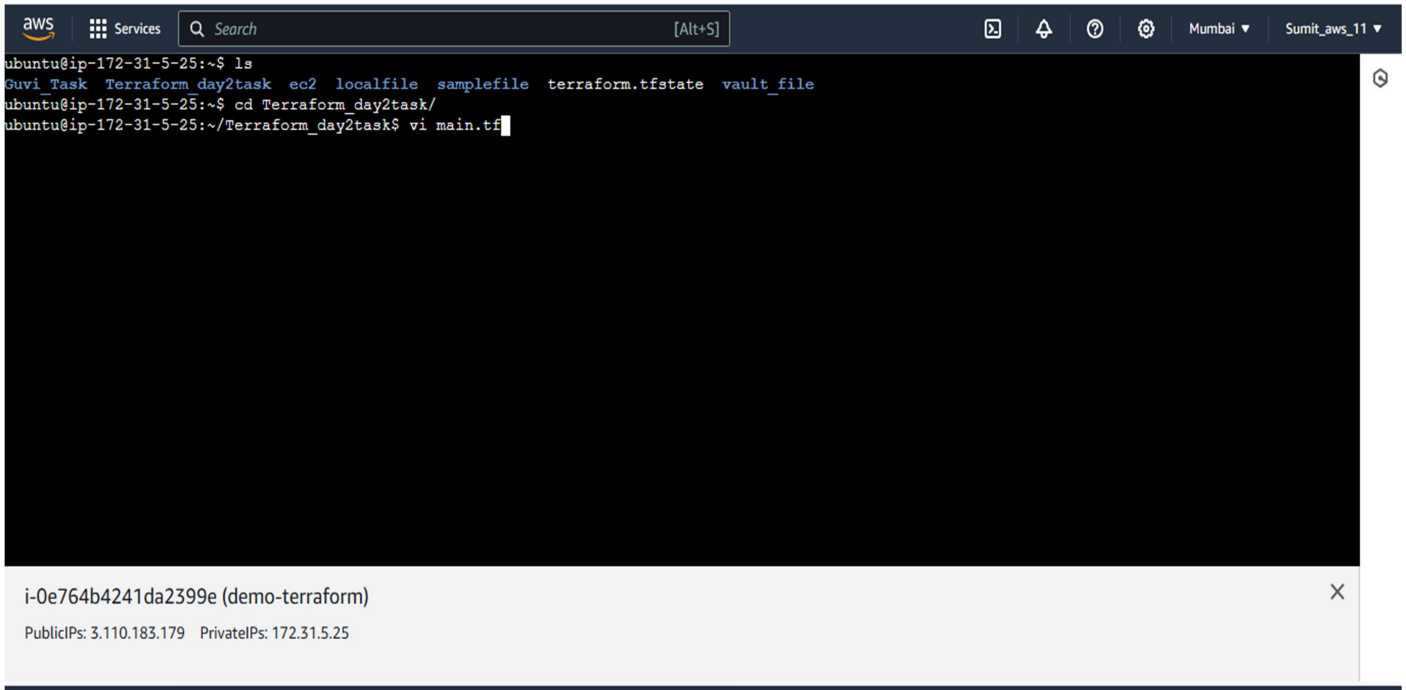
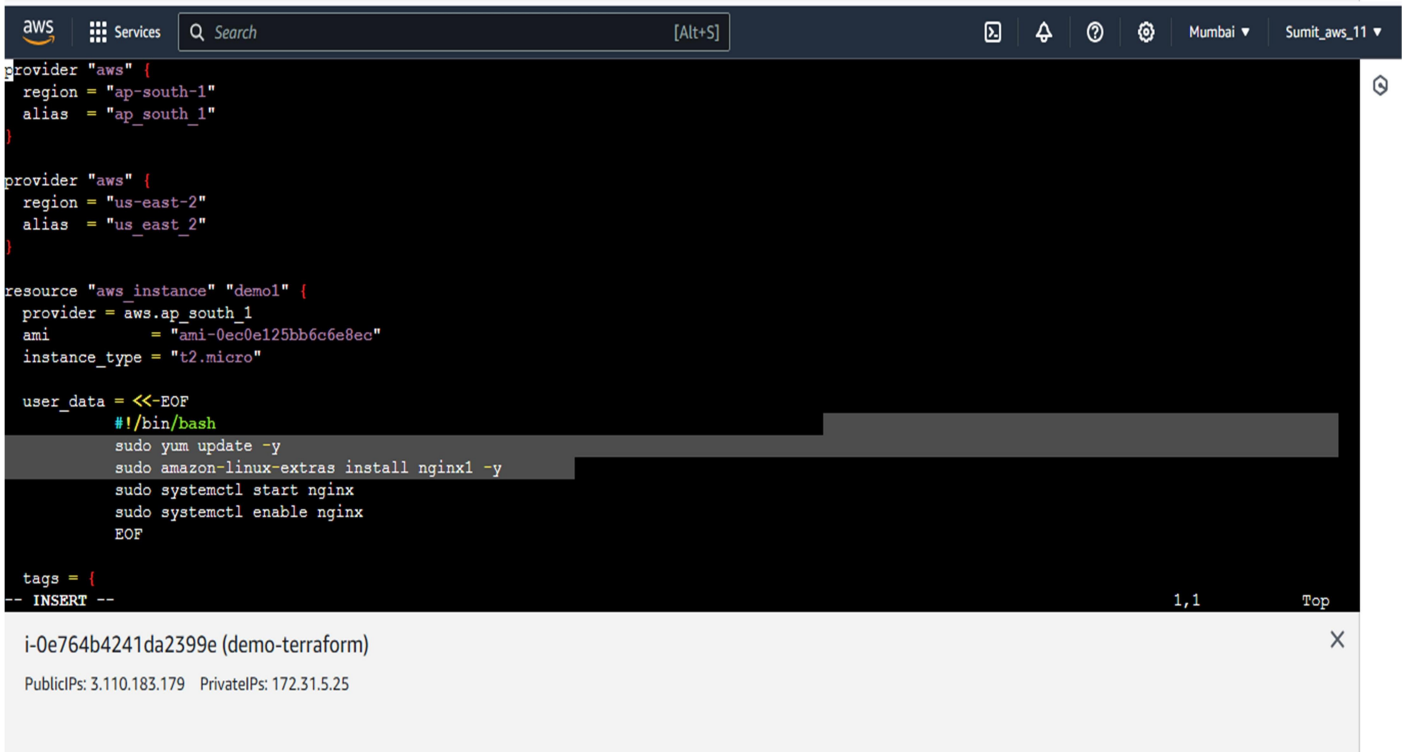
Name:- Sumit Dhabalddeb
Email id :-
sumitdhal2000@gmail.com
Course :- DevOps , ID – CT6WDS1258

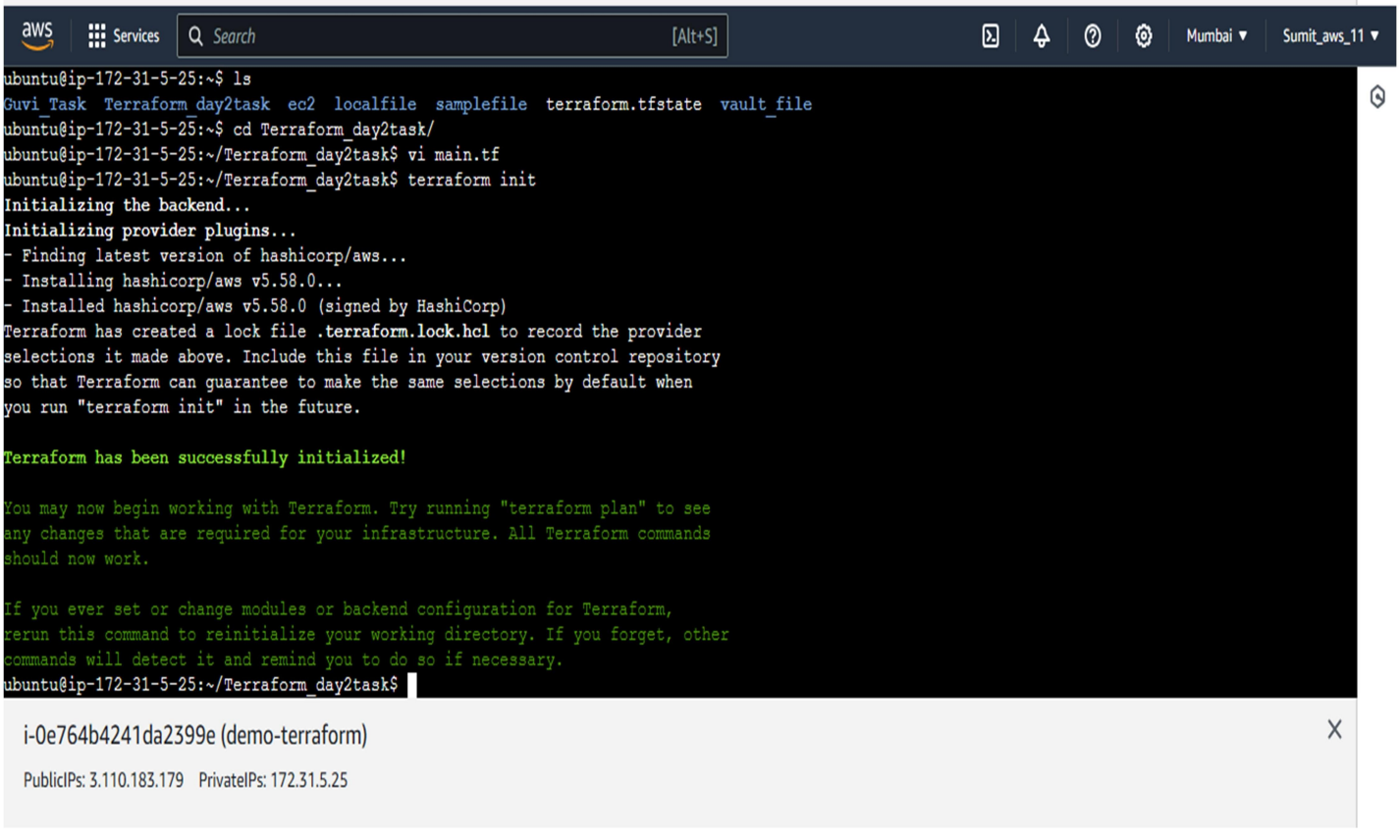
Terraform

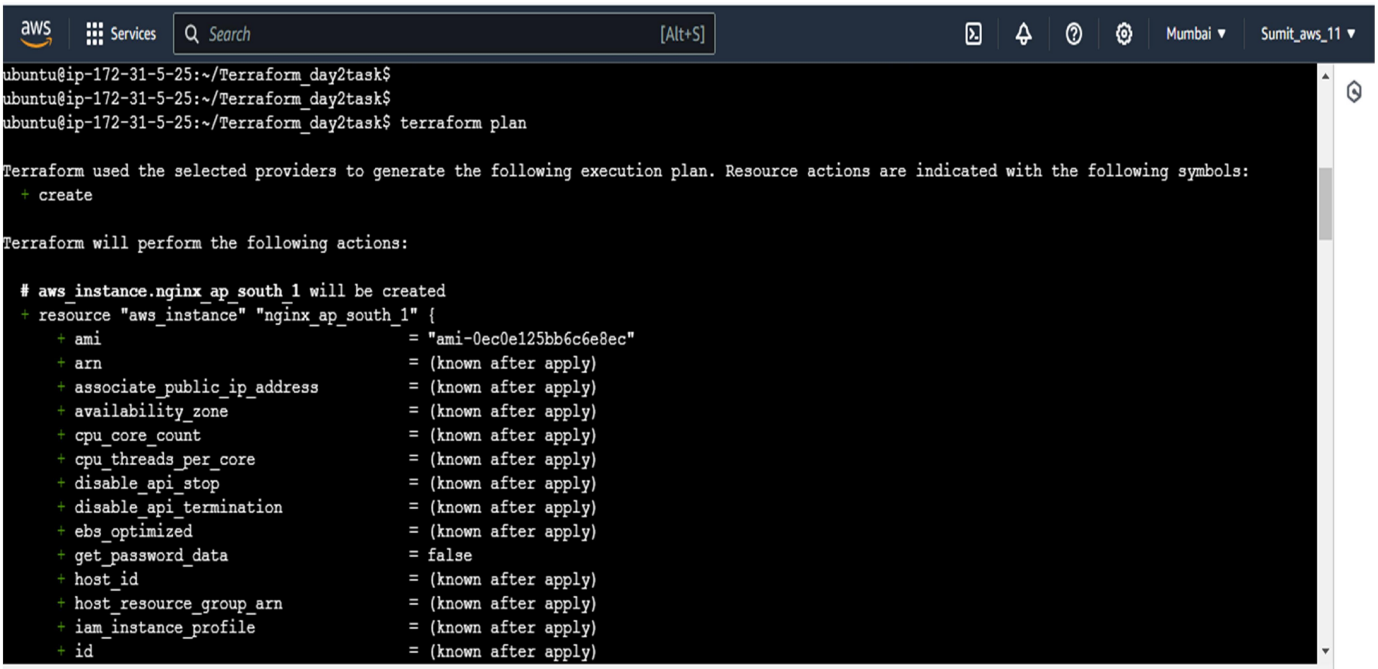
TASK: Use terraform to create and manage infrastructure as code (Iac). This project helps you learn how to automate the prpvisioning of cloud resources. Write Terraform configuration files to provision infrastructure (e.g AWS EC2 Instance). Apply and Destroy infrastructure using Terraform Command

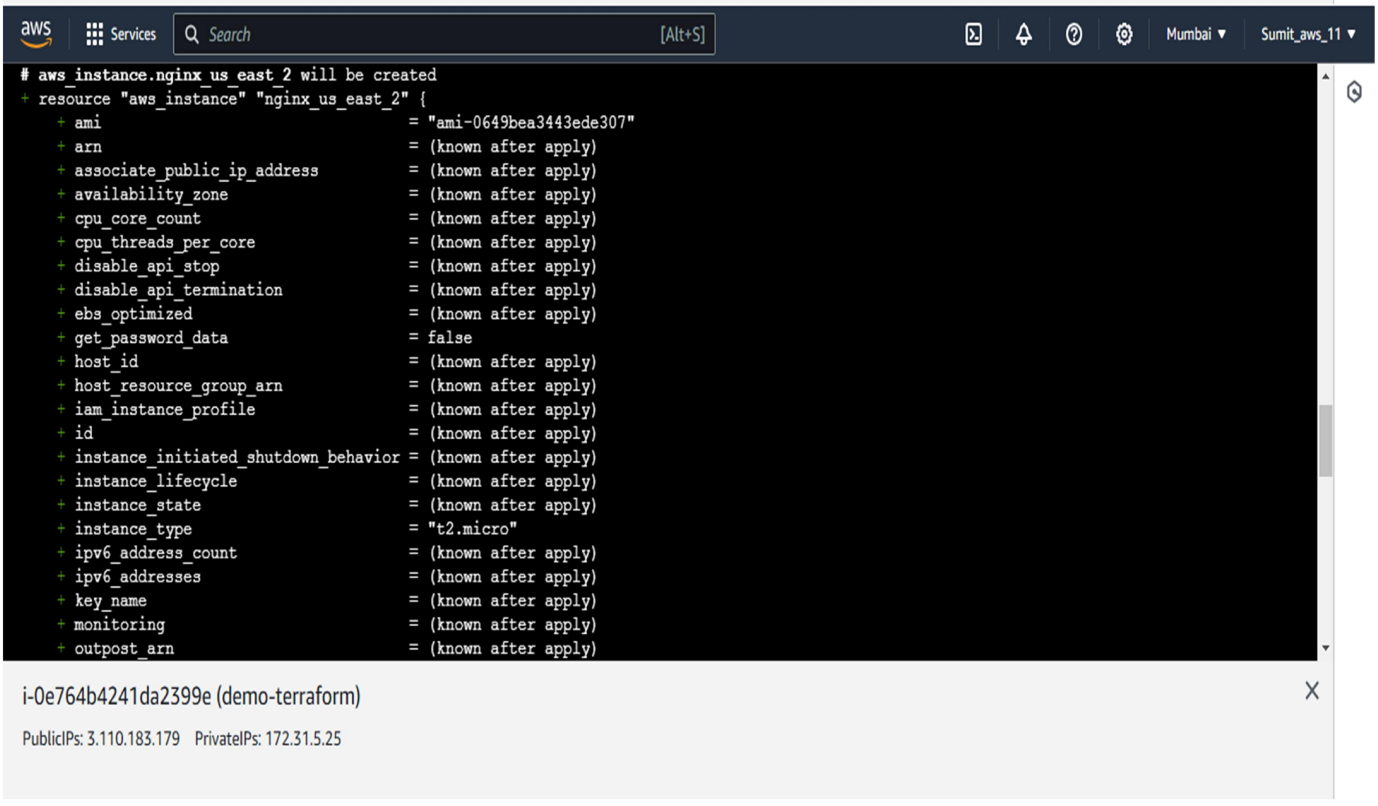


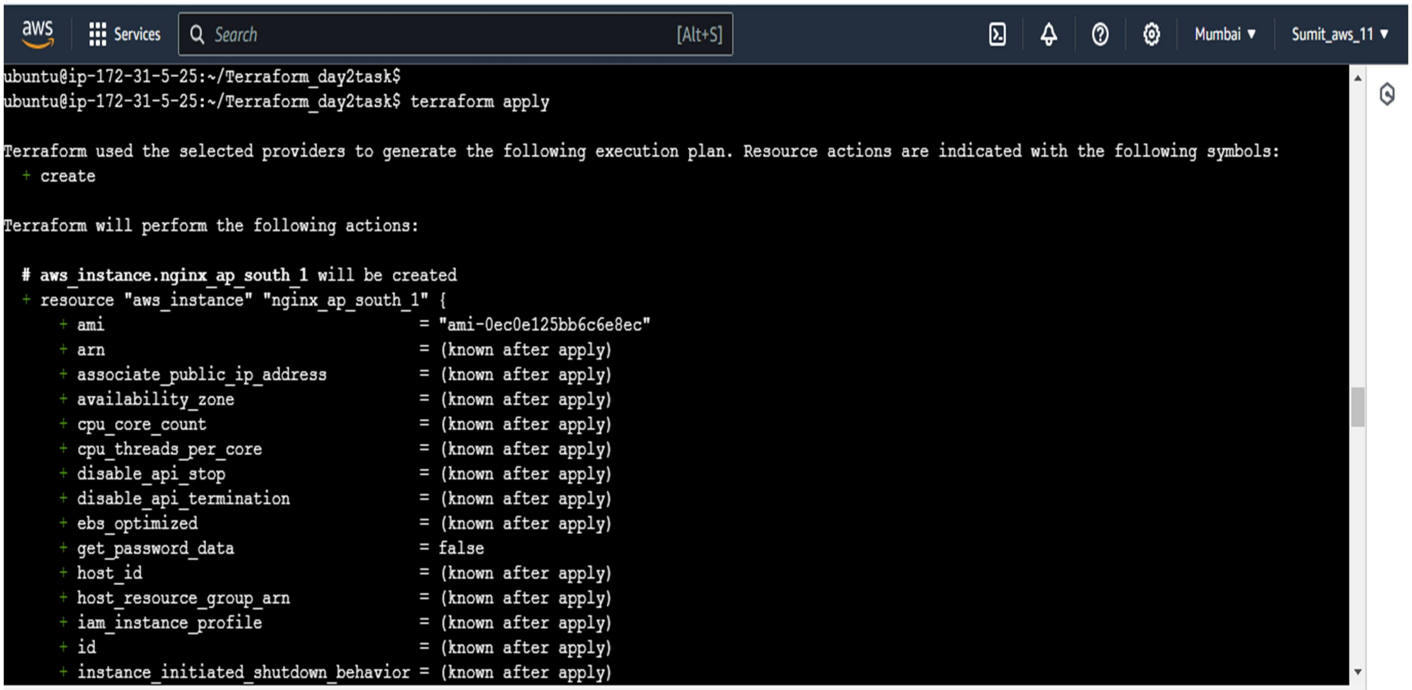


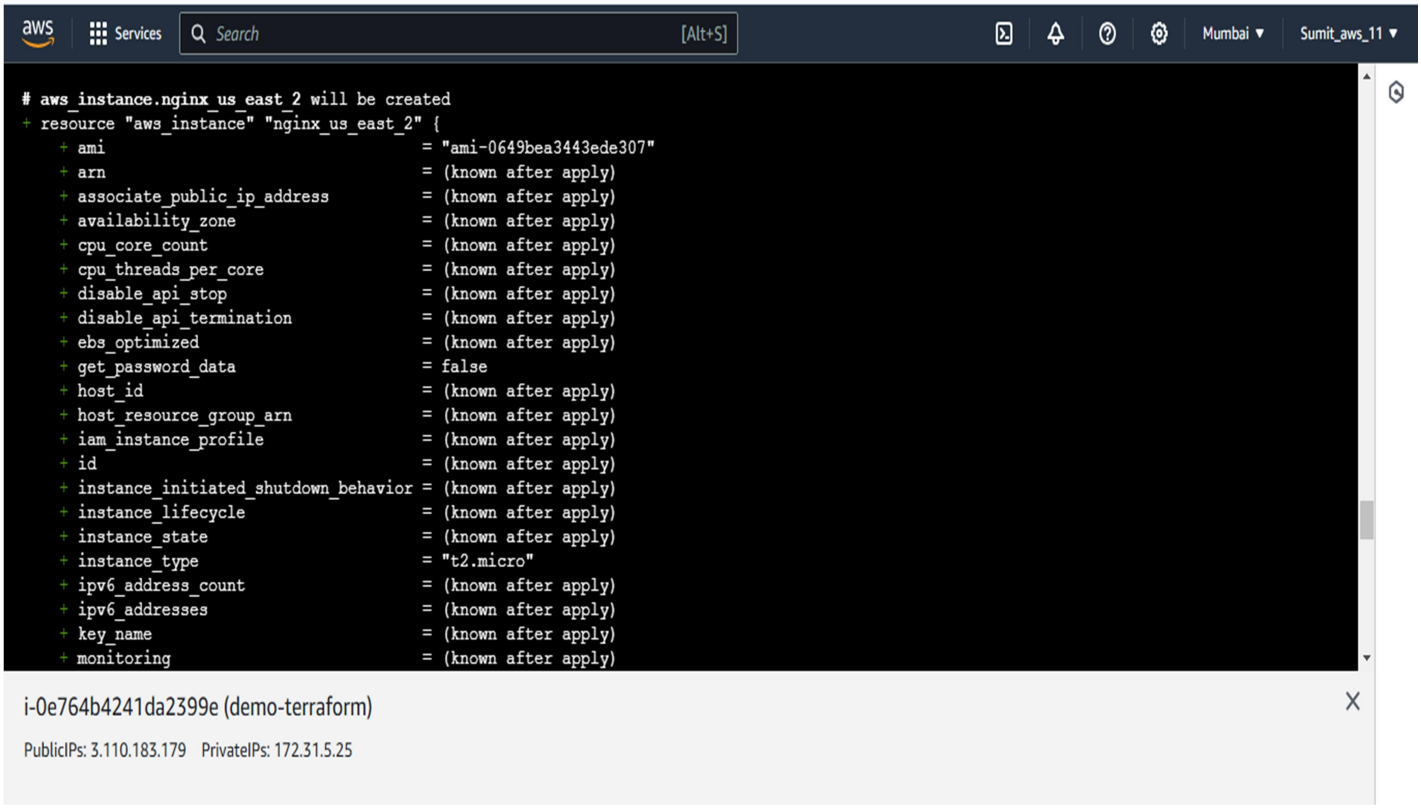

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
tags = {
Name = "nginxserver-ap-south-1"
}
}
resource "aws_instance" "demo2" {
provider = aws.us_east_2
ami = "ami-0649bea3443ede307"
instance_type = "t2.micro"
user_data = <<-EOF
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install nginx1 -y
sudo systemctl start nginx
sudo systemctl enable nginx
EOF
tags = {
Name = "nginxserver-us-east-2"
}
}
output "ap_south_1_instance_id" {
-- INSERT --
40,1
57%
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

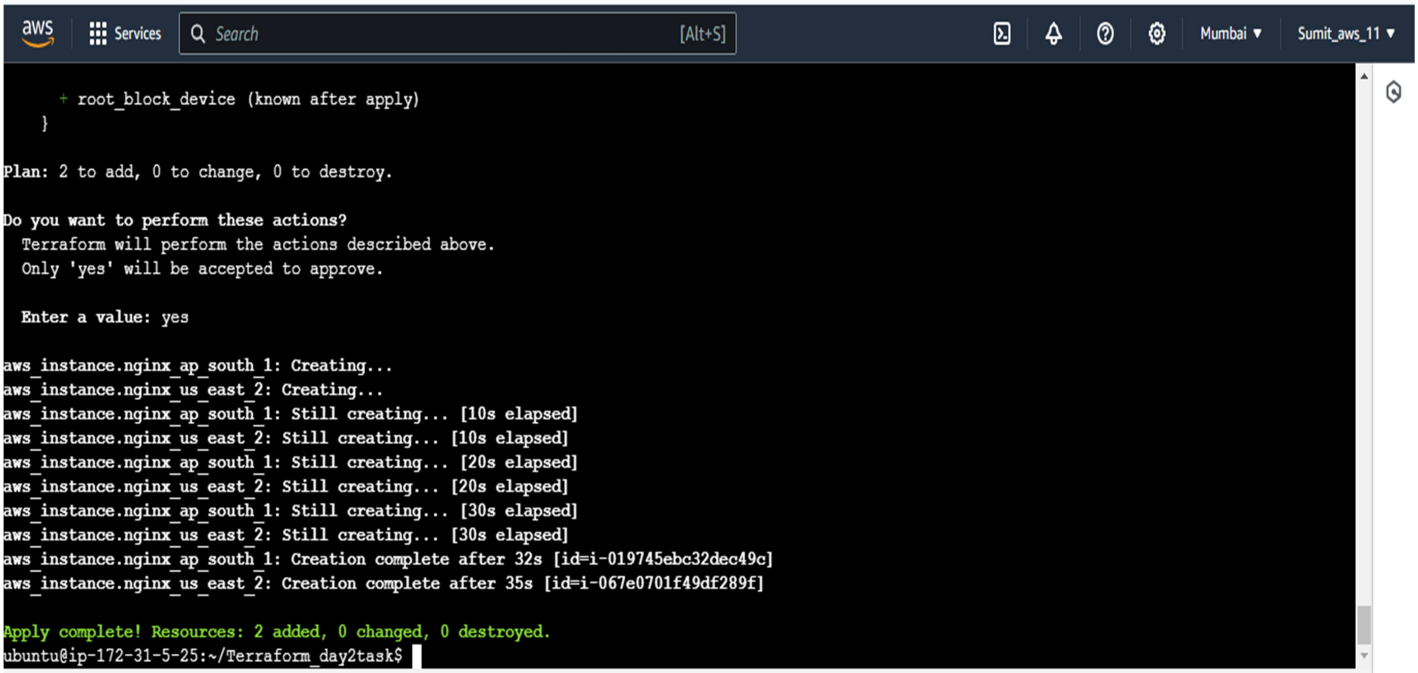

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
ubuntu@ip-172-31-5-25:~$ ls
Guvi_Task Terraform_day2task ec2 localfile samplefile terraform.tfstate vault_file
ubuntu@ip-172-31-5-25:~$ cd Terraform_day2task/
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ vi main.tf
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.58.0...
- Installed hashicorp/aws v5.58.0 (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-172-31-5-25:~/Terraform_day2task$
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

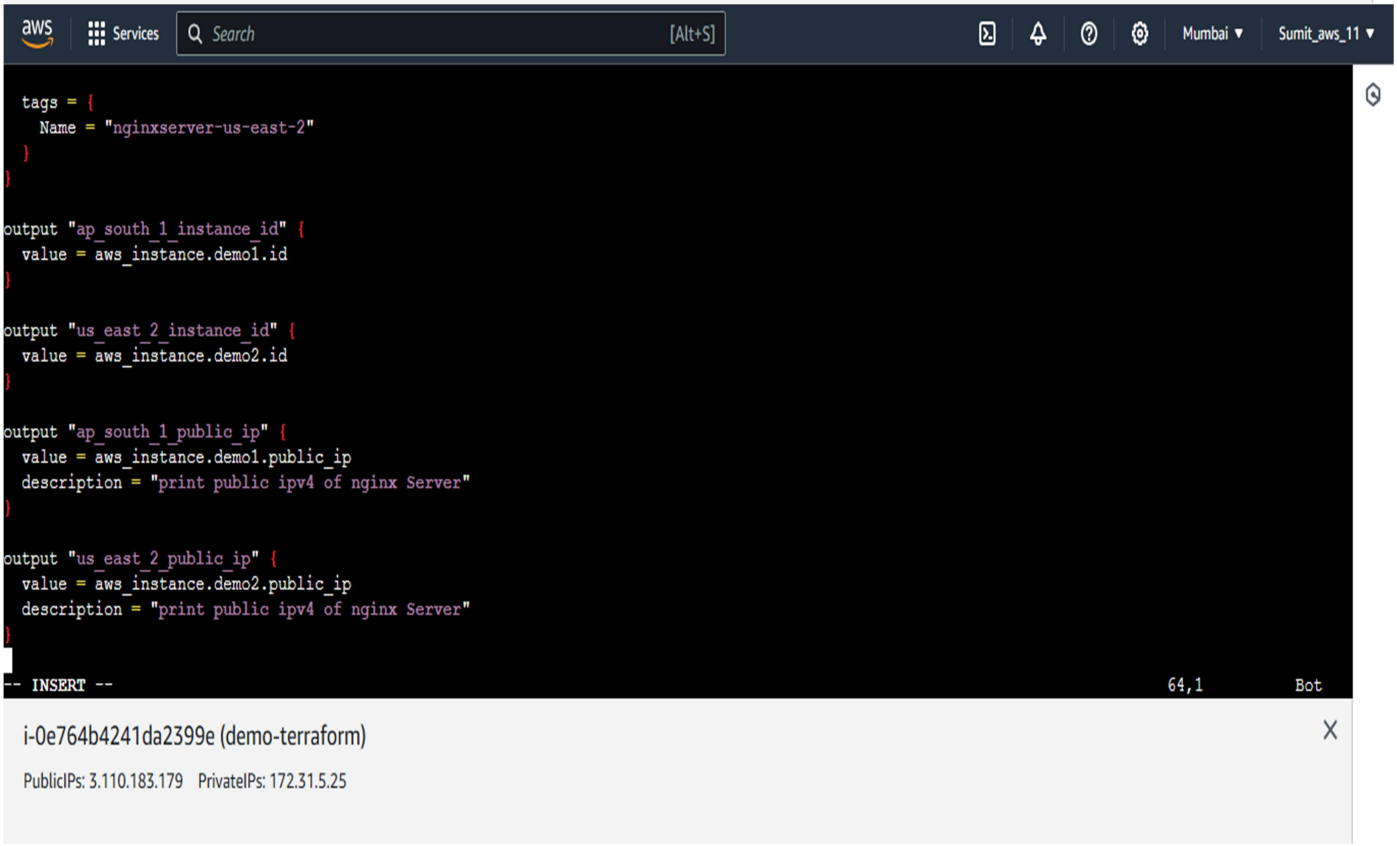

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.nginx_ap_south_1 will be created
+ resource "aws_instance" "nginx_ap_south_1" {
+ ami = "ami-0ec0e125bb6c6e8ec"
+ 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)
+ 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)
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

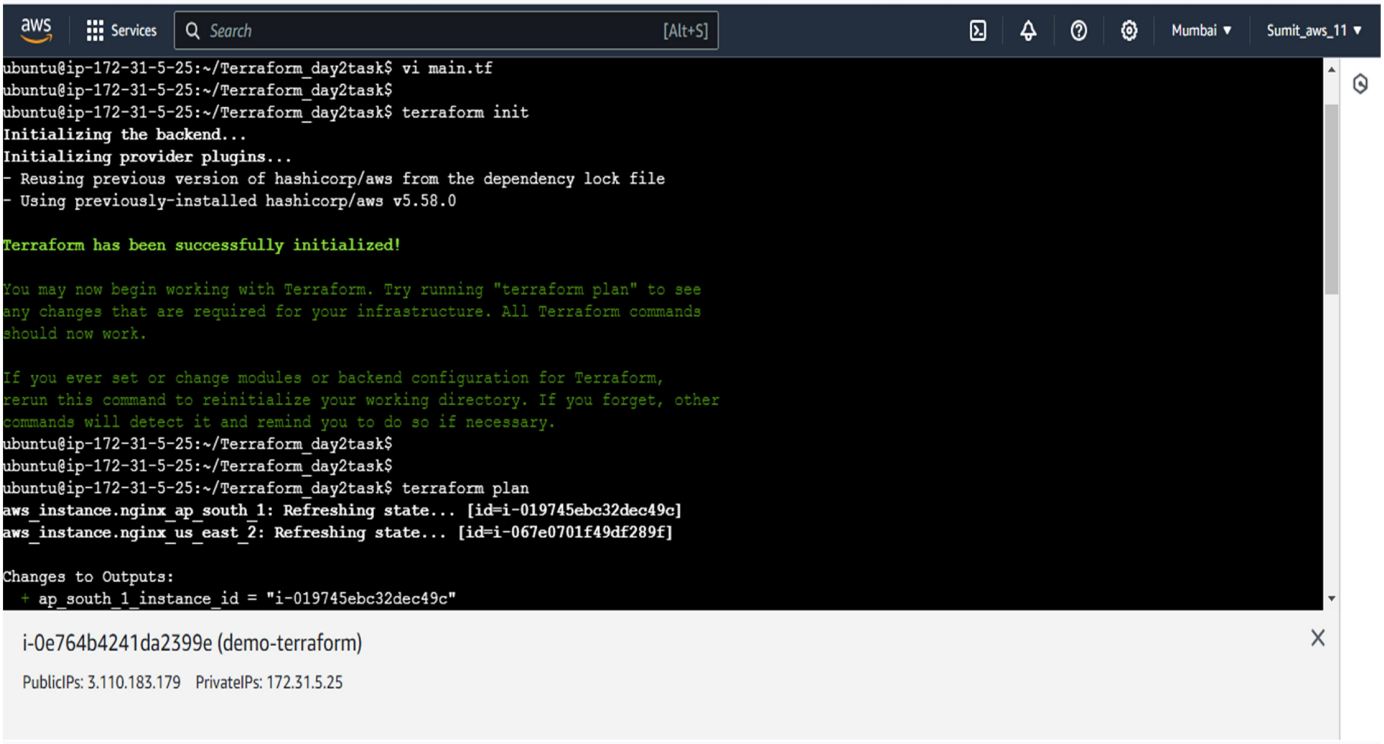

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
# aws_instance.nginx_us_east_2 will be created
+ resource "aws_instance" "nginx_us_east_2" {
+ ami = "ami-0649bea3443ede307"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (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)
+ 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)
+ monitoring = (known after apply)
+ outpost_arn = (known after apply)
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

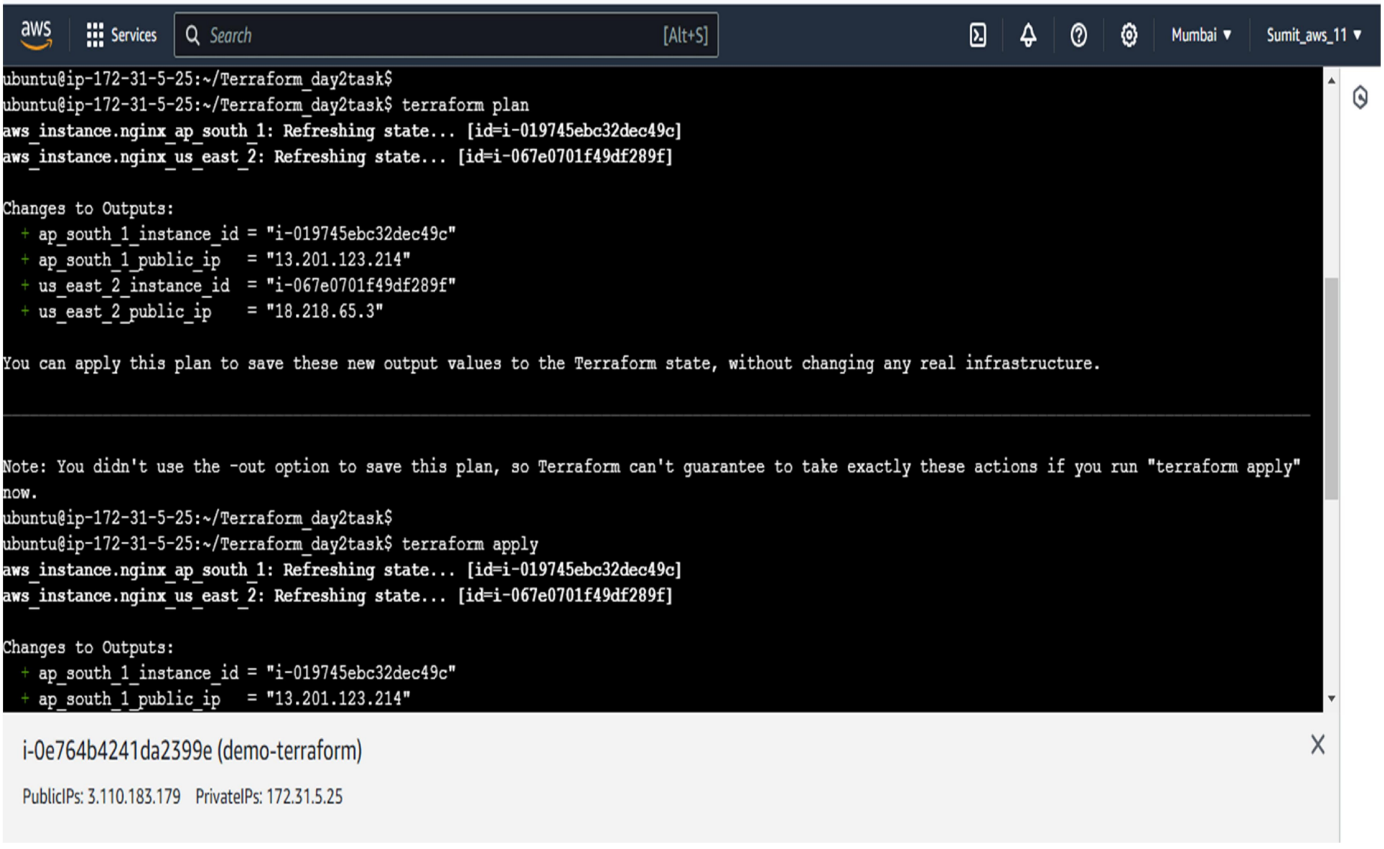

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.nginx_ap_south_1 will be created
+ resource "aws_instance" "nginx_ap_south_1" {
+ ami = "ami-0ec0e125bb6c6e8ec"
+ 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)
+ 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)
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

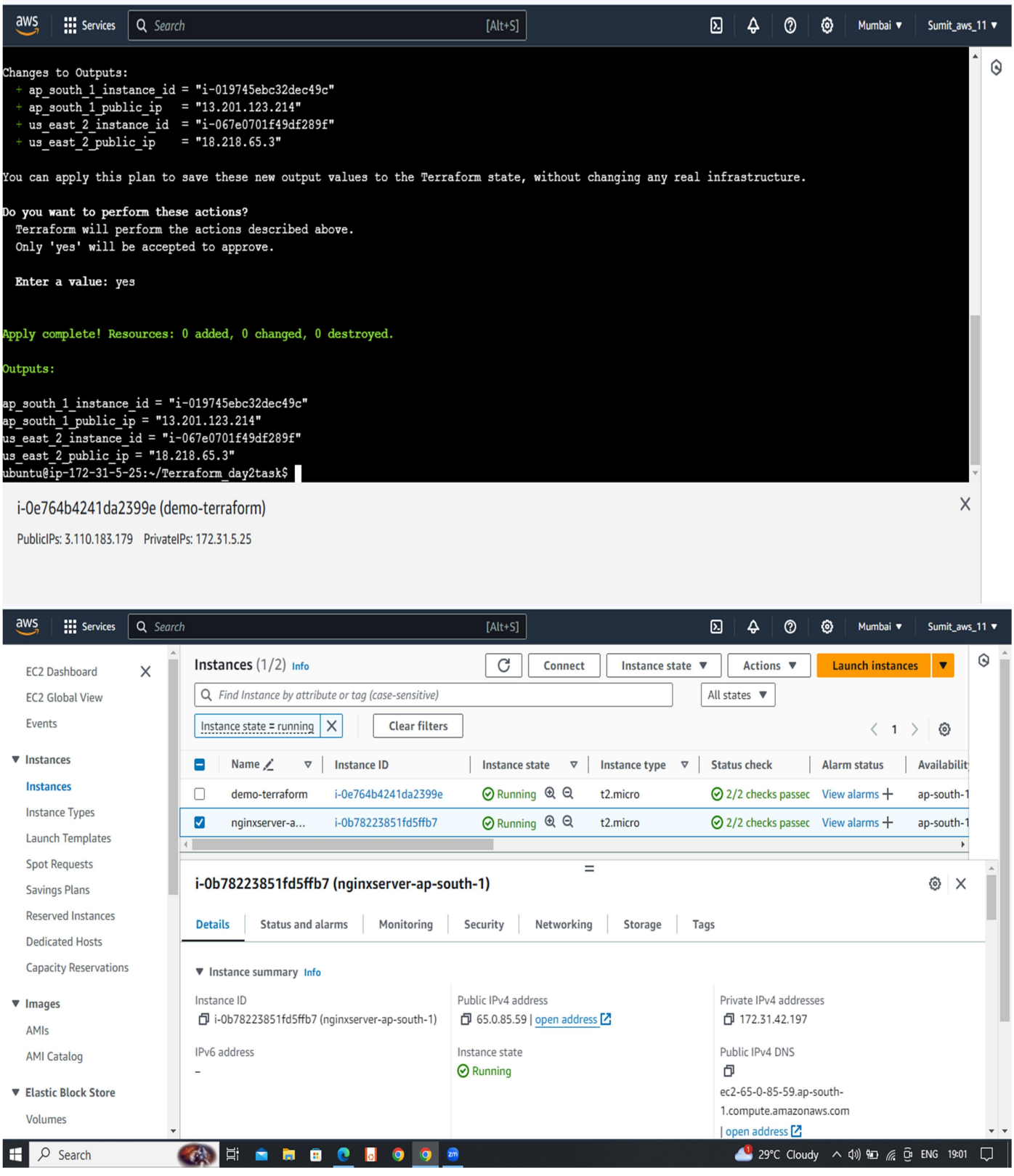

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
# aws_instance.nginx_us_east_2 will be created
+ resource "aws_instance" "nginx_us_east_2" {
+ ami = "ami-0649bea3443ede307"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (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)
+ 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)
+ monitoring = (known after apply)
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25


aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
+ root_block_device (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_instance.nginx_ap_south_1: Creating...
aws_instance.nginx_us_east_2: Creating...
aws_instance.nginx_ap_south_1: Still creating... [10s elapsed]
aws_instance.nginx_us_east_2: Still creating... [10s elapsed]
aws_instance.nginx_ap_south_1: Still creating... [20s elapsed]
aws_instance.nginx_us_east_2: Still creating... [20s elapsed]
aws_instance.nginx_ap_south_1: Still creating... [30s elapsed]
aws_instance.nginx_us_east_2: Still creating... [30s elapsed]
aws_instance.nginx_ap_south_1: Creation complete after 32s [id=i-019745ebc32dec49c]
aws_instance.nginx_us_east_2: Creation complete after 35s [id=i-067e0701f49df289f]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

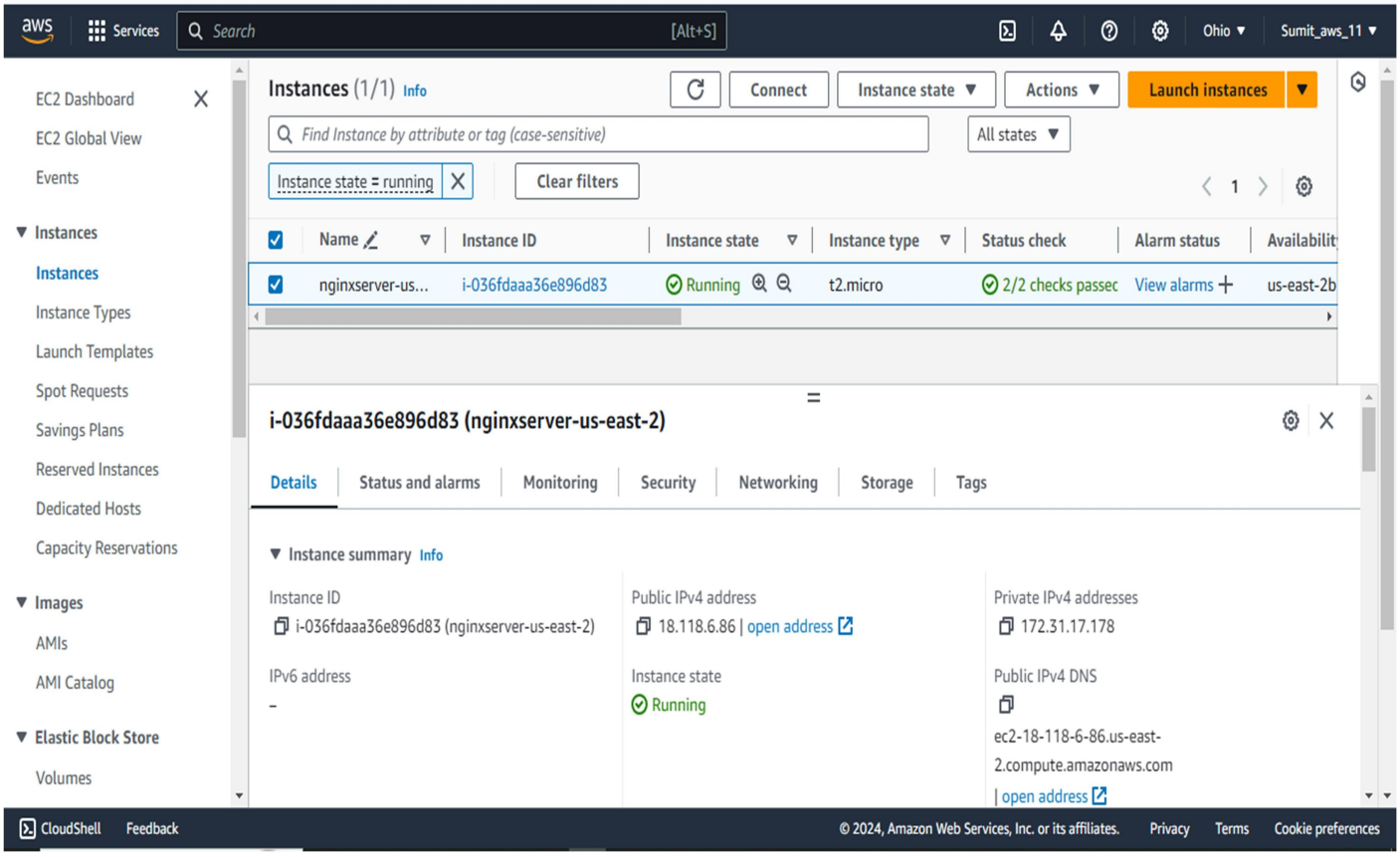

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
tags = {
Name = "nginxserver-us-east-2"
}
}
output "ap_south_1_instance_id" {
value = aws_instance.demo1.id
}
output "us_east_2_instance_id" {
value = aws_instance.demo2.id
}
output "ap_south_1_public_ip" {
value = aws_instance.demo1.public_ip
description = "print public ipv4 of nginx Server"
}
output "us_east_2_public_ip" {
value = aws_instance.demo2.public_ip
description = "print public ipv4 of nginx Server"
}
-- INSERT --
64,1
Bot
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

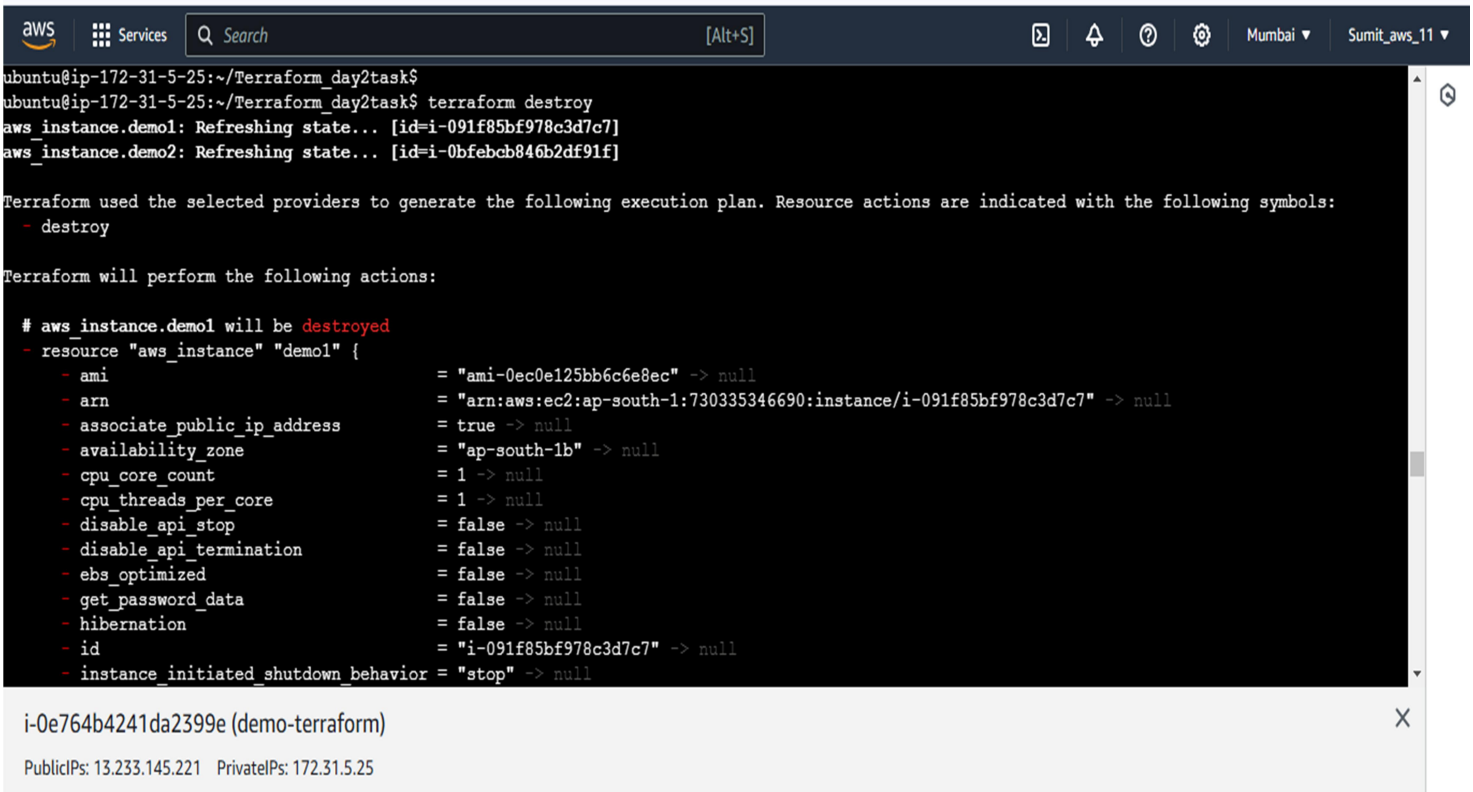

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ vi main.tf
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.58.0
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-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ terraform plan
aws_instance.nginx_ap_south_1: Refreshing state... [id=i-019745ebc32dec49c]
aws_instance.nginx_us_east_2: Refreshing state... [id=i-067e0701f49df289f]
Changes to Outputs:
+ ap_south_1_instance_id = "i-019745ebc32dec49c"
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

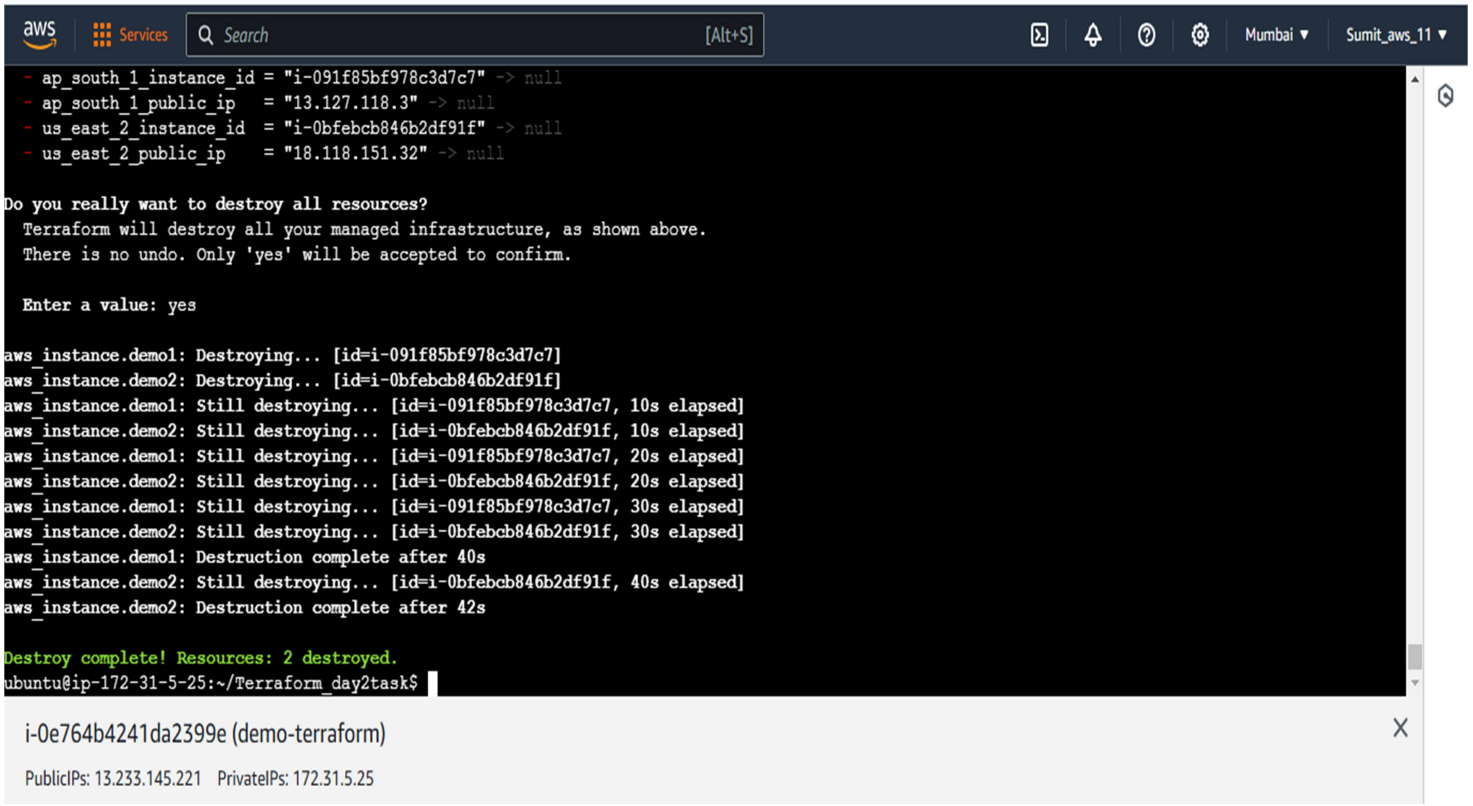

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ terraform plan
aws_instance.nginx_ap_south_1: Refreshing state... [id=i-019745ebc32dec49c]
aws_instance.nginx_us_east_2: Refreshing state... [id=i-067e0701f49df289f]
Changes to Outputs:
+ ap_south_1_instance_id = "i-019745ebc32dec49c"
+ ap_south_1_public_ip = "13.201.123.214"
+ us_east_2_instance_id = "i-067e0701f49df289f"
+ us_east_2_public_ip = "18.218.65.3"
You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.
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-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ terraform apply
aws_instance.nginx_ap_south_1: Refreshing state... [id=i-019745ebc32dec49c]
aws_instance.nginx_us_east_2: Refreshing state... [id=i-067e0701f49df289f]
Changes to Outputs:
+ ap_south_1_instance_id = "i-019745ebc32dec49c"
+ ap_south_1_public_ip = "13.201.123.214"
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25


aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
Changes to Outputs:
+ ap_south_1_instance_id = "i-019745ebc32dec49c"
+ ap_south_1_public_ip = "13.201.123.214"
+ us_east_2_instance_id = "i-067e0701f49df289f"
+ us_east_2_public_ip = "18.218.65.3"
You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.
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
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
Outputs:
ap_south_1_instance_id = "i-019745ebc32dec49c"
ap_south_1_public_ip = "13.201.123.214"
us_east_2_instance_id = "i-067e0701f49df289f"
us_east_2_public_ip = "18.218.65.3"
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 3.110.183.179 PrivateIPs: 172.31.5.25

aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
EC2 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
Instances (1/2) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
demo-terraform
i-0e764b4241da2399e
Running
t2.micro
2/2 checks passec
View alarms
nginxserver-a...
i-0b78223851fd5ffb7
Running
t2.micro
2/2 checks passec
View alarms
i-0b78223851fd5ffb7 (nginxserver-ap-south-1)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0b78223851fd5ffb7 (nginxserver-ap-south-1)
Public IPv4 address
65.0.85.59 | open address
Private IPv4 addresses
172.31.42.197
IPv6 address
-
Instance state
Running
Public IPv4 DNS
ec2-65-0-85-59.ap-south-1.compute.amazonaws.com | open address
Search
29°C Cloudy
ENG
19:01


aws
Services
Search
[Alt+S]
Ohio
Sumit_aws_11
EC2 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
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
nginxserver-us...
i-036fdaaa36e896d83
Running
t2.micro
2/2 checks passed
View alarms
us-east-2b
i-036fdaaa36e896d83 (nginxserver-us-east-2)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-036fdaaa36e896d83 (nginxserver-us-east-2)
IPv6 address
-
Public IPv4 address
18.118.6.86 | open address
Instance state
Running
Private IPv4 addresses
172.31.17.178
Public IPv4 DNS
ec2-18-118-6-86.us-east-2.compute.amazonaws.com | open address
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
ubuntu@ip-172-31-5-25:~/Terraform_day2task$ terraform destroy
aws_instance.demo1: Refreshing state... [id=i-091f85bf978c3d7c7]
aws_instance.demo2: Refreshing state... [id=i-0bfebcb846b2df91f]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy
Terraform will perform the following actions:
# aws_instance.demo1 will be destroyed
- resource "aws_instance" "demo1" {
- ami = "ami-0ec0e125bb6c6e8ec" -> null
- arn = "arn:aws:ec2:ap-south-1:730335346690:instance/i-091f85bf978c3d7c7" -> null
- associate_public_ip_address = true -> null
- availability_zone = "ap-south-1b" -> null
- cpu_core_count = 1 -> null
- cpu_threads_per_core = 1 -> null
- disable_api_stop = false -> null
- disable_api_termination = false -> null
- ebs_optimized = false -> null
- get_password_data = false -> null
- hibernation = false -> null
- id = "i-091f85bf978c3d7c7" -> null
- instance_initiated_shutdown_behavior = "stop" -> null
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 13.233.145.221 PrivateIPs: 172.31.5.25


aws
Services
Search
[Alt+S]
Mumbai
Sumit_aws_11
- ap_south_1_instance_id = "i-091f85bf978c3d7c7" -> null
- ap_south_1_public_ip = "13.127.118.3" -> null
- us_east_2_instance_id = "i-0bfebcb846b2df91f" -> null
- us_east_2_public_ip = "18.118.151.32" -> 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.demo1: Destroying... [id=i-091f85bf978c3d7c7]
aws_instance.demo2: Destroying... [id=i-0bfebcb846b2df91f]
aws_instance.demo1: Still destroying... [id=i-091f85bf978c3d7c7, 10s elapsed]
aws_instance.demo2: Still destroying... [id=i-0bfebcb846b2df91f, 10s elapsed]
aws_instance.demo1: Still destroying... [id=i-091f85bf978c3d7c7, 20s elapsed]
aws_instance.demo2: Still destroying... [id=i-0bfebcb846b2df91f, 20s elapsed]
aws_instance.demo1: Still destroying... [id=i-091f85bf978c3d7c7, 30s elapsed]
aws_instance.demo2: Still destroying... [id=i-0bfebcb846b2df91f, 30s elapsed]
aws_instance.demo1: Destruction complete after 40s
aws_instance.demo2: Still destroying... [id=i-0bfebcb846b2df91f, 40s elapsed]
aws_instance.demo2: Destruction complete after 42s
Destroy complete! Resources: 2 destroyed.
ubuntu@ip-172-31-5-25:~/Terraform_day2task$
i-0e764b4241da2399e (demo-terraform)
PublicIPs: 13.233.145.221 PrivateIPs: 172.31.5.25