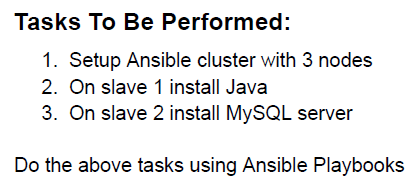
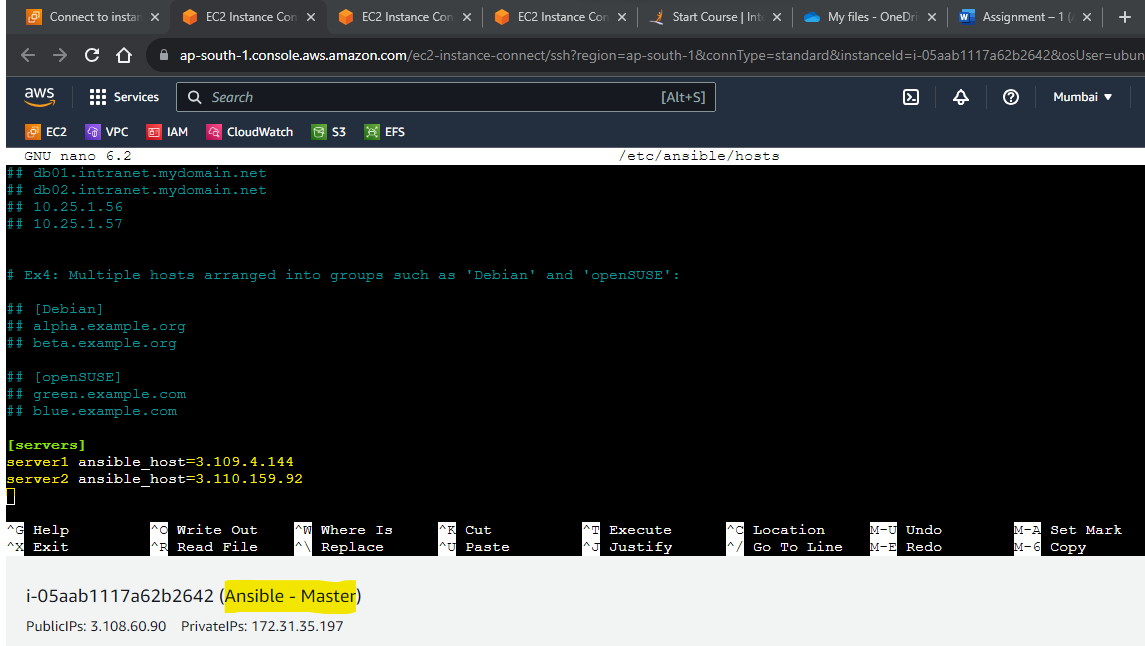
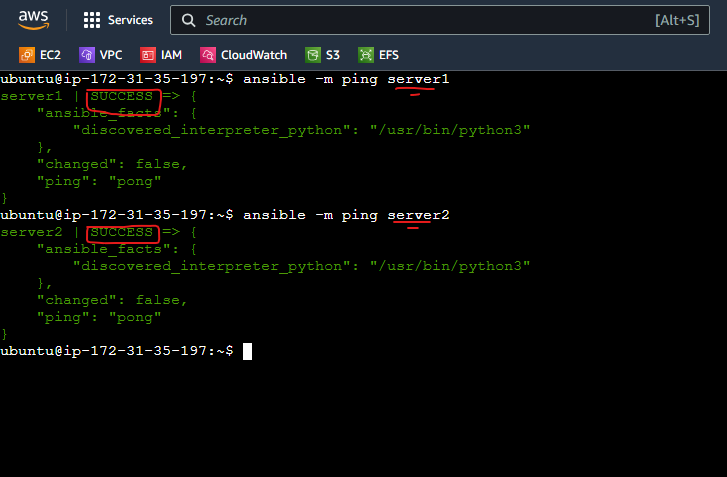
**Assignment – 1 (Ansible)**



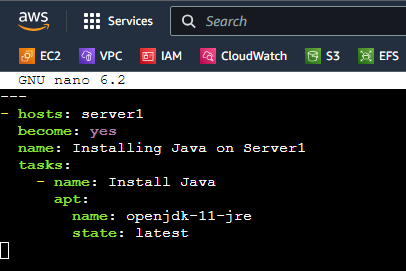
1. Setup Ansible Master and Slave by installing
   1. python on both the machines (Master and Slave)
   2. Generate a ssh-keygen in Ansible Master, copy the contents of id\_rsa.pub file in ./.ssh Directory.
   3. Paste the contents into both the Slave Machine under ./.ssh/authorized\_keys file.
   4. Configure and Install Ansible on Master Machine. Add both the Slave Machine Public IP under /etc/ansible/hosts file.



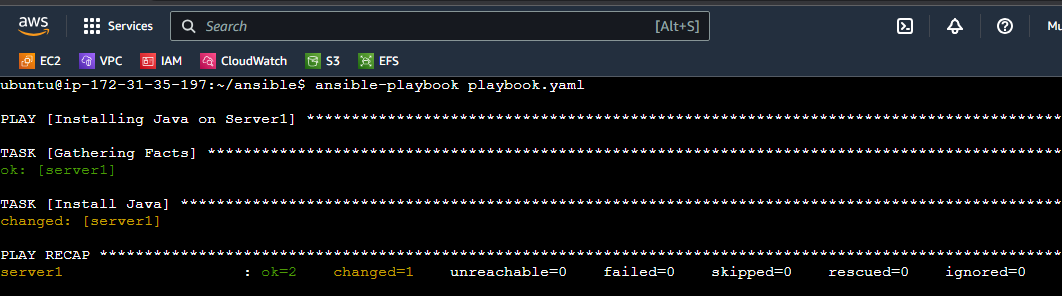
* 1. Try pinging both the slave machine via Master.



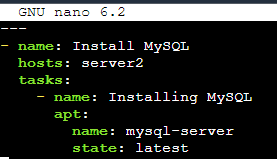
2. Create a playbook to install java on Slave 1. The below script will install java on Slave 1.



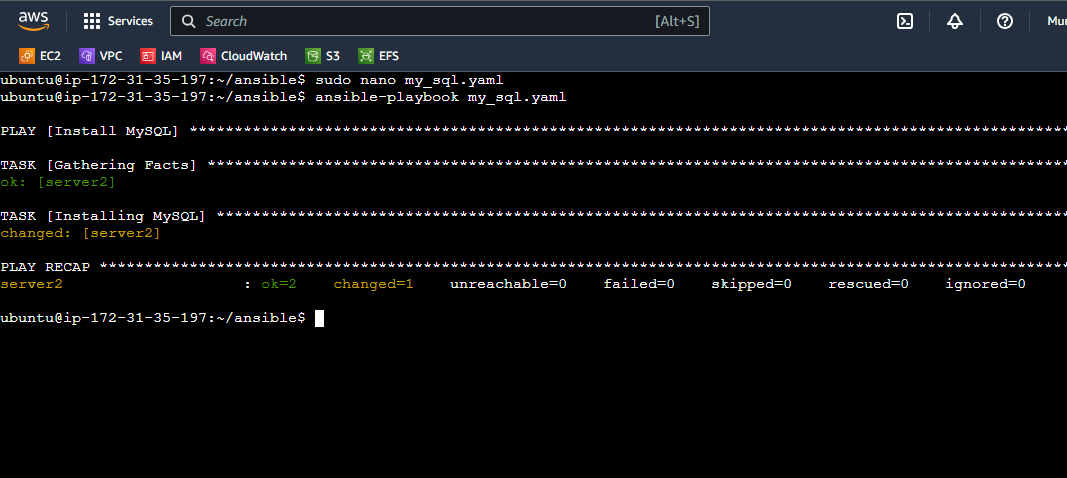
3. Run the playbook, you will receive the below output if succeded.



5. Create another playbook to install MySQL on Slave 2.

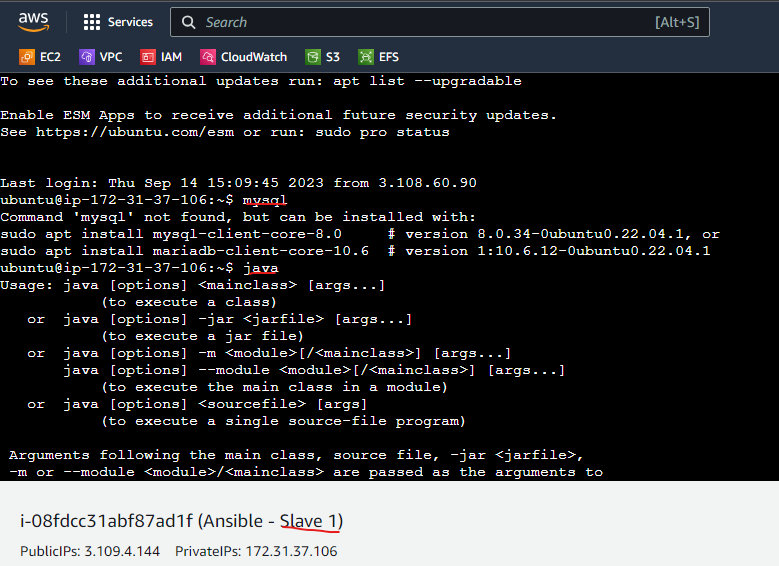


6. Execute the 2nd Playbook to Install MySQL on Slave 2.

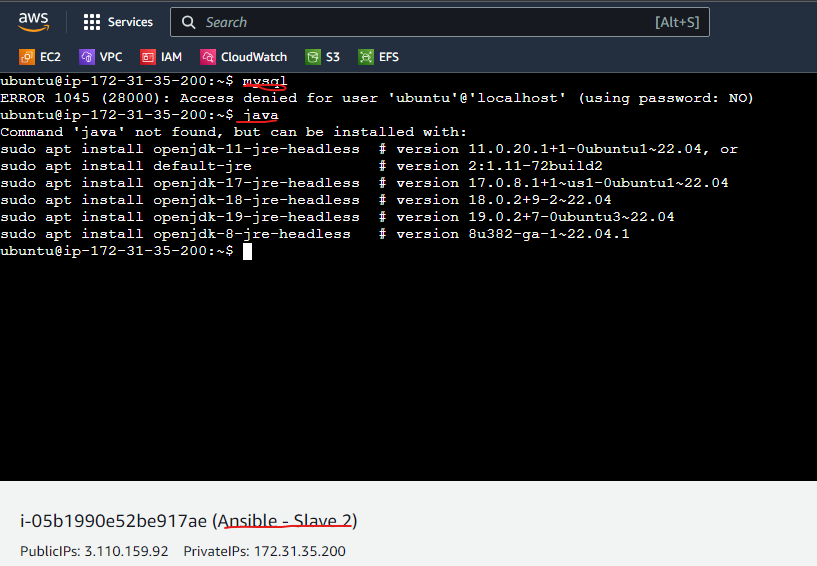


7. Cross check if both the packages are installed on Slave 1 and Slave 2.

A. Check java on Slave 1

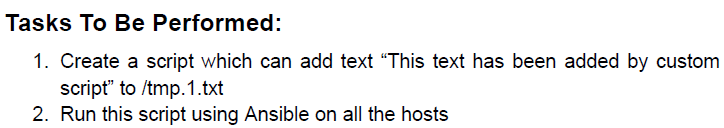


B. Check mysql on Slave 2

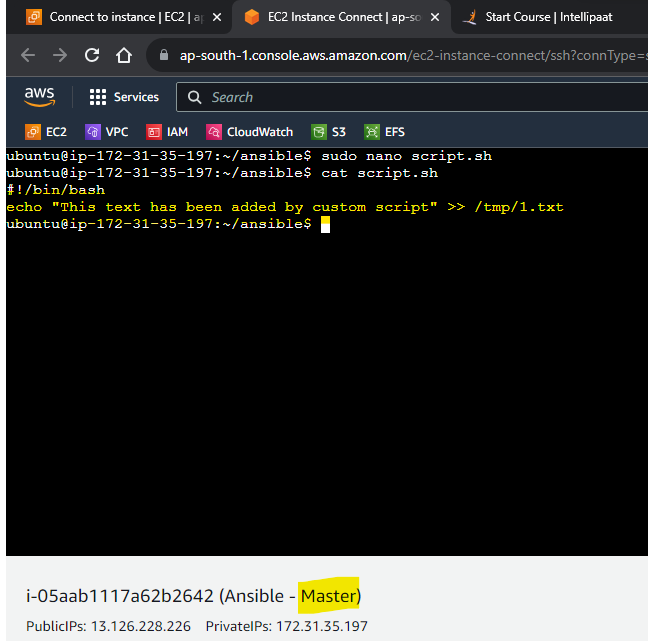


As you can observe, both the Slave Machines are being Managed by Master. Both the packages are installed on respective Slaves.

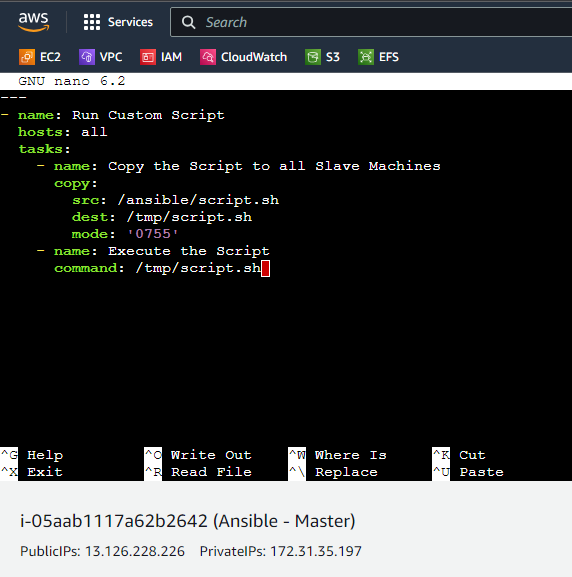
**Assignment 2 – Ansible**

****

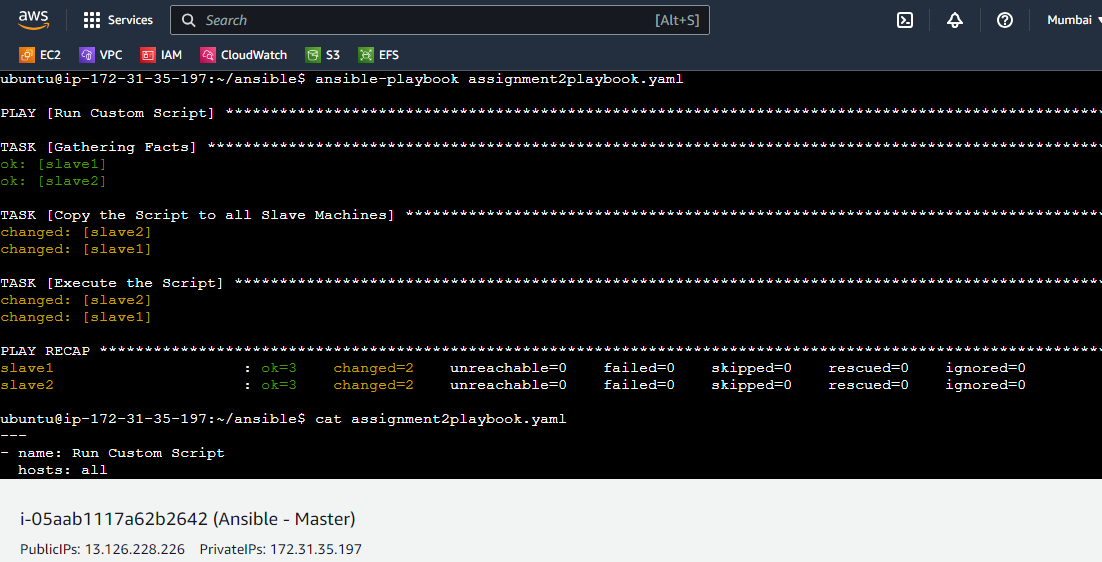
1. Create a Script which would add the content into the desired path.



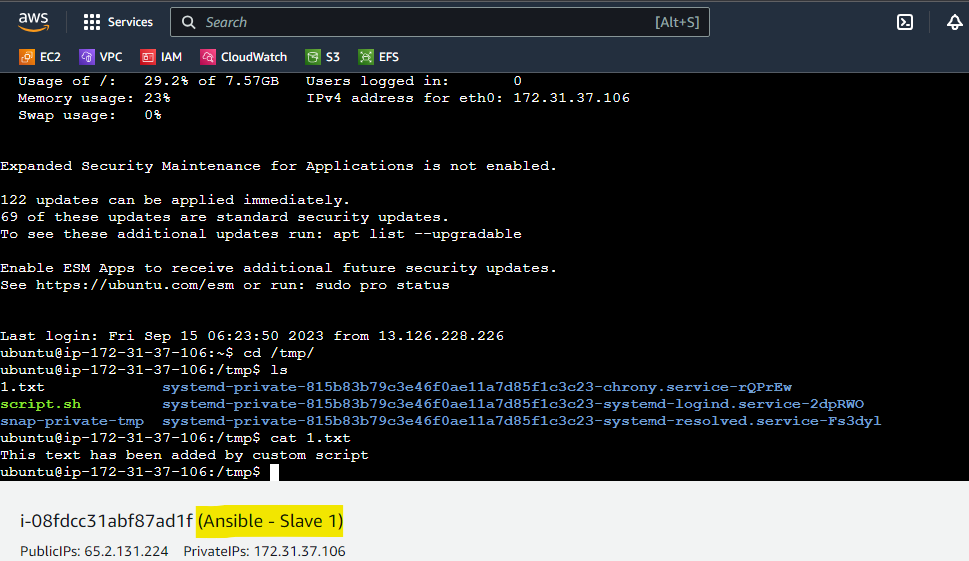
1. Create a Playbook which would copy the file from master to all the slave and then execute the script on all the slaves.



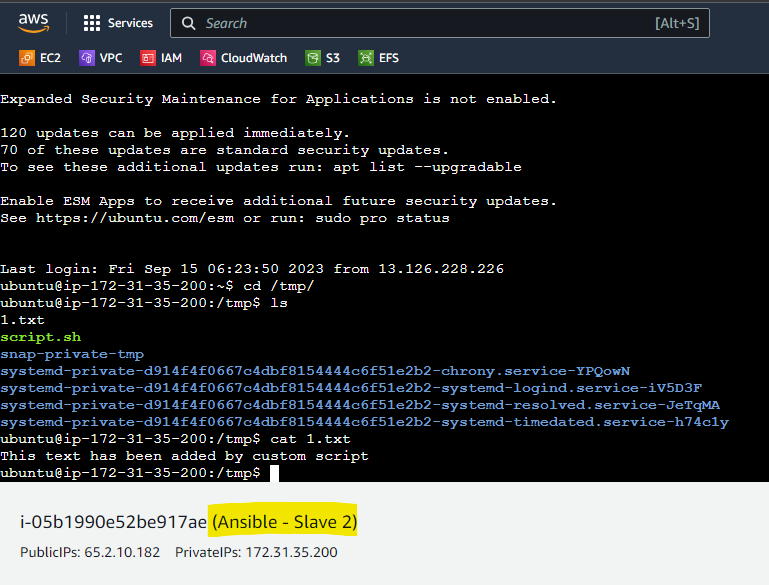
1. Use chmod +x script.sh command to make the Script Executable, and run the playbook using ansible-playbook command.



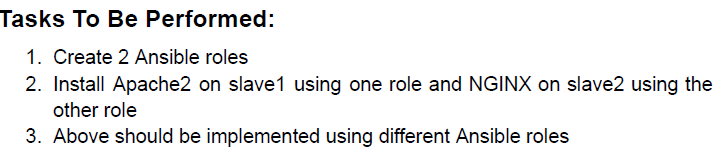
1. Check the /tmp path on Slave Machines if the Ansible Playbook ran successfully or not.
   1. Slave 1



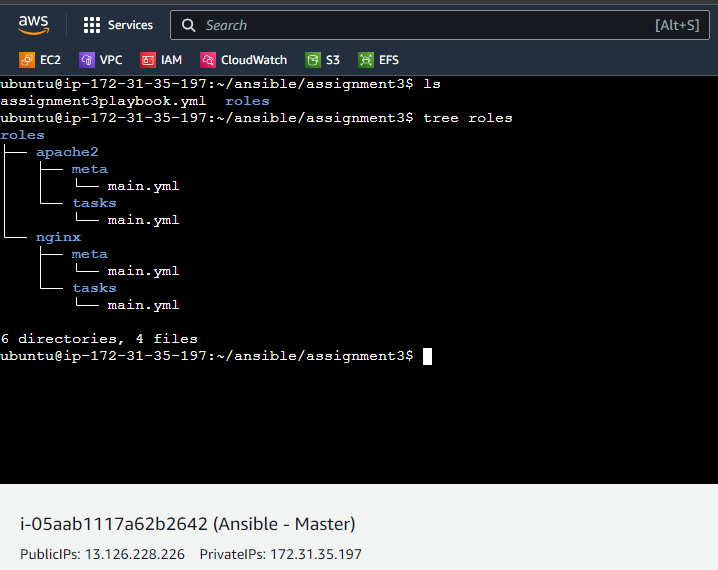
* 1. Slave 2



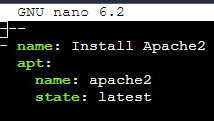
**Assignment - 3 (Ansible)**

****

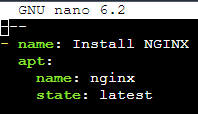
1. Create a directory “roles” and create 2 more directories name it according to the role names, in our case “apache2” and “nginx”. Create another directory named tasks on both the role.



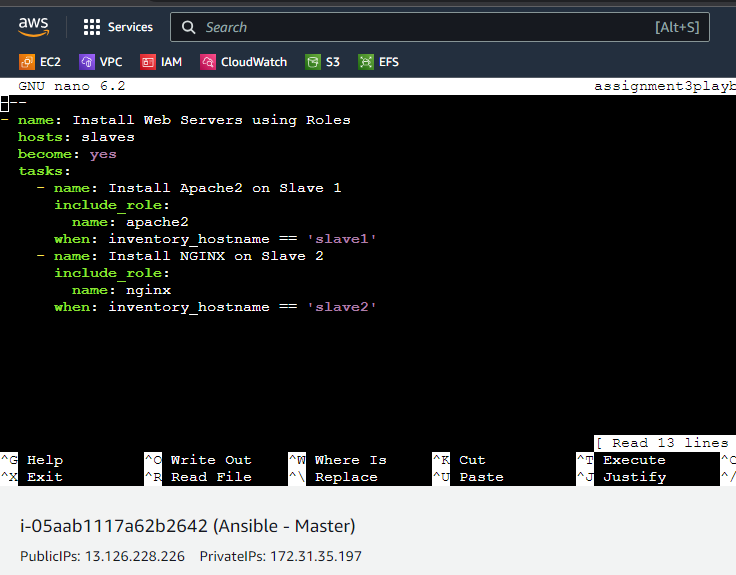
1. Create main.yml file inside /roles/apache2/tasks and /roles/nginx/tasks.
   1. apache2 main.yml content:



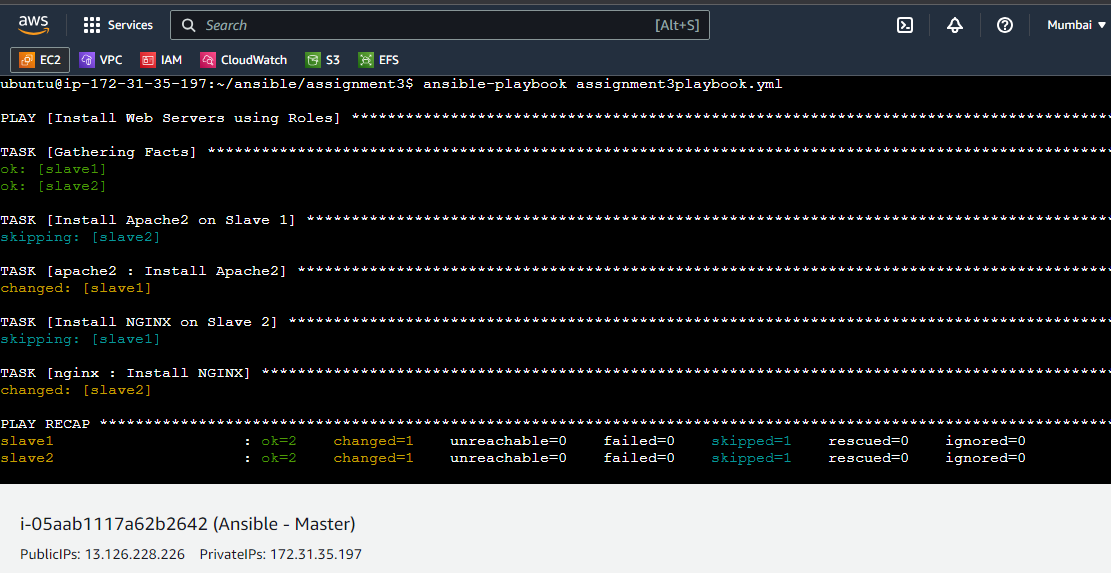
* 1. nginx main.yml content:



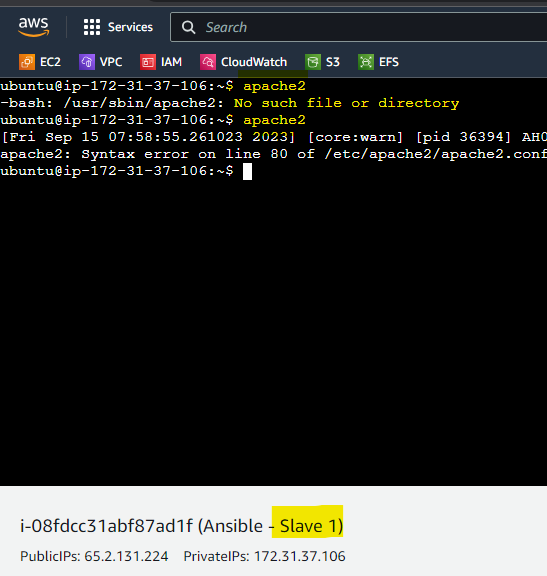
1. Create a Ansible Playbook and include both the roles in the playbook.



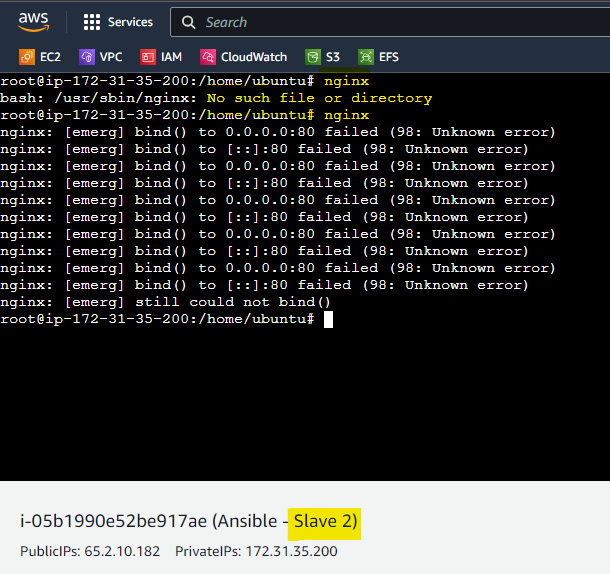
1. Execute the Playbook.



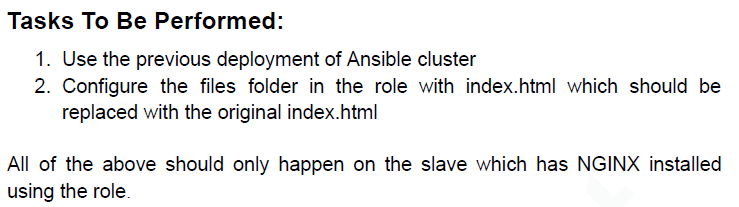
1. The playbook executed and performed the task successfully.
   1. Observe the change in Slave 1 before and after installing apache2 (Verification).



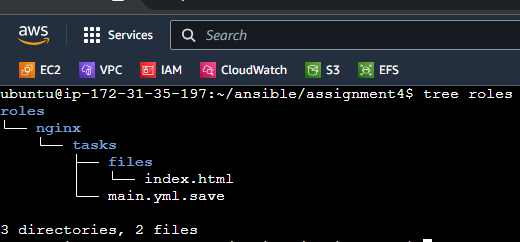
* 1. Observe the change in Slave 2 before and after installing nginx (Verification).



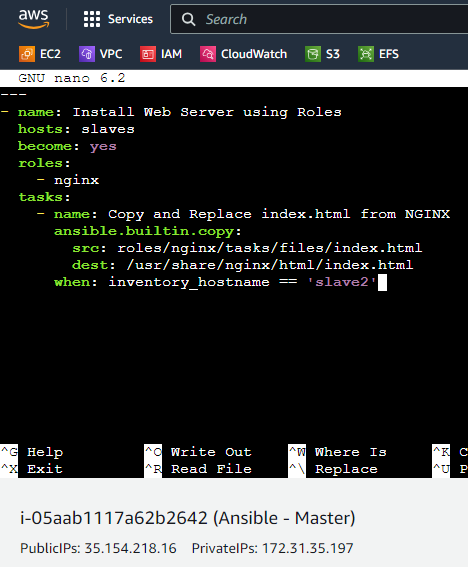
**Assignment – 4 (Ansible)**



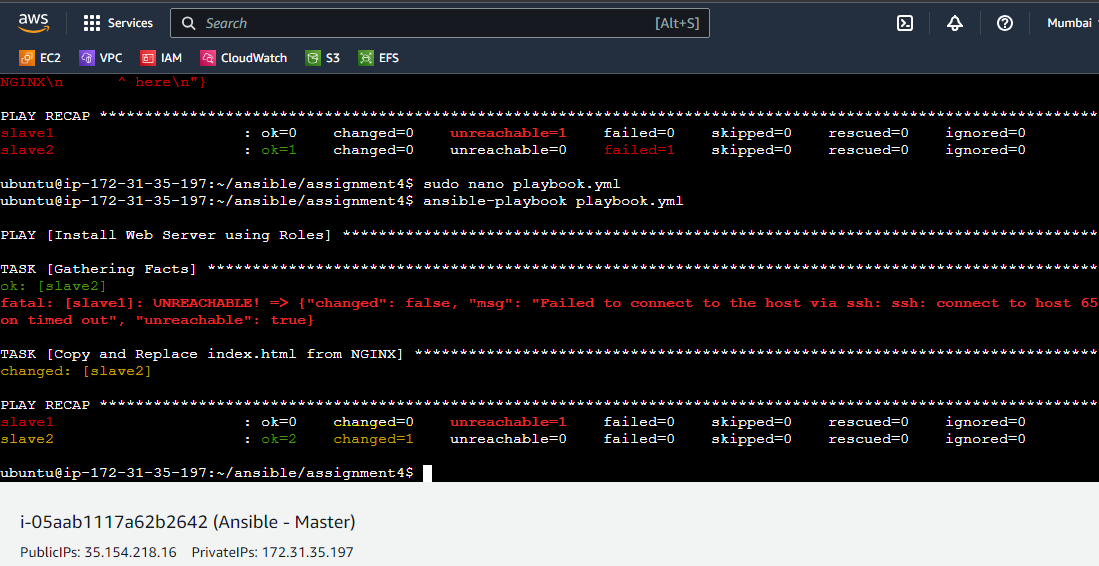
1. Create a files folder and create index.html file with html content inside it.



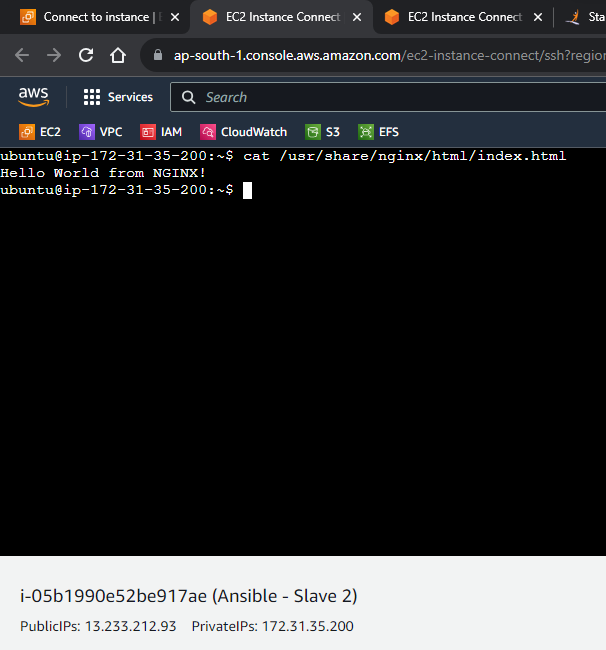
1. Create a playbook with a task stating copy the file from master to slave.



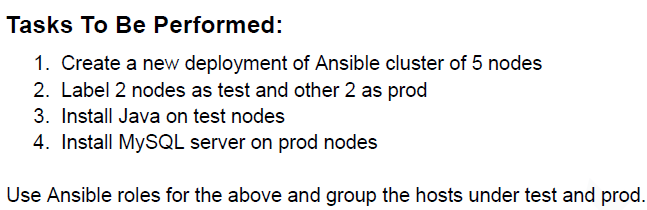
1. Execute the Playbook. (slave1 is Stopped in my AWS, hence it’s unreachable)



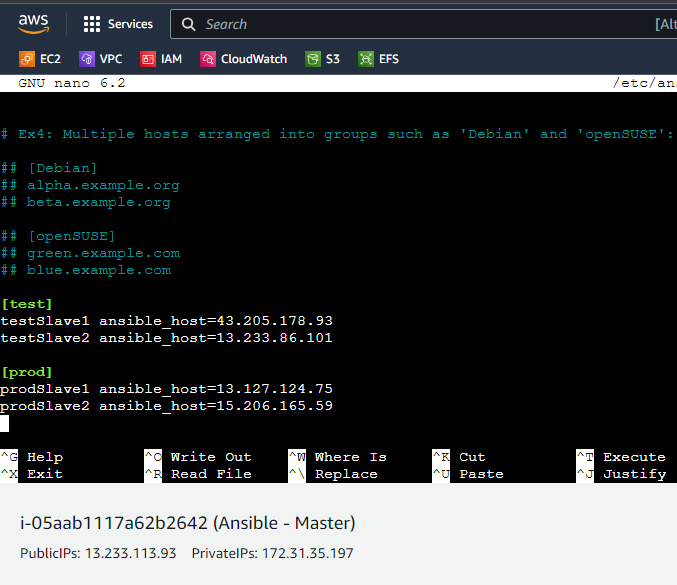
1. Once executed, login to Slave 2 machine and verify if the index.html has been replaced.



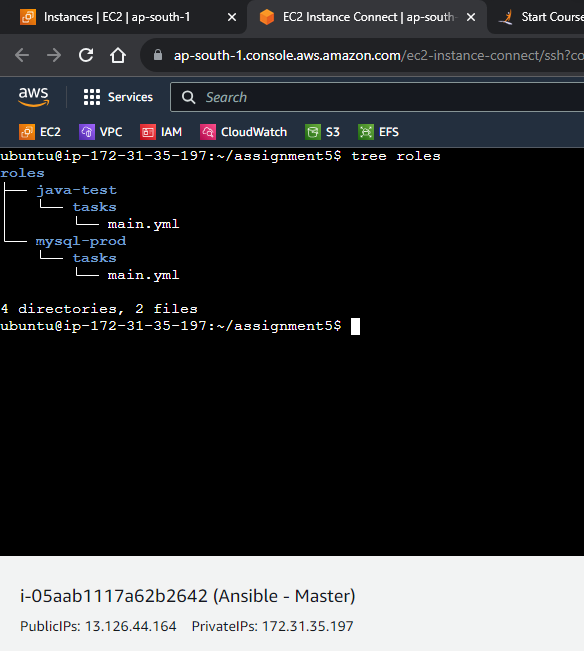
**Assignment – 5 (Ansible)**

****

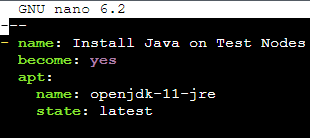
1. Group the Ansible Hosts under /etc/ansible/hosts as test and prod.



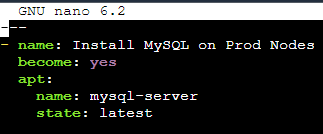
1. Create 2 Roles for Java and MySQL.



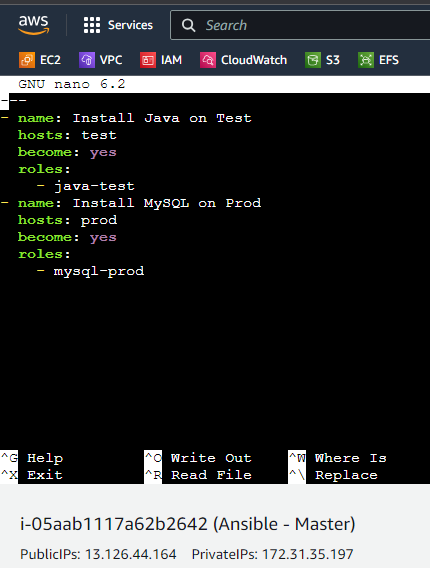
1. Inside roles/java-test/tasks/main.yml, write a playbook to install java.



1. Inside roles/mysql-prod/tasks/main.yml, write a playbook to install mysql.



1. Create a playbook to execute the roles.



1. Execute the playbook. As yo ucan see, all the 4 nodes have been sucessfully modified and installed the respective packages (prod / test)

