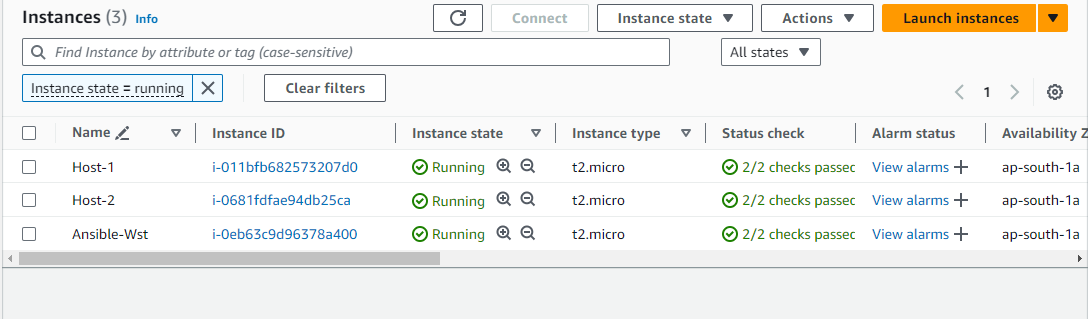
**Ansible Assignment**

Step-1: Launch 3 ec2 machines



Do all the authentication steps

* Creation of user in all three machine which will act as a service account
* Give sudo access to Ansible-wst, Host-1, Host-2
* Update Sshd\_config and restart it again in all three machine
* Generate ssh-keygen from Ansible-Wst and copy it to host

Ssh-copy-id username@private\_ip\_of\_host

* Install ansible on Ansible-WST
* Do changes in ansible.cfg
* Add hostpattern in host

Step-2: Create playbook

Vi test.yaml

---

- hosts: devlinux

user: vaishnavi

become: yes

connection: ssh

gather\_facts: yes

tasks:

- name: Install java

yum:

pkg: java-11-amazon-corretto.x86\_64

state: present

- name: Install git

yum:

pkg: git

state: present

- name: Install maven

yum:

pkg: maven

state: present

- name: dir\_create

command: "{{item}}"

loop:

- rm -rf git

- mkdir git

- name: Git\_clone

ansible.builtin.git:

repo: "https://github.com/Shantanumajan6/project.git"

dest: /home/vaishnavi/git

- name: Maven\_build

shell: mvn clean install

args:

chdir: /home/vaishnavi/git/

- hosts: devlinux

user: vaishnavi

become: yes

connection: ssh

gather\_facts: yes

tasks:

- name: Install docker

yum:

pkg: docker

state: present

- name: Start docker service

service:

name: docker

state: started

- name: Install Docker compose

get\_url:

url: https://github.com/docker/compose/releases/download/1.29.2/docker-compose-{{ ansible\_system }}-{{ ansible\_architecture }}

dest: /usr/local/bin/docker-compose

mode: 0755

- name: Create a symbolic link for docker-compose

ansible.builtin.file:

src: /usr/local/bin/docker-compose

dest: /usr/bin/docker-compose

state: link

- name: Install Python 3

apt:

name: python3

state: present

update\_cache: yes

when: ansible\_os\_family == "Debian"

- name: Install pip3

apt:

name: python3-pip

state: present

update\_cache: yes

when: ansible\_os\_family == "Debian"

- name: Install Docker SDK for Python

pip:

name: docker

state: present

executable: pip3

- name: Pull an image

community.docker.docker\_image:

name: tomcat:latest

source: pull

- name: Delete container

community.docker.docker\_container:

name: cont\_1

state: absent

force\_kill: true

- name: Run Docker container

community.docker.docker\_container:

name: cont\_1

image: tomcat:latest

state: started

ports:

- "8080:8080"

volumes:

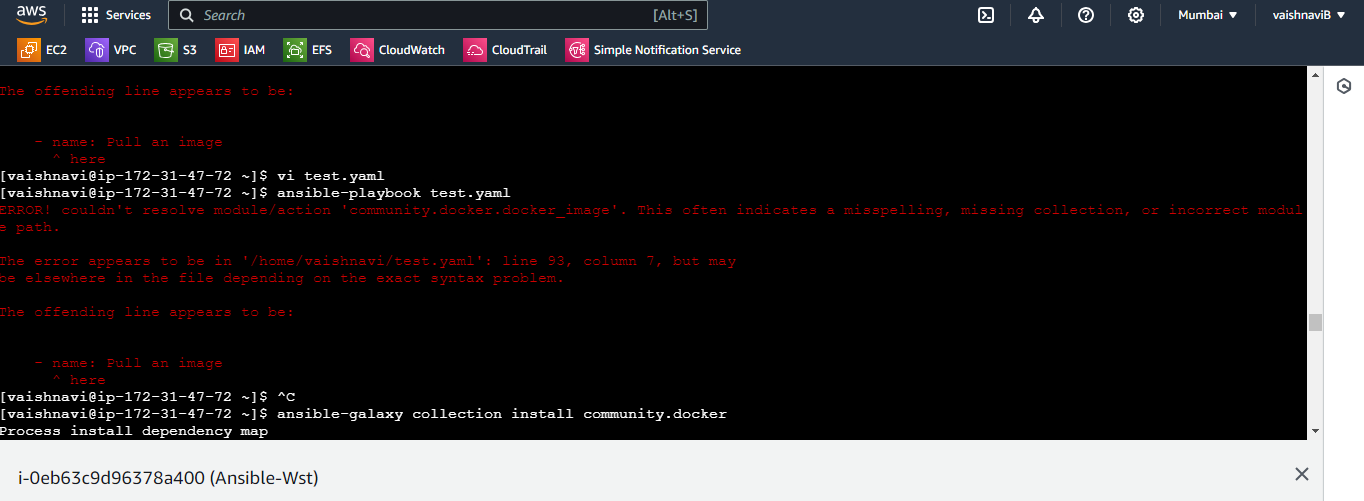
- /home/vaishnavi/git/target:/usr/local/tomcat/webapps

restart\_policy: always

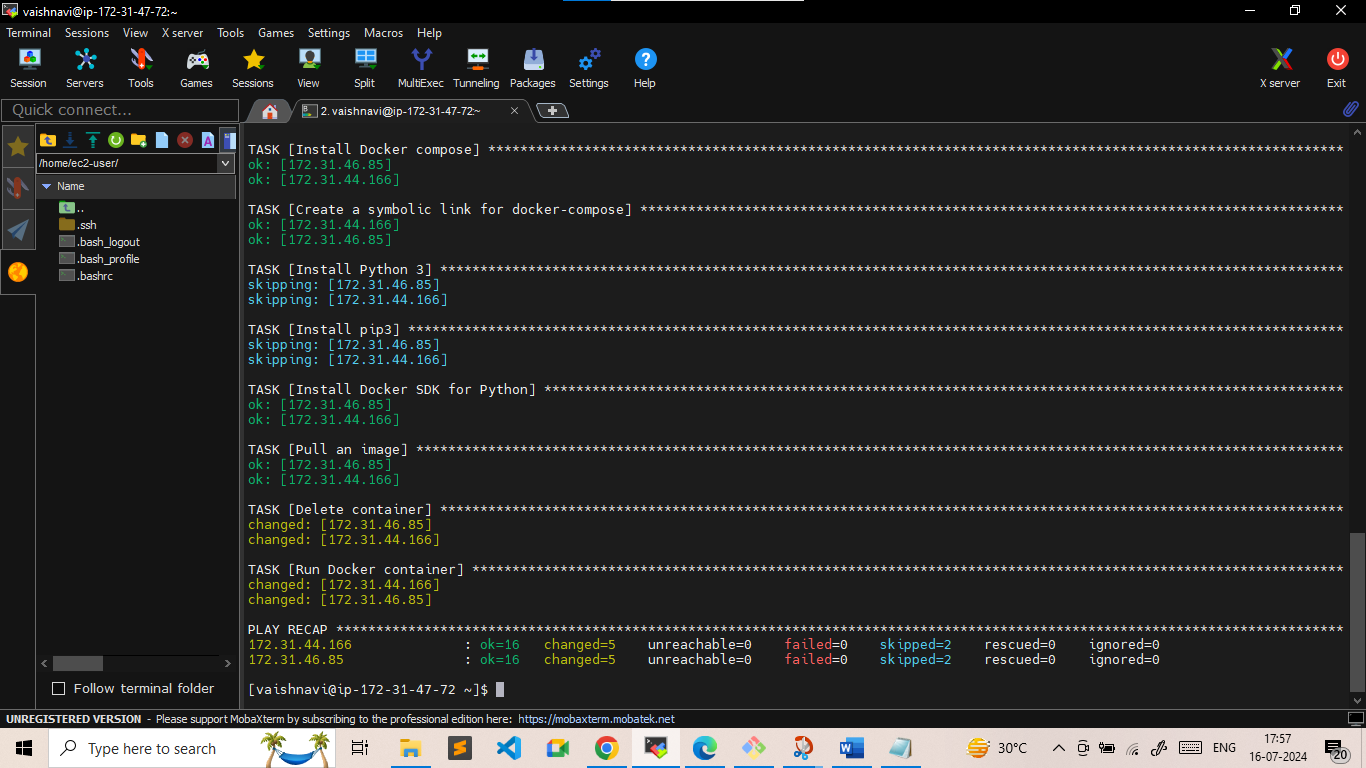
Step-3: Run a yaml script

run this before running a script: **ansible-galaxy collection install community.docker**

an otherwise it will show error



* Ansible-playbook test.yaml



* Access it with publicip of host-1 & host-2

