r requirements.txt

WORKDIR /app

RUN pwd

RUN ls

ENTRYPOINT ["python", "Changemanager.py"]

Step 5:

**Update code to Git hub:**

**1)** git init

**2)** git add .

**3)** git commit -m "First Commit"

**4)** git remote add origin <https://github.com/rkalidh/Simplilearnproject.git>

Upload of project from scratch require git pull origin master.

5) git push origin master

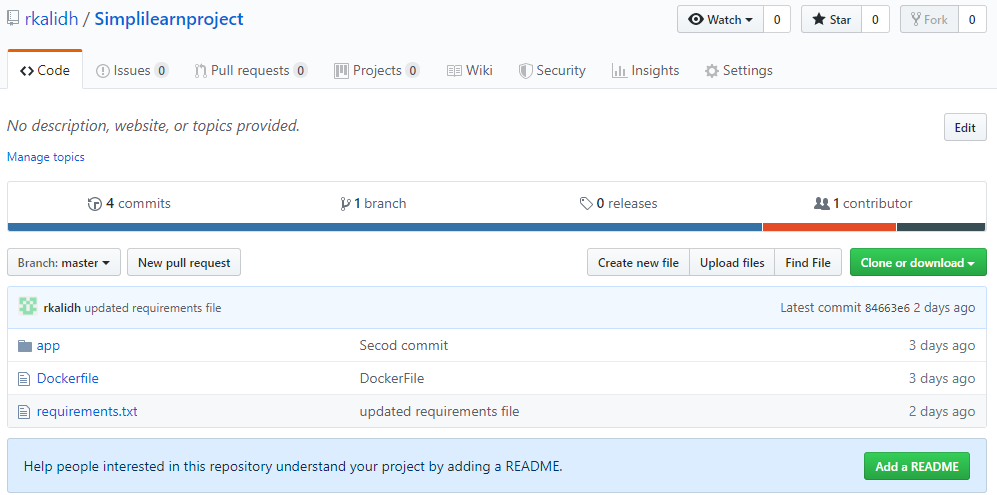
You will be able to see files are pushed to repository in github:

Server files:



**Git hub repository**:

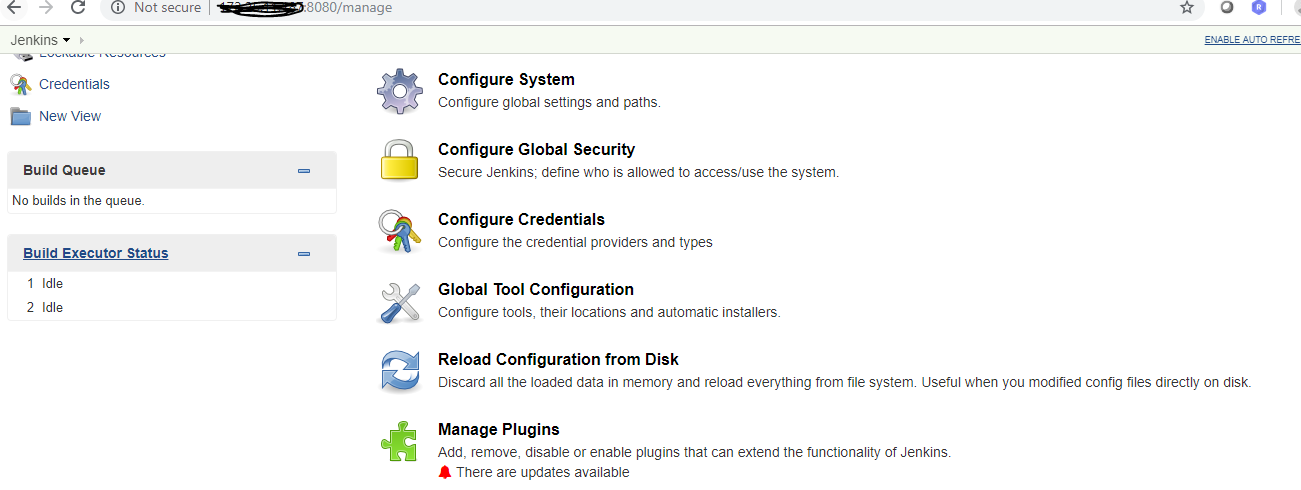
[**https://github.com/rkalidh/Simplilearnproject**](https://github.com/rkalidh/Simplilearnproject)



**Now code is committed in Github: Integrate Jenkins with Git hub as a source code manager.**

Step1:

**Login in Jenkins: Goto -> Manage Jenkins ->manage plugins**

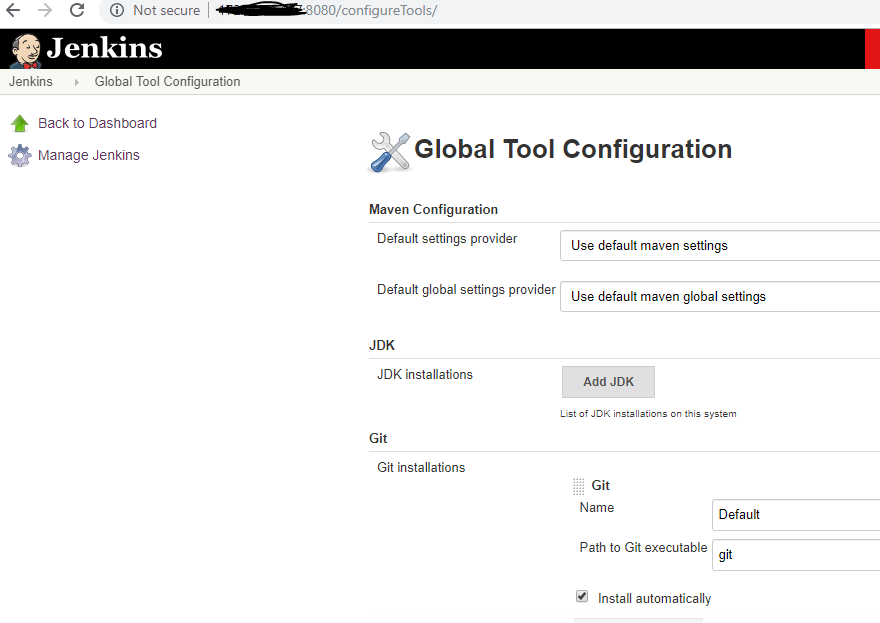


Step 2:

**Go to available plugins -> select Git hub integration plugin and select same to install plugin without restart**

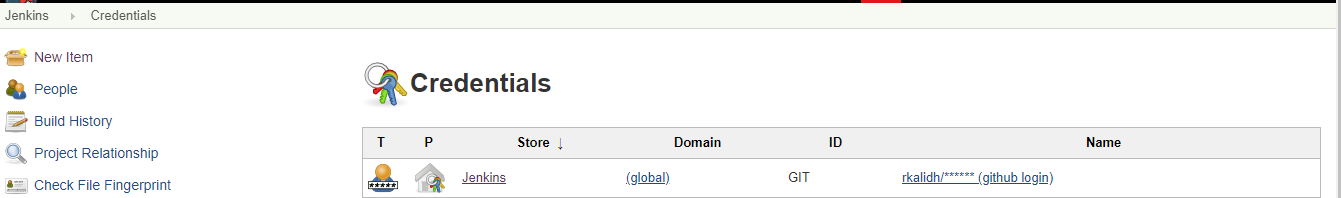
Step 3:

**Go to Manage Jenkins -> Global tool configuration -> Git installations – mention name and path.**

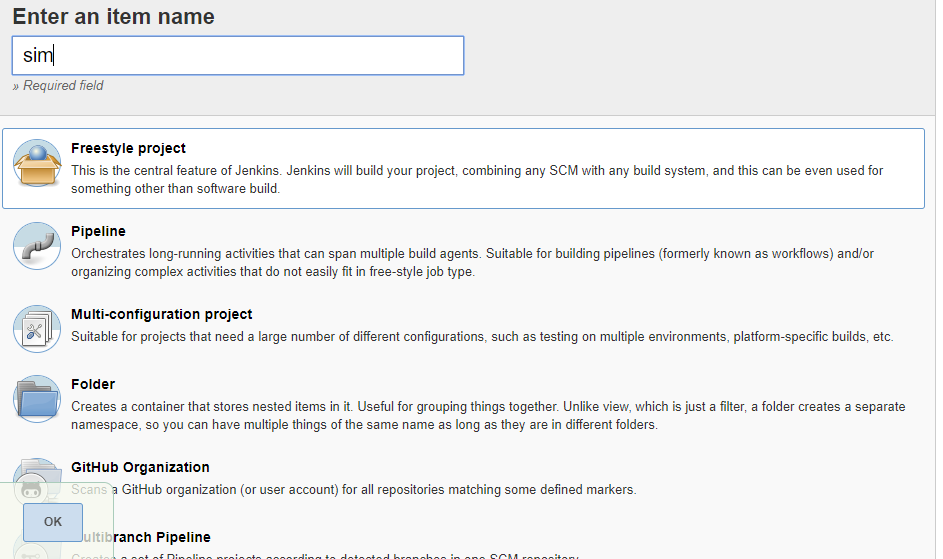


**Now build the container using jobs in Jenkins:**

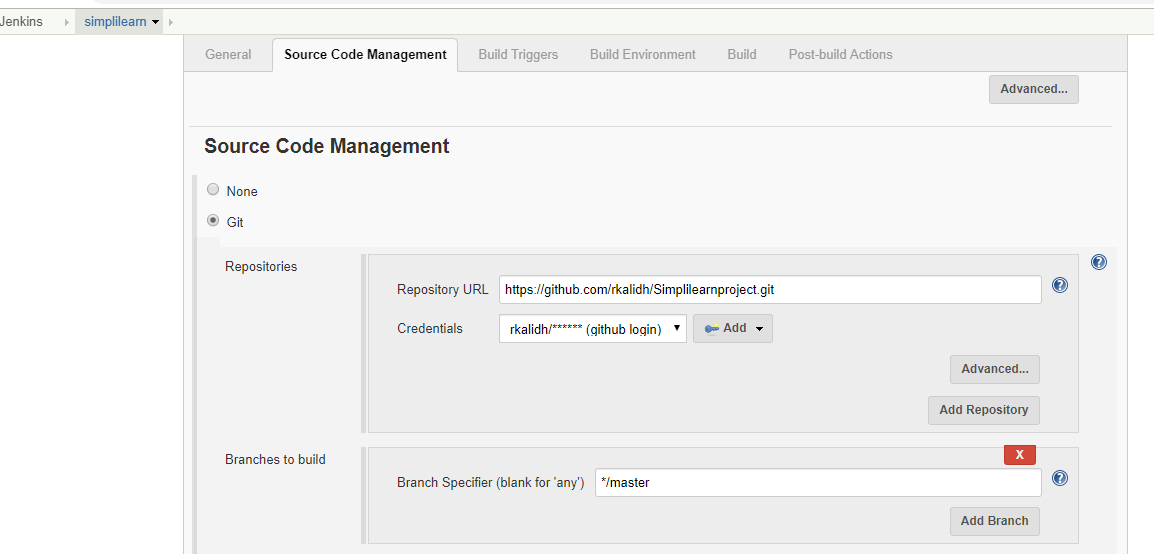
Ensure you added your Github Credentials in Jenkins->credentials:



**Configure new Item – Give name as simplilearn and select free style project.**



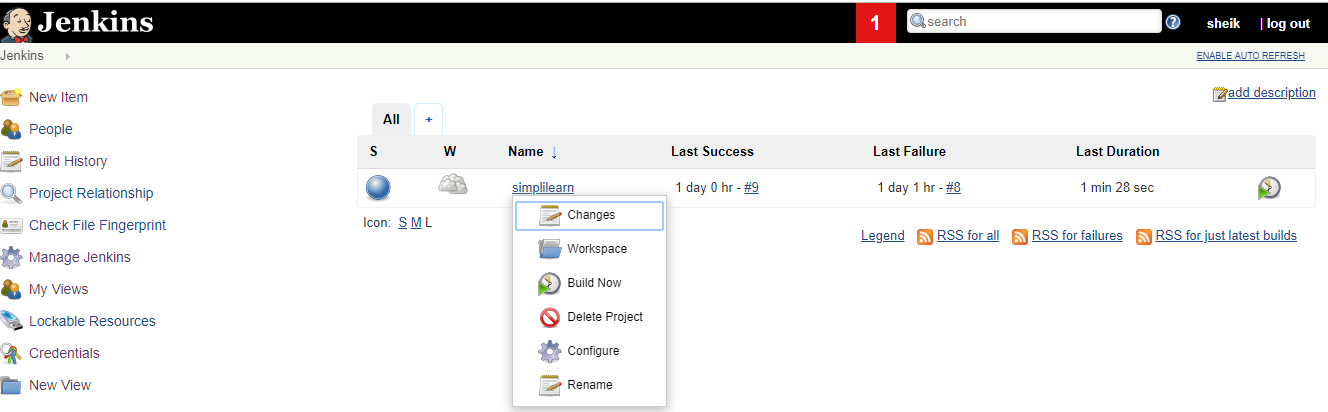
**Give Git hub name and add credentials in source code management.**



**In build section-> Select Execute shell and execute below commands:**

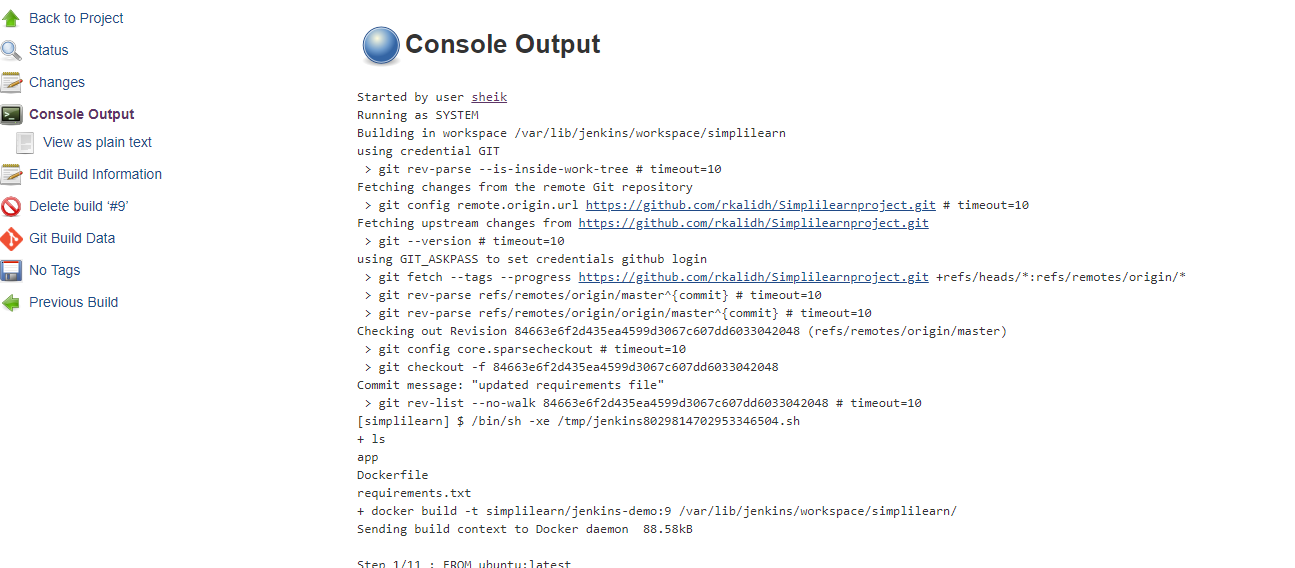


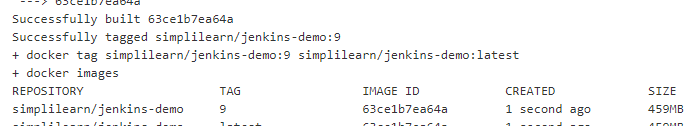
**Once completed, please save the job and click on build now:**



**You will be able to see console output as below: [ output is attached in Doc]**





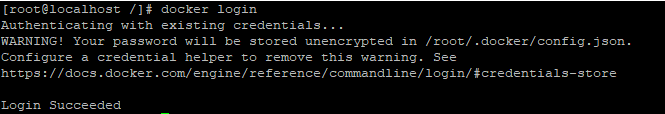


**Now run the image with container name using docker command:**

**docker run --name jenkinsintegration-new simplilearn/jenkins-demo:9**

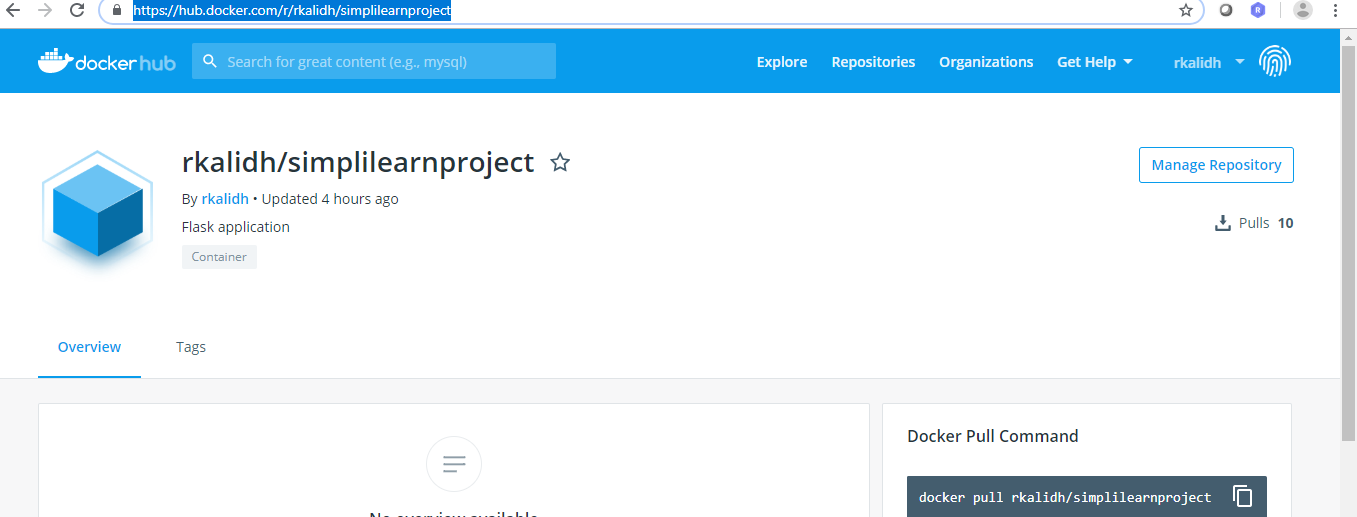


**Pus h the image to Docker hub and ensure to have docker account**



Push the image to docker hub:

Docker repository name: <https://hub.docker.com/r/rkalidh/simplilearnproject>



**docker push rkalidh/simplilearnproject**

**Verify whether you are able to login in container and execute some commands.**

**docker exec –it Jenkinsintegration-new /bin/bash**

