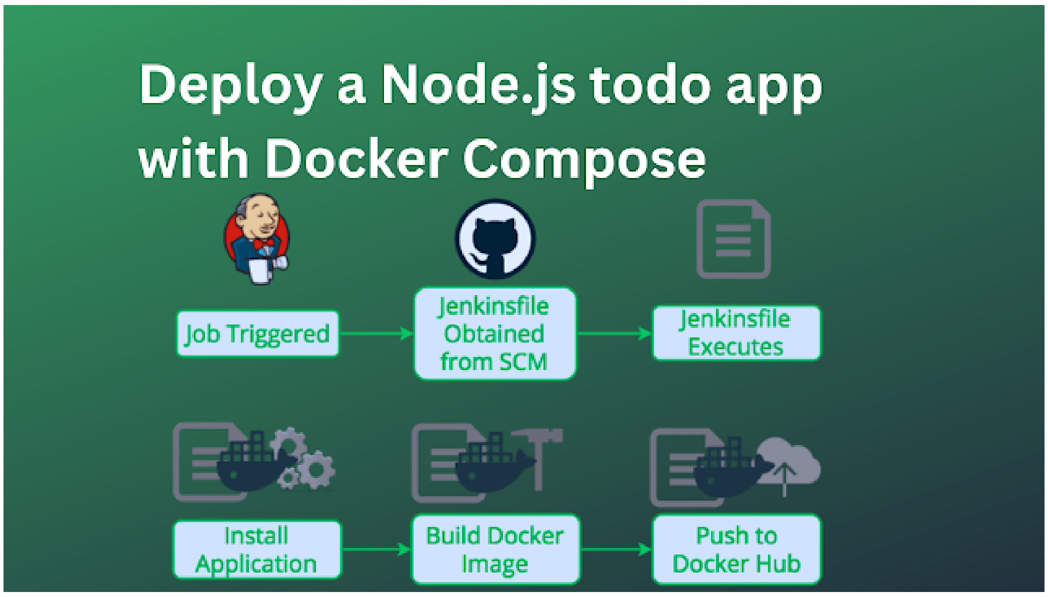
Step-by-Step Guide to Deploying a React-Django App with Jenkins Pipeline, Docker, and Push DockerHub

March 30, 2023



Introduction

In this article, we will go through the steps to deploy a demo React Django app using Jenkins Groovy Pipeline, Docker, and pushing it to DockerHub. We will start with setting up the project and creating the necessary files, and then we will configure the Jenkins server to automate the deployment process. Finally, we will push the Docker image to DockerHub.

Prerequisites

- Jenkins server
- Docker
- Git

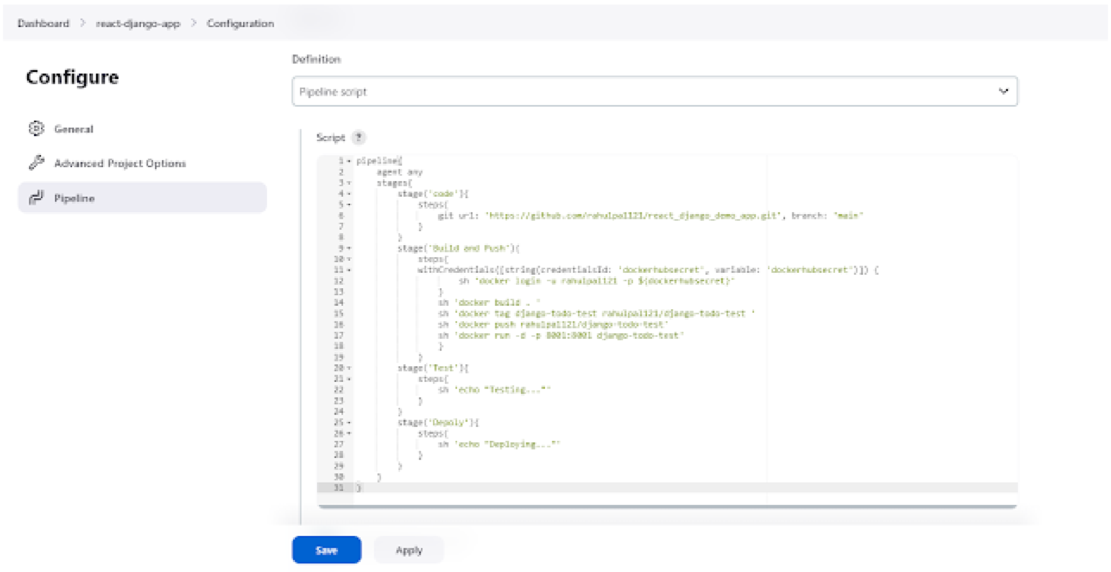
Setting up the project

1. Create a new directory and navigate to it in your terminal.

2. Run the "git init" command to initialize a new Git repository:

3. Clone the app from the git repository.

4 Now install and setup Jenkins with Java which required for Jenkins.

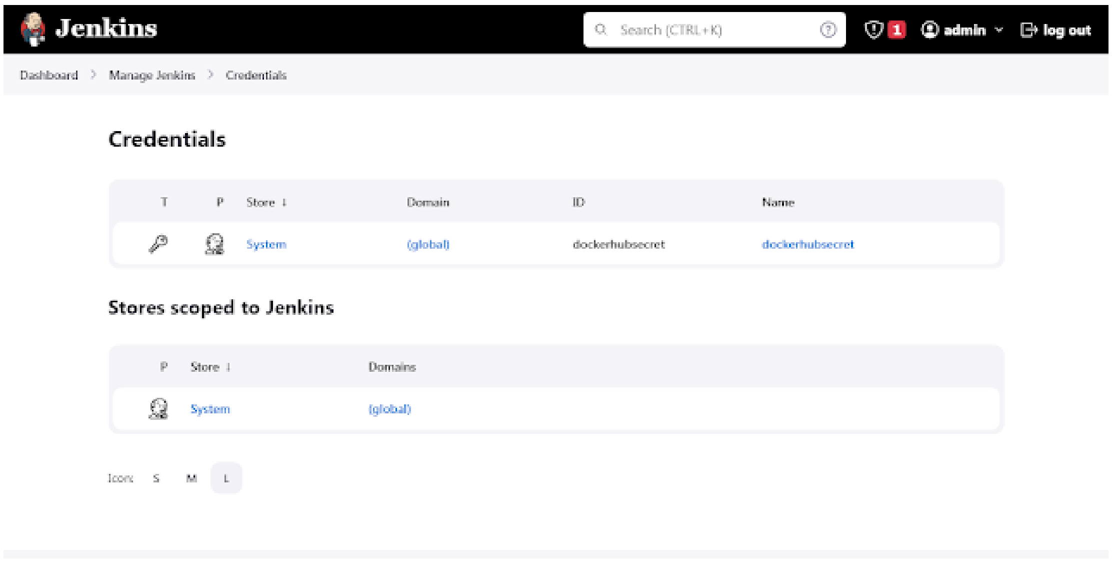


Definition

Pipeline script

Script

```
pipeline{
    agent any
    stages{
        stage('code'){
            steps{
                git url: 'https://github.com/rahulpal121/react_django_demo_app.git', branch: 'main'
            }
        }
        stage('Build and Push'){
            steps{
            withCredentials([string(credentialsId: 'dockerhubsecret', variable: 'dockerhubsecret')]) {
                    sh 'docker login -u rahulpal121 -p ${dockerhubsecret}'
                }
                sh 'docker build . '
                sh 'docker tag django-todo-test rahulpal121/django-todo-test '
                sh 'docker push rahulpal121/django-todo-test'
                sh 'docker run -d -p 8001:8001 django-todo-test'
                }
            }
        stage('Test'){
            steps{
                sh 'echo "Testing..."'
            }
        }
        stage('Depoly'){
            steps{
                sh 'echo "Deploying..."'
            }
        }
    }
}
```

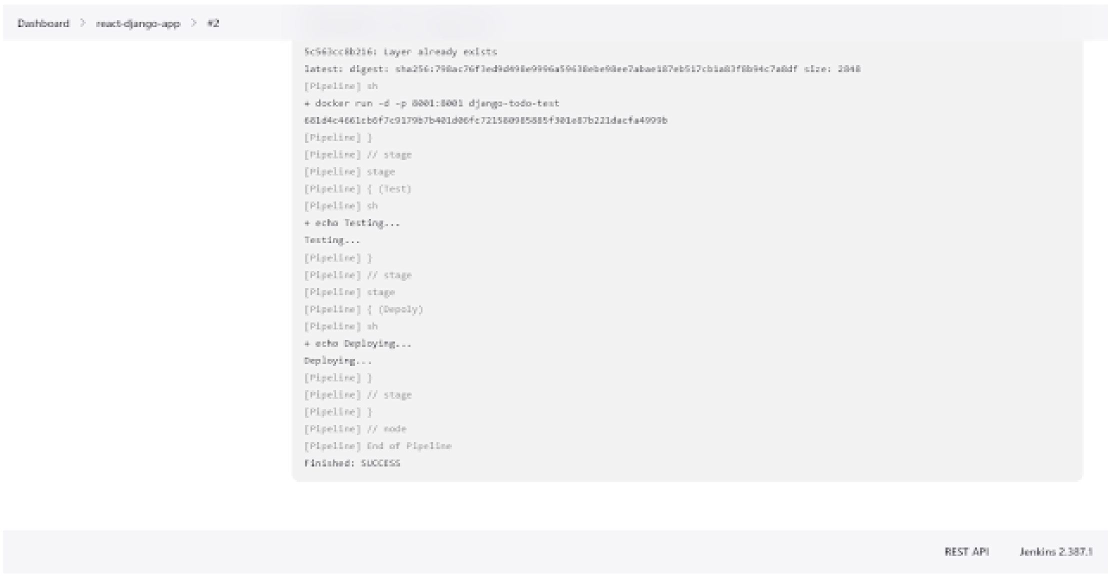


Credentials

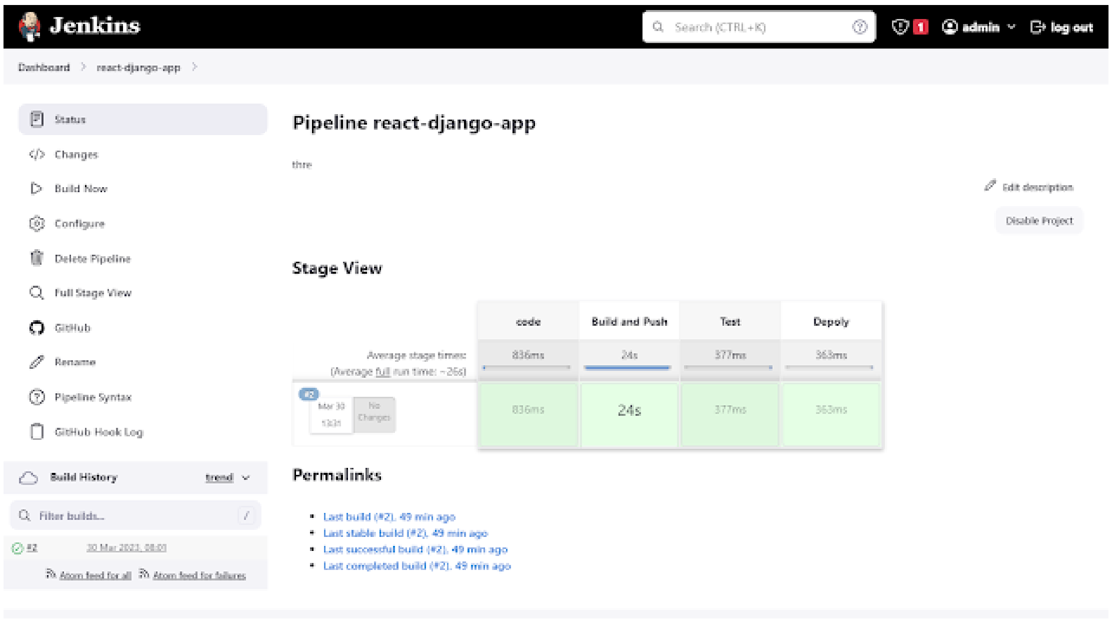
T	P	Store ↓	Domain	ID	Name
		System	(global)	dockerhubsecret	dockerhubsecret

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)



```
5c563cc8b216: Layer already exists
latest: digest: sha256:798ac76f3ed9d498e9996a59638ebe98ee7abae187eb517cb1a83f8b94c7a8df size: 2048
[Pipeline] sh
+ docker run -d -p 8001:8001 django-todo-test
681d4c4661cb6f7c9179b7b401d06fc721580985885f301e87b221dacfa4999b
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] sh
+ echo Testing...
Testing...
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Depoly)
[Pipeline] sh
+ echo Deploying...
Deploying...
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Pipeline react-django-app

thre

Stage View

	code	Build and Push	Test	Depoly
Average stage times: (Average full run time: ~26s)	836ms	24s	377ms	363ms
#2 Mar 30 13:31 No Changes	836ms	24s	377ms	363ms

#2 30 Mar 2023, 08:01

Atom feed for all   Atom feed for failures

Permalinks

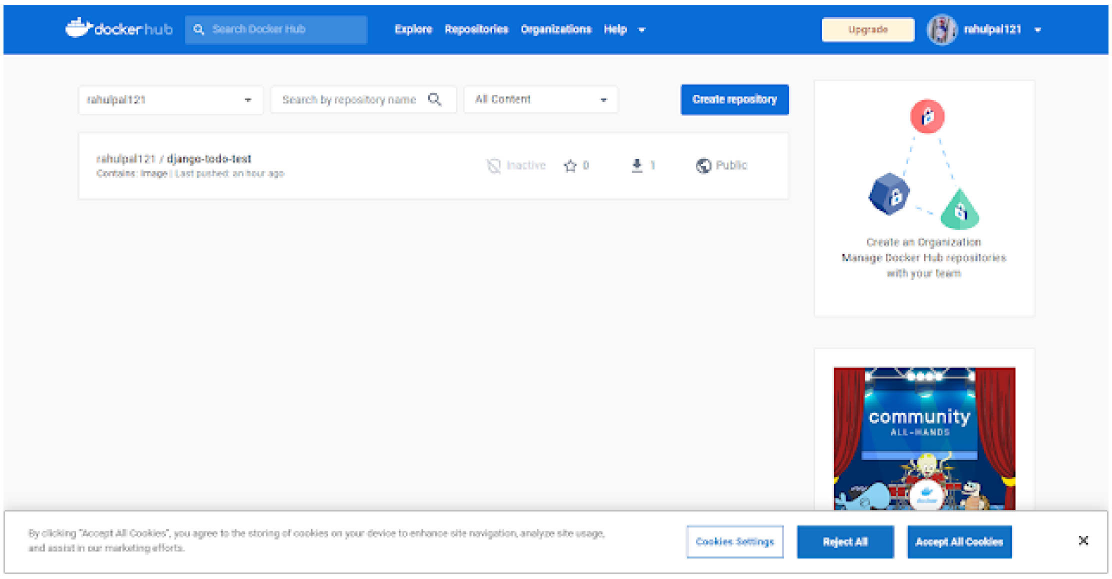
- Last build (#2), 49 min ago
- Last stable build (#2), 49 min ago
- Last successful build (#2), 49 min ago
- Last completed build (#2), 49 min ago

Jenkins Pipeline

To create a Jenkins pipeline using Groovy syntax with secret text credentials and a Docker push task, follow these steps:

1. Open the Jenkins dashboard and click on the "New Item" button.
2. Name your pipeline and select "Pipeline" as the project type.

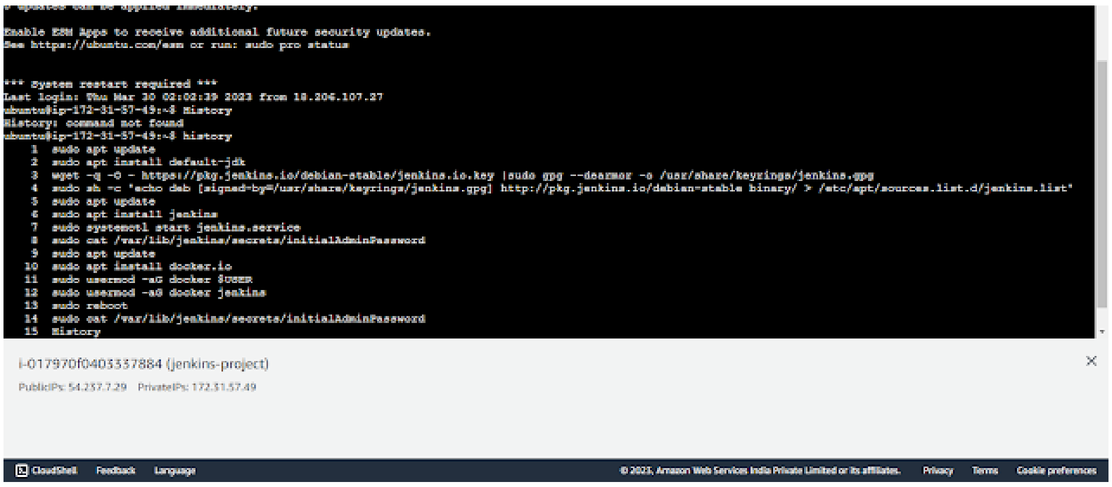
4. Write your pipeline script in the editor, including stages, steps, and any necessary parameters or variables.
5. Define your credentials as secret text in the Jenkins Credentials Manager.
6. In your pipeline script, use the withCredentials block to access your secret text credential by ID.
7. Add a dockerBuildAndPush step to your script to build and push your Docker image to a container registry.
8. Test your script by clicking the "Pipeline Syntax" button and running a syntax check.
9. Save your script and run the pipeline by clicking the "Build Now" button on the job page.



rahulpal121 / django-todo-test
Contains: Image | Last pushed: an hour ago

Create an Organization
Manage Docker Hub repositories with your team

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

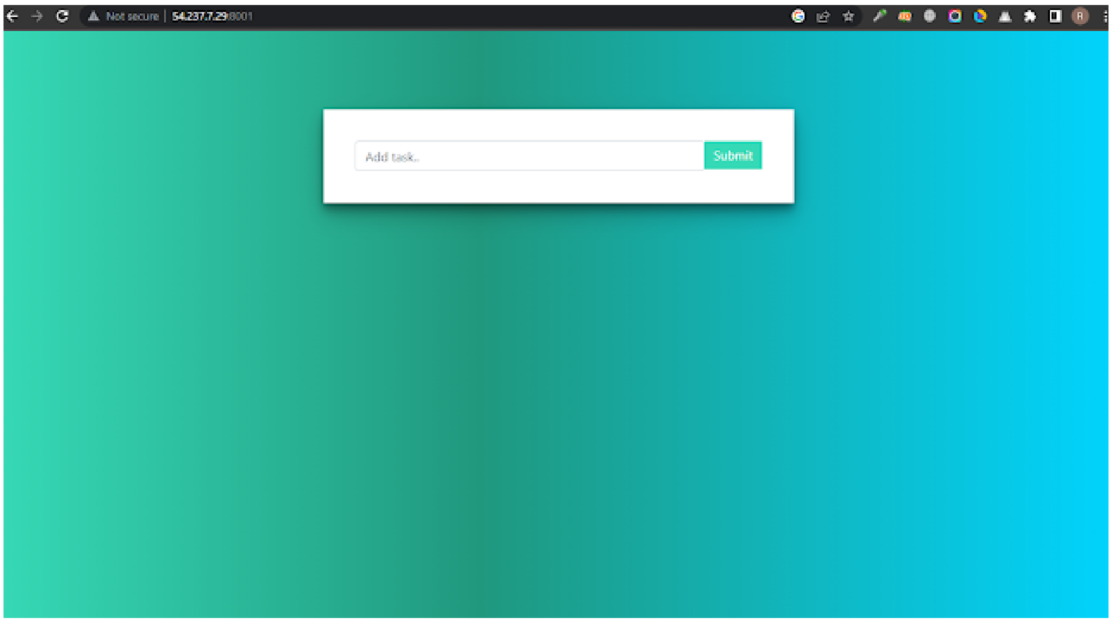


```
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


*** System restart required ***
Last login: Thu Mar 30 02:02:39 2023 from 18.206.107.27
ubuntu@ip-172-31-57-49:~$ History
History: command not found
ubuntu@ip-172-31-57-49:~$ history
    1  sudo apt update
    2  sudo apt install default-jdk
    3  wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |sudo gpg --dearmor -o /usr/share/keyrings/jenkins.gpg
    4  sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins.gpg] http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
    5  sudo apt update
    6  sudo apt install jenkins
    7  sudo systemctl start jenkins.service
    8  sudo cat /var/lib/jenkins/secrets/initialAdminPassword
    9  sudo apt update
   10  sudo apt install docker.io
   11  sudo usermod -aG docker $USER
   12  sudo usermod -aG docker jenkins
   13  sudo reboot
   14  sudo cat /var/lib/jenkins/secrets/initialAdminPassword
   15  History
```

i-017970f0403337884 (jenkins-project)

PublicIPs: 54.237.7.29 PrivateIPs: 172.31.57.49

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences



By following these steps and using the withCredentials block to securely access your secret text credentials, as well as including a Docker push task in your pipeline script, you can create a Jenkins pipeline using Groovy syntax to automate your software delivery process and deploy your application to a container registry.

Conclusion

to DockerHub. We started with setting up the project and creating the necessary files, and then we configured the Jenkins server to automate the deployment process. Finally, we pushed the Docker image to DockerHub. By following these steps, you can deploy your own React Django app using Jenkins and Docker