Terraform Task

Task Description:

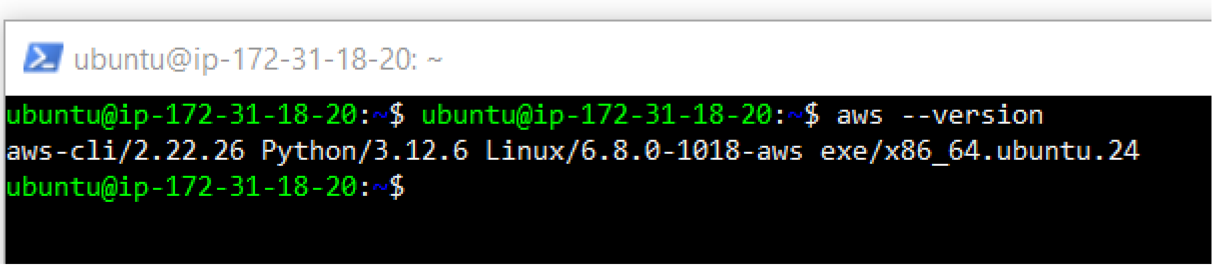
Launch Linux EC2 instances in two regions using a single Terraform file.

Techstacks needs to be used :

- AWS EC2
- Terraform
- AWS CLI

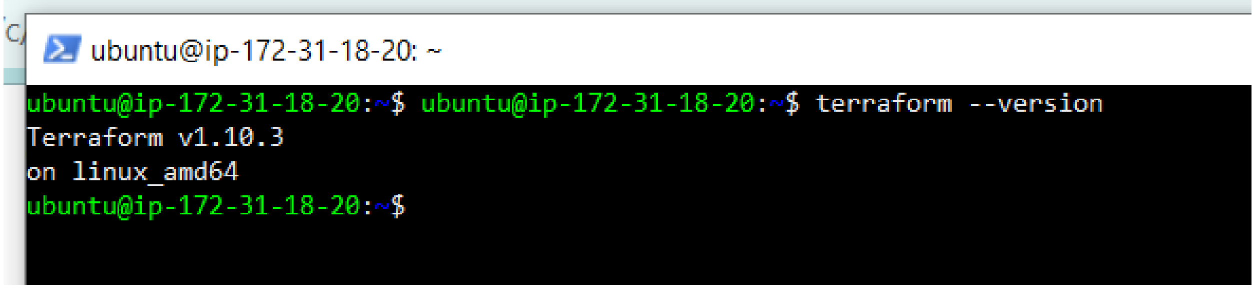
Step1

- Install AWS CLI



```
ubuntu@ip-172-31-18-20:~$ ubuntu@ip-172-31-18-20:~$ aws --version
aws-cli/2.22.26 Python/3.12.6 Linux/6.8.0-1018-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-18-20:~$
```

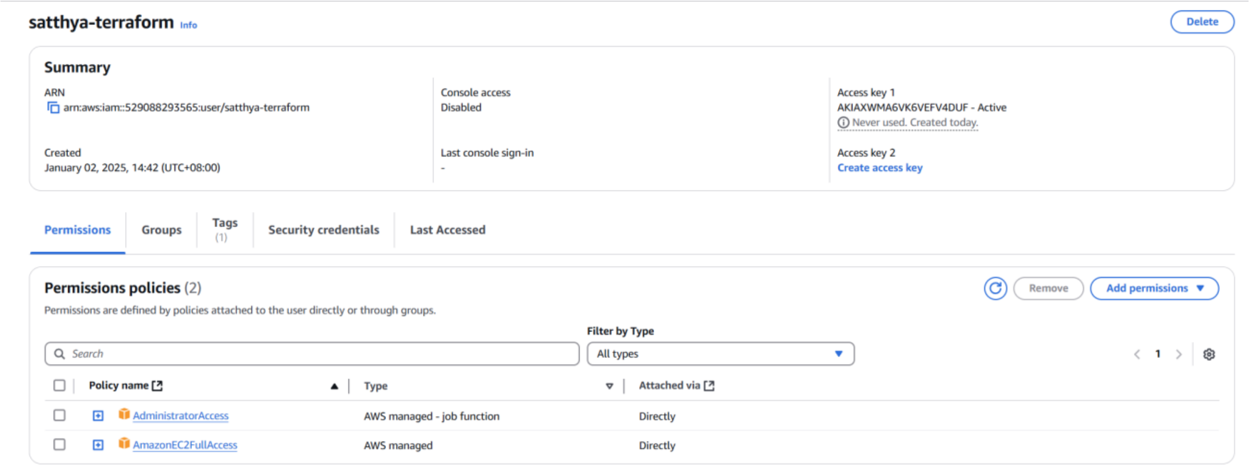
- Install Terraform



```
ubuntu@ip-172-31-18-20:~$ ubuntu@ip-172-31-18-20:~$ terraform --version
Terraform v1.10.3
on linux_amd64
ubuntu@ip-172-31-18-20:~$
```

Step2

- Set up IAM user and credential.



satthya-terraform Info

Delete

Summary

ARN
arn:aws:iam::529088293565:user/satthya-terraform

Created
January 02, 2025, 14:42 (UTC+08:00)

Console access
Disabled

Last console sign-in
-

Access key 1
AKIAXWMA6VK6VEFV4DUF - Active
Never used. Created today.

Access key 2
Create access key

Permissions | Groups | Tags (1) | Security credentials | Last Accessed

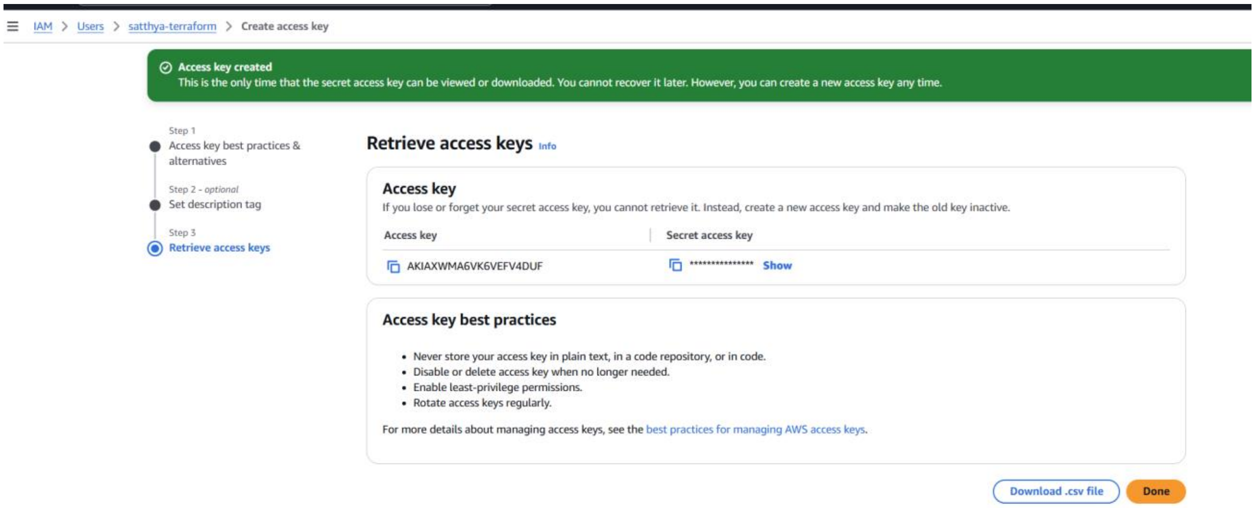
Permissions policies (2)

Permissions are defined by policies attached to the user directly or through groups.

Remove | Add permissions ▼

Filter by Type: All types

□	Policy name	Type	Attached via
□	AdministratorAccess	AWS managed - job function	Directly
□	AmazonEC2FullAccess	AWS managed	Directly



IAM > Users > satthya-terraform > Create access key

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAXWMA6VK6VEFV4DUF	*************** Show

Access key best practices

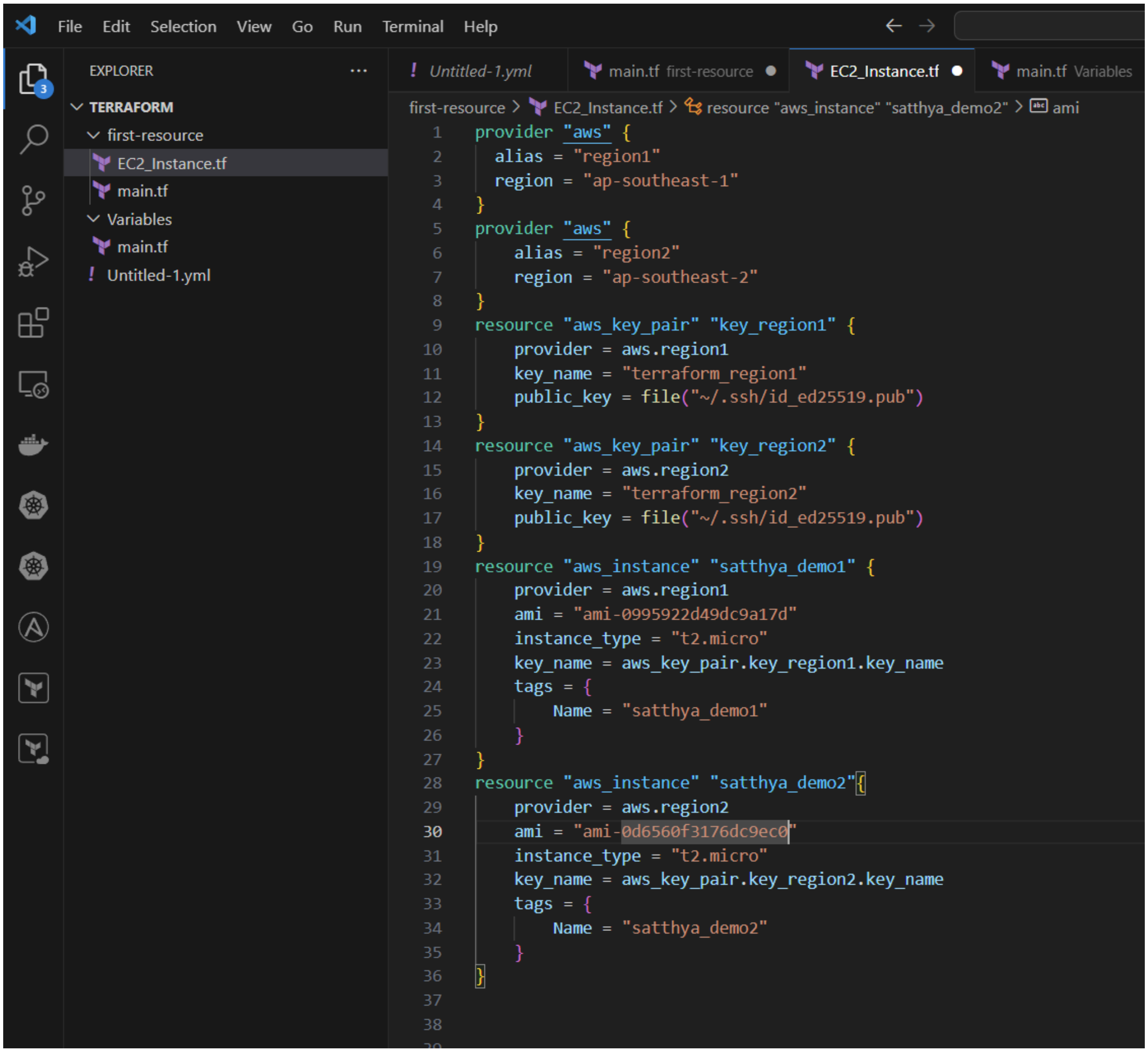
- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

Download .csv file | Done

Step3

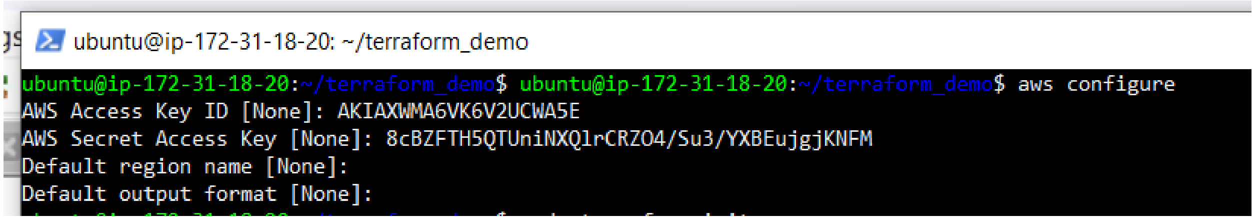
- Create Terraform Configuration file



```
provider "aws" {
  alias = "region1"
  region = "ap-southeast-1"
}
provider "aws" {
    alias = "region2"
    region = "ap-southeast-2"
}
resource "aws_key_pair" "key_region1" {
    provider = aws.region1
    key_name = "terraform_region1"
    public_key = file("~/.ssh/id_ed25519.pub")
}
resource "aws_key_pair" "key_region2" {
    provider = aws.region2
    key_name = "terraform_region2"
    public_key = file("~/.ssh/id_ed25519.pub")
}
resource "aws_instance" "satthya_demo1" {
    provider = aws.region1
    ami = "ami-0995922d49dc9a17d"
    instance_type = "t2.micro"
    key_name = aws_key_pair.key_region1.key_name
    tags = {
        Name = "satthya_demo1"
    }
}
resource "aws_instance" "satthya_demo2"{
    provider = aws.region2
    ami = "ami-0d6560f3176dc9ec0"
    instance_type = "t2.micro"
    key_name = aws_key_pair.key_region2.key_name
    tags = {
        Name = "satthya_demo2"
    }
}
```

Step4

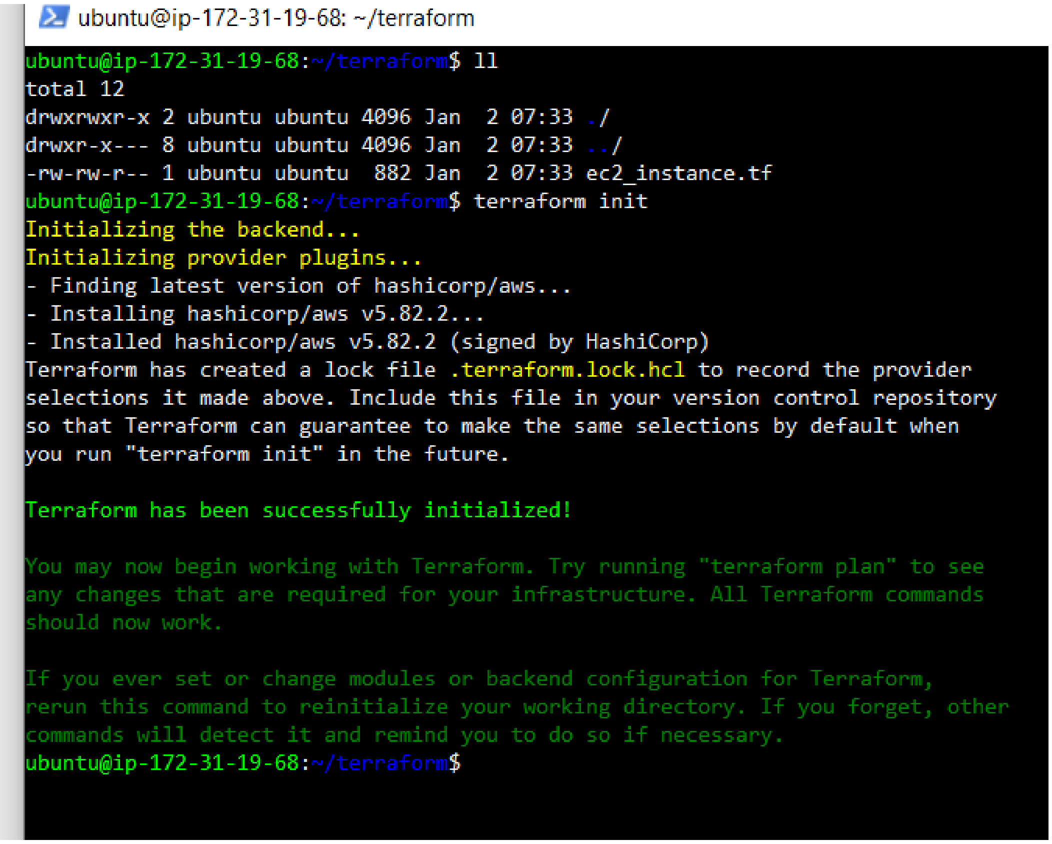
- AWS Configure



ubuntu@ip-172-31-18-20: ~/terraform_demo

```
ubuntu@ip-172-31-18-20:~/terraform_demo$ ubuntu@ip-172-31-18-20:~/terraform_demo$ aws configure
AWS Access Key ID [None]: AKIAXWMA6VK6V2UCWA5E
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

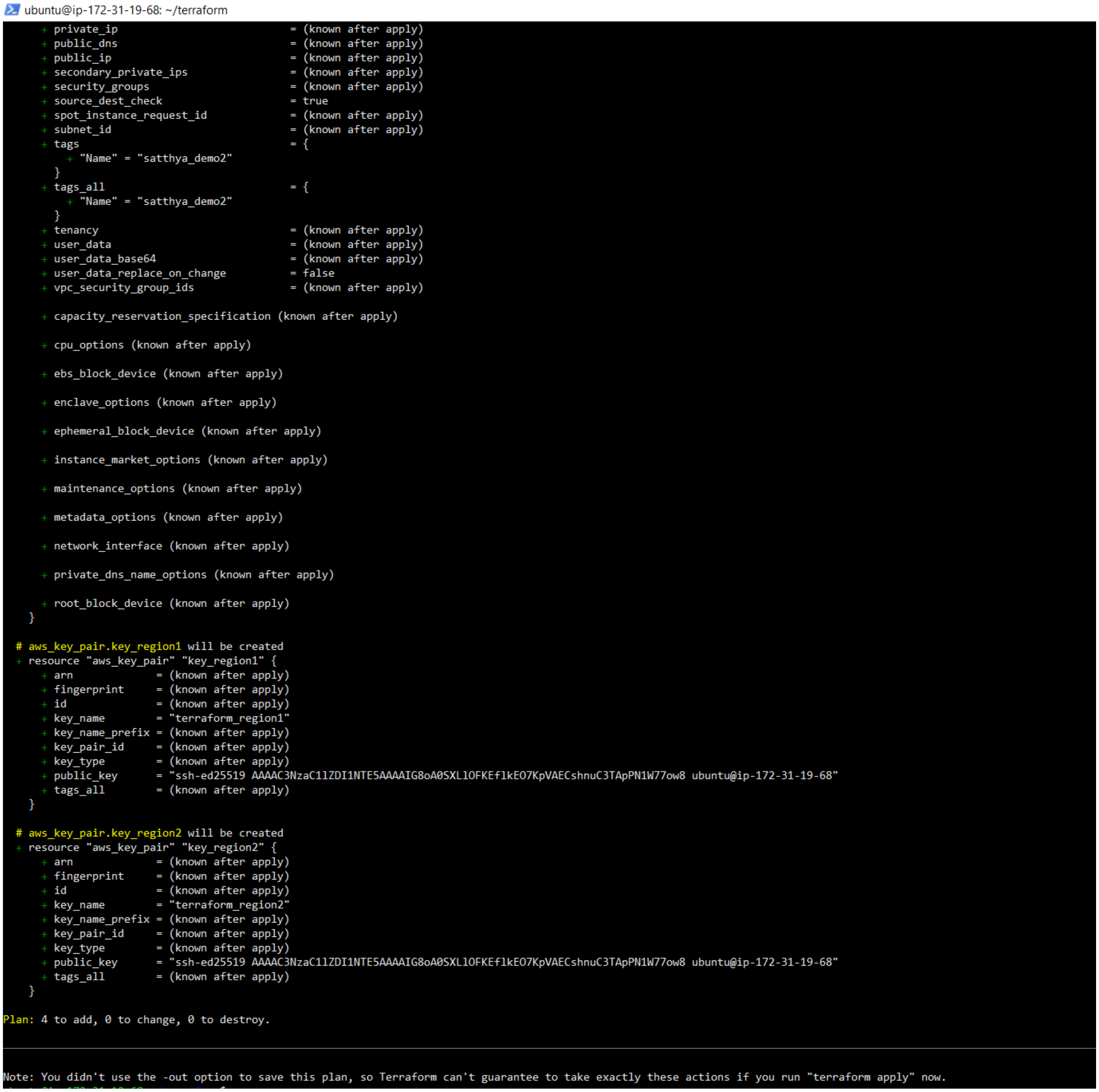
- terraform init



ubuntu@ip-172-31-19-68: ~/terraform

```
ubuntu@ip-172-31-19-68:~/terraform$ ll
total 12
drwxrwxr-x 2 ubuntu ubuntu 4096 Jan  2 07:33 ./
drwxr-x--- 8 ubuntu ubuntu 4096 Jan  2 07:33 ../
-rw-rw-r-- 1 ubuntu ubuntu  882 Jan  2 07:33 ec2_instance.tf
ubuntu@ip-172-31-19-68:~/terraform$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.82.2...
- Installed hashicorp/aws v5.82.2 (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-172-31-19-68:~/terraform$
```

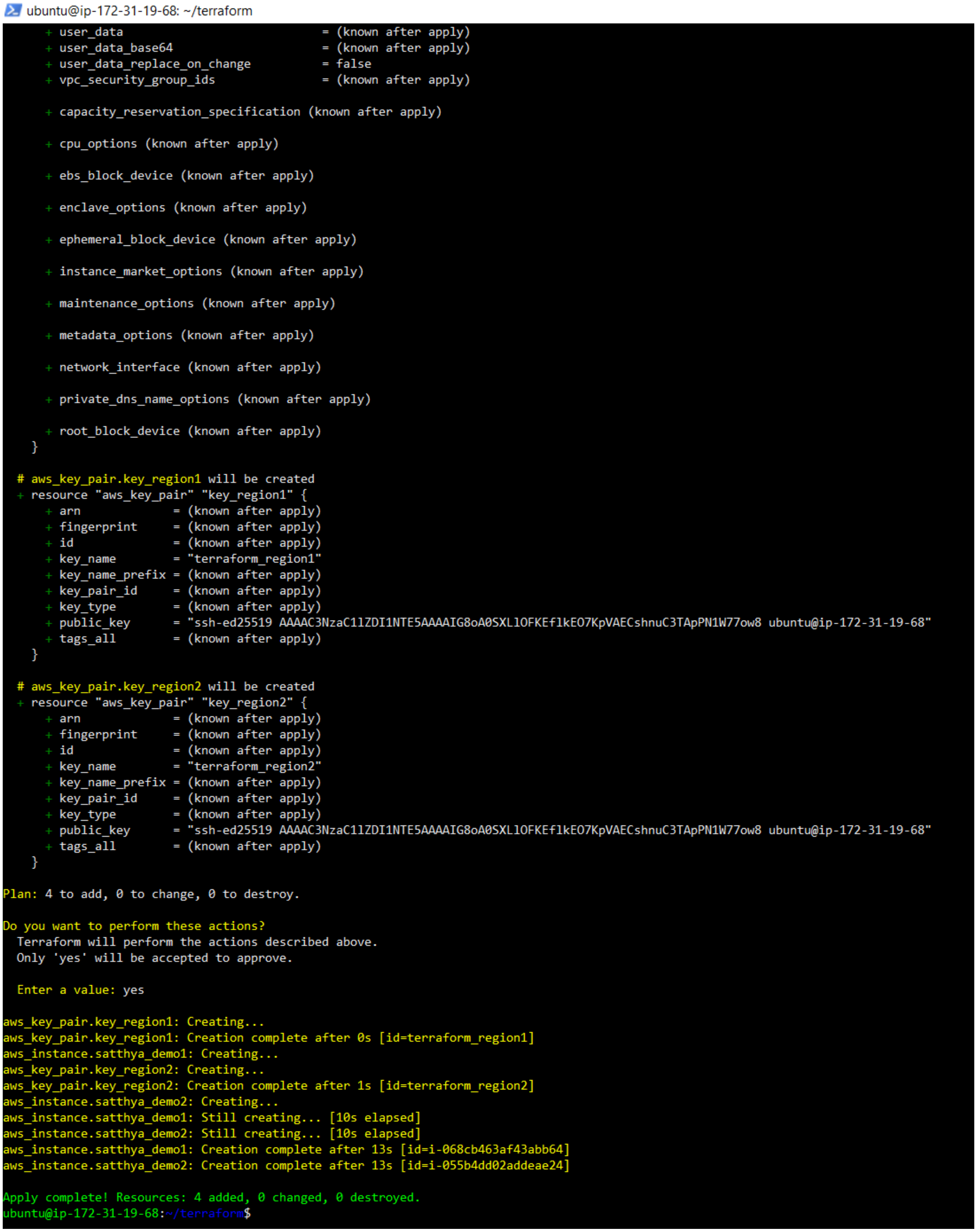
- terraform plan



ubuntu@ip-172-31-19-68: ~/terraform

```
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "satthya_demo2"
        }
      + tags_all                             = {
          + "Name" = "satthya_demo2"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

  # aws_key_pair.key_region1 will be created
  + resource "aws_key_pair" "key_region1" {
      + arn             = (known after apply)
      + fingerprint     = (known after apply)
      + id              = (known after apply)
      + key_name        = "terraform_region1"
      + key_name_prefix = (known after apply)
      + key_pair_id     = (known after apply)
      + key_type        = (known after apply)
      + public_key      = "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIG8oA0SXLlOFKEflkEO7KpVAECshnuC3TApPN1W77ow8 ubuntu@ip-172-31-19-68"
      + tags_all        = (known after apply)
    }

  # aws_key_pair.key_region2 will be created
  + resource "aws_key_pair" "key_region2" {
      + arn             = (known after apply)
      + fingerprint     = (known after apply)
      + id              = (known after apply)
      + key_name        = "terraform_region2"
      + key_name_prefix = (known after apply)
      + key_pair_id     = (known after apply)
      + key_type        = (known after apply)
      + public_key      = "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIG8oA0SXLlOFKEflkEO7KpVAECshnuC3TApPN1W77ow8 ubuntu@ip-172-31-19-68"
      + tags_all        = (known after apply)
    }

Plan: 4 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.
```

- terraform apply

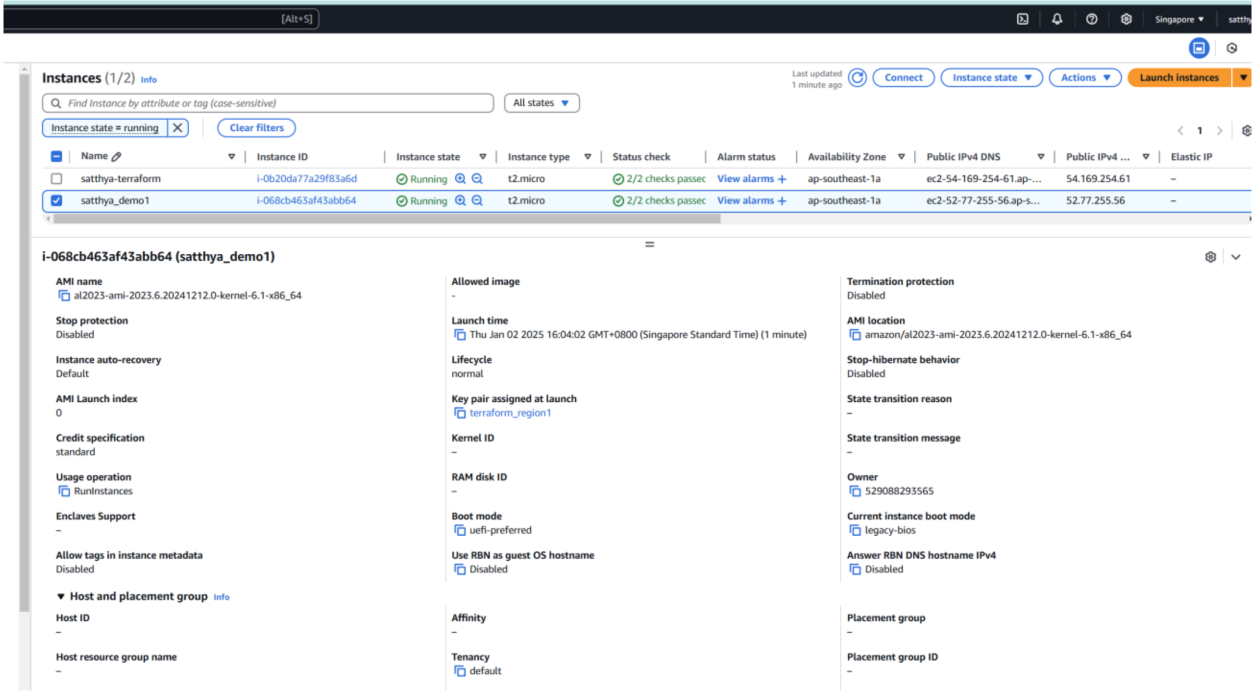


ubuntu@ip-172-31-19-68: ~/terraform

```
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

  # aws_key_pair.key_region1 will be created
  + resource "aws_key_pair" "key_region1" {
      + arn             = (known after apply)
      + fingerprint     = (known after apply)
      + id              = (known after apply)
      + key_name        = "terraform_region1"
      + key_name_prefix = (known after apply)
      + key_pair_id     = (known after apply)
      + key_type        = (known after apply)
      + public_key      = "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIG8oA0SXLlOFKEflkEO7KpVAECshnuC3TApPN1W77ow8 ubuntu@ip-172-31-19-68"
      + tags_all        = (known after apply)
    }

  # aws_key_pair.key_region2 will be created
  + resource "aws_key_pair" "key_region2" {
      + arn             = (known after apply)
      + fingerprint     = (known after apply)
      + id              = (known after apply)
      + key_name        = "terraform_region2"
      + key_name_prefix = (known after apply)
      + key_pair_id     = (known after apply)
      + key_type        = (known after apply)
      + public_key      = "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIG8oA0SXLlOFKEflkEO7KpVAECshnuC3TApPN1W77ow8 ubuntu@ip-172-31-19-68"
      + tags_all        = (known after apply)
    }

Plan: 4 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_key_pair.key_region1: Creating...
aws_key_pair.key_region1: Creation complete after 0s [id=terraform_region1]
aws_instance.satthya_demo1: Creating...
aws_key_pair.key_region2: Creating...
aws_key_pair.key_region2: Creation complete after 1s [id=terraform_region2]
aws_instance.satthya_demo2: Creating...
aws_instance.satthya_demo1: Still creating... [10s elapsed]
aws_instance.satthya_demo2: Still creating... [10s elapsed]
aws_instance.satthya_demo1: Creation complete after 13s [id=i-068cb463af43abb64]
aws_instance.satthya_demo2: Creation complete after 13s [id=i-055b4dd02addeae24]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-19-68:~/terraform$
```

Result

Region1



Instances (1/2) Info

Last updated 1 minute ago

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☐	satthya-terraform	i-0b20da77a29f83a6d	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-1a	ec2-54-169-254-61.ap-...	54.169.254.61	–
☑	satthya_demo1	i-068cb463af43abb64	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-1a	ec2-52-77-255-56.ap-s...	52.77.255.56	–

i-068cb463af43abb64 (satthya_demo1)

AMI name
al2023-ami-2023.6.20241212.0-kernel-6.1-x86_64

Stop protection
Disabled

Instance auto-recovery
Default

AMI Launch index
0

Credit specification
standard

Usage operation
RunInstances

Enclaves Support
–

Allow tags in instance metadata
Disabled

Allowed image
-

Launch time
Thu Jan 02 2025 16:04:02 GMT+0800 (Singapore Standard Time) (1 minute)

Lifecycle
normal

Key pair assigned at launch
terraform_region1

Kernel ID
–

RAM disk ID
–

Boot mode
uefi-preferred

Use RBN as guest OS hostname
Disabled

Termination protection
Disabled

AMI location
amazon/al2023-ami-2023.6.20241212.0-kernel-6.1-x86_64

Stop-hibernate behavior
Disabled

State transition reason
–

State transition message
–

Owner
529088293565

Current instance boot mode
legacy-bios

Answer RBN DNS hostname IPv4
Disabled

▼ Host and placement group Info

Host ID
–

Host resource group name
–

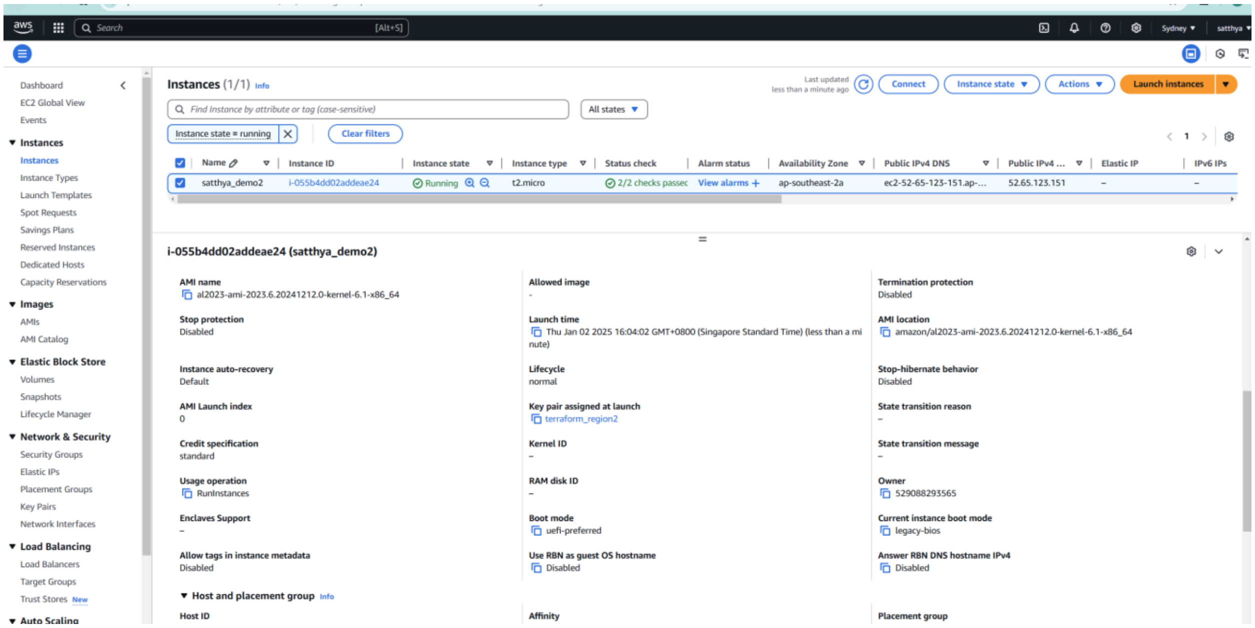
Affinity
–

Tenancy
default

Placement group
–

Placement group ID
–

Region2



Instances (1/1) Info

Last updated less than a minute ago

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
☑	satthya_demo2	i-055b4dd02addeae24	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-2a	ec2-52-65-123-151.ap-...	52.65.123.151	–	–

i-055b4dd02addeae24 (satthya_demo2)

AMI name
al2023-ami-2023.6.20241212.0-kernel-6.1-x86_64

Stop protection
Disabled

Instance auto-recovery
Default

AMI Launch index
0

Credit specification
standard

Usage operation
RunInstances

Enclaves Support
–

Allow tags in instance metadata
Disabled

Allowed image
-

Launch time
Thu Jan 02 2025 16:04:02 GMT+0800 (Singapore Standard Time) (less than a minute)

Lifecycle
normal

Key pair assigned at launch
terraform_region2

Kernel ID
–

RAM disk ID
–

Boot mode
uefi-preferred

Use RBN as guest OS hostname
Disabled

Termination protection
Disabled

AMI location
amazon/al2023-ami-2023.6.20241212.0-kernel-6.1-x86_64

Stop-hibernate behavior
Disabled

State transition reason
–

State transition message
–

Owner
529088293565

Current instance boot mode
legacy-bios

Answer RBN DNS hostname IPv4
Disabled

▼ Host and placement group Info

Host ID

Affinity

Placement group