

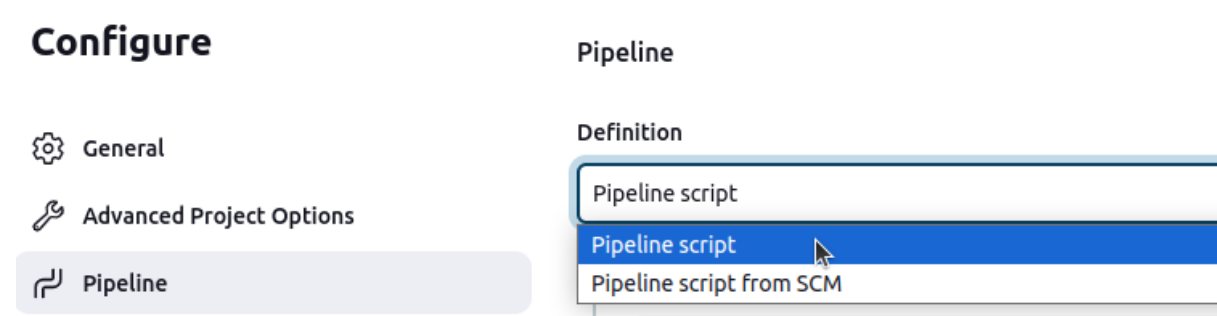
Steps to create a New Pipeline Project in Jenkins

Create New Job

- From the Jenkins dashboard, click on New Item.
- Enter a name for your job (e.g., DjangoAppPipeline).
- Select Pipeline and click OK.

Configure Pipeline Script

- In the job configuration page, scroll down to the Pipeline section.
- Choose Pipeline script from the Definition dropdown.



Enter Pipeline Script

Here's the basic pipeline script for our Django application

```
pipeline {
  agent any

  stages {
    stage('Checkout Code from GitHub') {
      steps {
        script {
          echo "Cleaning workspace before checkout"
          cleanWs() // Clean the workspace
          echo "Cloning the latest code from the GitHub repository"
          sh """
            git clone --branch dev
            https://github.com/ayush-prajapati01/fundoo-notes-copy.git $WORKSPACE
          """
        }
      }
    }
  }
}
```

```

    }
}

stage('Sync Application to Backend Server') {
    steps {
        script {
            echo "Syncing the application to backend/app server"
            sh """
            rsync -avz $WORKSPACE ubuntu@172.31.1.175:/tmp
            """
        }
    }
}

stage('Copy Files to Destination Directory') {
    steps {
        script {
            echo "Copying files to the destination directory on the app server"
            sh """
            ssh ubuntu@172.31.1.175 'cp -rfv /tmp/fundoo-pipeline/*
/home/ubuntu/fundoo-notes-copy/'
            """
        }
    }
}

stage('Install Dependencies') {
    steps {
        script {
            echo "Installing dependencies on the app server"
            sh """
            ssh ubuntu@172.31.1.175 'source /home/ubuntu/myenv/bin/activate
&& pip3 install -r /home/ubuntu/fundoo-notes-copy/requirements.txt && pip3 install
django'
            """
        }
    }
}

stage('Run Migrations') {
    steps {
        script {
            echo "Running database migrations"
            sh """
            ssh ubuntu@172.31.1.175 'source /home/ubuntu/myenv/bin/activate
&& python3 /home/ubuntu/fundoo-notes-copy/fundoo_notes/manage.py migrate'
            """
        }
    }
}

```

```

}


stage('Start Django Server') {
    steps {
        script {
            echo "Starting Django server on port 8000"
            sh """
            ssh ubuntu@172.31.1.175 'sudo systemctl restart
fundoo-notes.service'
            """
        }
    }
}

stage('Check Server Status') {
    steps {
        script {
            echo "Checking the status of the Django server"
            sh """
            ssh ubuntu@172.31.1.175 'sudo systemctl status
fundoo-notes.service'
            """
        }
    }
}
}

```

Save and Build

1. Click on the Save button at the bottom of the configuration page.

 Pipeline

```
77
78
79
80
81
82
83
84
85
```

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save

Apply

2. To run your job, go back to the job page and click on Build Now.

Dashboard > fundoo-pipeline >

Status

</> Changes

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

✓ fundoo-pipeline

Fundoo Pipeline

Permalinks

- Last build (#6), 2 days 18 hr ago
- Last stable build (#6), 2 days 18 hr ago
- Last successful build (#6), 2 days 18 hr ago
- Last failed build (#4), 2 days 18 hr ago
- Last unsuccessful build (#4), 2 days 18 hr ago
- Last completed build (#6), 2 days 18 hr ago

3. Monitor the build process by clicking on the build number in the build history.

Build History

trend ▾

Filter...

✓ #6

18 Oct 2024, 12:29

✓ #5

18 Oct 2024, 12:27

✗ #4

18 Oct 2024, 12:20

✗ #3

18 Oct 2024, 12:20

✗ #2

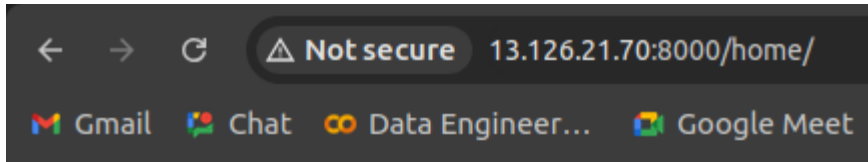
18 Oct 2024, 12:14

Atom feed for all

Atom feed for failures

Verify Deployment

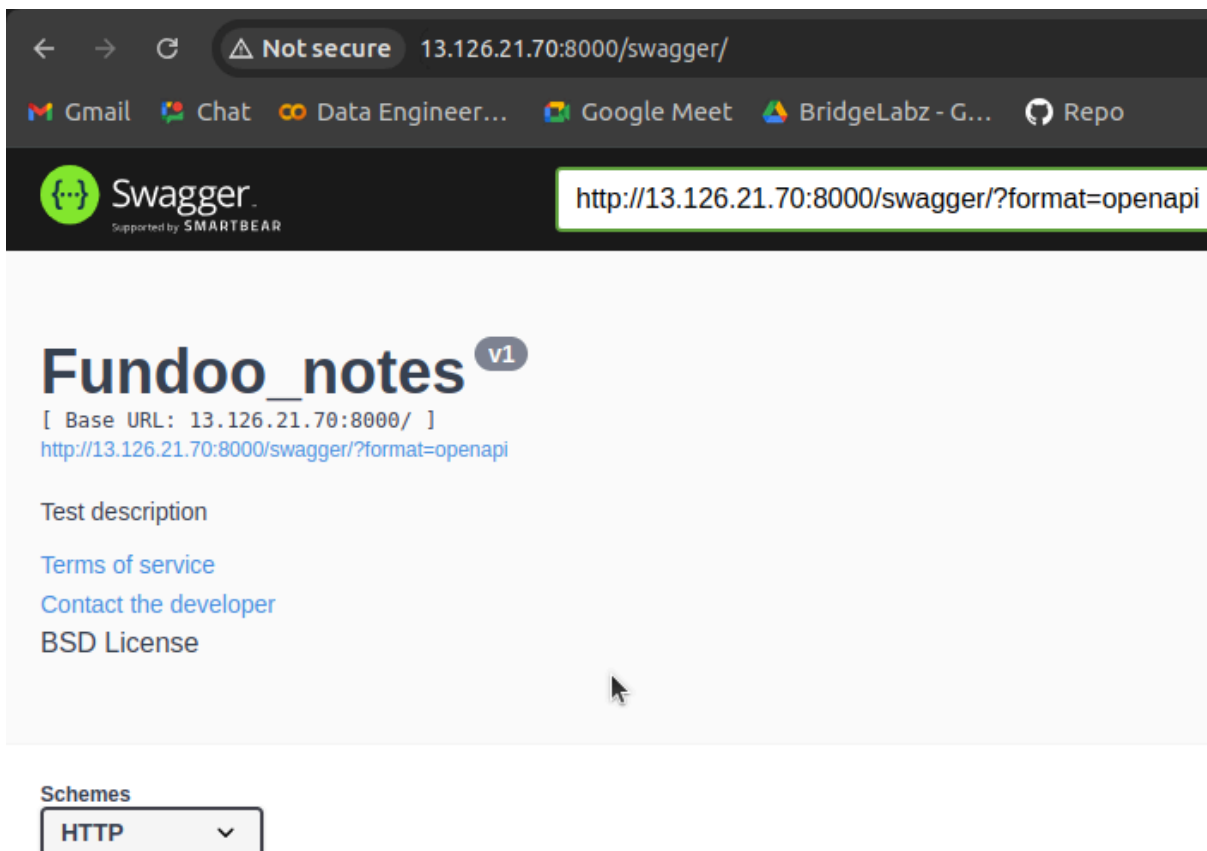
Once the build is complete, verify that your Django application is running correctly by accessing it via its public IP address or domain name.



Welcome, to Fundoo notes ayush !

Perform API testing

We can perform api testing using swagger to confirm our applications is running perfectly



Conclusion

By following these steps, you should be able to successfully configure and run a pipeline job in Jenkins that pulls your Django application code from a Git repository and executes necessary commands on your EC2 instance. This setup allows for continuous integration and deployment of your application as changes are made in your codebase.