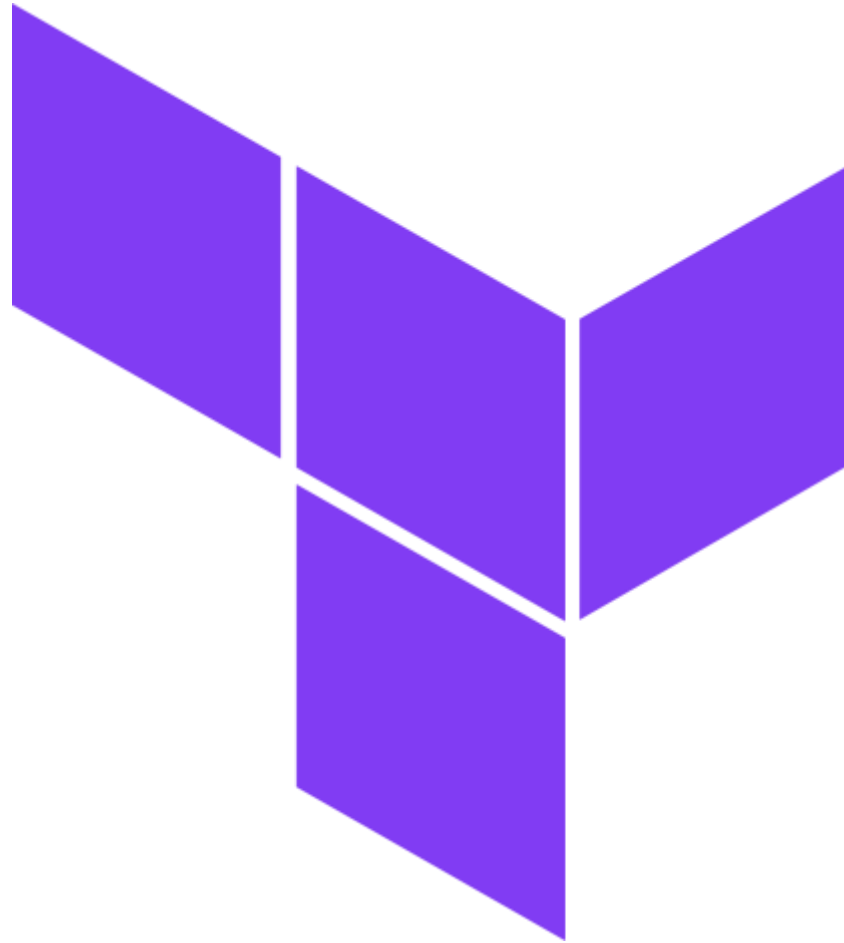


# **Module 8: Terraform Assignment**

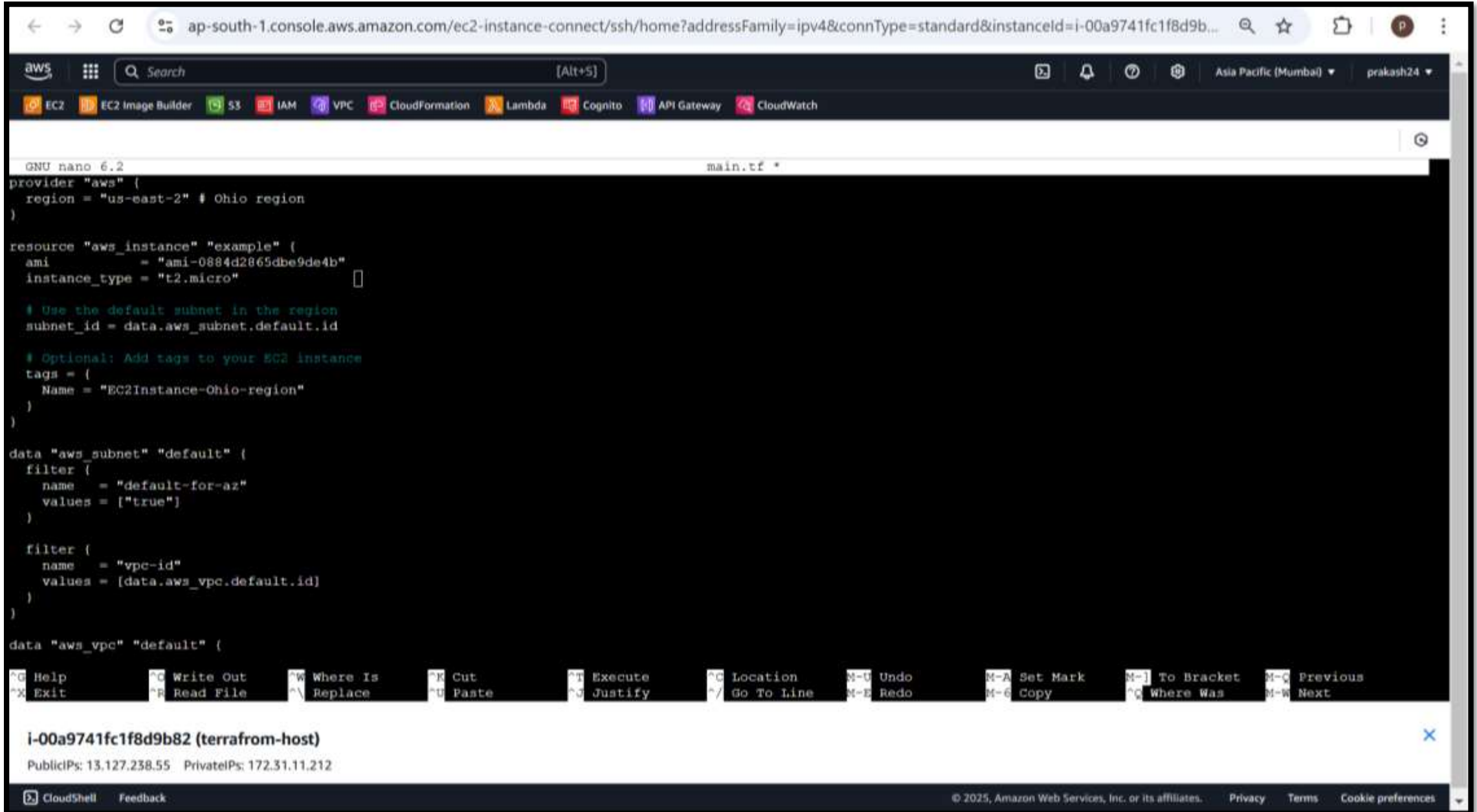


# Module 8: Terraform Assignment -1

## **Tasks To Be Performed:**

1. Create an EC2 service in the default subnet in the Ohio region

## Terraform Script main.tf file



The screenshot shows the AWS Management Console interface with a CloudShell session open. The browser address bar shows the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1f8d9b...`. The AWS navigation bar at the top includes links to EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user is logged in as 'prakash24' in the 'Asia Pacific (Mumbai)' region.

The CloudShell terminal window displays the GNU nano 6.2 editor editing a file named `main.tf`. The Terraform script content is as follows:

```
provider "aws" {
  region = "us-east-2" # Ohio region
}

resource "aws_instance" "example" {
  ami           = "ami-0884d2865dbe9de4b"
  instance_type = "t2.micro"

  # Use the default subnet in the region
  subnet_id = data.aws_subnet.default.id

  # Optional: Add tags to your EC2 instance
  tags = {
    Name = "EC2Instance-Ohio-region"
  }
}

data "aws_subnet" "default" {
  filter {
    name   = "default-for-az"
    values = ["true"]
  }

  filter {
    name   = "vpc-id"
    values = [data.aws_vpc.default.id]
  }
}

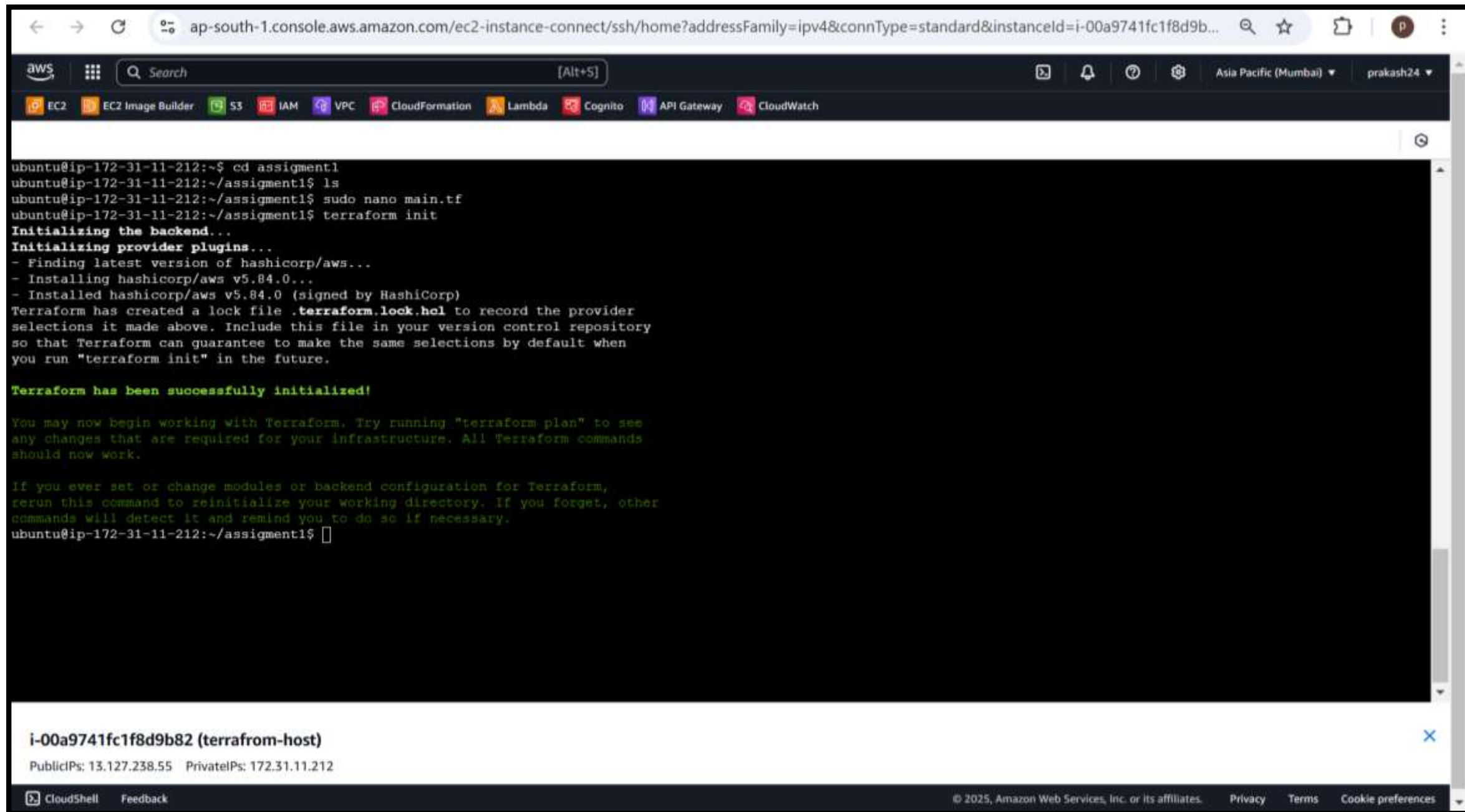
data "aws_vpc" "default" {
```

Below the editor, a table of nano editor shortcuts is displayed:

<code>^G</code> Help	<code>^O</code> Write Out	<code>^W</code> Where Is	<code>^K</code> Cut	<code>^T</code> Execute	<code>^C</code> Location	<code>M-U</code> Undo	<code>M-A</code> Set Mark	<code>M-I</code> To Bracket	<code>M-C</code> Previous
<code>^X</code> Exit	<code>^R</code> Read File	<code>^\</code> Replace	<code>^U</code> Paste	<code>^J</code> Justify	<code>^/_</code> Go To Line	<code>M-E</code> Redo	<code>M-6</code> Copy	<code>^G</code> Where Was	<code>M-W</code> Next

At the bottom of the terminal, the instance ID `i-00a9741fc1f8d9b82` is shown with the label `(terraform-host)`. Below this, the IP addresses are listed: `PublicIPs: 13.127.238.55` and `PrivateIPs: 172.31.11.212`. The footer of the console shows 'CloudShell', a 'Feedback' link, and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

# Running Terraform init Command



The screenshot shows an AWS CloudShell terminal window. The browser address bar at the top displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1f8d9b...`. The AWS navigation bar includes the AWS logo, a search bar, and various service icons (EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, CloudWatch). The user's location is set to 'Asia Pacific (Mumbai)' and the username is 'prakash24'.

The terminal output shows the following commands and their results:

```
ubuntu@ip-172-31-11-212:~$ cd assignment1
ubuntu@ip-172-31-11-212:~/assignment1$ ls
ubuntu@ip-172-31-11-212:~/assignment1$ sudo nano main.tf
ubuntu@ip-172-31-11-212:~/assignment1$ terraform init
```

**Initializing the backend...**

**Initializing provider plugins...**

- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (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.

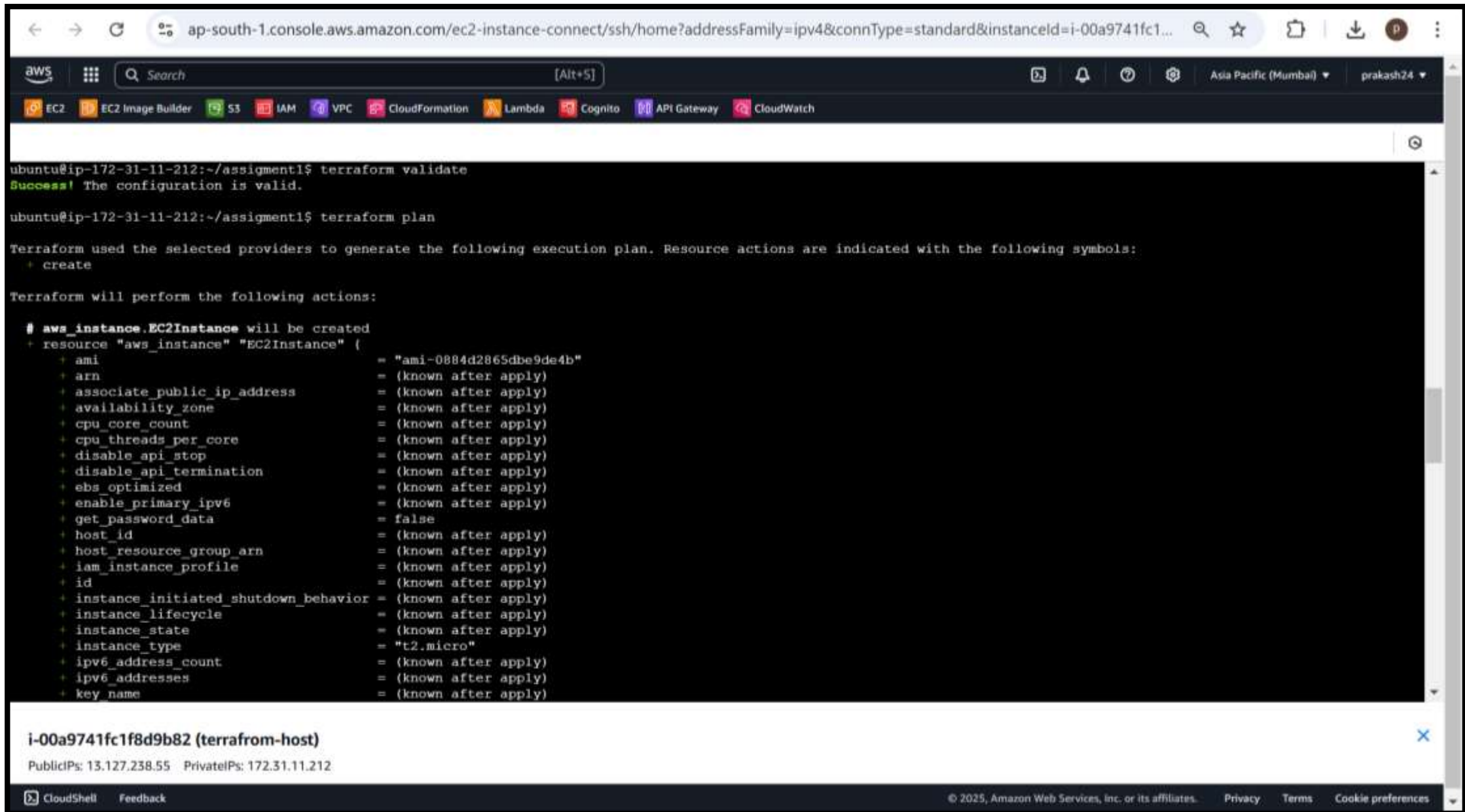
If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

```
ubuntu@ip-172-31-11-212:~/assignment1$
```

At the bottom of the terminal window, the instance ID `i-00a9741fc1f8d9b82` is listed with the label '(terraform-host)'. Below this, the public and private IP addresses are shown: `PublicIPs: 13.127.238.55` and `PrivateIPs: 172.31.11.212`.

The footer of the CloudShell window includes 'CloudShell' and 'Feedback' links on the left, and copyright information '© 2025, Amazon Web Services, Inc. or its affiliates.' along with 'Privacy', 'Terms', and 'Cookie preferences' links on the right.

## Running Terraform validate & Terraform plan Command



The screenshot shows the AWS Management Console interface. At the top, the browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1...`. The console header includes the AWS logo, a search bar, and navigation links for various services: EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user's location is set to Asia Pacific (Mumbai) and the username is prakash24.

The main content area displays a terminal window with the following commands and output:

```
ubuntu@ip-172-31-11-212:~/assignment1$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-11-212:~/assignment1$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

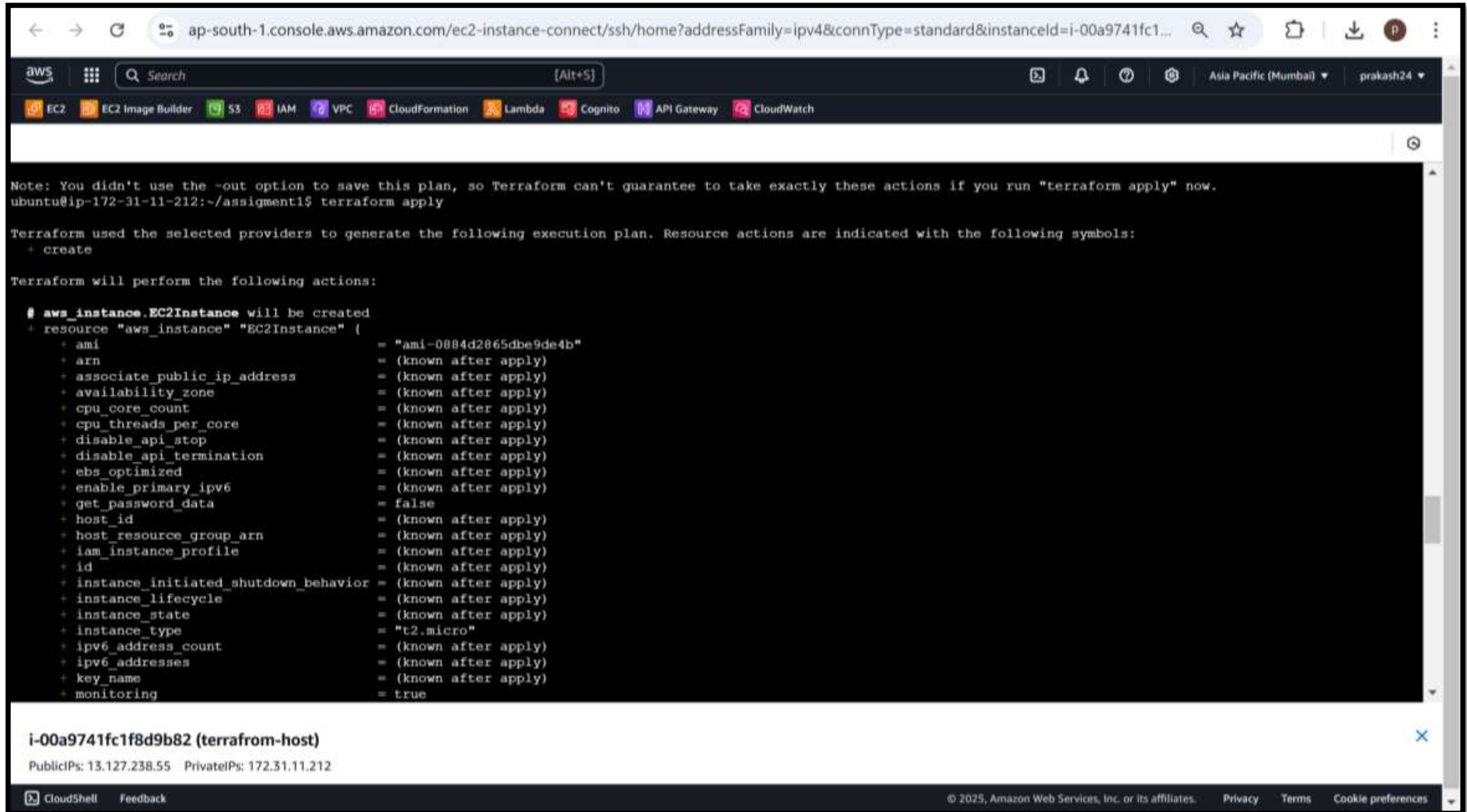
Terraform will perform the following actions:

# aws_instance.EC2Instance will be created
+ resource "aws_instance" "EC2Instance" {
  + ami                    = "ami-0884d2865dbe9de4b"
  + arn                    = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone       = (known after apply)
  + cpu_core_count          = (known after apply)
  + cpu_threads_per_core    = (known after apply)
  + disable_api_stop        = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized           = (known after apply)
  + enable_primary_ipv6     = (known after apply)
  + get_password_data       = false
  + host_id                 = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile    = (known after apply)
  + id                      = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle      = (known after apply)
  + instance_state          = (known after apply)
  + instance_type           = "t2.micro"
  + ipv6_address_count       = (known after apply)
  + ipv6_addresses          = (known after apply)
  + key_name                = (known after apply)
```

Below the terminal output, a notification box for the instance `i-00a9741fc1f8d9b82 (terraform-host)` is visible, showing Public IPs: 13.127.238.55 and Private IPs: 172.31.11.212.

The footer of the console includes links for CloudShell, Feedback, and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for Privacy, Terms, and Cookie preferences.

# Running Terraform apply Command



The screenshot shows the AWS Management Console interface. At the top, the browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1...`. The console header includes the AWS logo, a search bar, and navigation links for various services like EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user's location is set to Asia Pacific (Mumbai) and the user is logged in as prakash24.

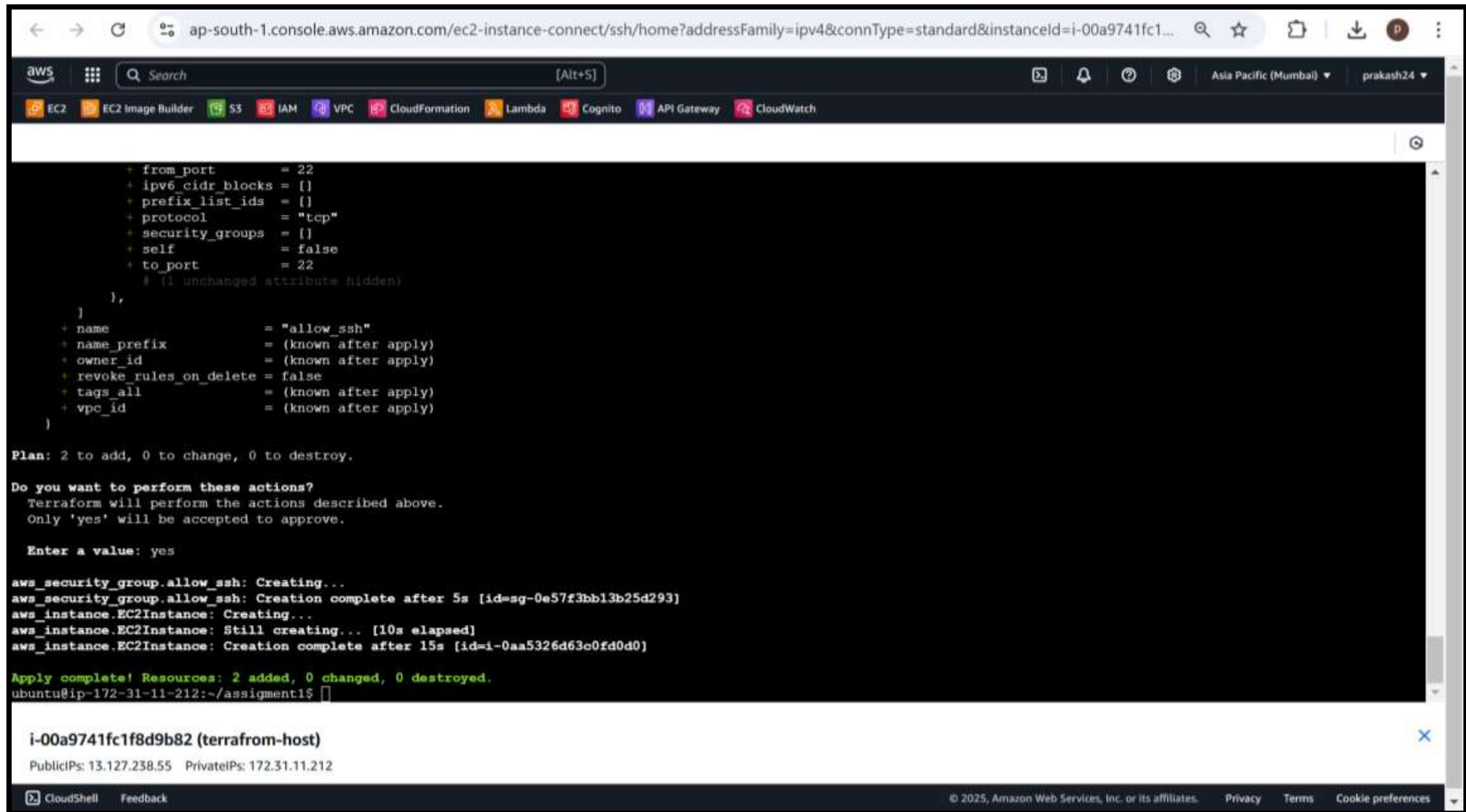
The main content area displays a terminal session. The first line is a note: "Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run 'terraform apply' now." The user enters the command `ubuntu@ip-172-31-11-212:~/assignment1$ terraform apply`. The output shows the execution plan for creating an EC2 instance. The plan indicates that the resource `aws_instance.EC2Instance` will be created. The following table lists the attributes and their values:

Attribute	Value
<code>ami</code>	<code>"ami-0884d2865dbe9de4b"</code>
<code>arn</code>	<code>"(known after apply)"</code>
<code>associate_public_ip_address</code>	<code>"(known after apply)"</code>
<code>availability_zone</code>	<code>"(known after apply)"</code>
<code>cpu_core_count</code>	<code>"(known after apply)"</code>
<code>cpu_threads_per_core</code>	<code>"(known after apply)"</code>
<code>disable_api_stop</code>	<code>"(known after apply)"</code>
<code>disable_api_termination</code>	<code>"(known after apply)"</code>
<code>ebs_optimized</code>	<code>"(known after apply)"</code>
<code>enable_primary_ipv6</code>	<code>"(known after apply)"</code>
<code>get_password_data</code>	<code>false</code>
<code>host_id</code>	<code>"(known after apply)"</code>
<code>host_resource_group_arn</code>	<code>"(known after apply)"</code>
<code>iam_instance_profile</code>	<code>"(known after apply)"</code>
<code>id</code>	<code>"(known after apply)"</code>
<code>instance_initiated_shutdown_behavior</code>	<code>"(known after apply)"</code>
<code>instance_lifecycle</code>	<code>"(known after apply)"</code>
<code>instance_state</code>	<code>"(known after apply)"</code>
<code>instance_type</code>	<code>"t2.micro"</code>
<code>ipv6_address_count</code>	<code>"(known after apply)"</code>
<code>ipv6_addresses</code>	<code>"(known after apply)"</code>
<code>key_name</code>	<code>"(known after apply)"</code>
<code>monitoring</code>	<code>true</code>

Below the terminal output, a summary box for the instance `i-00a9741fc1f8d9b82 (terraform-host)` is shown. It lists the public IP as `13.127.238.55` and the private IP as `172.31.11.212`. The bottom of the console shows the CloudShell logo and a feedback link.



## Terraform deployment done



The screenshot shows the AWS CloudShell interface. At the top, the browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1...`. The AWS navigation bar includes the AWS logo, a search bar, and various service icons (EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, CloudWatch). The user's location is set to Asia Pacific (Mumbai) and the account is prakash24.

The main terminal area displays the following Terraform configuration and output:

```
+ from_port      = 22
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol       = "tcp"
+ security_groups = []
+ self           = false
+ to_port        = 22
# (1 unchanged attribute hidden)
},
]
+ name           = "allow_ssh"
+ name_prefix    = (known after apply)
+ owner_id       = (known after apply)
+ revoke_rules_on_delete = false
+ tags_all       = (known after apply)
+ vpc_id         = (known after apply)
}
```

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?  
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.

Enter a value: yes

aws\_security\_group.allow\_ssh: Creating...  
aws\_security\_group.allow\_ssh: Creation complete after 5s [id=sg-0e57f3bb13b25d293]  
aws\_instance.EC2Instance: Creating...  
aws\_instance.EC2Instance: Still creating... [10s elapsed]  
aws\_instance.EC2Instance: Creation complete after 15s [id=i-0aa5326d63c0fd0d0]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

ubuntu@ip-172-31-11-212:~/assignment1\$

A notification banner at the bottom indicates the instance `i-00a9741fc1f8d9b82 (terraform-host)` is ready, with Public IPs: 13.127.238.55 and Private IPs: 172.31.11.212.

The footer includes the CloudShell logo, a feedback link, and copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

## Verifying the EC2 instance launched by terraform

The screenshot displays the AWS Management Console for the us-east-2 region. The left sidebar contains navigation links for Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store. The main content area shows the 'Instances (1/1)' page with a table listing the instance 'EC2Instance-Ohio-region' with ID 'i-0aa5326d63c0fd0d0', which is in a 'Running' state. Below the table, the 'Details' tab for this instance is active, showing a summary of its configuration including the instance ID, public and private IP addresses, and DNS names.

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:v=3;\$case=tags:true%5C,client:false;\$regex=tags:false%5C,client...

aws [Search] [Alt+S] United States (Ohio) prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

Instances (1/1) Info Last updated 1 minute ago [Connect] [Instance state] [Actions] [Launch instances]

Find Instance by attribute or tag (case-sensitive) All states

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avi
<input checked="" type="checkbox"/>	EC2Instance-Ohio-region	i-0aa5326d63c0fd0d0	Running	t2.micro	Initializing	View alarms +	us-

i-0aa5326d63c0fd0d0 (EC2Instance-Ohio-region)

Details Status and alarms Monitoring Security Networking Storage Tags

▼ Instance summary Info

Instance ID i-0aa5326d63c0fd0d0	Public IPv4 address 18.118.9.203   open address	Private IPv4 addresses 172.31.5.17
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-18-118-9-203.us-east-2.compute.amazonaws.com   open address
Hostname type	Private IP DNS name (IPv4 only)	

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

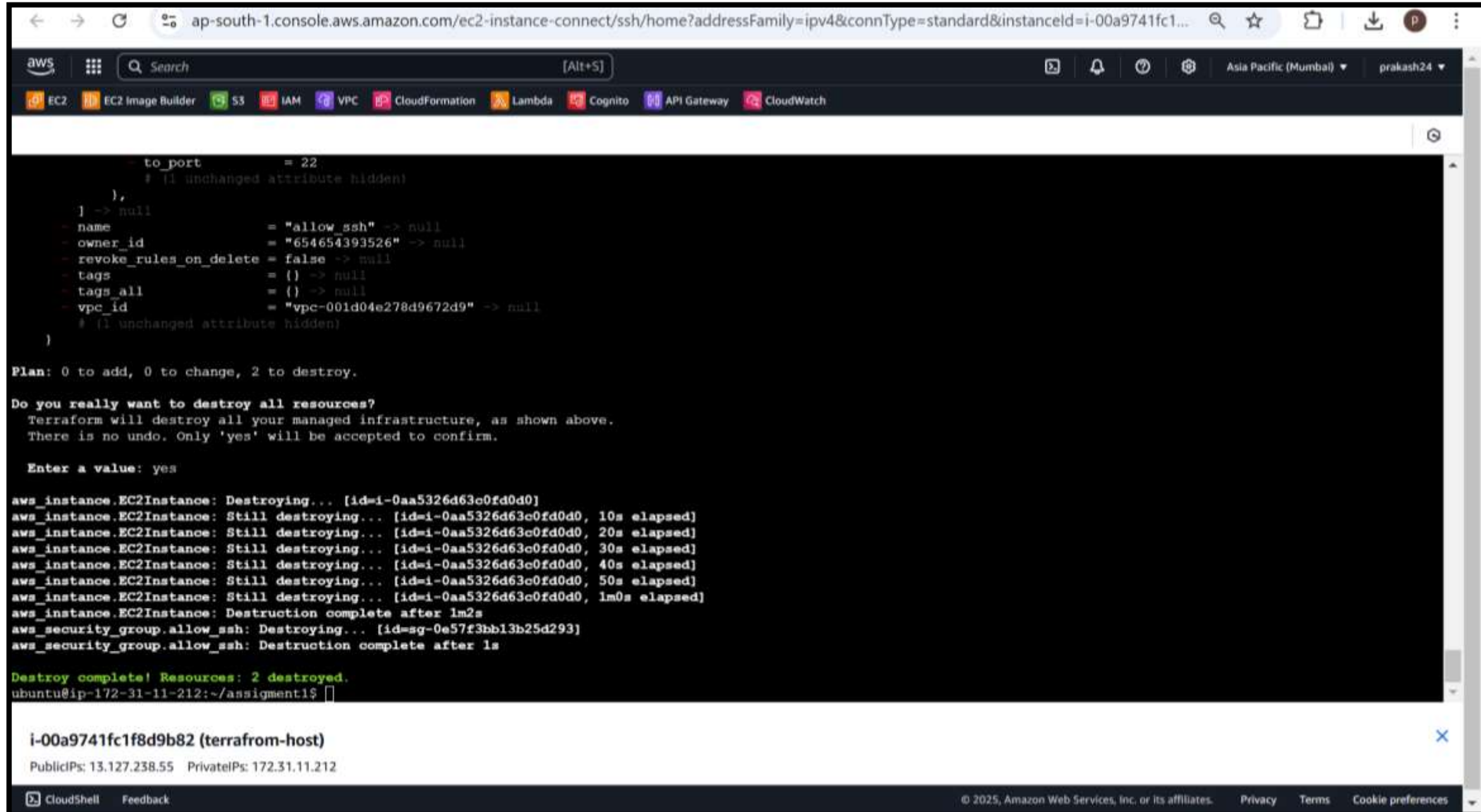


# Module 8: Terraform Assignment -2

## **Tasks To Be Performed:**

1. Destroy the previous deployment
2. Create a new EC2 instance with an Elastic IP

## Destroying EC2 instance launched by terraform using terraform destroy command



The screenshot shows the AWS Management Console interface with a terminal window open. The terminal displays the output of a Terraform destroy command. It lists resources to be destroyed, including an EC2 instance and a security group. The output shows the progress of destruction, with timestamps indicating the time taken for each resource. The terminal also shows the user's prompt and the command being executed.

```
to_port      = 22
# (1 unchanged attribute hidden)
},
] -> null
- name          = "allow_ssh" -> null
- owner_id      = "654654393526" -> null
- revoke_rules_on_delete = false -> null
- tags          = {} -> null
- tags_all      = {} -> null
- vpc_id        = "vpc-001d04e278d9672d9" -> null
# (1 unchanged attribute hidden)
}

Plan: 0 to add, 0 to change, 2 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_instance.EC2Instance: Destroying... [id=i-0aa5326d63c0fd0d0]
aws_instance.EC2Instance: Still destroying... [id=i-0aa5326d63c0fd0d0, 10s elapsed]
aws_instance.EC2Instance: Still destroying... [id=i-0aa5326d63c0fd0d0, 20s elapsed]
aws_instance.EC2Instance: Still destroying... [id=i-0aa5326d63c0fd0d0, 30s elapsed]
aws_instance.EC2Instance: Still destroying... [id=i-0aa5326d63c0fd0d0, 40s elapsed]
aws_instance.EC2Instance: Still destroying... [id=i-0aa5326d63c0fd0d0, 50s elapsed]
aws_instance.EC2Instance: Still destroying... [id=i-0aa5326d63c0fd0d0, 1m0s elapsed]
aws_instance.EC2Instance: Destruction complete after 1m2s
aws_security_group.allow_ssh: Destroying... [id=sg-0e57f3bb13b25d293]
aws_security_group.allow_ssh: Destruction complete after 1s

Destroy complete! Resources: 2 destroyed.
ubuntu@ip-172-31-11-212:~/assignment1$
```

i-00a9741fc1f8d9b82 (terraform-host)

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Verifying the EC2 instance terminated by terraform

←→↻

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#instances:v=3;\$case=tags:true%5C,client:false;\$regex=tags:false%5C,client...

🔍☆📄⬇️👤

aws

Search [Alt+S]

📺🔔🔗⚙️United States (Ohio)prakash24

EC2EC2 Image BuilderS3IAMVPCCloudFormationLambdaCognitoAPI GatewayCloudWatch

☰

Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Instances (1/1) Info

Last updated less than a minute ago

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Find Instance by attribute or tag (case-sensitive)

All states ▾

< 1 >

🔍🔍

✓	Name 🔗	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Avi
✓	EC2Instance-Ohio-region	i-0aa5326d63c0fd0d0	Terminated 🔍🔍	t2.micro	-	View alarms +	us-

i-0aa5326d63c0fd0d0 (EC2Instance-Ohio-region)

⚙️ ▾

Availability zone

us-east-2a

Carrier IP addresses (ephemeral)

-

Outpost ID

-

Use RBN as guest OS hostname

Disabled

▼ Network Interfaces (0) Info

Filter network interfaces

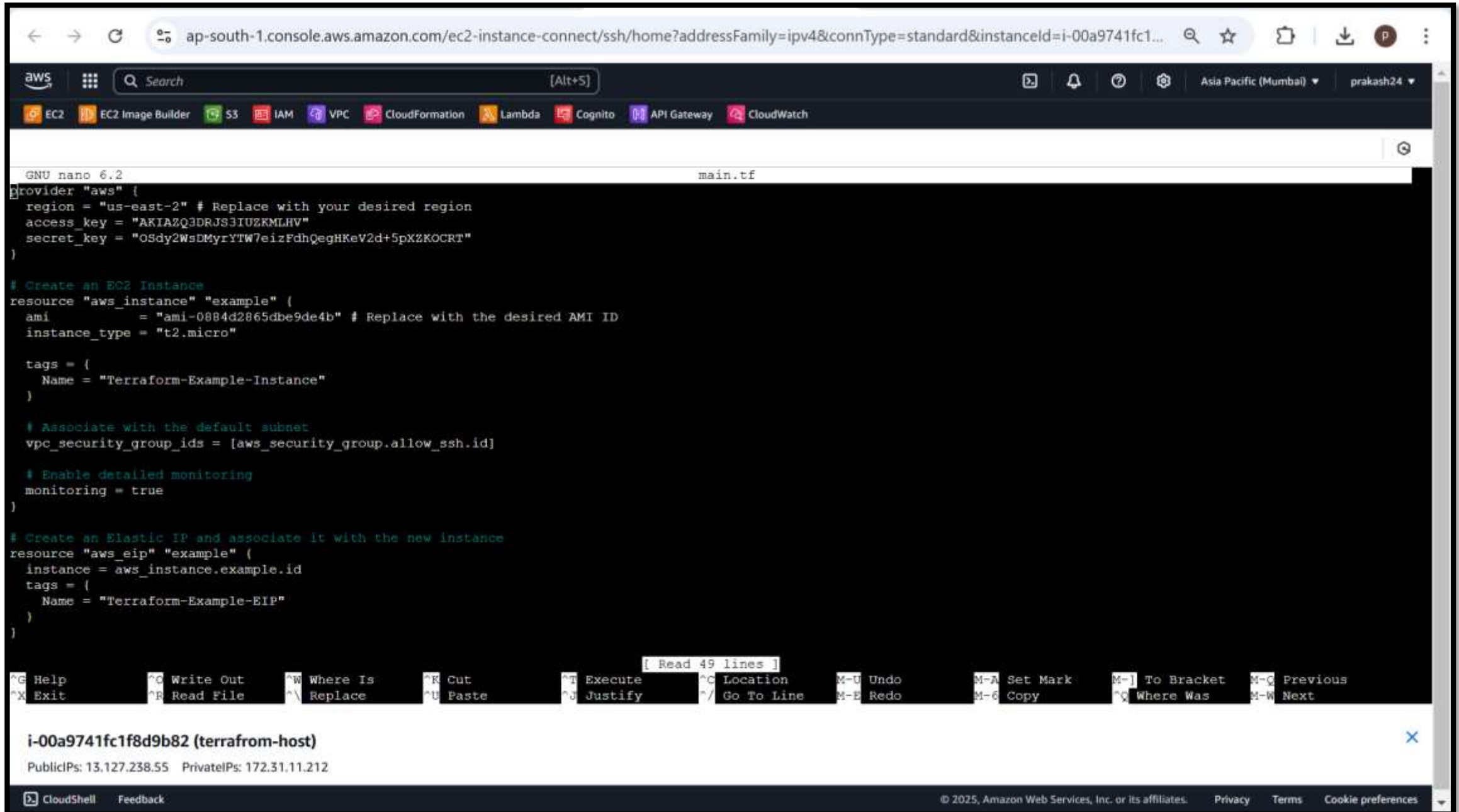
Interface ID	Device index	Card index	Description	Public IPv4 address	Private IPv4 address	Privat
--------------	--------------	------------	-------------	---------------------	----------------------	--------

No network interfaces attached to this instance

CloudShellFeedback

© 2025, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

## Terraform Script main.tf file



The screenshot shows the AWS Management Console interface. At the top, the browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1...`. The console header includes the AWS logo, a search bar, and navigation icons for various services like EC2, IAM, VPC, etc. The main content area is a CloudShell terminal window titled `main.tf`. The terminal shows the GNU nano 6.2 editor with a Terraform script for creating an EC2 instance and an Elastic IP. The script includes comments for replacing region, AMI ID, and instance type. The terminal output at the bottom shows the instance ID `i-00a9741fc1f8d9b82 (terraform-host)` and its public and private IP addresses. The footer of the console contains the CloudShell logo, feedback link, and copyright information for Amazon Web Services.

```
GNU nano 6.2 main.tf
provider "aws" {
  region = "us-east-2" # Replace with your desired region
  access_key = "AKIAZQ3DRJS3IUZKMLHV"
  secret_key = "OSdy2WsDMYrYTW7eizFdhQegHKeV2d+5pXZKOCRT"
}

# Create an EC2 Instance
resource "aws_instance" "example" {
  ami           = "ami-0884d2865dbe9de4b" # Replace with the desired AMI ID
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform-Example-Instance"
  }

  # Associate with the default subnet
  vpc_security_group_ids = [aws_security_group.allow_ssh.id]

  # Enable detailed monitoring
  monitoring = true
}

# Create an Elastic IP and associate it with the new instance
resource "aws_eip" "example" {
  instance = aws_instance.example.id
  tags = {
    Name = "Terraform-Example-EIP"
  }
}
```

[ Read 49 lines ]

^G Help	^O Write Out	^W Where Is	^R Cut	^T Execute	^C Location	M-U Undo	M-A Set Mark	M-I To Bracket	M-Q Previous
^X Exit	^R Read File	^N Replace	^U Paste	^J Justify	^/_ Go To Line	M-E Redo	M-G Copy	^Q Where Was	M-W Next

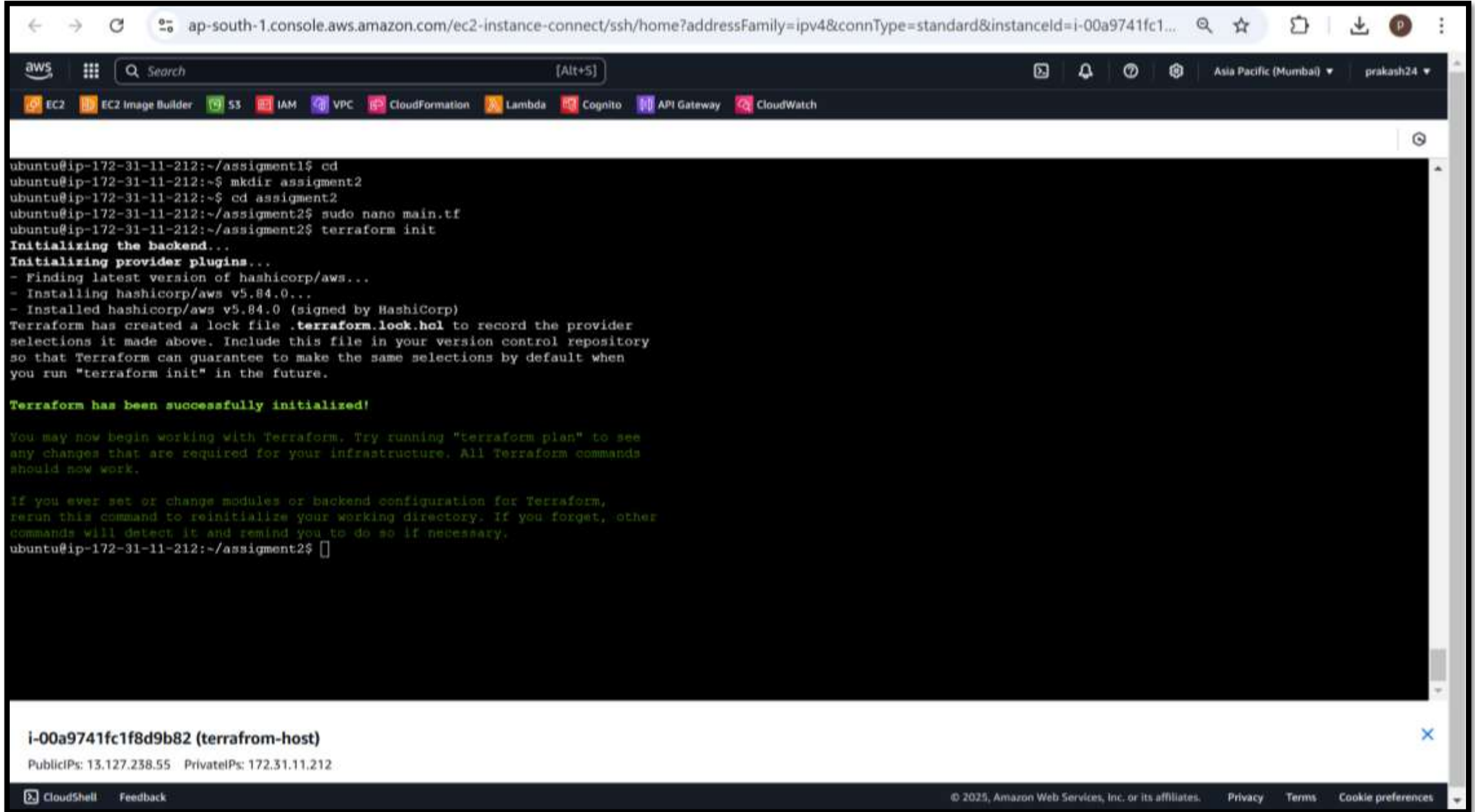
i-00a9741fc1f8d9b82 (terraform-host)

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Running Terraform init Command



The screenshot shows an AWS CloudShell terminal window. The browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1...`. The AWS navigation bar at the top includes the AWS logo, a search bar, and various service icons (EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, CloudWatch). The user's location is set to Asia Pacific (Mumbai) and the username is prakash24.

The terminal output shows the following commands and their results:

```
ubuntu@ip-172-31-11-212:~/assignment1$ cd
ubuntu@ip-172-31-11-212:~$ mkdir assignment2
ubuntu@ip-172-31-11-212:~$ cd assignment2
ubuntu@ip-172-31-11-212:~/assignment2$ sudo nano main.tf
ubuntu@ip-172-31-11-212:~/assignment2$ terraform init
```

**Initializing the backend...**

**Initializing provider plugins...**

- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (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.

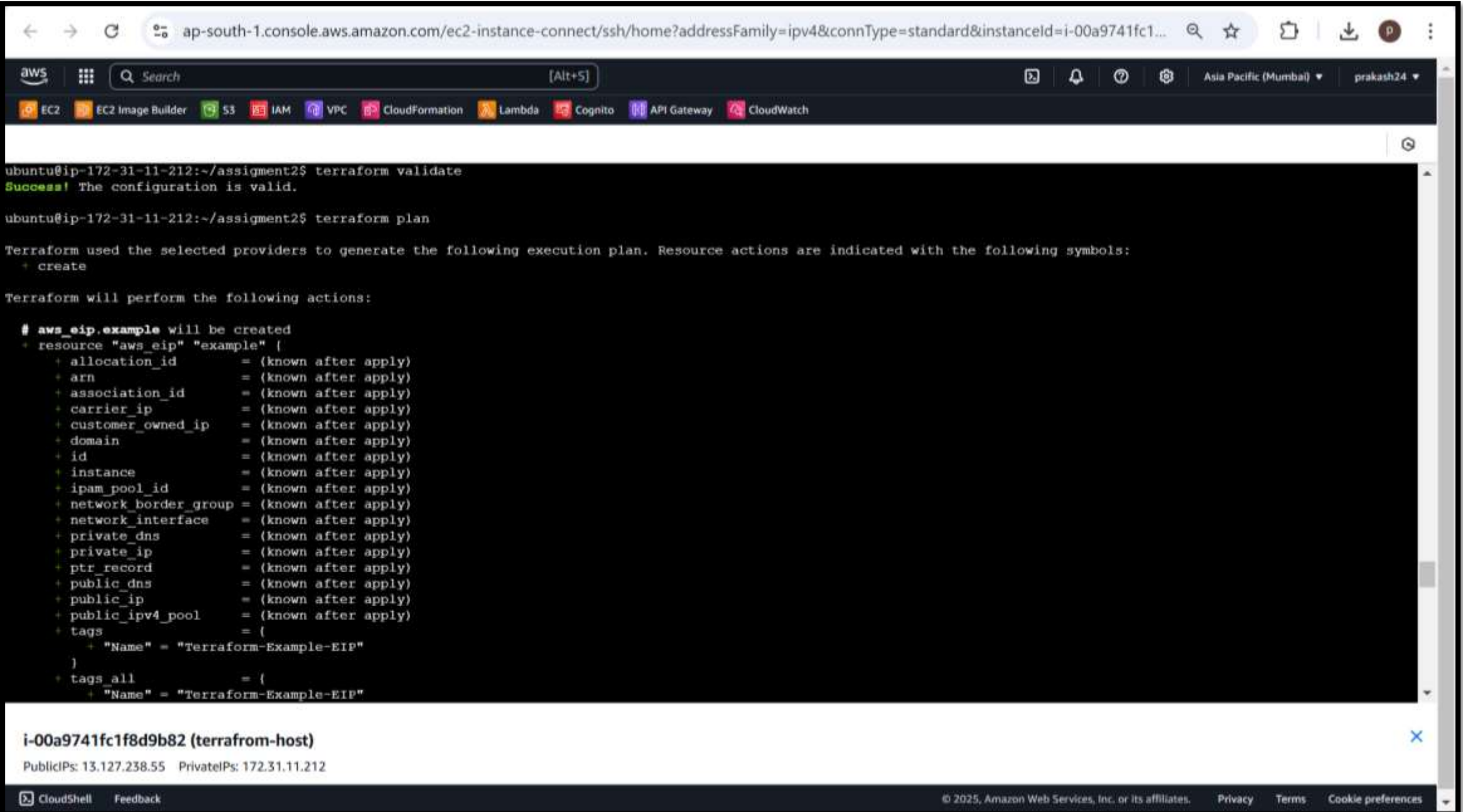
```
ubuntu@ip-172-31-11-212:~/assignment2$
```

At the bottom of the terminal window, the instance ID `i-00a9741fc1f8d9b82 (terraform-host)` is displayed, along with its Public IP (`13.127.238.55`) and Private IP (`172.31.11.212`).

The footer of the CloudShell window includes the CloudShell logo, a feedback link, and copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. It also links to Privacy, Terms, and Cookie preferences.



## Running Terraform validate & Terraform plan Command



The screenshot shows the AWS CloudShell interface. The terminal window displays the following commands and output:

```
ubuntu@ip-172-31-11-212:~/assignment2$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-11-212:~/assignment2$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_eip.example will be created
+ resource "aws_eip" "example" {
  + allocation_id      = (known after apply)
  + arn                = (known after apply)
  + association_id     = (known after apply)
  + carrier_ip         = (known after apply)
  + customer_owned_ip  = (known after apply)
  + domain             = (known after apply)
  + id                 = (known after apply)
  + instance           = (known after apply)
  + ipam_pool_id       = (known after apply)
  + network_border_group = (known after apply)
  + network_interface  = (known after apply)
  + private_dns        = (known after apply)
  + private_ip         = (known after apply)
  + ptr_record         = (known after apply)
  + public_dns         = (known after apply)
  + public_ip          = (known after apply)
  + public_ipv4_pool    = (known after apply)
  + tags               = {
    + "Name" = "Terraform-Example-EIP"
  }
  + tags_all           = {
    + "Name" = "Terraform-Example-EIP"
  }
}
```

**i-00a9741fc1f8d9b82 (terraform-host)**

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



# Running Terraform apply Command

The screenshot shows an AWS CloudShell terminal window. The browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1...`. The AWS console header is visible with the search bar and navigation icons for EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user is logged in as 'prakash24' in the 'Asia Pacific (Mumbai)' region.

```
+ prefix_list_ids = []
+ protocol        = "tcp"
+ security_groups = []
+ self            = false
+ to_port         = 22
# (1 unchanged attribute hidden)
},
+ name              = "allow_ssh"
+ name_prefix       = (known after apply)
+ owner_id          = (known after apply)
+ revoke_rules_on_delete = false
+ tags_all          = (known after apply)
+ vpc_id            = (known after apply)
}
```

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?  
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.

Enter a value: yes

aws\_security\_group.allow\_ssh: Creating...  
aws\_security\_group.allow\_ssh: Creation complete after 5s [id=sg-0a2429678d6c6182d]  
aws\_instance.example: Creating...  
aws\_instance.example: Still creating... [10s elapsed]  
aws\_instance.example: Creation complete after 15s [id=i-0f4ae25a002fdd875]  
aws\_eip.example: Creating...  
aws\_eip.example: Creation complete after 3s [id=eipalloc-01812cc77f52c1c5c]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

ubuntu@ip-172-31-11-212:~/assignment2\$

i-00a9741fc1f8d9b82 (terraform-host)

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Verifying the Elastic IP launched by terraform

The screenshot shows the AWS Management Console for the us-east-2 region. The left sidebar contains navigation links for Elastic Block Store, Network & Security (with Elastic IPs highlighted), Load Balancing, and Auto Scaling. The main content area is titled 'Elastic IP addresses (1)' and displays a table with one entry: 'Terraform-Example-EIP' with an allocated IPv4 address of 3.141.55.205. A notification banner at the bottom suggests viewing IP address usage with Public IP insights.

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Addresses:

aws [Search] [Alt+S] United States (Ohio) prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

**Elastic IP addresses (1)** [Actions](#) [Allocate Elastic IP address](#)

Find resources by attribute or tag

<input type="checkbox"/>	Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS rec
<input type="checkbox"/>	Terraform-Example-EIP	<a href="#">3.141.55.205</a>	Public IP	eipalloc-01812cc77f52c1c5c	-

View IP address usage and recommendations to release unused IPs with [Public IP insights](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Verifying the EC2 instance launched by terraform

← → ↺

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#instances:v=3;\$case=tags:true%5C,client:false,\$regex=tags:false%5C,client...

aws

Search [Alt+S]

United States (Ohio) prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

DashboardEC2 Global ViewEvents

InstancesInstancesInstance TypesLaunch TemplatesSpot RequestsSavings PlansReserved InstancesDedicated HostsCapacity Reservations

ImagesAMIsAMI Catalog

Elastic Block StoreVolumesSnapshotsLifecycle Manager

Instances (1/2) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

< 1 >

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avi
<input checked="" type="checkbox"/>	Terraform-Example-Instance	i-0f4ae25a002fdd875	Running	t2.micro	Initializing	View alarms +	us-
<input type="checkbox"/>	EC2Instance-Ohio-region	i-0aa5326d63c0fd0d0	Terminated	t2.micro	-	View alarms +	us-

i-0f4ae25a002fdd875 (Terraform-Example-Instance)

DetailsStatus and alarmsMonitoringSecurityNetworkingStorageTags

Networking details Info

Public IPv4 address3.141.55.205 | open address

Private IPv4 addresses172.31.6.86

VPC IDvpc-001d04e278d9672d9

Public IPv4 DNSec2-3-141-55-205.us-east-2.compute.amazonaws.com | open address

Private IP DNS name (IPv4 only)ip-172-31-6-86.us-east-2.compute.internal

Subnet ID

IPv6 addresses

Secondary private IPv4 addresses

CloudShellFeedback

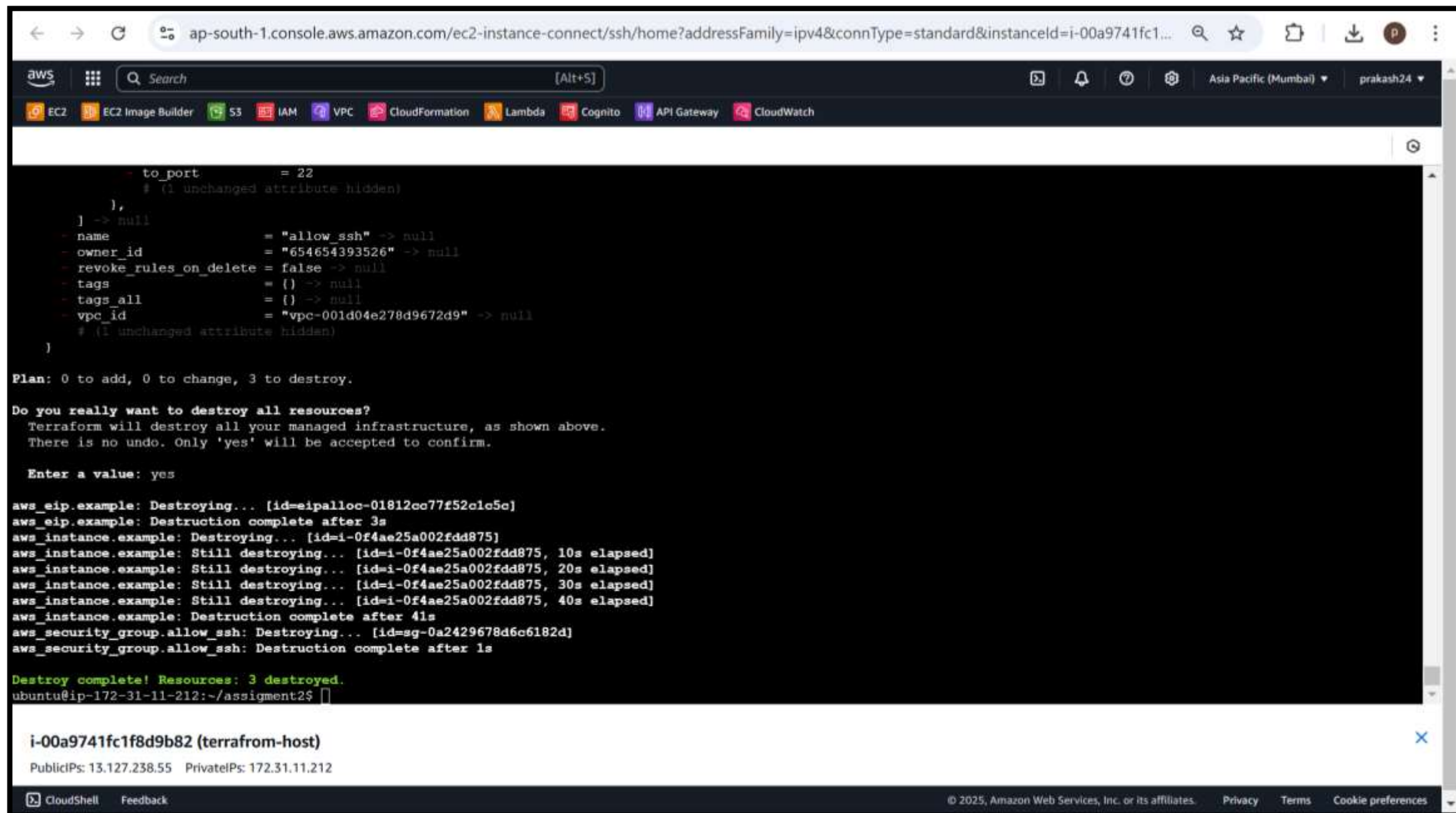
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Module 8: Terraform Assignment -3

## Tasks To Be Performed:

1. Destroy the previous deployment
2. Create 2 EC2 instances in Ohio and N.Virginia respectively
3. Rename Ohio's instance to 'hello-ohio' and Virginia's instance to 'hello-virginia'

## Destroying Resource launched by terraform using terraform destroy command



The screenshot shows the AWS Management Console interface for the 'ap-south-1' region. The top navigation bar includes the AWS logo, a search bar, and various service icons (EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, CloudWatch). The user is logged in as 'prakash24' in the 'Asia Pacific (Mumbai)' region.

The main content area displays the output of a Terraform destroy command. The output shows the Terraform plan, a confirmation prompt, and the successful destruction of three resources:

```
to_port      = 22
# (1 unchanged attribute hidden)
],
] -> null
- name        = "allow_ssh" -> null
- owner_id    = "654654393526" -> null
- revoke_rules_on_delete = false -> null
- tags        = {} -> null
- tags_all    = {} -> null
- vpc_id      = "vpc-001d04e278d9672d9" -> null
# (1 unchanged attribute hidden)
)

Plan: 0 to add, 0 to change, 3 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_eip.example: Destroying... [id=eipalloc-01812cc77f52c1c5c]
aws_eip.example: Destruction complete after 3s
aws_instance.example: Destroying... [id=i-0f4ae25a002fdd875]
aws_instance.example: Still destroying... [id=i-0f4ae25a002fdd875, 10s elapsed]
aws_instance.example: Still destroying... [id=i-0f4ae25a002fdd875, 20s elapsed]
aws_instance.example: Still destroying... [id=i-0f4ae25a002fdd875, 30s elapsed]
aws_instance.example: Still destroying... [id=i-0f4ae25a002fdd875, 40s elapsed]
aws_instance.example: Destruction complete after 41s
aws_security_group.allow_ssh: Destroying... [id=sg-0a2429678d6c6182d]
aws_security_group.allow_ssh: Destruction complete after 1s

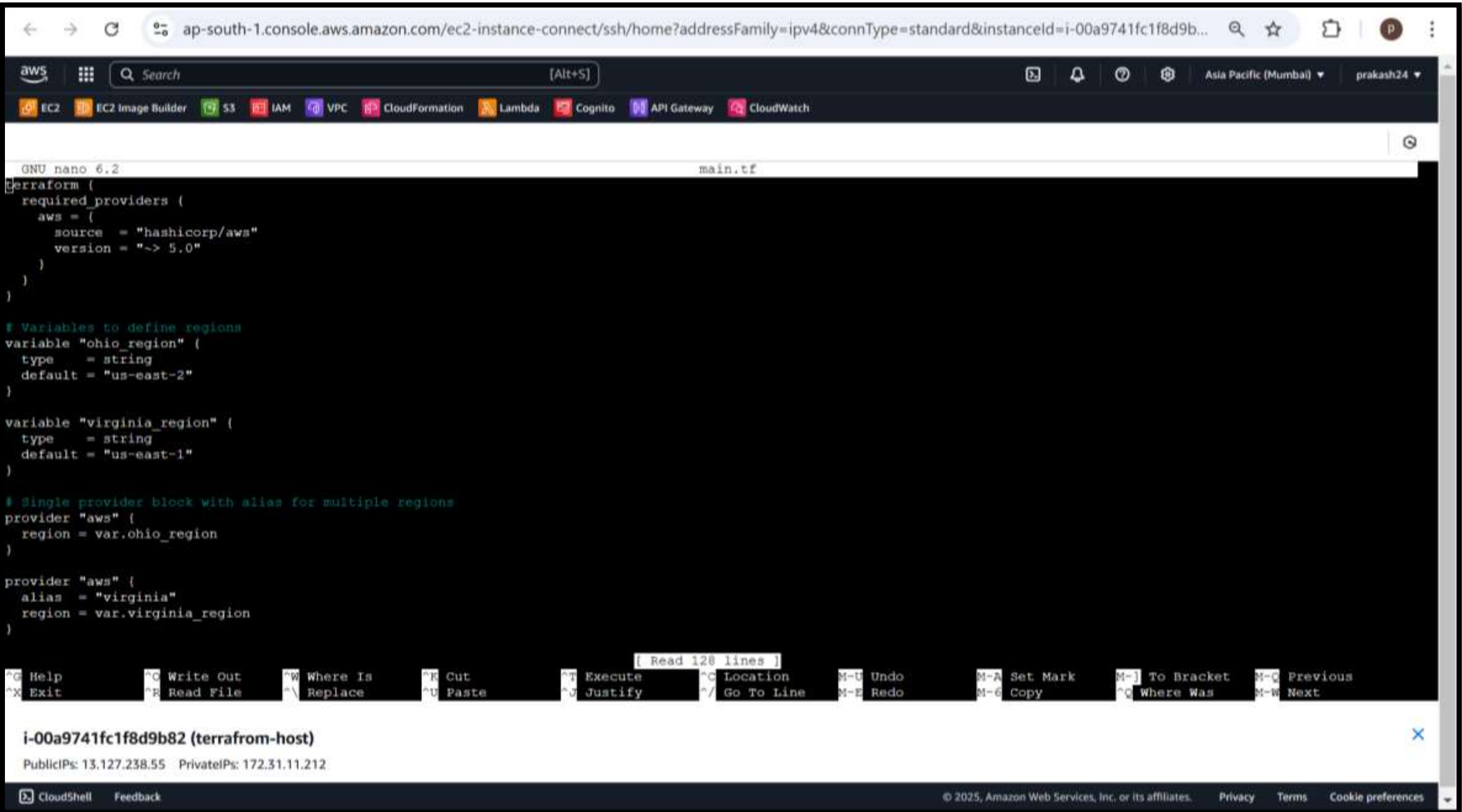
Destroy complete! Resources: 3 destroyed.
ubuntu@ip-172-31-11-212:~/assignment2$
```

At the bottom of the console, there is a summary for the instance 'i-00a9741fc1f8d9b82 (terraform-host)'. It shows the PublicIPs as 13.127.238.55 and PrivateIPs as 172.31.11.212.

The footer of the console includes the 'CloudShell' logo, a 'Feedback' link, and the copyright notice '© 2025, Amazon Web Services, Inc. or its affiliates.' along with links for 'Privacy', 'Terms', and 'Cookie preferences'.



## Terraform Script main.tf file



```
GNU nano 6.2 main.tf
Terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

# Variables to define regions
variable "ohio_region" {
  type = string
  default = "us-east-2"
}

variable "virginia_region" {
  type = string
  default = "us-east-1"
}

# Single provider block with alias for multiple regions
provider "aws" {
  region = var.ohio_region
}

provider "aws" {
  alias = "virginia"
  region = var.virginia_region
}

[ Read 128 lines ]

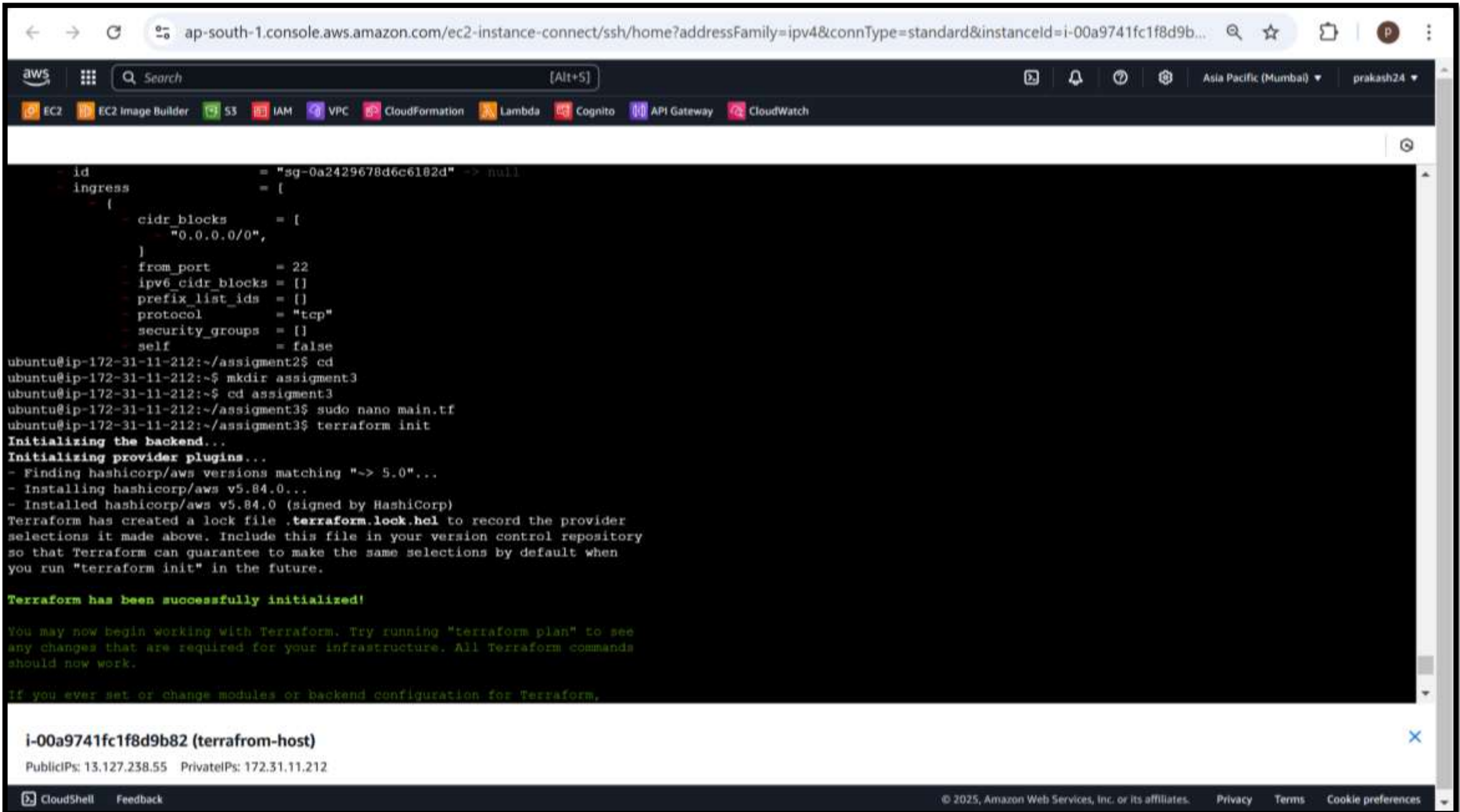
^G Help      ^C Write Out  ^W Where Is   ^R Cut        ^T Execute    ^L Location   M-U Undo      M-A Set Mark  M-I To Bracket M-Q Previous
^X Exit      ^R Read File  ^\ Replace    ^U Paste       ^J Justify    ^_ Go To Line  M-E Redo      M-G Copy     ^O Where Was  M-W Next

i-00a9741fc1f8d9b82 (terraform-host)
PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```



# Running Terraform init Command



The screenshot shows an AWS CloudShell terminal window. The top navigation bar includes the AWS logo, a search bar, and various service icons (EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, CloudWatch). The user is logged in as 'prakash24' in the 'Asia Pacific (Mumbai)' region. The terminal window displays the following commands and output:

```
ubuntu@ip-172-31-11-212:~/assignment2$ cd
ubuntu@ip-172-31-11-212:~$ mkdir assignment3
ubuntu@ip-172-31-11-212:~$ cd assignment3
ubuntu@ip-172-31-11-212:~/assignment3$ sudo nano main.tf
ubuntu@ip-172-31-11-212:~/assignment3$ terraform init
```

**Initializing the backend...**

**Initializing provider plugins...**

- Finding hashicorp/aws versions matching "> 5.0"...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (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,

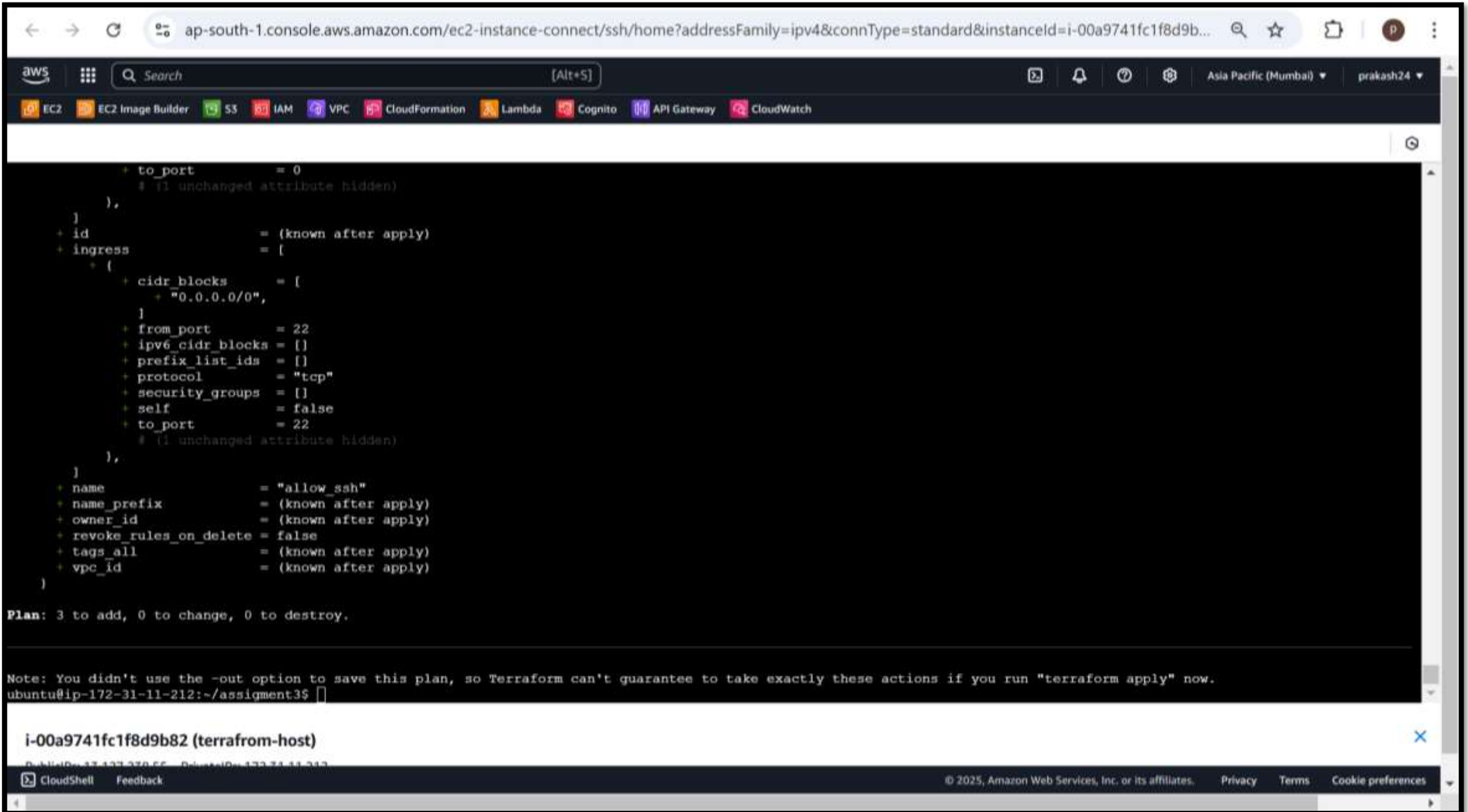
**i-00a9741fc1f8d9b82 (terraform-host)**

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Running Terraform validate & Terraform Plan Command



The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and various service icons (EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, CloudWatch). The main content area displays the output of a Terraform plan command. The plan shows three resources to be added, with no changes or deletions. The resources are:

- `aws_security_group_rule.allow_ssh`: A security group rule for SSH access.
- `aws_security_group.allow_ssh`: A security group for the EC2 instance.
- `aws_instance.ec2`: An EC2 instance.

The plan output is as follows:

```
+ to_port      = 0
+ (1 unchanged attribute hidden)
},
]
+ id           = (known after apply)
+ ingress      = [
+   {
+     cidr_blocks = [
+       "0.0.0.0/0",
+     ]
+     from_port    = 22
+     ipv6_cidr_blocks = []
+     prefix_list_ids = []
+     protocol      = "tcp"
+     security_groups = []
+     self          = false
+     to_port      = 22
+     (1 unchanged attribute hidden)
+   },
+ ]
+ name          = "allow_ssh"
+ name_prefix   = (known after apply)
+ owner_id      = (known after apply)
+ revoke_rules_on_delete = false
+ tags_all      = (known after apply)
+ vpc_id        = (known after apply)
}
```

**Plan:** 3 to add, 0 to change, 0 to destroy.

Note: You didn't use the `-out` option to save this plan, so Terraform can't guarantee to take exactly these actions if you run `"terraform apply"` now.

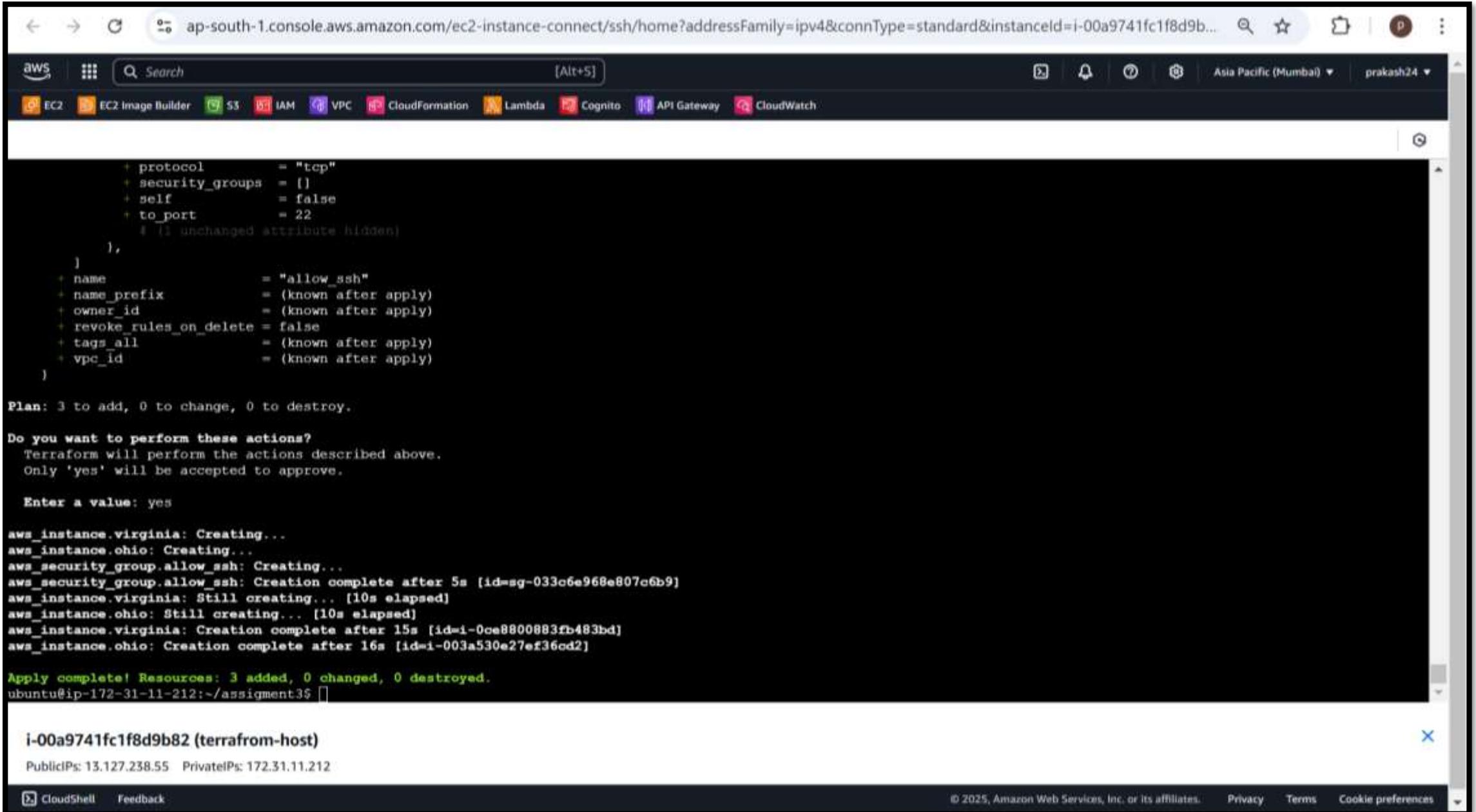
ubuntu@ip-172-31-11-212:~/assignment3\$

i-00a9741fc1f8d9b82 (terraform-host)

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Running Terraform validate apply Command



The screenshot shows an AWS CloudShell terminal window. The browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1f8d9b...`. The AWS navigation bar at the top includes icons for EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user's profile is 'prakash24' in the 'Asia Pacific (Mumbai)' region.

The terminal window shows the following Terraform configuration and execution:

```
+ protocol      = "tcp"
+ security_groups = []
+ self         = false
+ to_port      = 22
  # [! unchanged attribute hidden]
},
}
+ name          = "allow_ssh"
+ name_prefix   = (known after apply)
+ owner_id      = (known after apply)
+ revoke_rules_on_delete = false
+ tags_all      = (known after apply)
+ vpc_id        = (known after apply)
}
```

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?  
Terraform will perform the actions described above.  
Only 'yes' will be accepted to approve.

Enter a value: yes

aws\_instance.virginia: Creating...  
aws\_instance.ohio: Creating...  
aws\_security\_group.allow\_ssh: Creating...  
aws\_security\_group.allow\_ssh: Creation complete after 5s [id=sg-033c6e968e807c6b9]  
aws\_instance.virginia: Still creating... [10s elapsed]  
aws\_instance.ohio: Still creating... [10s elapsed]  
aws\_instance.virginia: Creation complete after 15s [id=i-0ce8800883fb483bd]  
aws\_instance.ohio: Creation complete after 16s [id=i-003a530e27ef36cd2]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

ubuntu@ip-172-31-11-212:~/assignment3\$

**i-00a9741fc1f8d9b82 (terraform-host)**

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Verifying the EC2 instance launched in Ohio region by terraform

← → ↻

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:instanceState=running

🔍 ☆ 🏠 P ⋮

aws

🔍 Search [Alt+S]

📄 🔔 ⓘ ⚙️ United States (Ohio) prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

☰

Dashboard <

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Instances (1) Info

Last updated less than a minute ago 🔄

Connect

Instance state ▼

Actions ▼

Launch instances ▼

🔍 Find Instance by attribute or tag (case-sensitive)

All states ▼

Instance state = running X Clear filters

< 1 > ⚙️

<input type="checkbox"/>	Name ✎	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Avi
<input type="checkbox"/>	hello-ohio	i-003a530e27ef36cd2	Running 🔍 🔍	t2.micro	2/2 checks passed	View alarms +	us-

Select an instance = ⚙️ ▼

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Verifying the EC2 instance launched in North Virginia region by terraform

← → ↺

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running

🔍 ☆ 📄 👤

aws

🔍 Search [Alt+S]

📧 🔔 ⓘ ⚙️ United States (N. Virginia) 👤 prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

☰

Dashboard <

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

CloudShell Feedback

Instances (1) Info

Last updated 1 minute ago 🔄

Connect

Instance state ▾

Actions ▾

Launch Instances

🔍 Find Instance by attribute or tag (case-sensitive)

All states ▾

Instance state = running ✕

Clear filters

< 1 > ⚙️

<input type="checkbox"/>	Name 🔗	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS
<input type="checkbox"/>	hello-virginia	i-0ce8800883fb483bd	Running 🔍 🔍	t2.micro	🕒 Initializing	<a href="#">View alarms +</a>	us-east-1a	ec2-54-196-205

Select an instance

=

⚙️ ▾

© 2025, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

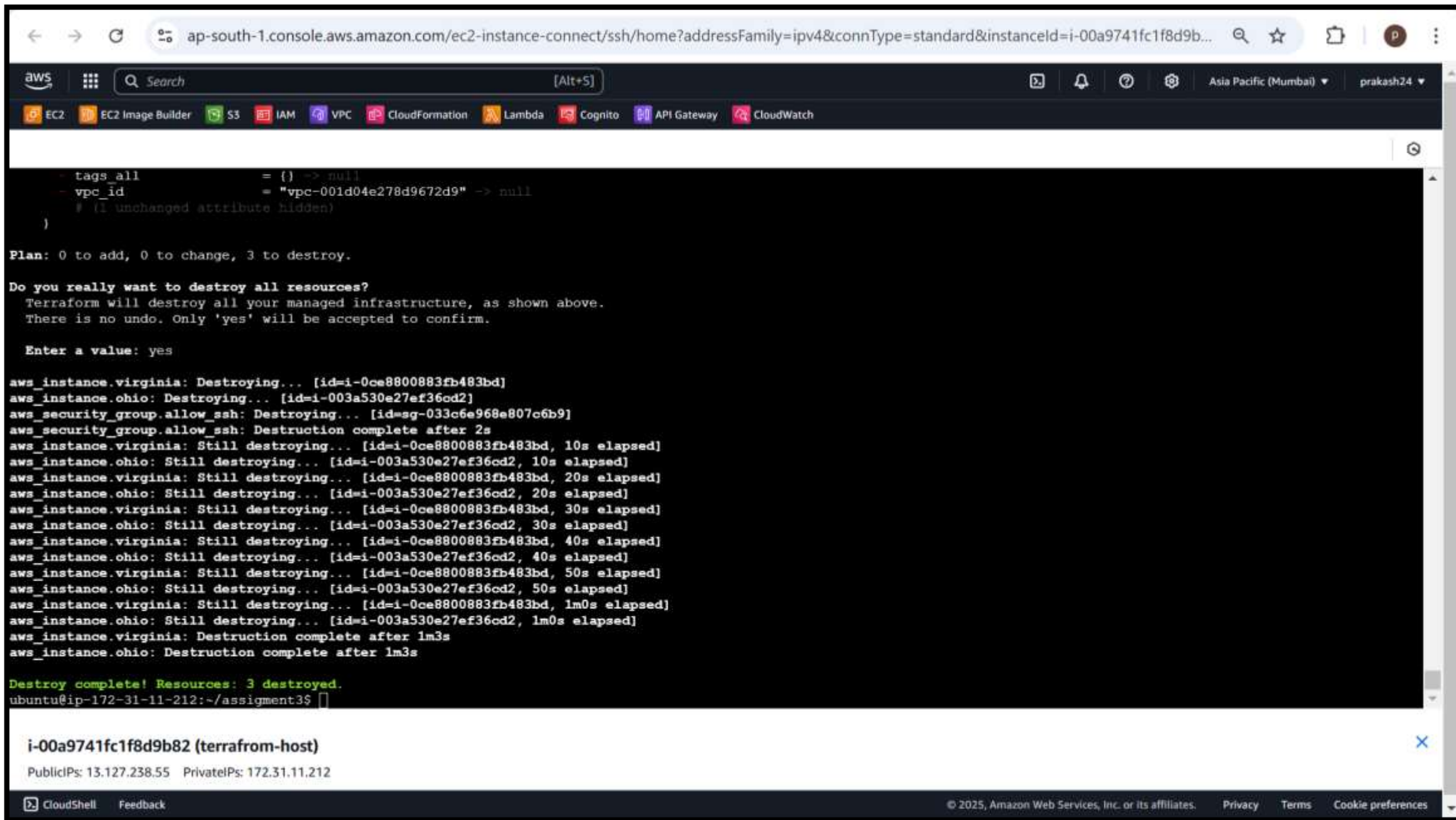
# Module 8: Terraform Assignment -4

## **Tasks To Be Performed:**

1. Destroy the previous deployments
2. Create a VPC with the required components using Terraform
3. Deploy an EC2 instance inside the VPC



## Destroying Resource launched by terraform using terraform destroy command



ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1f8d9b...

aws

Search [Alt+S]

Asia Pacific (Mumbai) prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

```
- tags_all      = {} -> null
- vpc_id        = "vpc-001d04e278d9672d9" -> null
# (1 unchanged attribute hidden)
}
```

Plan: 0 to add, 0 to change, 3 to destroy.

**Do you really want to destroy all resources?**  
Terraform will destroy all your managed infrastructure, as shown above.  
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws\_instance.virginia: Destroying... [id=i-0ce8800883fb483bd]  
aws\_instance.ohio: Destroying... [id=i-003a530e27ef36cd2]  
aws\_security\_group.allow\_ssh: Destroying... [id=sg-033c6e968e807c6b9]  
aws\_security\_group.allow\_ssh: Destruction complete after 2s  
aws\_instance.virginia: Still destroying... [id=i-0ce8800883fb483bd, 10s elapsed]  
aws\_instance.ohio: Still destroying... [id=i-003a530e27ef36cd2, 10s elapsed]  
aws\_instance.virginia: Still destroying... [id=i-0ce8800883fb483bd, 20s elapsed]  
aws\_instance.ohio: Still destroying... [id=i-003a530e27ef36cd2, 20s elapsed]  
aws\_instance.virginia: Still destroying... [id=i-0ce8800883fb483bd, 30s elapsed]  
aws\_instance.ohio: Still destroying... [id=i-003a530e27ef36cd2, 30s elapsed]  
aws\_instance.virginia: Still destroying... [id=i-0ce8800883fb483bd, 40s elapsed]  
aws\_instance.ohio: Still destroying... [id=i-003a530e27ef36cd2, 40s elapsed]  
aws\_instance.virginia: Still destroying... [id=i-0ce8800883fb483bd, 50s elapsed]  
aws\_instance.ohio: Still destroying... [id=i-003a530e27ef36cd2, 50s elapsed]  
aws\_instance.virginia: Still destroying... [id=i-0ce8800883fb483bd, 1m0s elapsed]  
aws\_instance.ohio: Still destroying... [id=i-003a530e27ef36cd2, 1m0s elapsed]  
aws\_instance.virginia: Destruction complete after 1m3s  
aws\_instance.ohio: Destruction complete after 1m3s

**Destroy complete! Resources: 3 destroyed.**  
ubuntu@ip-172-31-11-212:~/assignment3\$

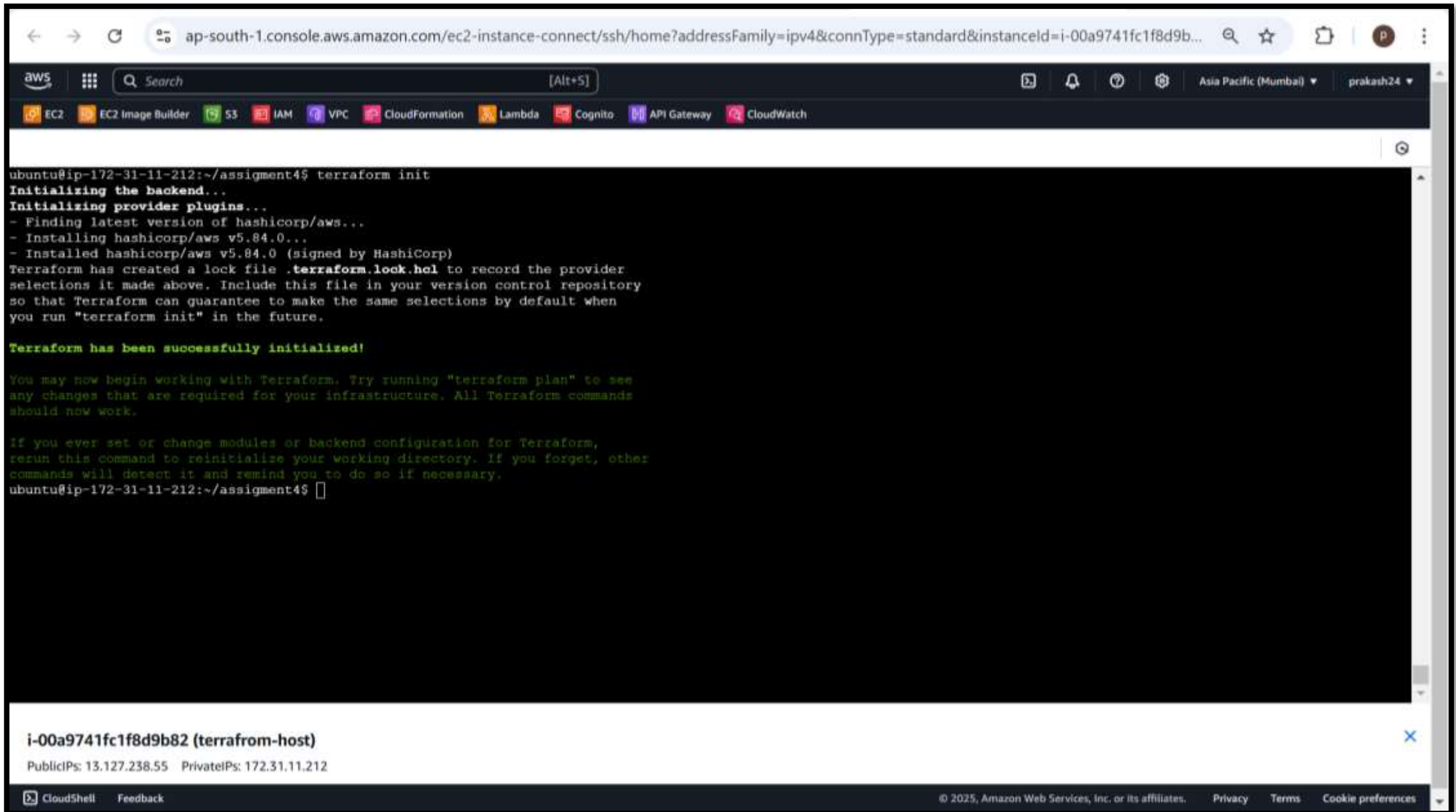
i-00a9741fc1f8d9b82 (terraform-host)

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Running Terraform init Command



The screenshot shows the AWS CloudShell interface. The terminal window displays the output of the 'terraform init' command. The output indicates that Terraform has successfully initialized the backend and provider plugins. It also provides instructions on how to use Terraform, including running 'terraform plan' to see changes and reinitializing the configuration if needed.

```
ubuntu@ip-172-31-11-212:~/assignment4$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (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-172-31-11-212:~/assignment4$
```

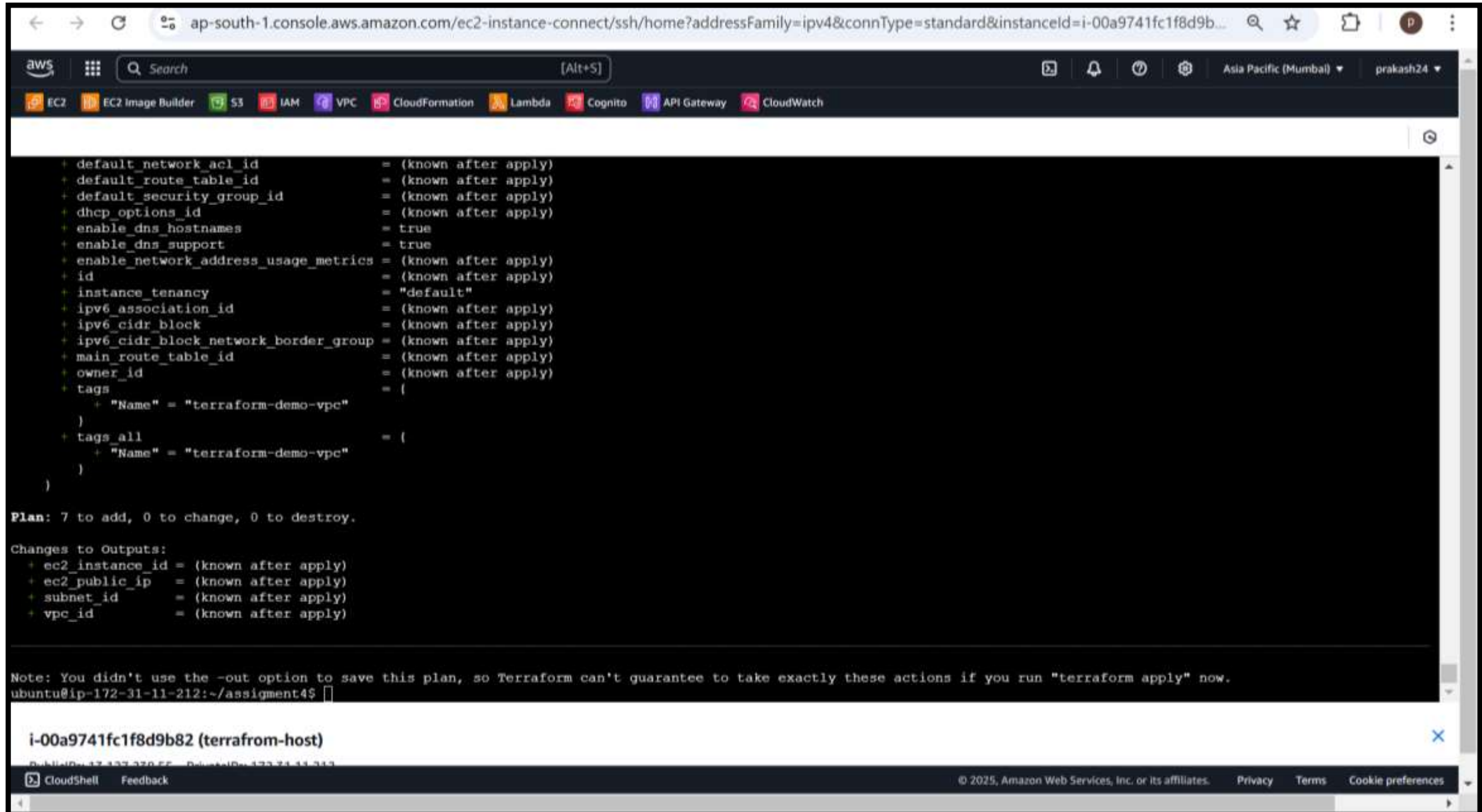
i-00a9741fc1f8d9b82 (terrafrom-host)

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Running Terraform validate & Terraform plan Command



The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, a search bar, and various service icons (EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, CloudWatch). The user is logged in as 'prakash24' in the 'Asia Pacific (Mumbai)' region. The main content area displays the output of a Terraform plan command, showing a list of resources to be added, with their values marked as '(known after apply)'. Below the plan output, a summary states: 'Plan: 7 to add, 0 to change, 0 to destroy.' This is followed by a section titled 'Changes to Outputs:' which lists four outputs: 'ec2\_instance\_id', 'ec2\_public\_ip', 'subnet\_id', and 'vpc\_id', all marked as '(known after apply)'. A note at the bottom of the console output area reads: 'Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.' The terminal prompt shows the user is at 'ubuntu@ip-172-31-11-212:~/assignment4\$'. At the bottom of the console window, there is a header for the EC2 instance 'i-00a9741fc1f8d9b82 (terraform-host)' and a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services.

```
+ default_network_acl_id      = (known after apply)
+ default_route_table_id     = (known after apply)
+ default_security_group_id  = (known after apply)
+ dhcp_options_id            = (known after apply)
+ enable_dns_hostnames        = true
+ enable_dns_support          = true
+ enable_network_address_usage_metrics = (known after apply)
+ id                          = (known after apply)
+ instance_tenancy            = "default"
+ ipv6_association_id         = (known after apply)
+ ipv6_cidr_block              = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id         = (known after apply)
+ owner_id                    = (known after apply)
+ tags                        = {
+   "Name" = "terraform-demo-vpc"
+ }
+ tags_all                    = {
+   "Name" = "terraform-demo-vpc"
+ }
)
```

**Plan:** 7 to add, 0 to change, 0 to destroy.

**Changes to Outputs:**

```
+ ec2_instance_id = (known after apply)
+ ec2_public_ip   = (known after apply)
+ subnet_id       = (known after apply)
+ vpc_id          = (known after apply)
```

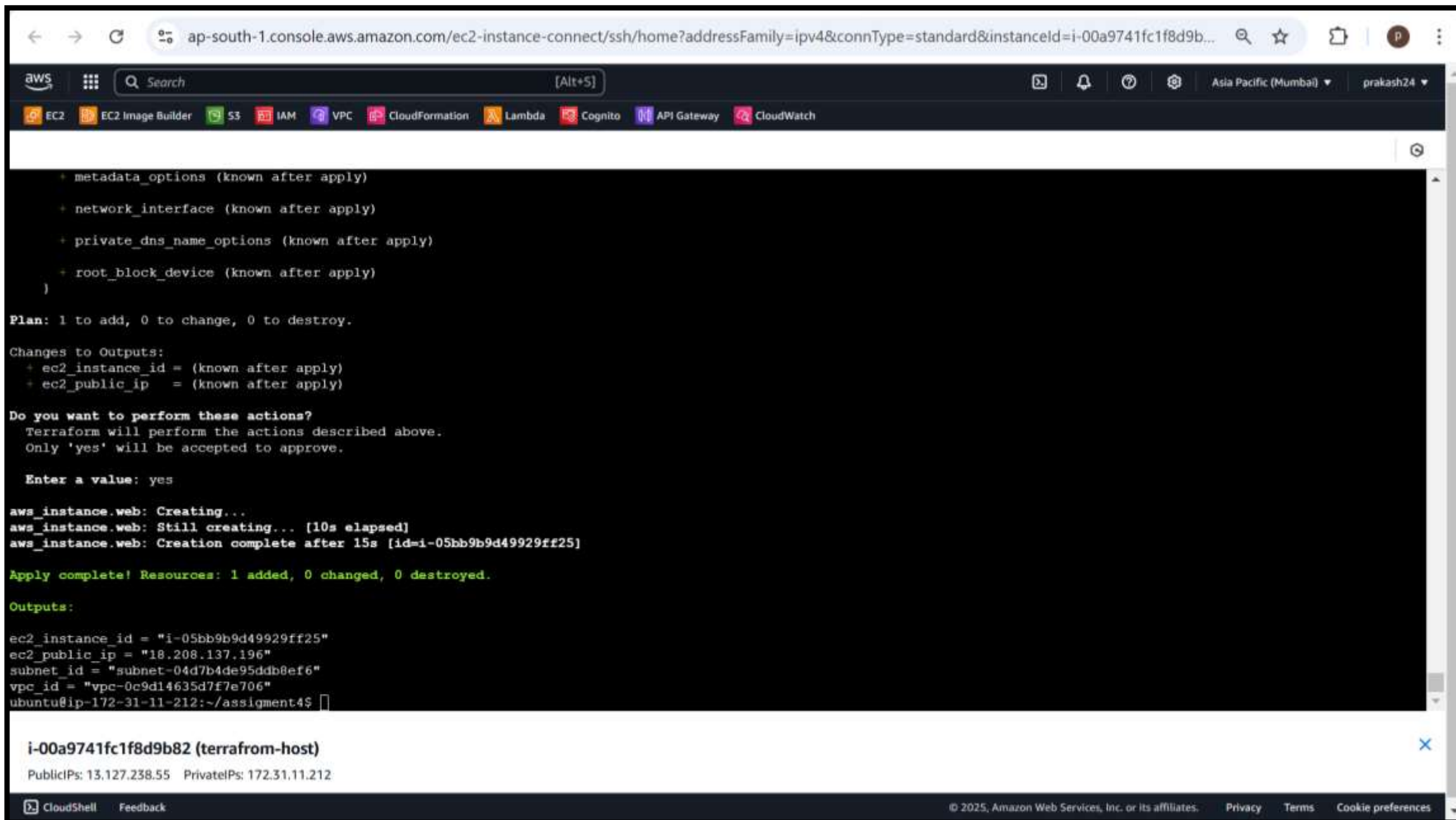
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

ubuntu@ip-172-31-11-212:~/assignment4\$

i-00a9741fc1f8d9b82 (terraform-host)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Running Terraform apply Command



The screenshot shows the AWS Management Console interface. At the top, the browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1f8d9b...`. The console header includes the AWS logo, a search bar, and navigation links for various services: EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user's location is set to Asia Pacific (Mumbai) and the username is prakash24.

The main content area displays the output of a Terraform `apply` command executed in a CloudShell session. The output shows the plan for adding a new EC2 instance, followed by the execution of the `apply` command. The instance is successfully created with ID `i-05bb9b9d49929ff25`. The output also shows the public IP address `18.208.137.196` and the subnet ID `subnet-04d7b4de95ddb8ef6`.

```
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
)

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ ec2_instance_id = (known after apply)
+ ec2_public_ip   = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.web: Creating...
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Creation complete after 15s [id=i-05bb9b9d49929ff25]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

ec2_instance_id = "i-05bb9b9d49929ff25"
ec2_public_ip   = "18.208.137.196"
subnet_id      = "subnet-04d7b4de95ddb8ef6"
vpc_id         = "vpc-0c9d14635d7f7e706"
ubuntu@ip-172-31-11-212:~/assignment4$
```

At the bottom of the console, a summary for the instance `i-00a9741fc1f8d9b82 (terraform-host)` is shown, including its Public IP (`13.127.238.55`) and Private IP (`172.31.11.212`).

The footer of the console includes the CloudShell logo, a feedback link, and copyright information: © 2025, Amazon Web Services, Inc. or its affiliates. It also includes links for Privacy, Terms, and Cookie preferences.

# Verifying a launched VPC by terraform

The screenshot shows the AWS Management Console VPC dashboard for the us-east-1 region. The left sidebar contains navigation links for VPC dashboard, EC2 Global View, and various VPC resources. The main content area displays 'Your VPCs (2)' with a table listing the VPCs. The first VPC is named 'terraform-demo-vpc' with ID 'vpc-0c9d14635d7f7e706' and is in an 'Available' state. The second VPC is unnamed with ID 'vpc-08ac7876d48a5c87d' and is also in an 'Available' state. Both VPCs have 'Block Public Access' set to 'Off' and 'IPv4 CIDR' of '10.0.0.0/16' and '172.31.0.0/16' respectively. The 'DHCP option set' for both is 'dopt-06e290acd'.

us-east-1.console.aws.amazon.com/vpconsole/home?region=us-east-1#vpcs:

aws [Alt+S] United States (N. Virginia) prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

VPC dashboard < EC2 Global View Filter by VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections

Security

- Network ACLs
- Security groups

PrivateLink and Lattice

- Getting started Updated
- Endpoints Updated
- Endpoint services

Your VPCs (2) Info Last updated less than a minute ago Actions Create VPC

Search

<input type="checkbox"/>	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option se
<input type="checkbox"/>	terraform-demo-vpc	<a href="#">vpc-0c9d14635d7f7e706</a>	Available	Off	10.0.0.0/16	-	<a href="#">dopt-06e290acd</a>
<input type="checkbox"/>	-	<a href="#">vpc-08ac7876d48a5c87d</a>	Available	Off	172.31.0.0/16	-	<a href="#">dopt-06e290acd</a>

Select a VPC above

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



# Verifying a launched EC2 inside the VPC by terraform

←→↻

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running

🔍☆📄👤

aws

Search [Alt+S]

📄🔔🔗🛠️

United States (N. Virginia) ▾

prakash24 ▾

EC2EC2 Image BuilderS3IAMVPCCloudFormationLambdaCognitoAPI GatewayCloudWatch

☰

Dashboard<

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Instances (1) Info

Last updated less than a minute ago

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Find Instance by attribute or tag (case-sensitive)

All states ▾

Instance state = running X

Clear filters

< 1 > ⚙️

<input type="checkbox"/>	Name <a href="#">🔗</a>	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public IPv4 DN!
<input type="checkbox"/>	terraform-demo-ec2	i-05bb9b9d49929ff25	Running <a href="#">🔗</a> <a href="#">🔗</a>	t2.micro	🕒 Initializing	<a href="#">View alarms +</a>	us-east-1d	ec2-18-208-137

Select an instance

⚙️ ▾

CloudShellFeedback

© 2025, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

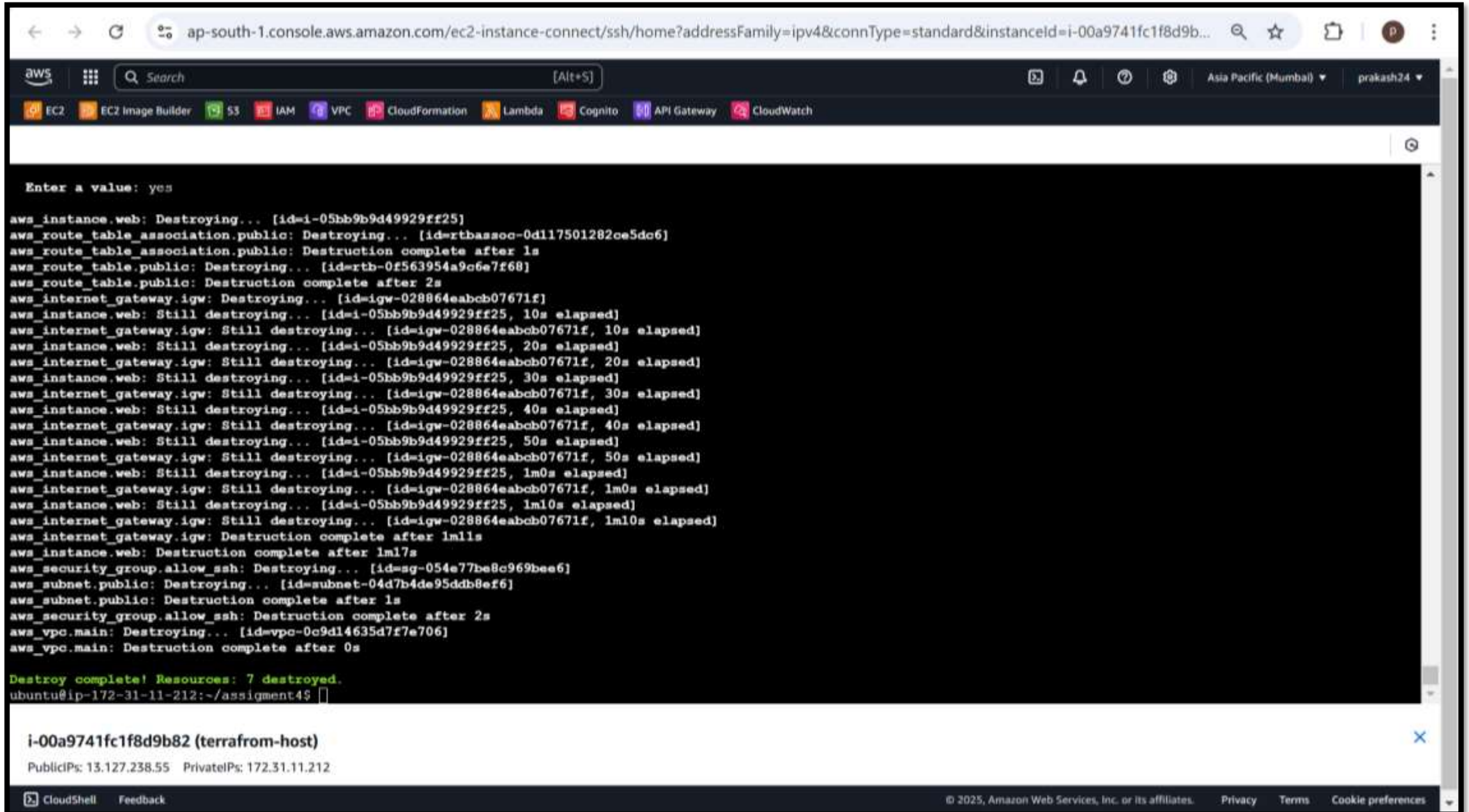


# Module 8: Terraform Assignment -5

## **Tasks To Be Performed:**

1. Destroy the previous deployments
2. Create a script to install Apache2
3. Run this script on a newly created EC2 instance
4. Print the IP address of the instance in a file on the local once deployed

## Destroying Resource launched by terraform using terraform destroy command



The screenshot shows the AWS Management Console interface. At the top, the browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-00a9741fc1f8d9b...`. The console header includes the AWS logo, a search bar, and navigation links for EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user's location is set to Asia Pacific (Mumbai) and the username is prakash24.

The main content area displays the output of a Terraform destroy command executed in a CloudShell session. The output shows the destruction of various AWS resources, including an EC2 instance, route table associations, a route table, an internet gateway, a security group, a subnet, and a VPC. The resources are destroyed in a specific order, with some resources being destroyed after others. The output is as follows:

```
Enter a value: yes
aws_instance.web: Destroying... [id=i-05bb9b9d49929ff25]
aws_route_table_association.public: Destroying... [id=rtbassoc-0d117501282ce5dc6]
aws_route_table_association.public: Destruction complete after 1s
aws_route_table.public: Destroying... [id=rtb-0f563954a9c6e7f68]
aws_route_table.public: Destruction complete after 2s
aws_internet_gateway.igw: Destroying... [id=igw-028864eabcb07671f]
aws_instance.web: Still destroying... [id=i-05bb9b9d49929ff25, 10s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-028864eabcb07671f, 10s elapsed]
aws_instance.web: Still destroying... [id=i-05bb9b9d49929ff25, 20s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-028864eabcb07671f, 20s elapsed]
aws_instance.web: Still destroying... [id=i-05bb9b9d49929ff25, 30s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-028864eabcb07671f, 30s elapsed]
aws_instance.web: Still destroying... [id=i-05bb9b9d49929ff25, 40s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-028864eabcb07671f, 40s elapsed]
aws_instance.web: Still destroying... [id=i-05bb9b9d49929ff25, 50s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-028864eabcb07671f, 50s elapsed]
aws_instance.web: Still destroying... [id=i-05bb9b9d49929ff25, 1m0s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-028864eabcb07671f, 1m0s elapsed]
aws_instance.web: Still destroying... [id=i-05bb9b9d49929ff25, 1m10s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-028864eabcb07671f, 1m10s elapsed]
aws_internet_gateway.igw: Destruction complete after 1m11s
aws_instance.web: Destruction complete after 1m17s
aws_security_group.allow_ssh: Destroying... [id=sg-054e77be8c969bee6]
aws_subnet.public: Destroying... [id=subnet-04d7b4de95ddb8ef6]
aws_subnet.public: Destruction complete after 1s
aws_security_group.allow_ssh: Destruction complete after 2s
aws_vpc.main: Destroying... [id=vpc-0c9d14635d7f7e706]
aws_vpc.main: Destruction complete after 0s

Destroy complete! Resources: 7 destroyed.
ubuntu@ip-172-31-11-212:~/assignment4$
```

Below the terminal output, the instance ID `i-00a9741fc1f8d9b82` is identified as the `(terraform-host)`. Below this, the public and private IP addresses are listed: `PublicIPs: 13.127.238.55` and `PrivateIPs: 172.31.11.212`.

The footer of the console shows the CloudShell logo, a feedback link, and the copyright notice: © 2025, Amazon Web Services, Inc. or its affiliates. It also includes links for Privacy, Terms, and Cookie preferences.

## Terraform Script main.tf file

The screenshot shows an AWS CloudShell session with the following components:

- Browser Address Bar:** `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-00a9741fc1f8d9b...`
- AWS Console Header:** Includes the AWS logo, a search bar, and navigation links for EC2, EC2 Image Builder, S3, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The region is set to Asia Pacific (Mumbai) and the user is prakash24.
- Terminal Editor:** A GNU nano 6.2 editor window titled `main.tf` containing the following Terraform script:

```
provider "aws" {
  region = "us-east-1"
  access_key = "AKIAZQ3DRJS3IUZKMLHV"
  secret_key = "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"
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "main-vpc"
  }
}

resource "aws_subnet" "public" {
  vpc_id = aws_vpc.main.id
  cidr_block = "10.0.1.0/24"
  map_public_ip_on_launch = true
}

resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.main.id
}

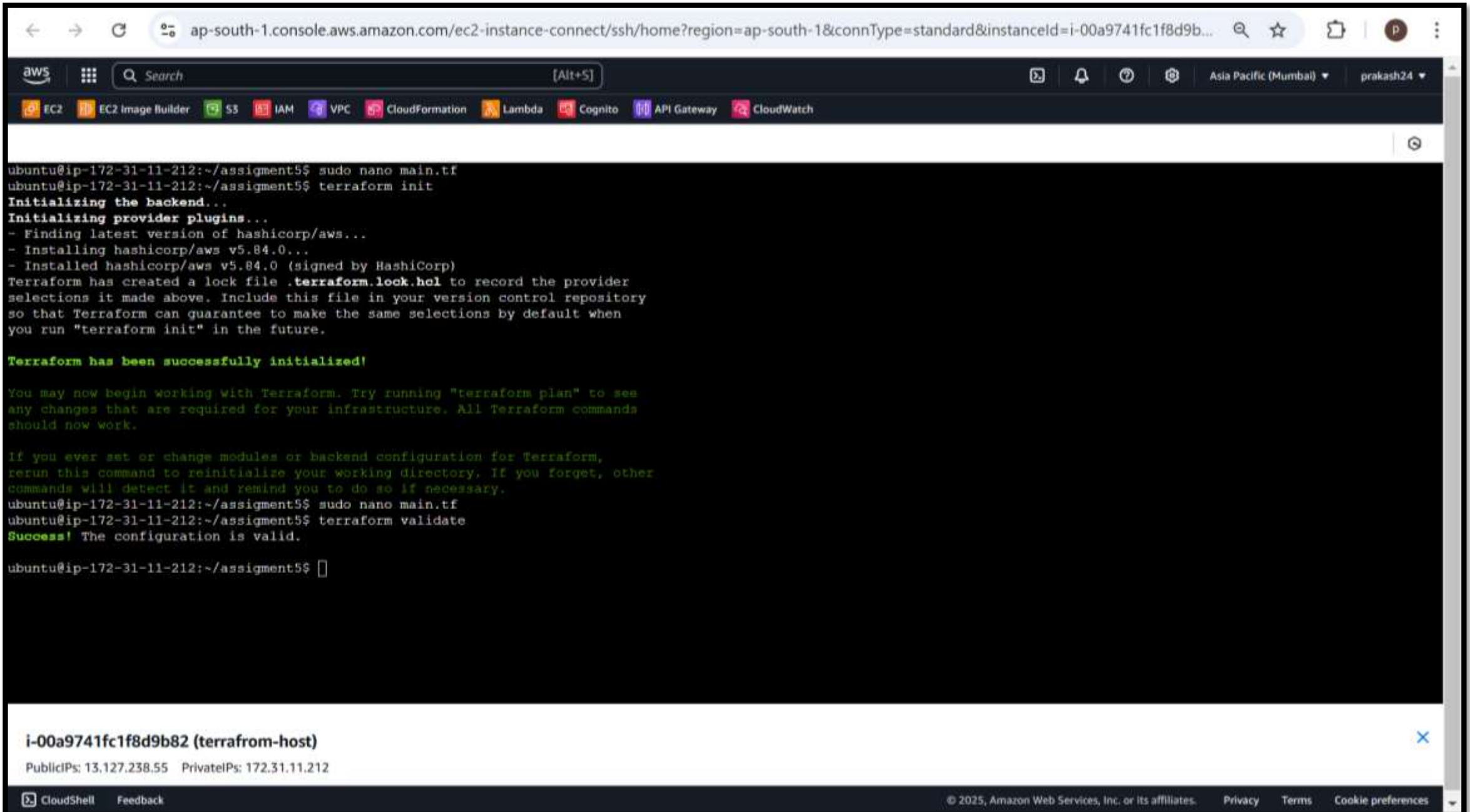
resource "aws_route_table" "public" {
  vpc_id = aws_vpc.main.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
}
```

The terminal shows a status bar indicating "[ Read 82 lines ]".

- Terminal Footer:** A table of keyboard shortcuts for the nano editor:

<code>^G</code> Help	<code>^O</code> Write Out	<code>^W</code> Where Is	<code>^K</code> Cut	<code>^T</code> Execute	<code>^C</code> Location	<code>M-U</code> Undo	<code>M-A</code> Set Mark	<code>M-I</code> To Bracket	<code>M-Q</code> Previous
<code>^X</code> Exit	<code>^R</code> Read File	<code>^\</code> Replace	<code>^U</code> Paste	<code>^J</code> Justify	<code>^/_</code> Go To Line	<code>M-E</code> Redo	<code>M-B</code> Copy	<code>^C</code> Where Was	<code>M-W</code> Next
- Instance Information:** A box at the bottom left shows the instance ID `i-00a9741fc1f8d9b82 (terrafrom-host)` and its IP addresses: Public IPs: 13.127.238.55, Private IPs: 172.31.11.212.
- Footer:** Includes a CloudShell logo, a feedback link, and a copyright notice: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

## Running Terraform init & Terraform Validate Command



The screenshot shows an AWS CloudShell terminal window. The browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-00a9741fc1f8d9b...`. The AWS navigation bar at the top includes the AWS logo, a search bar, and various service icons like EC2, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The user's profile is 'prakash24' in the 'Asia Pacific (Mumbai)' region.

The terminal output shows the following commands and results:

```
ubuntu@ip-172-31-11-212:~/assignment5$ sudo nano main.tf
ubuntu@ip-172-31-11-212:~/assignment5$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (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!

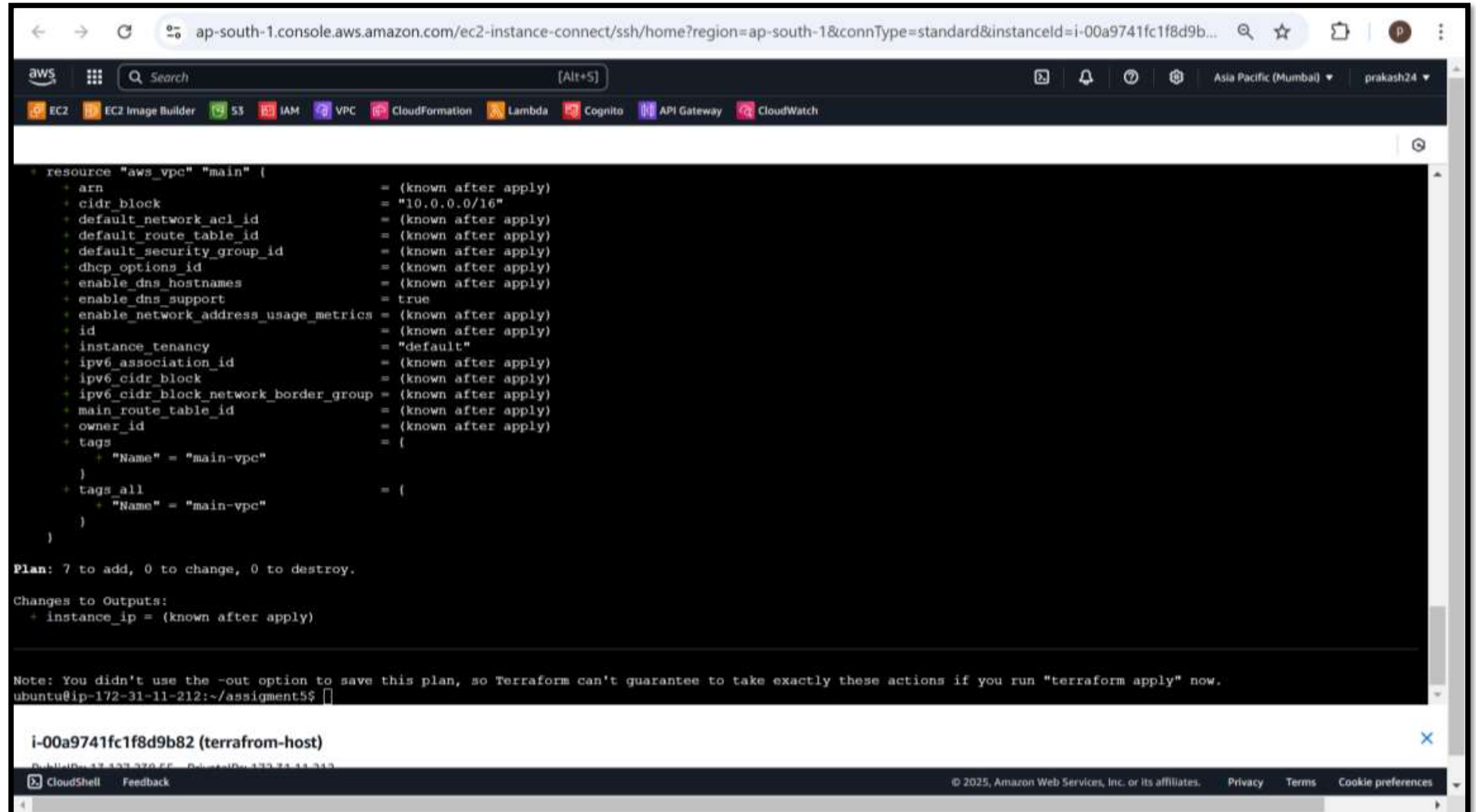
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

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-172-31-11-212:~/assignment5$ sudo nano main.tf
ubuntu@ip-172-31-11-212:~/assignment5$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-11-212:~/assignment5$
```

At the bottom of the terminal window, there is a status bar for the instance `i-00a9741fc1f8d9b82 (terraform-host)`, showing Public IPs: 13.127.238.55 and Private IPs: 172.31.11.212. The footer of the page includes 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for Privacy, Terms, and Cookie preferences.

## Running Terraform Plan Command



The screenshot shows the AWS CloudShell interface. At the top, the browser address bar displays the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-00a9741fc1f8d9b...`. The AWS console navigation bar is visible with various services like EC2, IAM, VPC, CloudFormation, Lambda, Cognito, API Gateway, and CloudWatch. The main area contains a terminal window with the following Terraform plan output:

```
+ resource "aws_vpc" "main" {
  + arn                                = (known after apply)
  + cidr_block                        = "10.0.0.0/16"
  + default_network_acl_id           = (known after apply)
  + default_route_table_id           = (known after apply)
  + default_security_group_id        = (known after apply)
  + dhcp_options_id                  = (known after apply)
  + enable_dns_hostnames              = (known after apply)
  + enable_dns_support                = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                               = (known after apply)
  + instance_tenancy                  = "default"
  + ipv6_association_id               = (known after apply)
  + ipv6_cidr_block                   = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id               = (known after apply)
  + owner_id                         = (known after apply)
  + tags                             = {
    + "Name" = "main-vpc"
  }
  + tags_all                         = {
    + "Name" = "main-vpc"
  }
}
```

Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:

```
+ instance_ip = (known after apply)
```

Note: You didn't use the `-out` option to save this plan, so Terraform can't guarantee to take exactly these actions if you run `"terraform apply"` now.

ubuntu@ip-172-31-11-212:~/assignment5\$

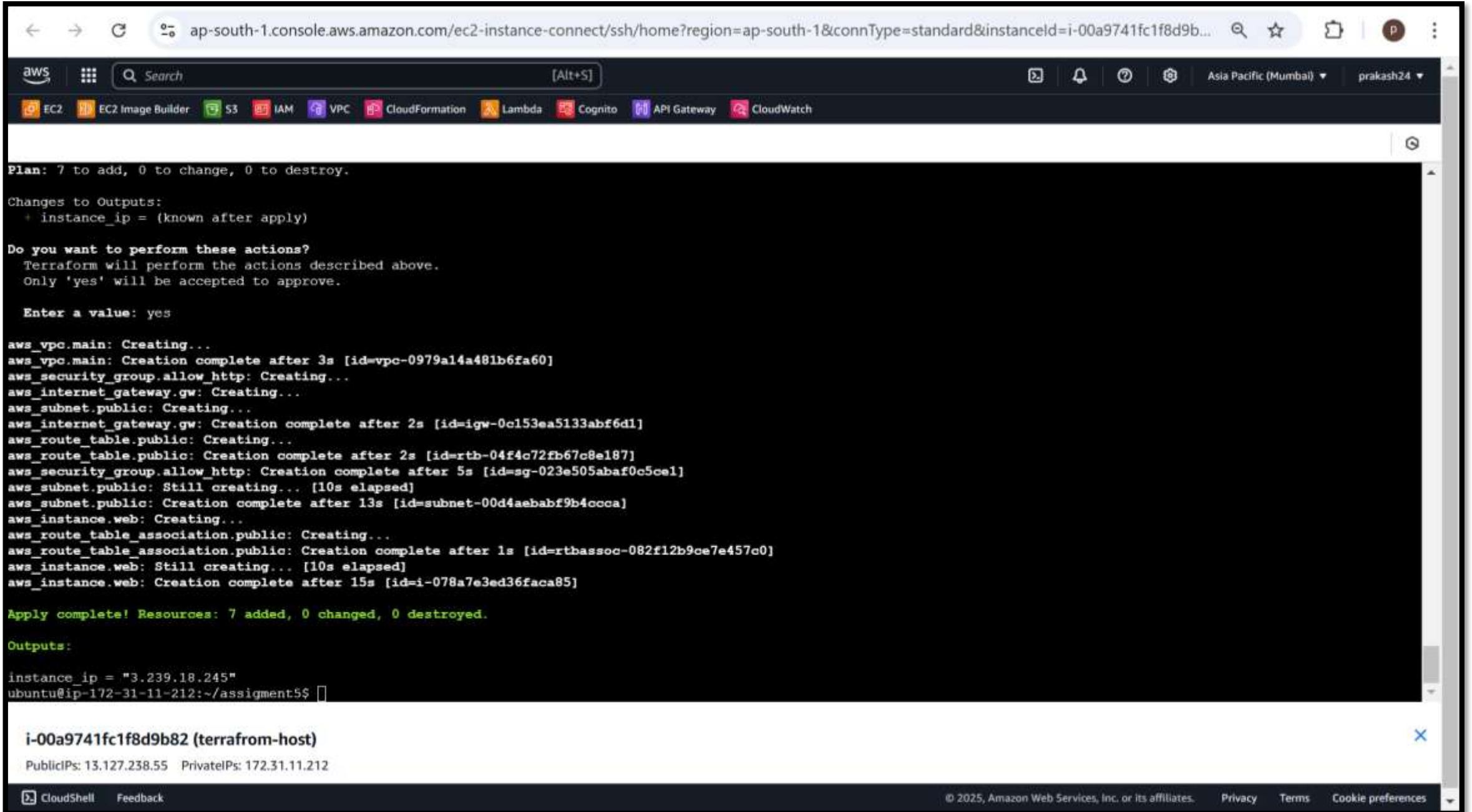
i-00a9741fc1f8d9b82 (terraform-host)

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



## Running Terraform Apply Command



The screenshot shows the AWS Management Console interface with a terminal window open. The terminal displays the output of a Terraform apply command. The output indicates that 7 resources were added, 0 were changed, and 0 were destroyed. The resources created include an AWS VPC, an Internet Gateway, a Subnet, a Route Table, a Security Group, and an EC2 Instance. The terminal also shows the IP addresses for the instance: PublicIPs: 13.127.238.55 and PrivateIPs: 172.31.11.212.

```
Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_ip = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.main: Creating...
aws_vpc.main: Creation complete after 3s [id=vpc-0979a14a481b6fa60]
aws_security_group.allow_http: Creating...
aws_internet_gateway.gw: Creating...
aws_subnet.public: Creating...
aws_internet_gateway.gw: Creation complete after 2s [id=igw-0c153ea5133abf6d1]
aws_route_table.public: Creating...
aws_route_table.public: Creation complete after 2s [id=rtb-04f4c72fb67c8e187]
aws_security_group.allow_http: Creation complete after 5s [id=sg-023e505abaf0c5ce1]
aws_subnet.public: Still creating... [10s elapsed]
aws_subnet.public: Creation complete after 13s [id=subnet-00d4aebabf9b4ccca]
aws_instance.web: Creating...
aws_route_table_association.public: Creating...
aws_route_table_association.public: Creation complete after 1s [id=rtbassoc-082f12b9ce7e457c0]
aws_instance.web: Still creating... [10s elapsed]
aws_instance.web: Creation complete after 15s [id=i-078a7e3ed36faca85]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

instance_ip = "3.239.18.245"
ubuntu@ip-172-31-11-212:~/assignment5$
```

i-00a9741fc1f8d9b82 (terraform-host)

PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Verifying a EC2 instance with Apache2 server launched by terraform

The screenshot displays the AWS Management Console for the us-east-1 region. The left-hand navigation pane includes sections for Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled 'Instances (1/1)' and shows a single instance named 'web-server' with ID 'i-078a7e3ed36faca85'. The instance is in a 'Running' state, using a 't2.micro' instance type, and is located in the 'us-east-1c' Availability Zone. Below the instance list, the 'Details' tab is selected, showing various attributes such as Instance ID, IPv6 address, Hostname type, Answer private resource DNS name, Auto-assigned IP address, IAM Role, Public IPv4 address, Instance state, Private IP DNS name, Instance type, VPC ID, Subnet ID, Private IPv4 addresses, Public IPv4 DNS, Elastic IP addresses, and AWS Compute Optimizer finding.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;\$case=tags:true%5C,client:false;\$regex=tags:false%5C,client:false

Search [Alt+S]

United States (N. Virginia) prakash24

EC2 EC2 Image Builder S3 IAM VPC CloudFormation Lambda Cognito API Gateway CloudWatch

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive) All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
web-server	i-078a7e3ed36faca85	Running	t2.micro	Initializing	View alarms +	us-east-1c	-

i-078a7e3ed36faca85 (web-server)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary Info

Instance ID  
i-078a7e3ed36faca85

IPv6 address  
-

Hostname type  
IP name: ip-10-0-1-47.ec2.internal

Answer private resource DNS name  
-

Auto-assigned IP address  
3.239.18.245 [Public IP]

IAM Role  
-

Public IPv4 address  
3.239.18.245 | open address

Instance state  
Running

Private IP DNS name (IPv4 only)  
ip-10-0-1-47.ec2.internal

Instance type  
t2.micro

VPC ID  
vpc-0979a14a481b6fa60 (main-vpc)

Subnet ID  
subnet-00d4a9b9b9b4ccca

Private IPv4 addresses  
10.0.1.47

Public IPv4 DNS  
-

Elastic IP addresses  
-

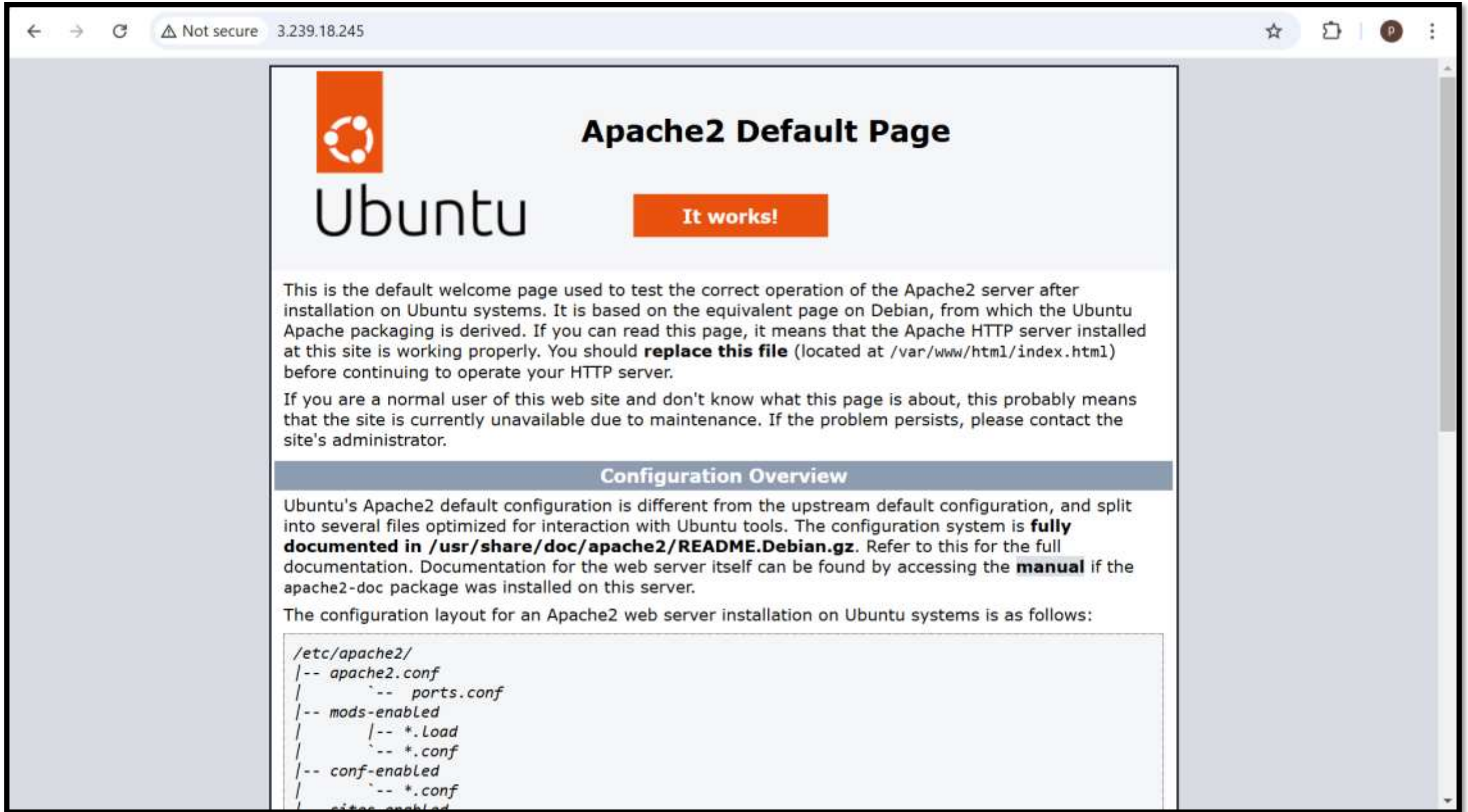
AWS Compute Optimizer finding  
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name  
-


CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Verifying Apache2 server launched by terraform



← → ↻ ⚠ Not secure 3.239.18.245 ☆ 📁 | p ⋮



# Ubuntu

## Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

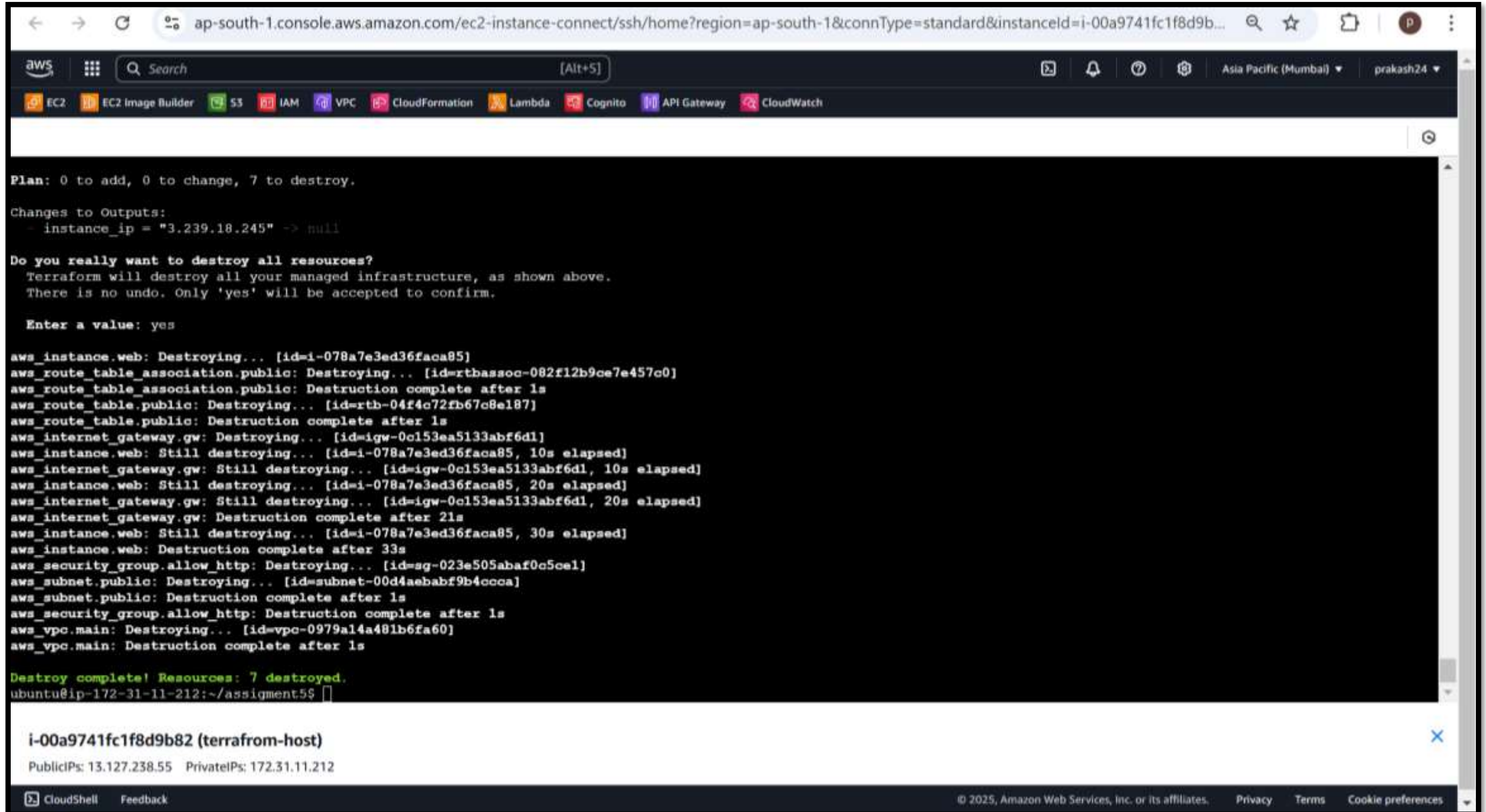
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
```



## Destroying Resource launched by terraform using terraform destroy command



The screenshot shows the AWS Management Console interface with a CloudShell terminal window open. The terminal displays the output of a Terraform destroy command. The output includes a plan summary, a confirmation prompt, and a detailed list of resources being destroyed, such as EC2 instances, route tables, internet gateways, security groups, and subnets. The terminal also shows the public and private IP addresses of the instance being destroyed.

Plan: 0 to add, 0 to change, 7 to destroy.

Changes to Outputs:

- instance\_ip = "3.239.18.245" -> null

Do you really want to destroy all resources?  
Terraform will destroy all your managed infrastructure, as shown above.  
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws\_instance.web: Destroying... [id=i-078a7e3ed36faca85]  
aws\_route\_table\_association.public: Destroying... [id=rtbassoc-082f12b9ce7e457c0]  
aws\_route\_table\_association.public: Destruction complete after 1s  
aws\_route\_table.public: Destroying... [id=rtb-04f4c72fb67c8e187]  
aws\_route\_table.public: Destruction complete after 1s  
aws\_internet\_gateway.gw: Destroying... [id=igw-0c153ea5133abf6d1]  
aws\_instance.web: Still destroying... [id=i-078a7e3ed36faca85, 10s elapsed]  
aws\_internet\_gateway.gw: Still destroying... [id=igw-0c153ea5133abf6d1, 10s elapsed]  
aws\_instance.web: Still destroying... [id=i-078a7e3ed36faca85, 20s elapsed]  
aws\_internet\_gateway.gw: Still destroying... [id=igw-0c153ea5133abf6d1, 20s elapsed]  
aws\_internet\_gateway.gw: Destruction complete after 21s  
aws\_instance.web: Still destroying... [id=i-078a7e3ed36faca85, 30s elapsed]  
aws\_instance.web: Destruction complete after 33s  
aws\_security\_group.allow\_http: Destroying... [id=sg-023e505abaf0c5ce1]  
aws\_subnet.public: Destroying... [id=subnet-00d4aebabf9b4ccca]  
aws\_subnet.public: Destruction complete after 1s  
aws\_security\_group.allow\_http: Destruction complete after 1s  
aws\_vpc.main: Destroying... [id=vpc-0979a14a481b6fa60]  
aws\_vpc.main: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.  
ubuntu@ip-172-31-11-212:~/assignment5\$

i-00a9741fc1f8d9b82 (terraform-host)  
PublicIPs: 13.127.238.55 PrivateIPs: 172.31.11.212

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences