Step 1: AWS Servers Setup

- **Server 1 → Jenkins Master**
  - Install Java & Jenkins
- **Server 2 → Jenkins Agent (Node)**
  - Install Java, Docker
  - Configure for Jenkins agent connection

Note: T3 Micro instance can be used, but for production-grade workloads, consider higher specs or RDS for database.

Step 2: Jenkins Agent Setup on AWS Server

```
# Update package lists
sudo apt update

# Install Java 17 (required for Jenkins)
sudo apt install openjdk-17-jdk -y

# Install Docker
sudo apt install docker.io -y

# Add current user to Docker group
sudo usermod -aG docker ubuntu
```

Tip: Log out and log back in to apply Docker group changes.

Step 3: Connect Jenkins Master to Agent Node

1. On Jenkins Master → **Manage Jenkins → Nodes → New Node**
2. Configure:
   - **Node name:** agent-1
   - **Remote root directory:** /home/ubuntu
   - **Launch method:** Launch agent via SSH
   - **Host: Agent AWS server IP**
   - **Credentials:** SSH private key
3. Setup SSH keys:

```
cd ~/.ssh

ubuntu@ip-172-31-14-95:~/.ssh$ ls

authorized_keys  id_ed25519  id_ed25519.pub
```

4. # On master server
5. cd ~/.ssh
6. vim id_rsa (private key)
7.
8. # On agent server
9. cd ~/.ssh
10. vim authorized_keys
11. # Paste master server public key here
12. Verify connection in Jenkins → Node should show **online**.

Step 4: Create Dockerfile

- Basic example for PHP / Web application:

```
FROM php:8.2-apache
COPY . /var/www/html/
EXPOSE 80
```

Tip: Ensure `.dockerignore` is present to skip unnecessary files.

Step 5: Create Jenkins Pipeline Job

1. Jenkins → **New Item → Pipeline**
2. Use the following **pipeline script**:

```
pipeline {
  agent { label 'agent-1' }

  stages {
    stage('Pull Code') {
      steps {
        git branch: 'main', url: 'https://github.com/<username>/<repo>.git'
      }
    }

    stage('Build Docker Image') {
      steps {
        sh 'docker build -t myapp:latest .'
      }
    }

    stage('Run Container') {
      steps {
        sh 'docker stop myapp || true'
        sh 'docker rm myapp || true'
        sh 'docker run -d -p 8080:80 --name myapp myapp:latest'
      }
    }
```

```
   }

   post {
      success {
         echo 'Deployment Successful!'
      }
      failure {
         echo 'Deployment Failed!'
      }
   }
}
////////////////

pipeline {

   agent { label 'Vinod' }


   stages {


      stage('Pull Code') {

         steps {

            git branch: 'main', url: 'https://github.com/SheetalChive/JenkinsProject.git'

         }

      }


      stage('Build Docker Image') {

         steps {

            sh 'docker build -t jenkinsproject:latest .'

         }

      }


      stage('Run Container') {

         steps {

            // Stop old container if exists

            sh 'docker stop jenkinsproject || true'


            // Remove old container if exists

            sh 'docker rm jenkinsproject || true'
```

```
                // Run new container on port 8000
                sh 'docker run -d -p 8000:80 --name jenkinsproject jenkinsproject:latest'
            }
        }
    }
}
```