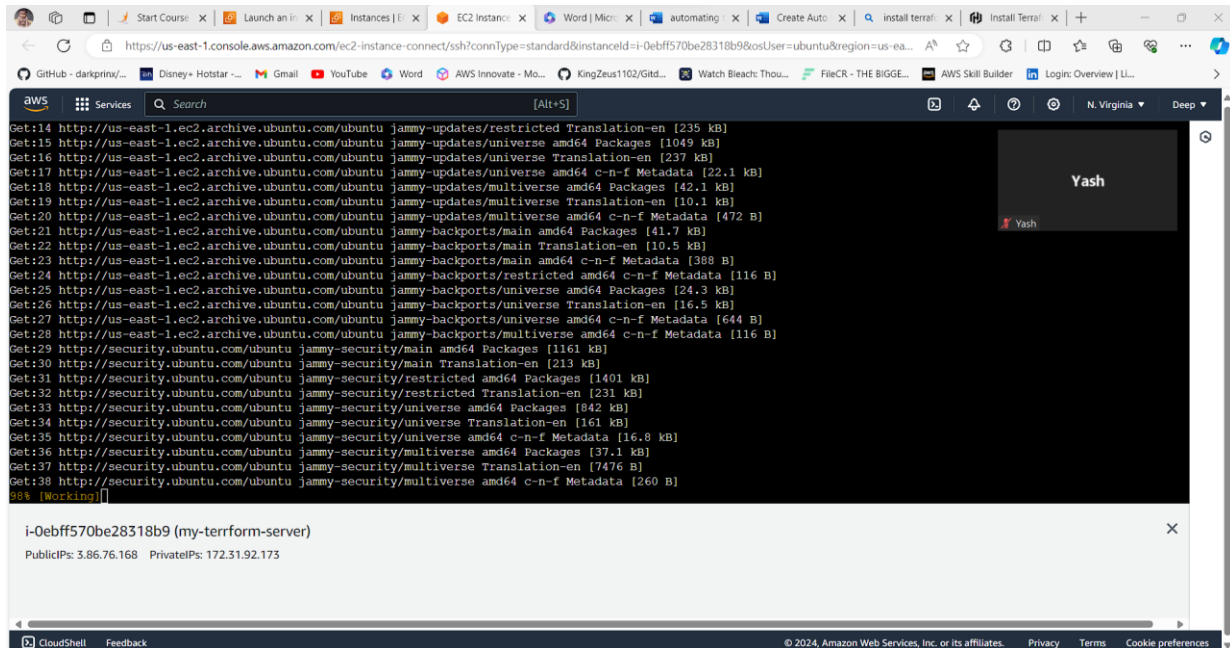


# automating the infrastructure creation and the deployment procedure

1) launch a instance and install terraform from the official portal.

Sudo apt update

The commands can be found here: [Install Terraform](#) | [Terraform](#) | [HashiCorp Developer](#)



The screenshot shows an AWS CloudShell terminal window with the following output for the 'sudo apt update' command:

```
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [235 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1049 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [237 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.1 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41.7 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24.3 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1161 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [213 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1401 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [231 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [842 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [161 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
98% [Working]
```

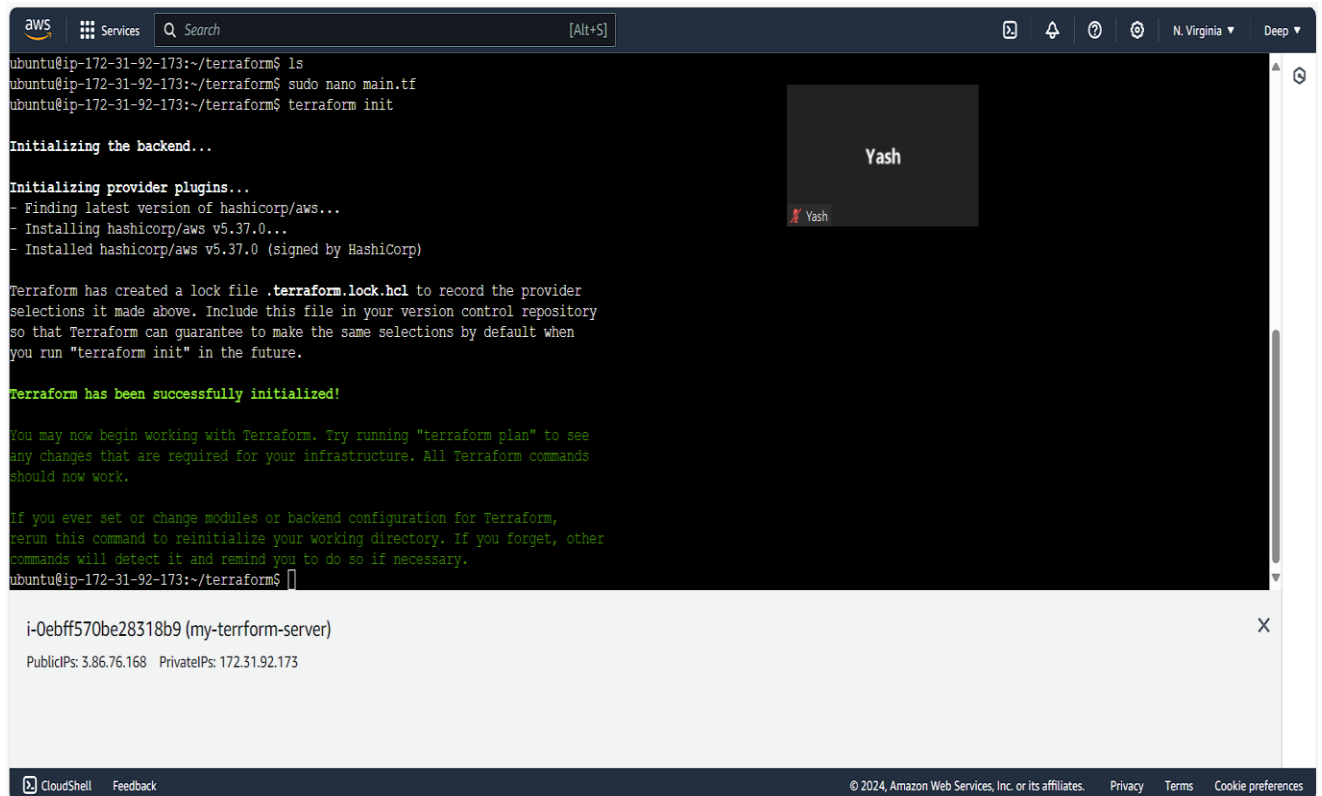
Below the terminal output, a box displays the instance details:

```
i-0ebff570be28318b9 (my-terraform-server)
PublicIPs: 3.86.76.168 PrivateIPs: 172.31.92.173
```

After successful installation, create a working dir using mkdir terraform

Cd terraform

Here we need to initialize terraform. We can do so by creating a main.tf file.



The screenshot shows an AWS CloudShell terminal window. The terminal output shows the user running `ls`, `sudo nano main.tf`, and `terraform init`. The output indicates that Terraform has been successfully initialized, creating a lock file `.terraform.lock.hcl` and installing the `hashicorp/aws` provider. A video call window with a participant named 'Yash' is visible in the background. Below the terminal, a metadata box shows the instance ID `i-0ebff570be28318b9` (my-terraform-server) and its public and private IP addresses.

```
aws Services Search [Alt+S]
ubuntu@ip-172-31-92-173:~/terraform$ ls
ubuntu@ip-172-31-92-173:~/terraform$ sudo nano main.tf
ubuntu@ip-172-31-92-173:~/terraform$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.37.0...
- Installed hashicorp/aws v5.37.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

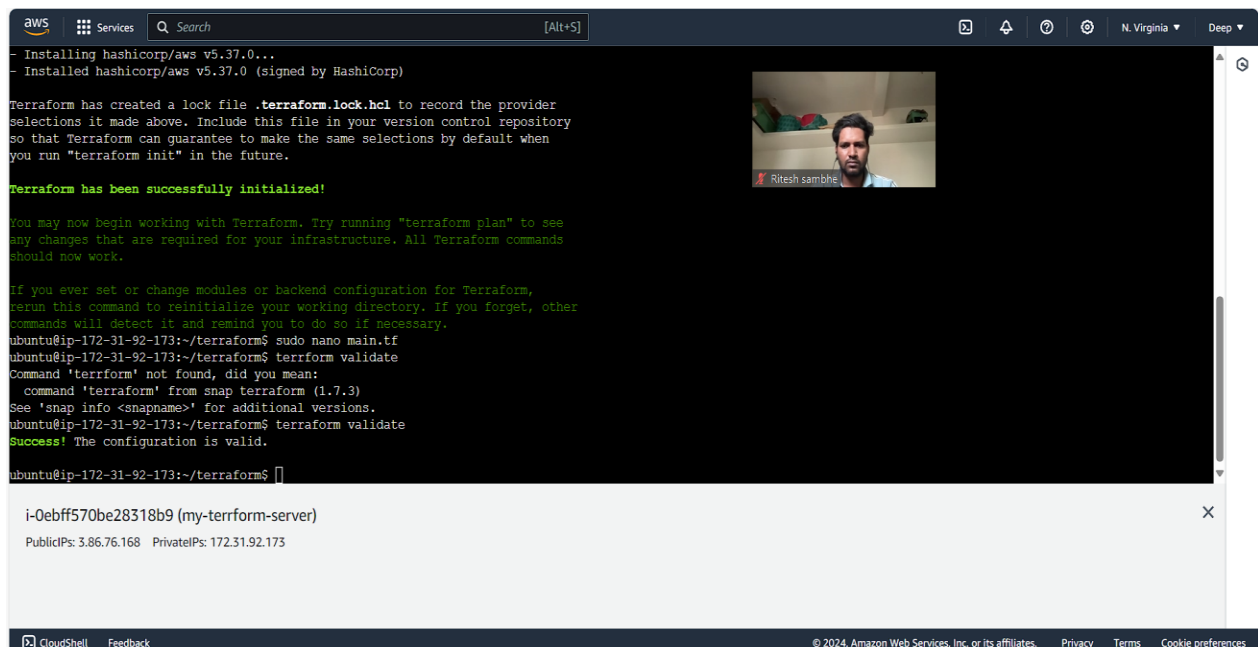
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-92-173:~/terraform$
```

i-0ebff570be28318b9 (my-terraform-server)  
PublicIPs: 3.86.76.168 PrivateIPs: 172.31.92.173

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Validate the created script using **terraform validate**.



The screenshot shows the same AWS CloudShell terminal window. The user has run `terraform validate`, which outputs a success message: `Success! The configuration is valid.` A video call window with a participant named 'Ritesh sambhe' is visible in the background. The metadata box at the bottom remains the same.

```
aws Services Search [Alt+S]
- Installing hashicorp/aws v5.37.0...
- Installed hashicorp/aws v5.37.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

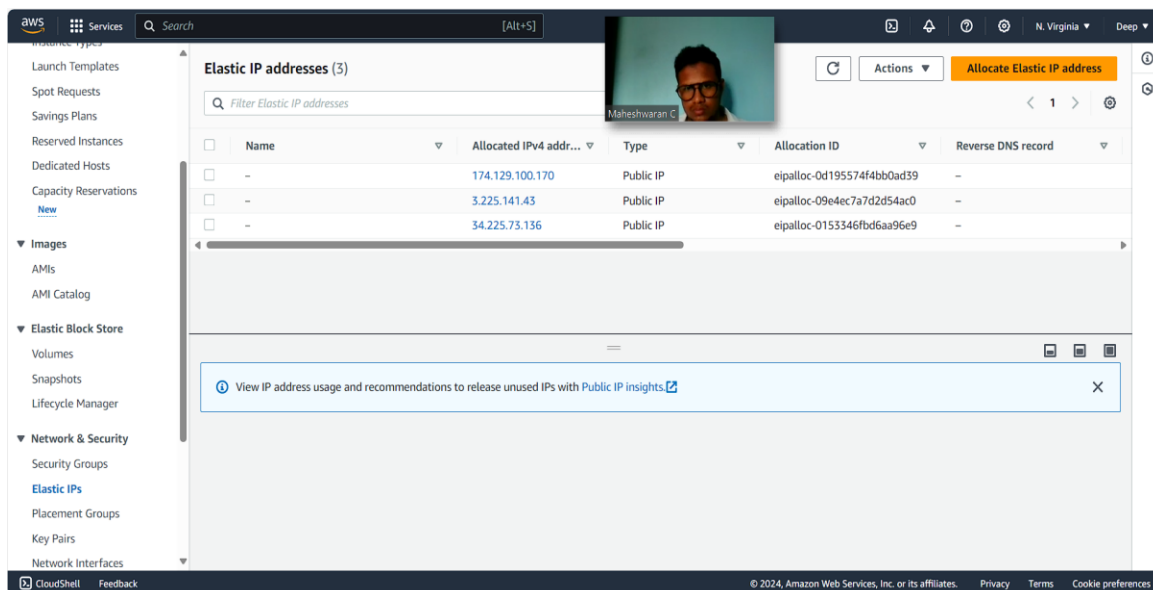
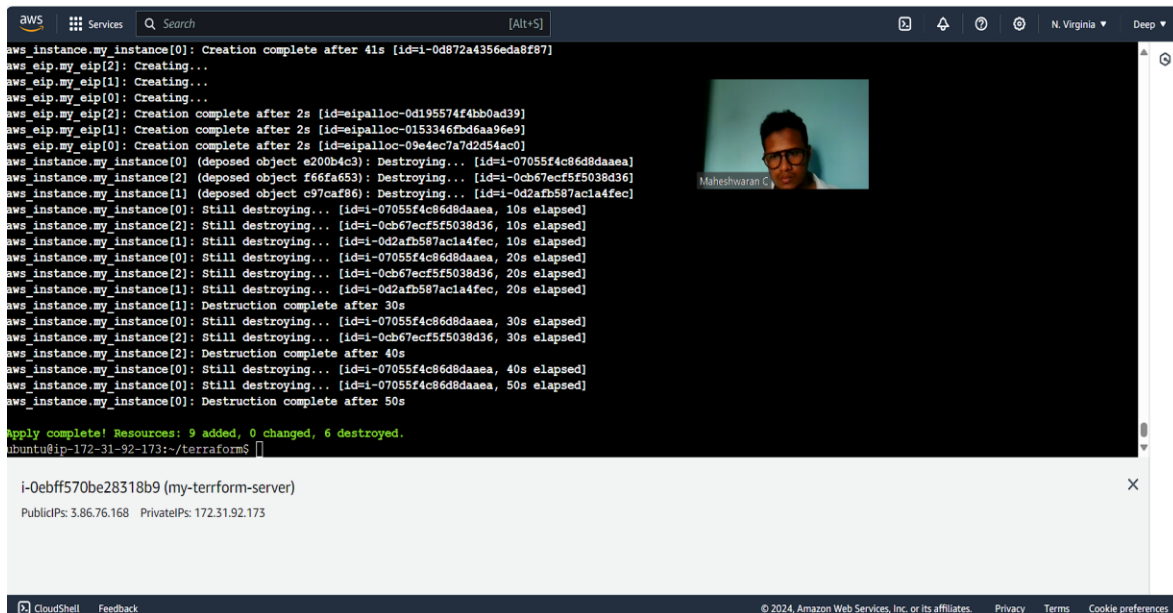
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-92-173:~/terraform$ sudo nano main.tf
ubuntu@ip-172-31-92-173:~/terraform$ terraform validate
Command 'terraform' not found, did you mean:
  command 'terraform' from snap terraform (1.7.3)
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-92-173:~/terraform$ terraform validate
Success! The configuration is valid.
ubuntu@ip-172-31-92-173:~/terraform$
```

i-0ebff570be28318b9 (my-terraform-server)  
PublicIPs: 3.86.76.168 PrivateIPs: 172.31.92.173

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Do a **terraform plan** and **terraform apply** to execute the changes.

Post approval the resources will be created as in the script.



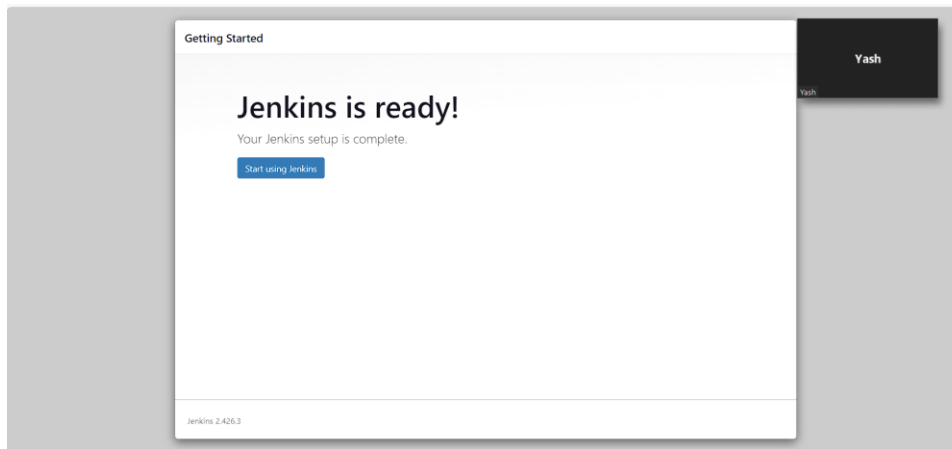
Install Jenkins and ansible on one the machines -

For Jenkins, java is a pre-requisite, do a `sudo apt install openjdk-11-jdk -y`

Then follow the ansible installation guide.

After successful installation confirm the version,

Jenkins Dashboard setup:



Establish keyless connection between ansible master and slaves:

Ansible -m ping slave\_hosts

```
aws Services Search [Alt+S] N. Virginia Deep
[slave_hosts]
3.225.141.43
174.129.100.170

ubuntu@ip-10-0-1-108:~$ ansible -m ping slave_hosts
The authenticity of host '3.225.141.43 (3.225.141.43)' can't be established.
ED25519 key fingerprint is SHA256:JUKdgHlxN0+1BFBheYLI9WlXk/bqpus7DIIYLRCAE.
This key is not known by any other names
The authenticity of host '174.129.100.170 (174.129.100.170)' can't be established.
ED25519 key fingerprint is SHA256:QgRtTA4Mey7u4DVodyJ5sxtPYIcqD0/sCFyyZsJNio.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
3.225.141.43 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

For jenkins master-agent configuration

Do a ssh-keygen on each of the agents and copy id\_rsa pvt key in to credential of the jenkins.

This will be Later used by jenkins to authenticate and execute the pipeline

Jenkins

Search (CTRL+K)

Deepak R

log out

Dashboard

Manage Jenkins

Credentials

System

Global credentials (unrestricted)

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
agent2-key	ubuntu	SSH Username with private key	
agent1-key	ubuntu	SSH Username with private key	

Icons: S M L

REST API

Jenkins 2.426.3

Setup pipeline:

Dashboard

test

Configuration

Configure

General

Advanced Project Options

Pipeline

Pipeline

Definition

Pipeline script

Script

```
6-  steps {
7-      script {
8-          // Clone the Github repository
9-          git branch: 'main', url: 'https://github.com/your_username/your_repository.git'
10-      }
11-  }
12-
13-  stage('Execute script') {
14-      steps {
15-          // Replace 'your_script.sh' with the name of your script
16-          sh './ansible.sh'
17-      }
18-  }
19-
20-  }
21-
22- }
```

try sample Pipeline...

☒ Use Groovy Sandbox

Save

Apply