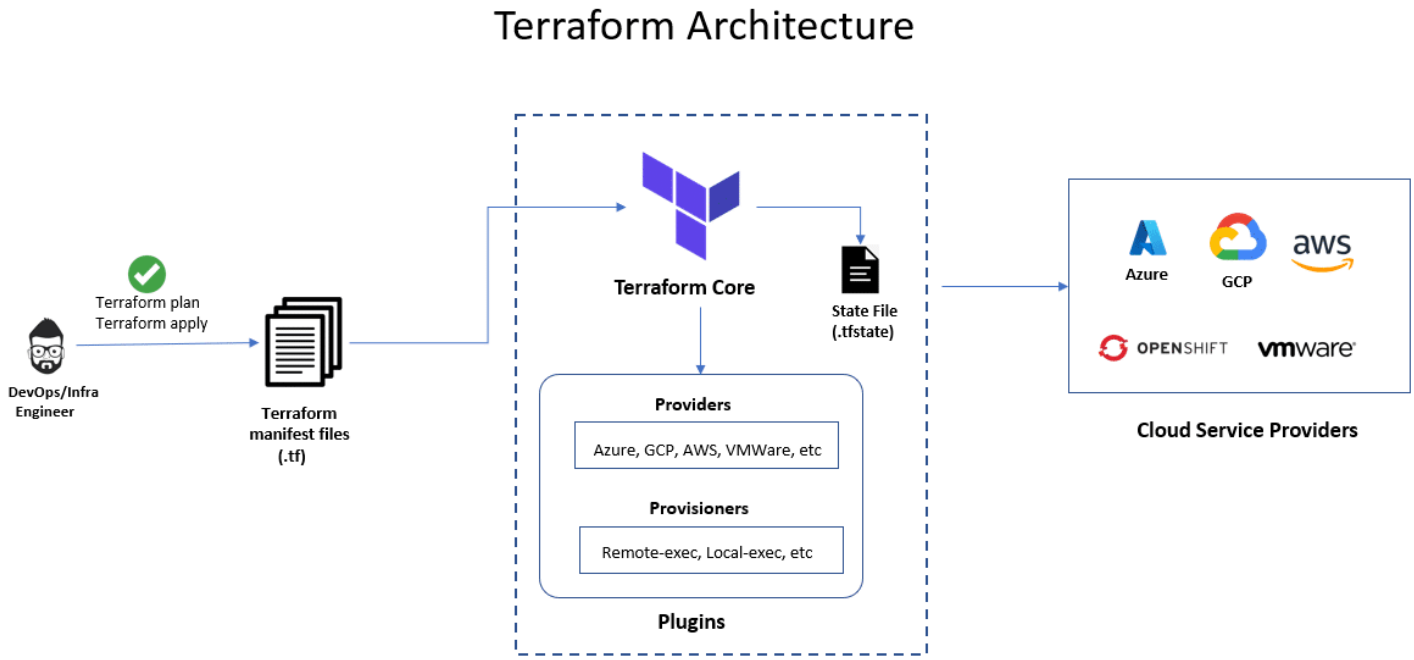
1. **Create an Ec2 instance with 8080 port number using terraform.**

Terraform: Terraform is an open-source infrastructure as code (IaC) tool developed by HashiCorp. It enables users to define and provision infrastructure resources in a declarative configuration language, allowing them to describe the desired state of their infrastructure and then automatically create and manage those resources.

Terraform is widely used in DevOps and cloud computing environments to automate the provisioning and management of infrastructure, making it easier to scale and maintain complex systems.

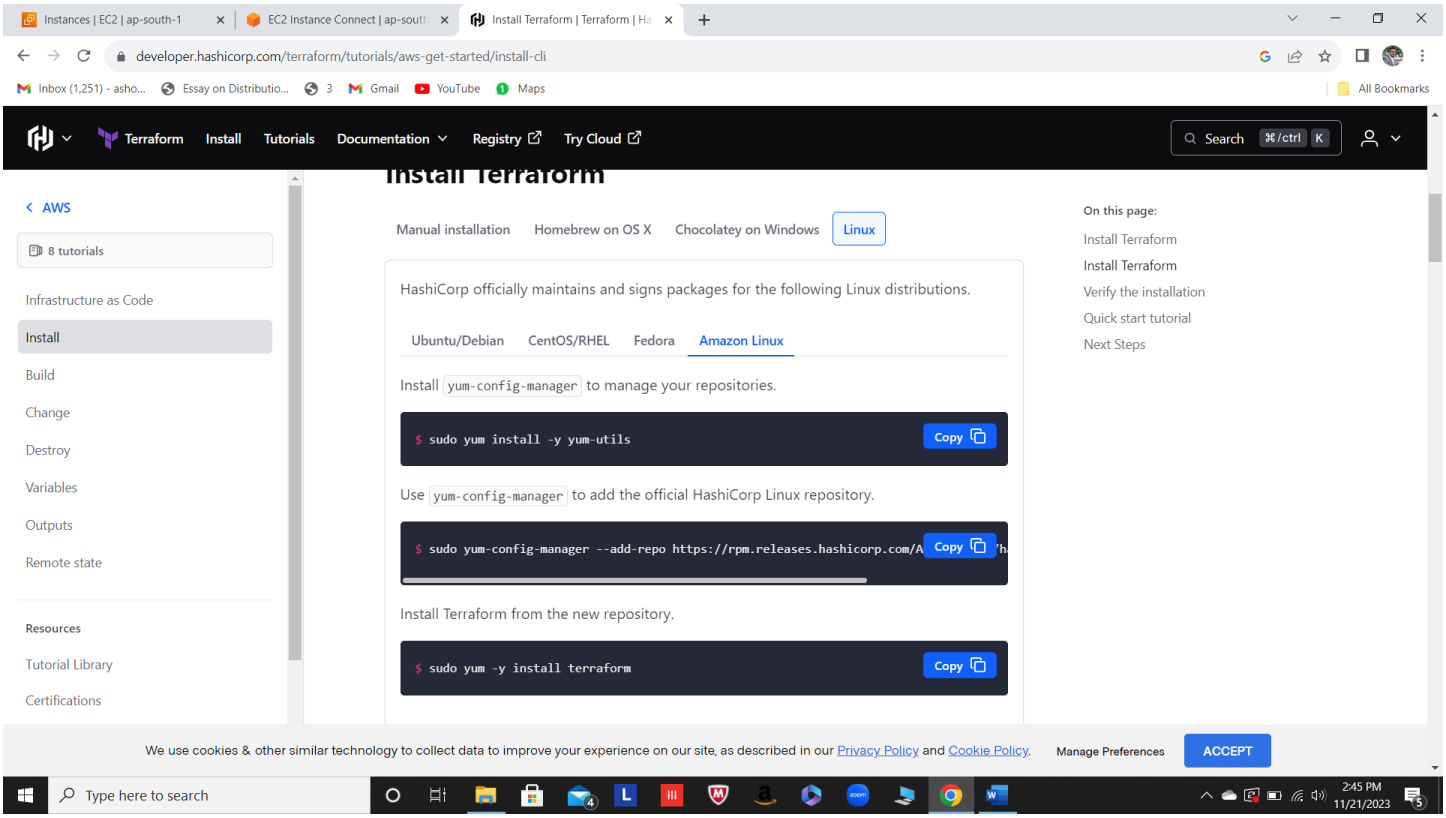


- Install yum-config-manager to manage your repositories.
  sudo yum install -y yum-utils

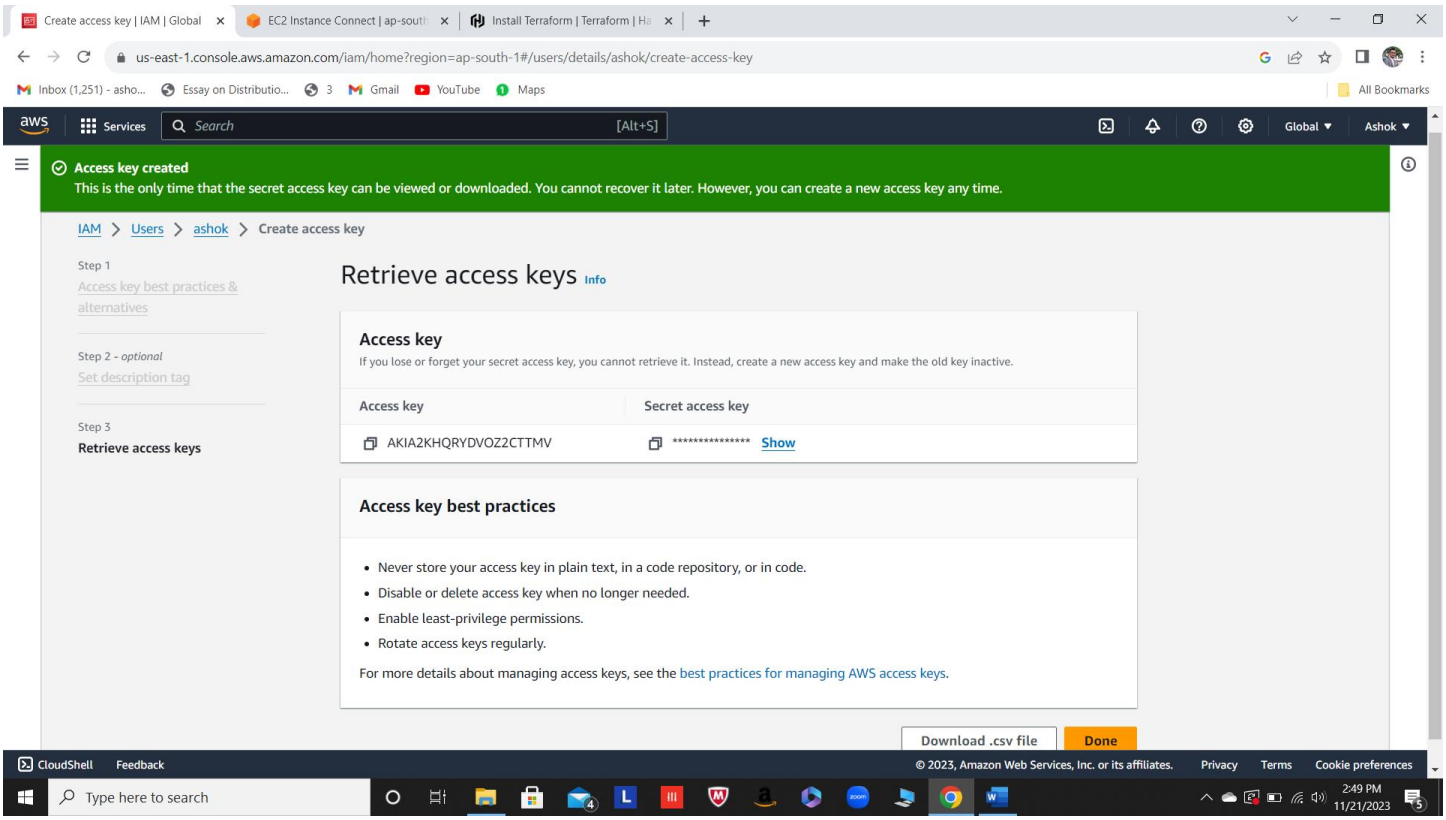
- Use yum-config-manager to add the official HashiCorp Linux repository.
  sudo yum-config-manager --add-repo
  https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

- Install Terraform from the new repository.
  sudo yum -y install terraform

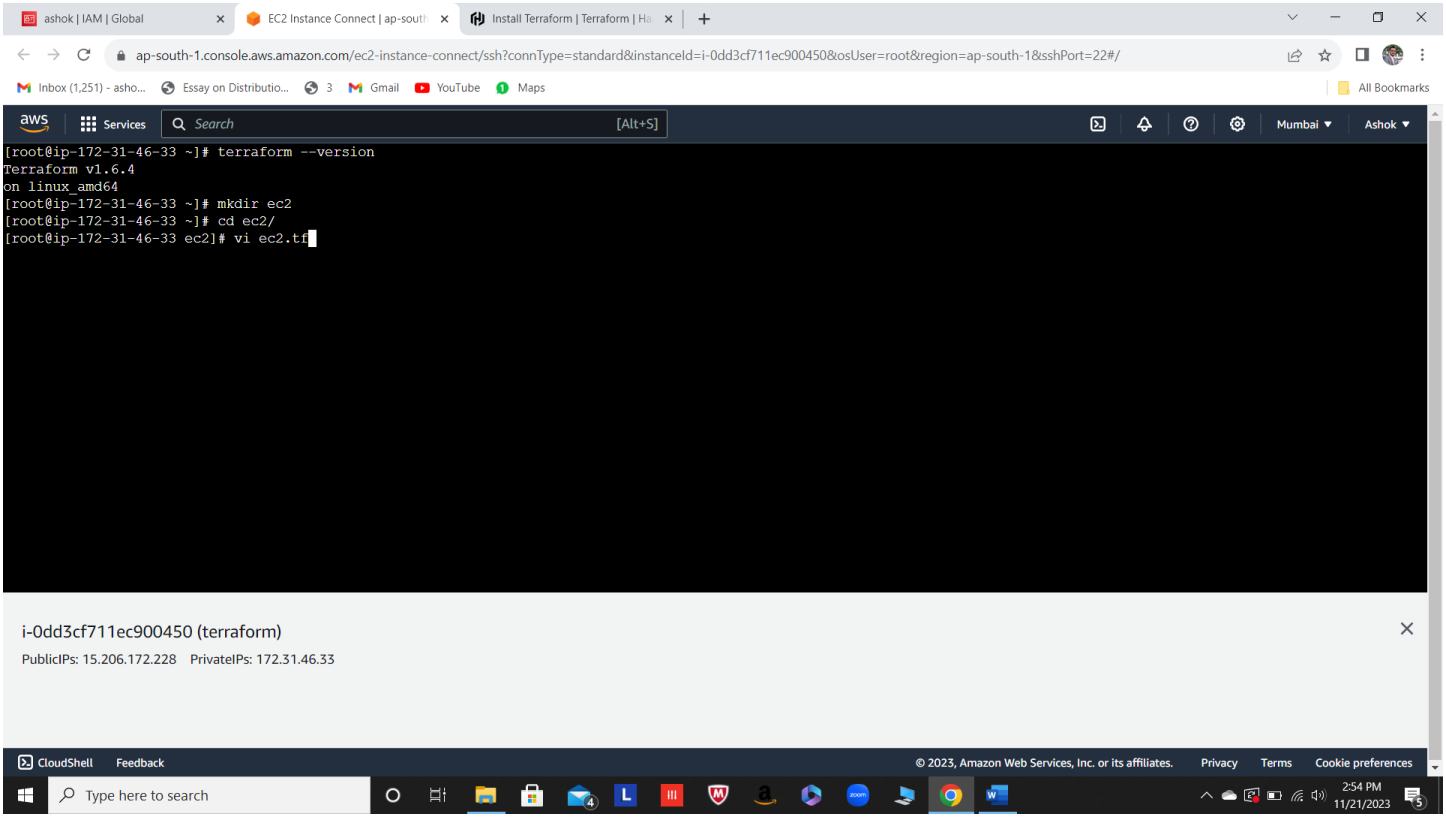
- Verify the installation
  Terraform --version


Instances | EC2 | ap-south-1
EC2 Instance Connect | ap-south
Install Terraform | Terraform | Ha
developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli
Terraform
Install
Tutorials
Documentation
Registry
Try Cloud
Search ⌘/ctrl K
AWS
8 tutorials
Infrastructure as Code
Install
Build
Change
Destroy
Variables
Outputs
Remote state
Resources
Tutorial Library
Certifications
Install Terraform
Manual installation
Homebrew on OS X
Chocolatey on Windows
Linux
HashiCorp officially maintains and signs packages for the following Linux distributions.
Ubuntu/Debian
CentOS/RHEL
Fedora
Amazon Linux
Install yum-config-manager to manage your repositories.
$ sudo yum install -y yum-utils
Copy
Use yum-config-manager to add the official HashiCorp Linux repository.
$ sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/A
Copy
Install Terraform from the new repository.
$ sudo yum -y install terraform
Copy
On this page:
Install Terraform
Install Terraform
Verify the installation
Quick start tutorial
Next Steps
We use cookies & other similar technology to collect data to improve your experience on our site, as described in our Privacy Policy and Cookie Policy.
Manage Preferences
ACCEPT
Type here to search
2:45 PM
11/21/2023

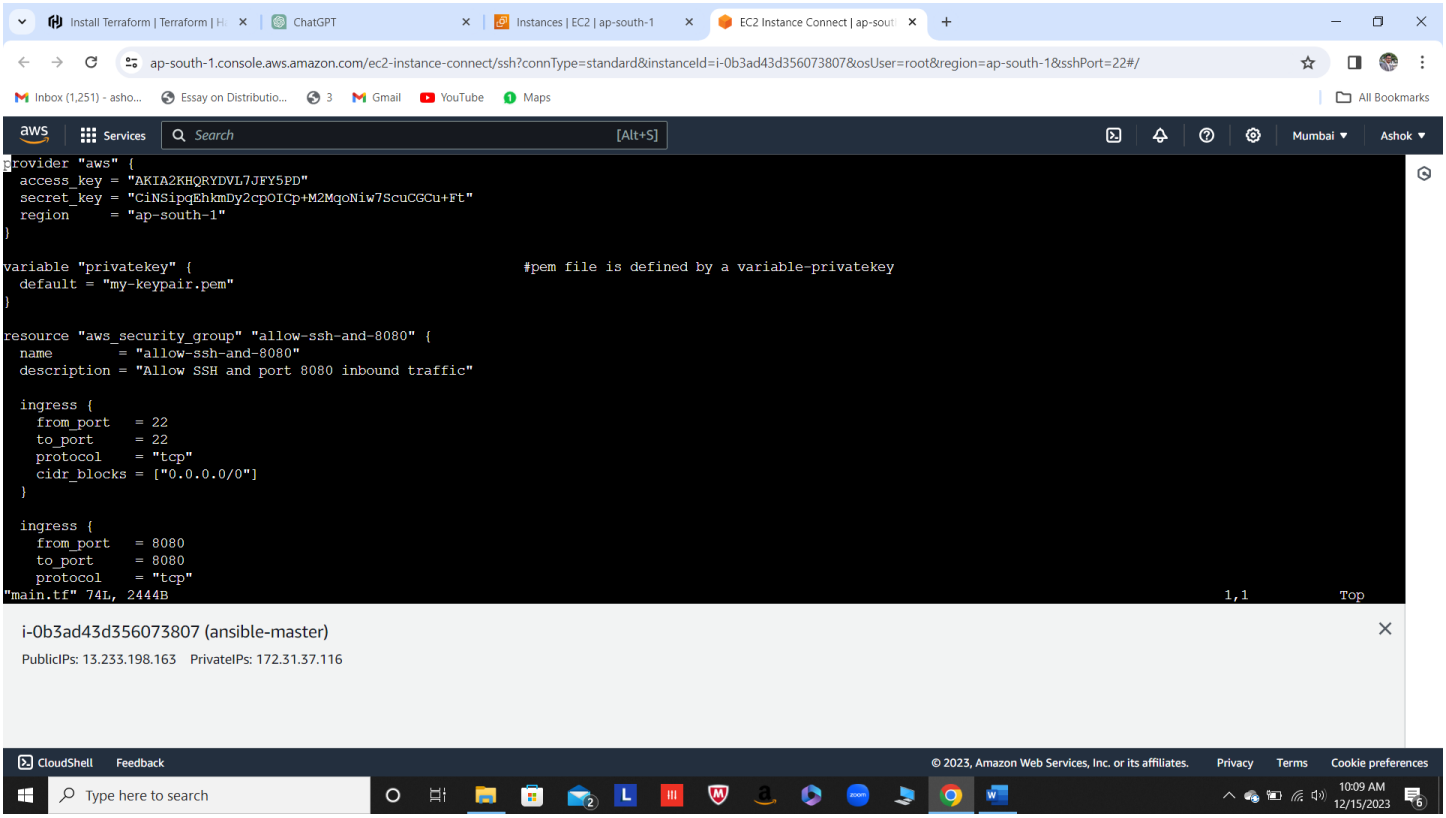
. IAM user

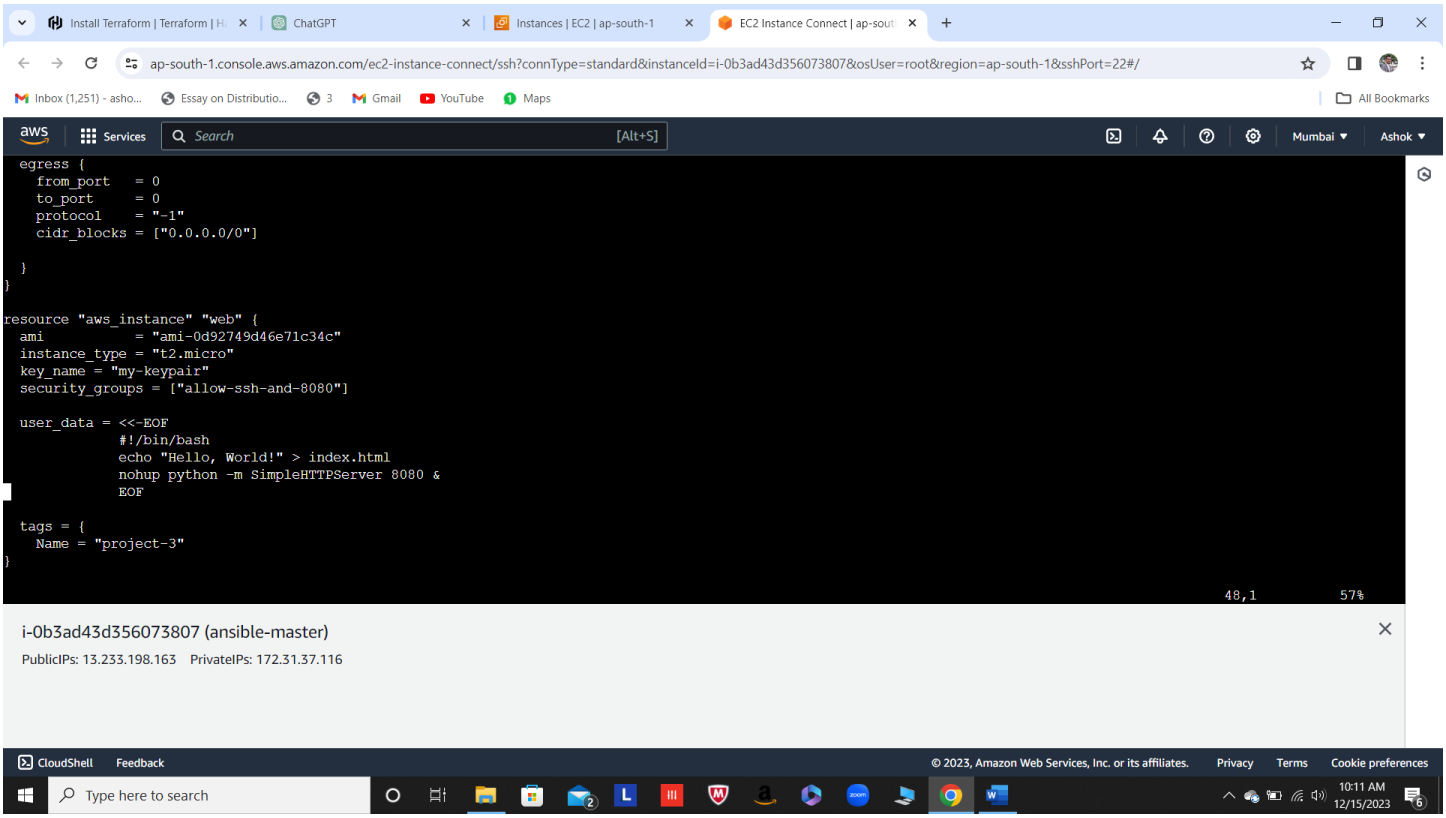

Create access key | IAM | Global
us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/details/ashok/create-access-key
Services
Search
[Alt+S]
Global
Ashok
Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.
IAM > Users > ashok > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Retrieve access keys
Info
Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.
Access key
Secret access key
AKIA2KHQRYDVOZ2CTTMV
****
Show
Access key best practices
Never store your access key in plain text, in a code repository, or in code.
Disable or delete access key when no longer needed.
Enable least-privilege permissions.
Rotate access keys regularly.
For more details about managing access keys, see the best practices for managing AWS access keys.
Download .csv file
Done
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
2:49 PM
11/21/2023

- terraform --version



. vi ec2.tf





- **Terraform init:** Initializes a Terraform working directory. This command is used to set up the backend, download necessary plugins, and prepare the directory for Terraform configuration.

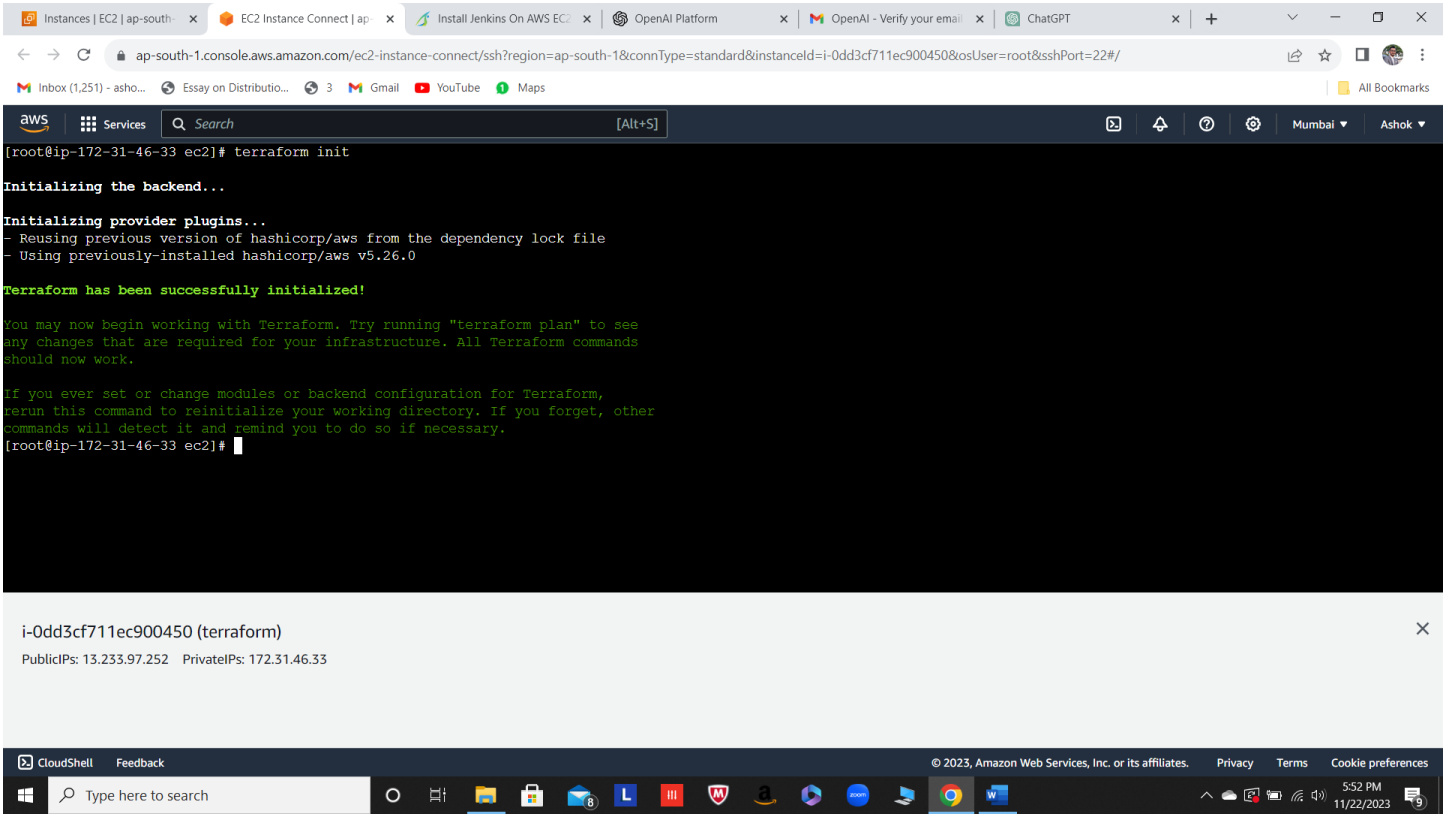
- **Terraform validate:** Validates the syntax and format of the Terraform configuration files. It checks for errors without actually executing the configuration.

- **Terraform plan:** Generates an execution plan, showing what actions Terraform will take to create or modify infrastructure. This command helps you understand the changes before applying them.

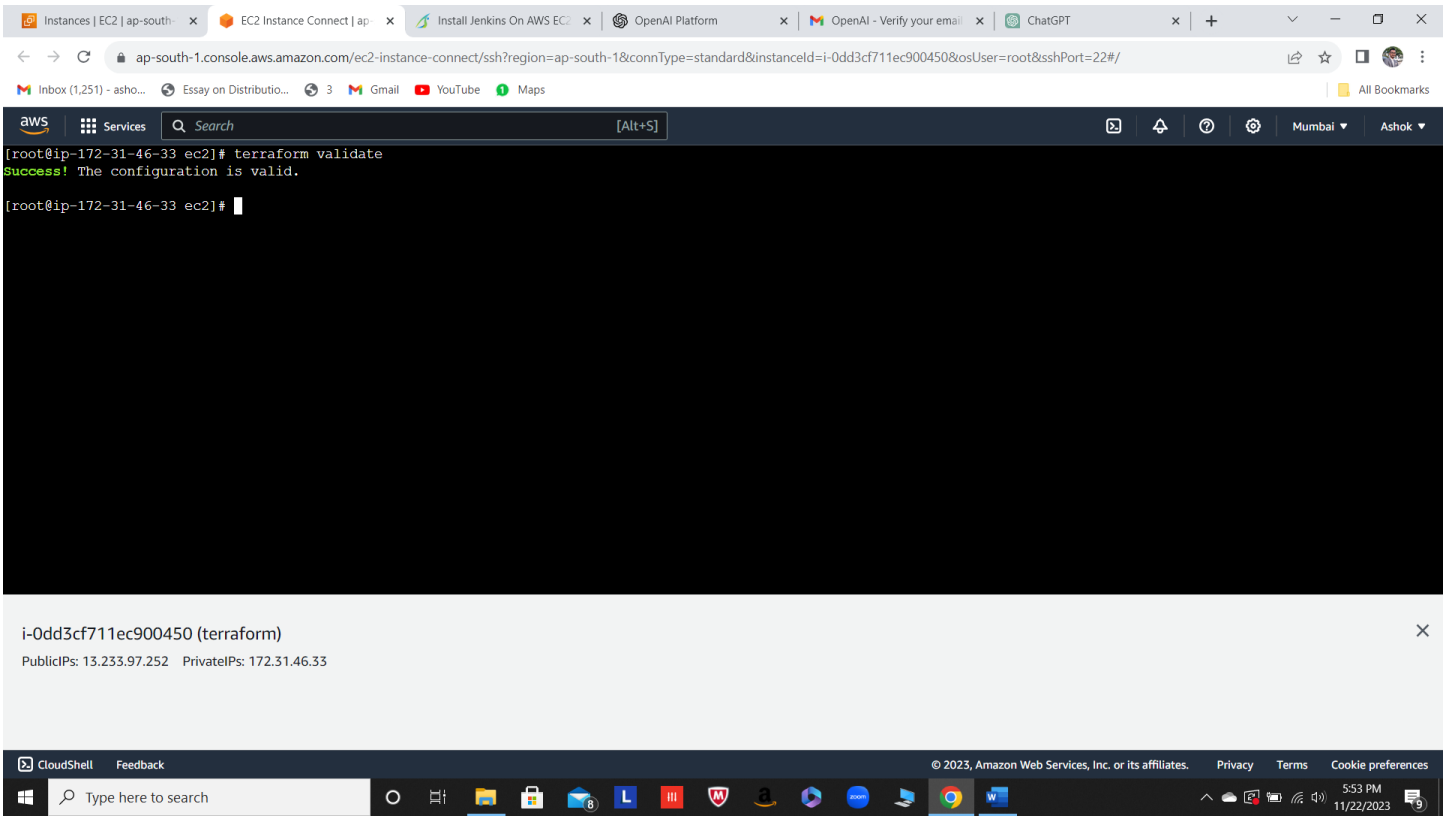
- **Terraform apply:** Applies the changes specified in the Terraform configuration. This command creates or modifies infrastructure according to the execution plan generated by **terraform plan.**

- **Terraform destroy:** Destroys the Terraform-managed infrastructure. It deletes all resources created by Terraform based on the configuration.

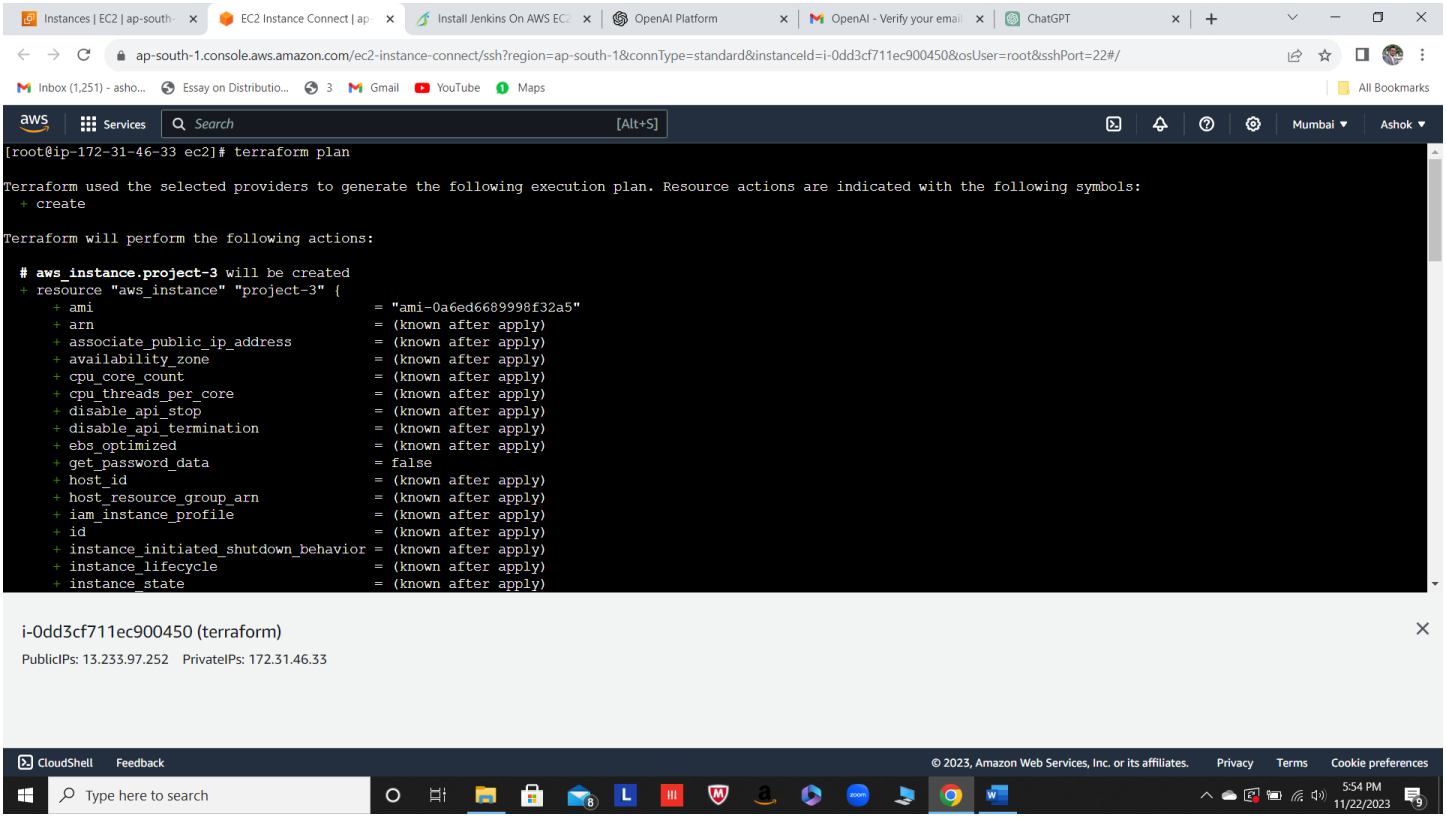
- Terraform init

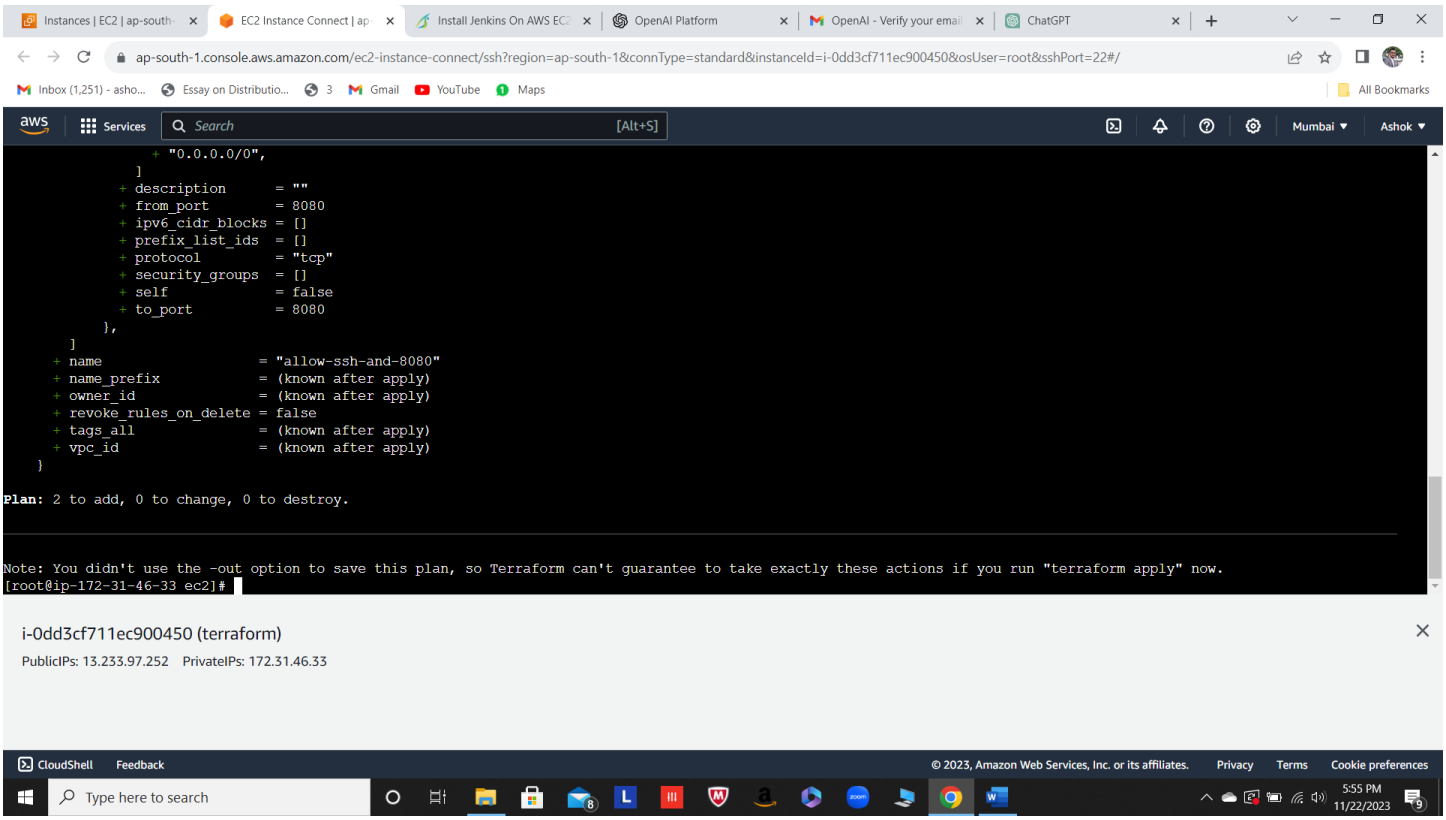


- Terraform validate

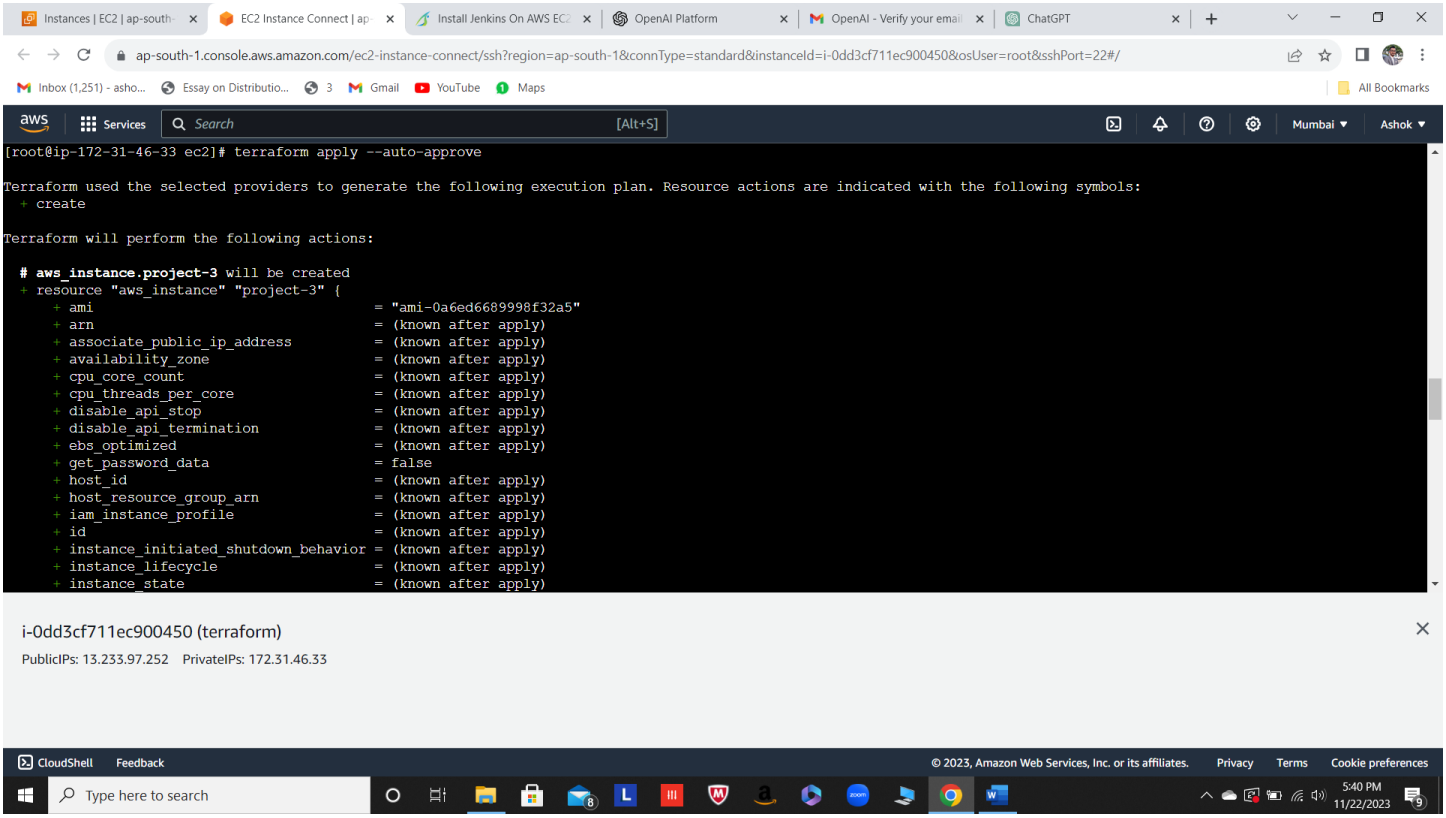


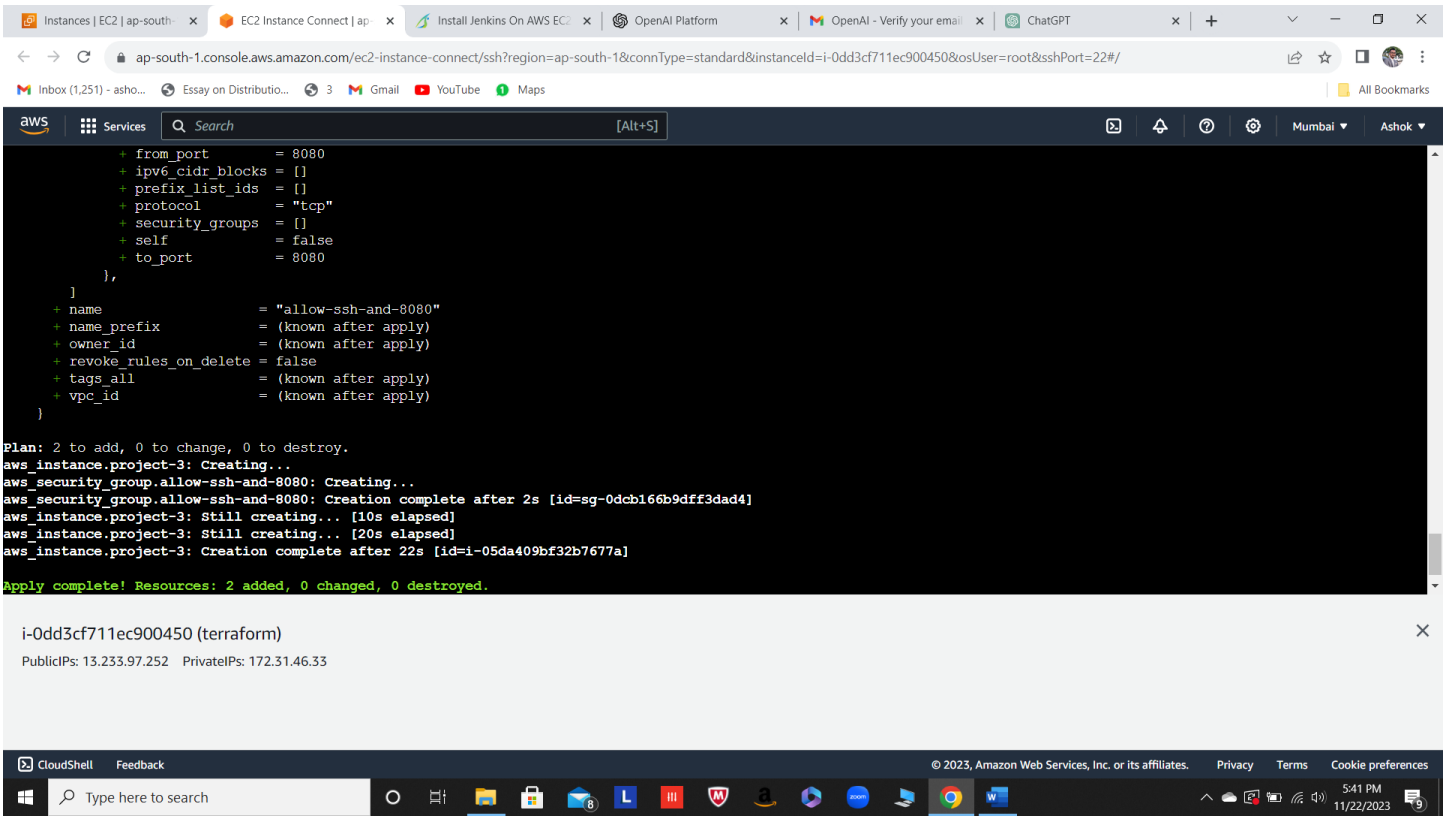
- Terraform plan



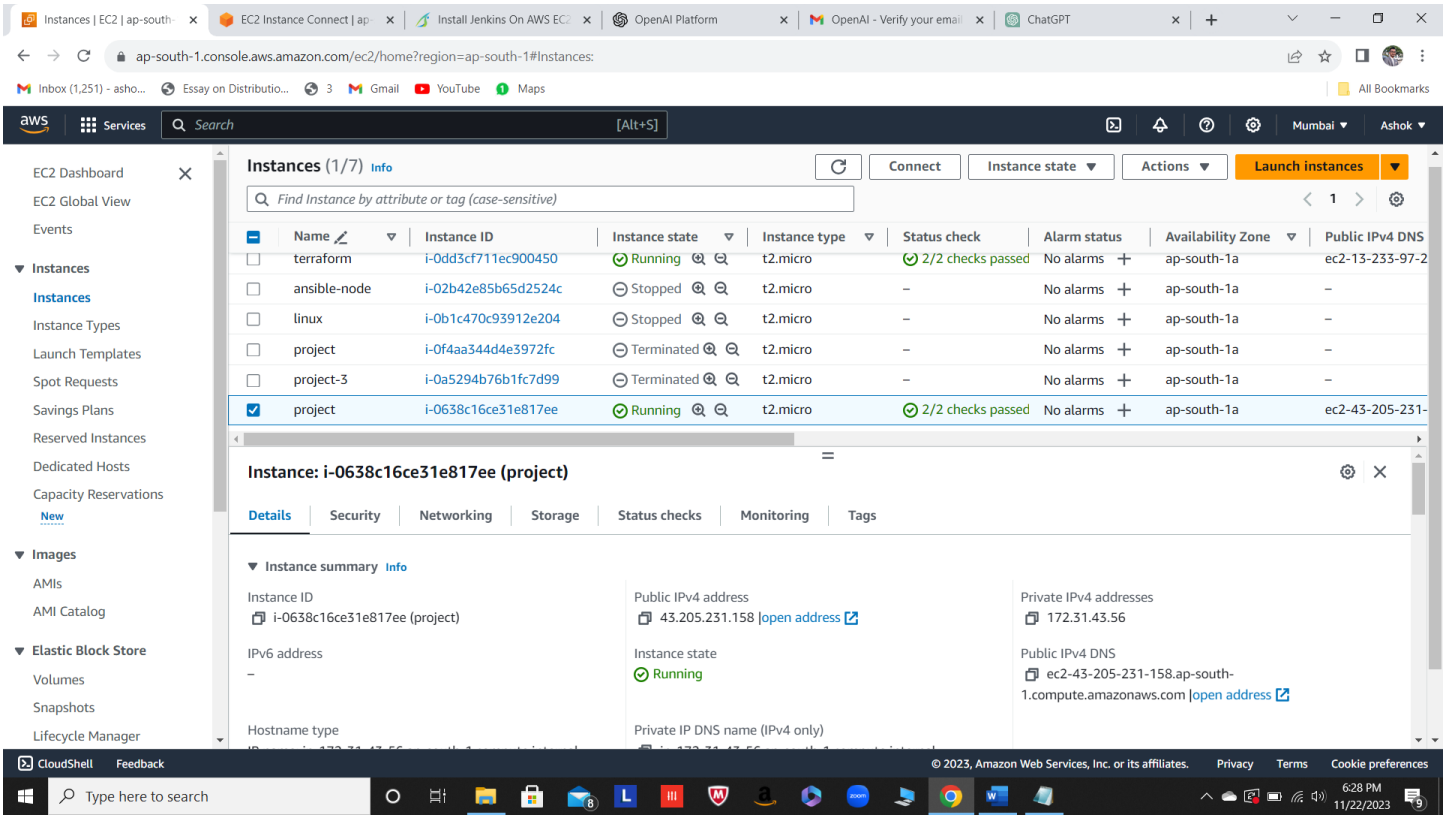


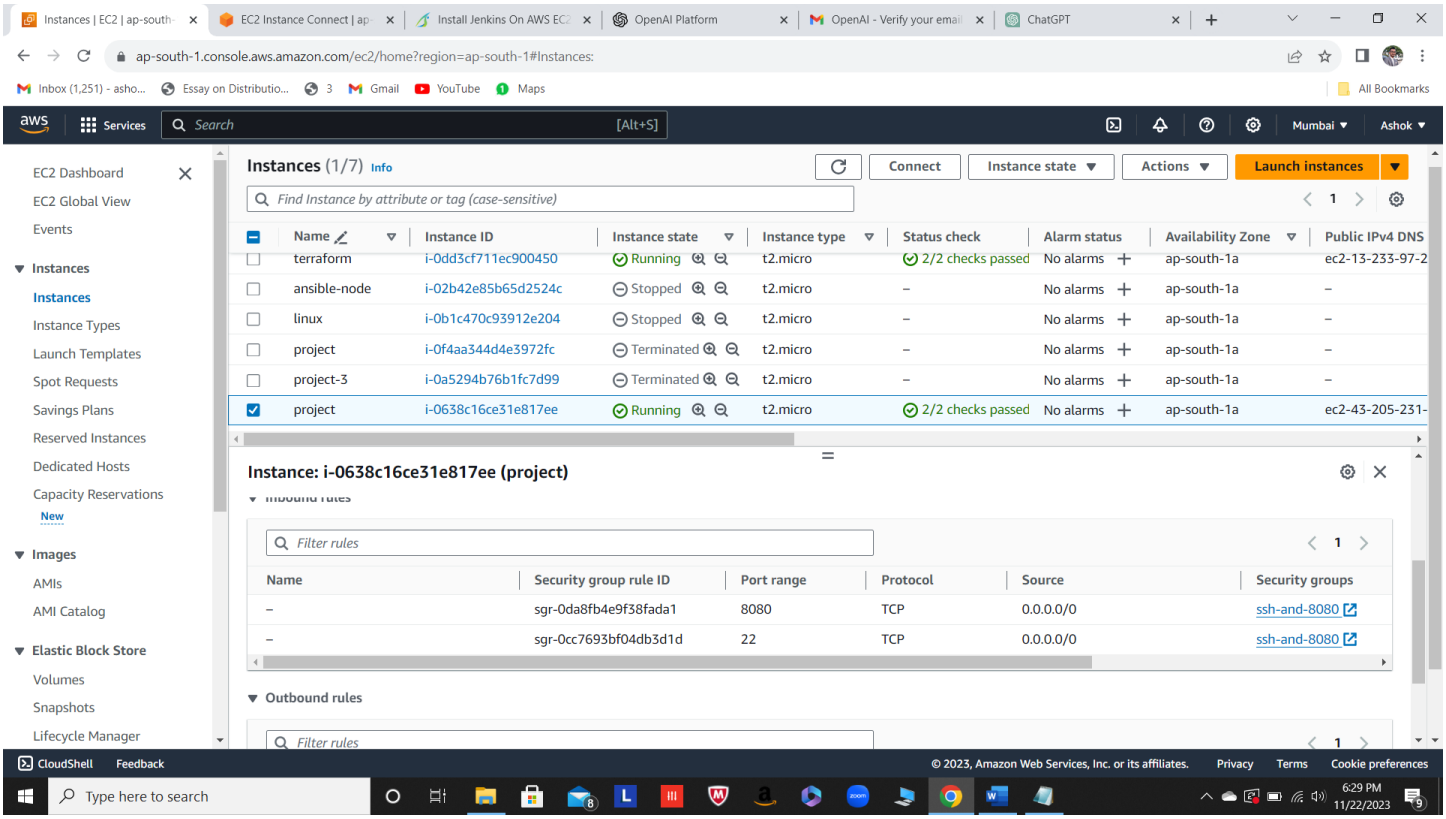
- Terraform apply –auto-approve

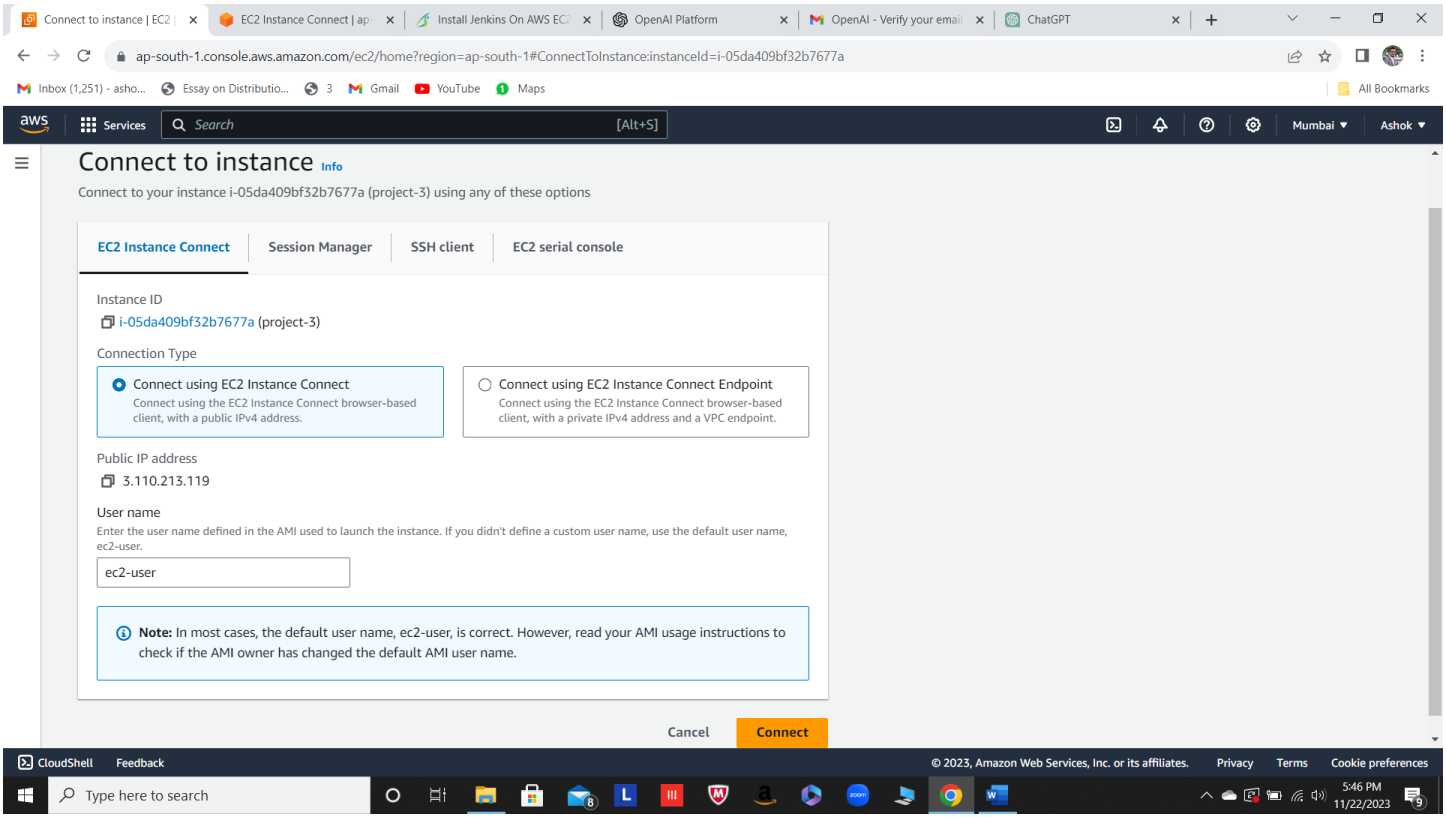


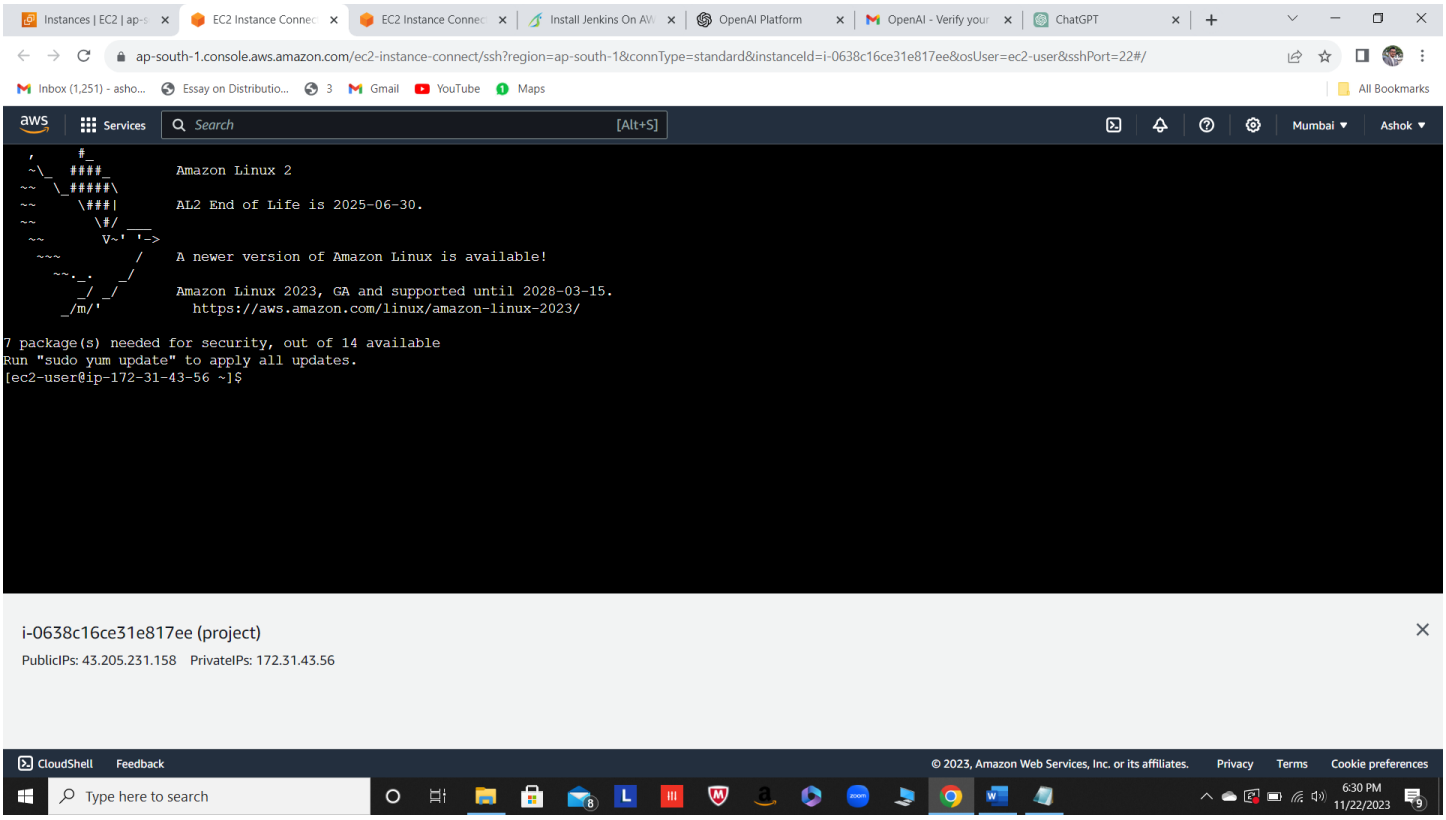


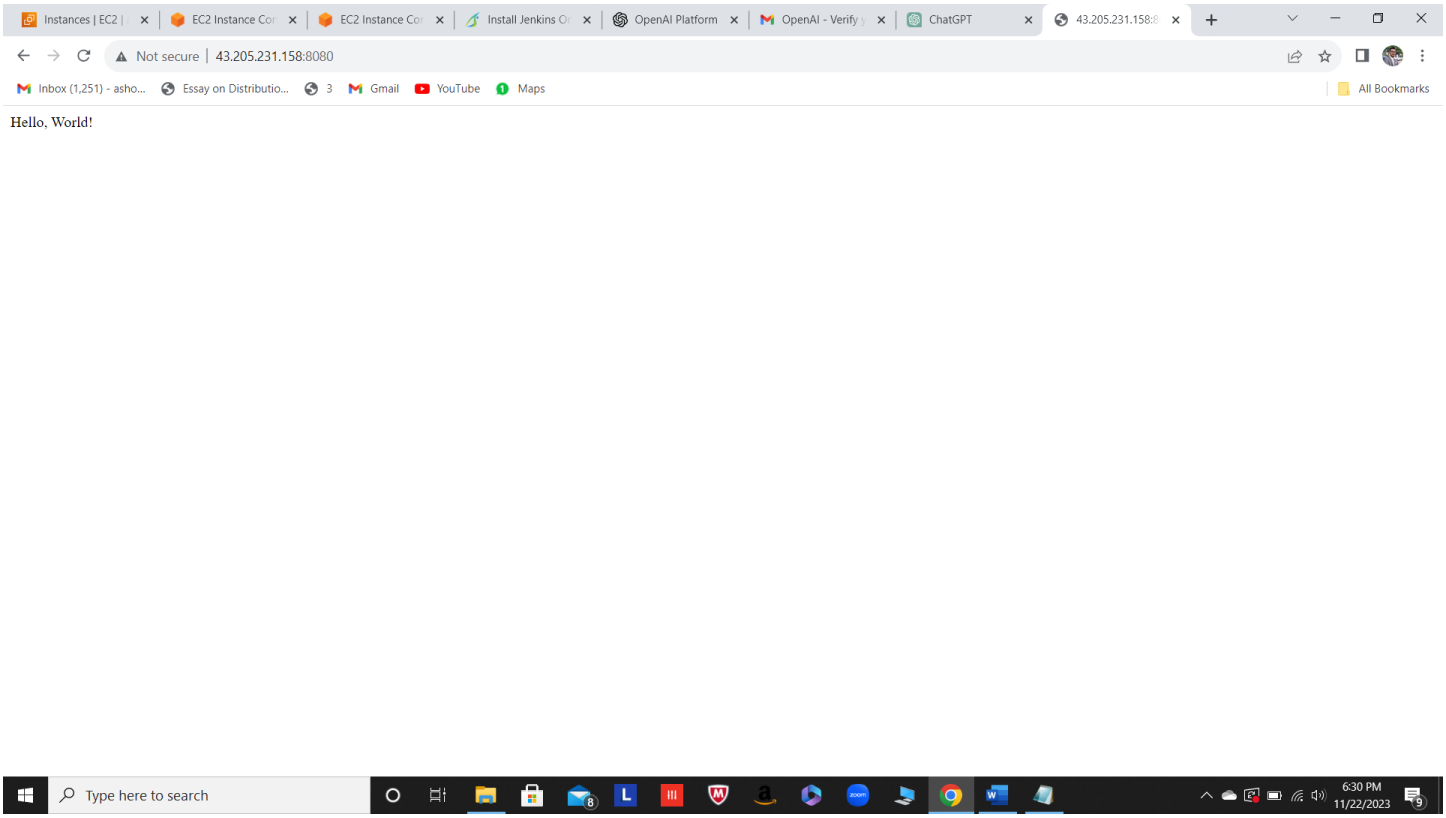
- I have created a single terraform file for the whole infrastructure. The file will have '.tf' as extension. After describing the provider which we are going to use mentioning region, I have created the security group which allow the ports 8080 and 22 to perform the specific work. Port 8080 -Hellow World, Port 22 - SSH.



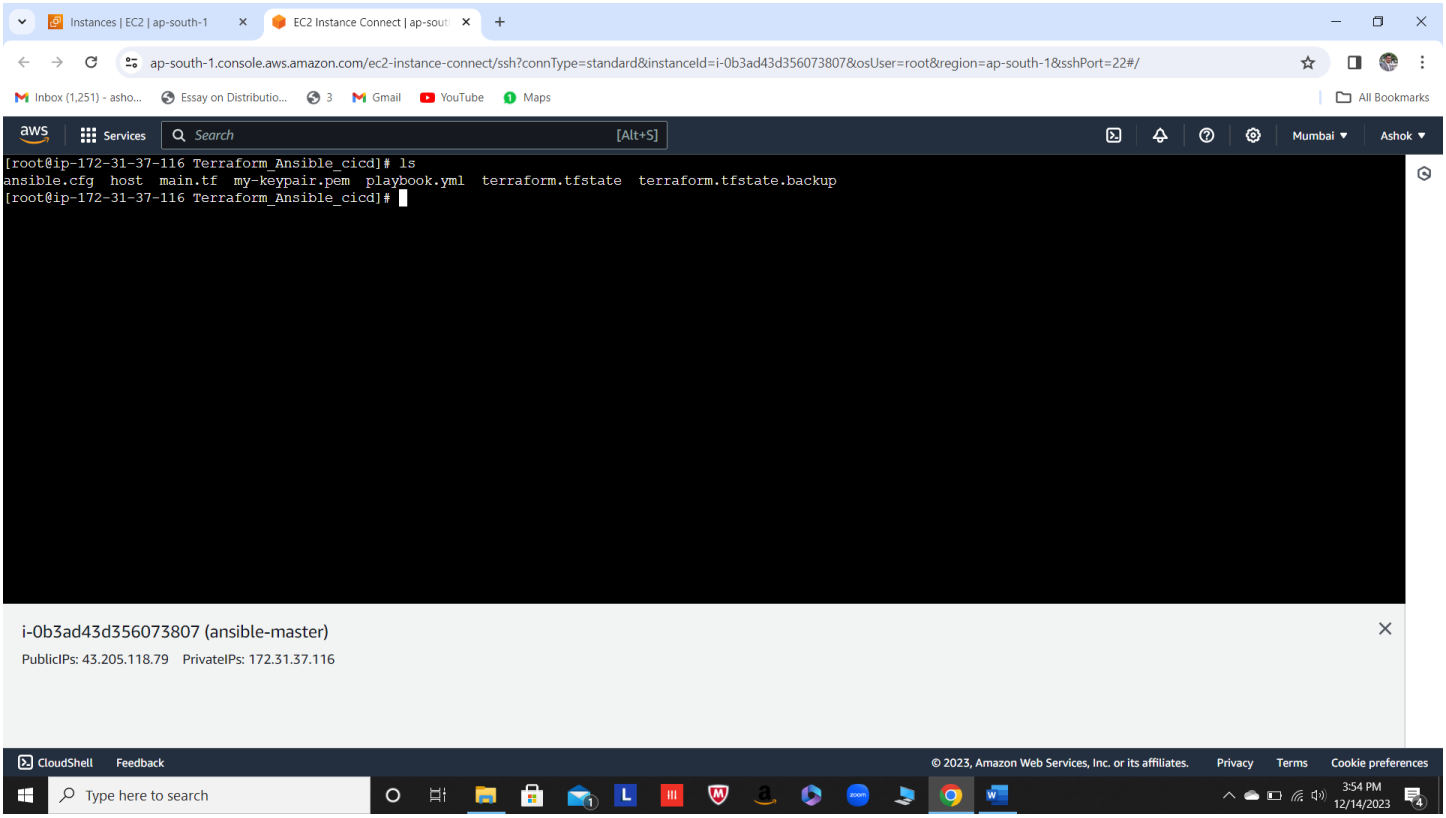



Connect to instance | EC2
EC2 Instance Connect | ap-
Install Jenkins On AWS EC2
OpenAI Platform
OpenAI - Verify your email
ChatGPT
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-05da409bf32b7677a
Inbox (1,251) - asho...
Essay on Distributio...
Gmail
YouTube
Maps
All Bookmarks
aws
Services
Search
[Alt+S]
Mumbai
Ashok
Connect to instance Info
Connect to your instance i-05da409bf32b7677a (project-3) using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-05da409bf32b7677a (project-3)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IP address
3.110.213.119
User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.
ec2-user
Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.
Cancel
Connect
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
5:46 PM
11/22/2023

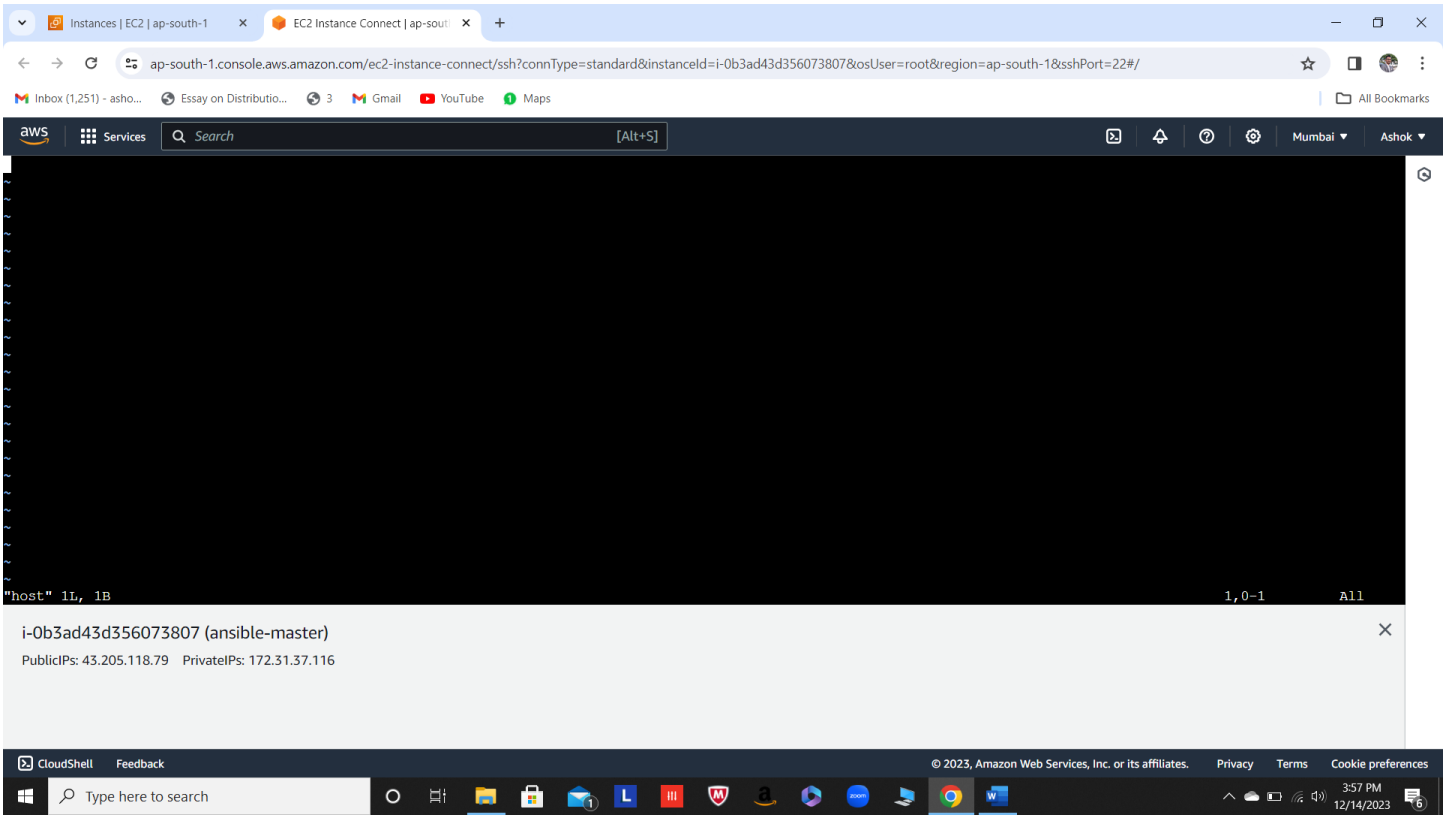

Instances | EC2 | ap-s
EC2 Instance Connec
EC2 Instance Connec
Install Jenkins On AW
OpenAI Platform
OpenAI - Verify your
ChatGPT
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0638c16ce31e817ee&osUser=ec2-user&sshPort=22#/
Inbox (1,251) - asho...
Essay on Distributio...
Gmail
YouTube
Maps
All Bookmarks
aws
Services
Search
[Alt+S]
Mumbai
Ashok
Amazon Linux 2
AL2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
7 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-43-56 ~]$
i-0638c16ce31e817ee (project)
PublicIPs: 43.205.231.158 PrivateIPs: 172.31.43.56
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
6:30 PM
11/22/2023


Instances | EC2 |
EC2 Instance Con
EC2 Instance Con
Install Jenkins On
OpenAI Platform
OpenAI - Verify y
ChatGPT
43.205.231.158:8
Not secure | 43.205.231.158:8080
Inbox (1,251) - asho...
Essay on Distributio...
3
Gmail
YouTube
Maps
All Bookmarks
Hello, World!
Type here to search
6:30 PM
11/22/2023

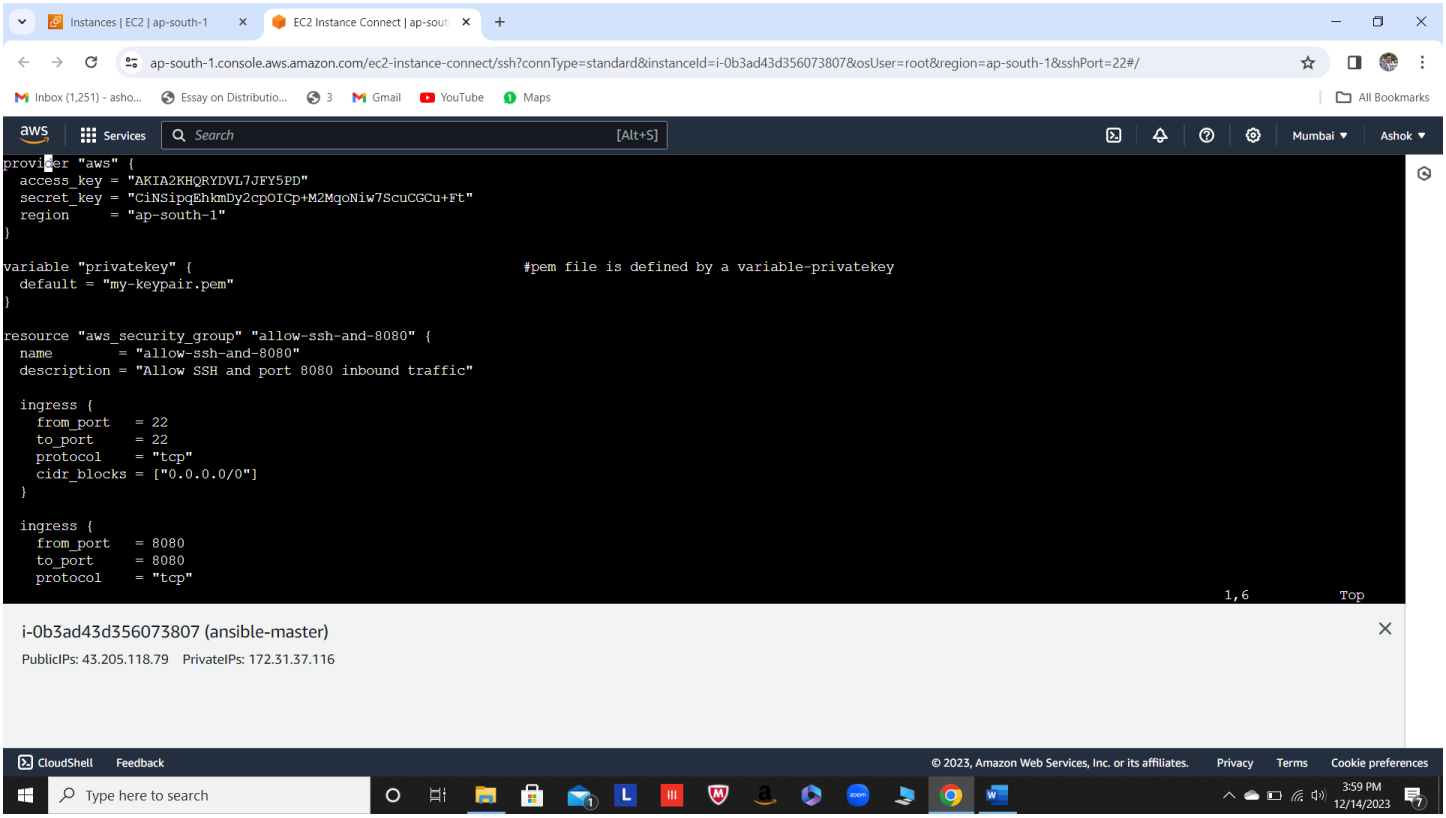
2. Run an Ansible Playbook with Terraform.



- vi host (empty file)



- vi main.tf



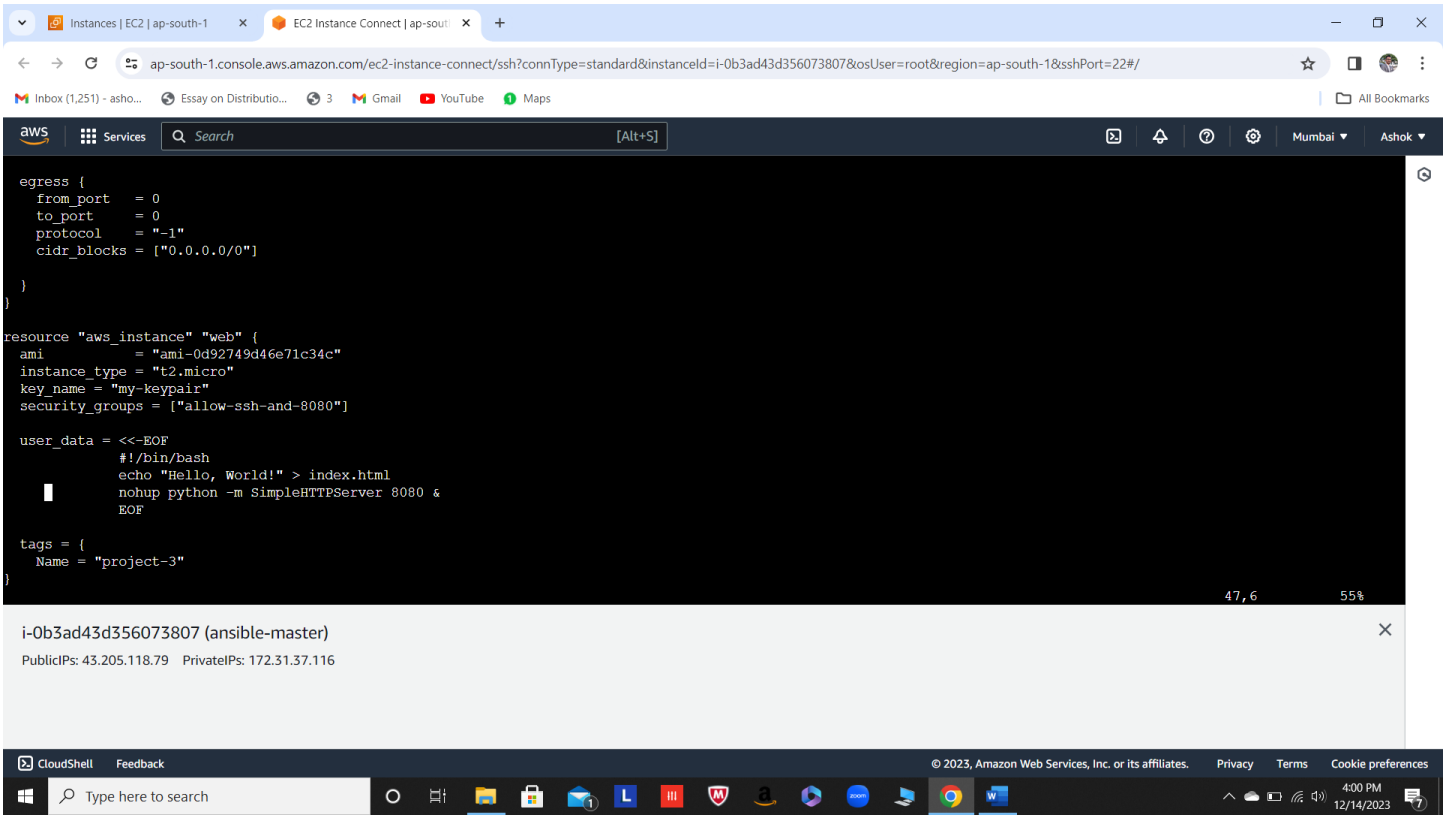
```
provider "aws" {
  access_key = "AKIA2KHQRYDVL7JFY5PD"
  secret_key = "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"
  region     = "ap-south-1"
}

variable "privatekey" {                                  #pem file is defined by a variable-privatekey
  default = "my-keypair.pem"
}

resource "aws_security_group" "allow-ssh-and-8080" {
  name        = "allow-ssh-and-8080"
  description = "Allow SSH and port 8080 inbound traffic"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 8080
    to_port     = 8080
    protocol    = "tcp"
```

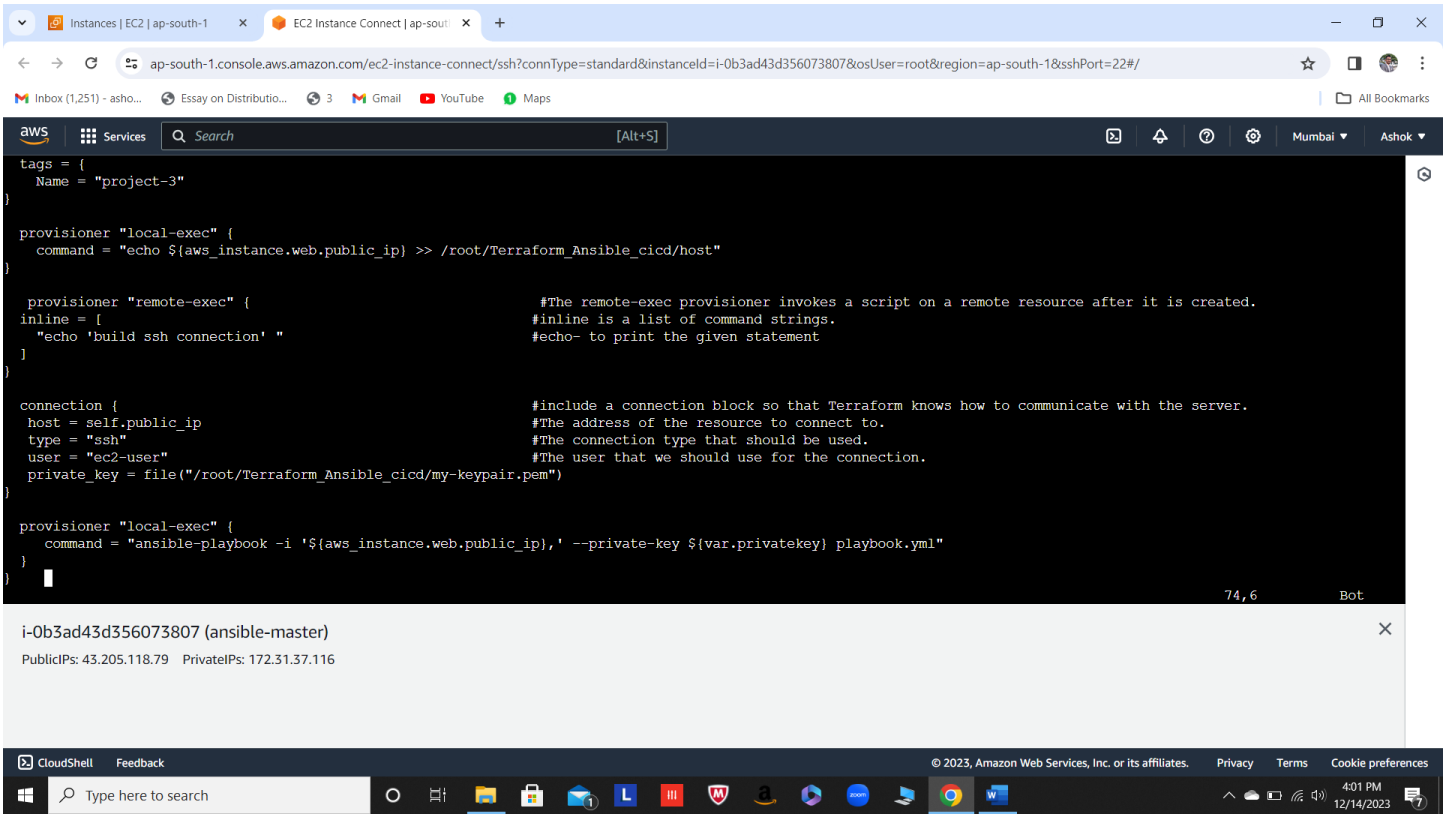
```
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]

  }
}

resource "aws_instance" "web" {
  ami           = "ami-0d92749d46e71c34c"
  instance_type = "t2.micro"
  key_name = "my-keypair"
  security_groups = ["allow-ssh-and-8080"]

  user_data = <<-EOF
              #!/bin/bash
              echo "Hello, World!" > index.html
              nohup python -m SimpleHTTPServer 8080 &
              EOF

  tags = {
    Name = "project-3"
}
```



- **Provider:** In Terraform, a provider is a plugin that enables Terraform to interact with a specific cloud or infrastructure platform. Providers are responsible for translating the desired state described in your Terraform configuration into API calls to create, update, or delete resources on the target platform. Providers handle the communication between Terraform and the underlying infrastructure.

- **Variable:** You declare variables using the **variable** block in your Terraform configuration. Variables can have a name, a type, an optional default value, and other constraints.

- **Security group:** In the context of cloud computing, a security group is a fundamental component used to control inbound and outbound traffic to resources such as virtual machines (VMs) or instances in a network. Security groups act as virtual firewalls that define rules governing the traffic flow to and from these resources. They are commonly used in cloud platforms like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).Security groups play a crucial role in ensuring the security and isolation of resources within cloud environments, and they are an essential part of infrastructure as code (IaC) practices using tools like Terraform.

- **Aws instance:** In Amazon Web Services (AWS), an instance refers to a virtual server in the cloud. Instances are the fundamental building blocks of computing resources on AWS, providing on-demand and scalable compute capacity. AWS instances are fundamental to the AWS cloud computing model, providing the compute capacity needed to run applications and services. They are a central component in building scalable, flexible, and resilient cloud architectures.

- **Provisioner:** In Terraform, provisioners are a mechanism for defining and executing scripts or commands on a resource as part of the resource creation or update process. Provisioners are typically used to perform tasks such as initializing software, configuring resources, or running custom scripts on instances after they have been created. There are two main types of provisioners in Terraform: local-exec and remote-exec.

  Local-Exec Provisioner: The **local-exec** provisioner allows you to run commands on the machine running Terraform, rather than on the resource being managed. This can be useful for tasks that don't require execution on the target resource, such as local script execution or interacting with local tools.

  Remote-Exec Provisioner: The **remote-exec** provisioner connects to the created resource via SSH and executes commands or scripts on that resource. This is commonly used for configuring software on remote instances.

  Note:

- While provisioners can be helpful, it's often preferable to use configuration management tools (e.g., Ansible, Chef, Puppet) for complex software configuration tasks. Terraform provisioners are more suitable for simple tasks or situations where external tools are not applicable.
- Using provisioners should be done cautiously, as they can introduce challenges in terms of idempotence, reliability, and maintainability. It's recommended to consider infrastructure as code best practices and explore alternative solutions when possible.
- Terraform also supports other provisioners, such as file, template, and null provisioners, which offer additional capabilities for managing resources.

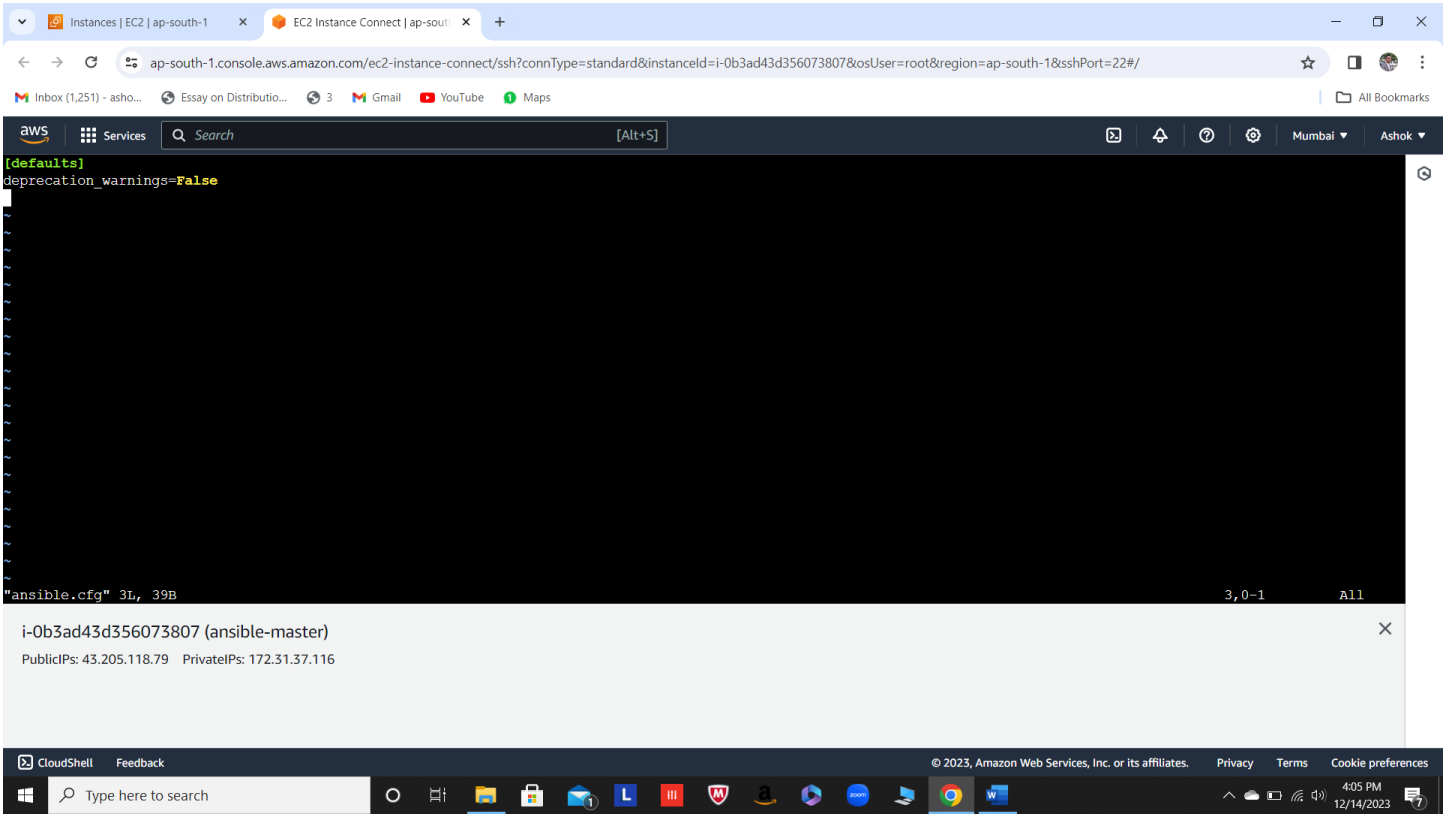
- **Ansible.cfg:** The **ansible.cfg** file is a configuration file used by Ansible to define settings and preferences for running Ansible commands and playbooks. It is not directly related to Terraform but is used when you are managing infrastructure with Ansible, which is a complementary tool to Terraform for configuration management and automation.

  In the context of using Ansible with Terraform, you might have an Ansible configuration file (**ansible.cfg**) to specify settings such as the location of inventory files, where to find roles, default user, and other options.

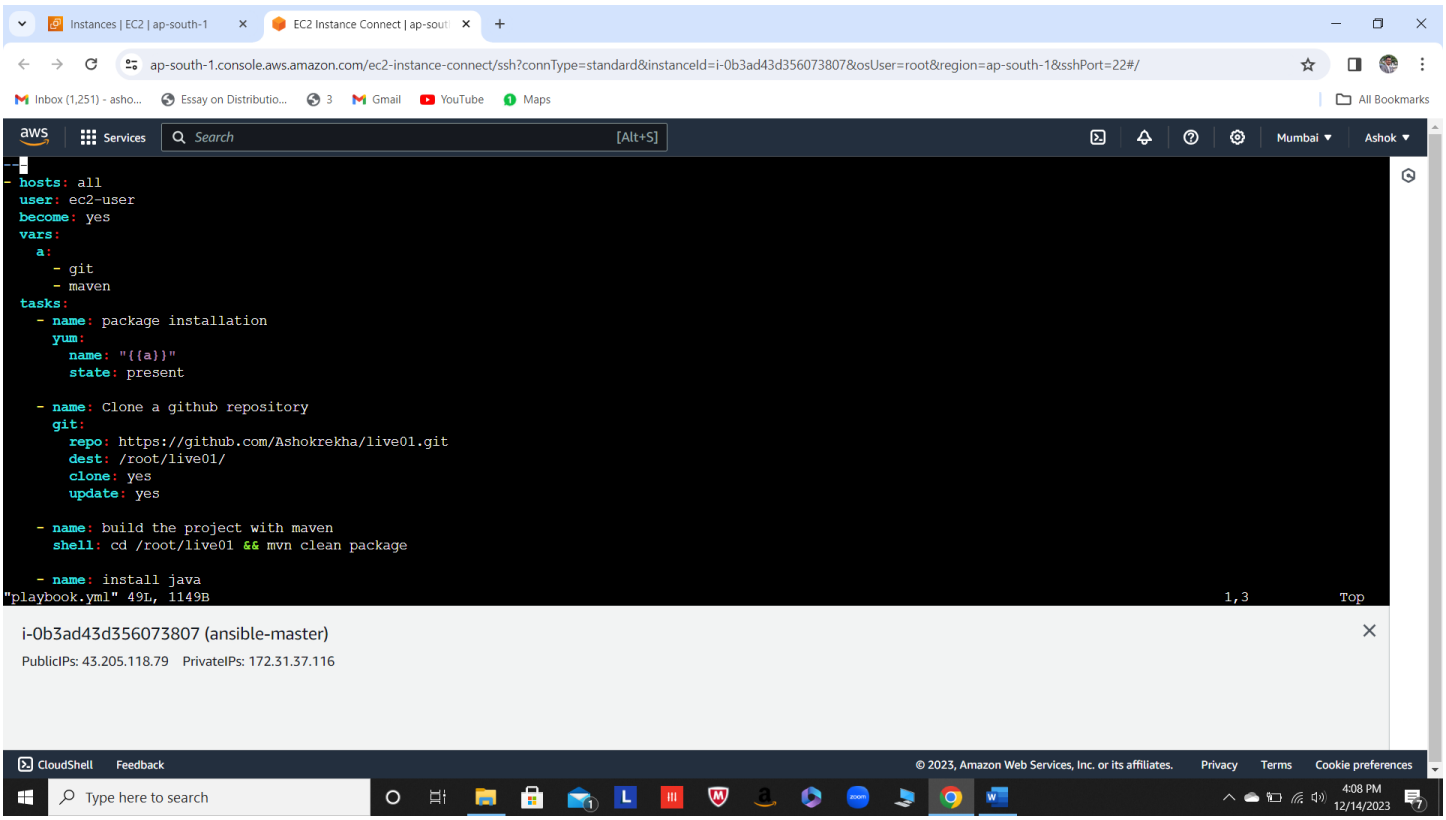
- **Host:** In Ansible, the inventory file (often referred to as the "hosts file") is a crucial component that defines the target hosts or nodes on which Ansible should operate. The inventory file is used to specify the hosts and groups of hosts that Ansible will manage. The default name for the inventory file is **hosts.**
  Usage with Terraform:
  When integrating Ansible with Terraform, Terraform can generate an inventory file dynamically based on the infrastructure it provisions. The generated inventory file can then be used by Ansible to configure and manage the provisioned resources.

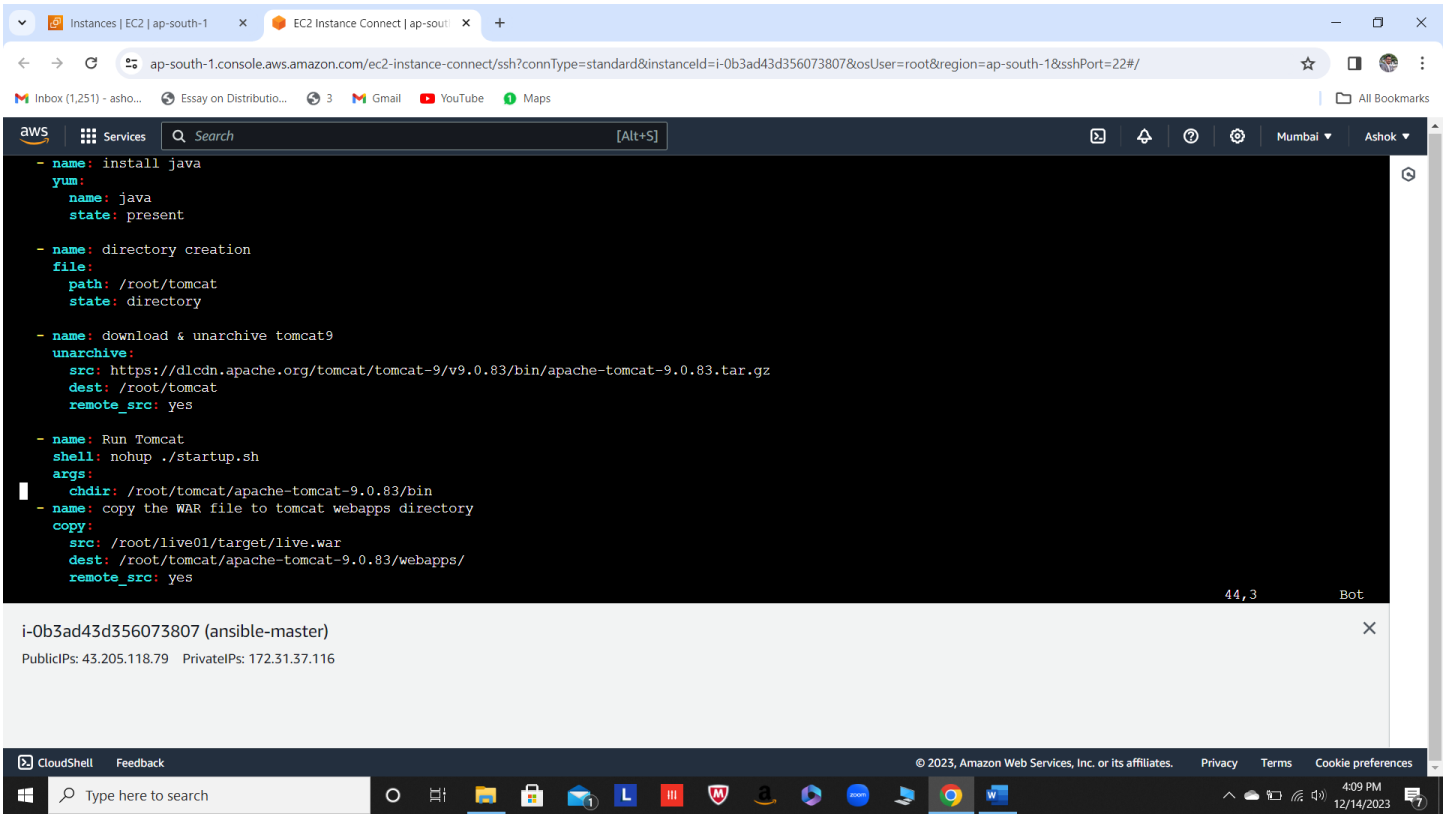
  Terraform can output the necessary information (IP addresses, SSH keys, etc.) to create a dynamic inventory script or file that Ansible can use.

- vi ansible.cfg

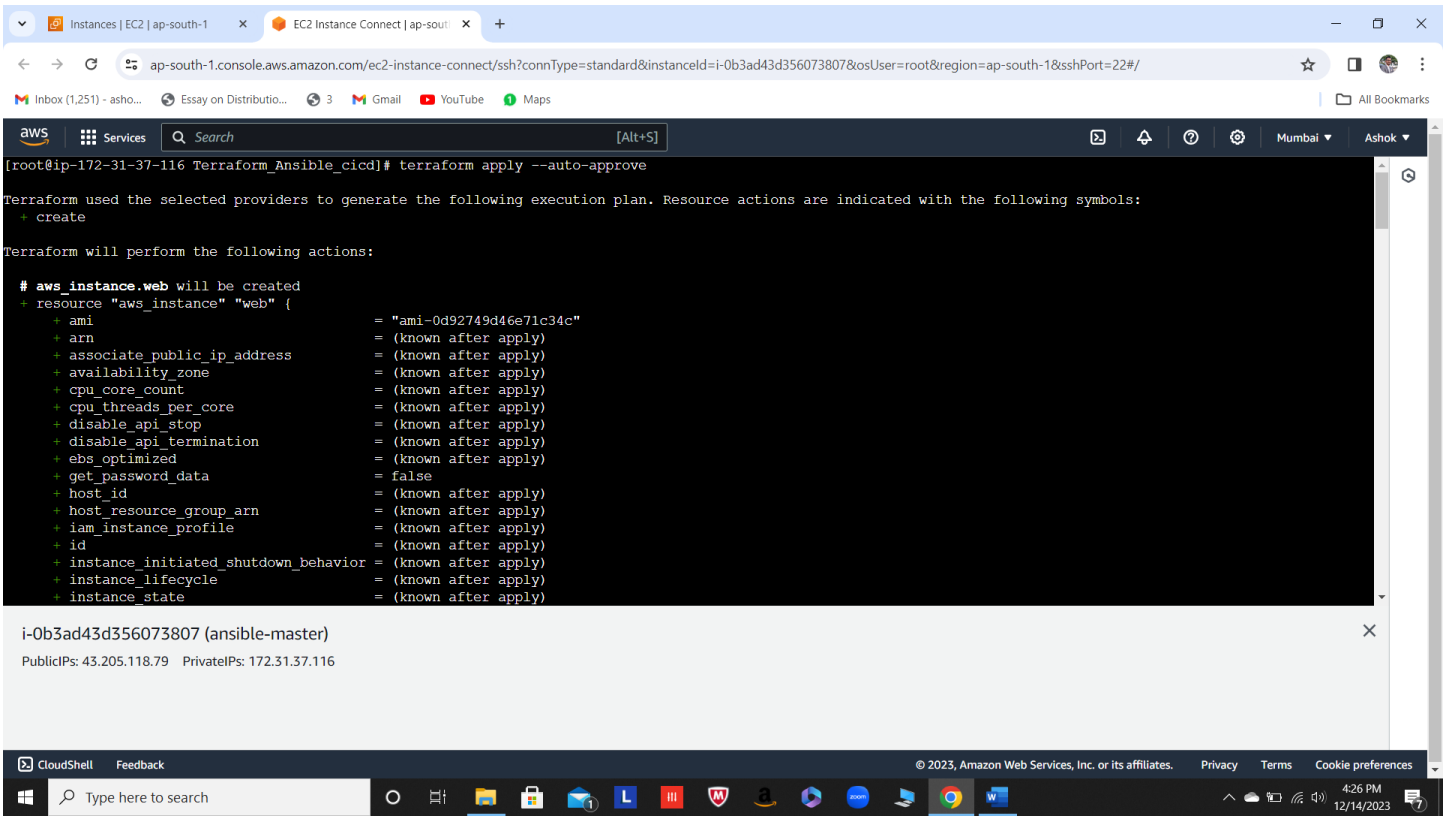


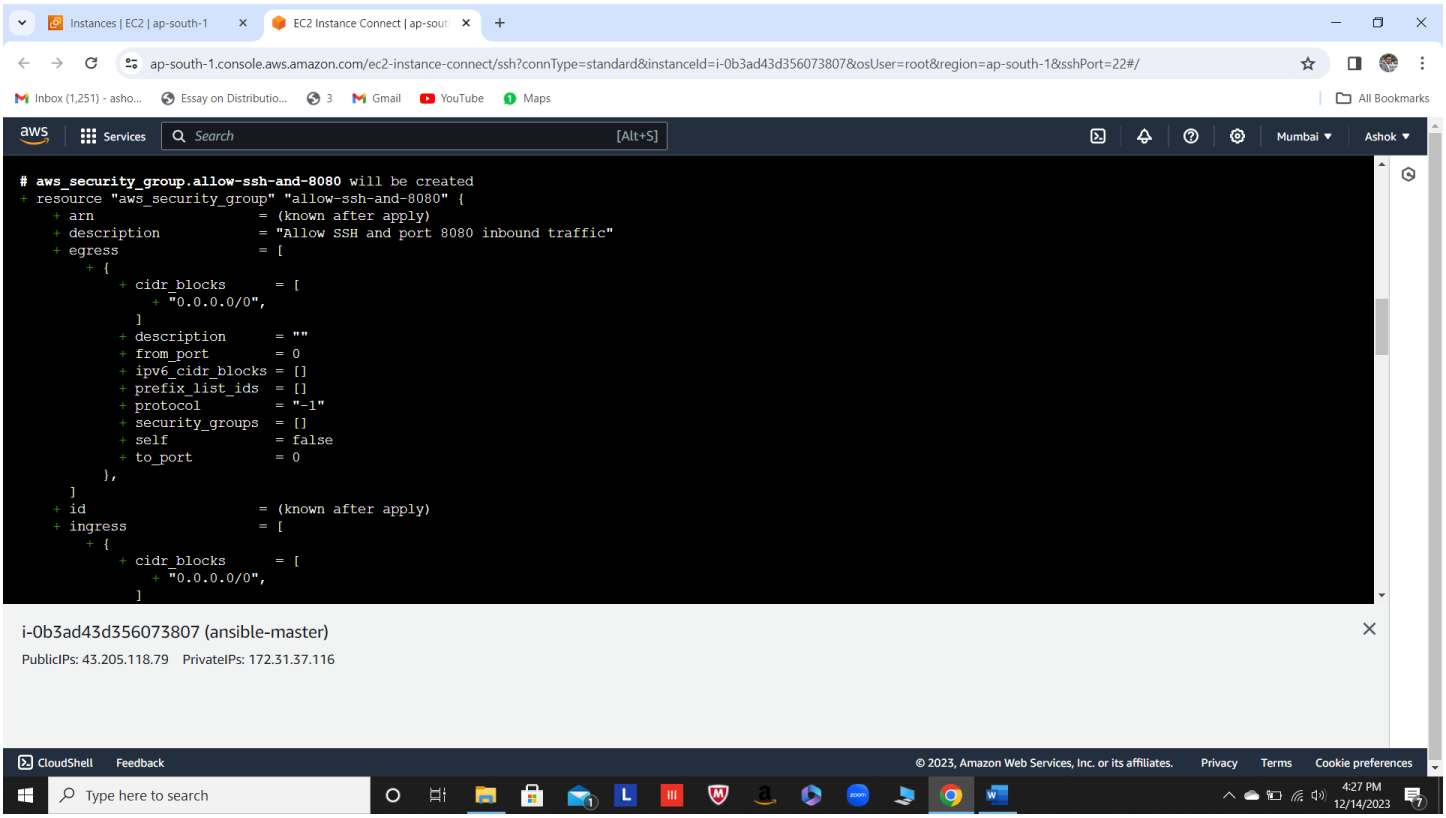
- vi playbook.yml (Ansible playbook)

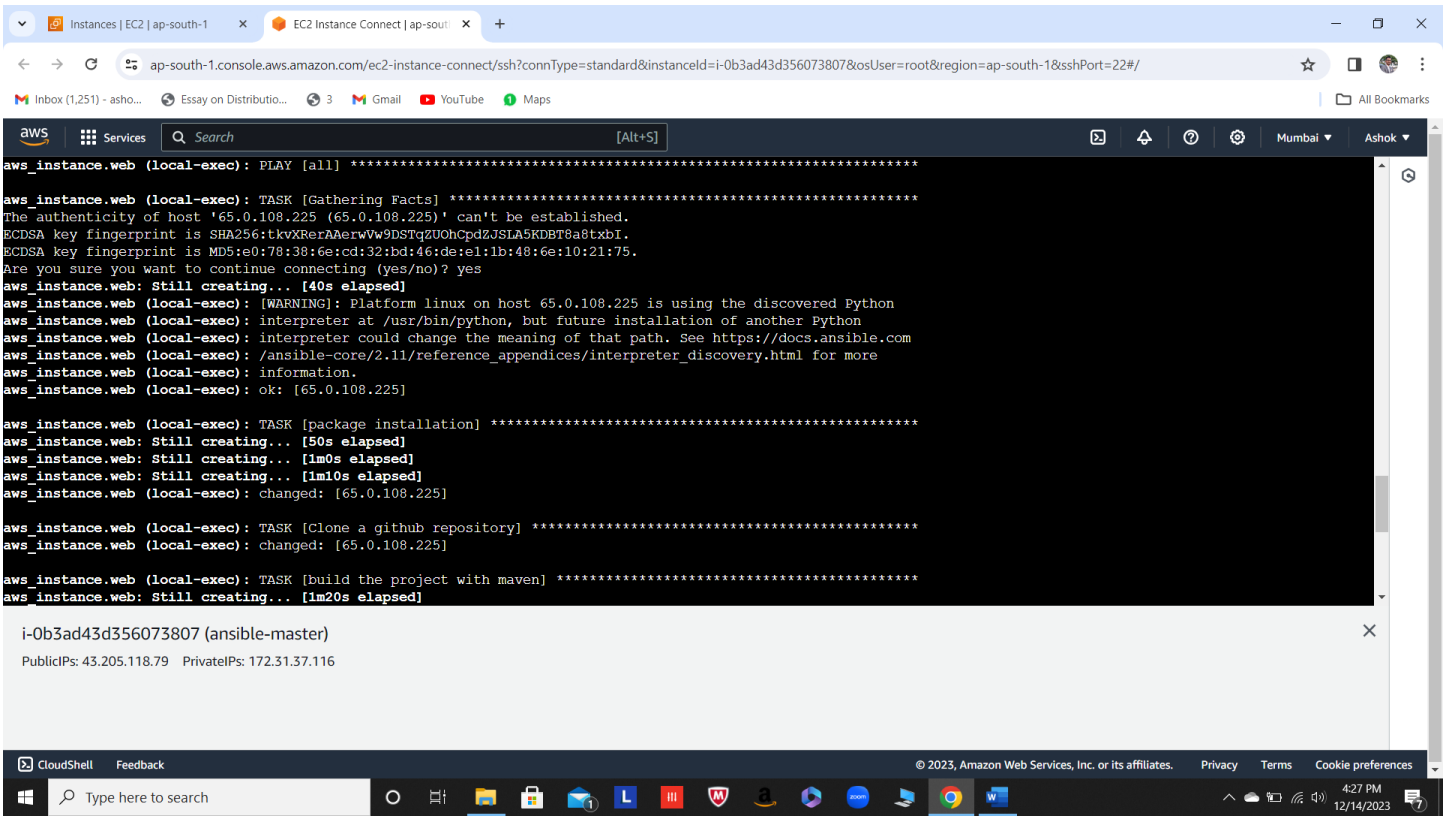


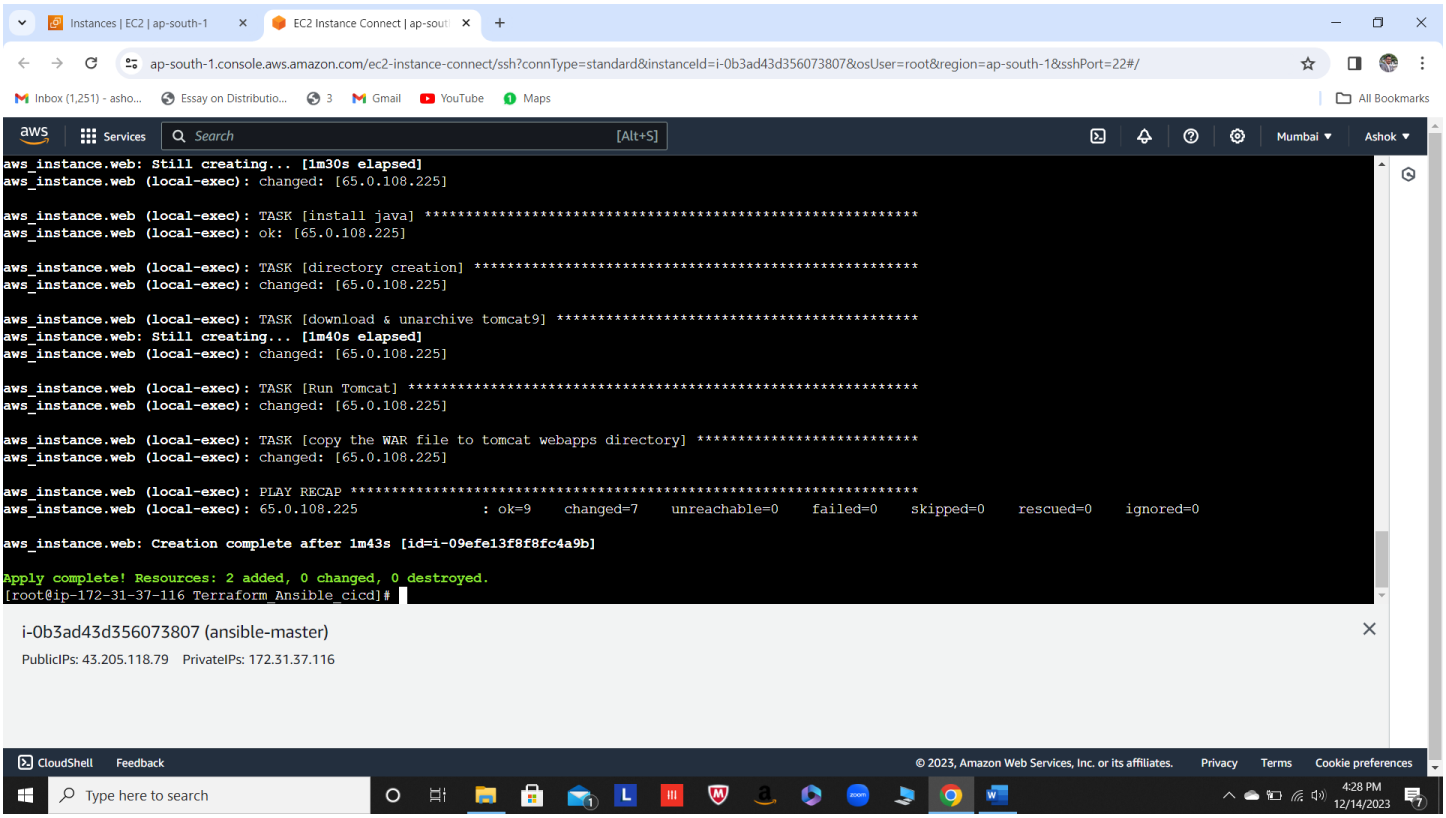


- terraform apply –auto-approve

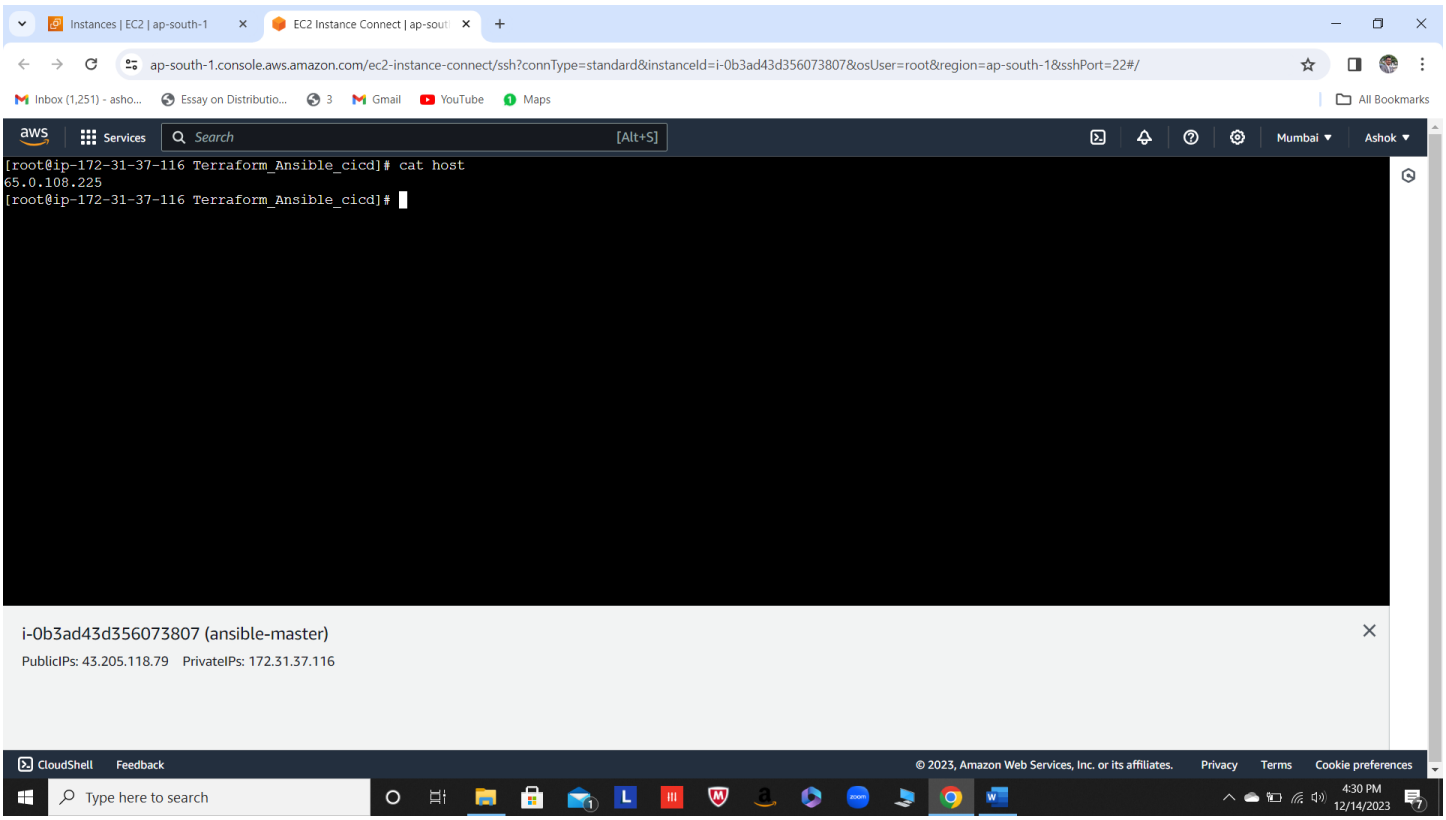



Instances | EC2 | ap-south-1
EC2 Instance Connect | ap-sout
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0b3ad43d356073807&osUser=root&region=ap-south-1&sshPort=22#/
Inbox (1,251) - asho...
Essay on Distributio...
Gmail
YouTube
Maps
All Bookmarks
aws
Services
Search
[Alt+S]
Mumbai
Ashok
# aws_security_group.allow-ssh-and-8080 will be created
+ resource "aws_security_group" "allow-ssh-and-8080" {
+ arn = (known after apply)
+ description = "Allow SSH and port 8080 inbound traffic"
+ egress = [
+ {
+ cidr_blocks = [
+ "0.0.0.0/0",
]
+ description = ""
+ from_port = 0
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "-1"
+ security_groups = []
+ self = false
+ to_port = 0
},
]
+ id = (known after apply)
+ ingress = [
+ {
+ cidr_blocks = [
+ "0.0.0.0/0",
]
i-0b3ad43d356073807 (ansible-master)
PublicIPs: 43.205.118.79 PrivateIPs: 172.31.37.116
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
4:27 PM
12/14/2023

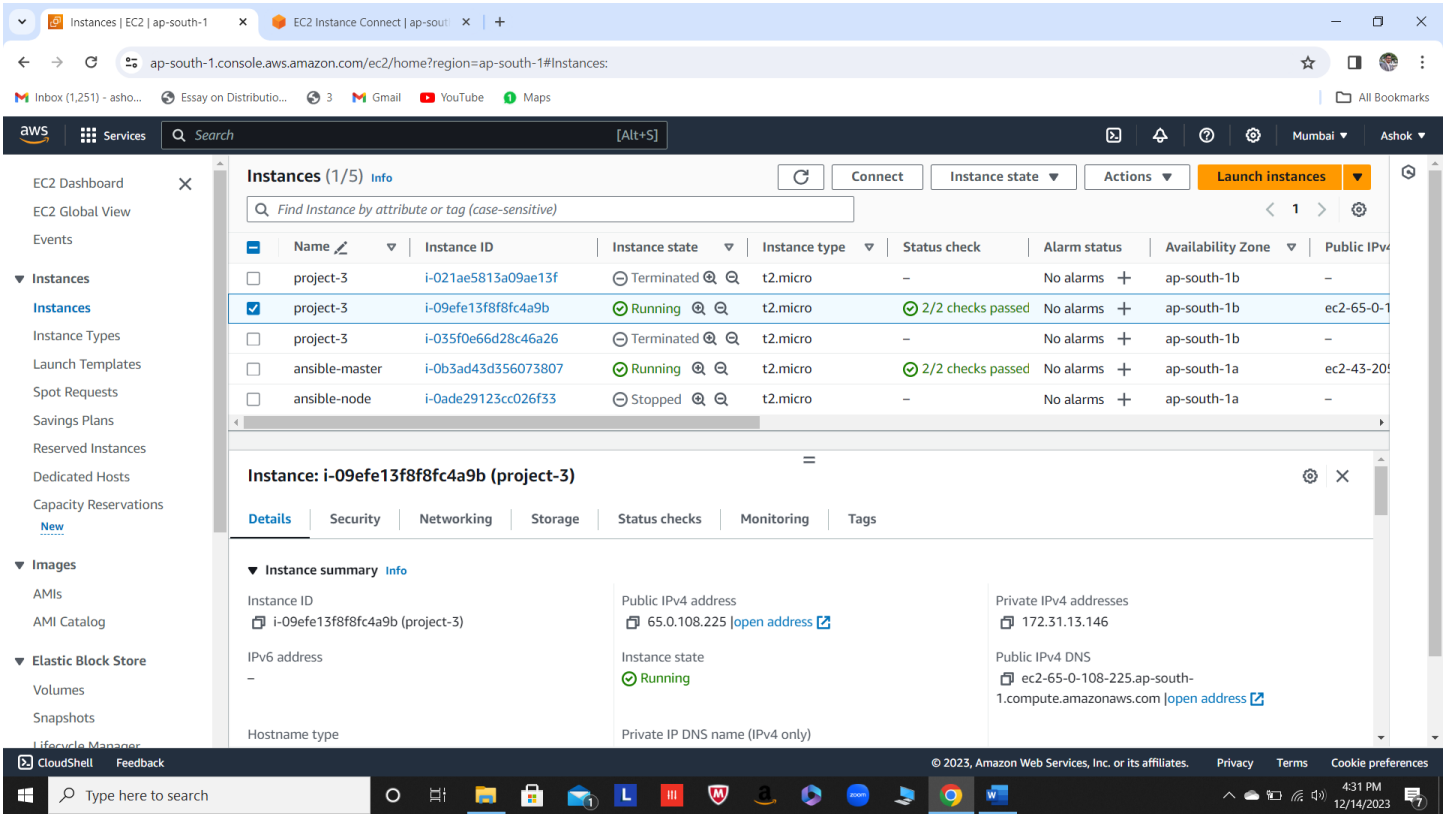

Instances | EC2 | ap-south-1
EC2 Instance Connect | ap-sout
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0b3ad43d356073807&osUser=root&region=ap-south-1&sshPort=22#/
Inbox (1,251) - asho...
Essay on Distributio...
Gmail
YouTube
Maps
All Bookmarks
aws
Services
Search
[Alt+S]
Mumbai
Ashok
aws_instance.web (local-exec): PLAY [all] ***
aws_instance.web (local-exec): TASK [Gathering Facts] ***
The authenticity of host '65.0.108.225 (65.0.108.225)' can't be established.
ECDSA key fingerprint is SHA256:tkvXRerAAerwVw9DSTqZUOhCpdZJSLA5KDBT8a8txbI.
ECDSA key fingerprint is MD5:e0:78:38:6e:cd:32:bd:46:de:e1:1b:48:6e:10:21:75.
Are you sure you want to continue connecting (yes/no)? yes
aws_instance.web: Still creating... [40s elapsed]
aws_instance.web (local-exec): [WARNING]: Platform linux on host 65.0.108.225 is using the discovered Python
aws_instance.web (local-exec): interpreter at /usr/bin/python, but future installation of another Python
aws_instance.web (local-exec): interpreter could change the meaning of that path. See https://docs.ansible.com
aws_instance.web (local-exec): /ansible-core/2.11/reference_appendices/interpreter_discovery.html for more
aws_instance.web (local-exec): information.
aws_instance.web (local-exec): ok: [65.0.108.225]
aws_instance.web (local-exec): TASK [package installation] **
aws_instance.web: Still creating... [50s elapsed]
aws_instance.web: Still creating... [1m0s elapsed]
aws_instance.web: Still creating... [1m10s elapsed]
aws_instance.web (local-exec): changed: [65.0.108.225]
aws_instance.web (local-exec): TASK [Clone a github repository] ***
aws_instance.web (local-exec): changed: [65.0.108.225]
aws_instance.web (local-exec): TASK [build the project with maven] **
aws_instance.web: Still creating... [1m20s elapsed]
i-0b3ad43d356073807 (ansible-master)
PublicIPs: 43.205.118.79 PrivateIPs: 172.31.37.116
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
4:27 PM
12/14/2023

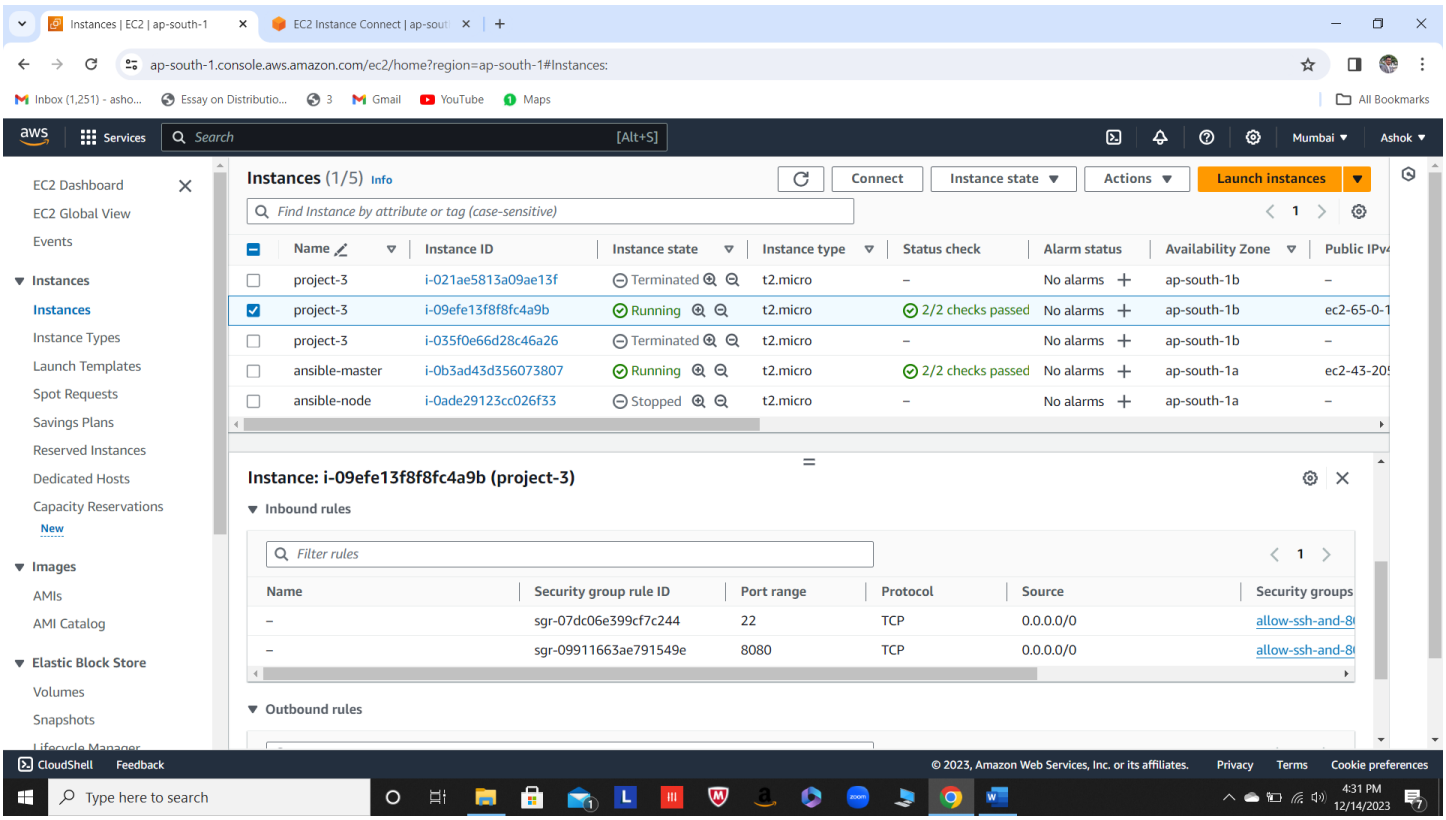


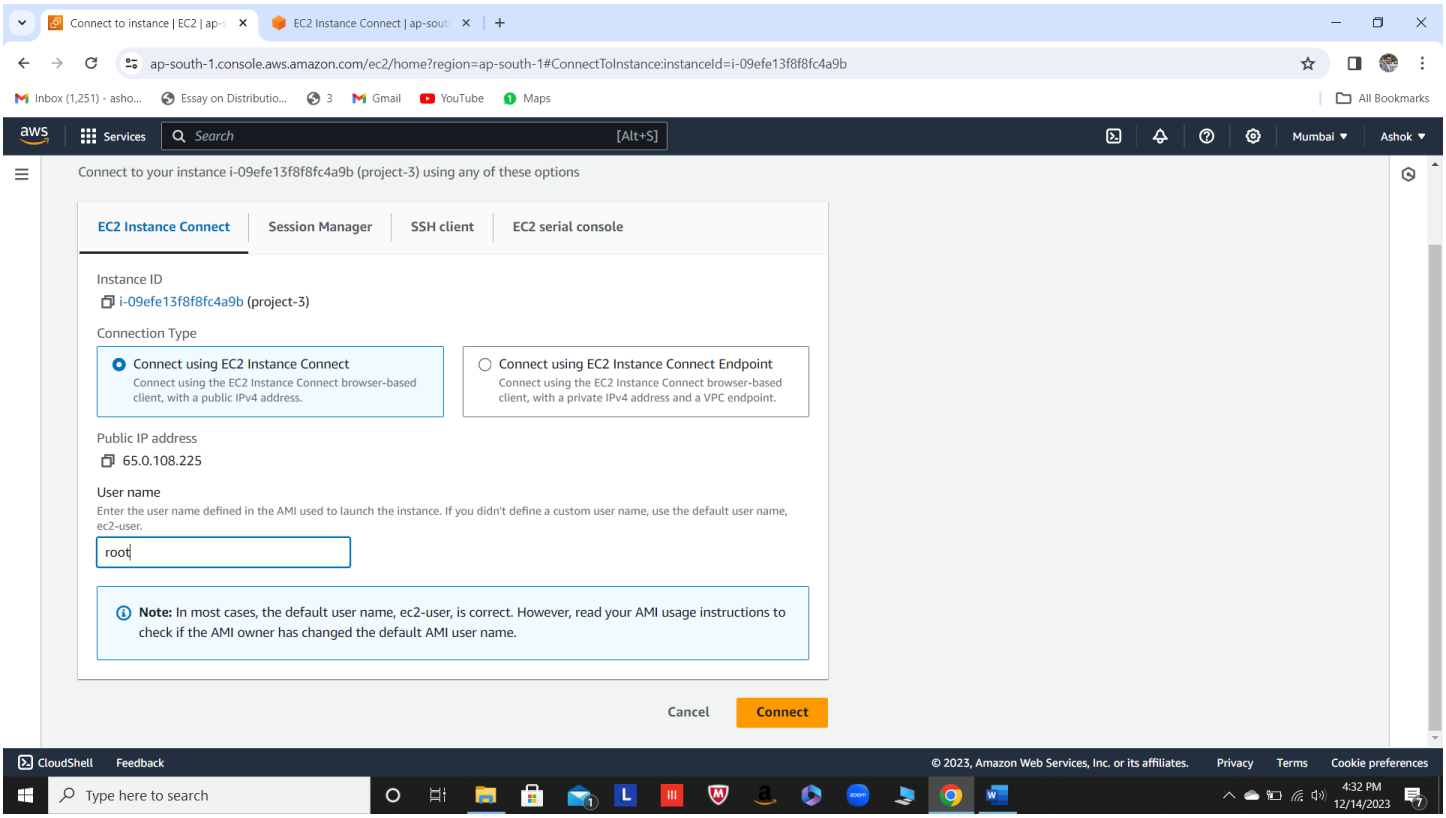
. cat host (Auto generated public ip)

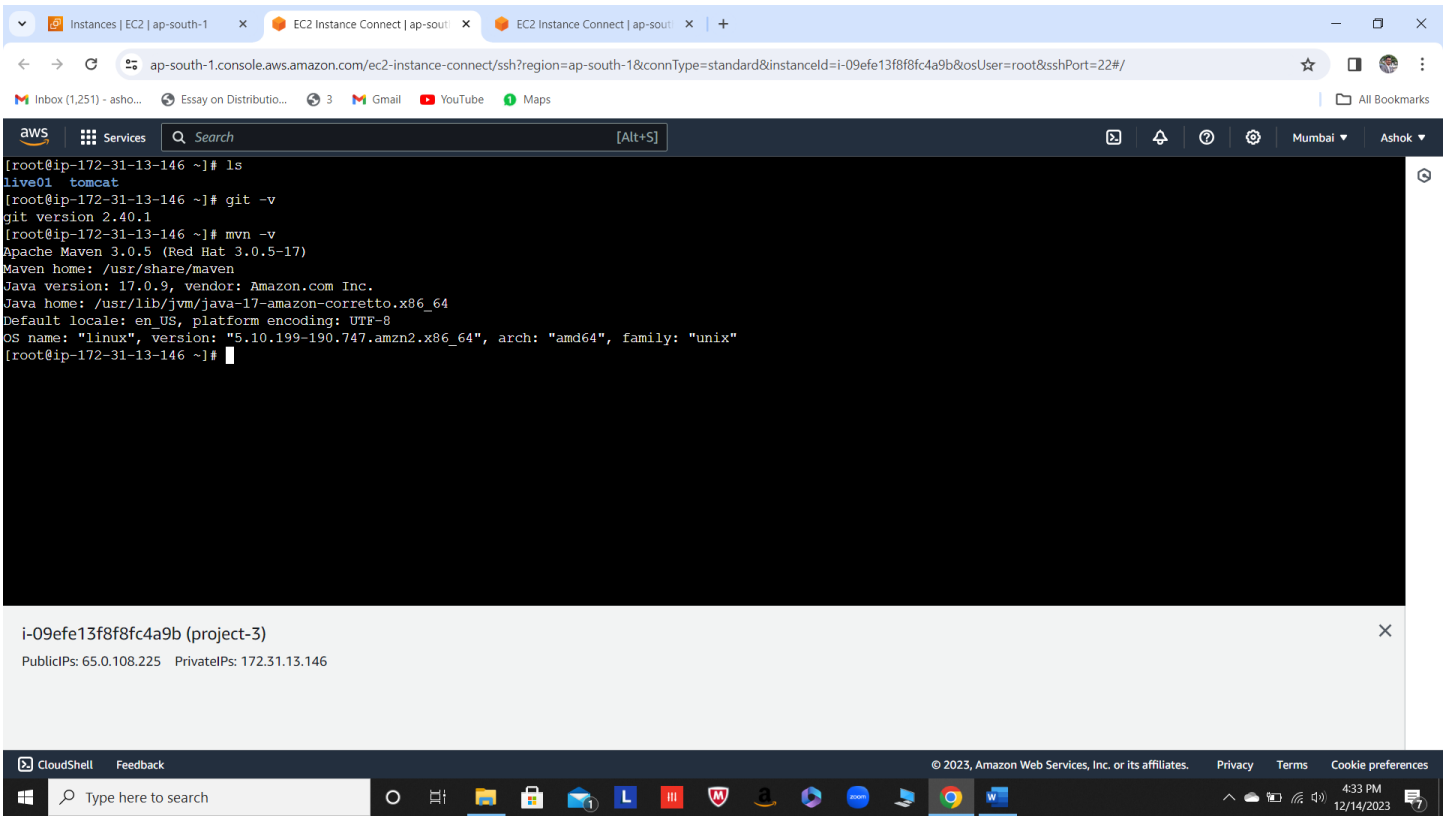


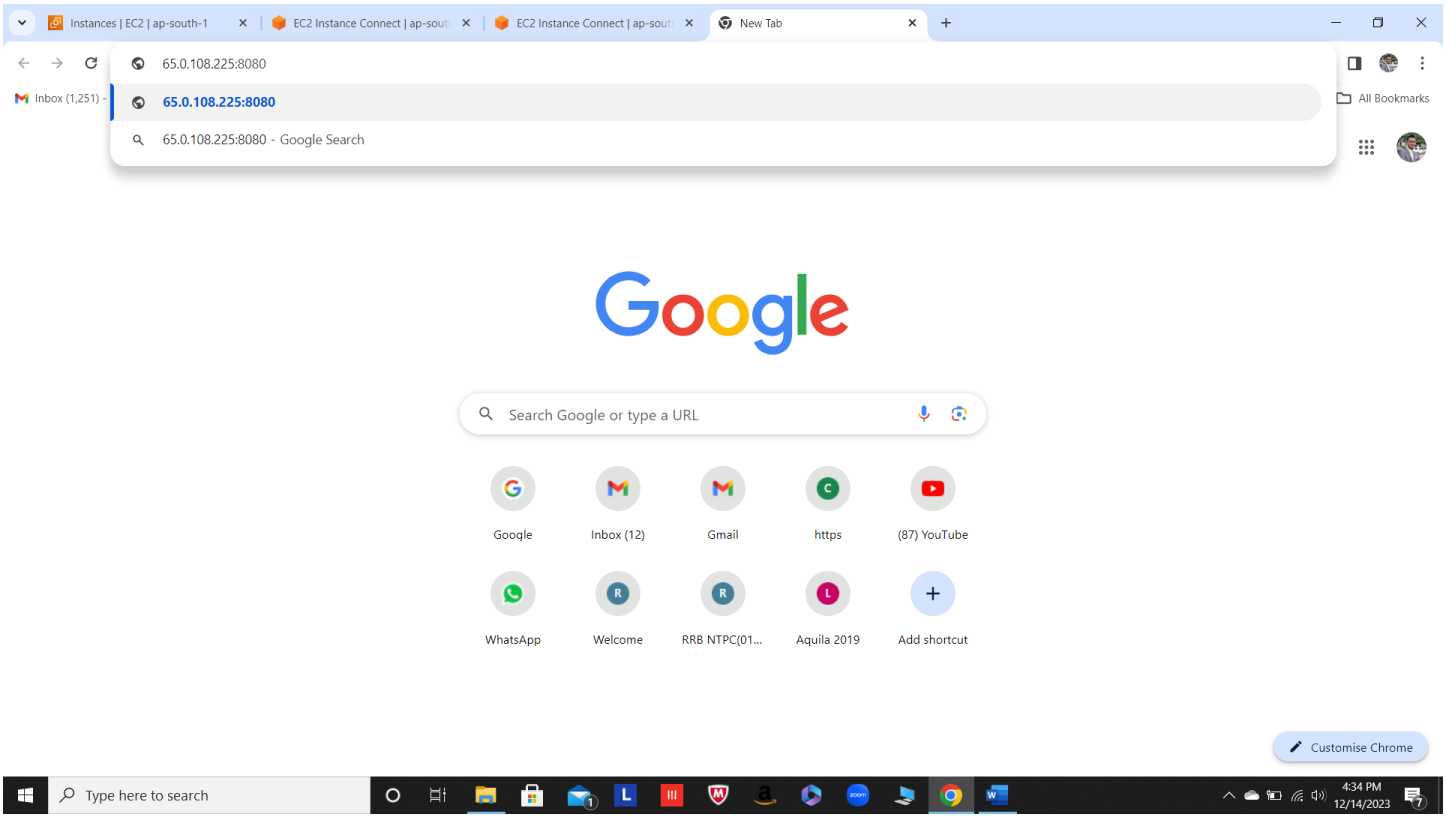
- Now check created instance “project-3” and playbook.yml output

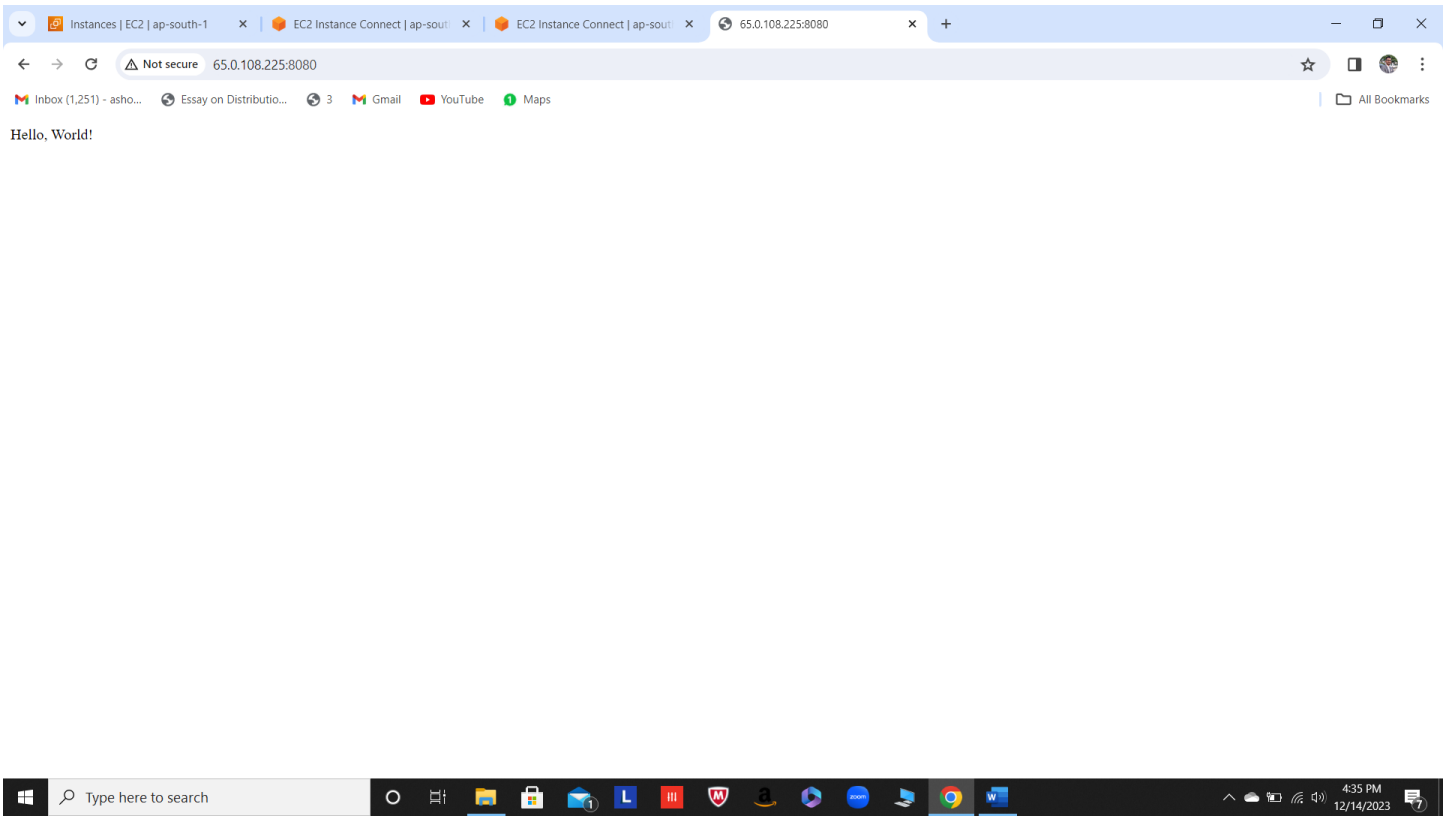





Connect to instance | EC2 | ap-s
EC2 Instance Connect | ap-sout
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-09efe13f8f8fc4a9b
Inbox (1,251) - asho...
Essay on Distributio...
Gmail
YouTube
Maps
All Bookmarks
aws
Services
Search
[Alt+S]
Mumbai
Ashok
Connect to your instance i-09efe13f8f8fc4a9b (project-3) using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-09efe13f8f8fc4a9b (project-3)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IP address
65.0.108.225
User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.
root
Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.
Cancel
Connect
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
4:32 PM
12/14/2023


Instances | EC2 | ap-south-1
EC2 Instance Connect | ap-sout
EC2 Instance Connect | ap-sout
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-09efe13f8f8fc4a9b&osUser=root&sshPort=22#/
Inbox (1,251) - asho...
Essay on Distributio...
Gmail
YouTube
Maps
All Bookmarks
aws
Services
Search
[Alt+S]
Mumbai
Ashok
[root@ip-172-31-13-146 ~]# ls
live01 tomcat
[root@ip-172-31-13-146 ~]# git -v
git version 2.40.1
[root@ip-172-31-13-146 ~]# mvn -v
Apache Maven 3.0.5 (Red Hat 3.0.5-17)
Maven home: /usr/share/maven
Java version: 17.0.9, vendor: Amazon.com Inc.
Java home: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.10.199-190.747.amzn2.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-13-146 ~]#
i-09efe13f8f8fc4a9b (project-3)
PublicIPs: 65.0.108.225 PrivateIPs: 172.31.13.146
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
4:33 PM
12/14/2023


Instances | EC2 | ap-south-1
EC2 Instance Connect | ap-sout
EC2 Instance Connect | ap-sout
New Tab
65.0.108.225:8080
65.0.108.225:8080
65.0.108.225:8080 - Google Search
Inbox (1,251)
All Bookmarks
Google
Search Google or type a URL
Google
Inbox (12)
Gmail
https
(87) YouTube
WhatsApp
Welcome
RRB NTPC(01...
Aquila 2019
Add shortcut
Customise Chrome
Type here to search
4:34 PM
12/14/2023


Instances | EC2 | ap-south-1
EC2 Instance Connect | ap-sout
EC2 Instance Connect | ap-sout
65.0.108.225:8080
Not secure
65.0.108.225:8080
Inbox (1,251) - asho...
Essay on Distributio...
3
Gmail
YouTube
Maps
All Bookmarks
Hello, World!
Type here to search
4:35 PM
12/14/2023