# Integrating Jenkins with Ansible

Integrating Jenkins with Ansible allows you to automate infrastructure provisioning, configuration management, and application deployment. This integration is beneficial for DevOps workflows, enabling continuous delivery (CD) and infrastructure as code (IaC) practices.

## Overview

**Jenkins**: Jenkins is an open-source automation server that facilitates continuous integration and delivery of software projects by automating parts of the software development lifecycle.

**Ansible**: Ansible is an open-source automation tool used for configuration management, application deployment, and task automation. It uses a simple, human-readable language (YAML) to describe automation jobs and does not require a client to be installed on managed nodes.

By integrating Jenkins with Ansible, you can:

- Automate the deployment of applications.
- Manage configurations and environments.
- Execute Ansible playbooks directly from Jenkins jobs or pipelines.

## Step-by-Step Setup

### 1. Prerequisites

- **Ubuntu machine**: A server or VM with Ubuntu installed.
- **Jenkins installed**: Jenkins must be installed and running. (Refer to previous setup steps if Jenkins is not installed).
- **Ansible installed**: Ansible must be installed on the Jenkins server.

### 2. Install Jenkins on Ubuntu

If Jenkins is not already installed, follow these steps:

1. **Update System Packages**:
   ```
   sudo apt update
   sudo apt upgrade -y
   ```

2. **Install Java**:
   Jenkins requires Java to run. Install OpenJDK:
   ```
   sudo apt install openjdk-11-jdk -y
   ```

3. **Add Jenkins Repository and Install Jenkins**:
   ```
   wget -q -O -
   https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo
   apt-key add -

   sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable
binary/ > /etc/apt/sources.list.d/jenkins.list'

   sudo apt update

   sudo apt install jenkins -y
   ```

4. **Start and Enable Jenkins Service**:
   ```
   sudo systemctl start jenkins

   sudo systemctl enable jenkins
   ```

5. **Access Jenkins**:
   Open your web browser and go to
   `http://your_server_ip_or_domain:8080` to complete the Jenkins setup.

**3. Install Ansible on Ubuntu**

1. **Add Ansible PPA**:
   ```
   sudo apt update
   sudo apt install software-properties-common -y
   sudo add-apt-repository --yes --update ppa:ansible/ansible
   ```

2. **Install Ansible**:
   ```
   sudo apt install ansible -y
   ```

3. **Verify Ansible Installation**:
   `ansible --version`

4. This should display the installed Ansible version and confirm that Ansible is installed correctly.

**4. Configure Jenkins for Ansible Integration**

1. **Install Ansible Plugin in Jenkins**:
   - Go to Jenkins Dashboard → Manage Jenkins → Manage Plugins.
   - In the "Available" tab, search for "Ansible" and install the "Ansible" plugin.
   - Restart Jenkins after the plugin is installed.

2. **Configure Ansible in Jenkins**:
   - Go to Jenkins Dashboard → Manage Jenkins → Global Tool Configuration.
   - Scroll down to the "Ansible" section and click "Add Ansible".
   - Provide a name for this Ansible installation (e.g., "Ansible").
   - Ensure that the "Install automatically" checkbox is **unchecked** because Ansible is already installed on the server.
   - Specify the path to the Ansible executable (default is `/usr/bin/ansible`).

3. **Add Ansible Credentials in Jenkins**:
   - Go to Jenkins Dashboard → Manage Jenkins → Manage Credentials.
   - Add new credentials that Jenkins will use to connect to remote servers.
   - Select "SSH Username with private key" and provide:
     - **Username**: The SSH username for the remote server.
     - **Private Key**: The private SSH key that matches the public key on the remote server.

**5. Create a Jenkins Job to Run Ansible Playbook**

1. **Create a New Jenkins Job**:
   - Go to Jenkins Dashboard → New Item → Freestyle Project.
   - Enter a name for the job and select "Freestyle project".

- Click "OK" to create the job.

2. **Configure Source Code Management**:
   - If your Ansible playbook is stored in a version control system (like Git), configure the source code management section to pull the playbook from the repository.
   - Select "Git" and enter the repository URL and credentials if needed.

3. **Add Build Step to Run Ansible Playbook**:
   - In the "Build" section, click "Add build step" and select "Invoke Ansible Playbook".
   - Configure the Ansible playbook execution:
     - **Playbook Path**: The path to your Ansible playbook file (relative to the workspace if using SCM).
     - **Inventory**: Specify the inventory file or inline inventory.
     - **Credentials**: Select the SSH credentials added earlier for connecting to remote hosts.
     - **Additional Parameters**: Add any extra Ansible parameters if needed (e.g., `-vv` for verbose output).

4. **Save and Build the Job**:
   - Save the Jenkins job configuration.
   - Click "Build Now" to run the job.

**6. Create a Jenkins Pipeline to Run Ansible Playbook (Optional)**

1. **Create a New Pipeline Job**:
   - Go to Jenkins Dashboard → New Item → Pipeline.
   - Enter a name for the pipeline job and select "Pipeline".
   - Click "OK" to create the pipeline job.

2. **Define the Pipeline Script**:
   Here is a simple pipeline script to run an Ansible playbook:
   groovy
   ```
   pipeline {
   ```

```
    agent any

    stages {
        stage('Checkout') {
            steps {
                git url:
'https://github.com/your-repo/your-ansible-playbook.git',
credentialsId: 'your-credentials-id'
            }
        }

        stage('Run Ansible Playbook') {
            steps {
                ansiblePlaybook playbook:
'your-playbook.yml', inventory: 'your-inventory',
credentialsId: 'your-credentials-id'
            }
        }
    }
}
```

- Replace `'https://github.com/your-repo/your-ansible-playbook.git'` with the URL of your Git repository.
- Replace `'your-playbook.yml'` with the path to your Ansible playbook file.
- Replace `'your-inventory'` with your inventory file or inline inventory.
- Replace `'your-credentials-id'` with the credentials ID for SSH access.

3. **Save and Run the Pipeline**:
   - Save the pipeline configuration.
   - Click "Build Now" to run the pipeline.

Integrating Jenkins with Ansible on Ubuntu allows you to automate various tasks related to infrastructure provisioning, configuration management, and application deployment. This setup streamlines CI/CD workflows and enhances the efficiency of managing infrastructure as code. By using Jenkins to orchestrate Ansible playbooks, you can achieve seamless automation and maintain consistency across different environments.