

Installing Jenkins Using Ansible

Project Overview

This project demonstrates the process of setting up Jenkins on an AWS EC2 instance using Ansible. It includes the creation of a master-slave architecture, installation of Ansible, Jenkins setup, and customization of Jenkins configurations through Ansible playbooks and templates.

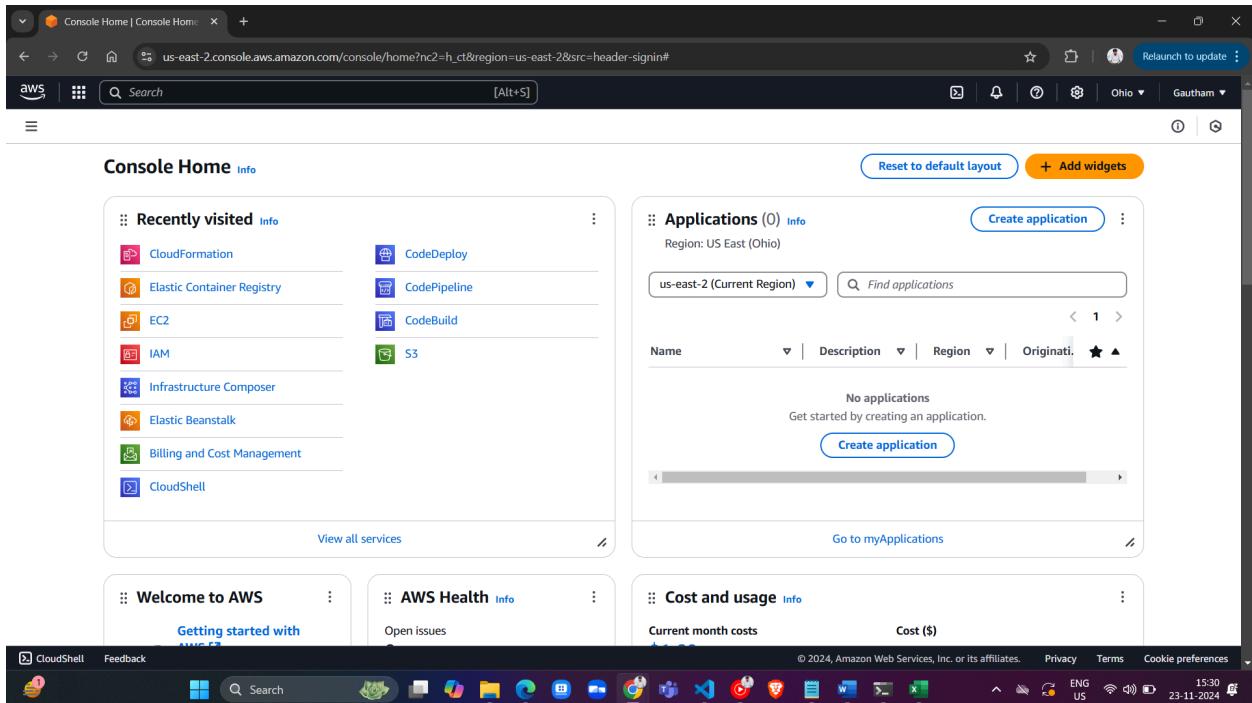
Prerequisites

- **AWS Account:** Ensure an AWS account with permissions to create and manage EC2 instances.
- **Putty Suite:** Includes PuTTY, Pageant, and related applications for SSH connectivity.
- **Key Pair:** A pre-created AWS EC2 key pair for secure instance access.

Procedures

Step 1. Launch EC2 Instances

Login: Sign in to the AWS Management Console.



Navigate to EC2: Select **EC2** under the **Compute** category.

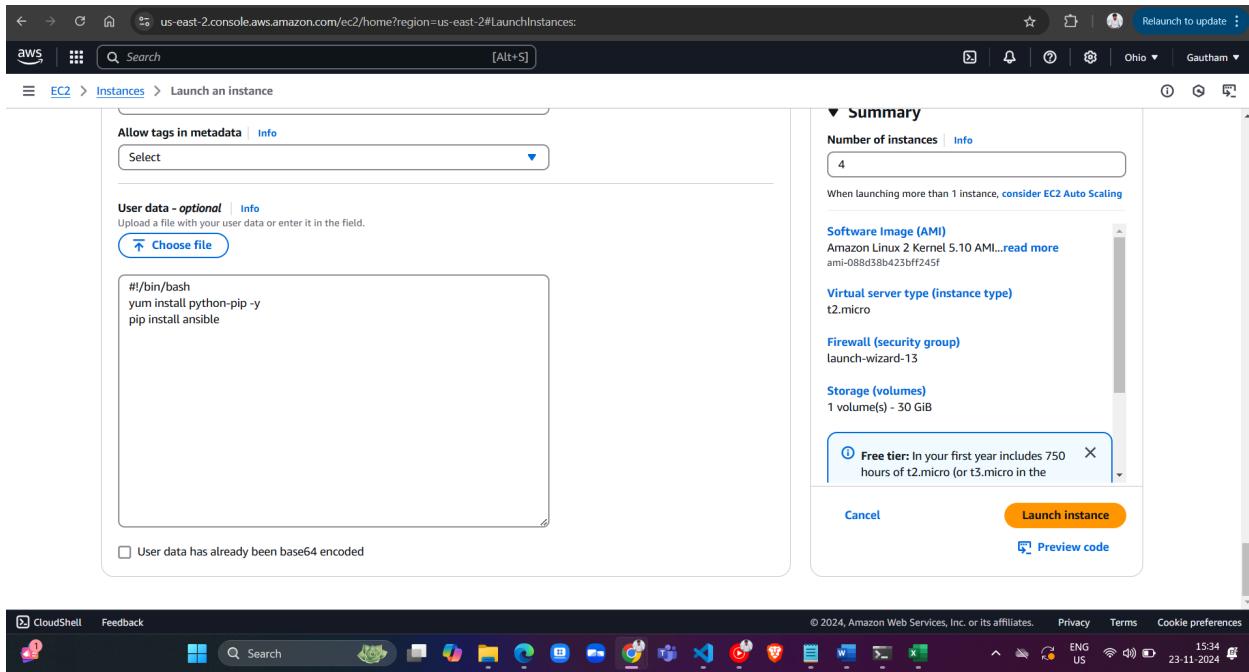
Launch Instance:

- From the EC2 dashboard, click **Launch Instance**.
- Choose the **Amazon Linux 2 AMI (HVM)** edition as the base image.

Configure Instance:

- Set the number of instances to **4** (1 master and 3 slaves).
- Select the key pair for secure access.
- Enter the following script in the **User Data** section to pre-install Python and Ansible:

```
#!/bin/bash
yum install python-pip -y
pip install ansible
```



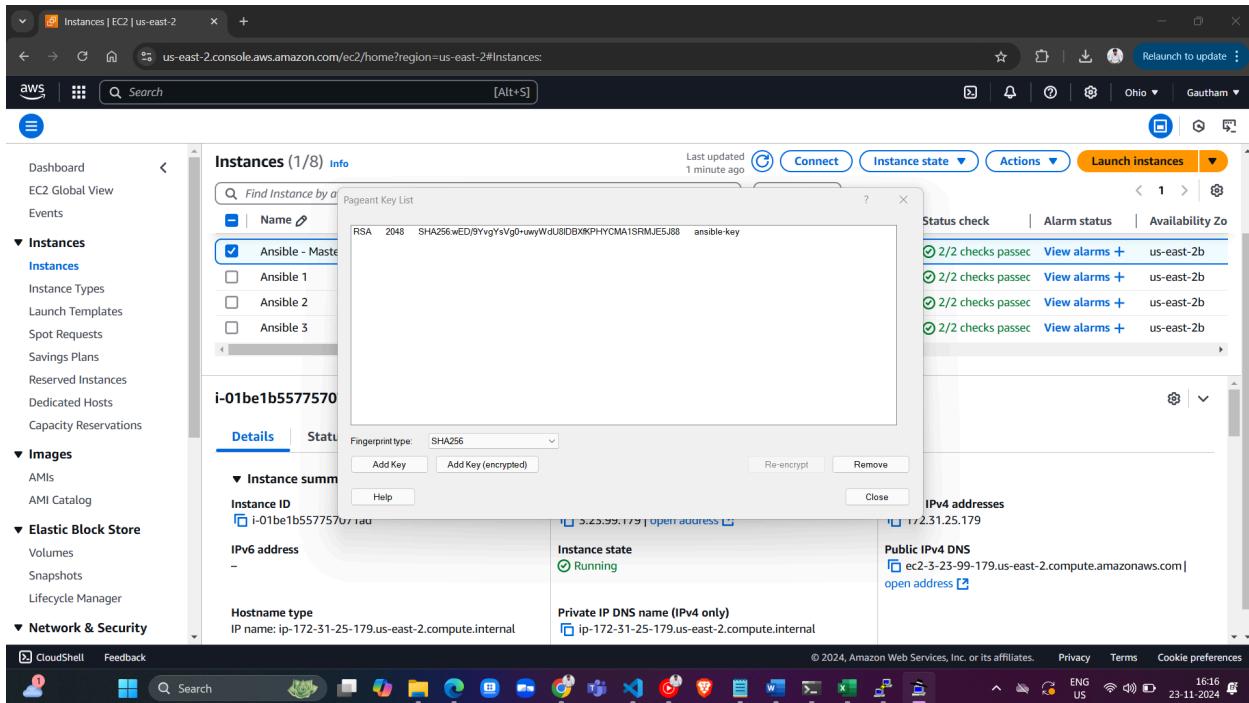
Launch Instances: Click **Launch** and wait for the instances to be ready.

Name	Instance ID	Instance state	Type	Status check	Availability Zone
Ansible - Master	i-01be1b557757071ad	Running	t2.micro	2/2 checks passed	us-east-2b
Ansible 1	i-0120a003be1...	Running	t2.micro	2/2 checks passed	us-east-2b
Ansible 2	i-052b08f2ecf3...	Running	t2.micro	2/2 checks passed	us-east-2b
Ansible 3	i-0b05236a4f7...	Running	t2.micro	2/2 checks passed	us-east-2b

Step 2. Connect to the Master Instance

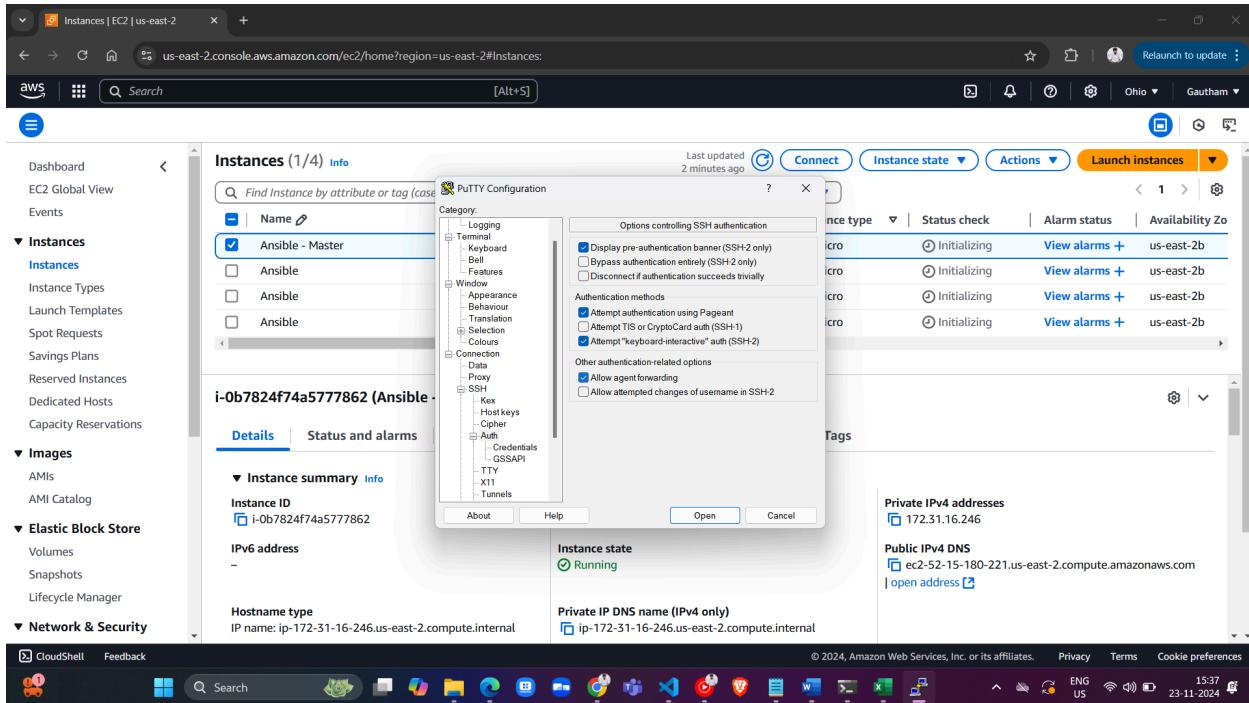
Add Private Key:

- Open **Pageant** and load your private key file.



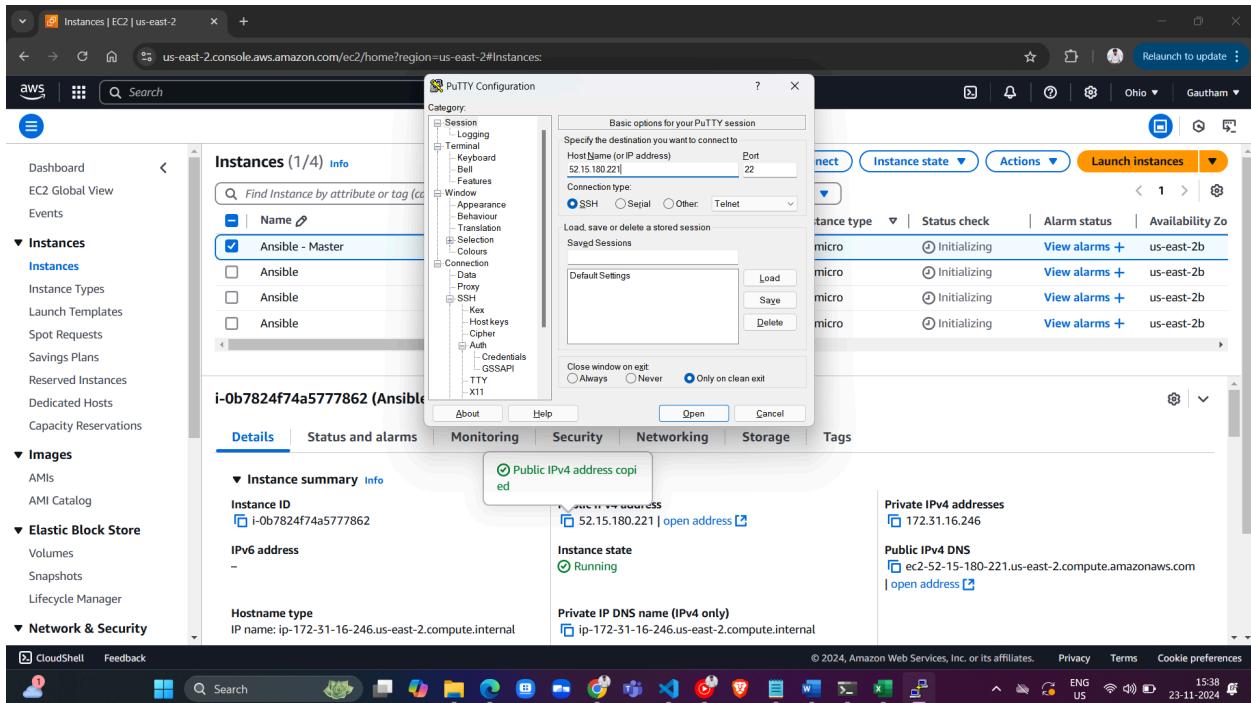
Enable SSH Authentication:

- Open PuTTY, navigate to **Connection > SSH > Auth**, and enable **Agent forwarding**.

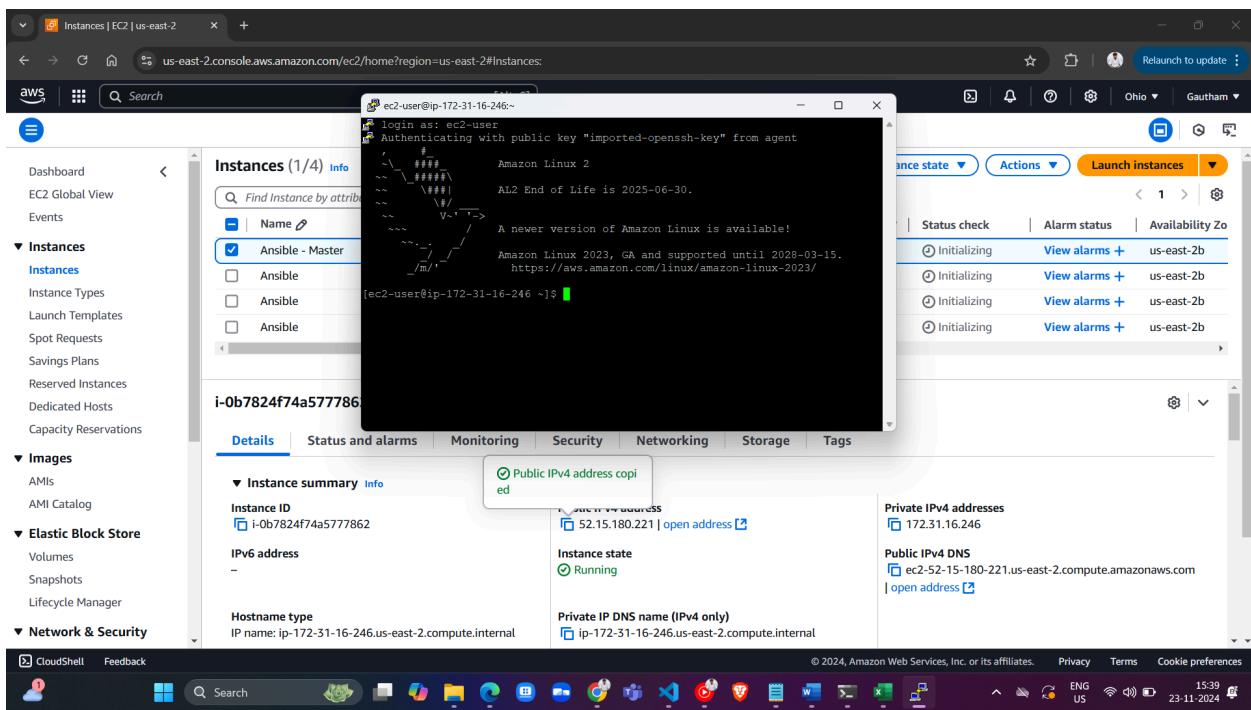


Connect:

- Enter the public IP address of the master instance in PuTTY and connect.



- Login as `ec2-user`.



Step 3. Verify Ansible Installation

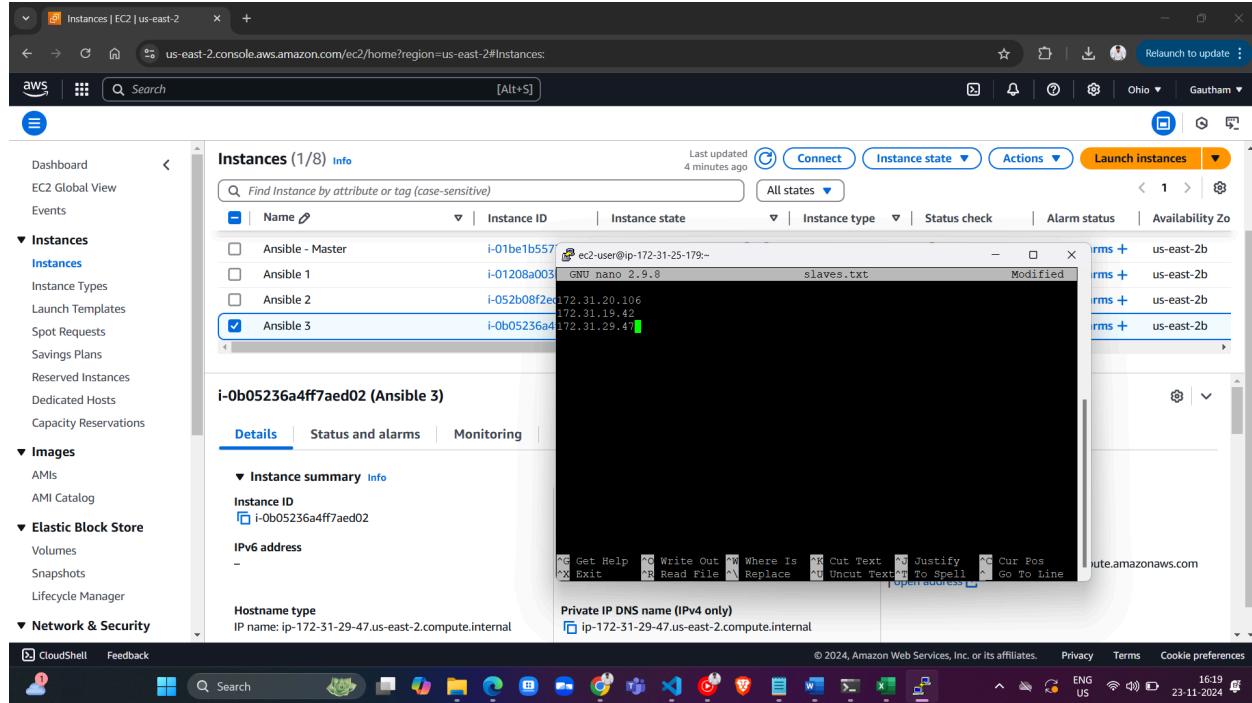
Prepare Inventory:

- Create a file named `slaves.txt` containing the private IPs of all slave instances.

`172.31.20.106`

`172.31.19.42`

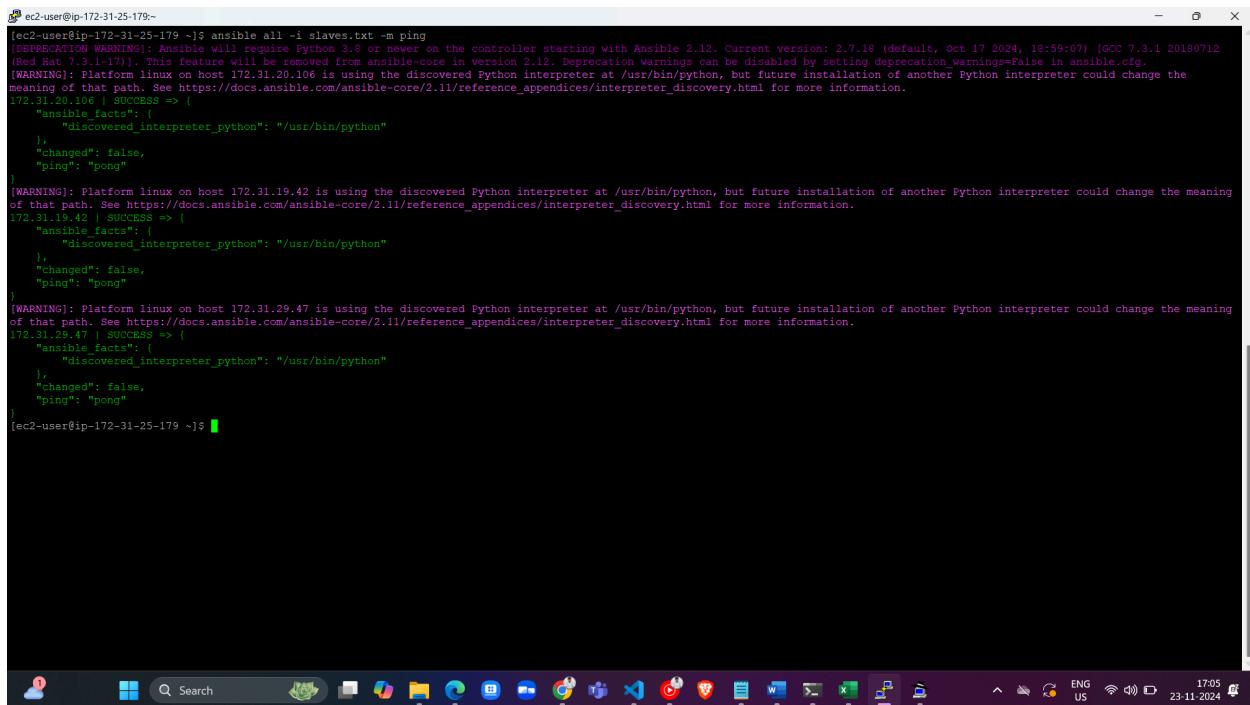
`172.31.29.47`



Ping Test:

- Run the following command to check connectivity:

```
ansible all -i slaves.txt -m ping
```



```
[ec2-user@ip-172-31-25-179 ~]$ ansible all -i slaves.txt -m ping
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller starting with Ansible 2.12. Current version: 2.7.18 (default, Oct 17 2024, 18:59:07) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]. This feature will be removed from ansible-core in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
[WARNING]: Platform linux on host 172.31.20.106 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.
172.31.20.106 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.19.42 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.
172.31.19.42 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.29.47 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.
172.31.29.47 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[ec2-user@ip-172-31-25-179 ~]$
```

- A successful message confirms Ansible is installed and properly configured.

Step 4. Create the Ansible Playbook

Assign Jenkins Role:

- Designate one slave instance as the Jenkins server for grouping.

Create Playbook:

- Write a playbook named `jenkins.yaml` to automate Jenkins installation:

```
- hosts: Jenkins_Server

become: yes

remote_user: ec2-user

vars:

  jenkins_port: JENKINS_PORT=8090

tasks:

  - name: Updating Packages
```

```
    yum:

      name: "*"

      state: present

  - name: Installing Java 17

    yum:

      name: java-17-amazon-corretto

      state: present

  - name: Import Jenkins Key

    rpm_key:

      key: https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

      state: present

  - name: Download Jenkins Repo

    get_url:

      url: https://pkg.jenkins.io/redhat-stable/jenkins.repo

      dest: /etc/yum.repos.d/jenkins.repo

      mode: '0777'

  - name: Install Jenkins

    yum:

      name: jenkins
```

```
state: present

-
- name: Start Jenkins

  service:

    name: jenkins

    state: started
```

- **Make 1 slave as Jenkin Server by grouping it**

A screenshot of a terminal window titled "ec2-user@ip-172-31-25-179:~". The window displays a list of IP addresses: 172.31.20.106, 172.31.19.42, [Jenkins_Server], and 172.31.29.47, each preceded by a small green arrow icon. Below this, there are several blank lines. The terminal is set against a dark background with white text. At the bottom of the screen, a Windows taskbar is visible, featuring the Start button, a search bar, and icons for various applications like File Explorer, Edge, and Excel. The system tray shows the date (23-11-2024), time (17:29), battery status, and network connectivity. A small green icon labeled "WGL" is also present in the taskbar.

```
ec2-user@ip-172-31-25-179:~$ hosts: Jenkins_Server
become: yes
remote_user: ec2-user
vars:
  jenkins_port: JENKINS_PORT=8090
tasks:
  - name: Updating Packages
    yum:
      name: "*"
      state: present

  - name: Installing Java 17
    yum:
      name: java-17-amazon-corretto
      state: present

  - name: Import Jenkins Key
    rpm_key:
      key: https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
      state: present

  - name: Download Jenkins Repo
    get_url:
      url: https://pkg.jenkins.io/redhat-stable/jenkins.repo
      dest: /etc/yum.repos.d/jenkins.repo
      mode: '0777'

  - name: Install Jenkins
    yum:
      name: jenkins
      state: present

  - name: Start Jenkins
    service:
      name: jenkins
      state: started
```

- **Execute Playbook:**

Run the playbook using:

```
ansible-playbook -i slaves.txt jenkins.yaml
```

```
[ec2-user@ip-172-31-25-179 ~]$ vi slaves.txt
[ec2-user@ip-172-31-25-179 ~]$ vi jenkins.yaml
[ec2-user@ip-172-31-25-179 ~]$ ansible-playbook -i slaves.txt jenkins.yaml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller starting with Ansible 2.12. Current version: 2.7.18 (default, Oct 17 2024, 18:59:07) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]. This feature will be removed from ansible-core in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.

PLAY [Jenkins_Server] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.29.47 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.

ok: [172.31.29.47]

TASK [Updating Packages] *****
ok: [172.31.29.47]

TASK [Installing Java 17] *****
changed: [172.31.29.47]

TASK [Import Jenkins Key] *****
changed: [172.31.29.47]

TASK [Download Jenkins Repo] *****
changed: [172.31.29.47]

TASK [Install Jenkins] *****
changed: [172.31.29.47]

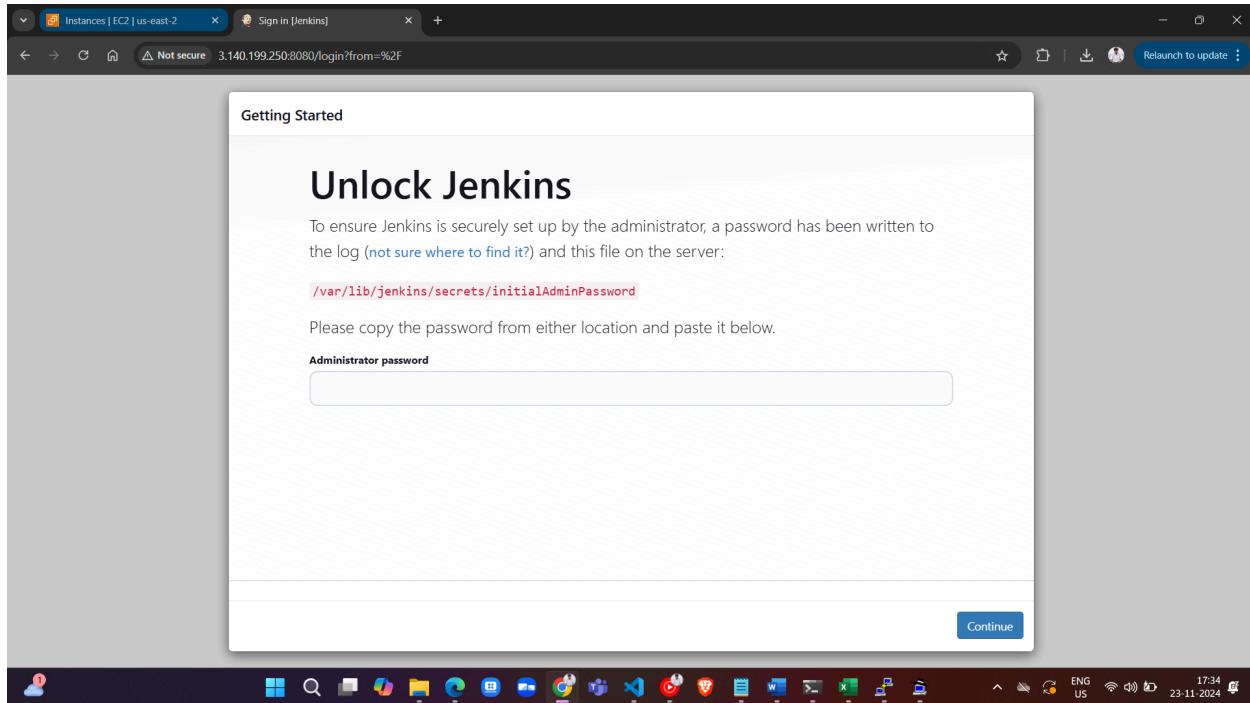
TASK [Start Jenkins] *****
changed: [172.31.29.47]

PLAY RECAP *****
172.31.29.47 : ok=7    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@ip-172-31-25-179 ~]$
```

- **Verify Jenkins:**

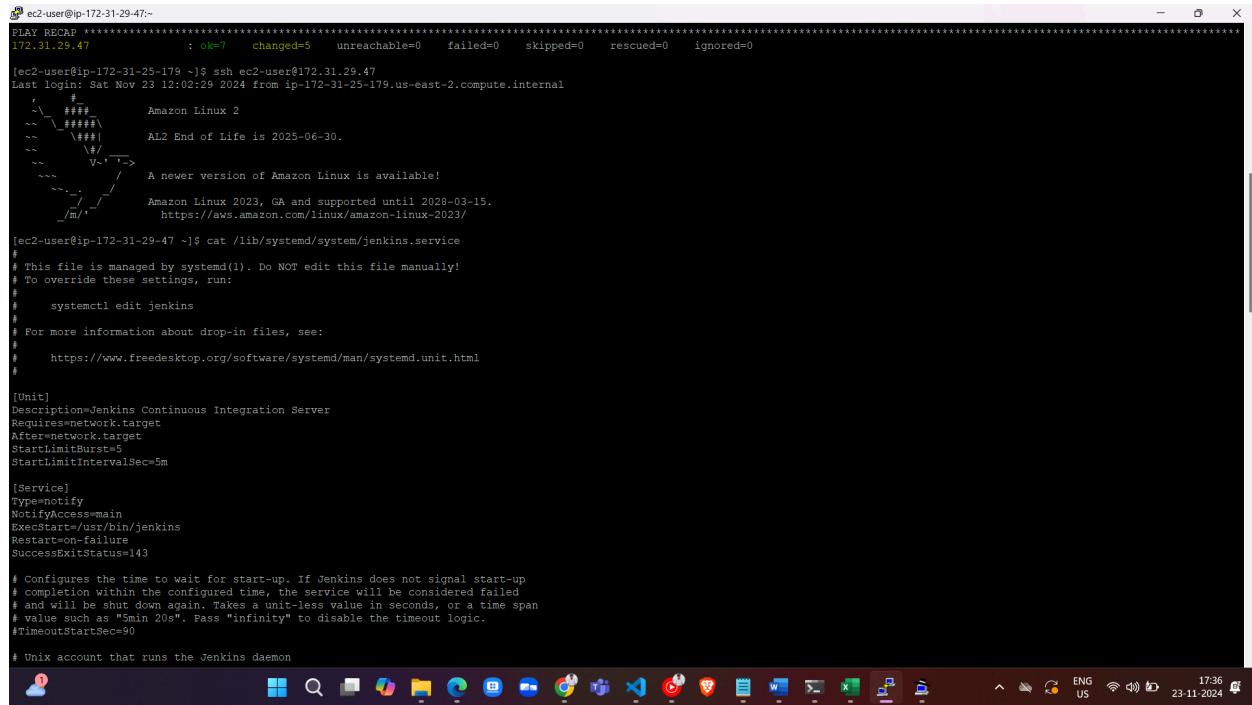
Open a browser and navigate to `<instance-ip>:8080` to access Jenkins.



Step 5. Create the Template File

Copy Jenkins Service File:

- SSH into the Jenkins slave instance:
`ssh ec2-user@<private_ip>`
- Verify the file `/lib/systemd/system/jenkins.service` exists.



```
PLAY RECAP ****
172.31.29.47 : ok=7    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
[ec2-user@ip-172-31-25-179 ~]$ ssh ec2-user@172.31.29.47
Last login: Sat Nov 23 12:02:29 2024 from ip-172-31-25-179.us-east-2.compute.internal
      _###_
     / \###\   Amazon Linux 2
    / \###\   AL2 End of Life is 2025-06-30.
   / \###\   V~*-->
  / \###\   A newer version of Amazon Linux is available!
 / \###\   Amazon Linux 2023, GA and supported until 2028-03-15.
 / \###\   https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-29-47 ~]$ cat /lib/systemd/system/jenkins.service
# This file is managed by systemd(1). Do NOT edit this file manually!
# To override these settings, run:
#   systemctl edit jenkins
# For more information about drop-in files, see:
#   https://www.freedesktop.org/software/systemd/man/systemd.unit.html
#
[Unit]
Description=Jenkins Continuous Integration Server
Requires=network.target
After=network.target
StartLimitBurst=5
StartLimitIntervalSec=5m

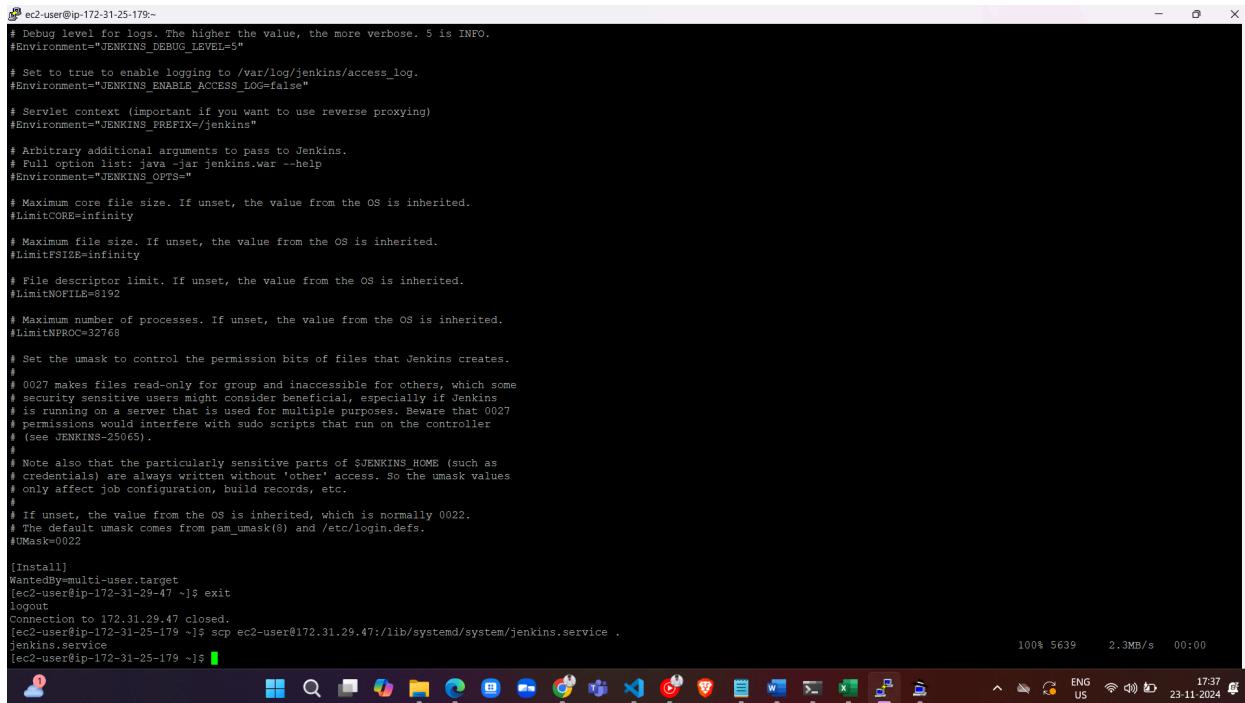
[Service]
Type=notify
NotifyAccess=main
ExecStart=/usr/bin/jenkins
Restart=on-failure
SuccessExitStatus=143

# Configures the time to wait for start-up. If Jenkins does not signal start-up
# completion within the configured time, the service will be considered failed
# and will be shut down again. Takes a unit-less value in seconds, or a time span
# value such as "3min 20s". Pass "infinity" to disable the timeout logic.
#TimeoutStartSec=90

# Unix account that runs the Jenkins daemon
[ec2-user@ip-172-31-29-47 ~]$
```

- Copy the file to the master instance:

```
scp ec2-user@<private_ip>:/lib/systemd/system/jenkins.service .
```



```

ec2-user@ip-172-31-25-179:~#
# Debug level for logs. The higher the value, the more verbose. 5 is INFO.
#Environment="JENKINS_DEBUG_LEVEL=5"

# Set to true to enable logging to /var/log/jenkins/access.log.
#Environment="JENKINS_ENABLE_ACCESS_LOG=false"

# Servlet context (important if you want to use reverse proxying)
#Environment="JENKINS_PREFIX=/jenkins"

# Arbitrary additional arguments to pass to Jenkins.
# Full option list: java -jar jenkins.war --help
#Environment="JENKINS_OPTS="

# Maximum core file size. If unset, the value from the OS is inherited.
#LimitCORE=infinity

# Maximum file size. If unset, the value from the OS is inherited.
#LimitFSIZE=infinity

# File descriptor limit. If unset, the value from the OS is inherited.
#LimitNOFILE=8192

# Maximum number of processes. If unset, the value from the OS is inherited.
#LimitNPROC=32768

# Set the umask to control the permission bits of files that Jenkins creates.
#
# 0027 makes files read-only for group and inaccessible for others, which some
# security sensitive users might consider beneficial, especially if Jenkins
# is running on a server that is used for multiple purposes. Beware that 0027
# permissions would interfere with sudo scripts that run on the controller
# (see JENKINS-25065).

# Note also that the particularly sensitive parts of $JENKINS_HOME (such as
# credentials) are always written without 'other' access. So the umask values
# only affect job configuration, build records, etc.

# If unset, the value from the OS is inherited, which is normally 0022.
# The default umask comes from pam_umask(8) and /etc/login.defs.
#UMask=0022

[Install]
WantedBy=multi-user.target
[ec2-user@ip-172-31-25-47 ~]$ exit
logout
Connection to 172.31.29.47 closed.
[ec2-user@ip-172-31-25-179 ~]$ scp ec2-user@172.31.29.47:/lib/systemd/system/jenkins.service .
jenkins.service
[ec2-user@ip-172-31-25-179 ~]$ 

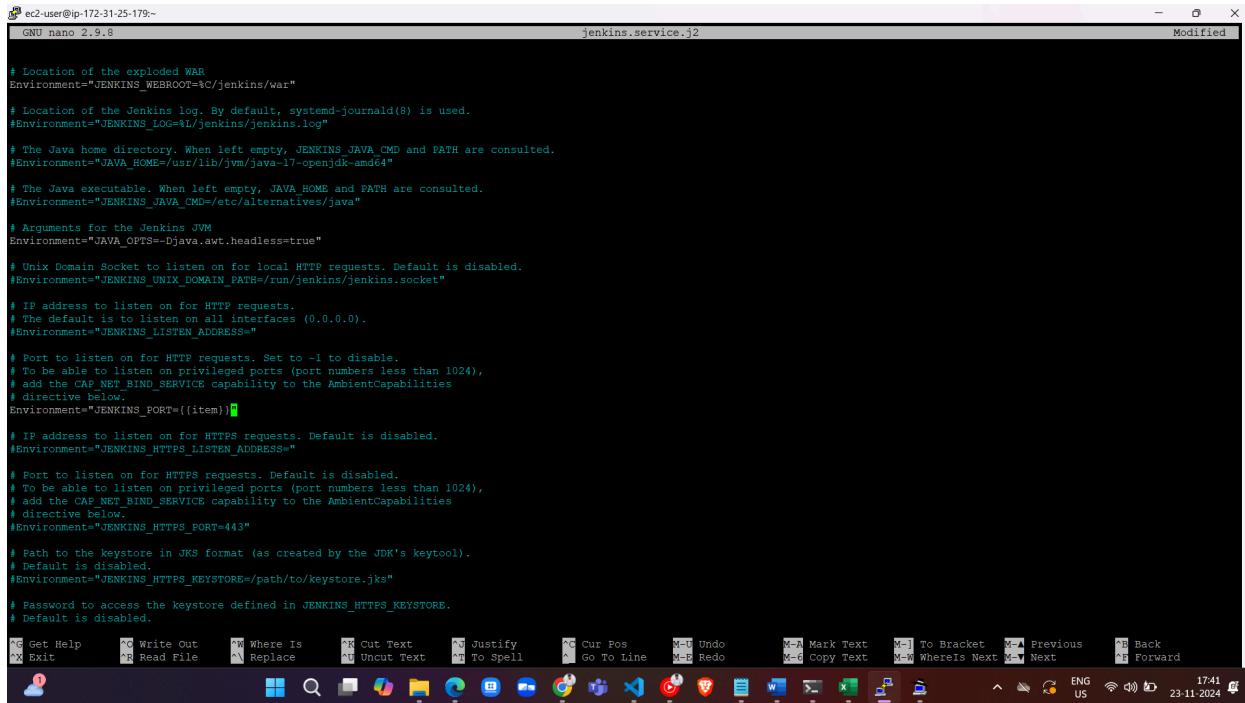
```

100% 5639 2.3MB/s 00:00

ENG US 17:37 23-11-2024

Customize File:

- Modify the file by adding templated variables surrounded by {{item}}.



```

ec2-user@ip-172-31-25-179:~#
GNU nano 2.9.8 jenkins.service.j2 Modified

# Location of the exploded WAR
Environment="JENKINS_WEBROOT=/C/jenkins/war"

# Location of the Jenkins log. By default, systemd-journald(8) is used.
#Environment="JENKINS_LOG=$L/jenkins/jenkins.log"

# The Java home directory. When left empty, JENKINS JAVA_CMD and PATH are consulted.
#Environment="JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64"

# The Java executable. When left empty, JAVA_HOME and PATH are consulted.
#Environment="JENKINS JAVA_CMD=/etc/alternatives/java"

# Arguments for the Jenkins JVM
Environment="JAVA_OPTS=-Djava.awt.headless=true"

# Unix Domain Socket to listen on for local HTTP requests. Default is disabled.
#Environment="JENKINS_UNIX_DOMAIN_PATH=/run/jenkins/jenkins.socket"

# IP address to listen on for HTTP requests.
# The default is to listen on all interfaces (0.0.0.0).
#Environment="JENKINS_LISTEN_ADDRESS="

# Port to listen on for HTTP requests. Set to -1 to disable.
# To be able to listen on privileged ports (port numbers less than 1024),
# add the CAP_NET_BIND_SERVICE capability to the AmbientCapabilities
# directive below.
Environment="JENKINS_PORT={{item}}"

# IP address to listen on for HTTPS requests. Default is disabled.
#Environment="JENKINS_HTTPS_LISTEN_ADDRESS="

# Port to listen on for HTTPS requests. Default is disabled.
# To be able to listen on privileged ports (port numbers less than 1024),
# add the CAP_NET_BIND_SERVICE capability to the AmbientCapabilities
# directive below.
#Environment="JENKINS_HTTPS_PORT=443"

# Path to the keystore in JKS format (as created by the JDK's keytool).
# Default is disabled.
#Environment="JENKINS_HTTPS_KEYSTORE=/path/to/keystore.jks"

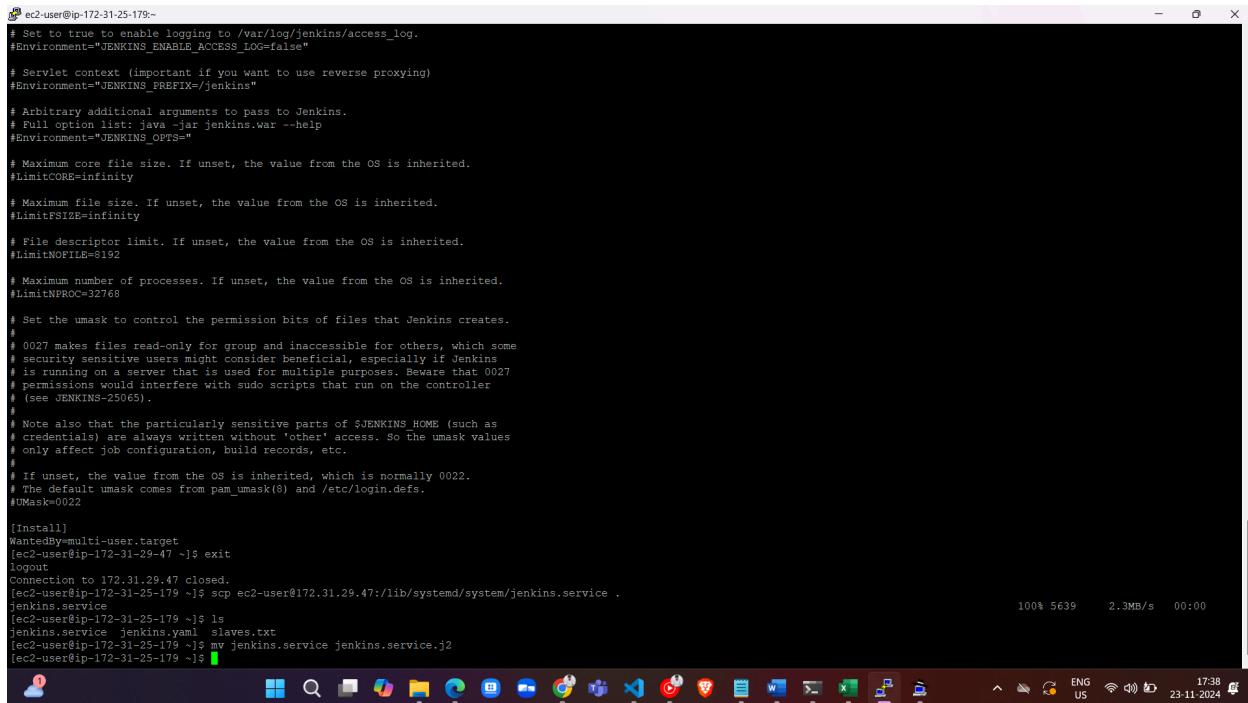
# Password to access the keystore defined in JENKINS_HTTPS_KEYSTORE.
# Default is disabled.

```

Modified

17:37 23-11-2024

- Save the file as [jenkins.service.j2](#)



```
ec2-user@ip-172-31-25-179:~$ cat jenkins.j2
# Set to true to enable logging to /var/log/jenkins/access_log.
#Environment="JENKINS_ENABLE_ACCESS_LOG=false"

# Servlet context (important if you want to use reverse proxying)
#Environment="JENKINS_PREFIX=/jenkins"

# Arbitrary additional arguments to pass to Jenkins.
# Full option list: java -jar Jenkins.war --help
#Environment="JENKINS_OPTS="

# Maximum core file size. If unset, the value from the OS is inherited.
#LimitCORE=infinity

# Maximum file size. If unset, the value from the OS is inherited.
#LimitFSIZE=infinity

# File descriptor limit. If unset, the value from the OS is inherited.
#LimitNOFILE=8192

# Maximum number of processes. If unset, the value from the OS is inherited.
#LimitNPROC=32768

# Set the umask to control the permission bits of files that Jenkins creates.
#
# 0027 makes files read-only for group and inaccessible for others, which some
# security sensitive users might consider beneficial, especially if Jenkins
# is running on a server that is used for multiple purposes. Beware that 0027
# permissions would interfere with sudo scripts that run on the controller
# (see JENKINS-25065).
#
# Note also that the particularly sensitive parts of $JENKINS_HOME (such as
# credentials) are always written without 'other' access. So the umask values
# only affect job configuration, build records, etc.
#
# If unset, the value from the OS is inherited, which is normally 0022.
# The default umask comes from pam_umask(8) and /etc/login.defs.
#UMask=0022

[Install]
WantedBy=multi-user.target
[ec2-user@ip-172-31-29-47 ~]$ exit
logout
Connection to 172.31.29.47 closed.
[ec2-user@ip-172-31-29-47 ~]$ scp ec2-user@172.31.29.47:/lib/systemd/system/jenkins.service .
jenkins.service
[ec2-user@ip-172-31-25-179 ~]$ ls
jenkins.service jenkins.yaml slaves.txt
[ec2-user@ip-172-31-25-179 ~]$ mv jenkins.service jenkins.service.j2
[ec2-user@ip-172-31-25-179 ~]$ 
```

Step 6. Apply the Template File

Modify Playbook:

- Add a task to `jenkins.yaml` to apply the template:

```
- name: Change Jenkins Port
  template:
    src: ./jenkins.service.j2
    dest: /lib/systemd/system/jenkins.service
    mode: '0777'
```

```

ec2-user@ip-172-31-25-179:~$ nano 2.9.8
GNU nano 2.9.8
jenkins.yaml

hosts: Jenkins_Server
become: yes
remote_user: ec2-user
vars:
  item: 8040
tasks:
  - name: Updating Packages
    yum:
      name: "*"
      state: present

  - name: Installing Java 17
    yum:
      name: java-17-amazon-corretto
      state: present

  - name: Import Jenkins Key
    rpm_key:
      key: https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
      state: present

  - name: Download Jenkins Repo
    get_url:
      url: https://pkg.jenkins.io/redhat-stable/jenkins.repo
      dest: /etc/yum.repos.d/jenkins.repo
      mode: '0777'

  - name: Install Jenkins
    yum:
      name: jenkins
      state: present

  - name: changing jenkins Parameters
    template:
      src: ./jenkins.service.j2
      dest: /usr/lib/systemd/system/jenkins.service
      mode: '0777'

  - name: Start Jenkins
    service:
      name: jenkins
      state: started

```

Execute Playbook:

- Run the modified playbook:

```
ansible-playbook -i slaves.txt jenkins.yaml
```

```

ec2-user@ip-172-31-25-179:~$ ansible-playbook -i slaves.txt jenkins.yaml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller starting with Ansible 2.12. Current version: 2.7.18 (default, Oct 17 2024, 18:59:07) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]. This feature will be removed from ansible-core in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.

PLAY [Jenkins_Server] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.29.47 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.11/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.29.47]

TASK [Updating Packages] ****
ok: [172.31.29.47]

TASK [Installing Java 17] ****
ok: [172.31.29.47]

TASK [Import Jenkins Key] ****
ok: [172.31.29.47]

TASK [Download Jenkins Repo] ****
ok: [172.31.29.47]

TASK [Install Jenkins] ****
ok: [172.31.29.47]

TASK [changing jenkins Parameters] ****
changed: [172.31.29.47]

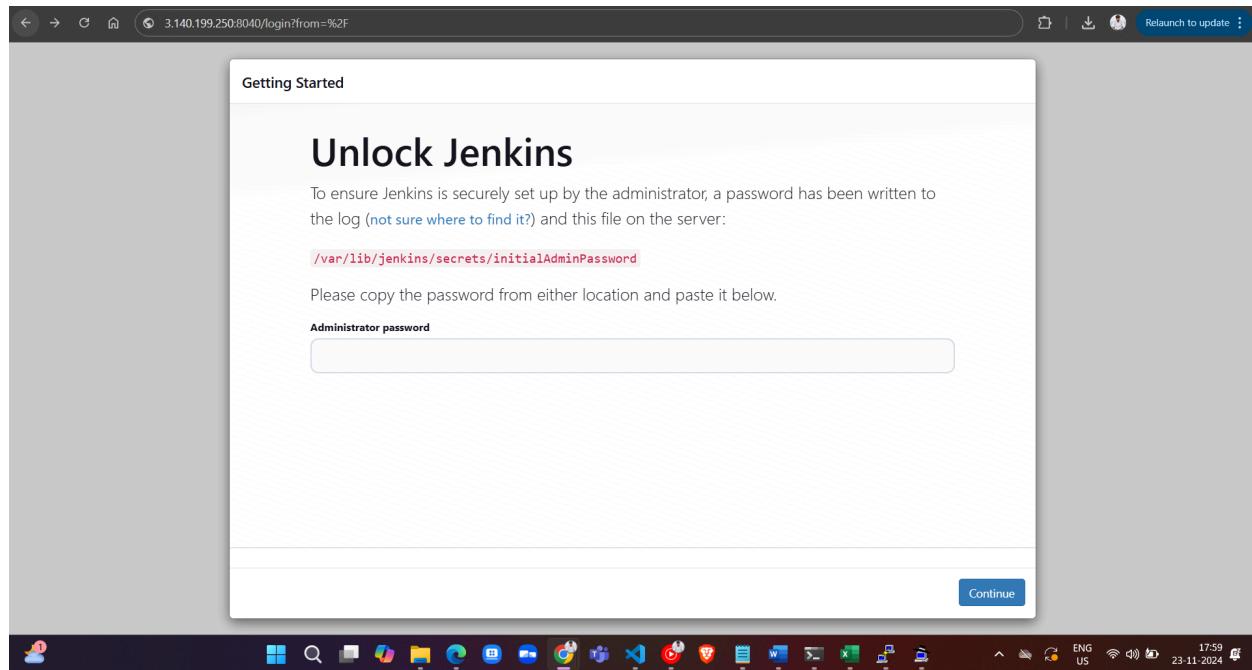
TASK [Start Jenkins] ****
ok: [172.31.29.47]

PLAY RECAP ****
172.31.29.47 : ok=6    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Verify Changes:

- Restart Jenkins to apply the configuration changes and confirm the new port is active



Conclusion

This project demonstrates a streamlined approach to deploying Jenkins using Ansible, leveraging automation to manage multiple EC2 instances effectively. By configuring a playbook, creating reusable templates, and ensuring secure connectivity, we simplified the installation and customization of Jenkins across a distributed environment. This method not only reduces manual effort but also ensures consistency and scalability for future deployments. With Ansible's robust capabilities, managing infrastructure becomes efficient, reliable, and adaptable to various use cases.