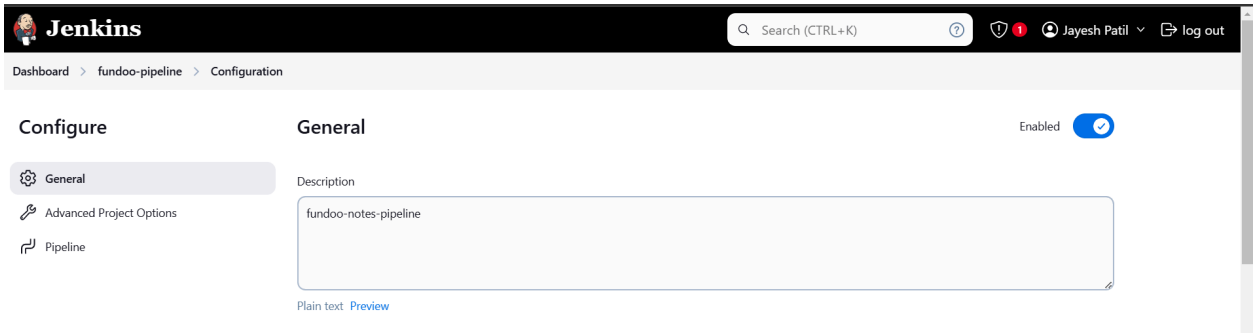
Steps to create a New Pipeline Project in Jenkins

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

    environment {
        WORKSPACE = "${env.WORKSPACE}" // Set the workspace environment variable
        REMOTE_HOST = '172.31.86.131' // Update this if the remote server IP changes
        REMOTE_USER = 'ubuntu' // Update if using a different SSH user
        VENV_PATH = '/home/ubuntu/myenv/bin/activate' // Path to the Python virtual environment
        APP_DIR = '/home/ubuntu/Aws_test_fundoo' // Final app directory
        TMP_DIR = '/tmp/fundoo-notes' // Temporary directory for syncing files
        GIT_REPO = 'https://github.com/jayeshpatil045/Aws_test_fundoo.git' // Your GitHub repository URL
        GIT_BRANCH = 'dev' // Specify the branch you want to clone
    }

    stages {
```

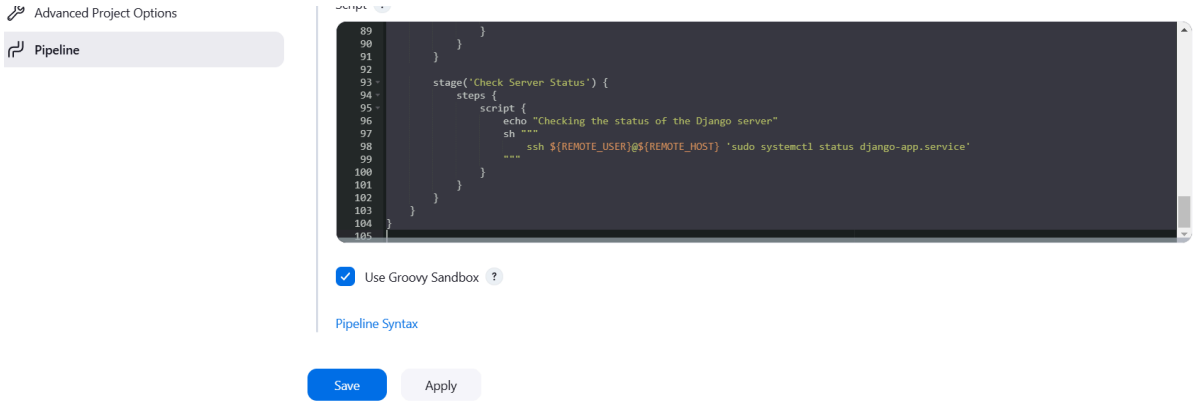
```
        stage('Checkout Code from GitHub') {
            steps {
                script {
                    echo "Cleaning workspace before checkout"
                    cleanWs() // Clean the workspace

                    echo "Cloning the latest code from your GitHub repository"
                    sh """
                        git clone --branch ${GIT_BRANCH} ${GIT_REPO} ${WORKSPACE}
                    """
                }
            }
        }

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

        stage('Copy Files to Destination Directory') {
            steps {
                script {
                    echo "Copying files to the destination directory on the app server"
                    sh """
                        ssh ${REMOTE_USER}@${REMOTE_HOST} 'cp -rfv ${TMP_DIR}/*
${APP_DIR}/'
                    """
                }
            }
        }

        stage('Install Dependencies') {
```

```
        steps {
            script {
                echo "Installing dependencies on the app server"
                sh """
                    ssh ${REMOTE_USER}@${REMOTE_HOST} '
                        source ${VENV_PATH} &&
                        pip install -r ${APP_DIR}/requirements.txt &&
                        pip install django
                    '
                """
            }
        }
    }

    stage('Run Migrations') {
        steps {
            script {
                echo "Running database migrations"
                sh """
                    ssh ${REMOTE_USER}@${REMOTE_HOST} '
                        source ${VENV_PATH} &&
                        python ${APP_DIR}/fundoo_notes/manage.py makemigrations &&
                        python ${APP_DIR}/fundoo_notes/manage.py migrate
                    '
                """
            }
        }
    }

    stage('Start Django Server') {
        steps {
            script {
                echo "Starting Django server on port 8000"
                sh """
                    ssh ${REMOTE_USER}@${REMOTE_HOST} 'sudo systemctl restart
django-app.service'
                """
            }
        }
```

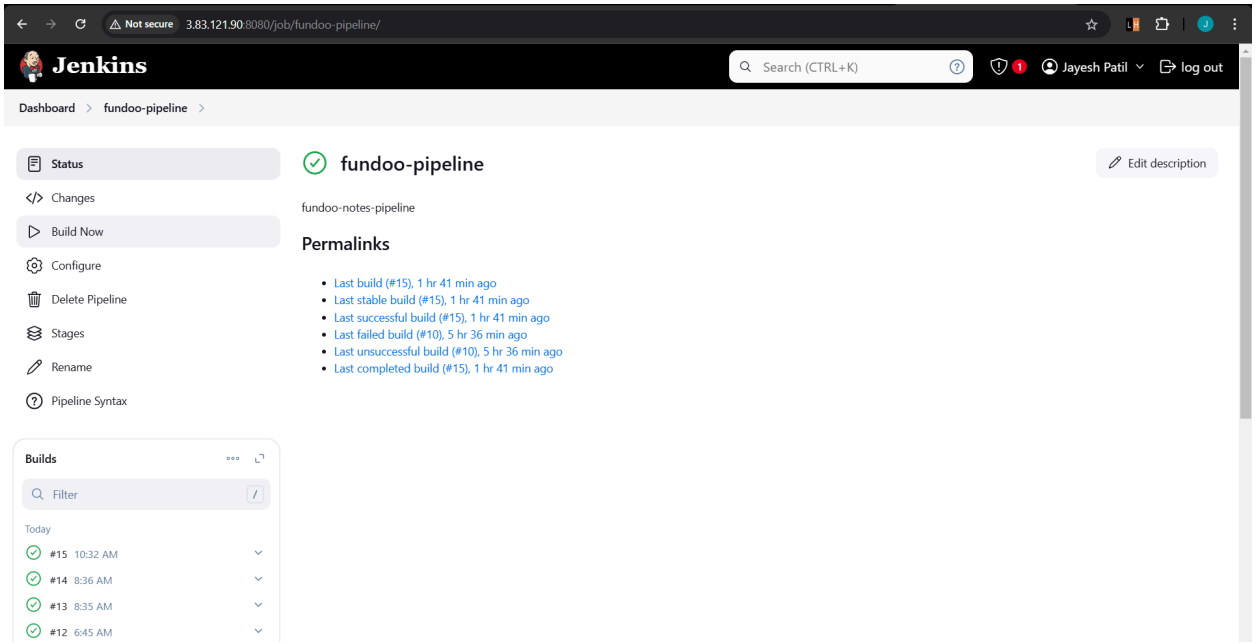
```
        }

        stage('Check Server Status') {
            steps {
                script {
                    echo "Checking the status of the Django server"
                    sh """
                        ssh ${REMOTE_USER}@${REMOTE_HOST} 'sudo systemctl status
django-app.service'
                    """
                }
            }
        }
    }
}
```

Save and Build

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

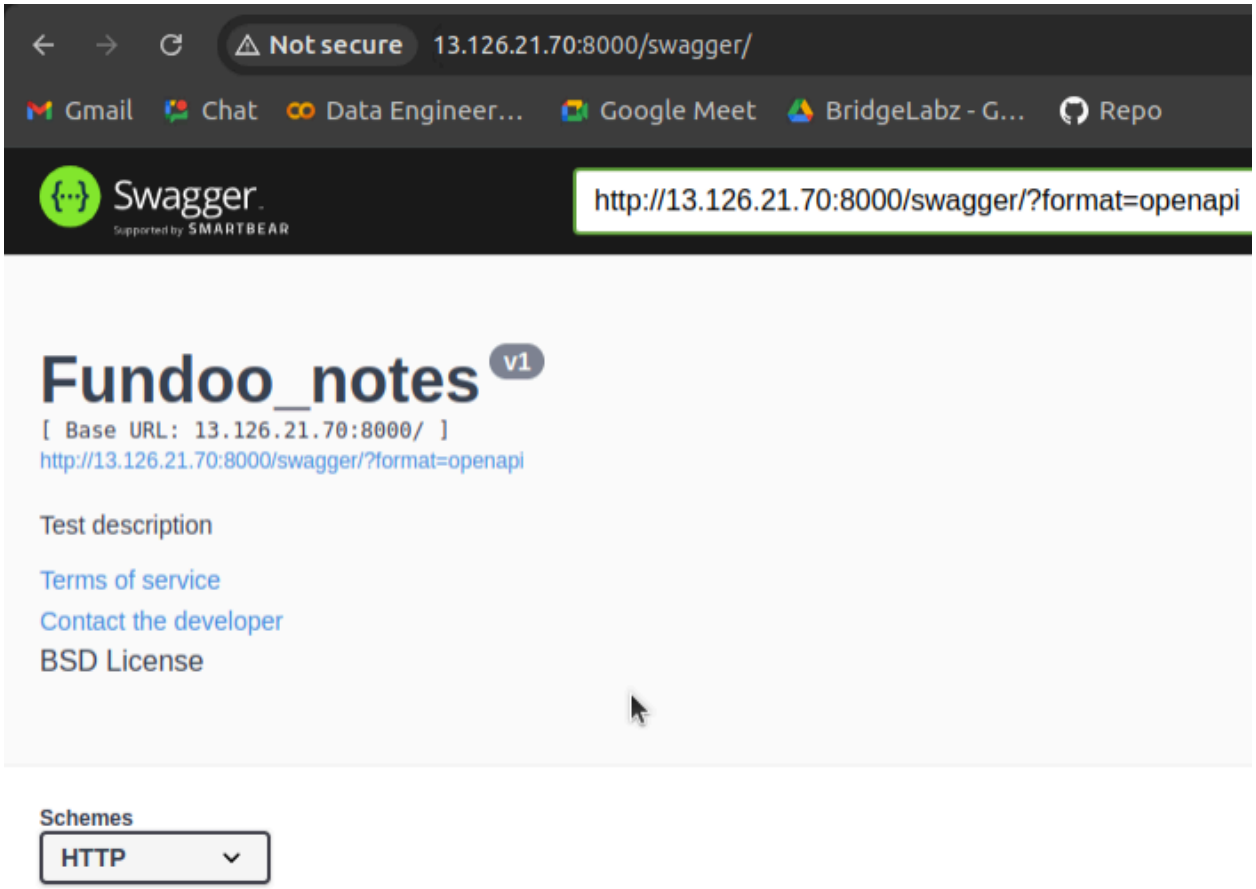


2. To run your job, go back to the job page and click on Build Now
3. Monitor the build process by clicking on the build number in the build history.



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.


Not secure
13.126.21.70:8000/swagger/
Gmail
Chat
Data Engineer...
Google Meet
BridgeLabz - G...
Repo
Swagger
Supported by SMARTBEAR
http://13.126.21.70:8000/swagger/?format=openapi
Fundoo_notes
v1
[Base URL: 13.126.21.70:8000/]
http://13.126.21.70:8000/swagger/?format=openapi
Test description
Terms of service
Contact the developer
BSD License
Schemes
HTTP