**ANSIBLE-02**

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

🡪create two servers with linux-2 image which will act as workers

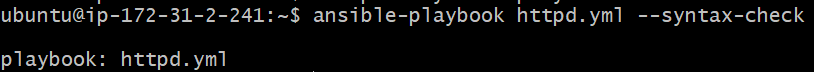
\***on ansible controller add inventory with name linux in hosts file\***

🡪on ansible controller create a file httpd.yml and write the playbook

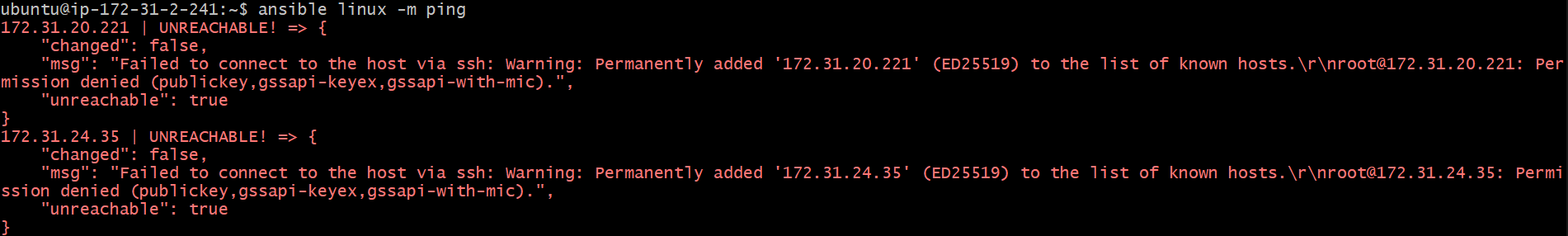
# vi apache.yml

|  |
| --- |
| ---  - hosts: all  become: yes  tasks:  - name: Install httpd  yum:  name: httpd  state: present  notify: Restart httpd  - name: Ensure httpd is started  service:  name: httpd  state: started  handlers:  - name: Restart httpd  service:  name: httpd  state: restarted |

# ansible-playbook httpd.yml –syntax-check **(check the syntax error on cli)**

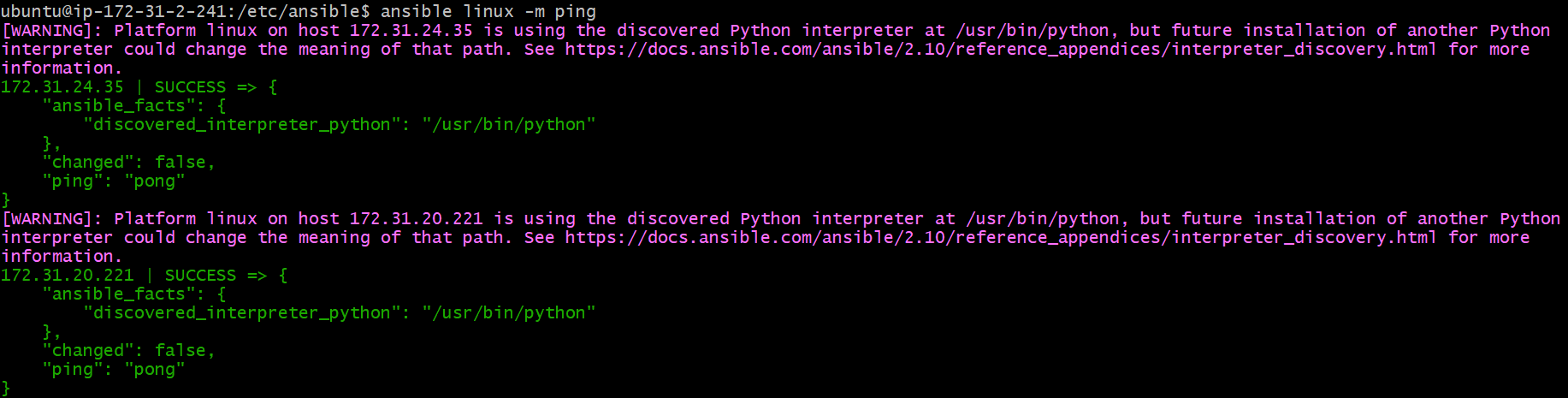


# ansible linux -m ping

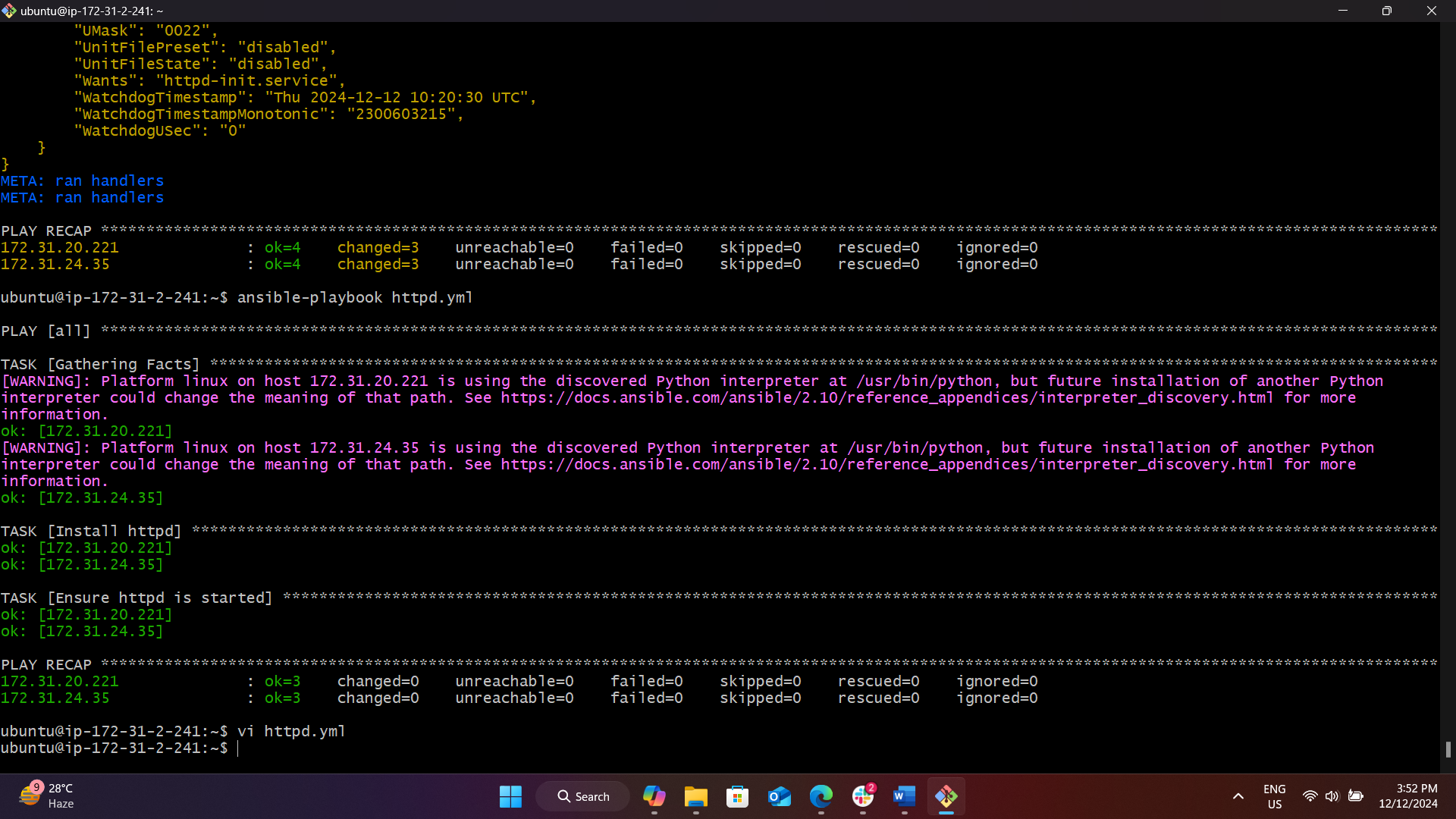


<**user permission issue , generate ssh-keygen and add to workers and restart sshd>**

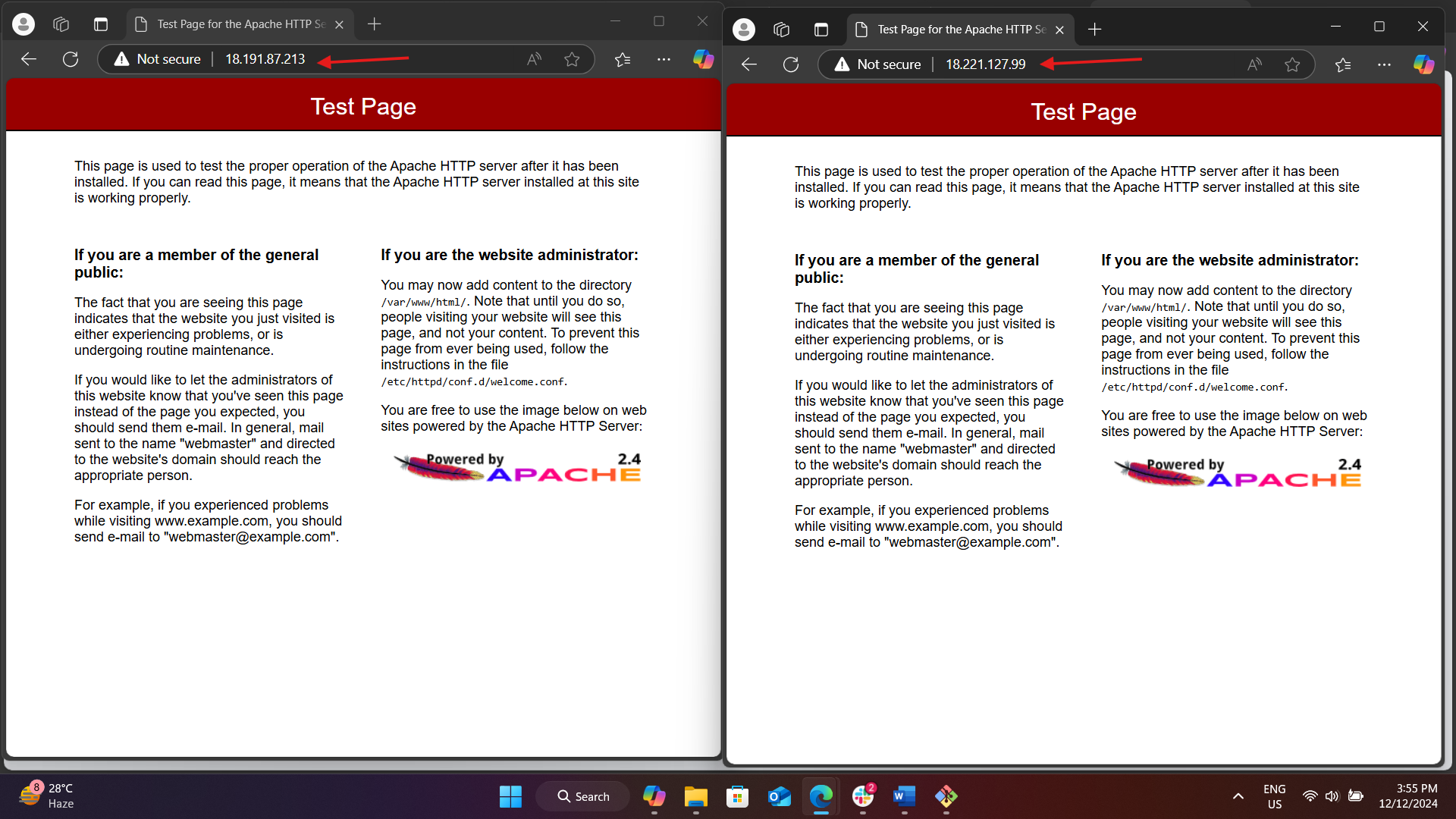
**<edit the ansible.cfg and add remote-user as ec2-user>**

****

# ansible-playbook httpd.yml



🡪checking on browser with public ip’s if it is accessable



**3) Write a ansible playbook to install apache tomcat.**

🡪create a playbook name tomcat.yml

**>get-url =** this helps with downloading packages from web

**>dest =** destination

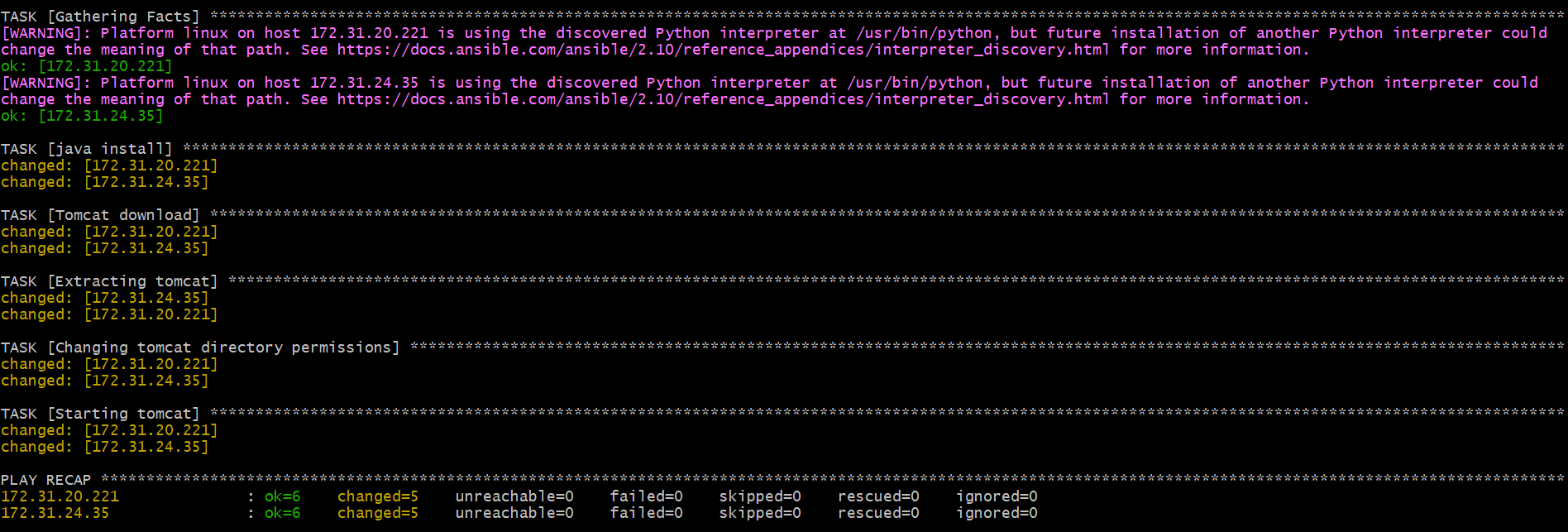
**>unarchive =** like the command like tar xvf

**>src =** source

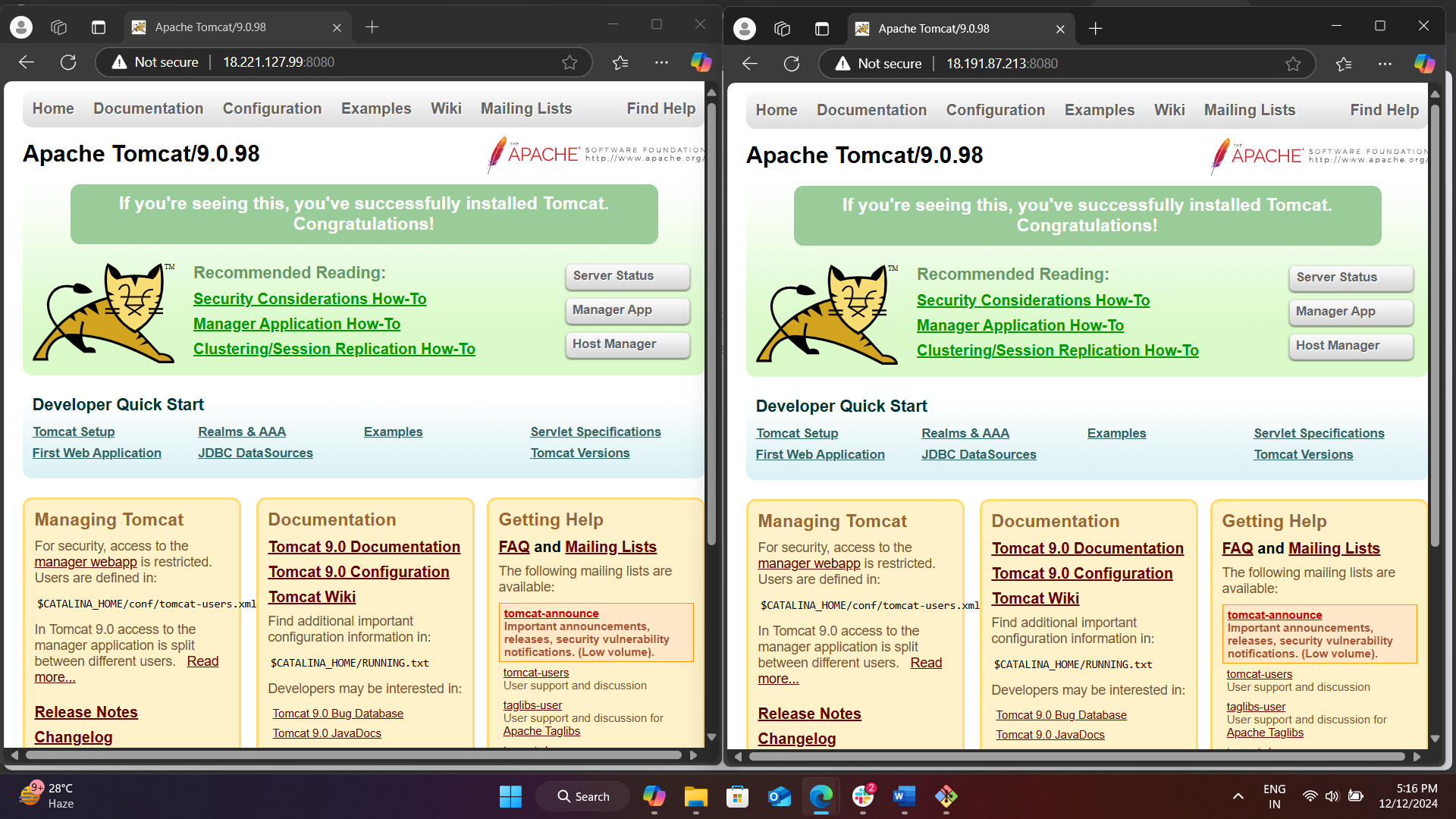
|  |
| --- |
| ---  - hosts: all  become: yes  tasks:  - name: java install  yum:  name: java-17\*  state: present  - name: Tomcat download  get\_url:  url: https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.98/bin/apache-tomcat-9.0.98.tar.gz  dest: /opt/apache-tomcat-9.0.98.tar.gz  - name: Extracting tomcat  unarchive:  src: /opt/apache-tomcat-9.0.98.tar.gz  dest: /opt  remote\_src: yes  - name: Changing tomcat directory permissions  file:  path: /opt/apache-tomcat-9.0.98  owner: ec2-user  group: ec2-user  recurse: yes  - name: Starting tomcat  command: /opt/apache-tomcat-9.0.98/bin/startup.sh |

# ansible-playbook httpd.yml –syntax-check **(check the syntax error on cli)**

# ansible-playbook tomcat.yml



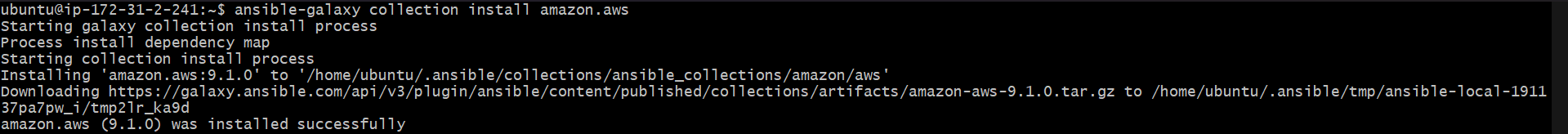
<**checking on browser if tomcat is accessable>**

****

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

**🡪install amazon aws on ansible**

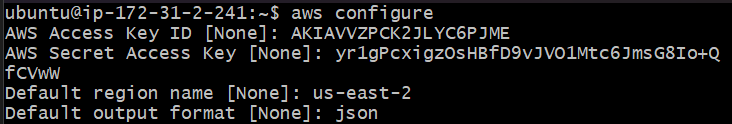
# ansible-galaxy collection install amazon.aws



# sudo apt install -y awscli

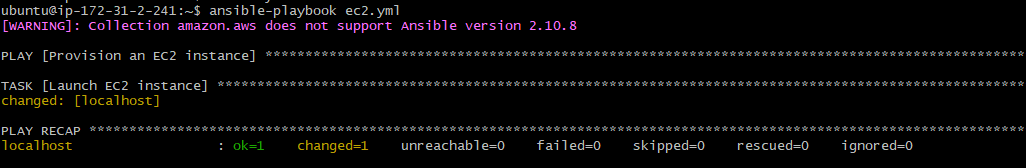


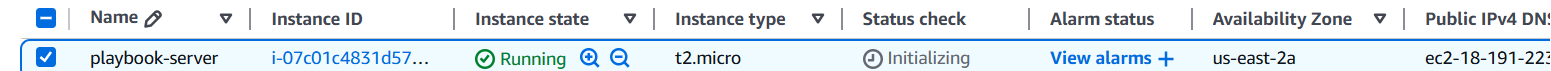
# aws configure



#create a playbook with name ec2.yml and add the following content

|  |
| --- |
| ---  - name: Provision an EC2 instance  hosts: localhost  gather\_facts: false  vars:  ansible\_python\_interpreter: /usr/bin/python3  aws\_access\_key: "your\_access\_key\_id"  aws\_secret\_key: "your\_secret\_access\_key"  tasks:  - name: Launch EC2 instance  amazon.aws.ec2\_instance:  name: playbook-server  key\_name: ohio  instance\_type: t2.micro  image\_id: ami-088d38b423bff245f  region: us-east-2  security\_groups: default  wait: yes  count: 1  aws\_access\_key: "{{ aws\_access\_key }}"  aws\_secret\_key: "{{ aws\_secret\_key }}" |



****

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

**🡪**create a inventory file node\_inventory with following content

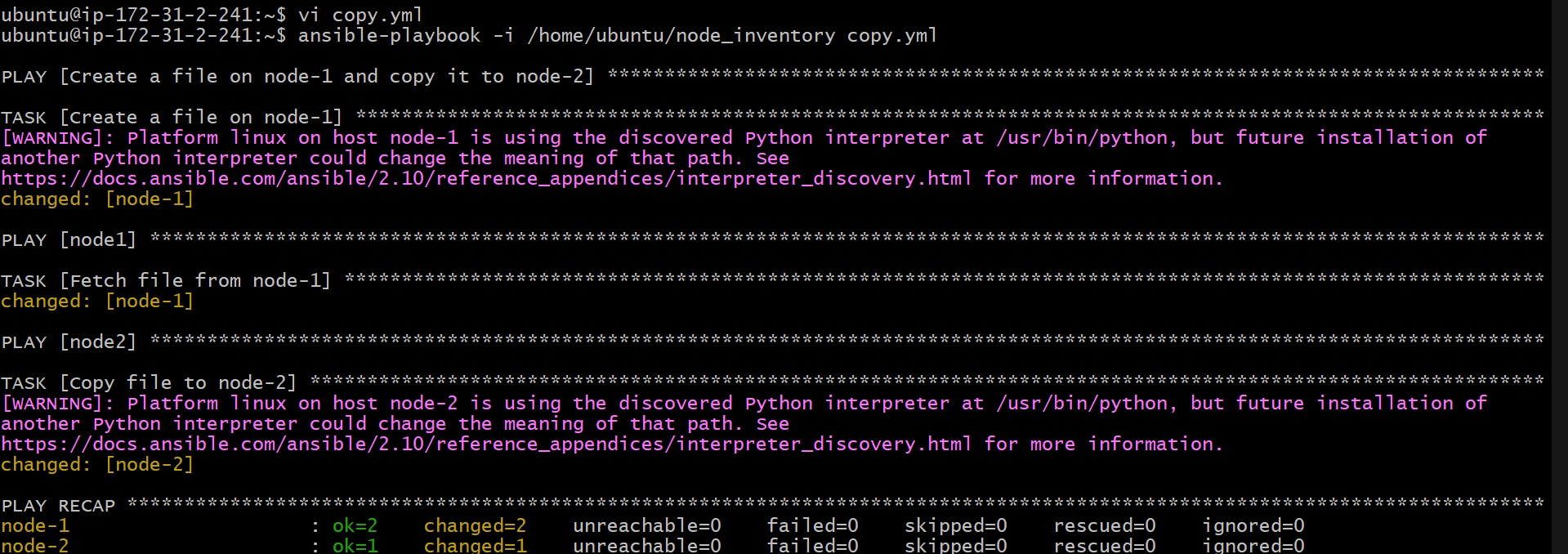
|  |
| --- |
| [node1]  node-1 ansible\_host=NODE\_1\_IP  [node2]  node-2 ansible\_host=NODE\_2\_IP |

**🡪**create a playbook copy.yml with following content

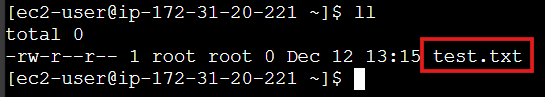
|  |
| --- |
| ---  - name: Create a file on node-1 and copy it to node-2  hosts: node1  gather\_facts: false  tasks:  - name: Create a file on node-1  file:  path: /home/ec2-user/test.txt  state: touch  - hosts: node1  gather\_facts: false  tasks:  - name: Fetch file from node-1  fetch:  src: /home/ec2-user/test.txt  dest: /tmp/test.txt  flat: yes  - hosts: node2  gather\_facts: false  tasks:  - name: Copy file to node-2  copy:  src: /tmp/test.txt  dest: /home/ec2-user/test.txt |

🡪 now run the command for playbook

# ansible-playbook -i /home/ubuntu/node\_inventory copy.yml



🡪check the node2 if file is available



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

**🡪**create a playbook create\_files.yml with following content

|  |
| --- |
| ---  - name: Create multiple files  hosts: localhost  tasks:  - name: Ensure the directory exists  file:  path: /home/ec2-user  state: directory  - name: Create files with different names  file:  path: "/home/ec2-user/{{ item }}"  state: touch  loop:  - file1.txt  - file2.txt  - file3.txt  - file4.txt  - file5.txt |

🡪run the playbook # ansible-playbook create\_files.yml

