

Drupal CI/CD Deployment Documentation

1. Overview

This document explains the **end-to-end CI/CD setup** for deploying a **Drupal application** using **GitLab + Jenkins + SSH (rsync)** without Docker. The setup supports **fully automated deployments** with **atomic releases** and **instant rollback**.

2. Architecture



3. Servers & Roles

Server	Hostname	Role	OS
GitLab	git.aksamity.com	Source Code Management	Ubuntu
Jenkins	jen.aksamity.com	CI/CD Automation	Rocky Linux
Production	dev-ayush.aksamity.com	Live Drupal Site	Rocky Linux

4. Production Directory Structure

```
/user/AYUSH
├─ current -> releases/20241217_1530 # Live site (symlink)
├─ releases/ # Versioned deployments
```

```

|   |— 20241217_1500
|   |— 20241217_1530
|   └─ shared/                                # Persistent data
|       └─ sites/default/
|           └─ files/
|               └─ settings.php

```

Key Points - `current` is always the live version - Each deploy creates a new folder under `releases/` - Rollback is done by switching the `current` symlink

5. GitLab Setup

Repository

- Repository Name: `drupal-ayush-prod`
- Namespace: `root`
- Repo URL:

```
git@git.aksamity.com:root/drupal-ayush-prod.git
```

Code Rules

Allowed in repo: - `core/` - `modules/` - `themes/` - `profiles/` - `index.php` - `composer.json` - `composer.lock`

NOT allowed in repo: - `sites/default/files/` - `sites/default/settings.php` - `vendor/`

6. SSH Trust Configuration

6.1 Jenkins → GitLab

- Jenkins generates an SSH key
- Public key added to GitLab (root user → SSH Keys)
- Test:

```
ssh -T git@git.aksamity.com
```

6.2 Jenkins → Production (deploy user)

Create deploy user

```
useradd -m -s /bin/bash deploy
usermod -aG apache deploy
```

Authorize Jenkins SSH key

```
mkdir -p /home/deploy/.ssh
chmod 700 /home/deploy/.ssh
# Paste Jenkins public key into authorized_keys
chmod 600 /home/deploy/.ssh/authorized_keys
chown -R deploy:deploy /home/deploy/.ssh
```

Test from Jenkins server

```
ssh deploy@192.168.2.97
```

7. Production Code Bootstrap

```
cd /user/AYUSH/current
git init
git branch -M main
```

.gitignore

```
sites/default/files/
sites/default/settings.php
vendor/
.git
```

```
git add .
git commit -m "Initial production code"
git remote add origin git@git.aksamity.com:root/drupal-ayush-prod.git
git push -u origin main
```

8. Jenkins Pipeline Configuration

Required Packages

```
dnf install -y git rsync
```

Jenkins Job: drupal-prod-deploy

```
pipeline {
  agent any

  environment {
    PROD_USER = "deploy"
    PROD_HOST = "192.168.2.97"
    PROD_PATH = "/user/AYUSH"
  }

  stages {
    stage('Checkout') {
      steps {
        git branch: 'main',
            url: 'git@git.aksamity.com:root/drupal-ayush-prod.git'
      }
    }

    stage('Deploy') {
      steps {
        sh '''
          TS=$(date +%Y%m%d_%H%M%S)
          RELEASE=${PROD_PATH}/releases/$TS

          ssh ${PROD_USER}@${PROD_HOST} "mkdir -p $RELEASE"

          rsync -avz
            --exclude=.git
            --exclude=sites/default/files
            --exclude=sites/default/settings.php
            --exclude=vendor
            ./ ${PROD_USER}@${PROD_HOST}::$RELEASE/

          ssh ${PROD_USER}@${PROD_HOST} "ln -sf $RELEASE ${PROD_PATH}/current"
          '''
      }
    }
  }
}
```

```
}  
}
```

9. Developer Workflow

Developers **ONLY** interact with GitLab.

```
git clone git@git.aksamity.com:root/drupal-ayush-prod.git  
# make changes  
git commit -m "feature / bugfix"  
git push
```

- No production access
- No Jenkins access
- Deployment happens automatically

10. Rollback Procedure (Zero Downtime)

List releases

```
ls -lt /user/AYUSH/releases
```

Rollback

```
ln -sf /user/AYUSH/releases/<PREVIOUS_RELEASE> /user/AYUSH/current
```

- Instant rollback
- No service restart
- No downtime

11. Automation Status

Feature	Status
Code push	Automated
Build & deploy	Automated
Zero downtime	Yes

Feature	Status
Rollback	Manual (1 command)
Developer access to prod	No

12. Best Practice Statement

Developers push code to GitLab. Jenkins automatically deploys the application to production using atomic releases, ensuring zero downtime and instant rollback capability.

End of Document