

```
ubuntu@ubuntu: ~/Desktop/ec2
ubuntu@ubuntu:~/Desktop/ec2$ nano ec2.tf
ubuntu@ubuntu:~/Desktop/ec2$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.48.0

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@ubuntu:~/Desktop/ec2$ 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.instance_ap_southeast_2 will be created
+ resource "aws_instance" "instance_ap_southeast_2" {
+   ami               = "ami-08060dc9757080771"
+   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)
```

1(creating tf file)

```
GNU nano 6.2 ec2.tf
# Specify providers for each region
provider "aws" {
  alias = "ap-southeast-2"
  region = "ap-southeast-2"
}

provider "aws" {
  alias = "us-east-1"
  region = "us-east-1"
}

# Create EC2 instances in each region
resource "aws_instance" "instance_ap_southeast_2" {
  provider = aws.ap_southeast_2
  ami      = "ami-08060dc9757080771" # Linux AMI ID for ap-southeast-2
  instance_type = "t2.micro"

  tags = {
    Name = "instance-ap-southeast-2"
  }
}

resource "aws_instance" "instance_us-east-1" {
  provider = aws.us-east-1
  ami      = "ami-04b78fa74e45c3917" # Linux AMI ID for us-east-1
  instance_type = "t2.micro"

  tags = {
    Name = "instance-us-east-1"
  }
}
```

2(terraform file)

```
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@ubuntu:~/Desktop/ec2$ 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:
```

3(terraform apply)

```

Enter a value: yes
aws_instance.instance_ap_southeast_2: Creating...
aws_instance.instance_us_east_1: Creating...
aws_instance.instance_ap_southeast_2: Still creating... [10s elapsed]
aws_instance.instance_us_east_1: Still creating... [10s elapsed]
aws_instance.instance_ap_southeast_2: Still creating... [20s elapsed]
aws_instance.instance_us_east_1: Still creating... [20s elapsed]
aws_instance.instance_ap_southeast_2: Still creating... [30s elapsed]
aws_instance.instance_us_east_1: Still creating... [30s elapsed]
aws_instance.instance_ap_southeast_2: Creation complete after 35s [id=i-0f52d1eb9e0263746]
aws_instance.instance_us_east_1: Creation complete after 37s [id=i-0e44a3a708e74a7fe]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

```

4(output)

The screenshot shows the AWS Management Console for the us-east-1 region. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, and various instance-related services. The main content area shows the 'Instances (2)' page. A table lists the instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
	i-0e8a4c46c9e5cd541	Terminated	t2.micro	-	View alarms +	us-east-1c	-
instance-us-ea...	i-0e44a3a708e74a7fe	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-18-20...

Below the table, a 'Select an instance' dialog box is open, showing the details of the selected instance.

5(created ec2 instance in n.virginia reg)

The screenshot shows the AWS Management Console for the ap-southeast-2 region. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, and various instance-related services. The main content area shows the 'Instances (1/4)' page. A table lists the instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
terraform	i-0289e31c797c6682e	Stopped	t2.micro	-	View alarms +	ap-southeast-2c	-
instance-ap-so...	i-0bc2295cb22f5d4e8	Terminated	t2.micro	-	View alarms +	ap-southeast-2c	-
instance-ap-so...	i-07ca48c4c9526b970	Terminated	t2.micro	-	View alarms +	ap-southeast-2c	-
instance-ap-so...	i-0f52d1eb9e0263746	Running	t2.micro	Initializing	View alarms +	ap-southeast-2c	ec2-3-27-2...

Below the table, the details for the selected instance 'i-0f52d1eb9e0263746 (instance-ap-southeast-2)' are shown. The 'Instance summary' section includes the following details:

- Platform: Ubuntu (Inferred)
- Platform details: Linux/UNIX
- AMI ID: ami-080660c9757080771
- AMI name: ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64
- Monitoring: disabled
- Termination protection: Disabled

6(created ec2 instance in sydney reg)