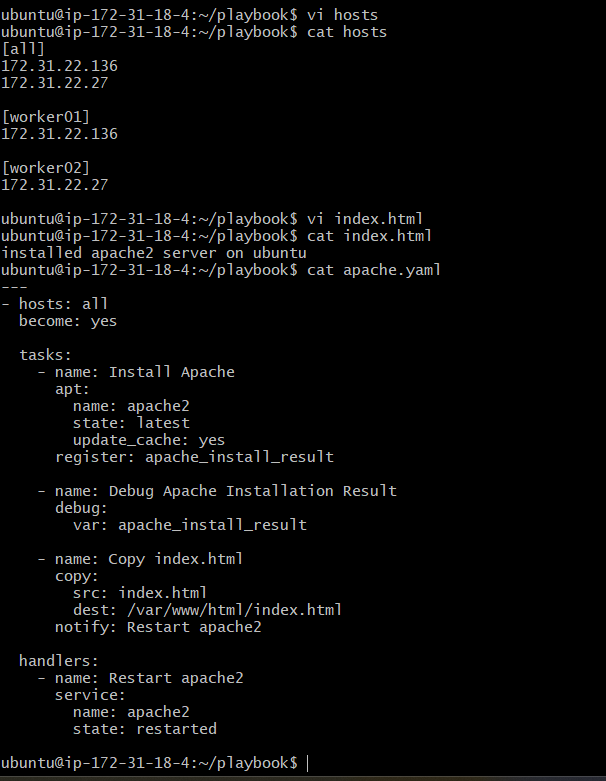
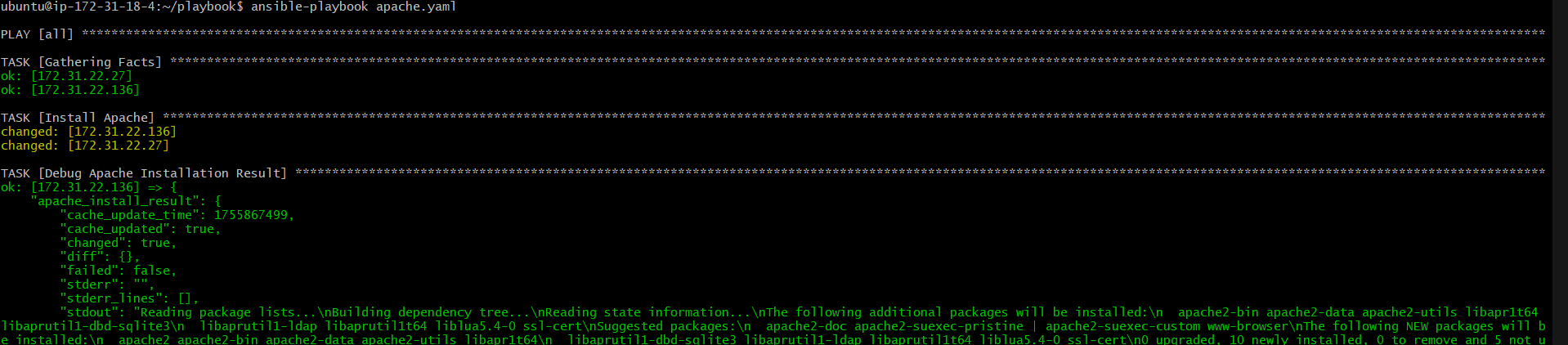
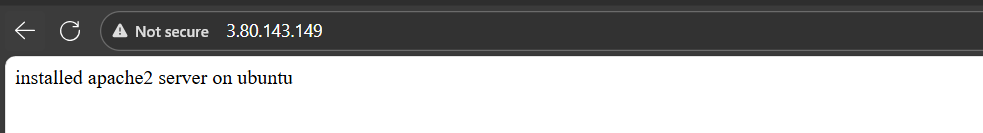
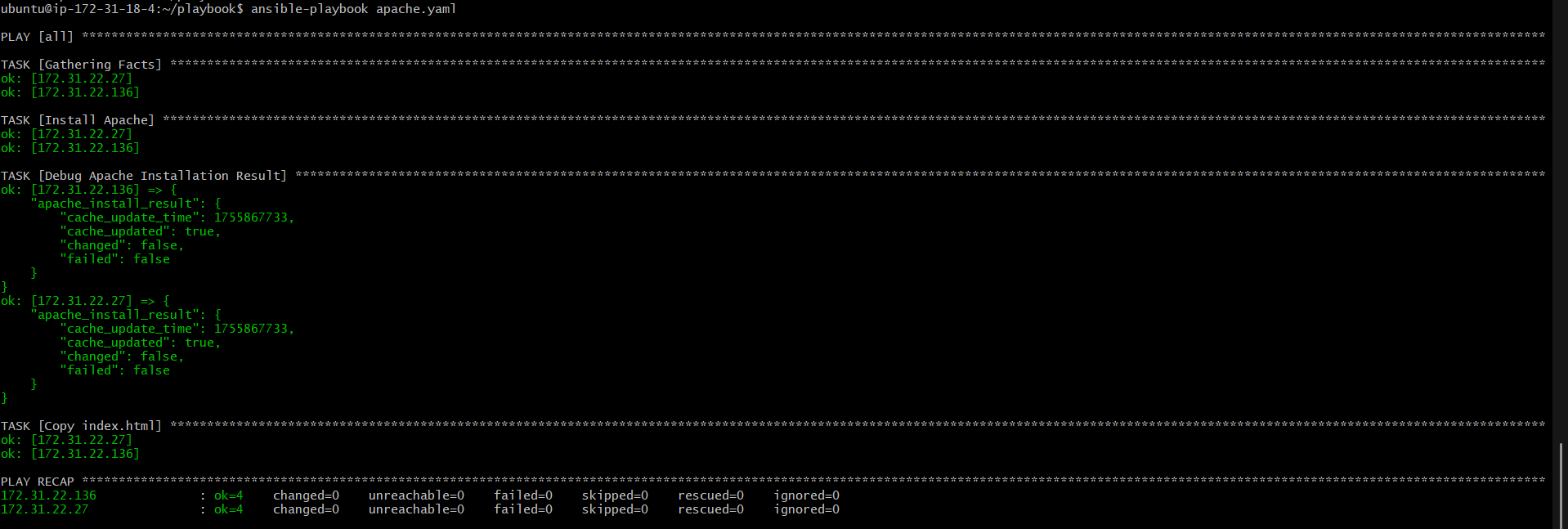
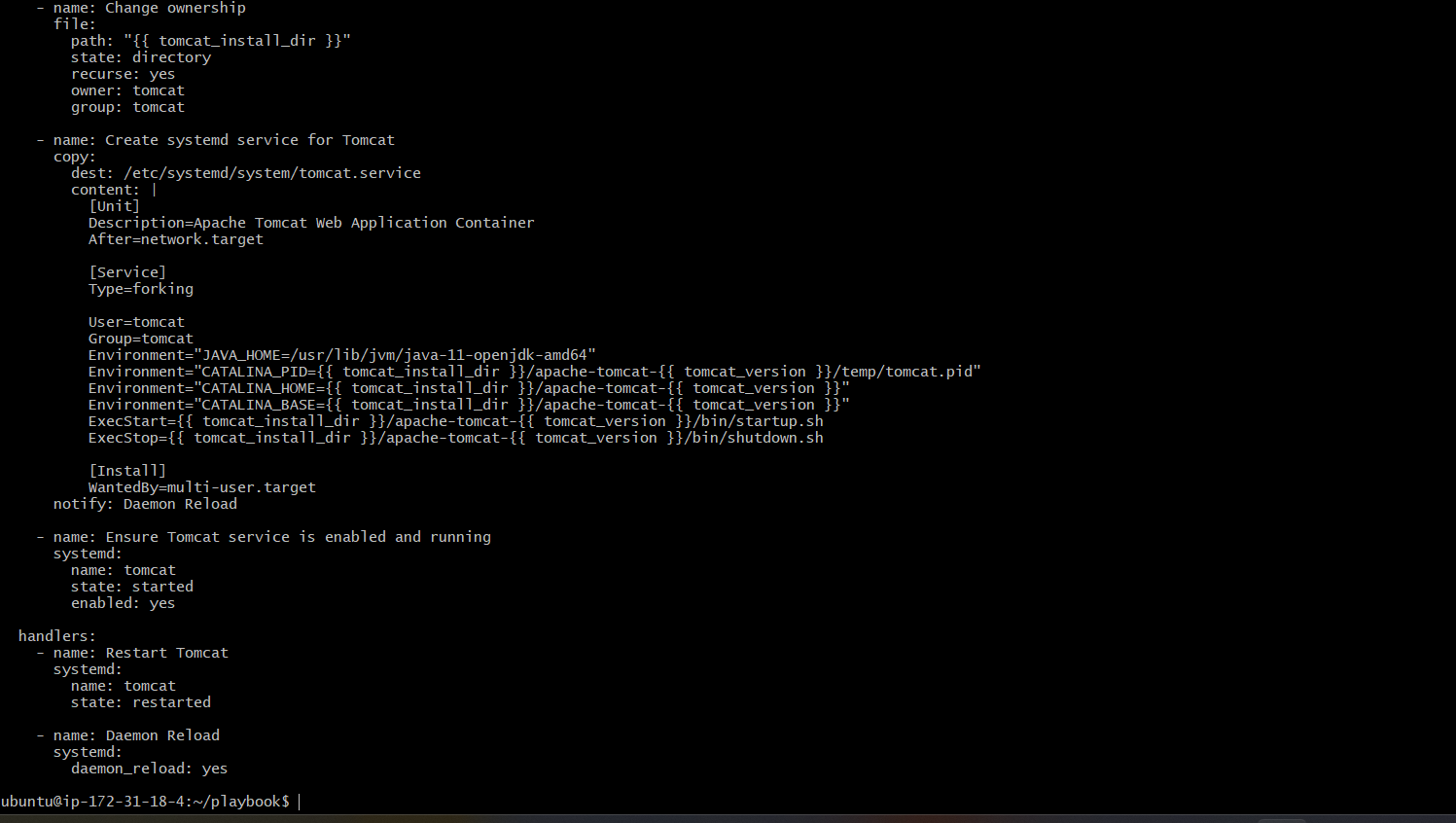
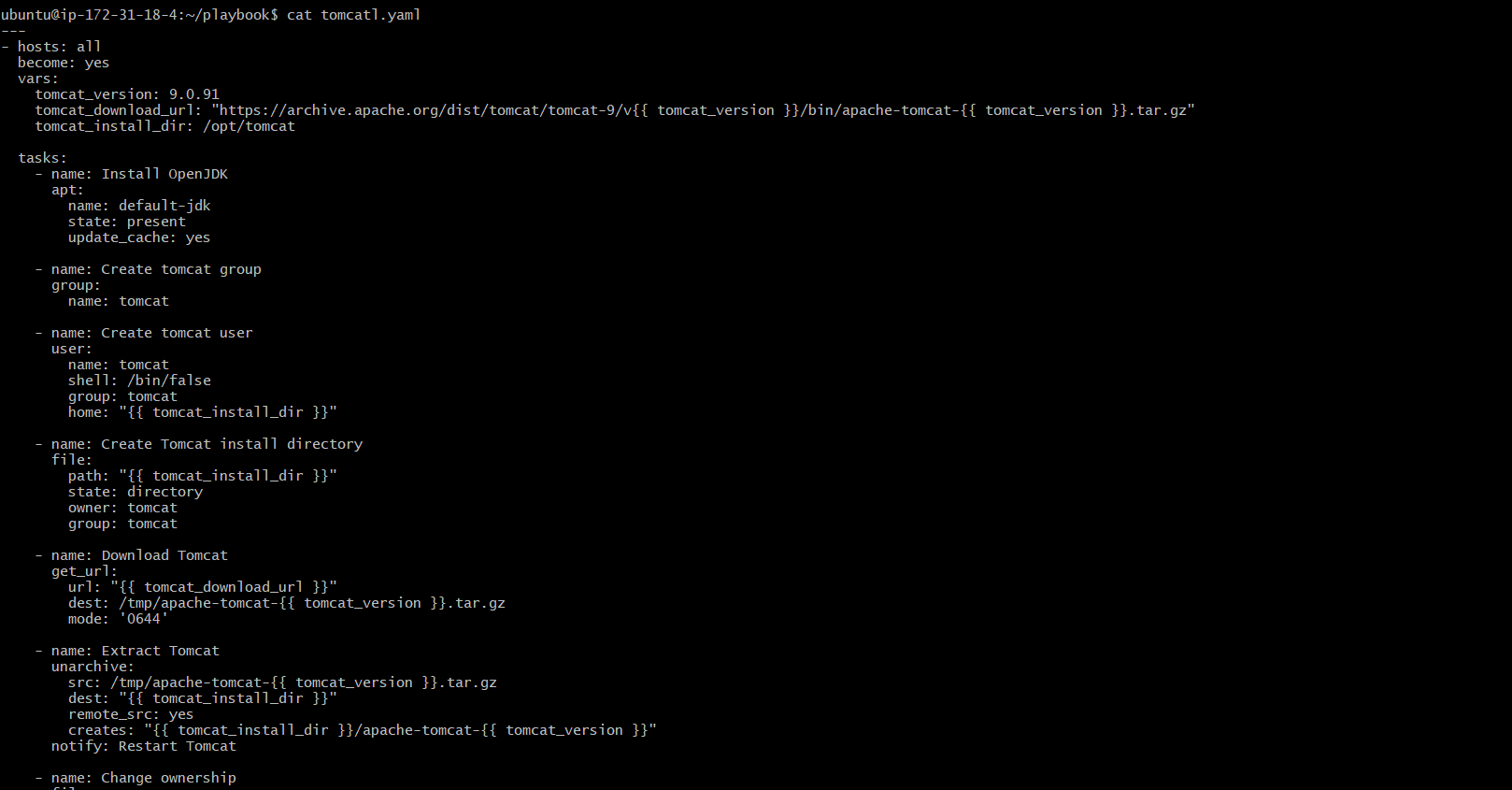
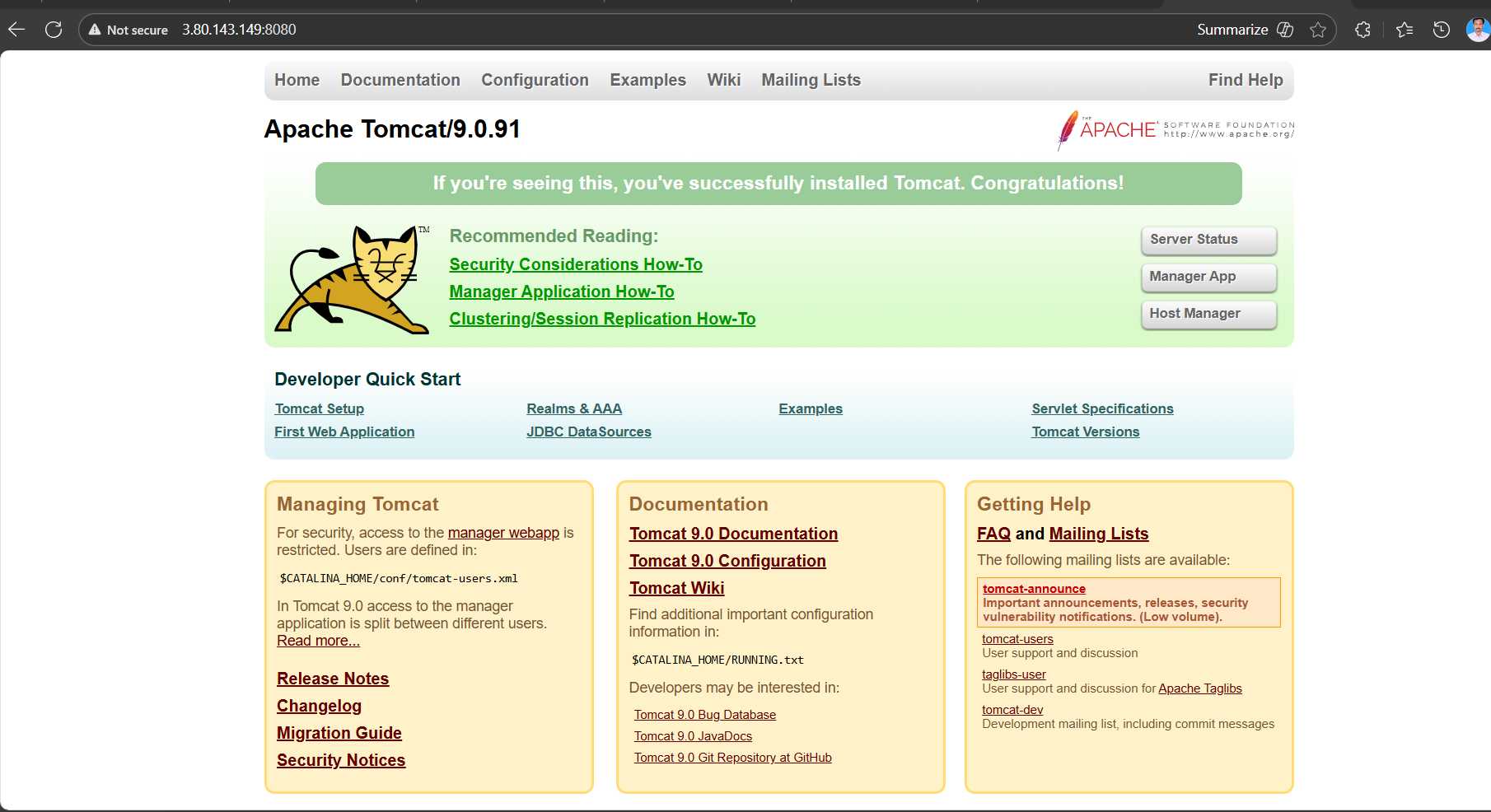
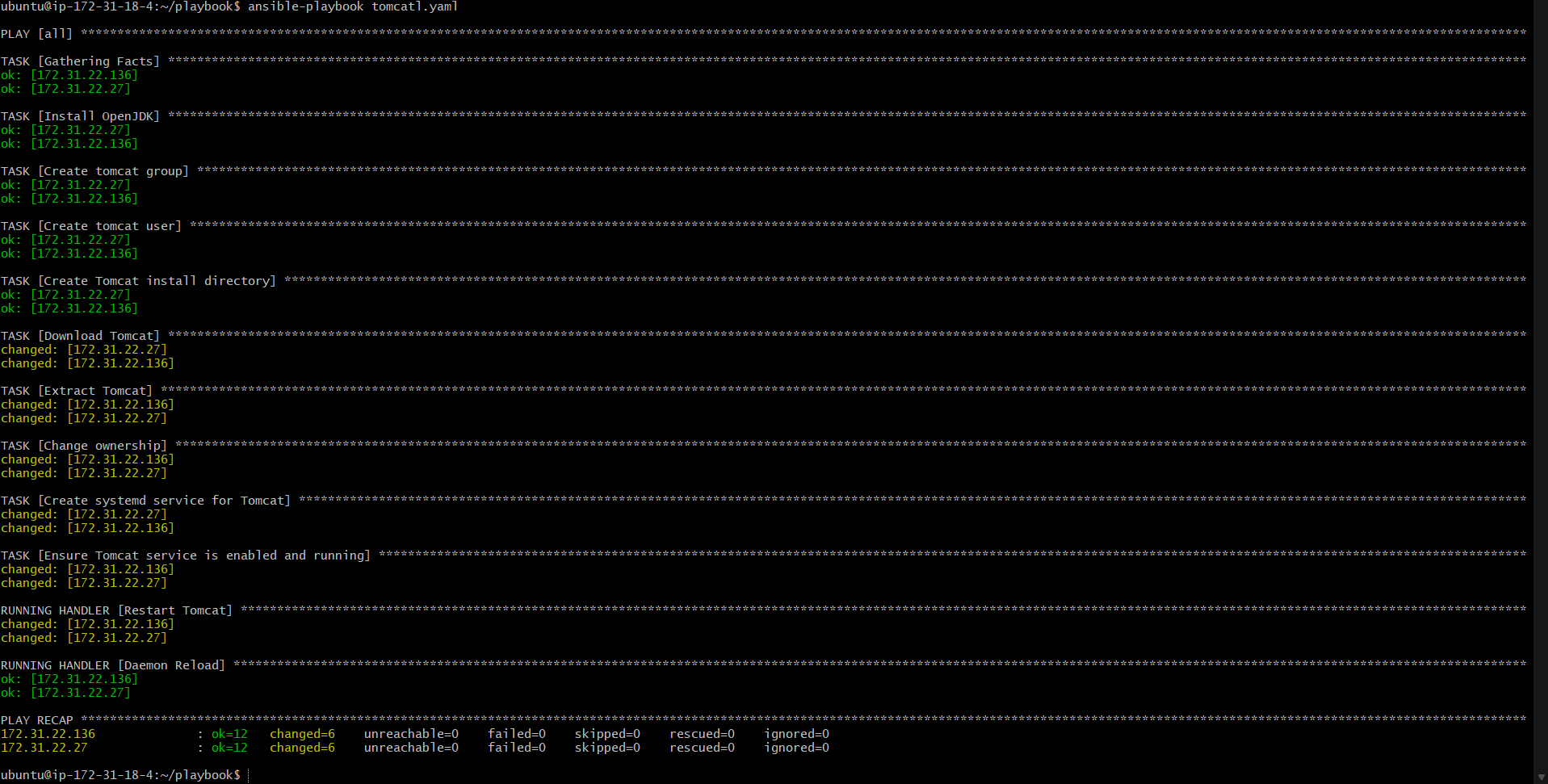
1. Watch ansible-02 video and write down notes.  
   2) Install httpd using ansible playbook, use handlers,notifiers.

      
3) Write a ansible playbook to install apache tomcat.





---  
- name: Install Apache Tomcat  
  hosts: all  
  become: yes  
  vars:  
    tomcat\_version: 9.0.97  
    install\_dir: /opt/tomcat  tasks:  
    - name: Install Java (required for Tomcat)  
      package:  
        name: "{{ 'java-11-openjdk' if ansible\_facts.os\_family == 'RedHat' else 'openjdk-11-jdk' }}"  
        state: present    - name: Create Tomcat directory  
      file:  
        path: "{{ install\_dir }}"  
        state: directory  
        mode: '0755'    - name: Download Tomcat  
      get\_url:  
        url: "[https://archive.apache.org/dist/tomcat/tomcat-9/v{{](https://archive.apache.org/dist/tomcat/tomcat-9/v%7B%7B) tomcat\_version }}/bin/apache-tomcat-{{ tomcat\_version }}.tar.gz"  
        dest: /tmp/tomcat.tar.gz    - name: Extract Tomcat  
      unarchive:  
        src: /tmp/tomcat.tar.gz  
        dest: "{{ install\_dir }}"  
        remote\_src: yes    - name: Start Tomcat  
      command: "{{ install\_dir }}/apache-tomcat-{{ tomcat\_version }}/bin/startup.sh"  
      args:  
        chdir: "{{ install\_dir }}/apache-tomcat-{{ tomcat\_version }}/bin/"

4) Write a ansible playbook to provision one ec2 on aws.

---

- name: Provision an EC2 instance on AWS

hosts: localhost

connection: local

gather\_facts: no

collections:

- amazon.aws

vars:

key\_name: ansible # must exist in AWS

instance\_type: t2.micro

image\_id: ami-0360c520857e3138f # example Ubuntu 20.04 AMI (us-east-1)

region: us-east-1

security\_group: default # must exist in AWS

subnet\_id: subnet-00eb4d5d1f0559ceb # must exist in AWS

tasks:

- name: Launch EC2 instance

ec2\_instance:

key\_name: "{{ key\_name }}"

instance\_type: "{{ instance\_type }}"

image\_id: "{{ image\_id }}"

region: "{{ region }}"

wait: yes

count: 1

network:

subnet\_id: "{{ subnet\_id }}"

assign\_public\_ip: yes

security\_groups:

- "{{ security\_group }}"

tags:

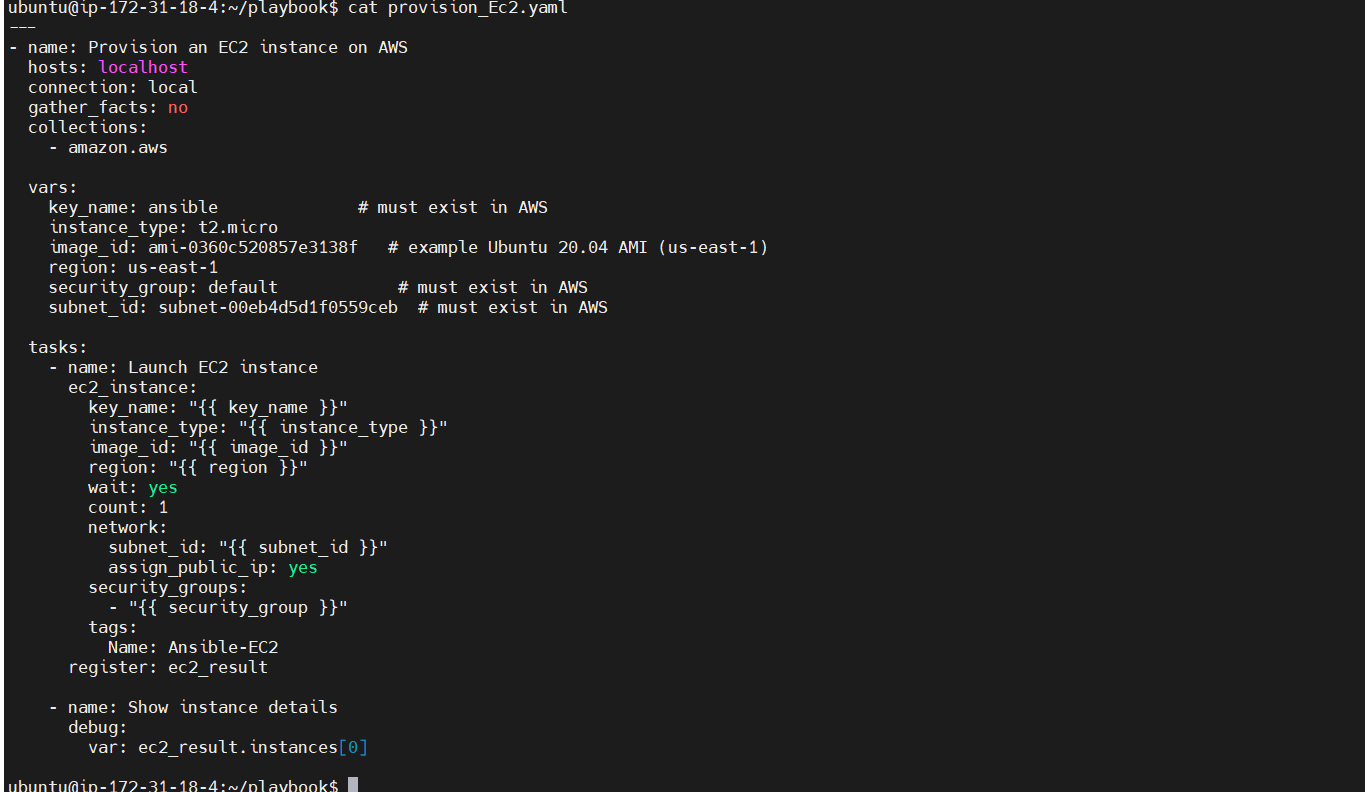
Name: Ansible-EC2

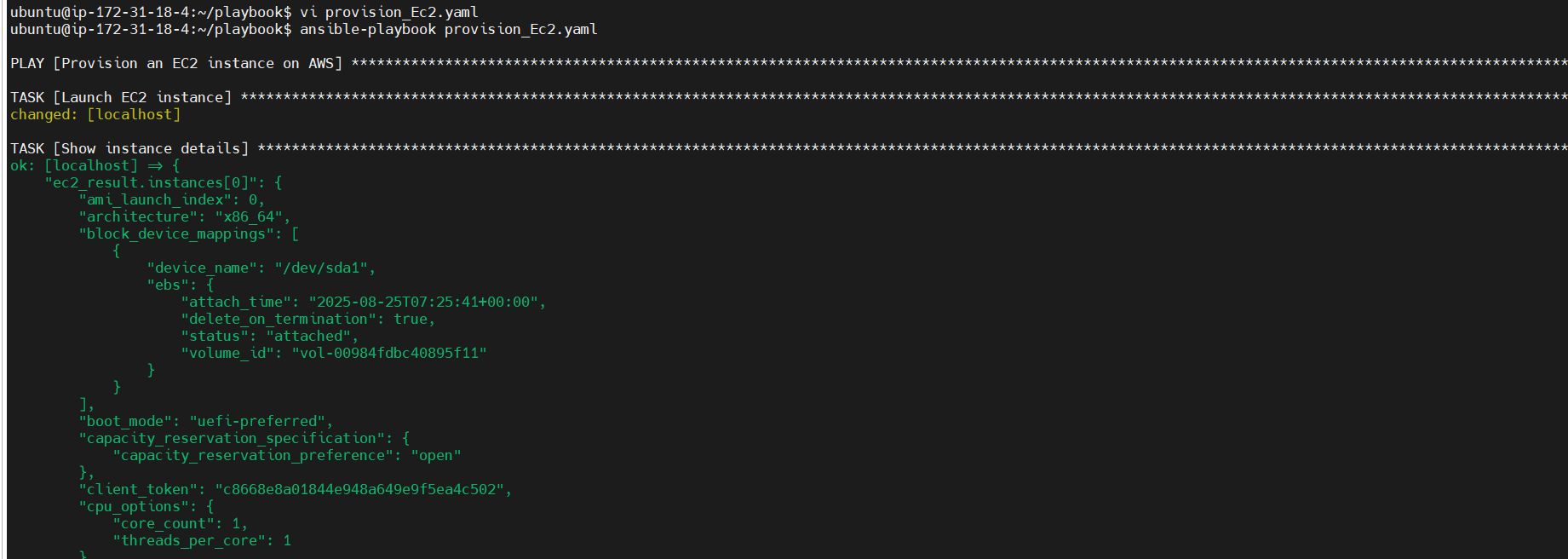
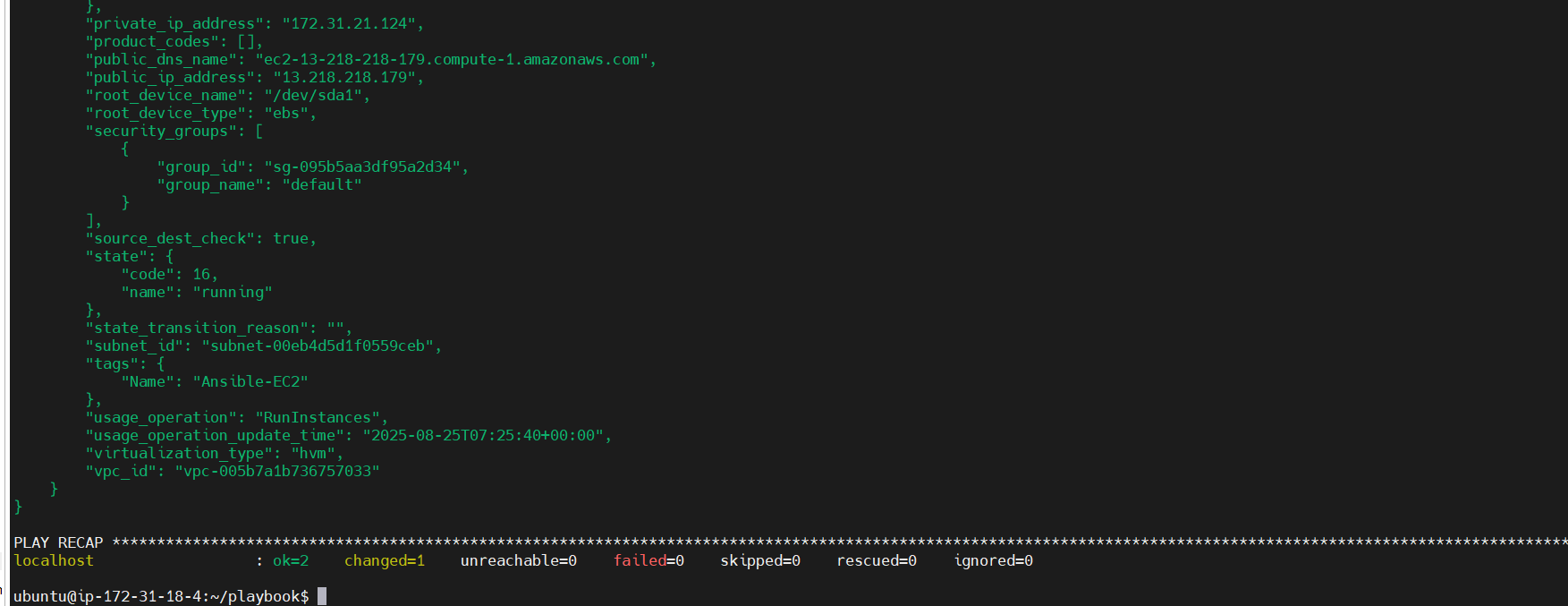
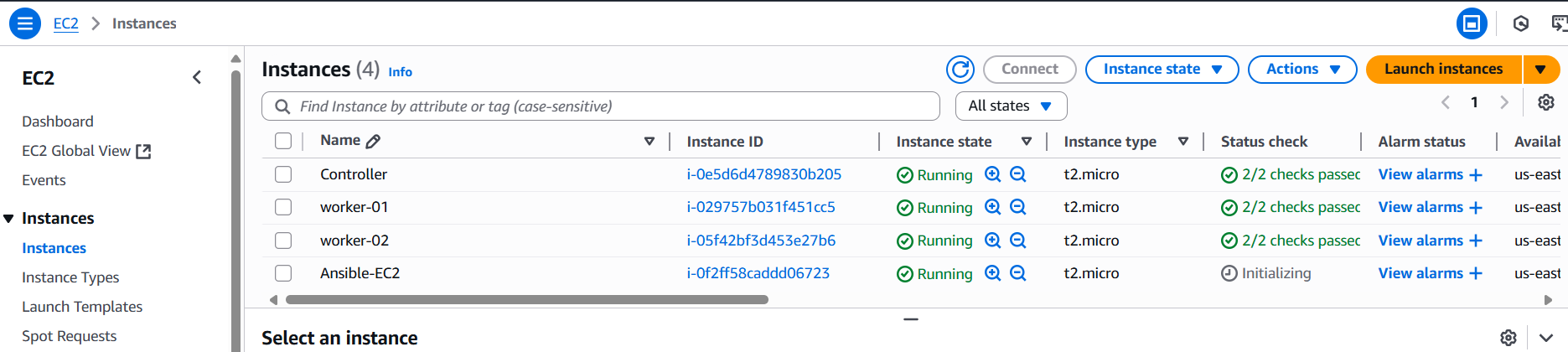
register: ec2\_result

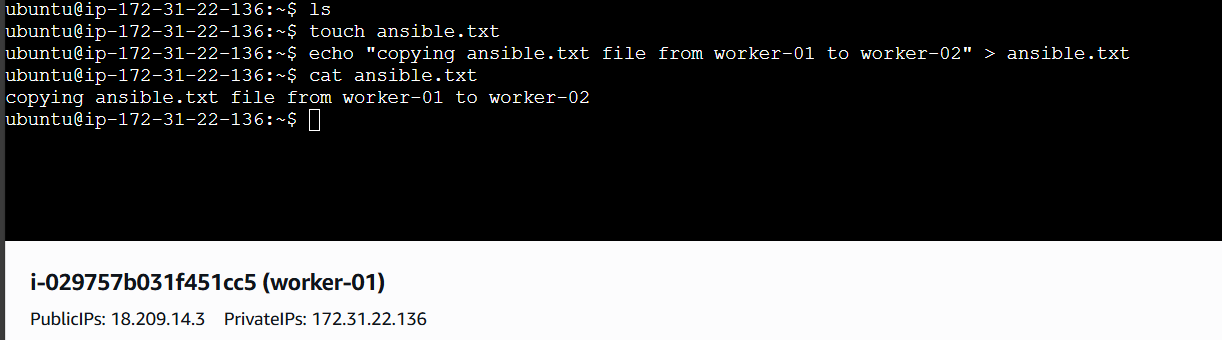
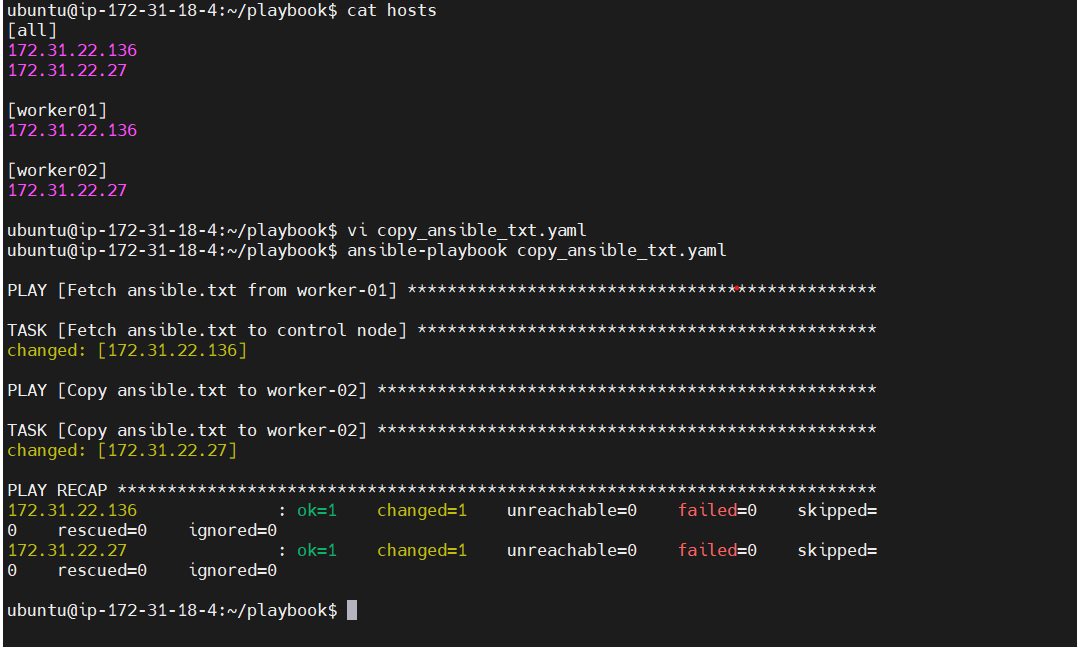
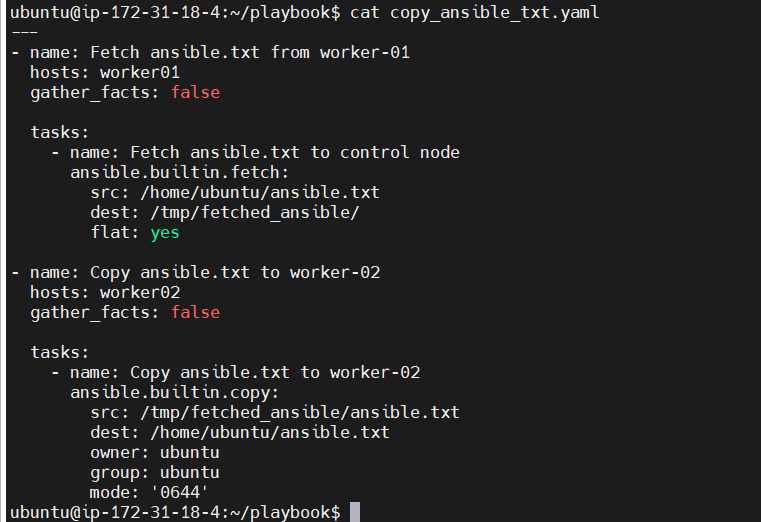
- name: Show instance details

debug:

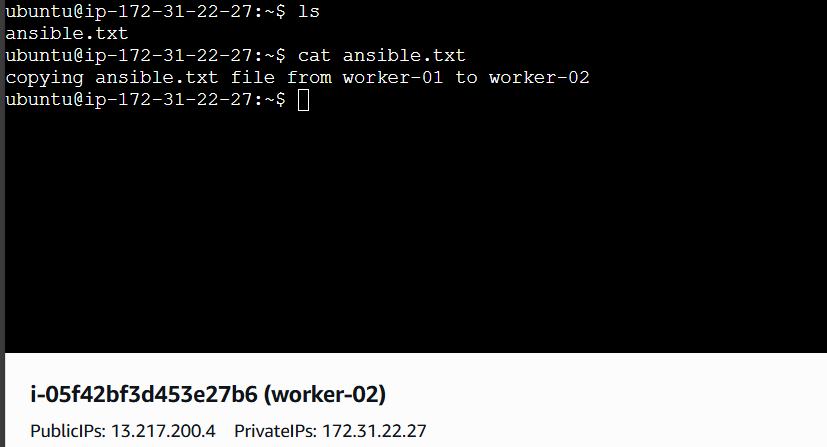
var: ec2\_result.instances[0]



    
5) Write a ansible playbook to copy one file from node-1 to node-2.

Copied to worker-02



6) Write a ansible playbook to create different files with different names using single playbook.

