Assignments No. 6		
Assignments Name	: Write an ansible-playbook to install nginx on target servers.	
Students Name	: 333047 Rohan N	
Division	TY-IT-C	**GR-No.** : 22010064
Professor Name	: Prof. Vishal Meshram	

Theory:

1) What is YAML

YAML is a human-readable data serialization language that is often used for writing configuration files. Depending on whom you ask, YAML stands for yet another markup language or YAML ain't markup language (a recursive acronym), which emphasizes that YAML is for data, not documents.

YAML is a popular programming language because it is designed to be easy to read and understand. It can also be used in conjunction with other programming languages. Because of its flexibility and accessibility, YAML is used by the Ansible automation tool to create automation processes, in the form of Ansible Playbooks.

2) Introduction to Ansible

Ansible is simple open source IT engine which automates application deployment, intra service orchestration, cloud provisioning and many other IT tools.

Ansible is easy to deploy because it does not use any agents or custom security infrastructure.

Ansible uses playbook to describe automation jobs, and playbook uses very simple language i.e. **YAML** (It's a human-readable data serialization language & is commonly used for configuration files, but could be used in many applications where data is being stored)which is very easy for humans to understand, read and write. Hence the advantage is that even the IT infrastructure support guys can read and understand the playbook and debug if needed (YAML – It is in human readable form).

Ansible is designed for multi-tier deployment. Ansible does not manage one system at time, it models IT infrastructure by describing all of your systems are interrelated.

Ansible is completely agentless which means Ansible works by connecting your nodes through ssh(by default). But if you want other method for connection like Kerberos, Ansible gives that option to you.

After connecting to your nodes, Ansible pushes small programs called as "Ansible Modules". Ansible runs that modules on your nodes and removes them when finished. Ansible manages your inventory in simple text files (These are the hosts file). Ansible uses the hosts file where one can group the hosts and can control the actions on a specific group in the playbooks.

Implementation:

1. Architecture:
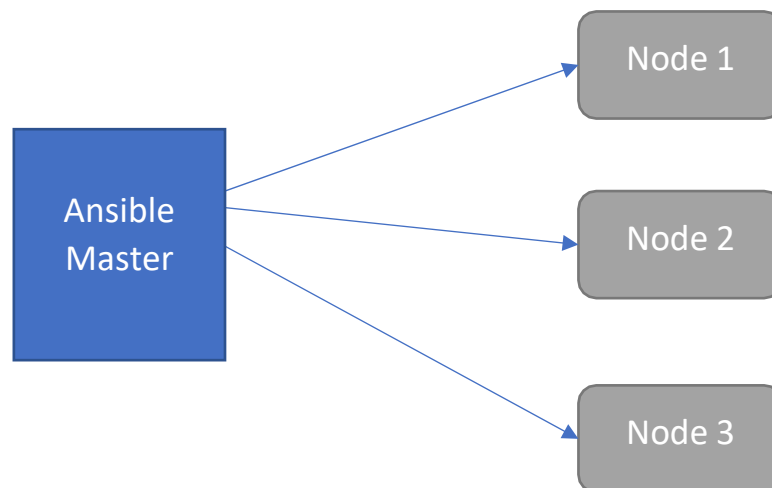
Figure1: Architecture Diagram

2. Steps

a) Create 4 ec2 instances of **Ubuntu machine.**
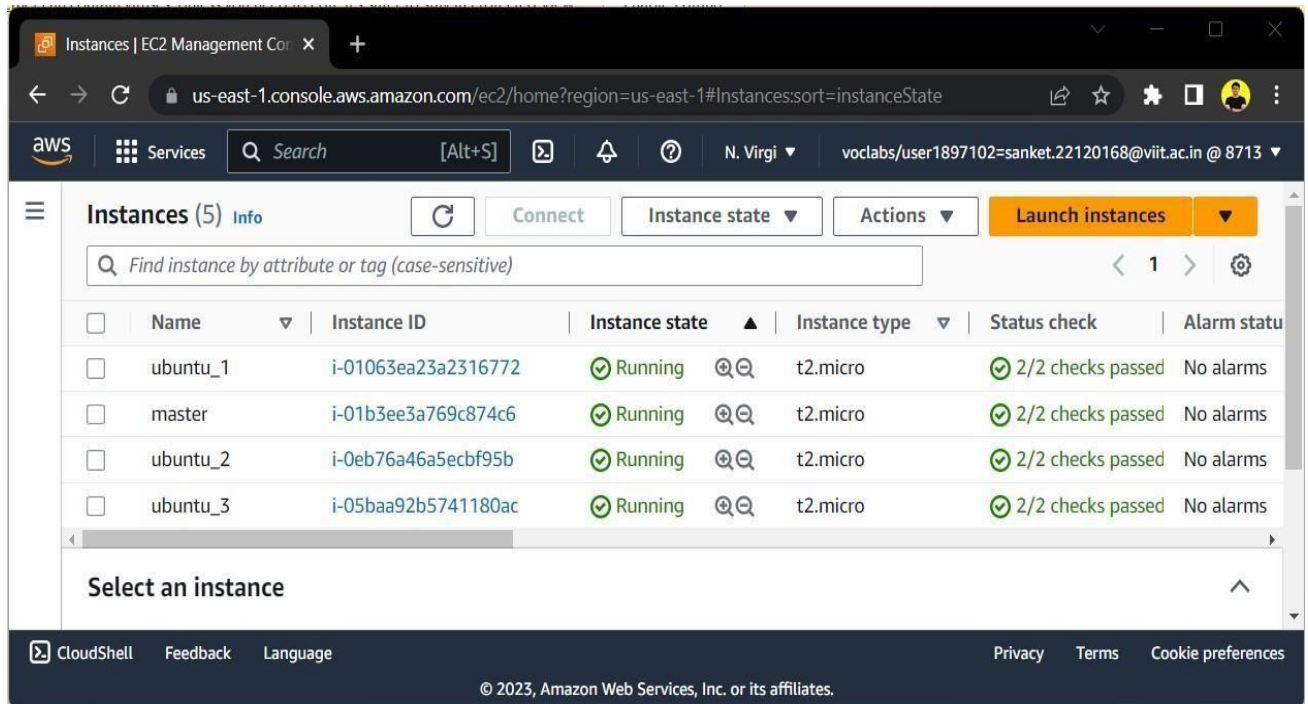
b) Connect to **"Ansible-Master"** server

c) Write following commands

1) > sudo -i
2) > apt update

3) Install ansible using command
   ➢ apt install ansible

a) Generate a ssh key on Ansible-master using command

ssh-keygen

a) Now login to Ansible-master and try to connect to ansible server using command
> **ssh ubuntu@private-ip**

a) copy the public key which is in .ssh folder into "authorized keys" on ansible-server1
commands:
1) ls ~/.ssh
2) cat ~/.ssh/id_rsa.pub
b) connect to ansible-server1 and again give command
> **ssh-keygen**
It will create the same files on ansible-server1

Department Of Information Technology

Now,

> **vim ~/.ssh/authorized_keys**
and Copy the public key

```
ubuntu@ip-172-31-83-59: ~/.:   ×   +   ∨                                    —   □   ×

ubuntu@ip-172-31-83-59:~$ python3 --version
Python 3.10.6
ubuntu@ip-172-31-83-59:~$ cd .ssh
ubuntu@ip-172-31-83-59:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-83-59:~/.ssh$ sudo nano authorized_keys

ubuntu@ip-172-31-83-59:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDJxmZ8H1C7FJqGjTVrOBCGYrqgB7KBnYyenTWQCmyc5gt+xPX/
6yh6biManD15lDG0eYuQggakxzp9J3Ax89vp6oJj01cmca0VBZ41d8XEQ8FMv+9g4yLFngkCF5Rs9BAqMo4EBsWb
dPlUl0jwQD4BMNtm3ASmMhJ8CrBZxcvk3FJzdEZudsH2Fgs47dI5olGhs/dGi9thz0M+viEaMEqhkh07+84QcTr9
JhBBPePRLi6iVqF+j0HKalC3ZFUQHk9eFqvYS7LYErz4XzUfKWivuMWqwbB1tMSVLOOsdPLNEy9gN6w810X6n+/6
32FU8hDFbWh+mJSqJlqion6cylEN ubuntu_server_1_key
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCYRXtfn6z1OgB5SyvM3k5zhsyF3X6AnKH+Uz1jRGDRUtQVTRm7
+39F+AX5ZA686qan+mx18ofFs7mKN+OWUSV96iB06QOrgvXuafcgY9+i5h6rHtA8v48efQiFZMRXKksrwQDTi9Oh
kFQ5kNspGl83Ohna4dq+I/D6bgJyY45mMrxO8KG0I3GfZ7Or5zExLMVUVb9es9Gg8FcGpIXxUIaBNfS2lKBGwyt6
X6IGYeMnJ0iNYpvL6SnZhJeeT2/XaptyhI9EntudfzE0K2sBV10/ZduuKnDvTecbHjcY3HkodPE3RjC3p9jhgxqR
bBhpRZsxYCxKQYlyrAKcge06NUtOQMLL2PlaB9mfLUwsD7wc2NezHRtVSAzaEhcHuwmeh672U0ttgWxfhqjy8j+T
zdCsoeWBIYw62oXAsmjNjbWeCQEpN0FJmRTe5mswq46WI1HjhkUhpNx+C9WR/kotm5Nb5hQzdAygEI7O4oKQQV95
vIu99BDYnKzjNamz7dLttSU= ubuntu@ip-172-31-89-14
ubuntu@ip-172-31-83-59:~/.ssh$
```

```
ubuntu@ip-172-31-89-14: ~/.s   ×   +   ∨                                    —   □   ×

}
ubuntu@ip-172-31-89-14:~/.ssh$ ansible -m ping all
ubuntu_1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu_3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu_2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-89-14:~/.ssh$
```
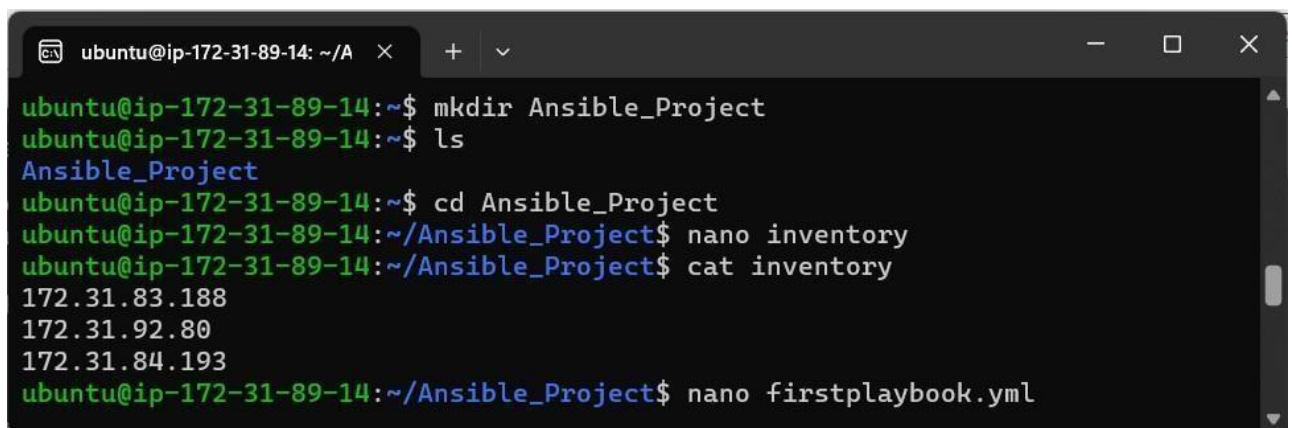
Create a playbook on Ansible-master

Step 1:- connect to "Ansible-Master"

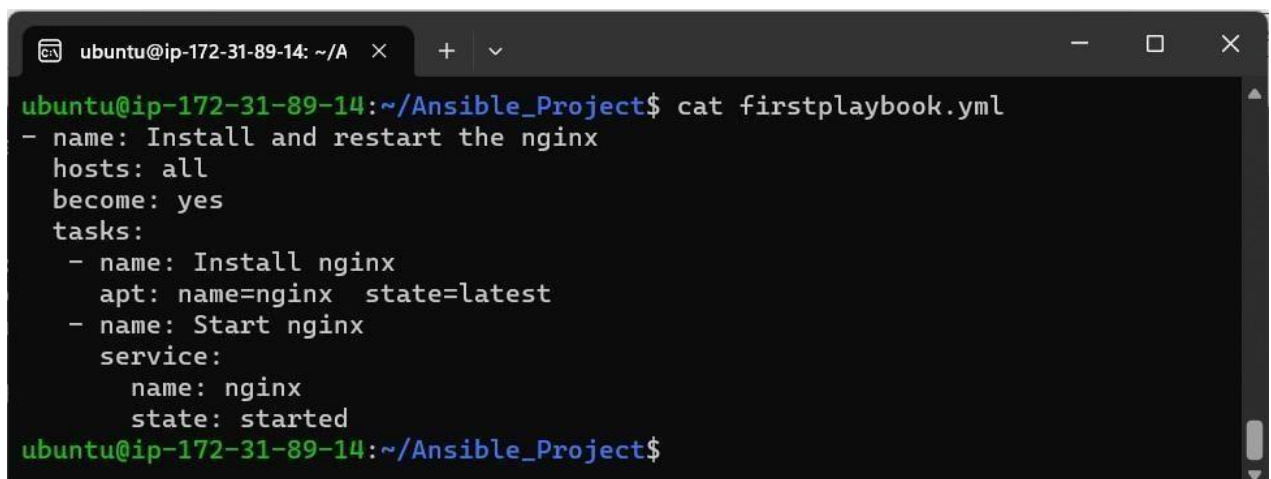**Step 2:-**create a new folder "ansible-project" using command

Step 3:

a) > cd ansible-project
b) > nano inventory
c) > write a private IP of "Ansible-server1" into inventory
d) > write a private IP of "Ansible-server2" into inventory
e) > write a private IP of "Ansible-server3" into inventory

```
ubuntu@ip-172-31-89-14:~$ mkdir Ansible_Project
ubuntu@ip-172-31-89-14:~$ ls
Ansible_Project
ubuntu@ip-172-31-89-14:~$ cd Ansible_Project
ubuntu@ip-172-31-89-14:~/Ansible_Project$ nano inventory
ubuntu@ip-172-31-89-14:~/Ansible_Project$ cat inventory
172.31.83.188
172.31.92.80
172.31.84.193
ubuntu@ip-172-31-89-14:~/Ansible_Project$ nano firstplaybook.yml
```

Task: Install Nginx and Start Nginx

Step 1: Create a new file called "first-playbook.yml"

```
ubuntu@ip-172-31-89-14:~/Ansible_Project$ cat firstplaybook.yml
- name: Install and restart the nginx
  hosts: all
  become: yes
  tasks:
   - name: Install nginx
     apt: name=nginx  state=latest
   - name: Start nginx
     service:
       name: nginx
       state: started
ubuntu@ip-172-31-89-14:~/Ansible_Project$
```

Execute the playbook by using command:

Ansible-playbook -i inventory first-playbook.yml

Verify the output:

Step 1: connect to any ansible-server1

Step run the command: sudo systemctl status nginx