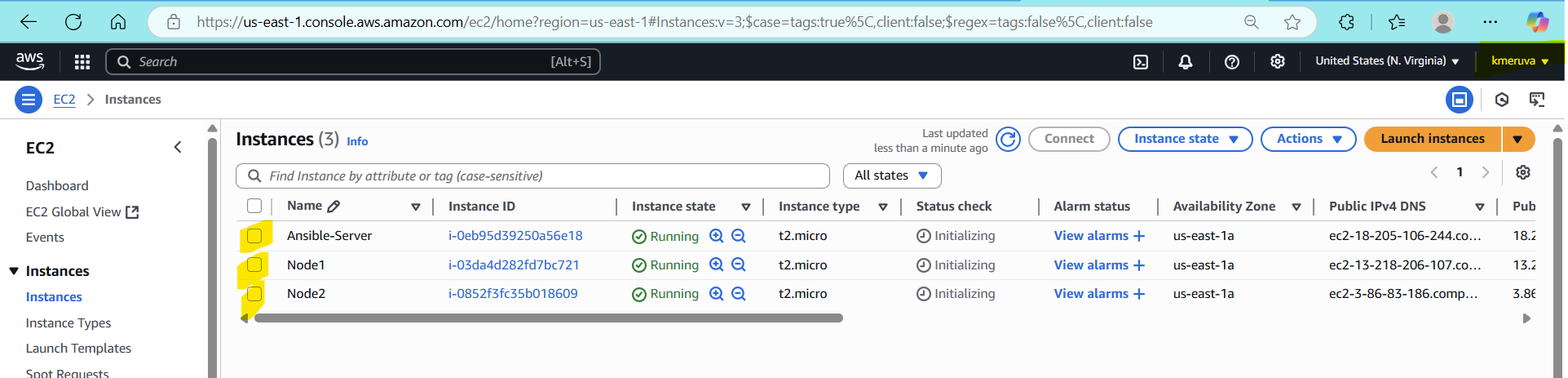
**L1 - Create and Execute Ansible Playbook to Setup Java Maven Application Build Server**

**Creating 3 EC2 instances. 1 Ansible server and node1 & node2**

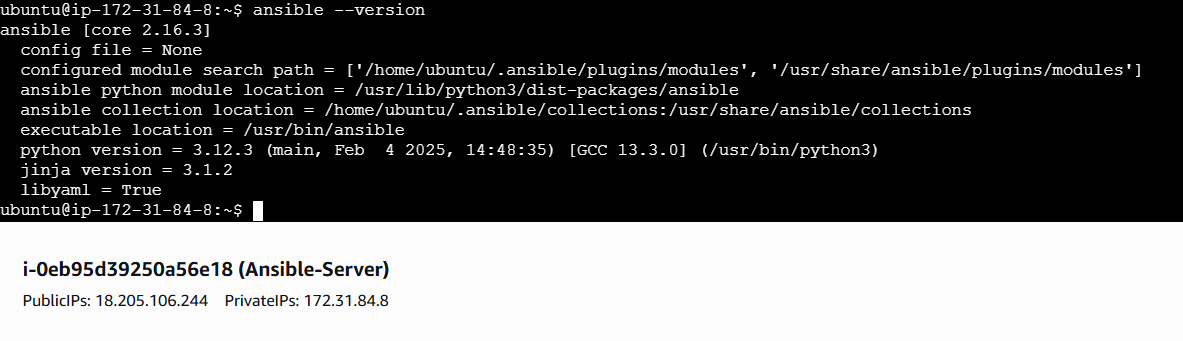


On Master Machine:

sudo apt update

sudo apt install ansible -y

ansible -- version



Create ssh connection using ssh-keygen and copying the pub into node1 & node2.

**Create ansible inventory file:**

mkdir ~/ansible-setup

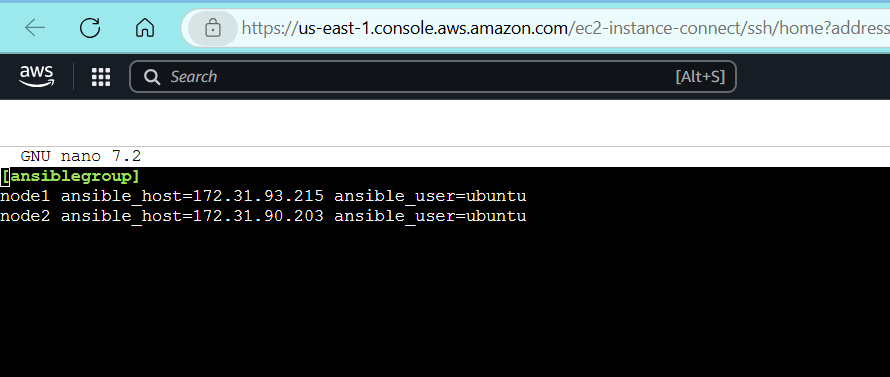
cd ~/ansible-setup

nano inventory

[ansiblegroup]

node1 ansible\_host=172.31.93.215 ansible\_user=ubuntu

node2 ansible\_host=172.31.90.203 ansible\_user=ubuntu

****

**Checking the ansible connection:** ansible -i inventory all -m ping

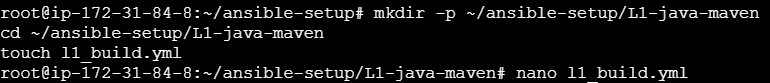
****

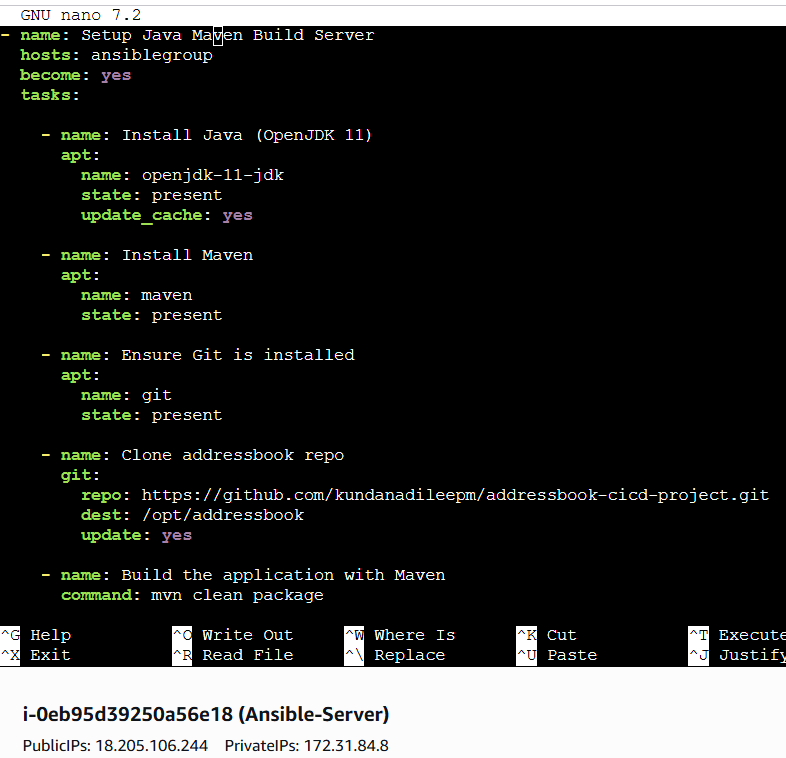
**Create folder and build.yml**

mkdir -p ~/ansible-setup/L1-java-maven

cd ~/ansible-setup/L1-java-maven

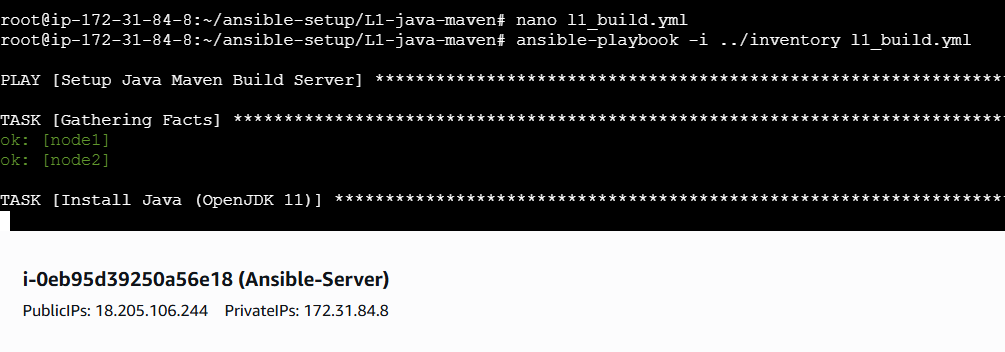
touch l1\_build.yml

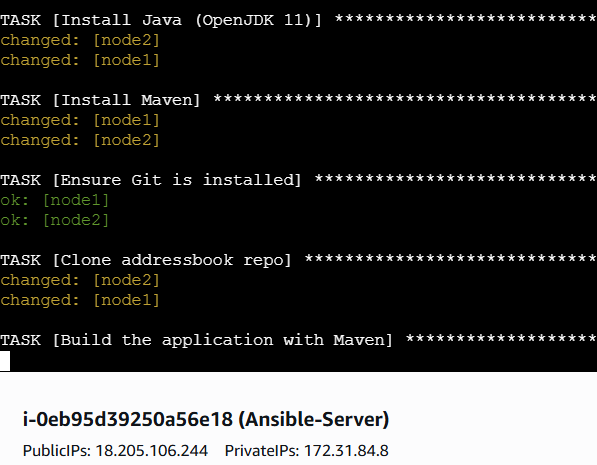
****

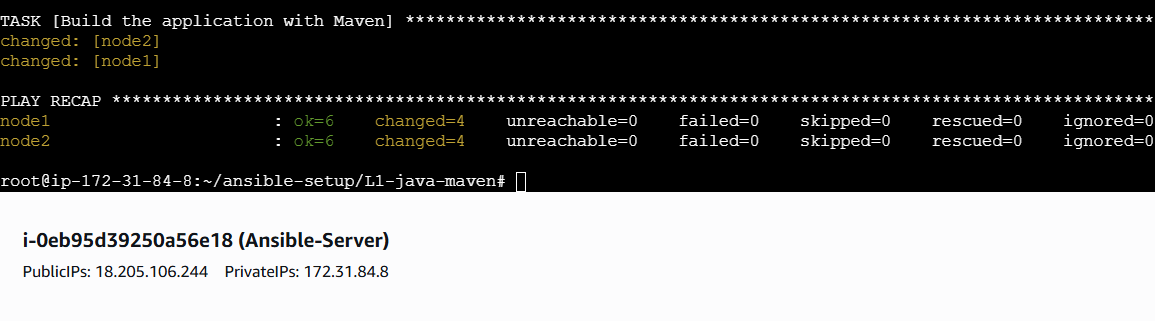
****

**Step 2: Run the Playbook**

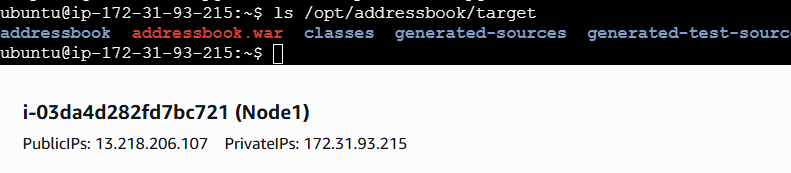
ansible-playbook -i ../inventory l1\_build.yml







.war file created in node1 & 2



**L2 - Create and Execute Ansible Playbook to Install Docker and Run Docker Application Image**

**On Master**

cd ~/ansible-setup

mkdir -p L2-docker-deploy/files

cd L2-docker-deploy

nano files/Dockerfile

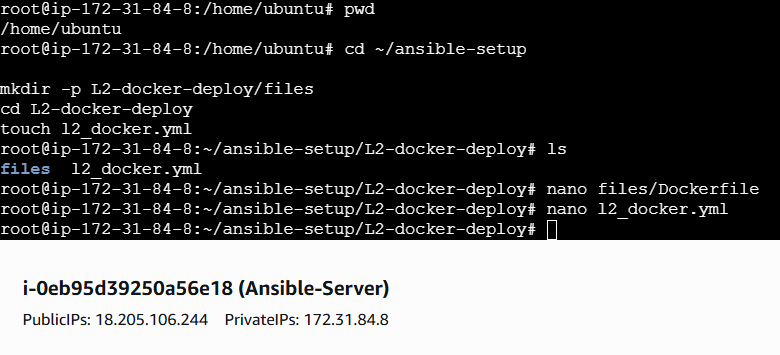
FROM tomcat:9.0

RUN rm -rf /usr/local/tomcat/webapps/\*

COPY addressbook.war /usr/local/tomcat/webapps/addressbook.war

EXPOSE 8080

CMD ["catalina.sh", "run"]



**touch l2\_docker.yml**

- name: L2 - Install Docker and Run Addressbook App

hosts: dockerserver

become: true

tasks:

- name: Add Docker GPG key

apt\_key:

url: https://download.docker.com/linux/ubuntu/gpg

state: present

- name: Add Docker repository

apt\_repository:

repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy stable

state: present

- name: Install Docker

apt:

name: docker-ce

state: present

update\_cache: yes

- name: Copy WAR file to node2

copy:

src: files/addressbook.war

dest: /home/ubuntu/addressbook.war

- name: Copy Dockerfile to node2

copy:

src: files/Dockerfile

dest: /home/ubuntu/Dockerfile

- name: Build Docker image

command: docker build -t addressbook-app .

args:

chdir: /home/ubuntu

- name: Run Docker container

docker\_container:

name: addressbook

image: addressbook-app

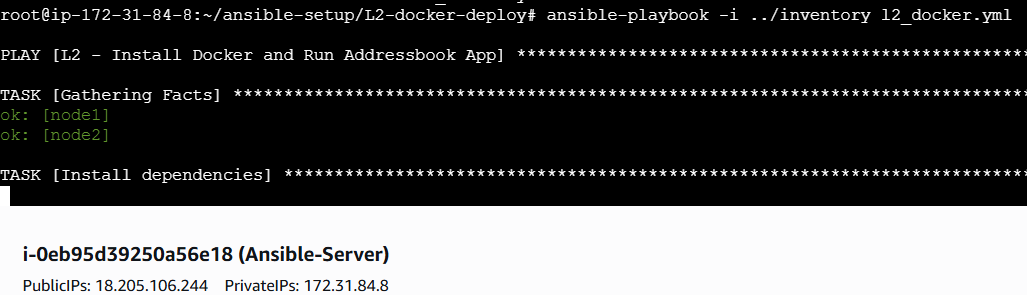
state: started

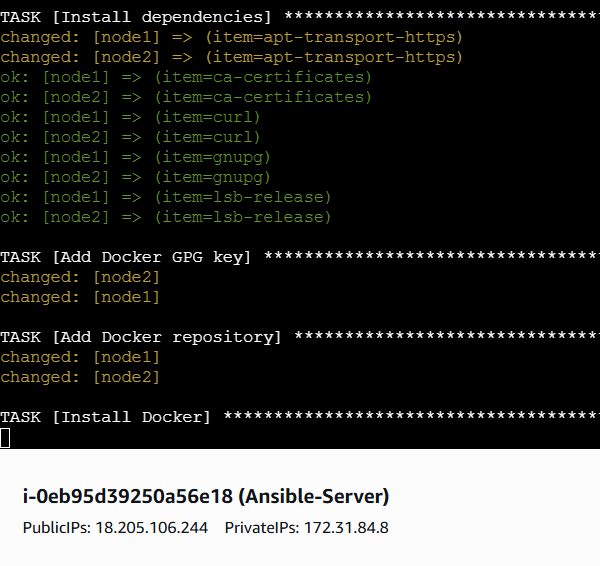
ports:

- "8080:8080"

**Step 2: Run the Playbook**

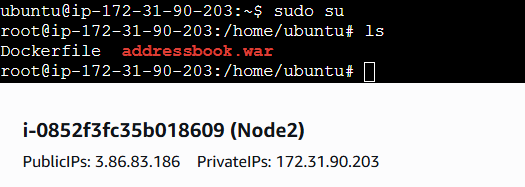
ansible-playbook -i ../inventory l2\_docker.yml



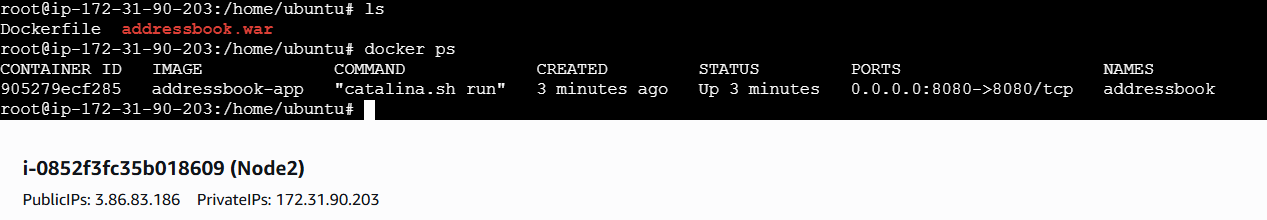




**On Node2 .war and dockerfile have been copied**



docker ps



<http://3.86.83.186:8080/addressbook/>

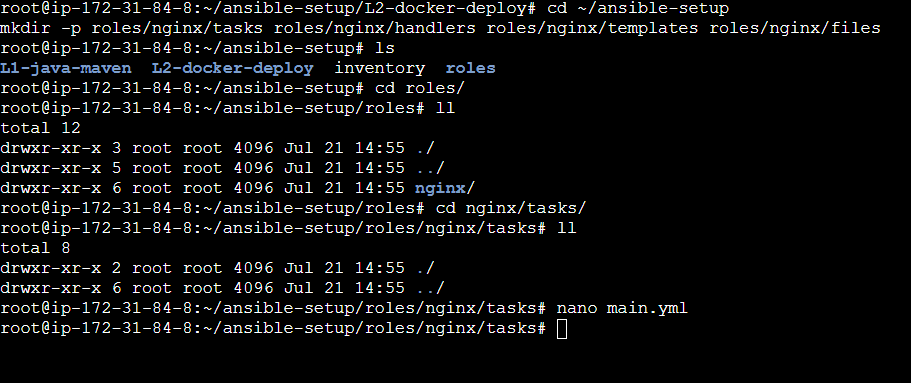


**L3 - Create Ansible Role to define the task, handler for Nginx Service Installation and invoke the role in Ansible playbook**

**On Master**

cd ~/ansible-setup

mkdir -p roles/nginx/tasks roles/nginx/handlers roles/nginx/templates roles/nginx/files



**Main.yml**

---

- name: Install Nginx

apt:

name: nginx

state: present

update\_cache: yes

- name: Deploy custom Nginx config

copy:

src: nginx.conf

dest: /etc/nginx/nginx.conf

backup: yes

notify: Restart Nginx

**Create Role Handler: roles/nginx/handlers/main.yml**

yaml

CopyEdit

---

- name: Restart Nginx

service:

name: nginx

state: restarted

**Create a Simple Nginx Config: roles/nginx/files/nginx.conf**

You can start with a default one or use this:

nginx

CopyEdit

events {}

http {

server {

listen 80;

location / {

return 200 'Hello from Nginx installed via Ansible Role!';

}

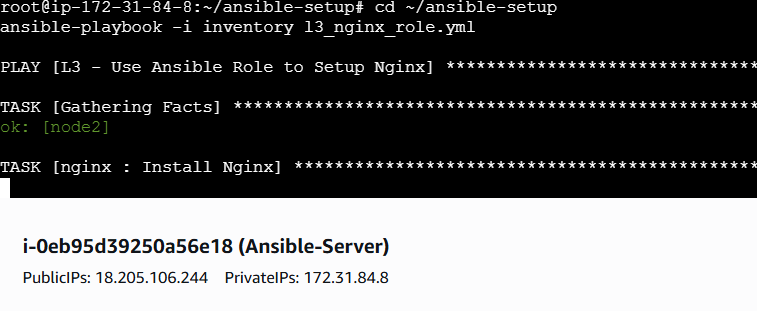
}

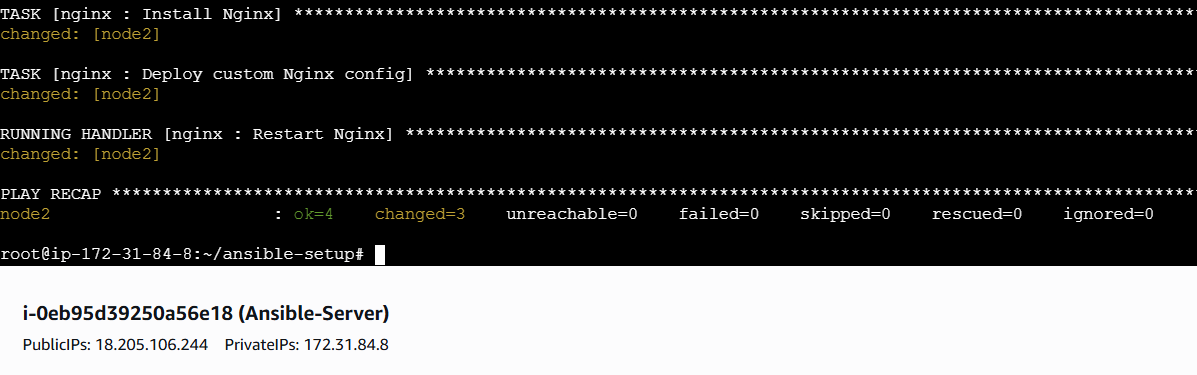
}

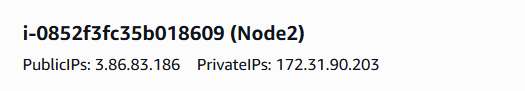
Now run below:

cd ~/ansible-setup

ansible-playbook -i inventory l3\_nginx\_role.yml





**Nginx is running on node2** 

http://3.86.83.186/

