# **Ansible Proof of Concept (POC)**

**Objective:**

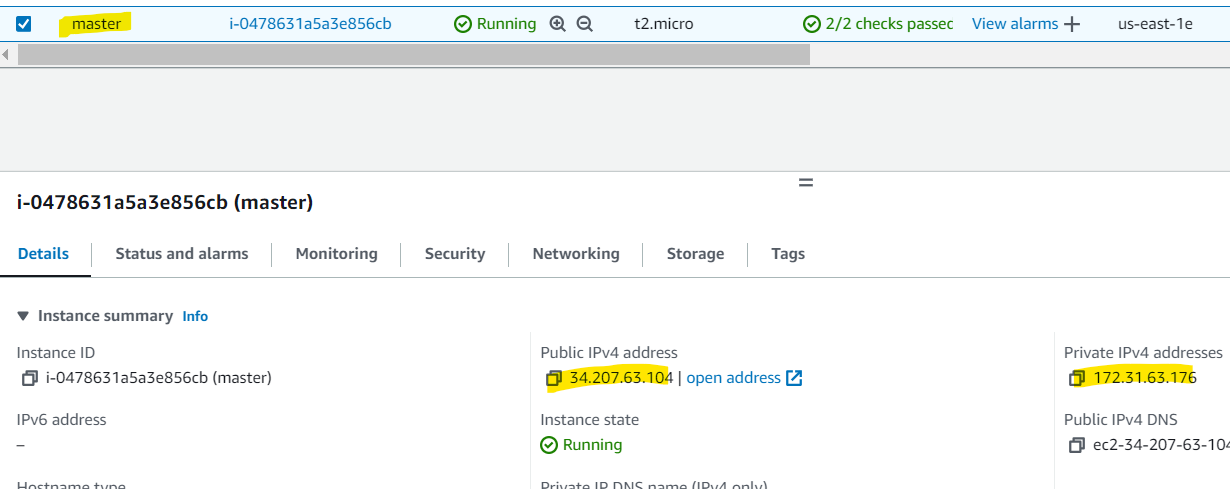
To demonstrate the capabilities of Ansible for automating task management, configuration, and application deployment across multiple servers using Ansible playbooks.

**Prerequisites:**

* **Master Node (Control Node):** A server where Ansible will be installed and run.
* **Slave Nodes (Managed Nodes):** Servers that Ansible will manage, accessible over SSH from the Master Node.
* **Basic knowledge of Linux command-line operations and SSH.**

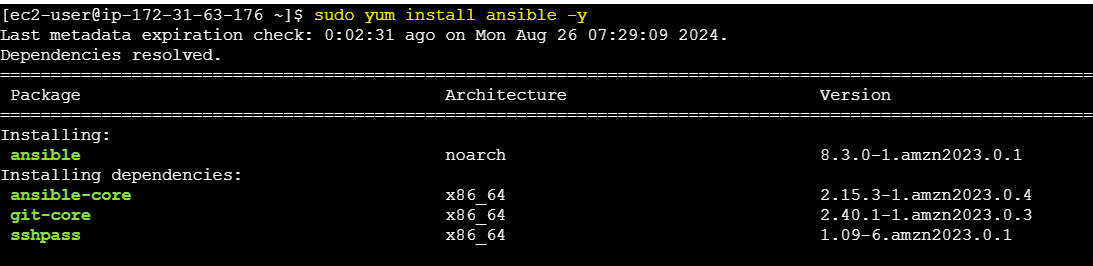
**Setup and Configuration:**

1. Setting Up the Master Node. On master ansible can be installed.

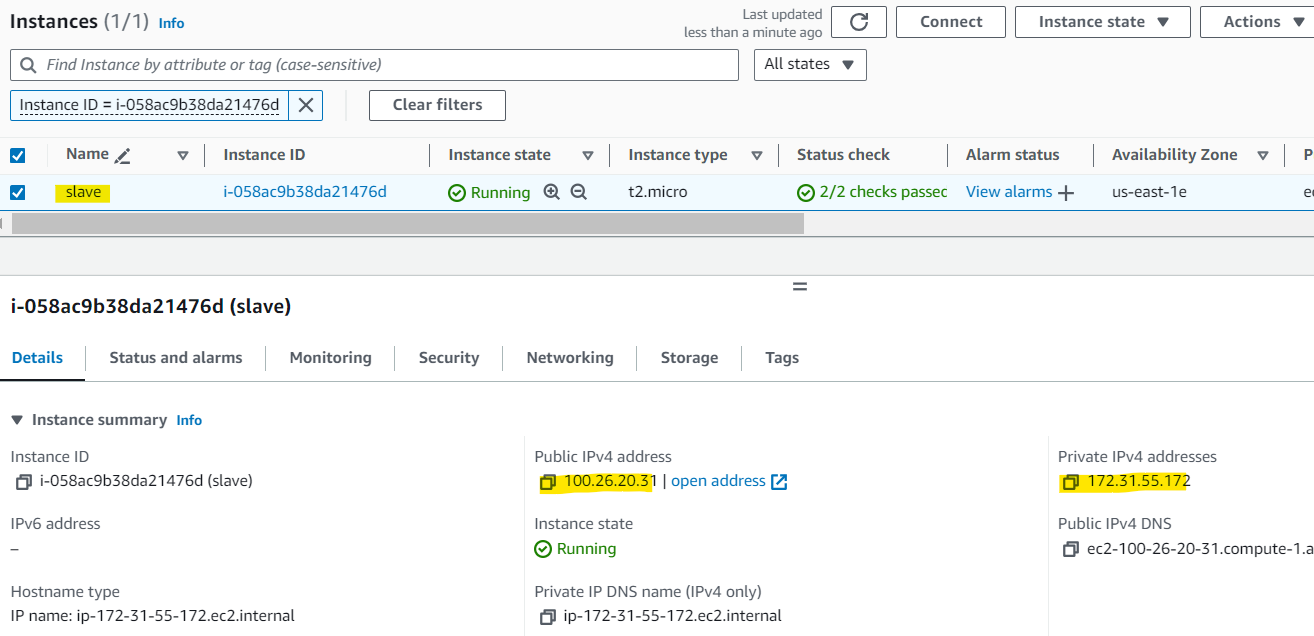


On master -$sudo yum update -y

-$sudo yum install ansible –y

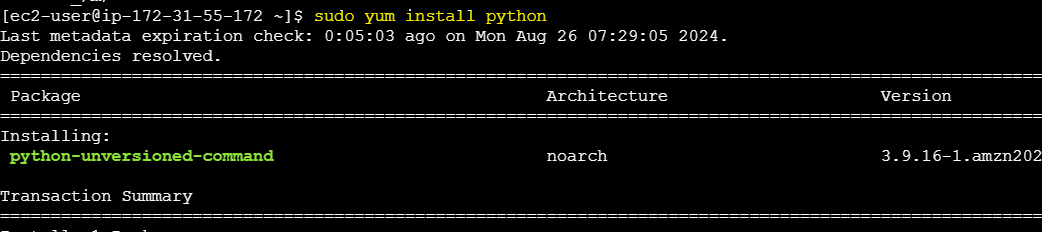


Instance 2- U**pdate the System and Install Python:** Python is required for Ansible to operate on managed nodes.



On slave -$sudo yum update -y

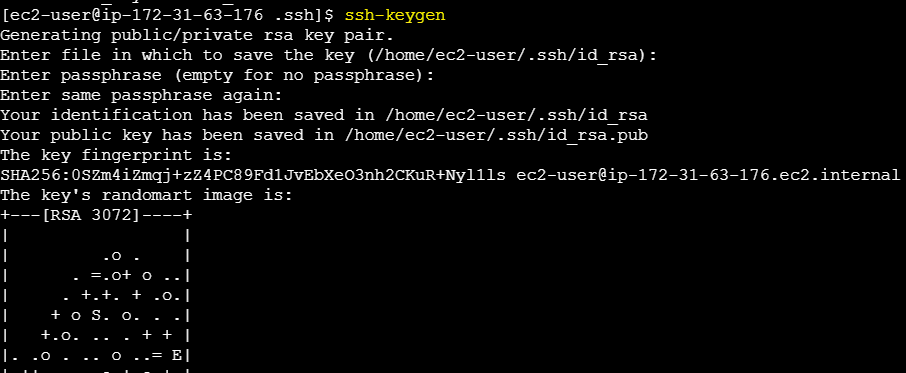
-$sudo yum install python



1. Generate public key on master to access the managed node through ssh port.

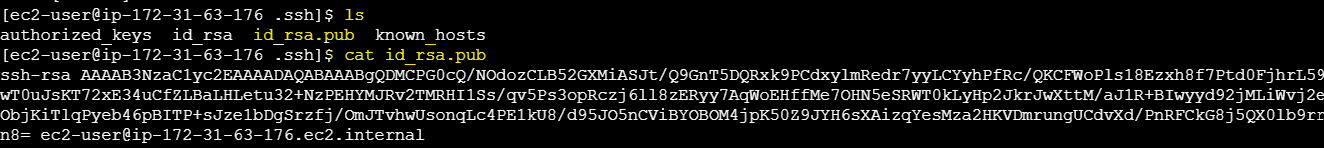
$cd /.ssh

$ssh-keygen



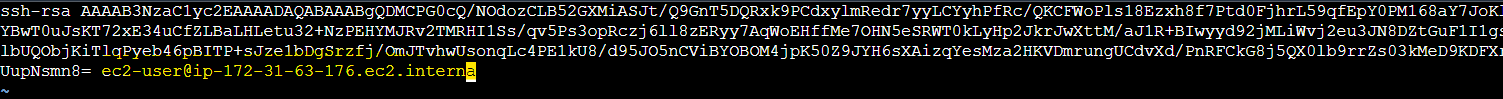
1. Above command generates a pub key. Open file (.pub) copy and paste it in the slave authorized keys to access.

$cat id\_rsa.pub



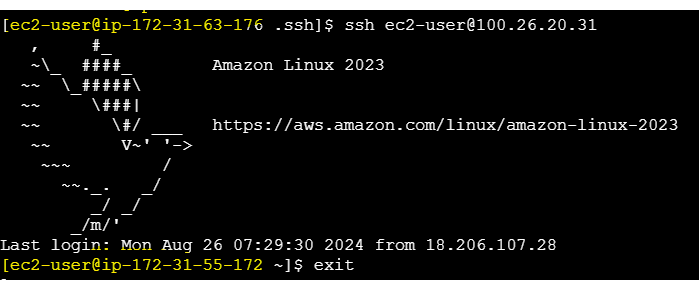
1. Connect to slave and open below path to past the public key

$vi /ssh/authorized\_keys



1. Now you can connect to slave from master

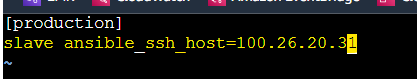
$ssh ec2-user@<ip\_address>



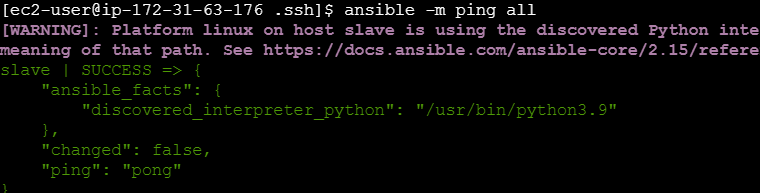
You can also check by using ping command

$sudo vi /etc/ansible/hosts

1. Target the node by configuring hosts file.



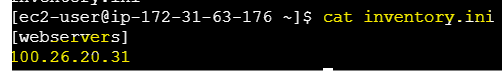
$ansible –m ping all



**Configuring the play books**

1. **Create an Ansible Inventory**

The inventory file lists all the managed nodes. You can use a simple INI format or YAML format to create inventory.



A playbook defines a series of tasks to be executed on your managed nodes.

**Playbook:** Install and configure httpd on web servers.

$vi playbook.yml

---

- name: Configure Apache HTTP Server

hosts: webservers

become: yes

tasks:

- name: Ensure httpd is installed

package:

name: httpd

state: present

- name: Ensure httpd service is running

service:

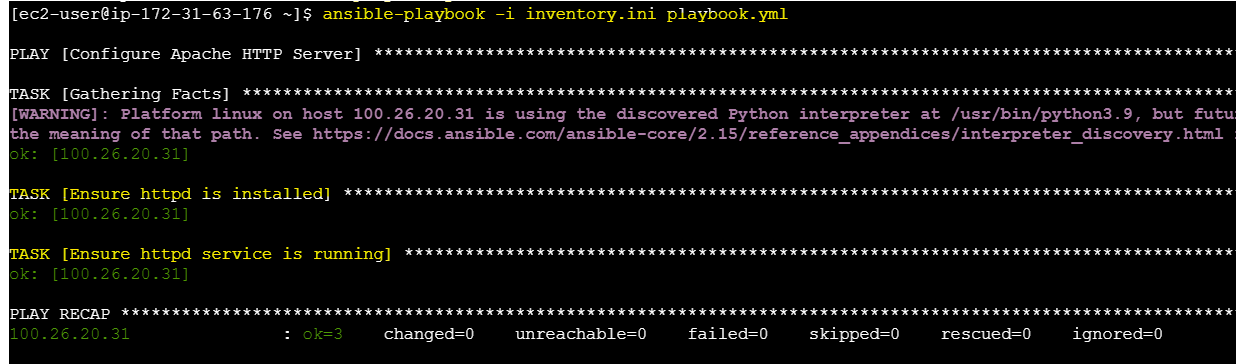
name: httpd

state: started

enabled: yes

1. Execute the playbook using the ansible-playbook command:

$ansible-playbook -i inventory.ini playbook.yml

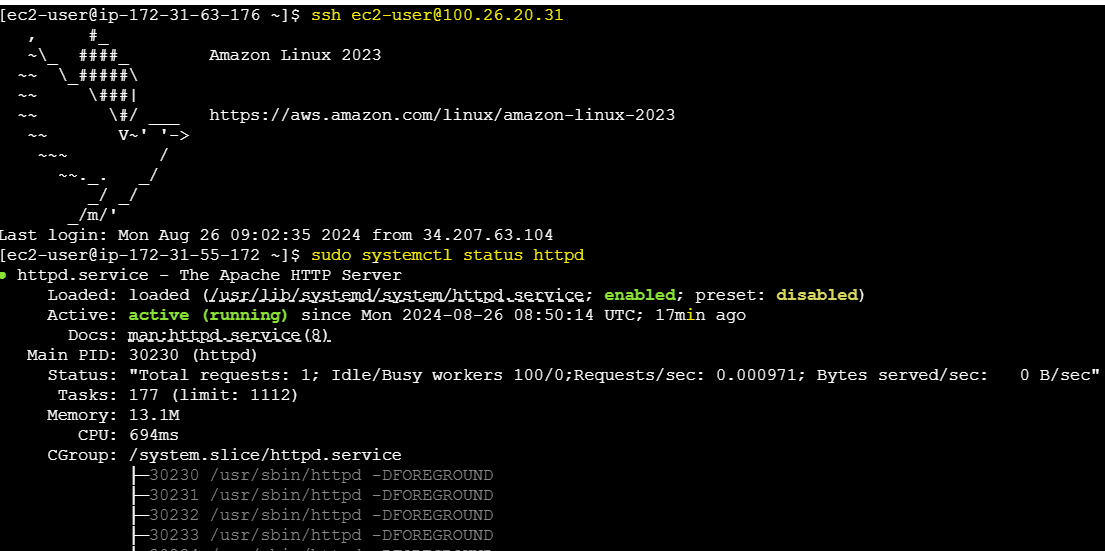


1. Verifying Results

Connect to SSH into the managed node and verify that httpd is installed and running:

Connect to slave $ ssh [ec2-user@100.26.20.31](mailto:ec2-user@100.26.20.31)

$ sudo systemctl status httpd



**Playbook 2**: Install and configure Jenkins deployment

---

- hosts: webservers

become: yes

vars:

java\_packages:

- java-17-amazon-corretto-devel

jenkins\_packages:

- jenkins

tasks:

- name: Download Jenkins repository file

get\_url:

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

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

become: true

- name: Import Jenkins-CI key

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

become: true

- name: Install Java

yum:

name: "{{ java\_packages }}"

state: present

become: true

- name: Install Jenkins

yum:

name: "{{ jenkins\_packages }}"

state: present

become: true

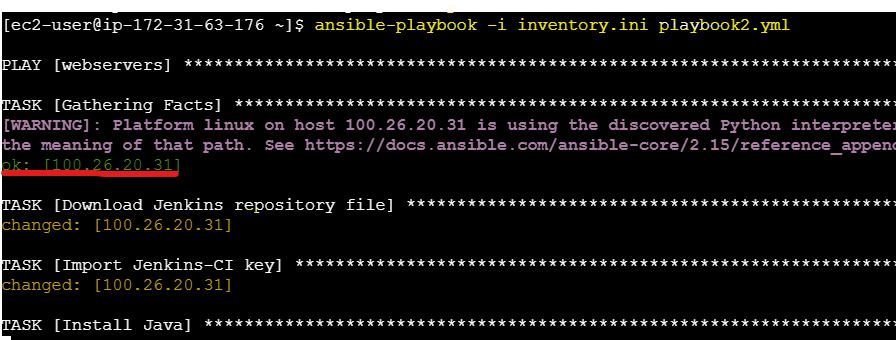
- name: Start Jenkins service

service:

name: jenkins

state: started

become: true



**Verifying Results:**

Connect to SSH into the managed node and verify that httpd is installed and running:

Connect to slave $ ssh [ec2-user@100.26.20.31](mailto:ec2-user@100.26.20.31)

$ sudo systemctl status Jenkins

