

## SCOPE:

Ansible is a popular automation tool that allows you to automate IT tasks such as configuration management, application deployment, and infrastructure management. Its scope is vast, and it can be used to automate a wide range of applications and systems.

Some of the popular use cases for Ansible automation include:

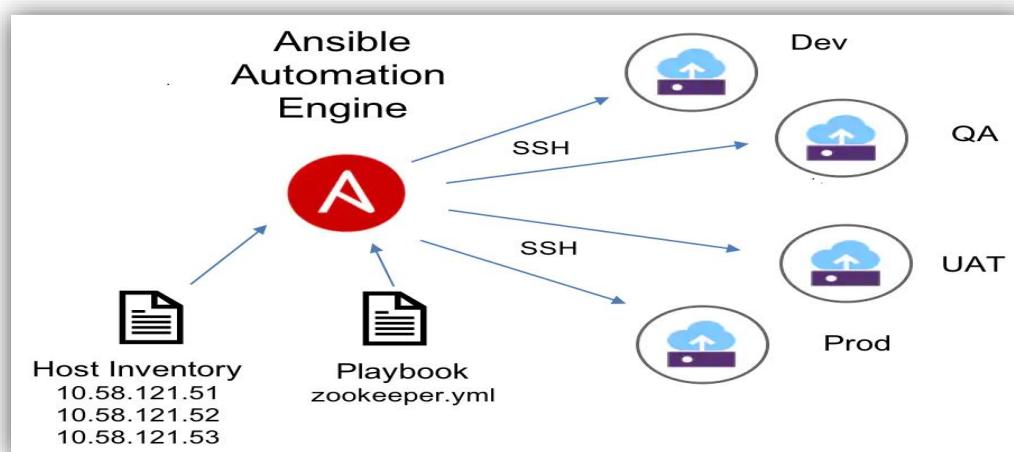
**Configuration management:** Ansible can be used to automate the configuration of servers, networking devices, and other IT resources. This includes tasks such as setting up users and permissions, installing software and packages, and configuring services.

**Application deployment:** Ansible can automate the deployment of applications and their dependencies, including web servers, databases, and load balancers. This can be particularly useful in environments where frequent updates and deployments are required.

**Infrastructure management:** Ansible can automate the provisioning of infrastructure resources such as virtual machines, containers, and cloud instances. This includes tasks such as creating and configuring network interfaces, storage volumes, and security groups.

**Security automation:** Ansible can be used to automate security-related tasks such as vulnerability scanning, patch management, and compliance auditing.

Overall, Ansible's flexibility and wide range of capabilities make it a valuable tool for automating various IT tasks and applications.



## PURPOSE:

The purpose of automating applications using Ansible is to simplify and streamline IT operations. By automating routine tasks, such as software installation, configuration management, and infrastructure provisioning, Ansible can save time and reduce the risk of errors that can occur with manual processes.

Some of the key benefits of automating applications with Ansible include

**Improved efficiency:** Ansible enables you to automate repetitive tasks, freeing up time for IT teams to focus on more strategic work.

**Consistency:** Automation with Ansible ensures that IT operations are standardized and consistent, reducing the risk of errors and security vulnerabilities.

**Speed:** Ansible can execute tasks much faster than manual processes, reducing the time it takes to deploy applications and infrastructure.

## TOOLS & TECHNOLOGIES USED:

### Ansible:

- Ansible is an open-source automation tool.
- It uses YAML-based playbooks to define desired system state.
- Ansible is agentless, lightweight, and easy to set up.



### Git & Git-Hub:

- Git is a distributed version control system.
- It allows developers to track changes to their code over time.
- Git enables collaboration among developers by allowing them to work on the same codebase simultaneously.



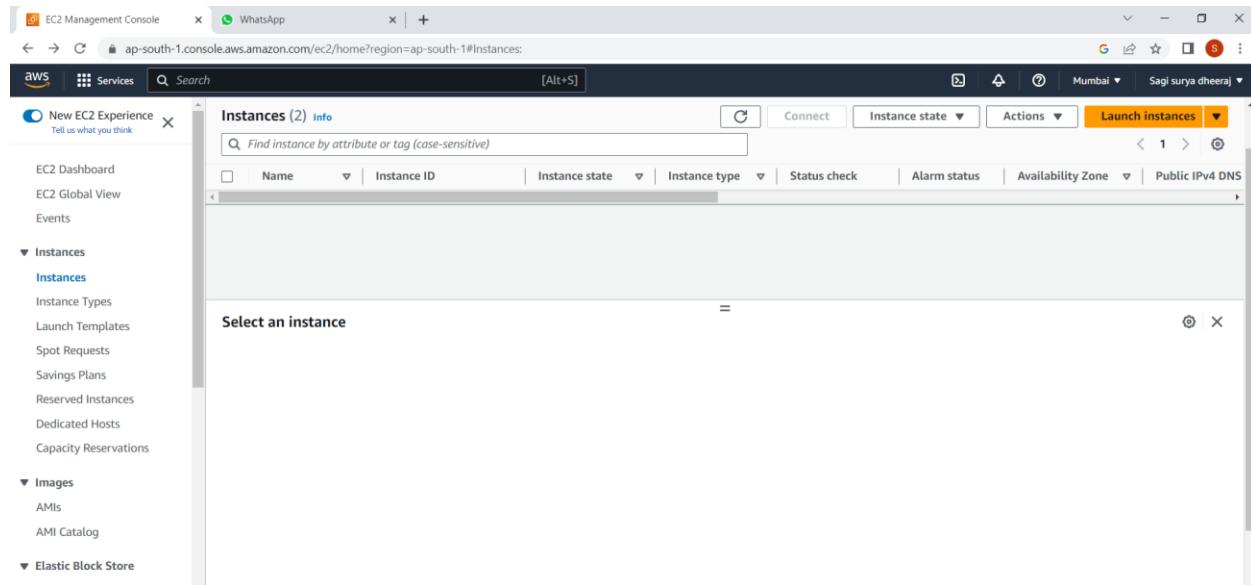
### AWS:

- AWS (Amazon Web Services) is a cloud computing platform that provides a wide range of services and solutions for businesses and individuals.
- It offers on-demand computing resources, such as virtual machines, storage, and databases, that can be accessed over the internet.



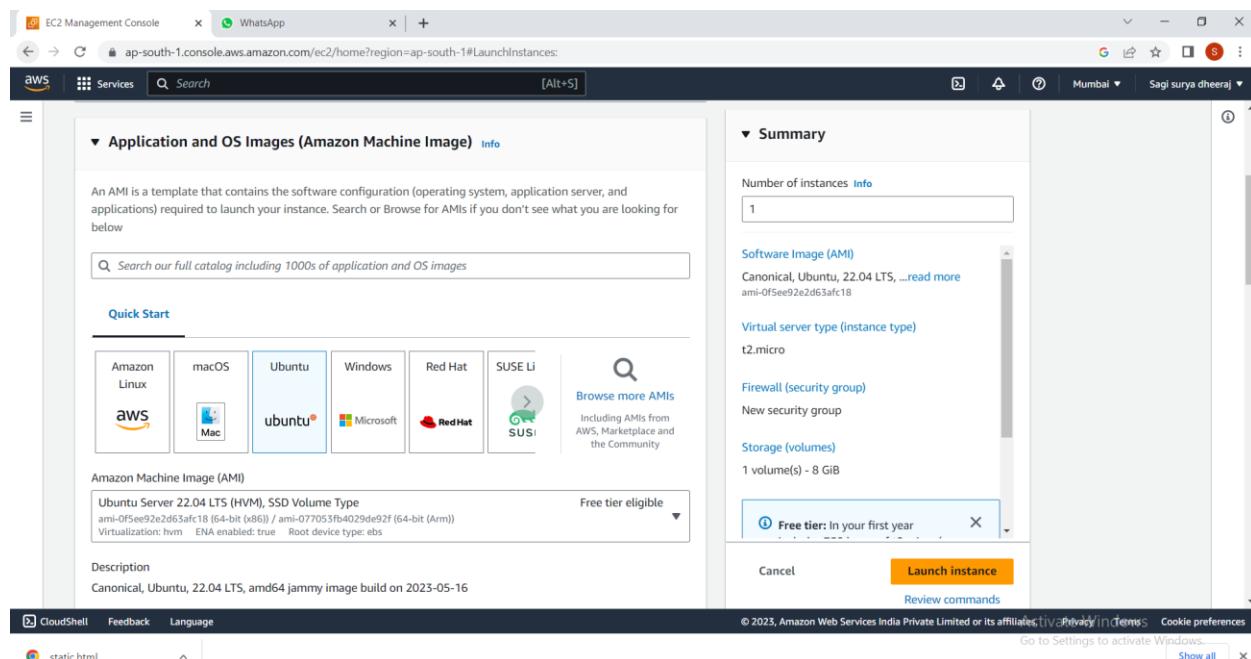
## PROOF OF CONCEPT:-

Login to aws.amazon.com with your credentials and navigate to console home.



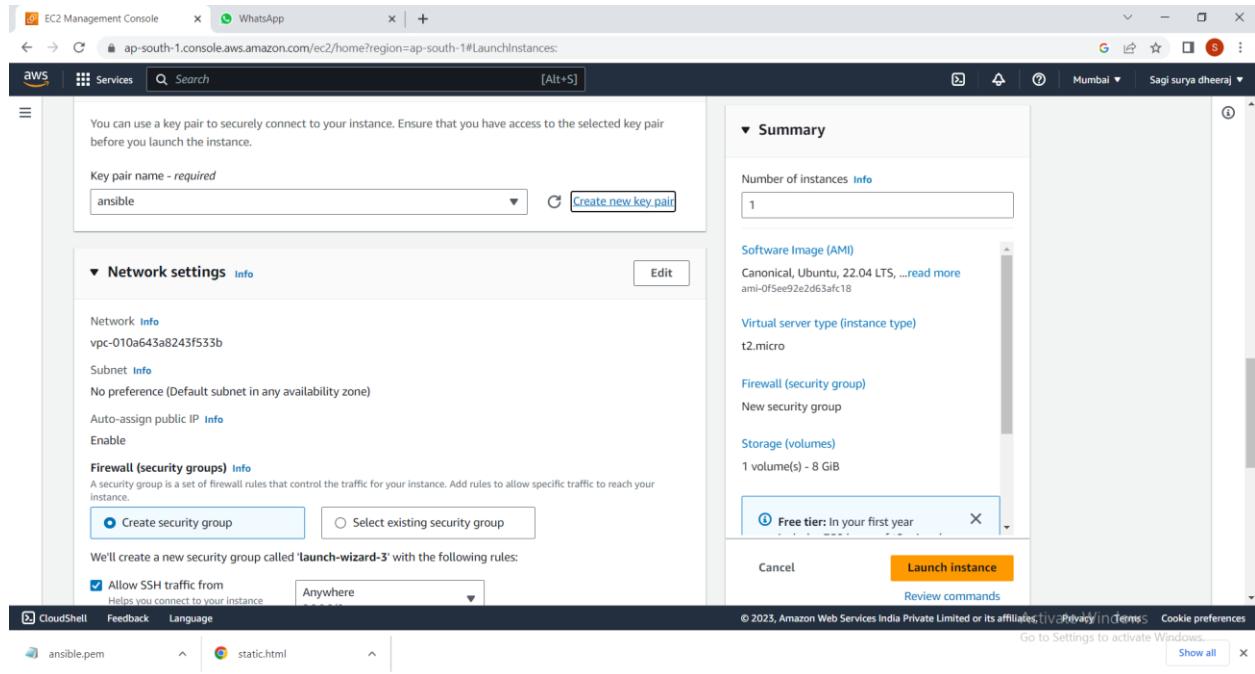
The screenshot shows the AWS EC2 Management Console interface. The left sidebar is collapsed, showing navigation links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store. The main content area is titled 'Instances (2) Info' and displays two instances. The first instance is an Amazon Linux 2 AMI (HVM, SSD Volume Type) with the ID ami-0f5ee92e2d65afc18. The second instance is a Canonical, Ubuntu, 22.04 LTS (HVM, SSD Volume Type) with the ID ami-077055fb4029de92f. The top right of the main area has buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. The status bar at the bottom indicates the user is in Mumbai and has signed in with Sagi surya dheeraj.

Click on instances and select AMI(Amazon Machine Image)ubuntu.



The screenshot shows the 'Launch Instances' wizard step 'Select an instance'. The left panel shows the 'Application and OS Images (Amazon Machine Image)' section, which includes a search bar, a 'Quick Start' section with icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux, and a link to 'Browse more AMIs'. The right panel shows the 'Summary' section, which includes fields for 'Number of instances' (set to 1), 'Software Image (AMI)' (set to Canonical, Ubuntu, 22.04 LTS), 'Virtual server type (instance type)' (set to t2.micro), 'Firewall (security group)' (New security group selected), and 'Storage (volumes)' (1 volume(s) - 8 GiB). A modal dialog at the bottom right says 'Free tier: In your first year' with 'Launch instance' and 'Review commands' buttons. The status bar at the bottom indicates the user is in Mumbai and has signed in with Sagi surya dheeraj.

Create a keypair, keypair allows you to connect to your instance.



Select your keypair and vpc,subnet and enable auto assign public Ip.Select number of instances as 2.

EC2 Management Console | WhatsApp | ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances: [Alt+S]

Services Search [Alt+S]

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type info: ssh | Protocol info: TCP | Port range info: 22

Source type info: Anywhere | Source info: e.g. SSH for admin desktop | Description - optional info: e.g. SSH for admin desktop

Remove

Security group rule 2 (All, All, 0.0.0.0/0)

Type info: All traffic | Protocol info: All | Port range info: All

Source type info: Anywhere | Source info: e.g. SSH for admin desktop | Description - optional info: e.g. SSH for admin desktop

Remove

⚠ Rules with source of 0.0.0.0/ allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule

Summary

Number of instances info: 2

When launching more than 1 instance, consider EC2 Auto Scaling.

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more ami-0f5ee92e2d63af18

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Cancel Launch instance Review commands

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy terms Cookie preferences Go to Settings to activate Windows. Show all

Click on create security group and select inbound security group rules. In our case select ALL Traffic, finally click on launch instance.

EC2 Management Console | WhatsApp | ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances: [Alt+S]

Services Search [Alt+S]

EC2 > Instances > Launch an instance

Success Successfully initiated launch of instances (i-0e883440057c8b6, i-0dc82bed3c31d9268)

Launch log

Next Steps

Q What would you like to do next with these instances, for example "create alarm" or "create backup"

1 2 3 4 5 6 >

Create billing and free tier usage alerts

To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Create billing alerts

Connect to your instance

Once your instance is running, log into it from your local computer.

Learn more

Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Connect an RDS database

Create a new RDS database

Learn more

Create EBS snapshot policy

Create a policy that automates the creation, retention, and deletion of EBS snapshots

Create EBS snapshot policy

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy terms Cookie preferences Go to Settings to activate Windows. Show all

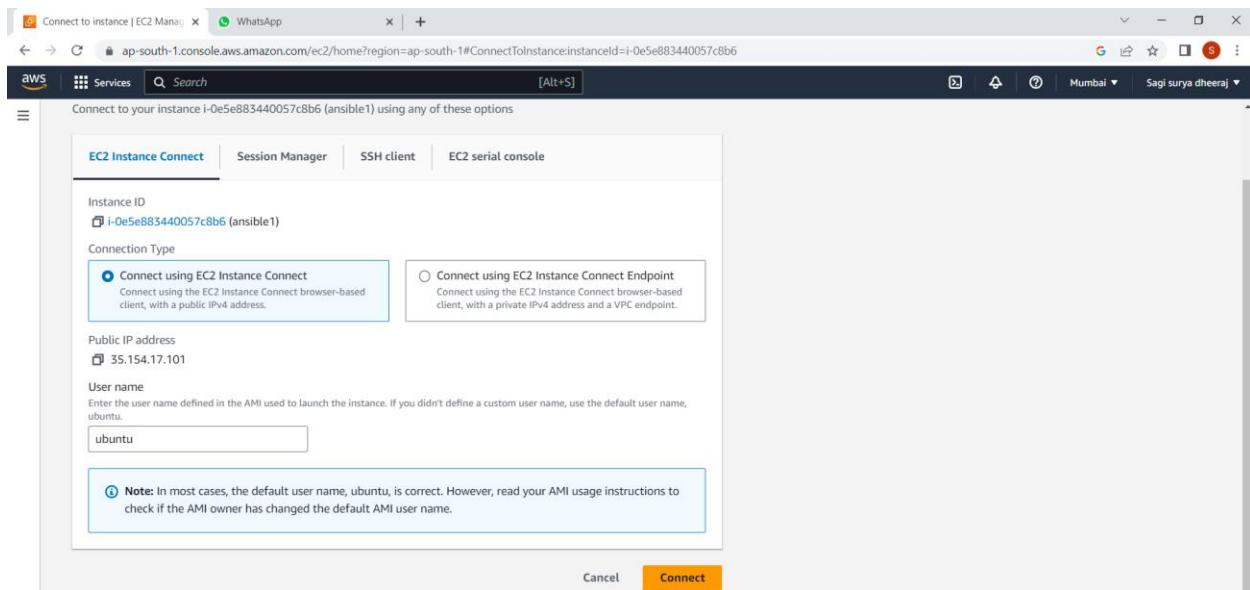
When instance launched.Name instances as ansible1,ansible2.

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays a table of instances:

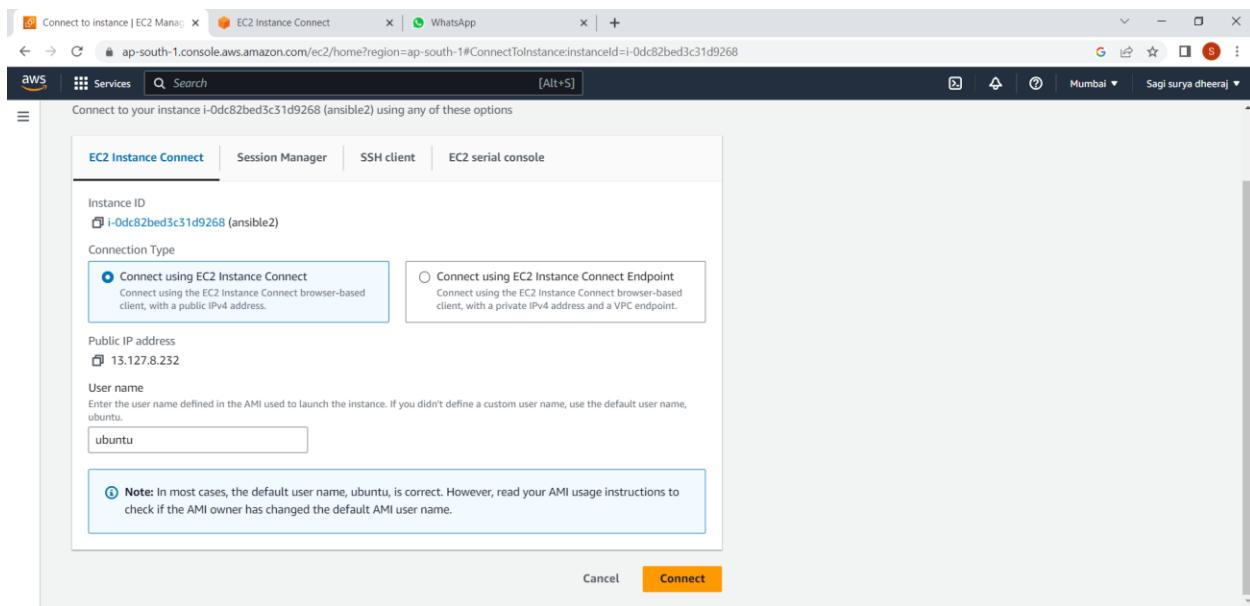
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
ansible1	i-0e5e883440057c8b6	Running	t2.micro	Initializing	No alarms	ap-south-1b	ec2-35-154-17-1
ansible2	i-0dc82bed3c31d9268	Running	t2.micro	Initializing	No alarms	ap-south-1b	ec2-13-127-8-23

Below the table, it says "Instances: i-0dc82bed3c31d9268 (ansible2), i-0e5e883440057c8b6 (ansible1)". The "Monitoring" section shows four charts for CPU utilization (%), Status check failed (any), Status check failed (instance), and Status check failed (system). All charts show "No unit" and "No data available." with a note to "Try adjusting the dashboard time range." The bottom of the screen shows the Windows taskbar with icons for CloudShell, Feedback, Language, ansibile.pem, static.html, and other system icons.

## Connect the ansible1 instance



## Simultaneously connect the ansible2.



We are successfully connected to ansible1.

```
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-13-147:~$
```

i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147

We are successfully connected to ansible2.

```
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

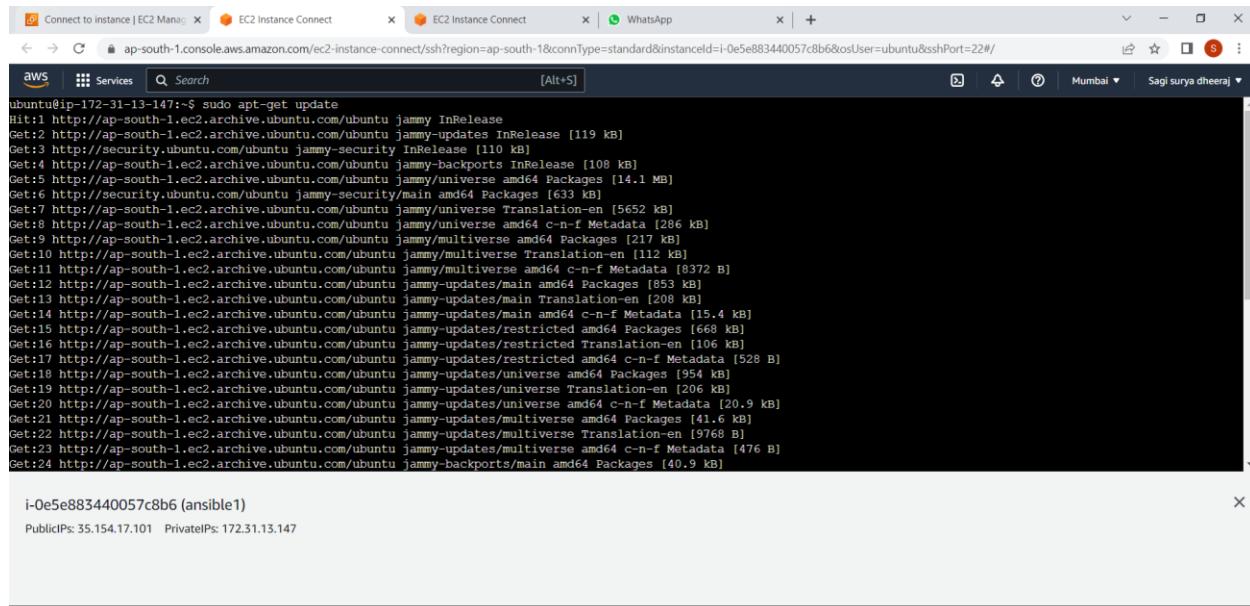
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-2-56:~$
```

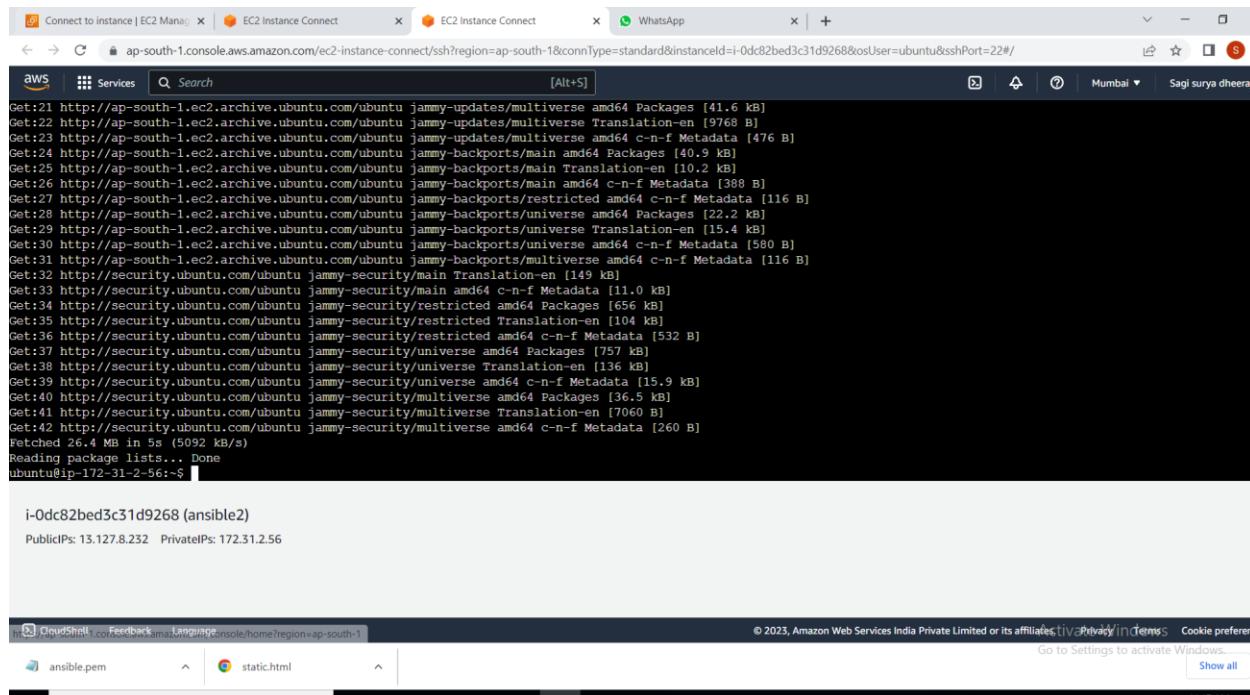
i-0dc82bed3c31d9268 (ansible2)  
PublicIPs: 13.127.8.232 PrivateIPs: 172.31.2.56

Execute this command in ansible1 and ansible2.



```
ubuntu@ip-172-31-13-147:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [633 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates amd64 c-n-f Metadata [8372 B]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [853 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [208 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [15.4 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [668 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [106 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [528 B]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [954 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [206 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [20.9 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [476 B]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [40.9 kB]

i-0e5e883440057c8b6 (ansible1)
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147
```



```
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [476 B]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [40.9 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [10.2 kB]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [22.2 kB]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [15.4 kB]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [580 B]
Get:31 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [149 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.0 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [656 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [104 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [532 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [757 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [136 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [15.9 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 26.4 MB in 9s (5092 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-2-56:~$ 
```

i-0dc82bed3c31d9268 (ansible2)

PublicIPs: 13.127.8.232 PrivateIPs: 172.31.2.56



## Execute the following commands in ansible1

```
ubuntu@ip-172-31-13-147:~$ sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties
The following packages will be upgraded:
  python3-software-properties software-properties-common
  upgraded, 0 newly installed, 0 to remove and 86 not upgraded.
Need to get 42.9 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 software-properties-common all 0.99.22.7 [14.1 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-software-properties all 0.99.22.7 [28.8 kB]
Fetched 42.9 kB in 0s (144 kB/s)
Reading database ... 64295 files and directories currently installed.
Preparing to unpack .../software-properties-common_0.99.22.7_all.deb ...
Unpacking software-properties-common (0.99.22.7) over (0.99.22.6) ...
Preparing to unpack .../python3-software-properties_0.99.22.7_all.deb ...
Unpacking python3-software-properties (0.99.22.7) over (0.99.22.6) ...
Setting up python3-software-properties (0.99.22.7) ...
Setting up software-properties-common (0.99.22.7) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Cleaning processes...
i-0e5e883440057c8b6 (ansible1)
Public IPs: 35.154.17.101 Private IPs: 172.31.13.147
```

cloudShell Feedback Language

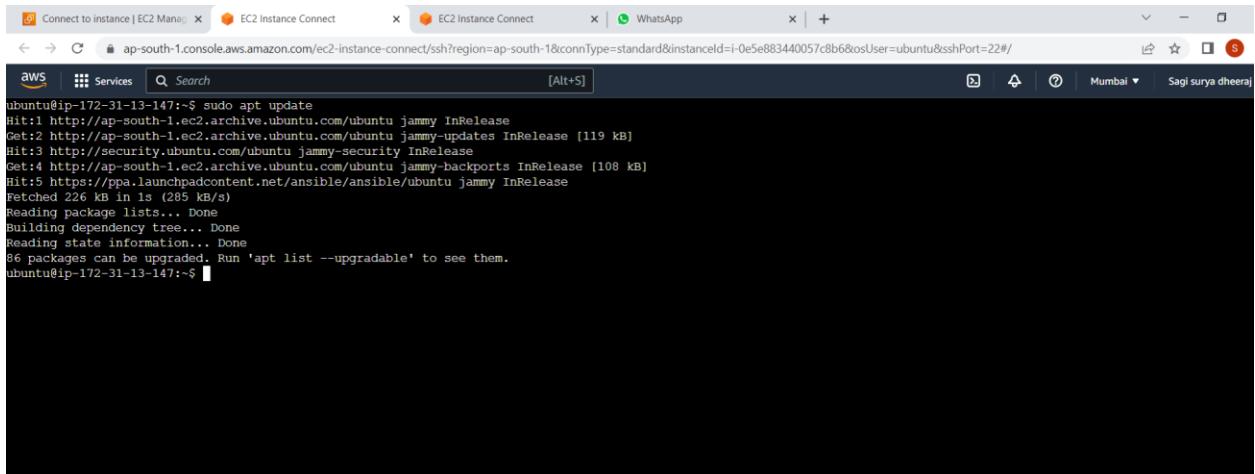
© 2023, Amazon Web Services India Private Limited or its affiliates. All rights reserved. Privacy Terms Go to Settings to activate

```
ubuntu@ip-172-31-13-147:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description: Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications—automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansibile/+archive/ubuntu/ansible
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint 6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease [18.0 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 Packages [1144 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main Translation-en [752 B]
Fetched 19.9 kB in 2s (10.0 kB/s)
Reading package lists... Done
i-0e5e883440057c8b6 (ansible1)
```

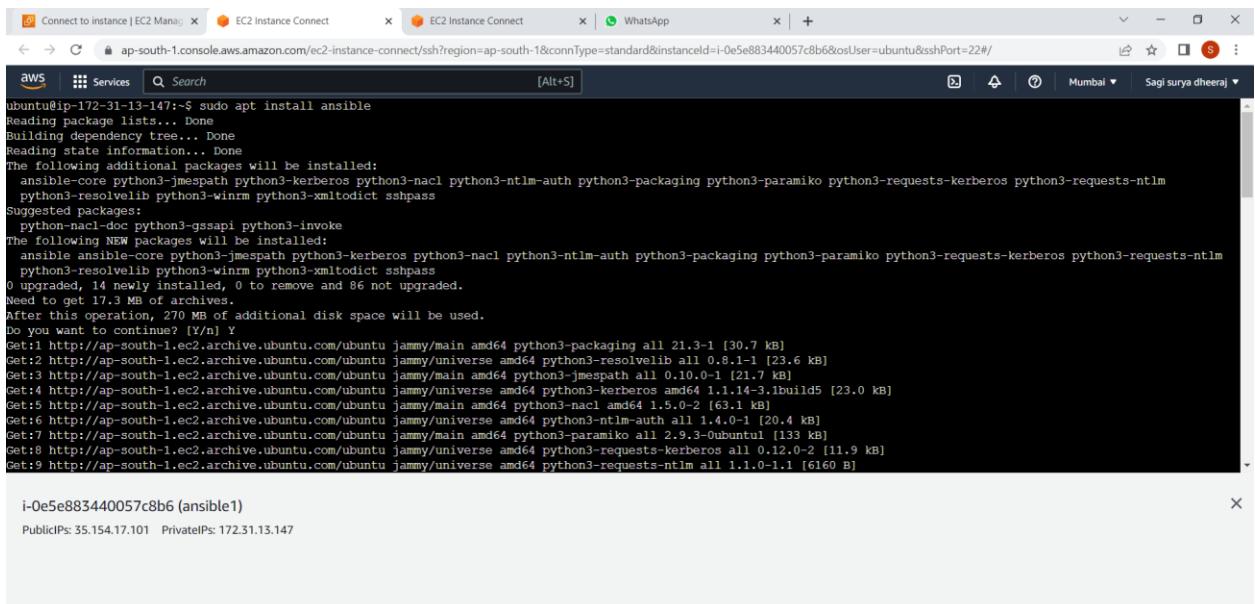
Public IPs: 35.154.17.101 Private IPs: 172.31.13.147



```
ubuntu@ip-172-31-13-147:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Hit:5 https://ppa.launchpadcontent.net/ansible/ubuntu jammy InRelease
Fetched 226 kB in 1s (285 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
86 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-13-147:~$
```

i-0e883440057c8b6 (ansible1)

Public IPs: 35.154.17.101 Private IPs: 172.31.13.147

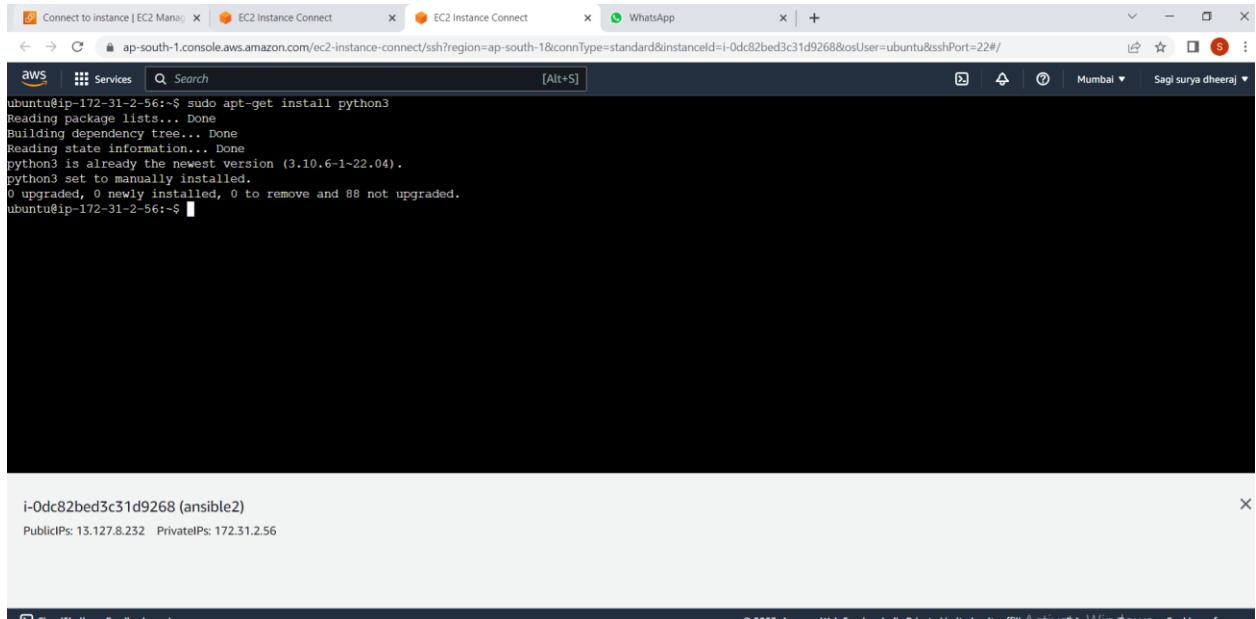


```
ubuntu@ip-172-31-13-147:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko python3-requests-kerberos python3-requests-ntlm
  python3-resolvelib python3-winrm python3-xmldict sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko python3-requests-kerberos python3-requests-ntlm
  python3-resolvelib python3-winrm python3-xmldict sshpass
0 upgraded, 14 newly installed, 0 to remove and 86 not upgraded.
Need to get 17.3 MB of archives.
After this operation, 270 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-packaging all 21.3-1 [30.7 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-resolvelib all 0.8.1-1 [23.6 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-jmespath all 0.10.0-1 [21.7 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-kerberos amd64 1.1.14-3.1build5 [23.0 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-nacl amd64 1.5.0-2 [63.1 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-ntlm-auth all 1.4.0-1 [20.4 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-paramiko all 2.9.3-0ubuntu1 [133 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-requests-ntlm all 1.1.0-1.1 [6160 B]
```

i-0e883440057c8b6 (ansible1)

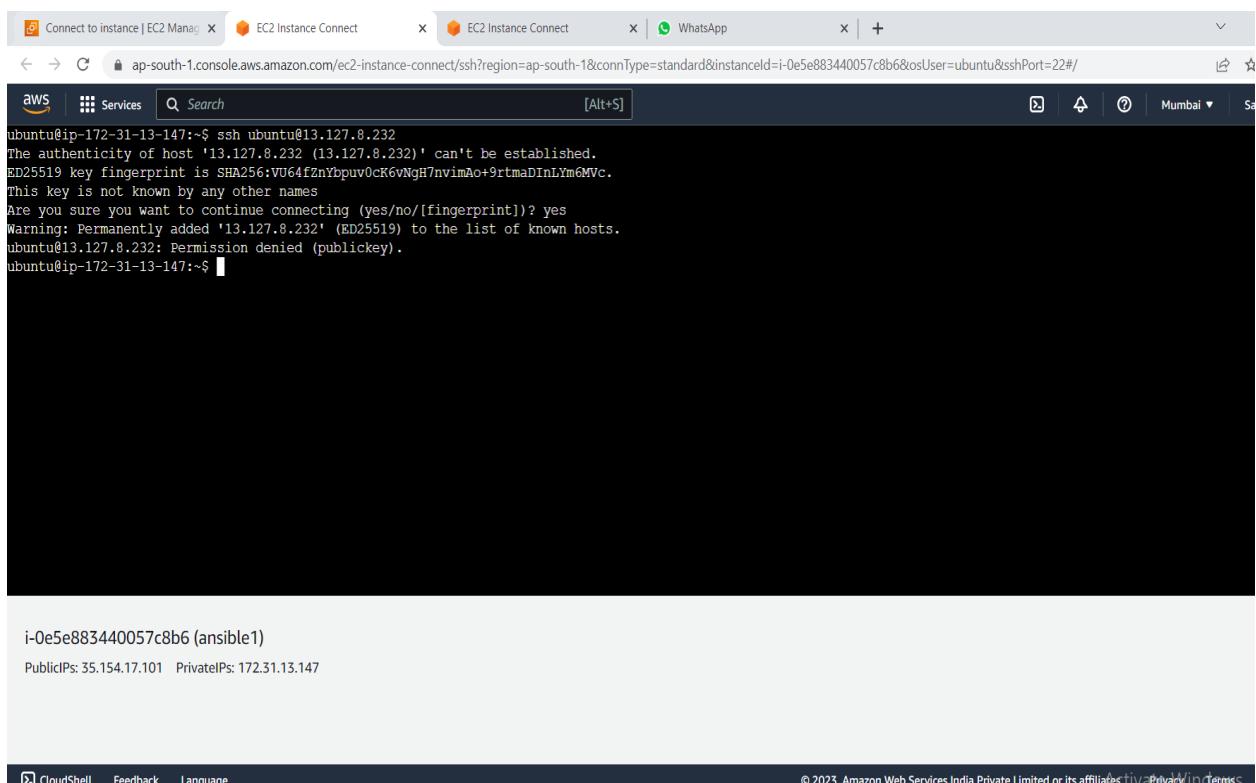
Public IPs: 35.154.17.101 Private IPs: 172.31.13.147

Execute this command in ansible2.



```
aws Connect to instance | EC2 Manager EC2 Instance Connect EC2 Instance Connect WhatsApp +  
← → C ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0dc82bed3c31d9268&osUser=ubuntu&sshPort=22#/ Mumbai Sagi surya dheeraj  
[aws] Services Search [Alt+S]  
ubuntu@ip-172-31-2-56:~$ sudo apt-get install python3  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3 is already the newest version (3.10.6-1~22.04).  
python3 set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 88 not upgraded.  
ubuntu@ip-172-31-2-56:~$  
  
i-0dc82bed3c31d9268 (ansible2)  
Public IPs: 13.127.8.232 Private IPs: 172.31.2.56  
  
CloudShell Feedback Language © 2023 Amazon Web Services India Private Limited or its affiliates. All rights reserved. Privacy Terms
```

Now execute the following the command:-ssh ubuntu@ip(ansible2 ip address) in ansible1.



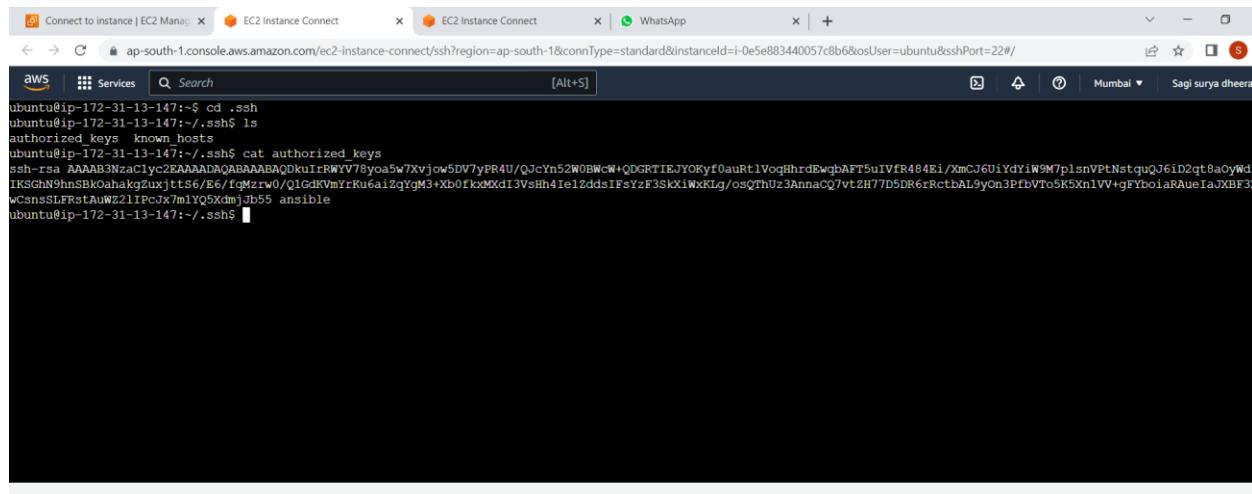
```
aws Connect to instance | EC2 Manager EC2 Instance Connect EC2 Instance Connect WhatsApp +  
← → C ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0e5e883440057c8b6&osUser=ubuntu&sshPort=22#/ Mumbai Sagi surya dheeraj  
[aws] Services Search [Alt+S]  
ubuntu@ip-172-31-13-147:~$ ssh ubuntu@13.127.8.232  
The authenticity of host '13.127.8.232 (13.127.8.232)' can't be established.  
ED25519 key fingerprint is SHA256:VU64fZnYbpuv0cK6vNgH7nvimAo+9rtmaDInLyM6MVC.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '13.127.8.232' (ED25519) to the list of known hosts.  
ubuntu@13.127.8.232: Permission denied (publickey).  
ubuntu@ip-172-31-13-147:~$  
  
i-0e5e883440057c8b6 (ansible1)  
Public IPs: 35.154.17.101 Private IPs: 172.31.13.147  
  
CloudShell Feedback Language © 2023 Amazon Web Services India Private Limited or its affiliates. All rights reserved. Privacy Terms
```

Now execute this in ansible1.

Now use cd .ssh to navigate in our linux environment.

Now use ls command after executing this command it shows:

authorized\_keys known\_hosts

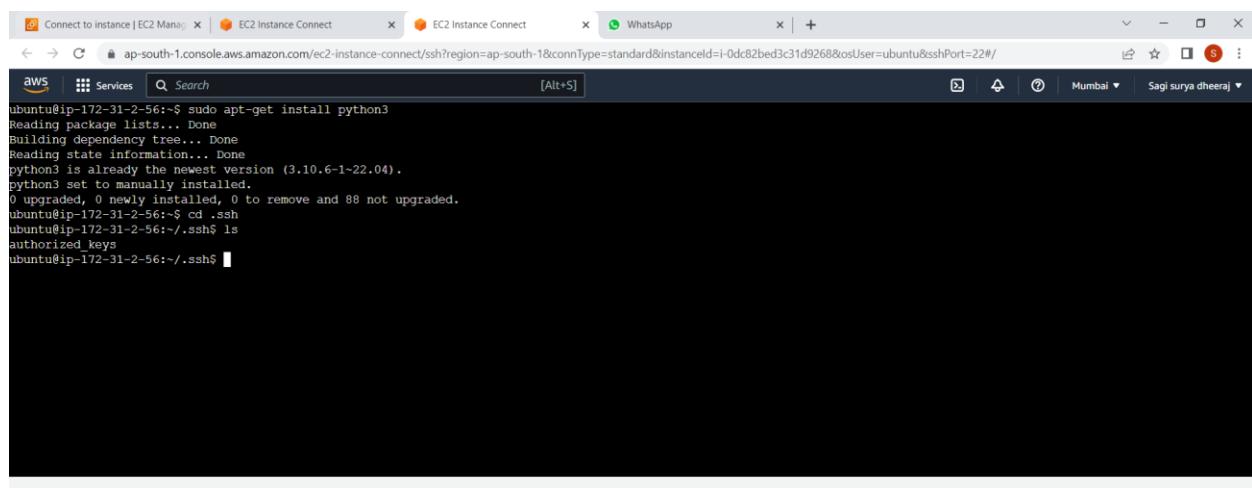


```
ubuntu@ip-172-31-13-147:~$ cd .ssh
ubuntu@ip-172-31-13-147:~/ssh$ ls
authorized_keys known_hosts
ubuntu@ip-172-31-13-147:~/ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAQABAAAQDkuIiRWYV7Byoa5w7XvjoW5DV7yPR4U/QJcYn5z20BWcW+QDGRTIBJYOKyf0auRt1VogIhrdEWqbAFT5uIVfR484EI/XmCJ6UiYdyiW9M7plsnVPtNstquQJ6iD2qt8aOyWdiIKSGhN9mSBKoahakg2uxjttS6/E6/fqMzrw0/qlGdKvmYrKu6aiZqYgM3+Xb0fkxMXdI3VsHh4ie1ZddsIFsYzF3SKXlWxKLg/osQThUz3AnnaCQ7vt2H77D5DR6rRctbAL9yon3PfbVTc5K5Xn1VV+gFYboiaRAueIaJXBFX3wCsnSLERstauWZ2L1TPcxJx7m1Qy5Xdmjb55 ansible
ubuntu@ip-172-31-13-147:~/ssh$
```

i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147

Now use cd .ssh command in ansible2 and execute ls command it shows:

authorized\_keys



```
ubuntu@ip-172-31-2-56:~$ sudo apt-get install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 88 not upgraded.
ubuntu@ip-172-31-2-56:~$ cd .ssh
ubuntu@ip-172-31-2-56:~/ssh$ ls
authorized_keys
ubuntu@ip-172-31-2-56:~/ssh$
```

i-0dc82bed3c31d9268 (ansible2)  
PublicIPs: 13.127.8.232 PrivateIPs: 172.31.2.56

Now execute the following command in ansible1 it will generate a keypair.

The screenshot shows a terminal window titled 'aws' with the following command and output:

```
ubuntu@ip-172-31-13-147:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Ch58/n3KL+amsM7SeTVIATZKwRlTp6ip3eUOzFVxqiw ubuntu@ip-172-31-13-147
The key's randomart image is:
+---[RSA 3072]----+
| ..oo = . |
| +..* .. |
| + oo . |
| .+o o |
| oE= S o |
| +o B.. o |
| o .+++. . |
| . .o=+o.o+ . |
| o=o===. |
+---[SHA256]----+
ubuntu@ip-172-31-13-147:~/.ssh$
```

At the bottom of the terminal, the session details are shown:

i-0e5e883440057c8b6 (ansible1)  
Public IPs: 35.154.17.101 Private IPs: 172.31.13.147

After that now execute ls command it shows:-

authorized\_keys id\_rsa id\_rsa.pub known\_hosts

Now execute cat id\_rsa.pub command it will generate a code copy that code.

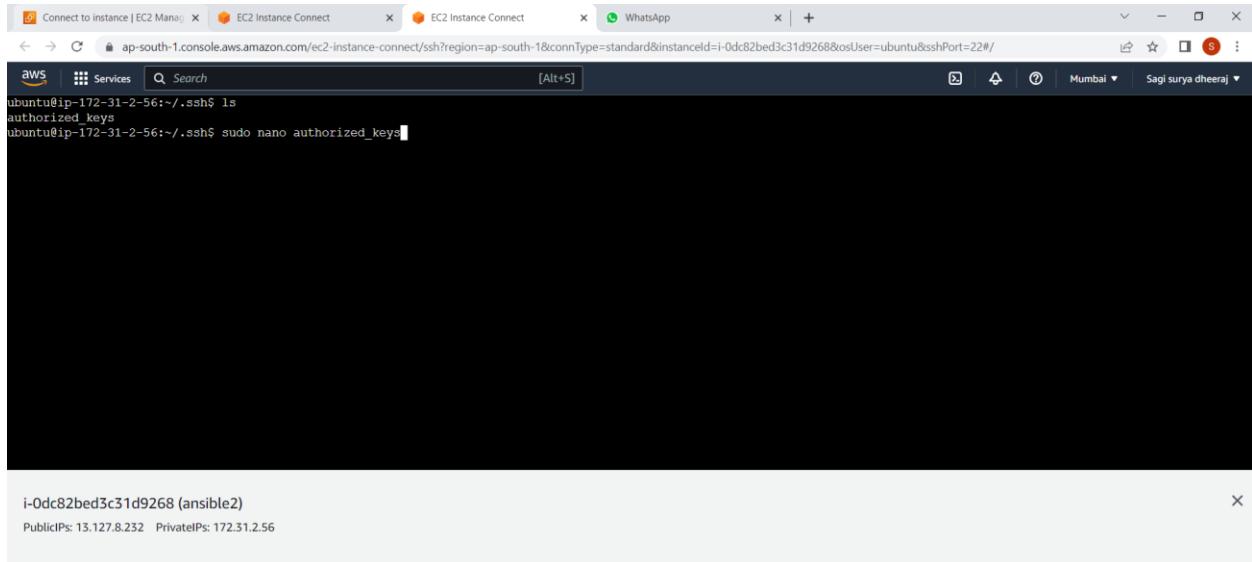
The screenshot shows a terminal window titled 'aws' with the following command and output:

```
ubuntu@ip-172-31-13-147:~/.ssh$ ls
authorized_keys id_rsa id_rsa.pub known_hosts
ubuntu@ip-172-31-13-147:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCv5DW1McBzCM17NBz4rHdEl16PqwGLKot5PemQpOEKGX6cy//WYGDVGK9gQUYDfYGFvzDvCaLrRWJ/W9L8RKJ0qTJvkPmM0j0ihB3qmAeU3zKO+479gCoeoGAUczlBRjjQS084f9w4d1RvJR/VrrtoW0ESIAoAEw2p0fpDtVNzBNM/Z1VD8M01Ah3uNE4f06VAd2rNT0YXKUzRk8i5VSw5tbaETw2IZ8D/Kq9Ry2TEWZKa67vqpOhSN6T0yA/41atfrn7qfcYyzb4FuxJHHTuIfMo6o6TtrKUokf6PT42VdR+cZELPx631rtjva1ohKxrVDGygxn6xBa08fz2n08ETM8VIdz+o396+VlupKQx0PcPgAmeJYsj3CcCXJluCDtrNyvxS1TWbqr9tM0E5/gutf3/DU94H/Sdg5vmta0/t9acRp+lgzd9UypSwFttK9puAzdkIdlMke6aojE0/veXFPqFANpb3KGwbllagshjM+XMVzwmTD4ldcYEN74E= ubuntu@ip-172-31-13-147:~/.ssh$
```

At the bottom of the terminal, the session details are shown:

i-0e5e883440057c8b6 (ansible1)  
Public IPs: 35.154.17.101 Private IPs: 172.31.13.147

Now execute the following code in ansible2.

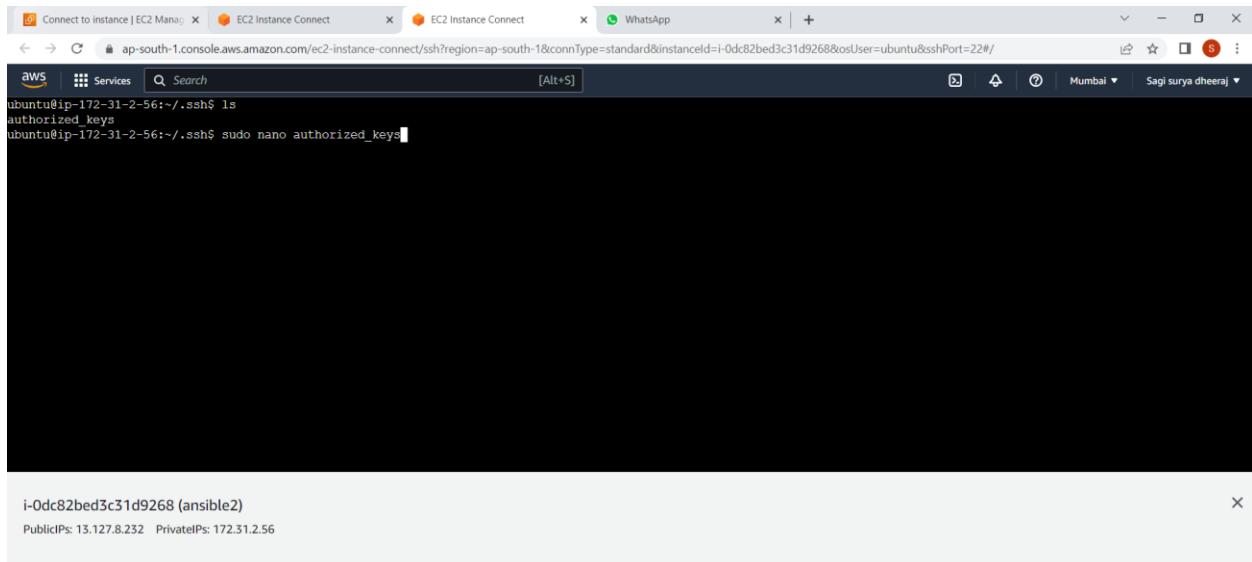


A screenshot of a terminal window titled "aws Services". The terminal shows the command "sudo nano authorized\_keys" being typed. Below the terminal, a status bar displays "i-0dc82bed3c31d9268 (ansible2)" and "PublicIPs: 13.127.8.232 PrivateIPs: 172.31.2.56".

```
ubuntu@ip-172-31-2-56:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-2-56:~/.ssh$ sudo nano authorized_keys
```

i-0dc82bed3c31d9268 (ansible2)  
PublicIPs: 13.127.8.232 PrivateIPs: 172.31.2.56

Paste the copied code in after executing sudo nano authorized\_keys.

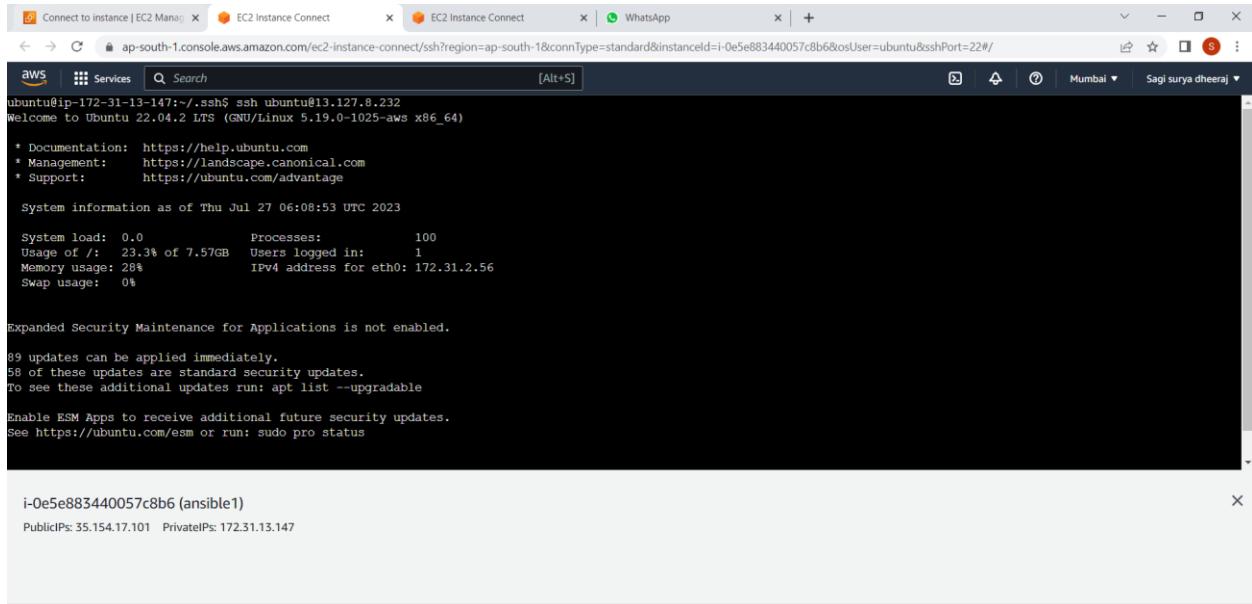


A screenshot of a terminal window titled "aws Services". The terminal shows the command "sudo nano authorized\_keys" being typed. Below the terminal, a status bar displays "i-0dc82bed3c31d9268 (ansible2)" and "PublicIPs: 13.127.8.232 PrivateIPs: 172.31.2.56".

```
ubuntu@ip-172-31-2-56:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-2-56:~/.ssh$ sudo nano authorized_keys
```

i-0dc82bed3c31d9268 (ansible2)  
PublicIPs: 13.127.8.232 PrivateIPs: 172.31.2.56

Now come to ansible1 and once again execute the ssh ubuntu@ip(ansible public ip address) command.



```
ubuntu@ip-172-31-13-147:~/.ssh$ ssh ubuntu@13.127.8.232
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Thu Jul 27 06:08:53 UTC 2023

System load: 0.0          Processes:      100
Usage of /: 23.3% of 7.57GB  Users logged in:     1
Memory usage: 28%          IPv4 address for eth0: 172.31.2.56
Swap usage:  0%

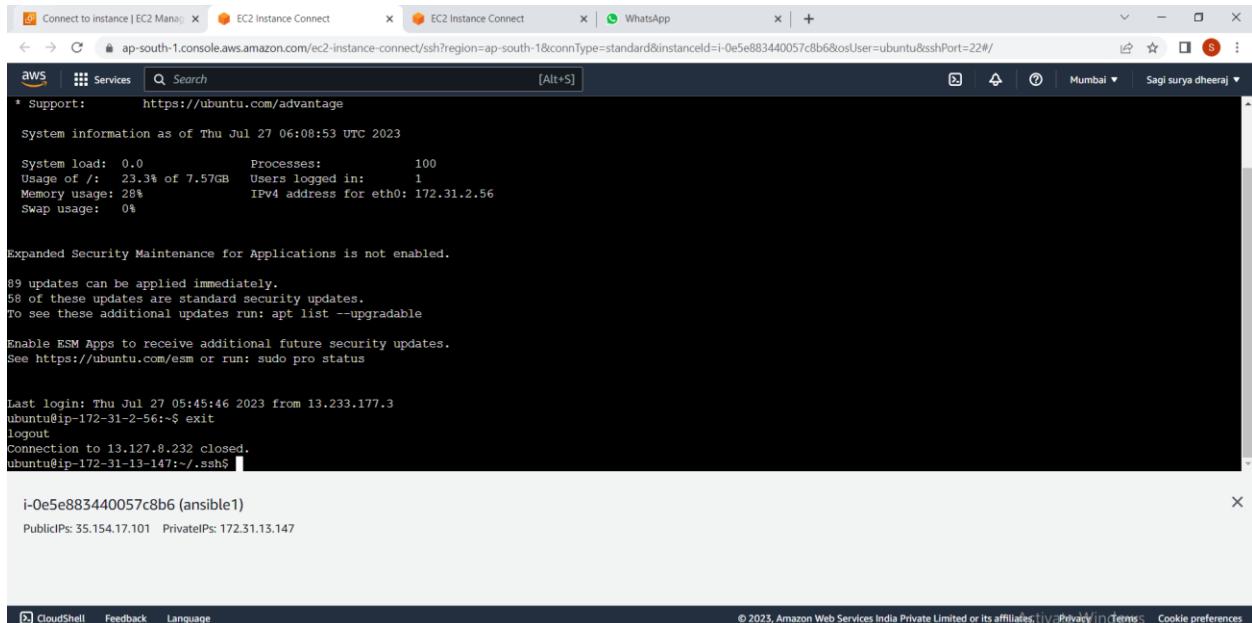
Expanded Security Maintenance for Applications is not enabled.

89 updates can be applied immediately.
58 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147

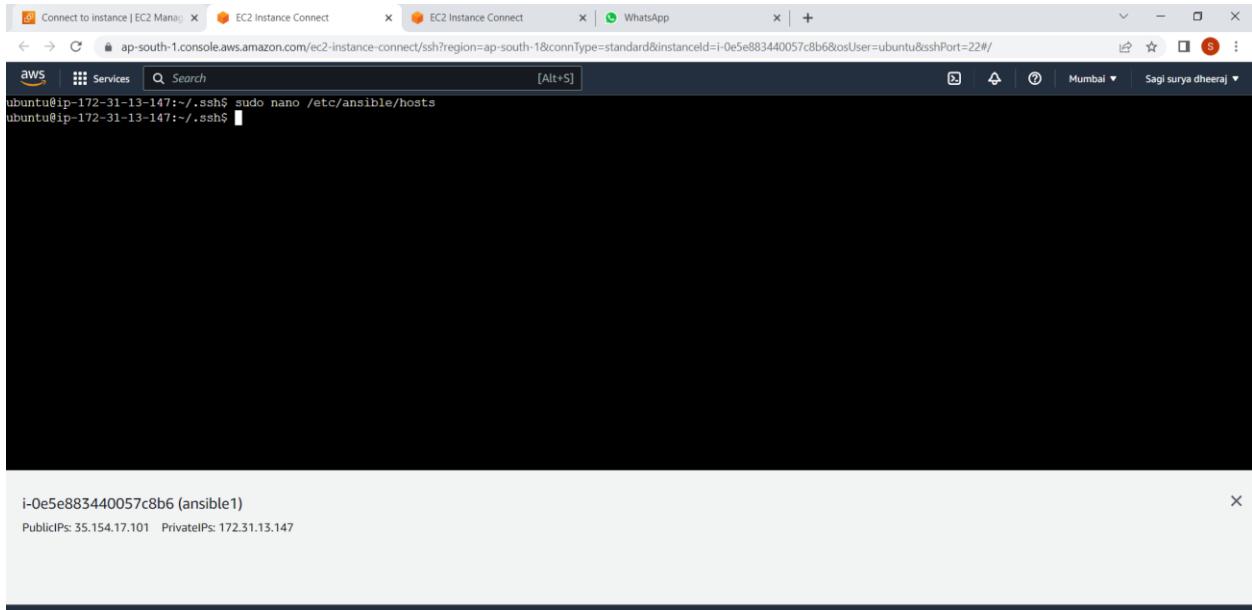
You can clearly see that the ansible1 got logged into ansible2.



```
ubuntu@ip-172-31-2-56:~$ exit
logout
Connection to 13.127.8.232 closed.
ubuntu@ip-172-31-13-147:~/.ssh$
```

i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147

Now execute the following commands in ansible1.



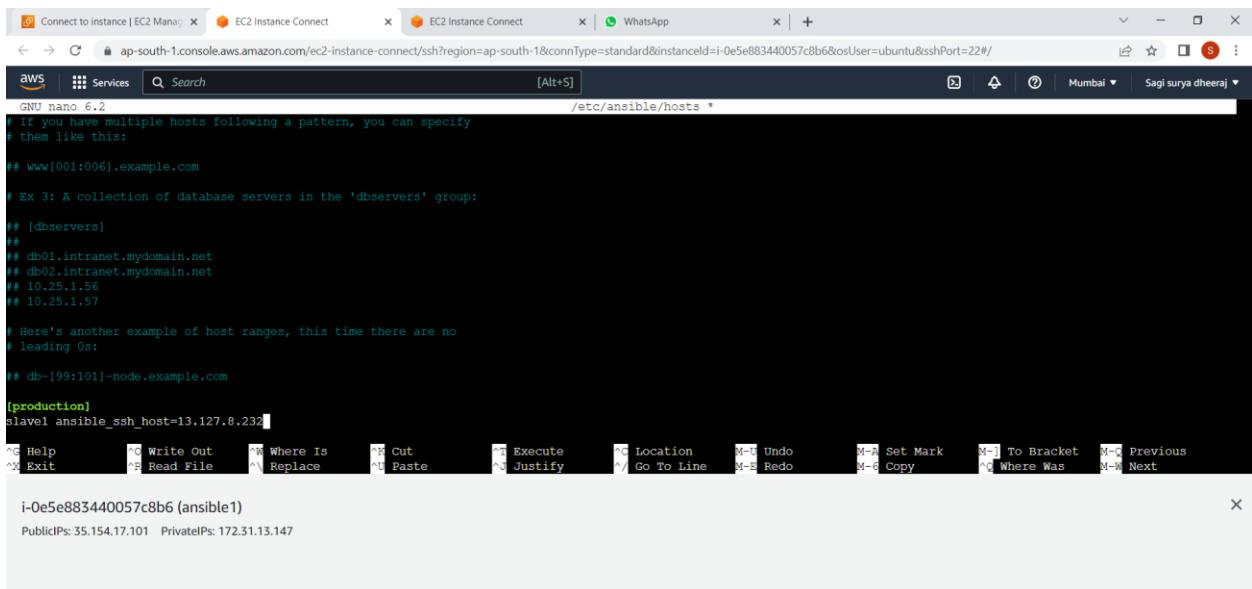
```
aws Services Search [Alt+S]
ubuntu@ip-172-31-13-147:~/.ssh$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-13-147:~/.ssh$
```

i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147

After running the command its shows like this

Here [production] is called as group which is created now and also slave which is here slave1.

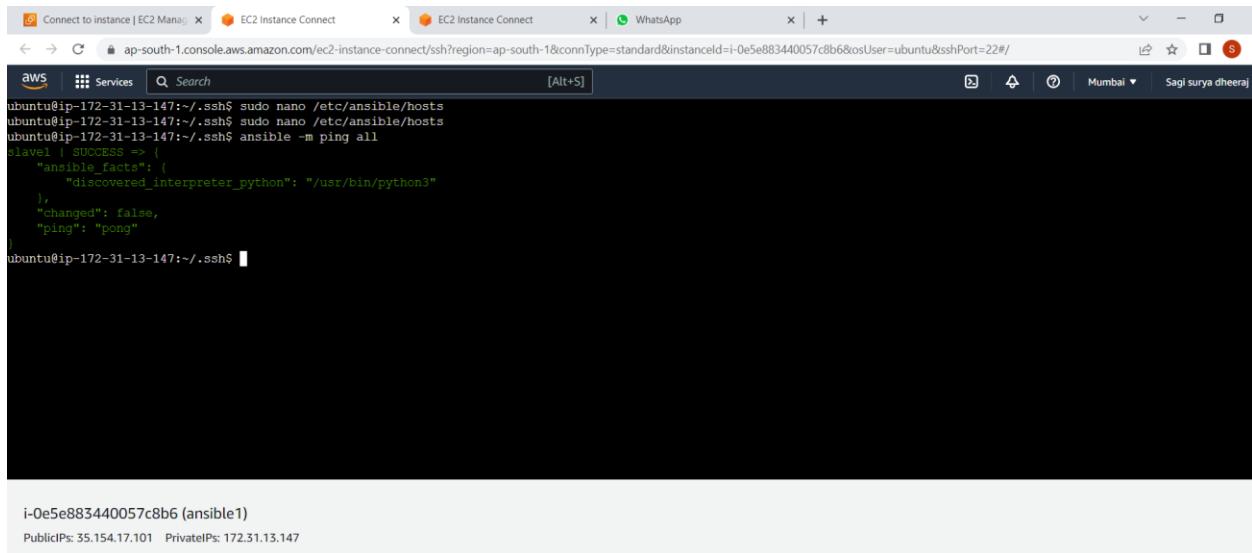
write the following command:-slave1 ansible\_ssh\_host=(ansible2 public ip address).



```
GNU nano 6.2
/etc/ansible/hosts *
# 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:
## [dbservers]
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57
# Here's another example of host ranges, this time there are no
# leading 0s:
## db-[99:101]-node.example.com
[production]
slave1 ansible_ssh_host=13.127.8.232
```

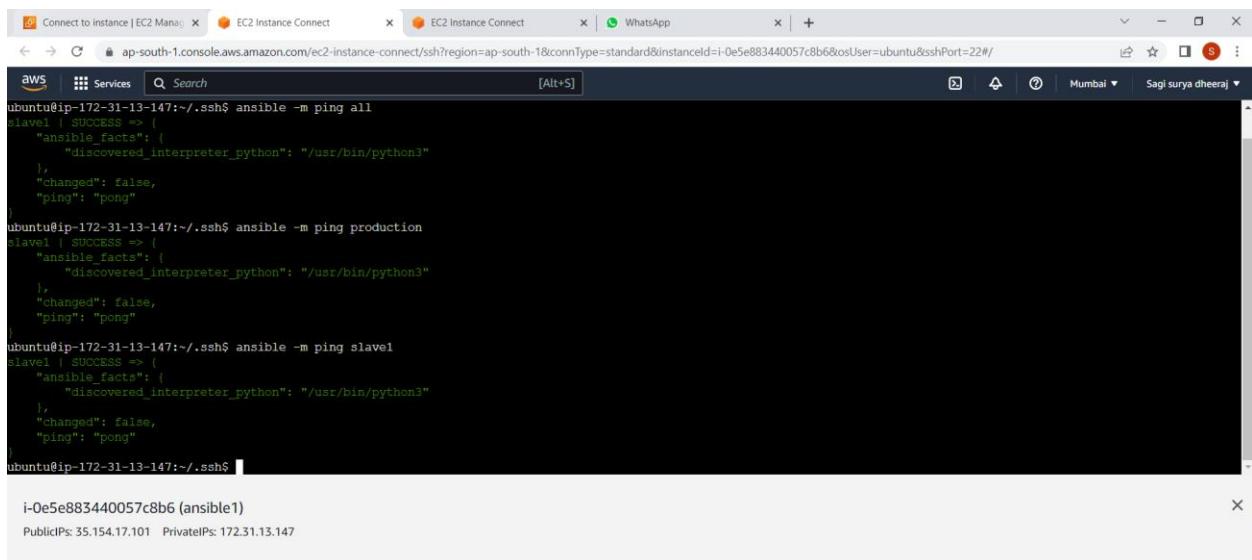
i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147

Execute the following commands in ansible1.



```
aws EC2 Instance Connect WhatsApp [Alt+S] Mumbai Sagi surya dheeraj
ubuntu@ip-172-31-13-147:~/.ssh$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-13-147:~/.ssh$ sudo nano /etc/ansible/hosts
ubuntu@ip-172-31-13-147:~/.ssh$ ansible -m ping all
slave1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-13-147:~/.ssh$
```

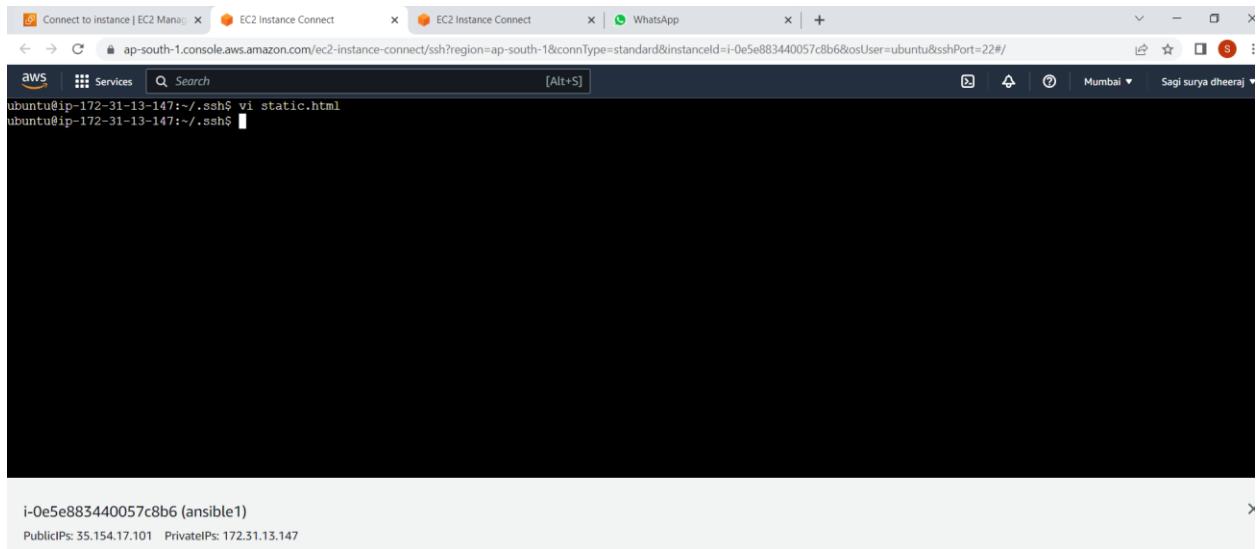
i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147



```
aws EC2 Instance Connect WhatsApp [Alt+S] Mumbai Sagi surya dheeraj
ubuntu@ip-172-31-13-147:~/.ssh$ ansible -m ping all
slave1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-13-147:~/.ssh$ ansible -m ping production
slave1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-13-147:~/.ssh$ ansible -m ping slave1
slave1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-13-147:~/.ssh$
```

i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147

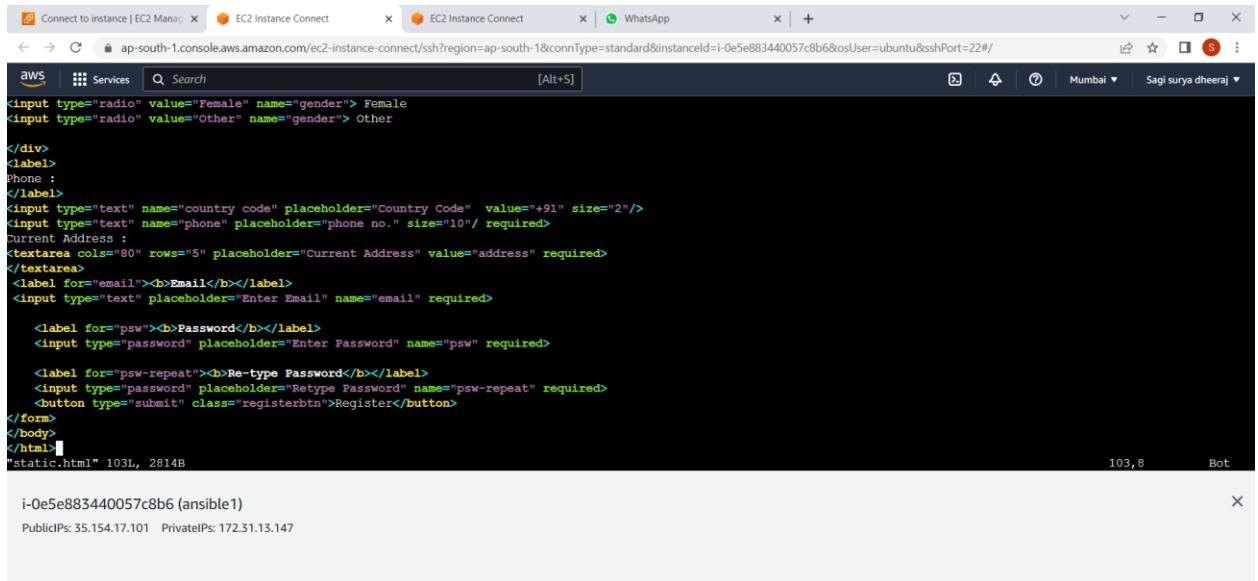
Now in ansible1 create a file.



```
ubuntu@ip-172-31-13-147:~/ssh$ vi static.html
```

The screenshot shows a terminal window titled "EC2 Instance Connect" with the command "vi static.html" entered. The window is part of a browser interface for managing AWS instances. At the bottom, it displays the instance ID "i-0e5e883440057c8b6 (ansible1)" and its public and private IP addresses.

Write the html code.



```
<input type="radio" value="Female" name="gender"> Female
<input type="radio" value="Other" name="gender"> Other

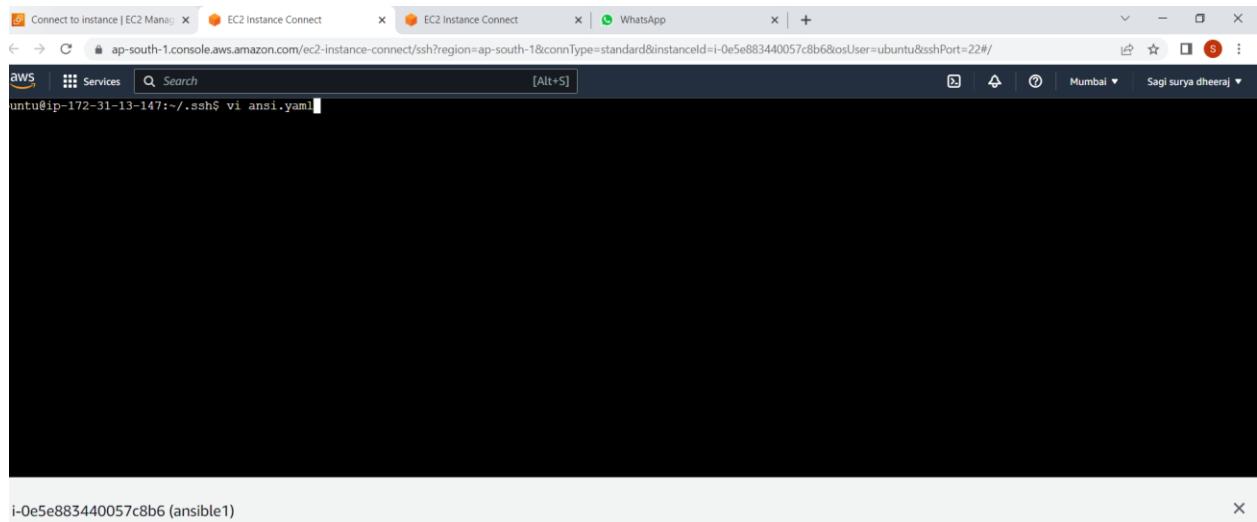
</div>
<label>
Phone :
</label>
<input type="text" name="country code" placeholder="Country Code" value="+91" size="2"/>
<input type="text" name="phone" placeholder="phone no." size="10" required>
Current Address :
<textarea cols="80" rows="5" placeholder="Current Address" value="address" required>
</textarea>
<label for="email"><b>Email</b></label>
<input type="text" placeholder="Enter Email" name="email" required>

<label for="psw"><b>Password</b></label>
<input type="password" placeholder="Enter Password" name="psw" required>

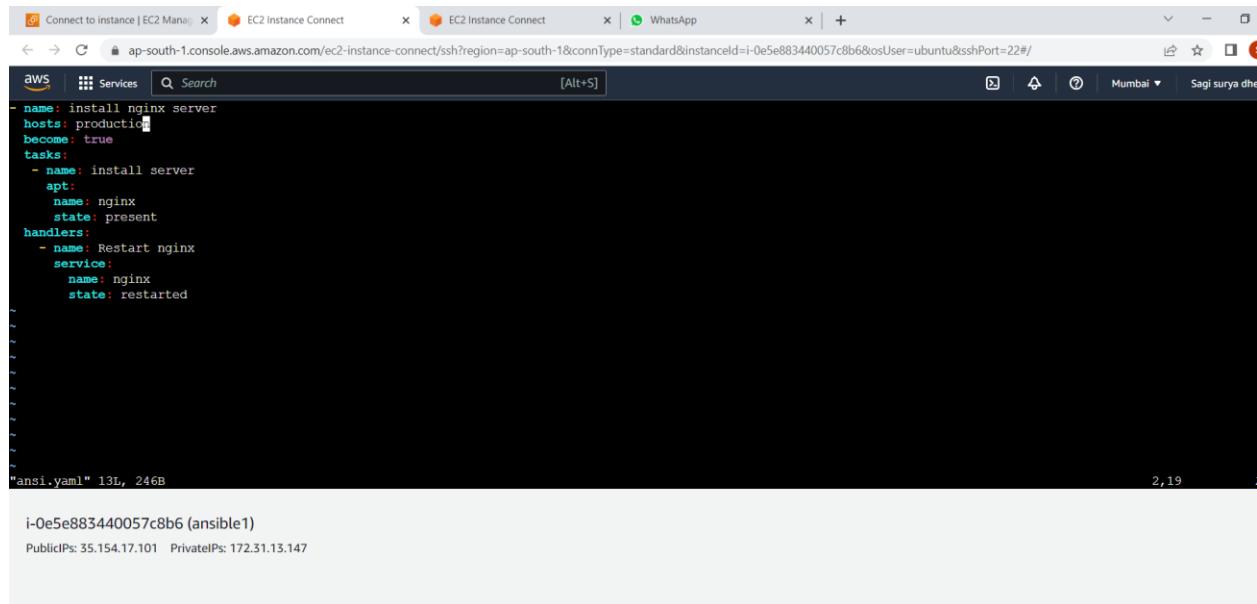
<label for="psw-repeat"><b>Re-type Password</b></label>
<input type="password" placeholder="Retype Password" name="psw-repeat" required>
<button type="submit" class="registerbtn">Register</button>
```

The screenshot shows a terminal window displaying the content of a static HTML file named "static.html". The file contains a registration form with fields for gender (radio buttons), country code (input), phone number (input), current address (textarea), email (input), password (input), and re-type password (input). A submit button is also present. The terminal shows the file has 103 lines and 2814 bytes.

Now execute the following yaml code,in place of hosts give group name i.e.production.



```
ubuntu@ip-172-31-13-147:~/ssh$ vi ansi.yaml
```



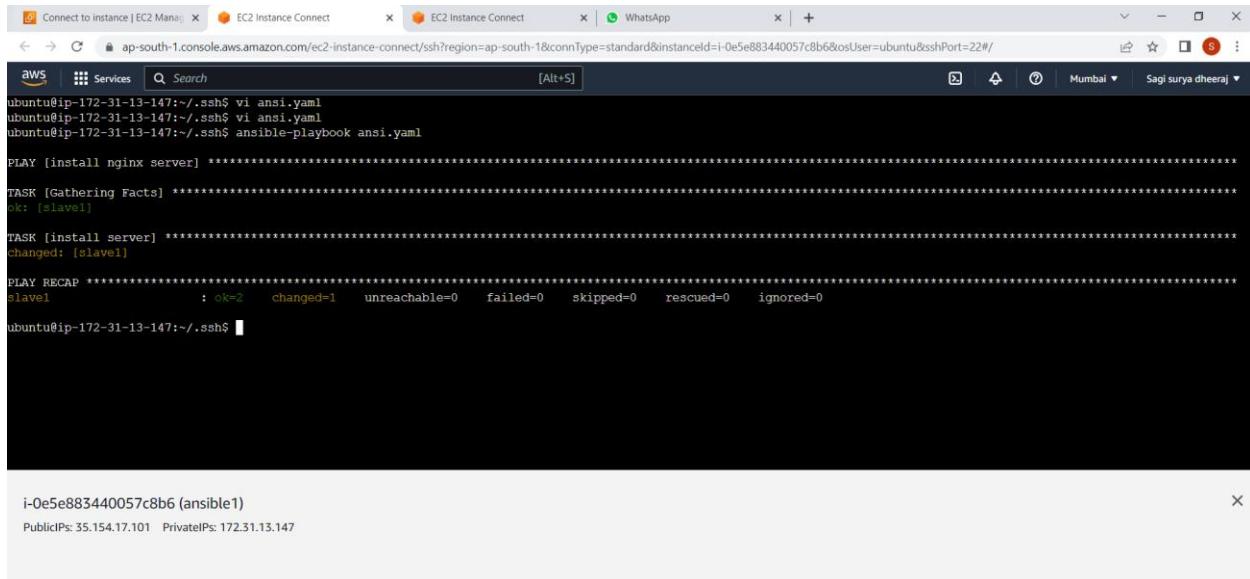
```
name: install nginx server
hosts: production
become: true
tasks:
- name: install server
  apt:
    name: nginx
    state: present
handlers:
- name: Restart nginx
  service:
    name: nginx
    state: restarted
```

"ansi.yaml" 13L, 246B 2,19

Now execute the following command.

### *Ansible-playbook:-*

The primary way of using ansible to automate tasks is through “playbooks”, an ‘ansible-playbook’ is the command used to execute the playbooks. Playbooks are written in YAML format.



```
ubuntu@ip-172-31-13-147:~/.ssh$ vi ansi.yaml
ubuntu@ip-172-31-13-147:~/.ssh$ vi ansi.yaml
ubuntu@ip-172-31-13-147:~/.ssh$ ansible-playbook ansi.yaml

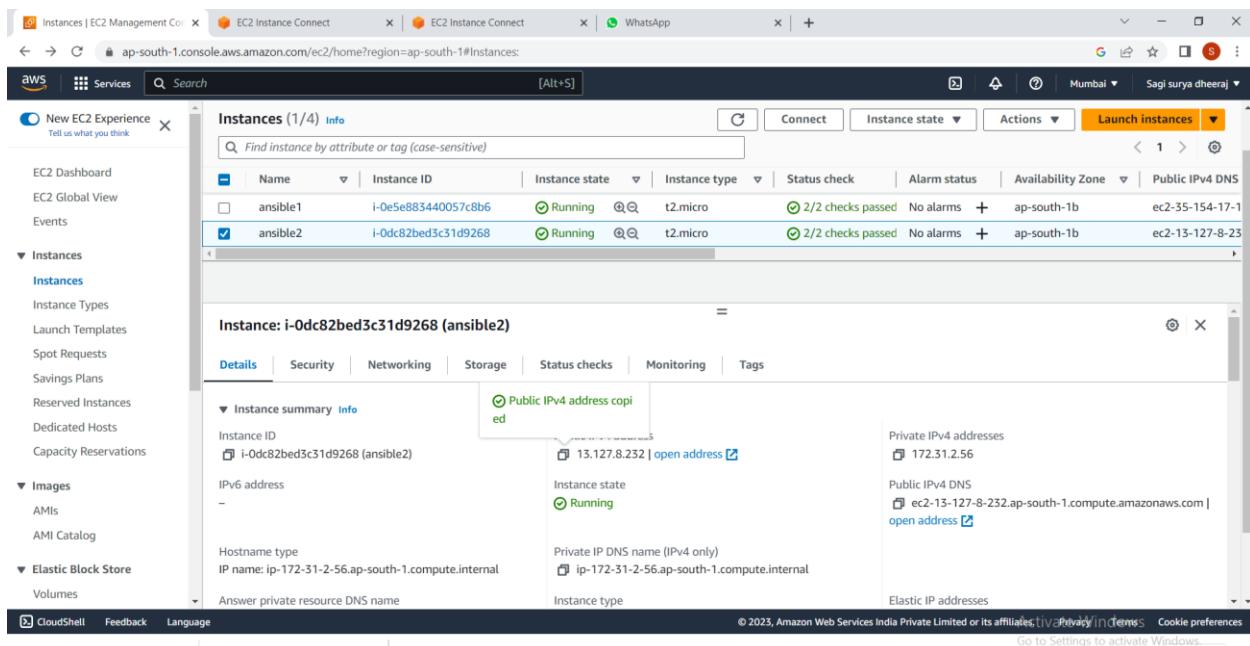
PLAY [install nginx server] ****
TASK [Gathering Facts] ****
ok: [slave1]
TASK [install server] ****
changed: [slave1]

PLAY RECAP ****
slave1 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-13-147:~/.ssh$
```

i-0e5e883440057c8b6 (ansible1)  
PublicIPs: 35.154.17.101 PrivateIPs: 172.31.13.147

Now here copy the public ip address of ansible2.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
ansible1	i-0e5e883440057c8b6	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-35-154-17-1
ansible2	i-0dc82bed3c31d9268	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-127-8-23

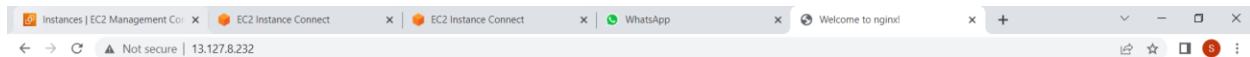
Instance: i-0dc82bed3c31d9268 (ansible2)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info  
Instance ID: i-0dc82bed3c31d9268 (ansible2)  
IPv6 address: -  
Hostname type: IP name: ip-172-31-2-56.ap-south-1.compute.internal  
Answer private resource DNS name: -

Public IPv4 address copied  
13.127.8.232 | open address  
Private IPv4 addresses  
172.31.2.56  
Public IPv4 DNS  
ec2-13-127-8-23.ap-south-1.compute.amazonaws.com | open address  
Elastic IP addresses

Paste it in the chrome.



## Welcome to nginx!

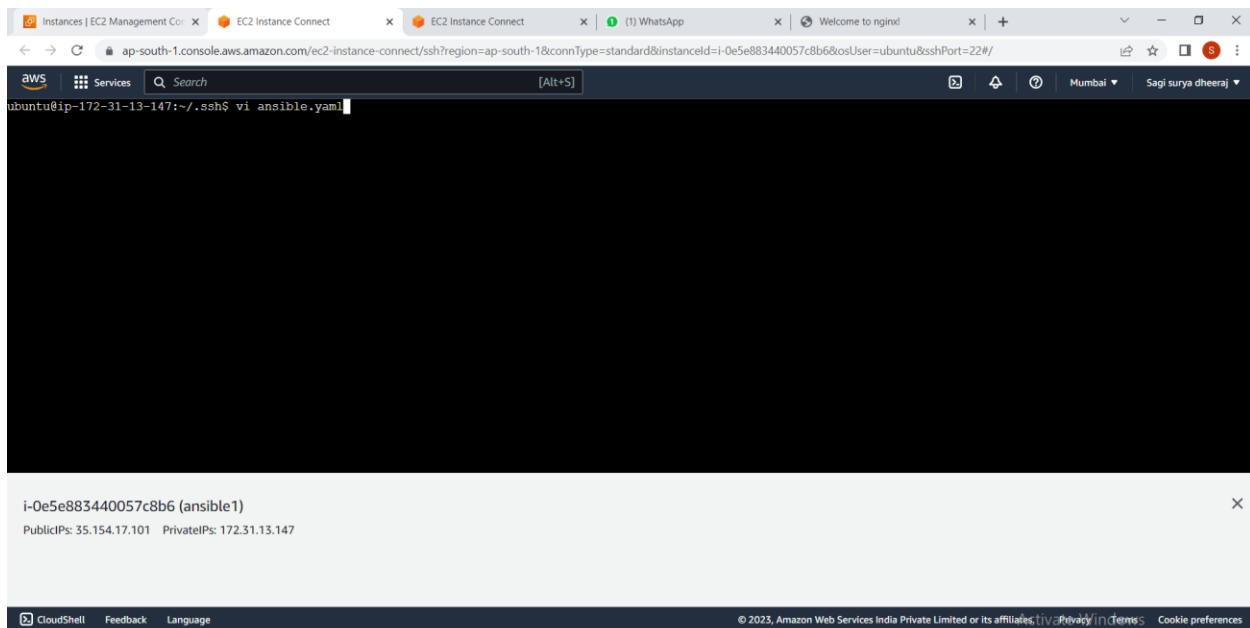
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

Activate Windows

Now create another file i.e.vi filename.yaml



Here src:give the file path of html file we have created before.ie./home/ubuntu/.ssh/static.html

The screenshot shows a terminal window within the AWS CloudWatch interface. The terminal is executing an Ansible playbook named 'ansible.yaml'. The playbook defines a host static web application with a task to copy an HTML file from '/home/ubuntu/.ssh/static.html' to '/var/www/html/index.html' and to restart the Nginx service.

```
- name: host static web application
  hosts: production
  become: true
  tasks:
    - name: copy html file to remote server
      copy:
        src: /home/ubuntu/.ssh/static.html
        dest: /var/www/html/index.html
      notify:
        - Restart Nginx
    handlers:
      - name: Restart Nginx
        service:
          name: nginx
          state: restarted
```

At the bottom of the terminal, it shows the instance details: i-0e5e883440057c8b6 (ansible1) with PublicIPs: 35.154.17.101 and PrivateIPs: 172.31.13.147.

The screenshot shows a terminal window within the AWS CloudWatch interface. The terminal is executing an Ansible playbook named 'ansible.yaml'. The output shows the playbook running on a single host ('host static web application') and performing a task to copy an HTML file to the remote server and restart the Nginx service. The command 'ansible-playbook ansible.yaml' is run at the prompt.

```
ubuntu@ip-172-31-13-147:~/ssh$ vi ansible.yaml
ubuntu@ip-172-31-13-147:~/ssh$ ansible-playbook ansible.yaml

PLAY [host static web application] ****
TASK [Gathering Facts] ****
ok: [slave1]

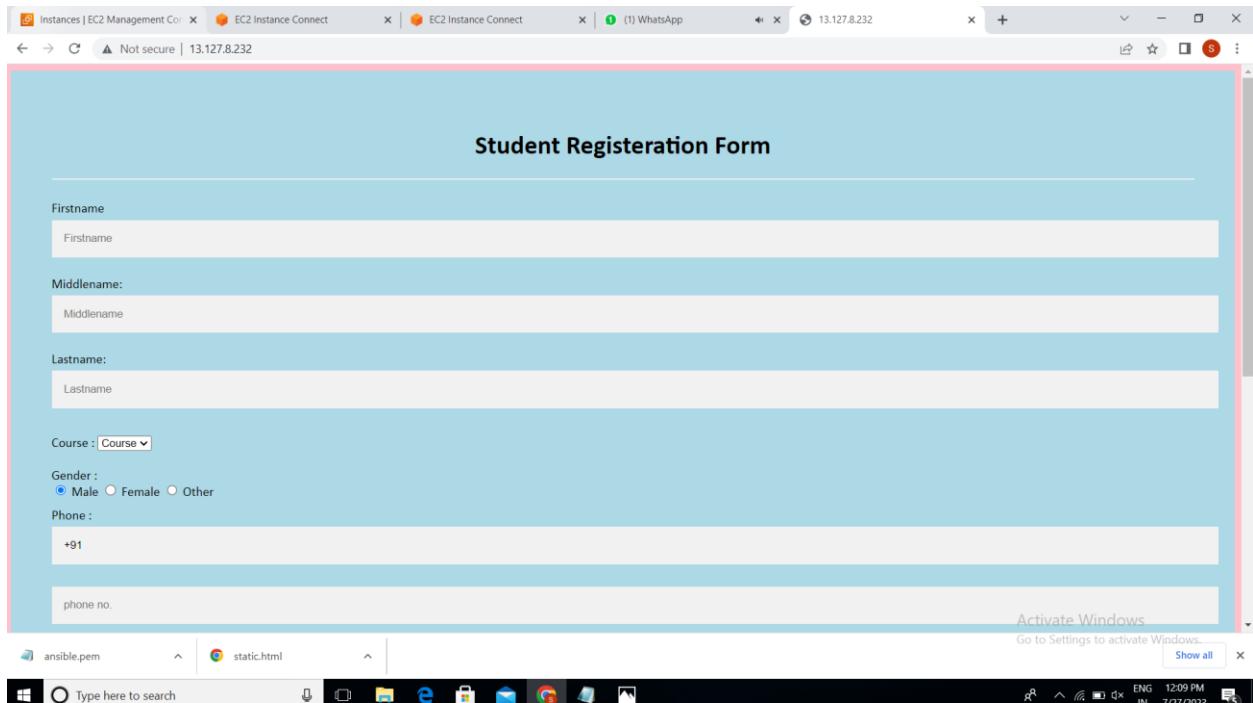
TASK [copy html file to remote server] ****
changed: [slave1]

RUNNING HANDLER [Restart Nginx] ****
changed: [slave1]

PLAY RECAP ****
slave1 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@ip-172-31-13-147:~/ssh$
```

At the bottom of the terminal, it shows the instance details: i-0e5e883440057c8b6 (ansible1) with PublicIPs: 35.154.17.101 and PrivateIPs: 172.31.13.147.

Now once again copy the public ip address of ansible2 and paste it, now our web page is open.



## OUTCOME:

**Improved efficiency:** Ansible can automate repetitive tasks, such as software installation, configuration management, and infrastructure provisioning. This can save time and reduce the risk of errors that can occur with manual processes. As a result, IT teams can be more efficient and productive, focusing on more strategic work that adds value to the organization.

**Increased consistency:** Automation with Ansible ensures that IT operations are standardized and consistent, reducing the risk of errors and security vulnerabilities. Ansible allows you to define playbooks and roles that can be used across your organization, ensuring that tasks are executed in a consistent manner. This can help to improve the reliability and stability of your IT environment.

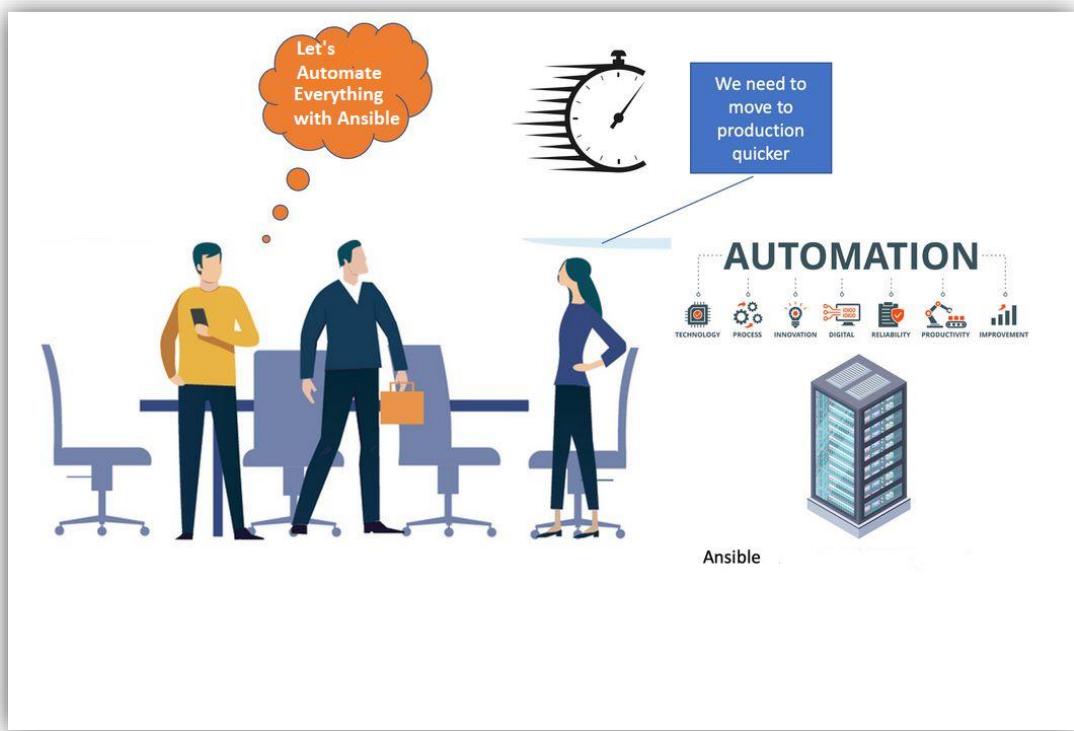
**Enhanced scalability:** Ansible is highly scalable and can automate tasks across large numbers of servers and devices. This makes it a valuable tool for organizations that need to manage complex IT environments. Ansible allows you to define groups of hosts and execute tasks across those groups, allowing you to easily scale your automation efforts as your environment grows.

**Reduced costs:** By automating routine tasks, Ansible can help organizations save time and reduce labour costs associated with manual processes. Ansible can also help organizations to avoid costly downtime by automating tasks such as system patching and updates.

**Improved collaboration:** Ansible provides a platform for collaboration between IT teams, allowing for the sharing of playbooks and roles that can be used across the organization. This can help to improve communication and coordination between teams, leading to a more efficient and effective IT operation.

**Enhanced security:** Ansible can help organizations maintain a strong security posture by automating tasks such as patch management, vulnerability scanning, and compliance auditing. Ansible can also help to enforce security policies across your environment by automating the configuration of firewalls, security groups, and other security-related tasks.

In summary, automating applications using Ansible can lead to significant improvements in efficiency, consistency, scalability, cost savings, collaboration, and security. These outcomes can help organizations to streamline their IT operations, improve productivity, and reduce the risk of downtime and security breaches.



## CONCLUSION:

In conclusion, automating applications using Ansible can provide numerous benefits for organizations looking to improve their IT operations. Ansible is a powerful open-source automation tool that can be used to automate a wide range of tasks, from software installation and configuration management to infrastructure provisioning and deployment.

By automating routine tasks, organizations can improve efficiency, consistency, and scalability, while reducing the costs associated with manual processes. Ansible can also help enhance collaboration between IT teams, allowing for the sharing of playbooks and roles that can be used across the organization.

Additionally, Ansible can help organizations maintain a strong security posture by automating tasks such as patch management, vulnerability scanning, and compliance auditing. By leveraging Ansible's capabilities, organizations can achieve greater efficiency, cost savings, and security, while freeing up IT teams to focus on more strategic work.