Title: Docker Volume-Based Multi-Branch Deployment via Jenkins Slave Node
Assignment: Docker | 4

Author: Vinay Hinukale

Objective:

Deploy three versions of an index.html webpage from three GitHub branches (2025Q1, 2025Q2, 2025Q3) into three Docker httpd containers using volumes. Each container will serve on a unique host port (80, 90, 8001) and will be deployed using a Jenkins pipeline running on a **Jenkins slave node**.

Jenkins Master Prerequisites

- Jenkins Master already installed and running.
- SSH access and credentials to the slave EC2 instance.

Jenkins Slave Node Configuration

1. **Login to Jenkins Master UI**
   Navigate to: Dashboard → Manage Jenkins → Nodes → New Node
2. **Configure Node**:
   - **Name**: slave
   - **Type**: Permanent Agent
   - **# of Executors**: 2
   - **Remote Root Directory**: /home/ec2-user/Jenkins
     - Create this directory on the slave EC2:
     - mkdir -p /home/ec2-user/Jenkins
     - sudo usermod -aG docker ec2-user
   - **Labels**: slave
   - **Usage**: Only build jobs with label expressions matching this node
   - **Launch Method**: Launch agents via SSH
   - **Host**: Slave EC2's Private IP

    - **Credentials**:
        - Type: SSH Username with private key
        - Scope: Global
        - Username: ec2-user (or jenkins if that user is created with sudo privileges)
        - Private Key: Enter directly (paste .pem or .ppk)
    - **Host Key Verification Strategy**: Manually trusted key Verification Strategy

3. **Apply and Save**

Jenkins Pipeline Job Configuration

1. **Create a New Pipeline Job**
    - Name: docker_multibranch_slave_pipeline
    - Type: Pipeline
2. **Paste the following Pipeline Script:**

```
pipeline {
  agent {
    label {
      label "slave"
    }
  }


  stages {
    stage("docker-install") {
      steps {
        sh '''
        sudo yum install docker -y
        sudo systemctl start docker
        docker stop Q1 Q2 Q3 || true
        docker rm Q1 Q2 Q3 || true
        '''
      }
    }

    stage("git-clone-b1") {
      steps {
        dir("2025Q1") {
          git url: "https://github.com/LegPro/docker-repo-1.git", branch: "2025Q1"
          sh '''
          docker run -itd --name Q1 -p 80:80 \
           -v /home/ec2-user/jenkins/workspace/docker-multibranch-pipeline/2025Q1:/usr/local/apache2/htdocs httpd
          docker exec Q1 chmod -R 777 /usr/local/apache2/htdocs
          '''
        }
      }
    }

    stage("git-clone-b2") {
      steps {
        dir("2025Q2") {
          git url: "https://github.com/LegPro/docker-repo-1.git", branch: "2025Q2"
          sh '''
          docker run -itd --name Q2 -p 90:80 \
           -v /home/ec2-user/jenkins/workspace/docker-multibranch-pipeline/2025Q2:/usr/local/apache2/htdocs httpd
          docker exec Q2 chmod -R 777 /usr/local/apache2/htdocs
          '''
        }
      }
    }

    stage("git-clone-b3") {
      steps {
        dir("2025Q3") {
          git url: "https://github.com/LegPro/docker-repo-1.git", branch: "2025Q3"
          sh '''
          docker run -itd --name Q3 -p 8001:80 \
            -v /home/ec2-user/jenkins/workspace/docker-multibranch-pipeline/2025Q3:/usr/local/apache2/htdocs httpd
          docker exec Q3 chmod -R 777 /usr/local/apache2/htdocs
          '''
        }
      }
    }
  }
}
```

✅ Tick: **Use Groovy Sandbox**

💾 Click: **Apply** and **Save**

Expected Outcome:

- Each GitHub branch (2025Q1, 2025Q2, 2025Q3) is cloned on the slave.
- Corresponding index.html files are volume-mounted into Docker containers.
- Containers are exposed at:
  - http://:80
  - http://:90

- http://:8001

Conclusion:

This documentation showcases how to integrate Docker volume usage with Jenkins slave nodes for automated multi-version web deployments across separate HTTPD containers. It enables isolated and reproducible CI/CD workflows for multi-branch repositories.