**Deployment Setup Using GitHub PAT and VirtualBox VMs**

You're moving from local Jenkins to multi-VM deployment where:

* **VM1** (staging) runs your app on port 8000
* **VM2** (production) runs your app on port 8001
* Web apps should stay running **in the background**
* GitHub repo is cloned using **PAT via HTTPS**

**1. Set Up the Two VMs in VirtualBox**

Ensure both VMs are prepared:

|  |  |  |
| --- | --- | --- |
| **VM** | **Hostname** | **Port** |
| 1 | webapp-staging | 8000 |
| 2 | webapp-release | 8001 |

Each VM should have:

* Python or required runtime installed

**2. Create a GitHub PAT**

1. Go to: <https://github.com/settings/tokens>
2. Click **“Generate new token”**
3. Give it:
   * Name: Jenkins deploy
   * Expiration: optional
   * Scope: ✅ repo
4. Copy the token.   
   Token Example:

ghp\_ABC123ThisIsYourTokenXYZ456

**3. Update Jenkins Pipeline (Using PAT for Git)**

Instead of sshagent, we directly SSH into the VM and use HTTPS Git clone using the PAT.

This exposes the token in logs if not handled properly. **Use Jenkins credentials if possible**, but this version is written for simplicity.

**🔧 Example Jenkinsfile (Declarative)**

pipeline {

agent any

environment {

GIT\_REPO = 'https://github.com/your-username/your-repo.git'

GIT\_USER = 'your-username'

GIT\_PAT = 'ghp\_YourRealGitHubPATHere' // Replace or store in Jenkins credentials

}

stages {

stage('Deploy to Staging') {

steps {

sh """

ssh user@vm1-ip << 'EOF'

pkill -f "python3 app.py" || true

rm -rf webapp || true

git clone https://${GIT\_USER}:${GIT\_PAT}@github.com/your-username/your-repo.git webapp

cd webapp

pip3 install -r requirements.txt

nohup python3 app.py --port=8000 > app.log 2>&1 &

EOF

"""

}

}

stage('Manual Approval') {

steps {

input message: 'Deploy to Production?'

}

}

stage('Deploy to Production') {

steps {

sh """

ssh user@vm2-ip << 'EOF'

pkill -f "python3 app.py" || true

rm -rf webapp || true

git clone https://${GIT\_USER}:${GIT\_PAT}@github.com/your-username/your-repo.git webapp

cd webapp

pip3 install -r requirements.txt

nohup python3 app.py --port=8001 > app.log 2>&1 &

EOF

"""

}

}

}

}

**🛠️ Optional: Use Jenkins Credentials**

Instead of hardcoding the PAT in the Jenkinsfile, add it to:

1. **Jenkins → Credentials → Global → Add Credentials**
   * Kind: Username with password
   * Username: your-username
   * Password: your PAT
   * ID: github-pat
2. Use in the pipeline:

withCredentials([usernamePassword(credentialsId: 'github-pat', usernameVariable: 'GIT\_USER', passwordVariable: 'GIT\_PAT')]) {

sh """

ssh user@vm1-ip << 'EOF'

git clone https://$GIT\_USER:$GIT\_PAT@github.com/your-username/your-repo.git

**4. Keep the App Running After Jenkins Ends**

Use nohup (simpler) or systemd (recommended for stability).

***Option A: Using nohup (already included)***

This keeps the app running in background, even after logout.

***Option B: systemd service (survives reboots)***

1. **On each VM, create a file:**

sudo nano /etc/systemd/system/webapp.service

1. **Example for staging (VM1):**

[Unit]

Description=Web App on 8000

After=network.target

[Service]

ExecStart=/usr/bin/python3 /home/user/webapp/app.py --port=8000

WorkingDirectory=/home/user/webapp

Restart=always

User=user

[Install]

WantedBy=multi-user.target

1. **Enable and start:**

sudo systemctl daemon-reexec

sudo systemctl enable webapp

sudo systemctl start webapp

Do the same on **VM2** with port 8001.

**Final Checklist**

* Two VMs with SSH access enabled
* GitHub repo is publicly accessible or PAT is valid
* Jenkins pipeline includes PAT securely
* Web apps run on ports 8000 and 8001
* Optionally use system or service file for auto-restart