

St. Francis Institute of Technology
Department of Computer Engineering
Five Days Student Development Programme
on
DevOps
(28th June to 2nd July 2021)

Assignment 03: **Ansible**

- 1. Launch two virtual machines having ubuntu and connect them over the same network.**
- 2. Install Ansible on one of the machines**
- 3. Write Ansible playbook for the following tasks**
 - **Install packages like {postgres-sql, Nginx} on the other machine**
 - **Create a username “yourname” in the remote machine**
 - **Create a folder name “your_Roll_no” in the home directory**
 - **Uninstall the nano service on a remote machine.**
 - **Take screenshots of the playbook and the play in execution**
 - **Name the play in the playbook as yourname-rollno**
- 4. Take screenshots of all the steps and prepare a report for the same.**
- 5. Upload the Ansible Playbook in the GitHub Repository you created in the first task**

NAME: Saravana sundar Nadar

ROLL NO: 72

PARTICIPANT ENROLLMENT ID: 192072

CLASS: SE CMPN A

➡ **Github Link for Ansible Playbook -**

<https://github.com/saravana-sn/Saravana-sundar-192072/blob/master/playbook-192072.yml>

ANSIBLE

1.Launching two virtual ubuntu Machines using amazon aws EC2 instances.

The screenshot displays the AWS Management Console interface for EC2 instances. The left sidebar shows navigation options like EC2 Dashboard, Events, Tags, Limits, and various instance types. The main content area shows a list of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. Two instances are listed: 'ansible-master' (i-098633598efca457b) and 'ansible-slave' (i-04eeb0de6b5721a8c), both in a 'Running' state. Below the list, the 'Networking' tab for the 'ansible-master' instance is expanded, showing details like Public IPv4 address (15.206.211.96), Private IPv4 addresses (172.31.4.62), VPC ID (vpc-a34d84c8), Subnet ID (subnet-db2b0c97), and Availability zone (ap-south-1b).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
ansible-master	i-098633598efca457b	Running	t2.micro	Initializing	No alarms	ap-south-1b	ec2-15-206-
ansible-slave	i-04eeb0de6b5721a8c	Running	t2.micro	Initializing	No alarms	ap-south-1b	ec2-3-108-2

Instance: i-098633598efca457b (ansible-master)

Networking details

Public IPv4 address	Private IPv4 addresses	VPC ID
15.206.211.96 open address	172.31.4.62	vpc-a34d84c8 VPC
Public IPv4 DNS	Private IPv4 DNS	Subnet ID
ec2-15-206-211-96.ap-south-1.compute.amazonaws.com open address	ip-172-31-4-62.ap-south-1.compute.internal	subnet-db2b0c97 Subnet
IPv6 addresses	Secondary private IPv4 addresses	Availability zone
-	-	ap-south-1b
Carrier IP addresses (ephemeral)	Outpost ID	

This screenshot shows the 'ansible-slave' instance details in the AWS Management Console. The instance is in a 'Running' state. The 'Networking' tab is selected, displaying the instance's network configuration. A tooltip indicates that the 'Public IPv4 DNS' has been copied. The instance is associated with the same VPC and Subnet as the 'ansible-master' instance.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
ansible-master	i-098633598efca457b	Running	t2.micro	Initializing	No alarms	ap-south-1b	ec2-15-206-
ansible-slave	i-04eeb0de6b5721a8c	Running	t2.micro	Initializing	No alarms	ap-south-1b	ec2-3-108-2

Instance: i-04eeb0de6b5721a8c (ansible-slave)

Networking details

Public IPv4 address	Private IPv4 addresses	VPC ID
3.108.215.65 open address	172.31.10.108	vpc-a34d84c8 VPC
Public IPv4 DNS	Private IPv4 DNS	Subnet ID
Public IPv4 DNS copied ec2-3-108-215-65.ap-south-1.compute.amazonaws.com open address	ip-172-31-10-108.ap-south-1.compute.internal	subnet-db2b0c97 Subnet
IPv6 addresses	Secondary private IPv4 addresses	Availability zone
-	-	ap-south-1b
Carrier IP addresses (ephemeral)	Outpost ID	

2. Installing Ansible on one of the machines - [ansible-master]

```
root@ip-172-31-4-62:/home/ubuntu# ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0]
root@ip-172-31-4-62:/home/ubuntu#
```

➡ Configuring both master and slave machines to connect with each other using SSH connection.

```
#blue.example.com
#192.168.100.1
#192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group
[slave_1]
172.31.10.108
#[webservers]
#alpha.example.org
#beta.example.org
#192.168.1.100
#192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

#www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group
```

```
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Just
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text   ^T To S
```

```

12. ansible-master
13. ansible-slave

root@ip-172-31-4-62:/home/ubuntu# ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0]
root@ip-172-31-4-62:/home/ubuntu# nano /etc/ansible/hosts
root@ip-172-31-4-62:/home/ubuntu# ssh-keygen -t rsa
ssh-keygen-t: command not found
root@ip-172-31-4-62:/home/ubuntu# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Vm0s9uDXFml0nXeAsWzcGCe+TsPFFZz0DebV88Ps5QU root@ip-172-31-4-62
The key's randomart image is:
+---[RSA 3072]---+
|
|      +o+.=|
|      *.0E0+|
|      = @o&=B|
|      + B B.BB|
|      S . B =.+|
|      . + o ..|
|      .      |
|      |      |
+---+

+----[SHA256]-----+
root@ip-172-31-4-62:/home/ubuntu# cd /root/.ssh/
root@ip-172-31-4-62:~/ssh# la -al
total 20
drwx----- 2 root root 4096 Jul 10 13:21 .
drwx----- 6 root root 4096 Jul 10 13:17 ..
-rw----- 1 root root 562 Jul 10 12:33 authorized_keys
-rw----- 1 root root 2610 Jul 10 13:21 id_rsa
-rw-r--r-- 1 root root 573 Jul 10 13:21 id_rsa.pub
root@ip-172-31-4-62:~/ssh#

```

```

+----[SHA256]-----+
root@ip-172-31-4-62:/home/ubuntu# cd /root/.ssh/
root@ip-172-31-4-62:~/ssh# la -al
total 20
drwx----- 2 root root 4096 Jul 10 13:21 .
drwx----- 6 root root 4096 Jul 10 13:17 ..
-rw----- 1 root root 562 Jul 10 12:33 authorized_keys
-rw----- 1 root root 2610 Jul 10 13:21 id_rsa
-rw-r--r-- 1 root root 573 Jul 10 13:21 id_rsa.pub
root@ip-172-31-4-62:~/ssh# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDAXzmPPg/upEUjPHXiAXaTLB5ZnK8iVwWCZL26EEazhfTANimx0Ppjvnc0Fbg0spI3tEZYknhr6LP5PtUnH1X9WwU8hzGIhVjhPbv41n
T1vSHWHZMzQ9M0P/91cy0Ext54y0xNFsyVDYdeMsVePS4rJ6RPWN+gIHZZoL8z+JsaIMMq+yQm1f/FyEfJiySr70tCFxY7WugLFzfl6YleH0owlSxteTBB3a2S5/kXDGs1Z3dBb3vyxbsb
JpNti7k0Y40nn8UA7mFssA0bDhsGY5Lm1cdHdpQEBZt+9LRSgNpw69Ukfo+fs/00YwQxd0XrU0mHQKrHmxgyu12jz4PCGcbIhFLFQJhSWPaFAIH6J8hnZkrHtcU83f2ZuaZTyXkY5WCgd0
4v0BhffULA8EPBDLVGEBz7B+sIEvgDCD0PJCfSpBDYUvG8vrMIMhBYeWSS+5L156LvteHJS5dLFEE= root@ip-172-31-4-62
root@ip-172-31-4-62:~/ssh#

```

```

root@ip-172-31-4-62:~/ssh# ansible -m ping all
172.31.10.108 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
root@ip-172-31-4-62:~/ssh# ansible -m setup all
172.31.10.108 | SUCCESS => {
  "ansible_facts": {
    "ansible_all_ipv4_addresses": [
      "172.31.10.108"
    ],
    "ansible_all_ipv6_addresses": [
      "fe80::884:35ff:fed7:4252"
    ],
    "ansible_apparmor": {
      "status": "enabled"
    },
    "ansible_architecture": "x86_64",
    "ansible_bios_date": "08/24/2006",
    "ansible_bios_version": "4.11.amazon",
    "ansible_cmdline": {
      "BOOT_IMAGE": "/boot/vmlinuz-5.4.0-1045-aws",
      "console": "ttyS0",
      "nvme_core.io_timeout": "4294967295",
      "panic": "-1"
    }
  }
}

```

3. Writing Ansible playbook for following tasks :

- Install packages like {postgres-sql, Nginx} on the other machine
- Create a username “yourname” in the remote machine
- Create a folder name “your_Roll_no” in the home directory
- Uninstall the nano service on a remote machine.
- Take screenshots of the playbook and the play in execution
- Name the play in the playbook as yourname-rollno

```
GNU nano 4.8                                playbook-192072.yml
--
#saravana-sundar-nadar-192072

- name: Saravana sundar Nadar - 192072
  hosts: slave_1
  remote_user: root
  vars:
    passwd: $6$S2n0uh0P$un8G.gV3s6vAN5XryMx0MkzDo90U4etX68z6DBzDZWTYQVnmuGGbJV0sQrnQCxE.Hutca/qrrqyvEp3Bkbqu0
  tasks:
    - name: Installing Packages
      become: yes
      become_user: root
      apt:
        pkg:
          - postgresql
          - nginx
        state: present
        update_cache: yes

    - name: Create user with my name-saravana
      user:
        name: saravana
        password: "{{ passwd }}"

    - name: Create Directory with my rollno-192072
      file:
        path: /home/192072
        state: directory

    - name: Uninstalling nano
```

[Read 31 lines]

```
- name: Create Directory with my rollno-192072
  file:
    path: /home/192072
    state: directory

- name: Uninstalling nano
  apt: name=nano state=absent
```

➡ Executing Created ansible playbook.

```
12. ansible-master 13. ansible-slave
root@ip-172-31-4-62:/home# nano playbook-192072.yml
root@ip-172-31-4-62:/home# ansible-playbook playbook-192072.yml

PLAY [Saravana sundar Nadar - 192072] *****

TASK [Gathering Facts] *****
ok: [172.31.10.108]

TASK [Installing Packages] *****
ok: [172.31.10.108]

TASK [Create user with my name-saravana] *****
changed: [172.31.10.108]

TASK [Create Directory with my rollno-192072] *****
changed: [172.31.10.108]

TASK [Uninstalling nano] *****
changed: [172.31.10.108]

PLAY RECAP *****
172.31.10.108 : ok=5 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

root@ip-172-31-4-62:/home#
```

➡ New user successfully created in client machine [ansible-slave] by ansible playbook.

```
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
landscape:x:110:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false
ec2-instance-connect:x:112:65534::/nonexistent:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
postgres:x:113:120:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
saravana:x:1001:1001::/home/saravana:/bin/sh
root@ip-172-31-10-108:~# su saravana
$
```

Upload the Ansible Playbook in the GitHub Repository you created in the first task

➡ Github Link for Ansible Playbook -

<https://github.com/saravana-sn/Saravana-sundar-192072/blob/master/playbook-192072.yml>