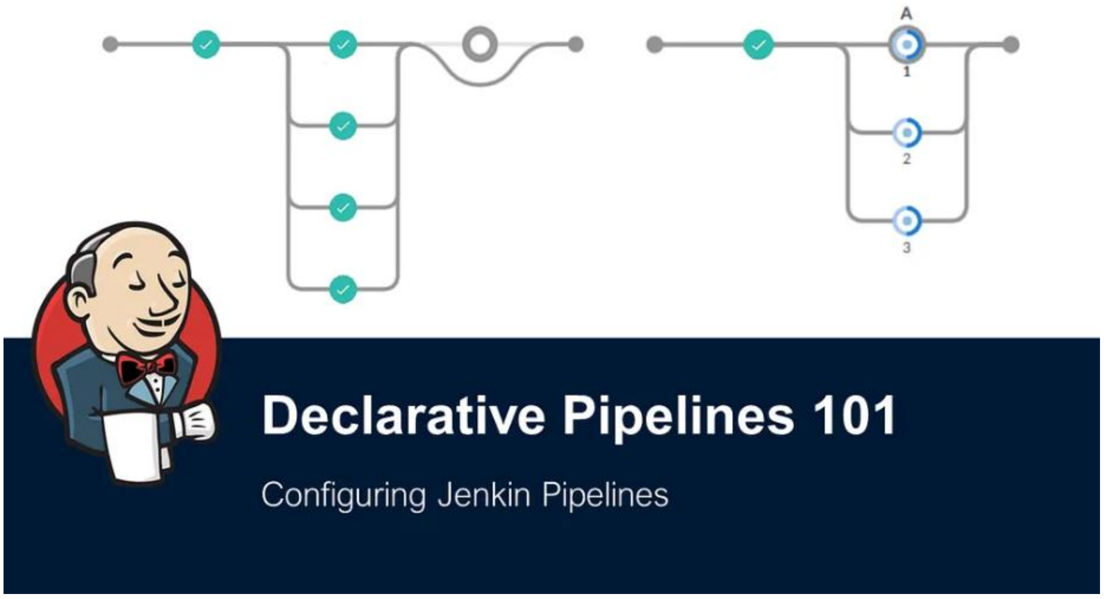
Day 27 : Jenkins Declarative Pipeline

This is [#90DaysofDevops](#) challenge under the guidance of [Shubham Londhe](#).

Day 26 TASK

Check this for task:

https://github.com/LondheShubham153/90DaysOfDevOps/blob/master/2023/day27/tasks.md



Docker Build and Run

docker build — you can use `sh 'docker build . -t <tag>'` in your pipeline stage block to run the docker build command. (Make sure you have docker installed with correct permissions.

docker run: you can use `sh 'docker run -d <image>'` in your pipeline stage block to build the container.

How will the stages look

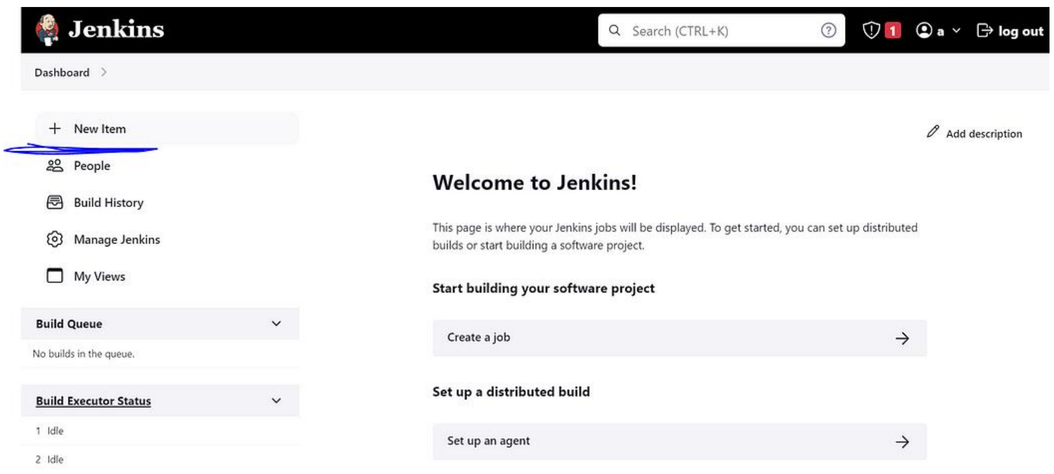
```
stages {
        stage('Build') {
            steps {
                sh 'docker build -t trainwithshubham/django-app:latest'
            }
            }
        }
```

Task-01

- Create a docker-integrated Jenkins declarative pipeline
- Use the above-given syntax using `sh` inside the stage block
- You will face errors in case of running a job twice, as the docker container will be already created, so for that do task 2.

First create an EC2 instance , install Jenkins on it. Access Jenkins using the public IP of the EC2 instance and the 8080 port.

Once you are done with the Jenkins installation and are able to access the same on the browser then open the Jenkins Dashboard and select "New Item".



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

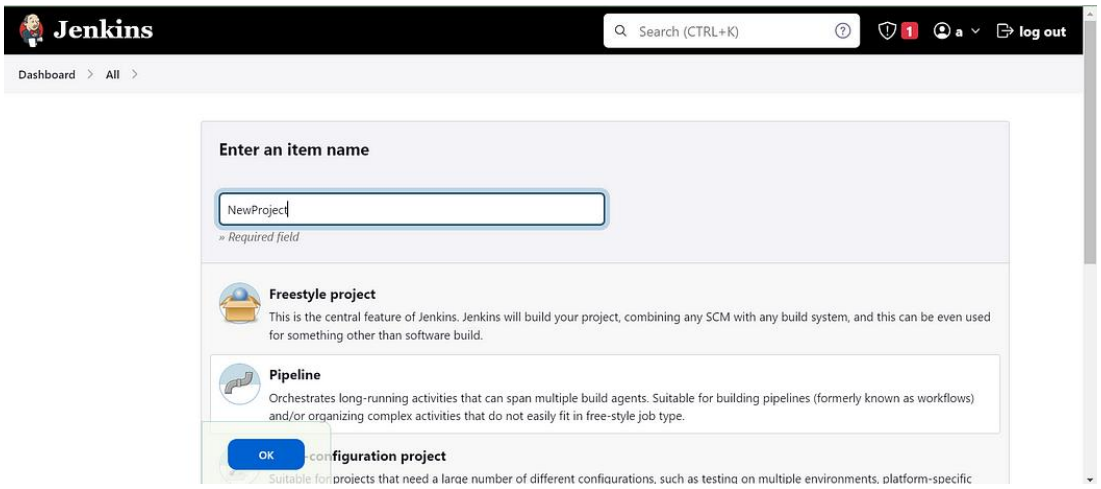
Set up a distributed build

No builds in the queue.

1 Idle

2 Idle

Here you need to add the Project Name and then select the project type as "pipeline" in this case because we will be creating the JenkinnFile further.



Enter an item name

NewProject

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

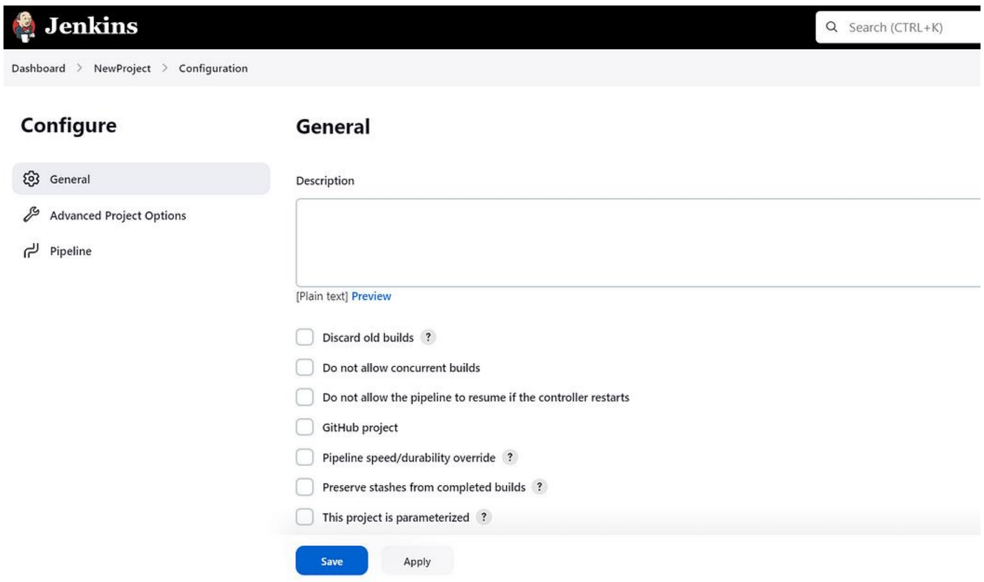
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

OK

configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific

Click on "OK".

Now we will land into the "Project Configuration" section.,



General

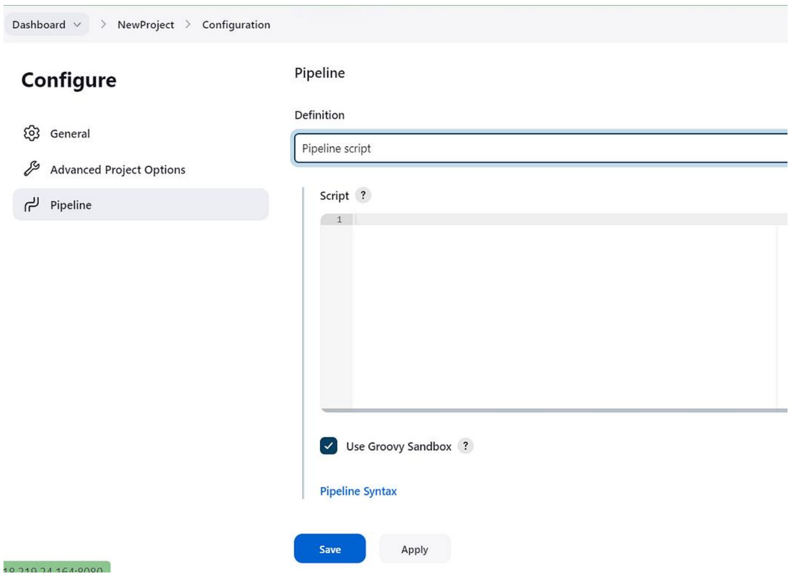
Description

[Plain text] Preview

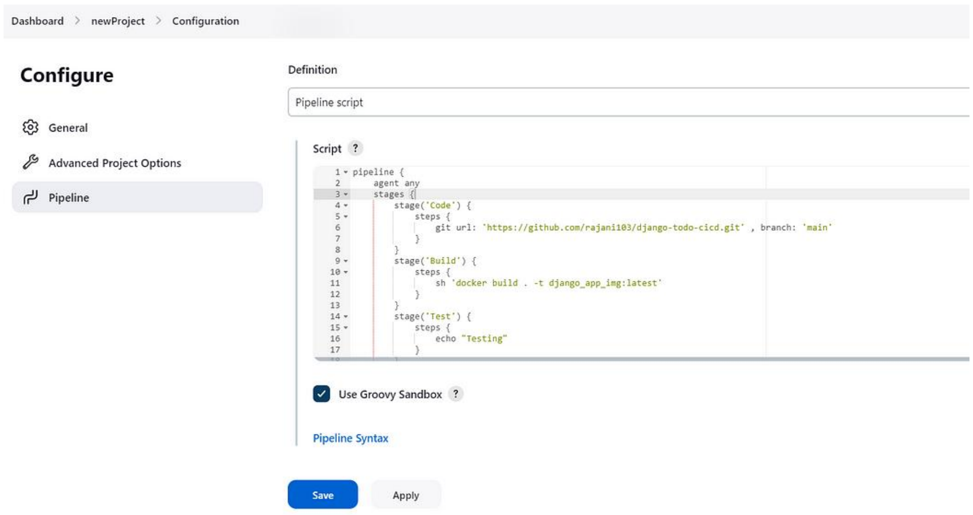
- [ ] Discard old builds ?
- [ ] Do not allow concurrent builds
- [ ] Do not allow the pipeline to resume if the controller restarts
- [ ] GitHub project
- [ ] Pipeline speed/durability override ?
- [ ] Preserve stashes from completed builds ?
- [ ] This project is parameterized ?

Save Apply

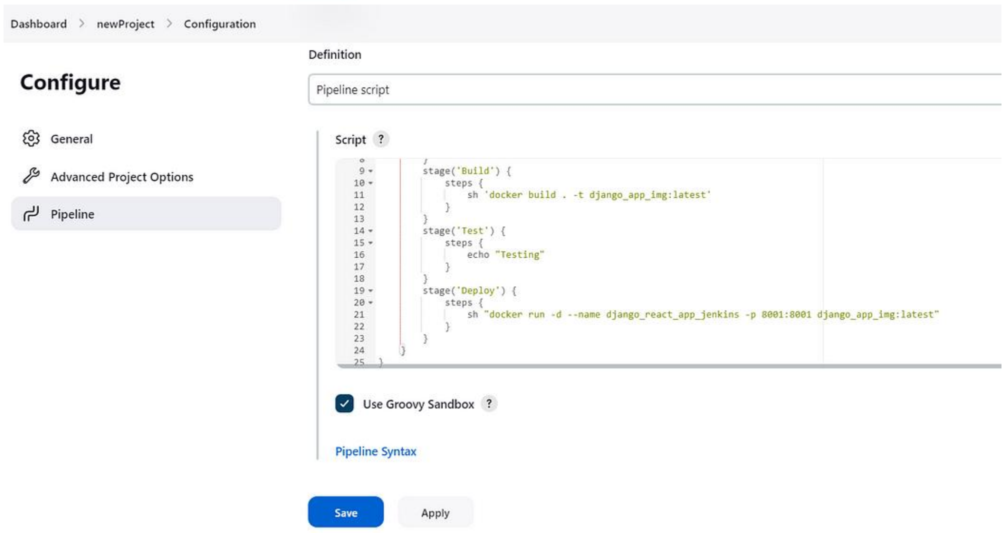
Here you need to go to the "pipeline" section and there you need to select the "Pipeline script" in definition.



Now we will write a basic Pipeline Script for for react-django app.



```
pipeline {
    agent any
    stages {
        stage('Code') {
            steps {
                git url: 'https://github.com/rajani103/django-todo-cicd.git' , branch: 'main'
            }
        }
        stage('Build') {
            steps {
                sh 'docker build . -t django_app_img:latest'
            }
        }
        stage('Test') {
            steps {
                echo "Testing"
            }
```

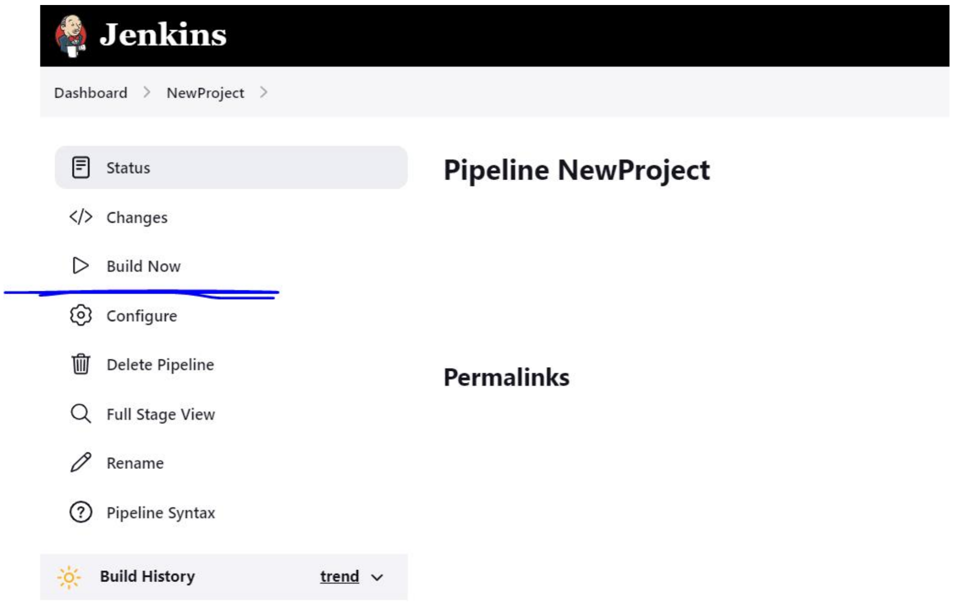


```
        stage('Build') {
            steps {
                sh 'docker build . -t django_app_img:latest'
            }
        }
        stage('Test') {
            steps {
                echo "Testing"
            }
        }
        stage('Deploy') {
            steps {
                sh "docker run -d --name django_react_app_jenkins -p 8001:8001 django_app_img:latest"
            }
        }
    }
}
```

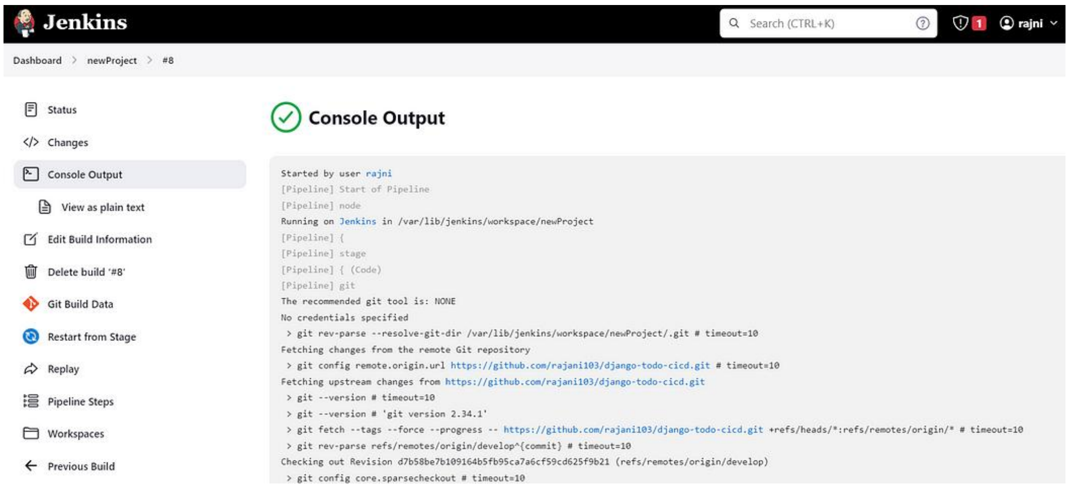
```
pipeline {
    agent any
    stages {
        stage('Code') {
            steps {
                git url: 'https://github.com/rajani103/django-todo-cicd.git' , branch: 'main'
            }
        }
        stage('Build') {
            steps {
                sh 'docker build . -t django_app_img:latest'
            }
        }
        stage('Test') {
            steps {
                echo "Testing"
            }
        }
        stage('Deploy') {
            steps {
                sh "docker run -d --name django_react_app_jenkins -p 8001:8001 django_app_img:latest"
            }
        }
    }
}
```

Click on the **save** button to save the Pipeline.

Now we can start the build process by manually clicking on the build now tab present.



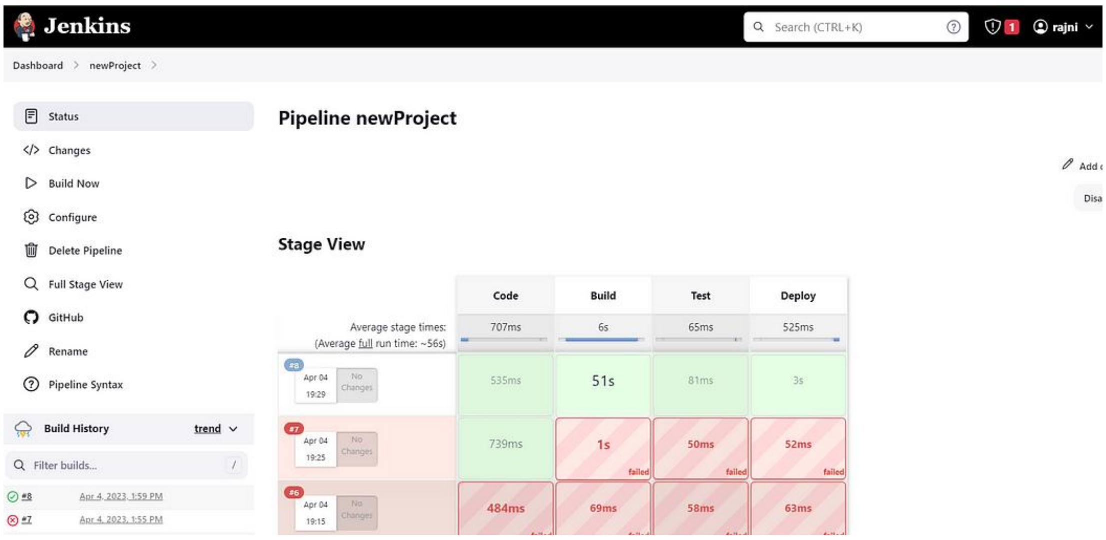
Once the build is complete you can check the output of the build by clicking on the "Console Output" tab present there.



```
Started by user rajni
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/newProject
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Code)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/newProject/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/rajani103/django-todo-cicd.git # timeout=10
Fetching upstream changes from https://github.com/rajani103/django-todo-cicd.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git fetch --tags --force --progress -- https://github.com/rajani103/django-todo-cicd.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/develop^{commit} # timeout=10
Checking out Revision d7b58be7b109164b5fb95ca7a6cf59cd625f9b21 (refs/remotes/origin/develop)
 > git config core.sparsecheckout # timeout=10
```

You can see here the pipeline is running successfully.

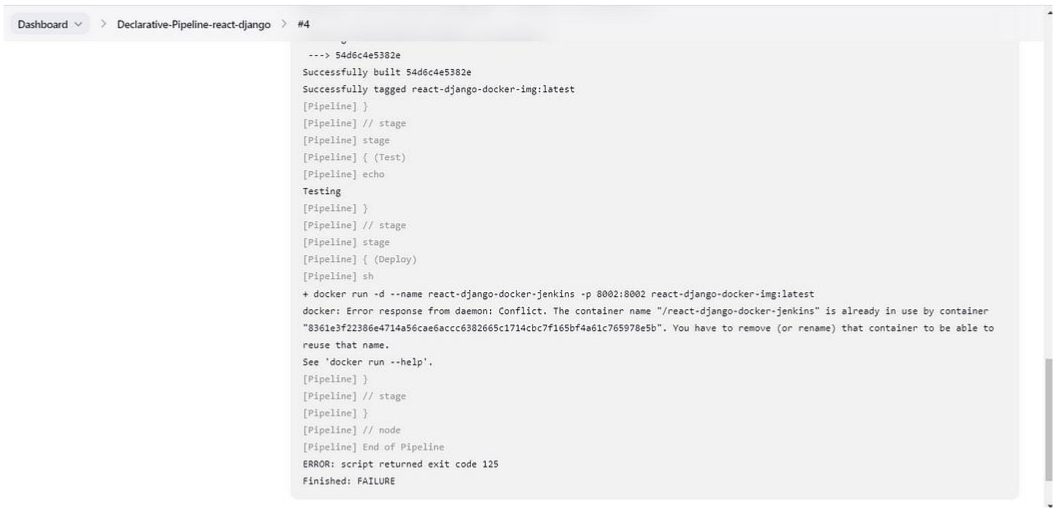
Now you check the multi stage view by clicking on the “Full Stage View” in the project main page.



	Code	Build	Test	Deploy
Average stage times: (Average full run time: ~56s)	707ms	6s	65ms	525ms
#8 Apr 04 19:29 No Changes	535ms	51s	81ms	3s
#7 Apr 04 19:25 No Changes	739ms	1s failed	50ms failed	52ms failed
#6 Apr 04 19:15 No Changes	484ms	69ms	58ms	63ms

- You will face errors in case of running a job twice, as the docker container will be already created, so for that do task 2.

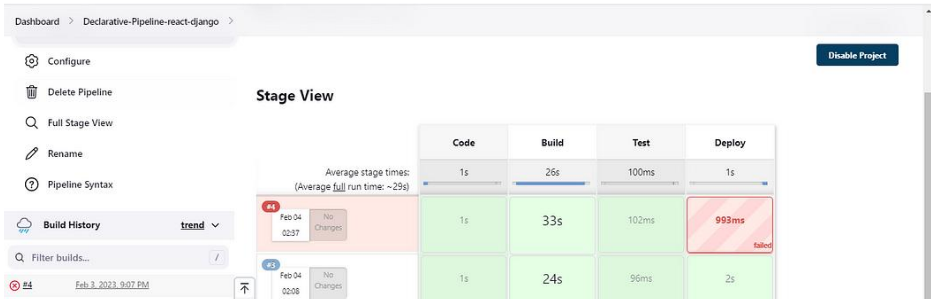
Run pipeline again by clicking on “**Build Now**” button .



Dashboard > Declarative-Pipeline-react-django > #4

```
 ---> 54d6c4e5382e
Successfully built 54d6c4e5382e
Successfully tagged react-django-docker-img:latest
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] echo
Testing
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] sh
+ docker run -d --name react-django-docker-jenkins -p 8002:8002 react-django-docker-img:latest
docker: Error response from daemon: Conflict. The container name "/react-django-docker-jenkins" is already in use by container
"8361e3f22386e4714a56cae6accc6382665c1714cbc7f165bf4a61c765978e5b". You have to remove (or rename) that container to be able to
reuse that name.
See 'docker run --help'.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 125
Finished: FAILURE
```

It will throw the error.



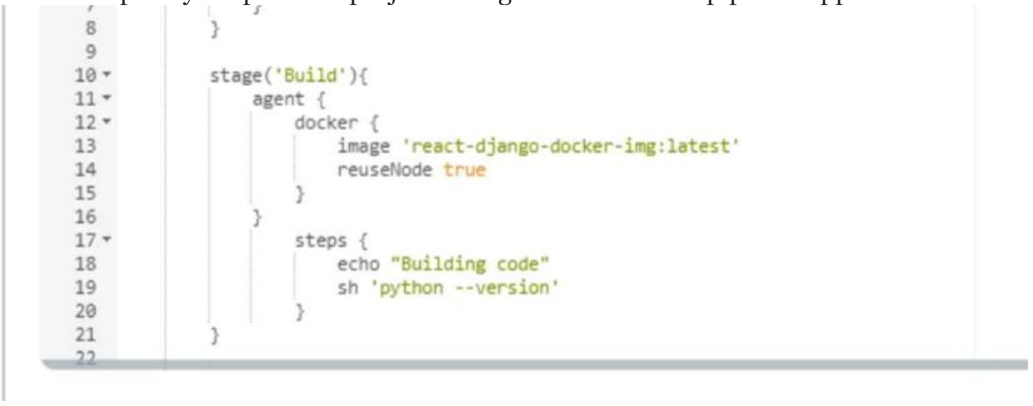
Stage View

	Code	Build	Test	Deploy
Average stage times: (Average full run time: ~29s)	1s	26s	100ms	1s
#4 Feb 04 02:37 No Changes	1s	33s	102ms	993ms failed
#3 Feb 04 02:08 No Changes	1s	24s	96ms	2s

#4 Feb 3, 2023, 9:07 PM

Task-02

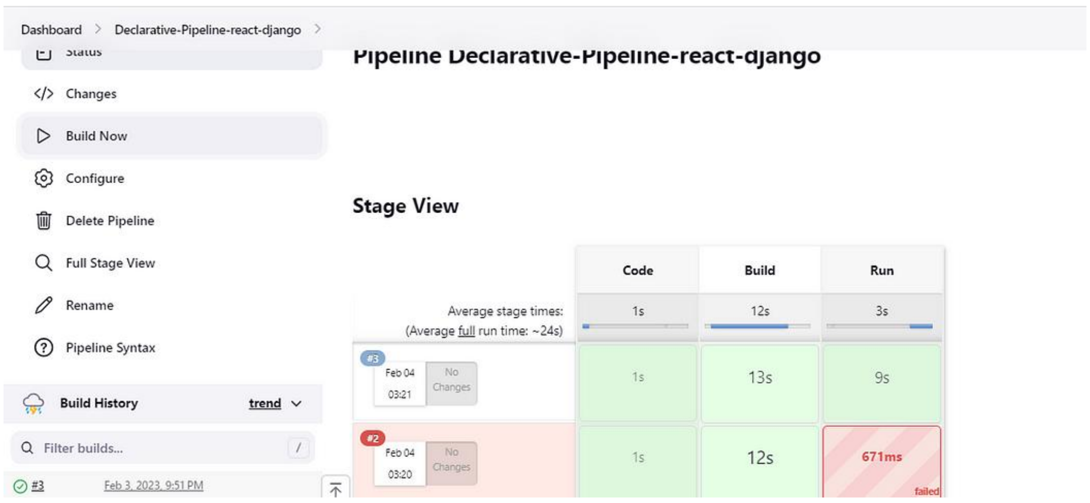
- Create a docker-integrated Jenkins declarative pipeline using the `docker` groovy syntax inside the stage block.
- You won’t face errors, you can Follow this documentation
- Complete your previous projects using this Declarative pipeline approach.



```
        }

        stage('Build'){
            agent {
                docker {
                    image 'react-django-docker-img:latest'
                    reuseNode true
                }
            }
            steps {
                echo "Building code"
                sh 'python --version'
            }
        }
```

Click on Save and then click on Build Now. After this it will work fine because we did dockerize the appliaction and then tried accessing the same.



Pipeline Declarative-Pipeline-react-django

Stage View

	Code	Build	Run
Average stage times: (Average full run time: ~24s)	1s	12s	3s
#3 Feb 04 03:21 No Changes	1s	13s	9s
#2 Feb 04 03:20 No Changes	1s	12s	671ms failed

#3 Feb 3, 2023, 9:51 PM

Please, feel free to drop any questions in the comments below. I would be happy to answer them.

If this post was helpful, please do follow and click the clap

_Thank you for reading

_Rajani