

**Course-End Project**

**Dockerized CI/CD pipelines with Docker image vulnerabilities scan**

**Steps to be followed:**

* + 1. Access Jenkins application
    2. Create Jenkins pipeline to perform CI CD for a Docker container
    3. Create Docker Hub Credentials and other necessary pre-requisites before running build.
    4. Execute Jenkins Build
    5. Access Deployed application on Docker container

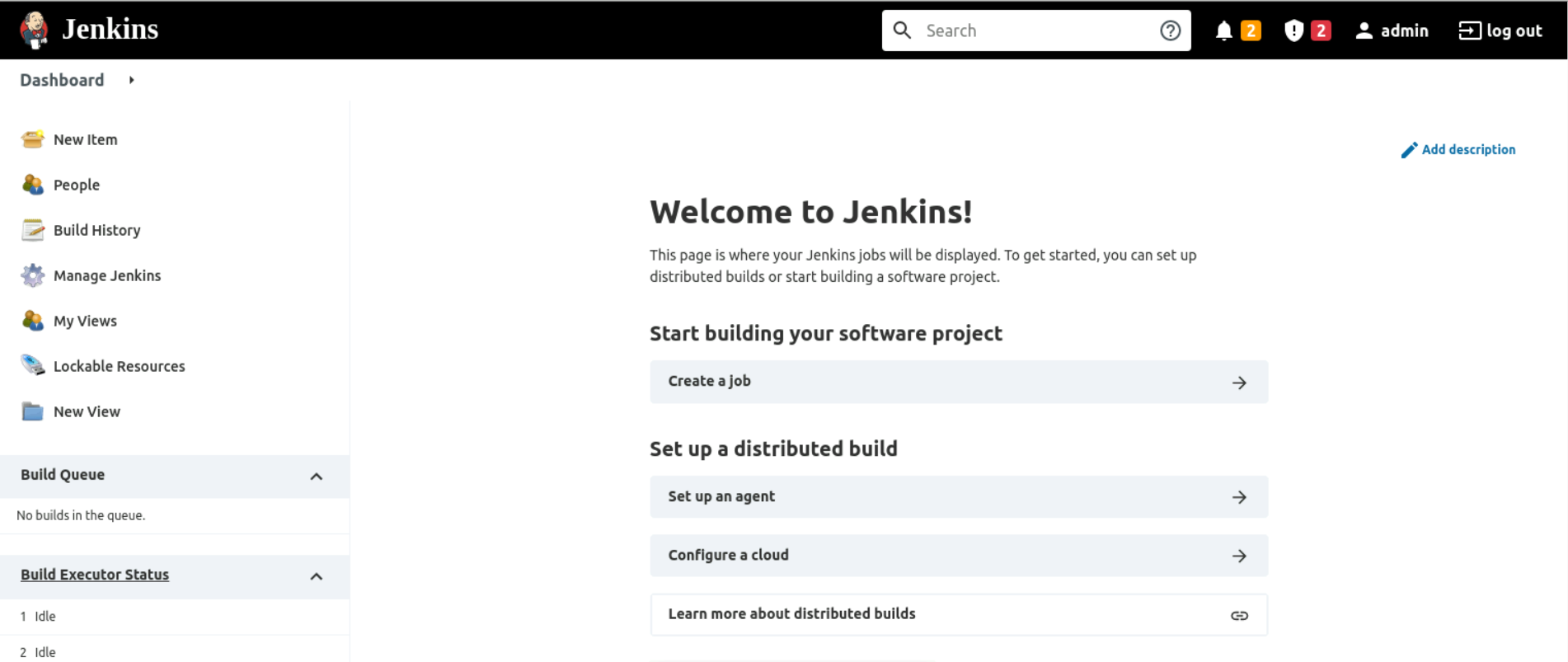
**Step 1: Access Jenkins application**

* 1. Open new browser and access Jenkins application deployed on a cloud instance with credentials as below:

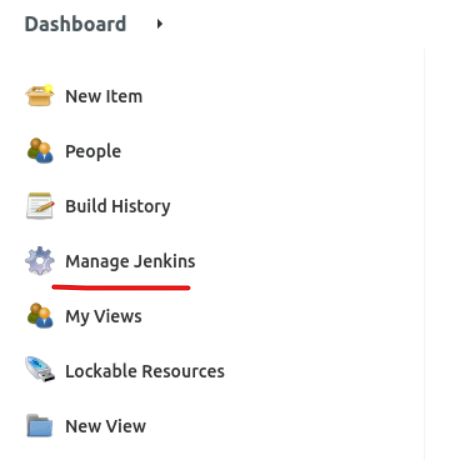
<http://localhost:8080/>

Username: admin

Password: admin



1.2 Once logged in Jenkins please make sure you have admin rights to create or manage Jenkin pipeline and Manage Jenkins access should be enabled for the specific user:

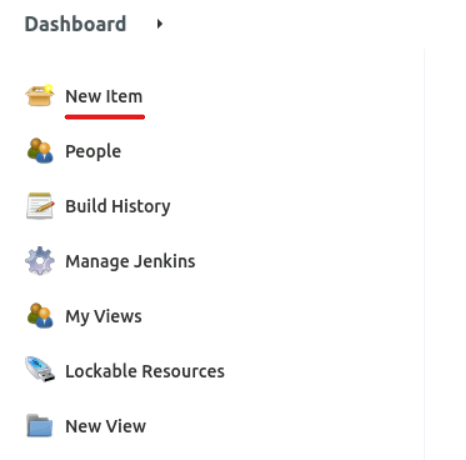


1.3 Before proceeding further open terminal and then install below package on Jenkins instance:

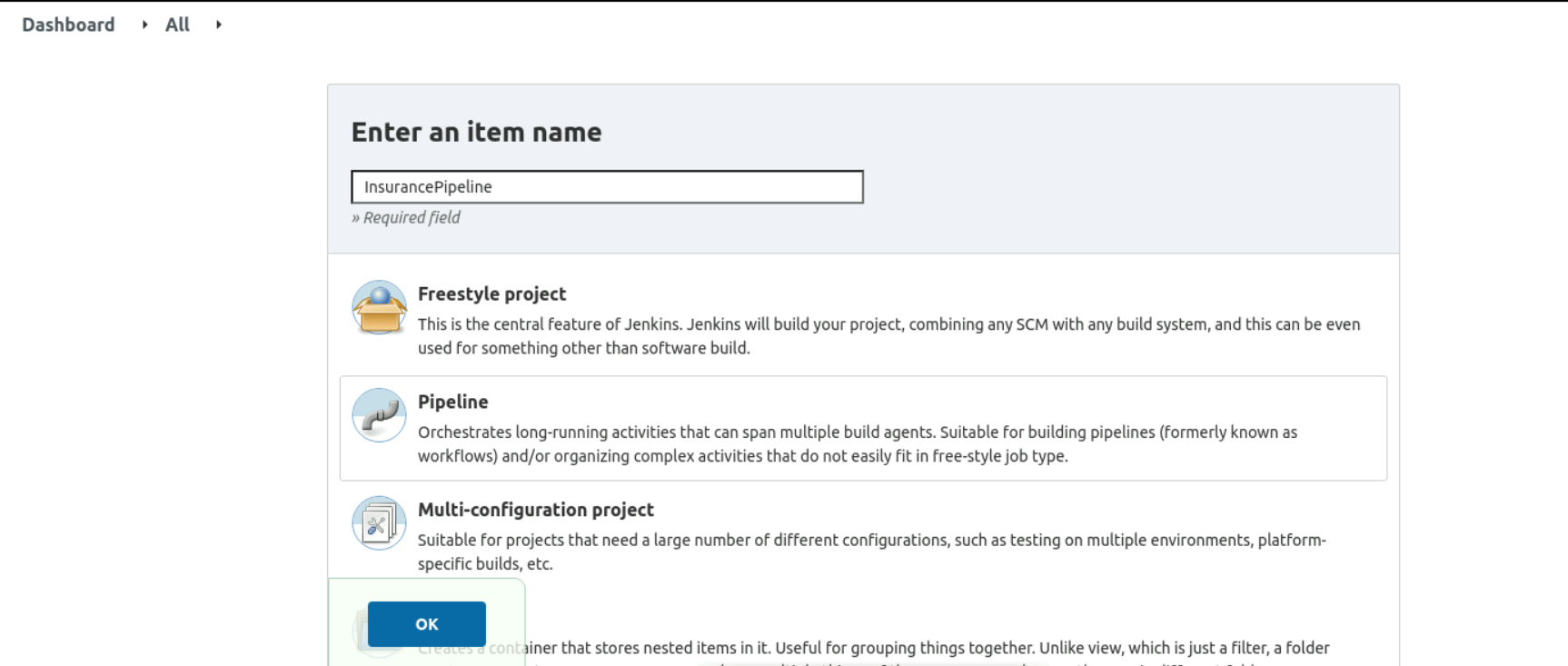
apt update && apt install docker.io

**Step 2: Create Jenkins pipeline to perform CI CD for a Docker container**

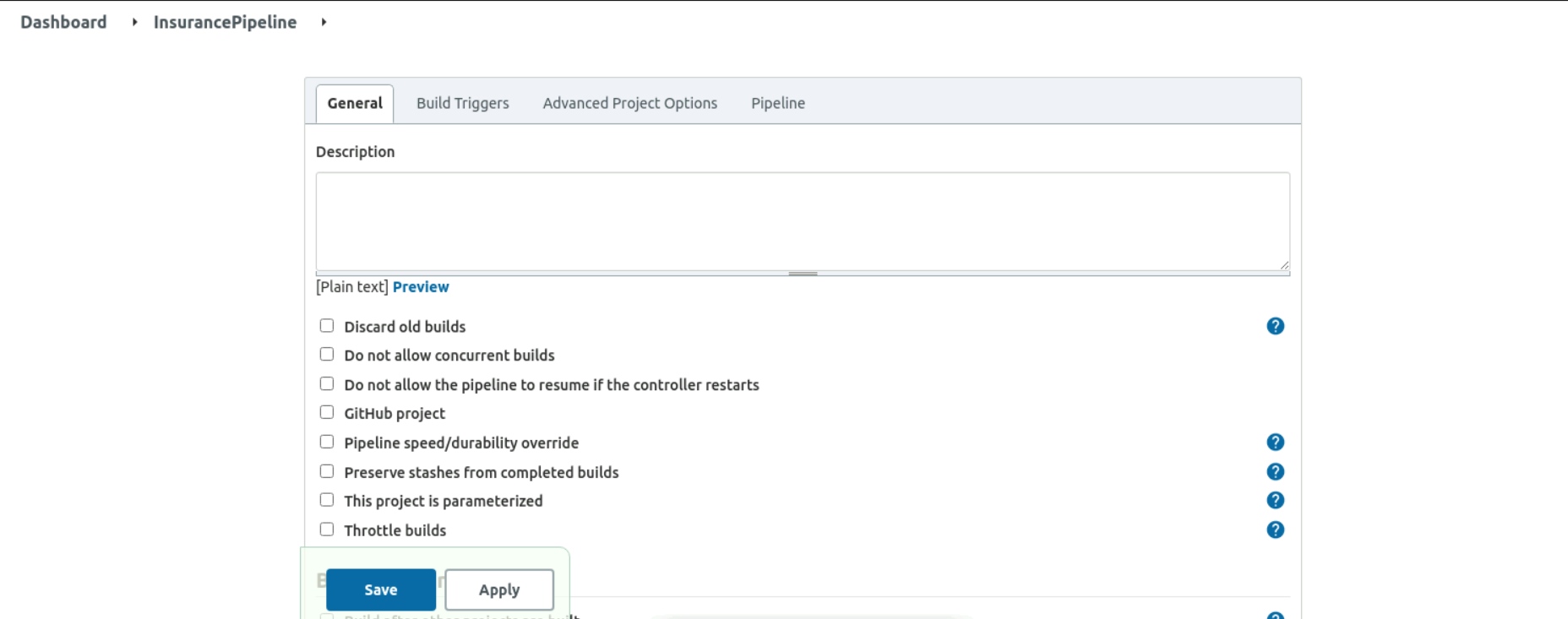
2.1 Access Jenkins application and click on New Item to create a new Jenkins job:



2.2 Select desired Jenkins pipeline job type and fill in job name as per project requirement:

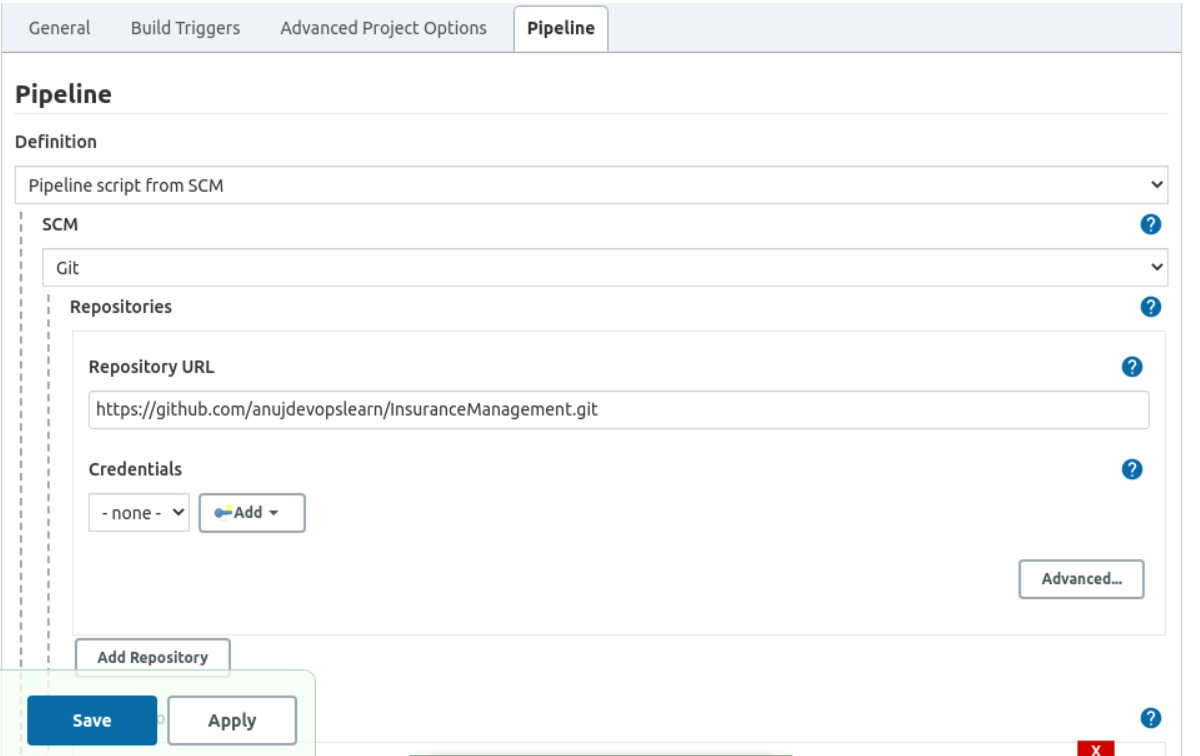


2.3 Once Clicked on Ok you will be navigated to Jenkins job configuration page where we can provide Jenkins job details such as parameters, Jenkins pipeline configuration etc.

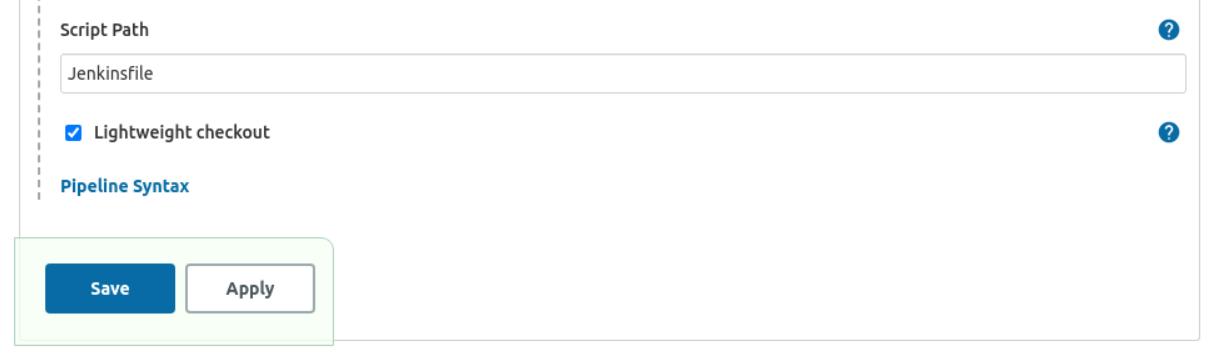


2.4 To execute Jenkins pipeline please fork below GitHub Repository and use your GitHub repository to configure Jenkins pipeline location.

https://github.com/anujdevopslearn/InsuranceManagement.git

****

**2.6 In above code repository we are using default Jenkins pipeline name but we can modify in case we are using a different pipeline name**:

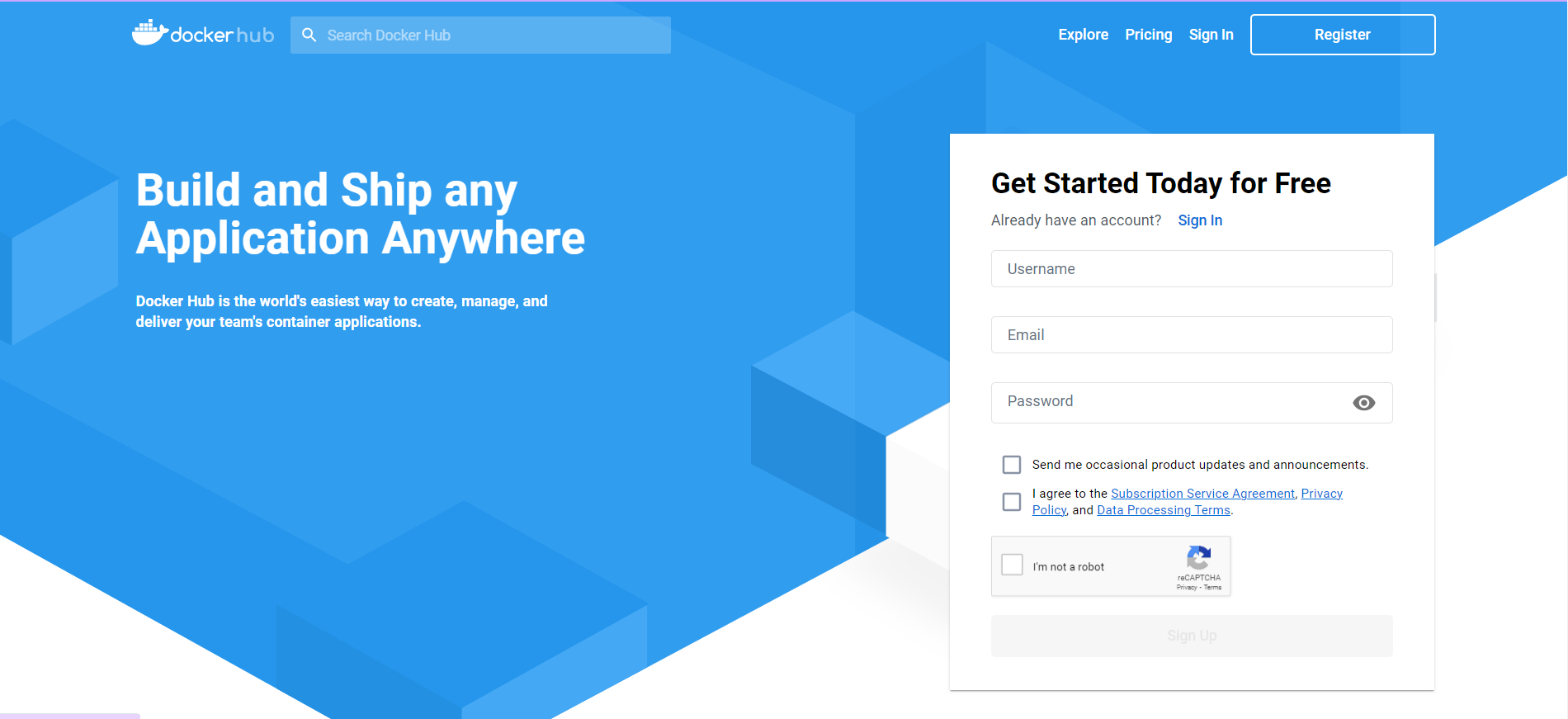


**Step 3: Create Docker Hub Credentials and other necessary pre-requisites before running build.**

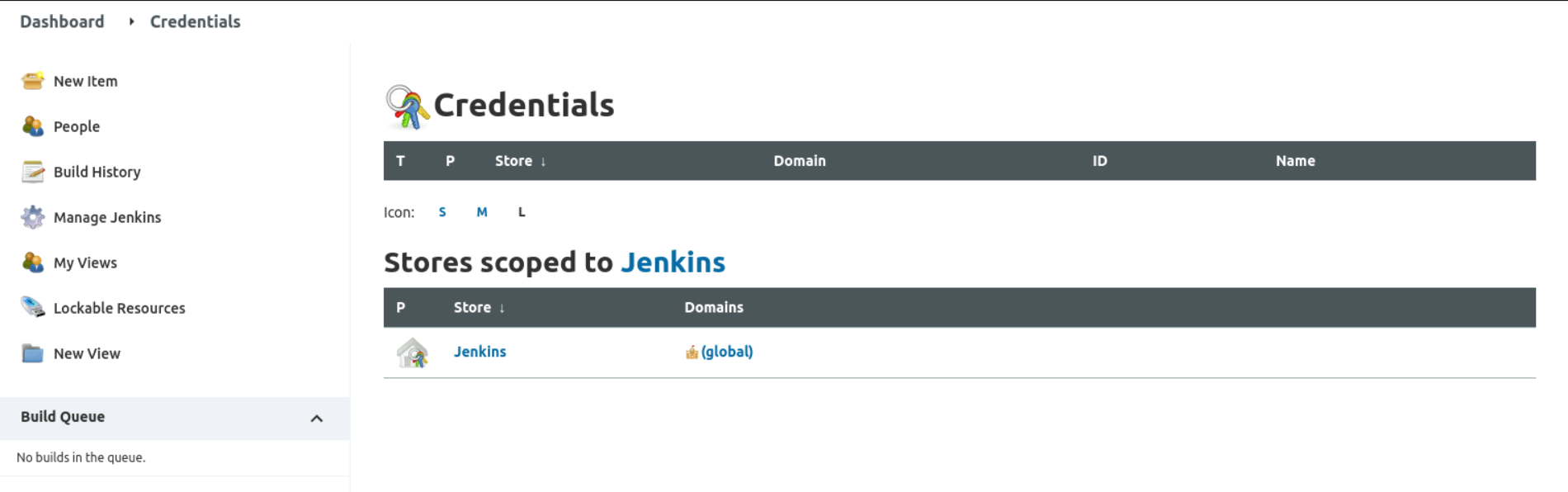
3.1 Once Jenkins job configuration are done, we have to create a Docker credential which will be used to push Docker image to Docker hub:

3.2 Navigate to Docker hub and create a new free account and keep Docker hub username and password with you before configuring Jenkins credential.

[**https://hub.docker.com/**](https://hub.docker.com/)

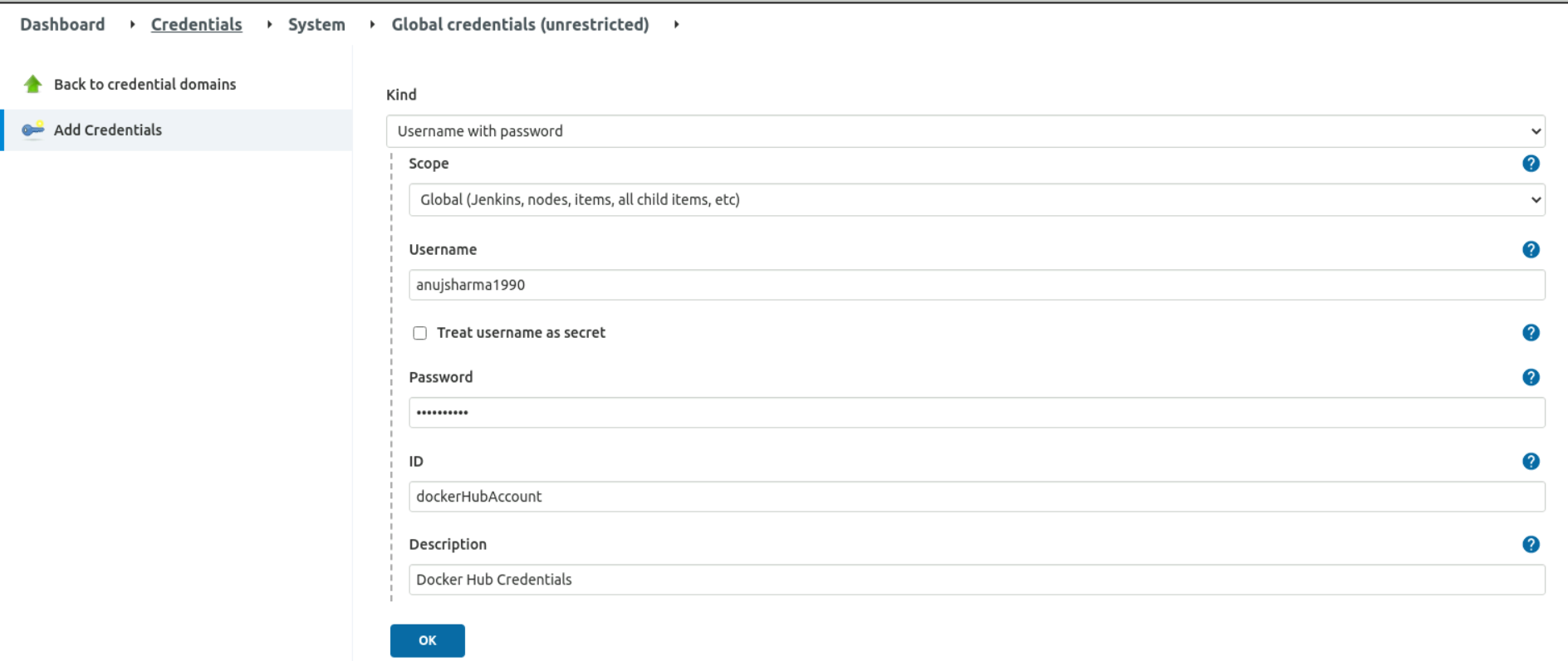
****

**3.3 Once Docker hub account is created, navigate to Manage Jenkins 🡪 Manage Credentials and then select global domain.**

****

**3.4 Click on new Credential by clicking on Add Credentials to create a new Docker hub credentials as per below details:**

**Note: Credential id in below screenshot should match what you have mentioned in Jenkins pipeline.**

****

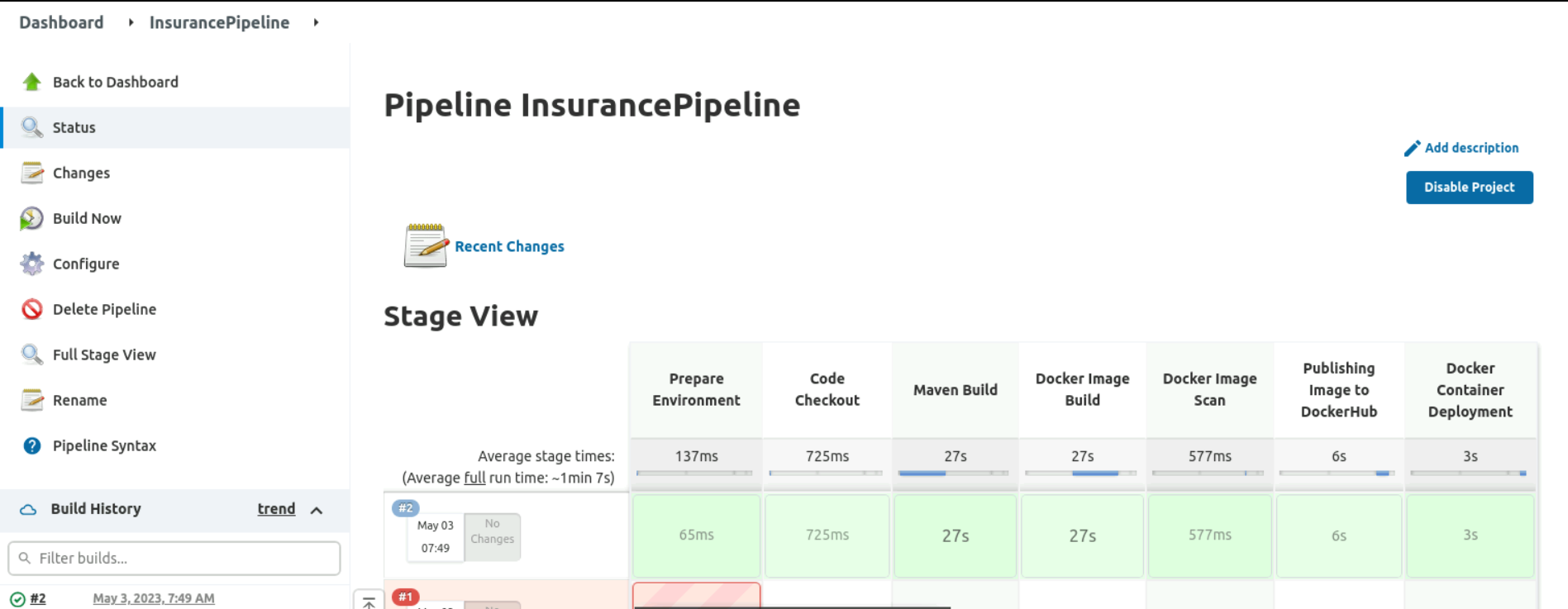
**3.5 Next provide full access to Docker Sock file using below command:**

**chmod 777 /var/run/docker.sock**

**Once done save both tool configuration and proceed with running Jenkins build.**

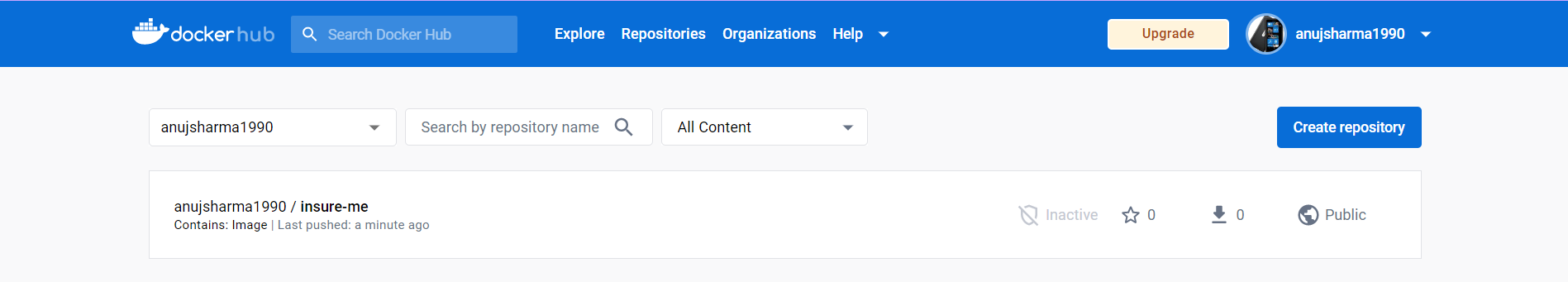
**Step 4: Execute Jenkins Build**

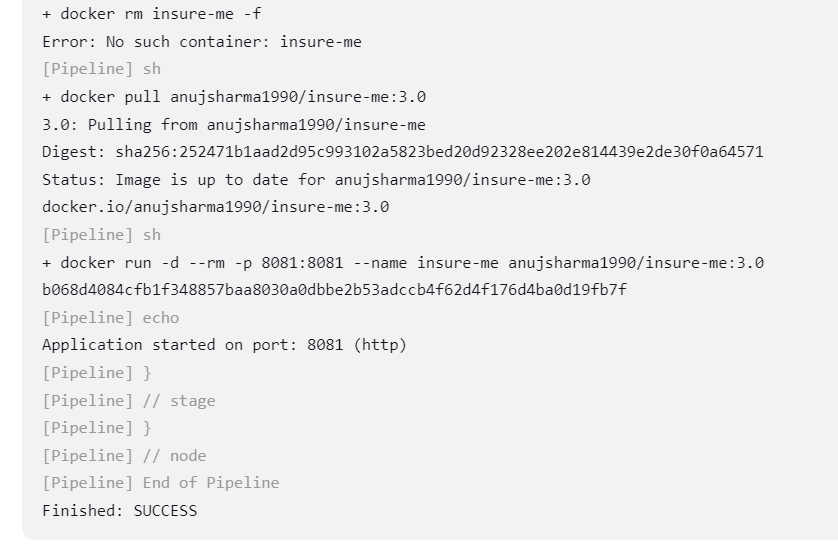
4.1 Navigate to Jenkins job created in step 2.6 and click on Build now to start running build for Jenkins job created.



Note: First time build may take more time than expected since lot of maven dependencies will be downloaded and cache locally first time. Subsequent executions will not take much time later.

Once Jenkins build is completed validate if Docker image really gets uploaded to Docker hub or not.





**Step 5: Access Deployed application on Docker container**

5.1 Once Docker container is deployed using server IP address and port 8081 to access application.

5.2 Use system default browser to access application deployed on Docker container.



