# Integrate Chef with a CI/CD pipeline for automated deployments

#### **Table of Contents**

- Description
- Problem Statement
- Prerequisites
- Implementation Steps
  - Step 1: Install Jenkins Using Docker
  - Step 2: Create a Windows Agent in Jenkins
  - Step 3: Start the Jenkins Agent on the Windows Node
  - Step 4: Create and Configure Credentials in Jenkins
  - Step 5: Add CHEF\_WORKSTATION\_PATH as a Global Environment Variable
  - Step 6: Run the Pipeline
  - Step-7: Update and Configure knife.rb for Jenkins Pipeline
- References

# **Description**

Integrating Chef with a CI/CD pipeline allows automated testing, configuration, and deployment of infrastructure changes. In this guide, you will configure a CI/CD pipeline using Chef to manage infrastructure and application deployment. This setup provides a consistent, testable approach to infrastructure provisioning and deployment.

#### **Problem Statement**

When managing infrastructure across environments, manual configurations can lead to inconsistencies. Automating deployment with Chef and integrating it into a CI/CD pipeline solves these issues by:

- Testing and validating configuration changes automatically.
- Applying configurations consistently across environments.
- Reducing human error in repetitive tasks.

# **Prerequisites**

#### **Software Required**

- Chef Workstation: To write and test Chef configurations.
- **Chef Server**: For managing node configurations.
- **Docker**: For installing Jenkins.
- Jenkins or another CI/CD tool: To set up automated testing and deployment.
- Git: For version control and Chef cookbook management.

#### **Hardware Requirement**

- Chef Server: 4 GB RAM, 2 CPU cores minimum.
- Workstation: 4 GB RAM, 2 CPU cores minimum.
- CI/CD Server (e.g., Jenkins): 8 GB RAM, 4 CPU cores recommended.

## **Implementation Steps**

#### **Step 1: Install Jenkins Using Docker**

#### 1. Create Docker Network:

• Create a Docker network named "jenkins" to allow communication between containers.

```
docker network create jenkins
```

#### 2. Run Docker Daemon Exposed with Port 2376:

• Start a Docker daemon with necessary configurations for Jenkins usage.

```
docker run --name jenkins-docker --rm --detach ^
    --privileged --network jenkins --network-alias docker ^
    -env DOCKER_TLS_CERTDIR=/certs ^
    --volume jenkins-docker-certs:/certs/client ^
    --volume jenkins-data:/var/jenkins_home ^
    --publish 2376:2376 ^
docker:dind
```

#### 3. Run Jenkins Image:

• Start Jenkins container using a custom image or the official Jenkins image, ensuring it's properly configured.

```
docker run --name jenkins-blueocean --restart=on-failure --detach ^
    --network jenkins --env DOCKER_HOST=tcp://docker:2376 ^
    --env DOCKER_CERT_PATH=/certs/client --env DOCKER_TLS_VERIFY=1 ^
    --volume jenkins-data:/var/jenkins_home ^
    --volume jenkins-docker-certs:/certs/client:ro ^
    --publish 8080:8080 --publish 50000:50000 vijaynvb/jenkins:1.0
```

• Open a browser and navigate to http://localhost:8080.

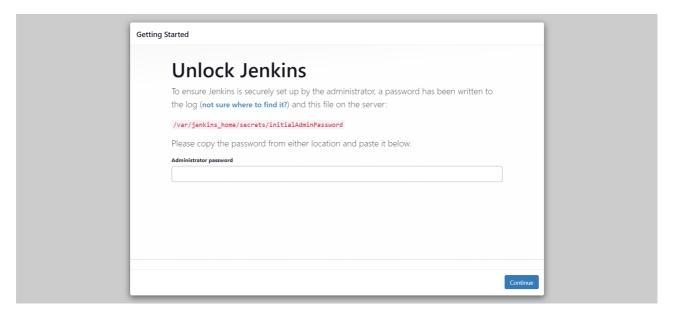
#### Setup admin details

Access the jenkins application in localhost:8080

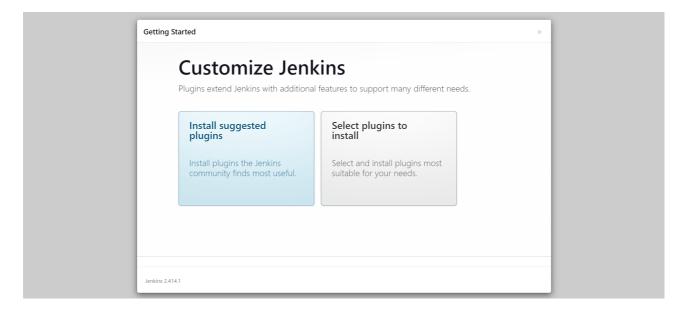
• Login with default credentials - To get the password, go to the shown directory or simply run docker logs containerId & get the password from logs.

```
Chiberally jay acousts to the Stricture Log Lo
```

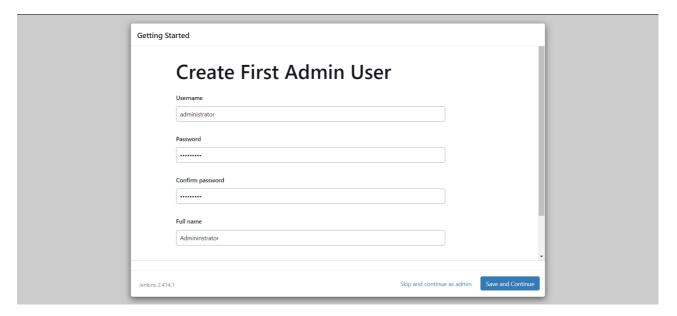
• copy the password from console paste it in login page.



• Select Install suggested plugins



Create new user 'administrator' with password '\*\*\*\*'



• Save and Continue

## **Step 2: Create a Windows Agent in Jenkins**

- 1. Go to Manage Jenkins:
  - o On the Jenkins dashboard, click **Manage Jenkins**.

+	New Item			
90	People			
<b>₽</b>	Build History			
0	Project Relationship			
	Check File Fingerprint			
<b>(3)</b>	Manage Jenkins			
	My Views			
	Open Blue Ocean			
	anage Nodes: roll down and click <b>Nodes</b> (or <b>M</b>	anage Nodes and Clouds).		
System	Configuration			
छ	System  Configure global settings and paths.	Tools  Configure tools, their locations and automatic installers.	\$ <del>54</del>	Plugins  Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

# 3. Create New Node:

Nodes

Add, remove, control and

Jenkins runs jobs on.

monitor the various nodes that

- Click **New Node** on the left sidebar.
- Give the node a name (e.g., windows-node).
- Select **Permanent Agent** and click **OK**.

Clouds

Add, remove, and configure

cloud instances to provision

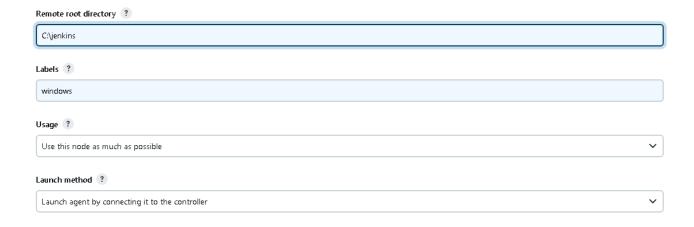
agents on-demand.

#### New node



#### 4. Configure the Windows Node:

- **Description**: Add an optional description.
- # Executors: Set the number of executors (typically 1).
- Remote Root Directory: Set the directory where Jenkins stores the workspace (e.g., C:\jenkins).
- Labels: Add a label (e.g., windows) to easily reference the node in your pipeline.
- Launch Method:
  - Select \*\*Launch agent by connecting it to the controller



5. **Save** the configuration.

#### Step 3: Start the Jenkins Agent on the Windows Node

1. Click on the windows-node





S	Name 1	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
₽	Built-In Node	Linux (amd64)	In sync	54.14 GB	864.90 MB	54.14 GB	Oms 🧔
X	windows- node		N/A	N/A	N/A	N/A	N/A 🧔
	Data obtained	15 min	15 min	15 min	15 min	15 min	15 min

#### 2. Run the Jenkins Agent on Windows:

- Open **Command Prompt** (CMD) on the Windows node.
- Copy the command from Run from agent command line (Windows) and run it in the windows
   Command Prompt.

Run from agent command line: (Windows)

```
curl.exe -sO http://172.19.4.192:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://172.19.4.192:8080/computer/windows%2Dnode/jenkins-agent.jnlp -secret
68d850d43c092ea18c2d2de56dfb17719fcac291f4dfa16180f79ce24fb2d5fc -workDir "C:\jenkins"
```

```
C:\Users\Administrator>curl.exe -sO http://172.19.4.192:8888/jnlp3ars/agent.jar & java -jar agent.jar -jnlpUrl http://172.19.4.192:8888/computer/windows%2Dnode/jenkins-agent.jnlp -secret 68d85043c892e318c2d2de5dfb17719fcac291f4dfa16180f70ce24fb2dfc -workDir "C:\jenkins"
NAV 20, 2024 11:02:128 May ong.jenkinsici.remoting.genjen.WorkDirManager initializeWorkDir
INFO: Using C:\jenkins\remoting as a remoting work directory
NAV 20, 2024 11:02:128 May ong.jenkinsici.remoting.genjen.WorkDirManager setupLogging
INFO: Setting up agent: windows-node
NAV 20, 2024 11:02:129 MA hudson.remoting.jnlp.Main createEngine
INFO: Using Remoting version: 311.vf2b_D-798b_ce99
NAV 20, 2024 11:02:129 MA hudson.remoting.ginle.workDirManager initializeWorkDir
INFO: Using Remoting version: 311.vf2b_D-798b_ce99
NAV 20, 2024 11:02:129 MA hudson.remoting.ginle.workDirManager initializeWorkDir
INFO: Using C:\jenkins\remoting as a remoting work directory
NAV 20, 2024 11:02:129 MA hudson.remoting.jnlp.MainScuilistener status
INFO: Remoting server among [http://172.19.4.192:80807]
NAV 20, 2024 11:02:129 MA hudson.remoting.jnlp.MainScuilistener status
INFO: Agent discovery successful
Agent address: 172.19.4.192
Agent part: 50808
INFO: Manager and Scuil Scuil
```

#### 3. Verify Connection:

- After running the command, the Jenkins agent should connect to the Jenkins master.
- The node will show as online on the Manage Nodes page in Jenkins.

#### **Step 4: Create and Configure Credentials in Jenkins**

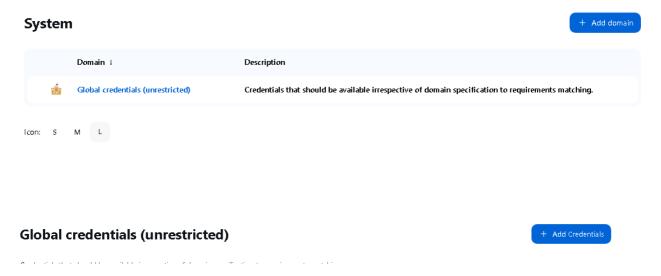
#### 4.1 Add Chef Server Credentials

• Go to Manage Jenkins > Credentials > System > Global Credentials>Add Credentials.

#### Security Credentials **Credential Providers** Secure Jenkins; define who is Configure credentials Configure the credential allowed to access/use the providers and types system. Users In-process Script Approval Create/delete/modify users that Allows a Jenkins administrator can log in to this Jenkins. to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions. 2 scripts pending approval.

# **Credentials**





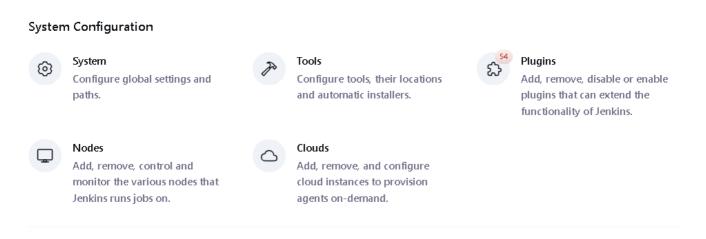
- Add a new credential:
  - **ID**: chef-server-id
  - Username: Chef Server username Password: Chef Server password



## Step 5: Add CHEF\_WORKSTATION\_PATH as a Global Environment Variable

#### **5.1 Configure Global Properties**

• Navigate to Manage Jenkins > Configure System.



- Enable Global Properties > Environment Variables.
- Add the following variable:

• Name: CHEF\_WORKSTATION\_PATH

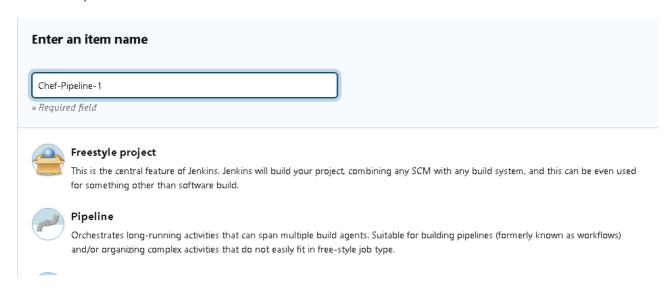
Value: C:/opscode/chef-workstation/bin



#### **Step 6: Run the Pipeline**

Once the agent is online, you can use the node in your pipeline as follows:

• Create a Pipeline



- Add the pipeline struture as shown below.
- Click on Apply and Save the configuration.
- Click on **Build Now** ro run the pipeline.

```
pipeline {
   agent { label 'windows' } // Ensure the pipeline runs on a Windows agent

   environment {
      CHEF_SERVER_CRED = credentials('sooryav') // Jenkins credential ID for

Chef server
      CHEF_ORG = 'chefserver02' // Chef organization name
      CHEF_NODE_IP = '172.19.3.101' // Target node IP (Ubuntu node)
      CHEF_NODE_USER = 'vagrant' // Node SSH username
      CHEF_NODE_PASS = 'vagrant' // Node SSH password
```

```
CHEF_WORKSTATION_PATH = "C:/opscode/chef-workstation/bin" // Chef
Workstation path on Windows
        CHEF_LOCAL_PATH = "C:/Users/Administrator/Downloads/chef-starter/chef-
repo" // Local Chef folder path
        PATH = "${env.CHEF WORKSTATION PATH};${env.PATH}" // Add Chef Workstation
path to the PATH
    }
    stages {
        stage('Use Local Chef Repository') {
            steps {
                echo 'Using local Chef repository...'
                echo Copying Chef repository from local path...
                xcopy /E /I /Y "${CHEF_LOCAL_PATH}" "%WORKSPACE%\\chef-repo"
            }
        }
        stage('Upload Cookbooks') {
            steps {
                echo 'Uploading cookbooks to Chef server...'
                bat """
                %CHEF_WORKSTATION_PATH%/knife cookbook upload webserver --config
%WORKSPACE%\\chef-repo\\.chef\\knife.rb --cookbook-path %WORKSPACE%\\chef-
repo\\cookbooks
            }
        }
    }
    post {
        success {
            echo 'Pipeline executed successfully! Chef configurations are applied
and verified.'
        }
        failure {
            echo 'Pipeline execution failed. Check the logs for detailed error
information.'
    }
}
```

• Check the pipeline status on **Console output**.

# **Console Output**

```
Started by user admin

[Pipeline] Start of Pipeline

[Pipeline] node

Running on windows-node in C:\jenkins\workspace\Chef-Pipeline-1

[Pipeline] {

[Pipeline] withCredentials

Masking supported pattern matches of %CHEF_SERVER_CRED% or %CHEF_SERVER_CRED_PSW%

[Pipeline] {

[Pipeline] withEnv

[Pipeline] stage

[Pipeline] stage

[Pipeline] {

[Use Local Chef Repository)

[Pipeline] echo

Using local Chef repository...

[Pipeline] bat
```

- The pipeline begins by cloning the Chef repository from GitHub into the Jenkins workspace.
- Upload the webserver cookbook to the Chef server using knife.

#### Step-7: Update and Configure knife.rb for Jenkins Pipeline

To fix the knife not found error, you need to rename your config.rb file to knife.rb and ensure it is configured properly to work with the Jenkins pipeline structure. Below are the steps:

#### Steps to Configure knife.rb

#### 1. Locate the config.rb File:

Navigate to the directory where config.rb is located. Typically, it will be in the .chef folder within your Chef repository.

```
cd C:/Users/Administrator/Downloads/chef-starter/chef-repo/.chef
```

#### 2. Rename config.rb to knife.rb:

• Rename the config.rb file using the command:

```
ren config.rb knife.rb
```

#### 3. Update knife.rb Configuration:

 Open the renamed knife.rb file in an editor (like Notepad++ or VS Code) and update it to work with your Jenkins pipeline structure:

# References

- 1. Chef Official Documentation: https://docs.chef.io/
- 2. **Jenkins Official Documentation**: https://www.jenkins.io/
- 3. Docker Documentation: https://docs.docker.com/
- 4. GitHub Repository Setup: https://github.com/