JENKINS CI/CD Pipeline (Development Cycle) With Gitlab, Nginx, Django

SYSTEM ARCHITECTURE

.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.

Section A

Context of the Project

```
There are 3 Systems that are important to be known in this architecture


1. Developer/Development Machine
2. Jenkins Server
3. Deployment/Staging Server
```

For the context of this Document and the Project we are going to work on these

Git and Gitlab Setup

```
 - Creating or Clong an existing Github Repository to Gitlab Project
- Ading SSH key to your Gitlab Account
```

Manual Deployment of the Gitlab Code to a Staging Server

```
 - Setup a Dedicated Deploy User to Staging Server
- Create and Add Deploy User SSH Key to GitLab
- Installing Required Dependencies for the Django Project (To avoid confliction while pip install command)
- Setting Up Database (Optional)
- Creating a virtual Enviornment
- Manually installing Project Dependancies with requirement.txt File
- Serving the application on localhost
- Serving the application wit gunicorn
- Setting up a Nginx Server
```

Redploying Manually

```
 - Redeploy application manually
- Adding a deploy script
- Deploy with deploy script
- Running sudo commands without password
```

Setting Up Jenkins Server

```
- Installing Jenkins
- Exposing Local Jenkins instance to Public with NGROK
- Installing required plugins
- Configure Jenkins to send email notification
```

CICD With Gitlab and Jenkins

```
- Install Gitlab Plugin in Jenkins
- Configure Gitlab-Jenkins Connections
- Adding Jenkinsfile to code
- Create Jenkins Pipeline For staging
- Run Pipeline manually
- Jenkins Workspace
- Configure Jenkinsfile for building and Testing Code
- Configuring the Deploy Stage
- Jenkins to SSH to deployment staging server
- Configuring WebHook to trigger the Pipleline on Push Events
- Testing the PipeLine
```