Terraform with AWS

Provisioning on AWS is quite easy and straightforward with Terraform.

Prerequisites

AWS CLI installed

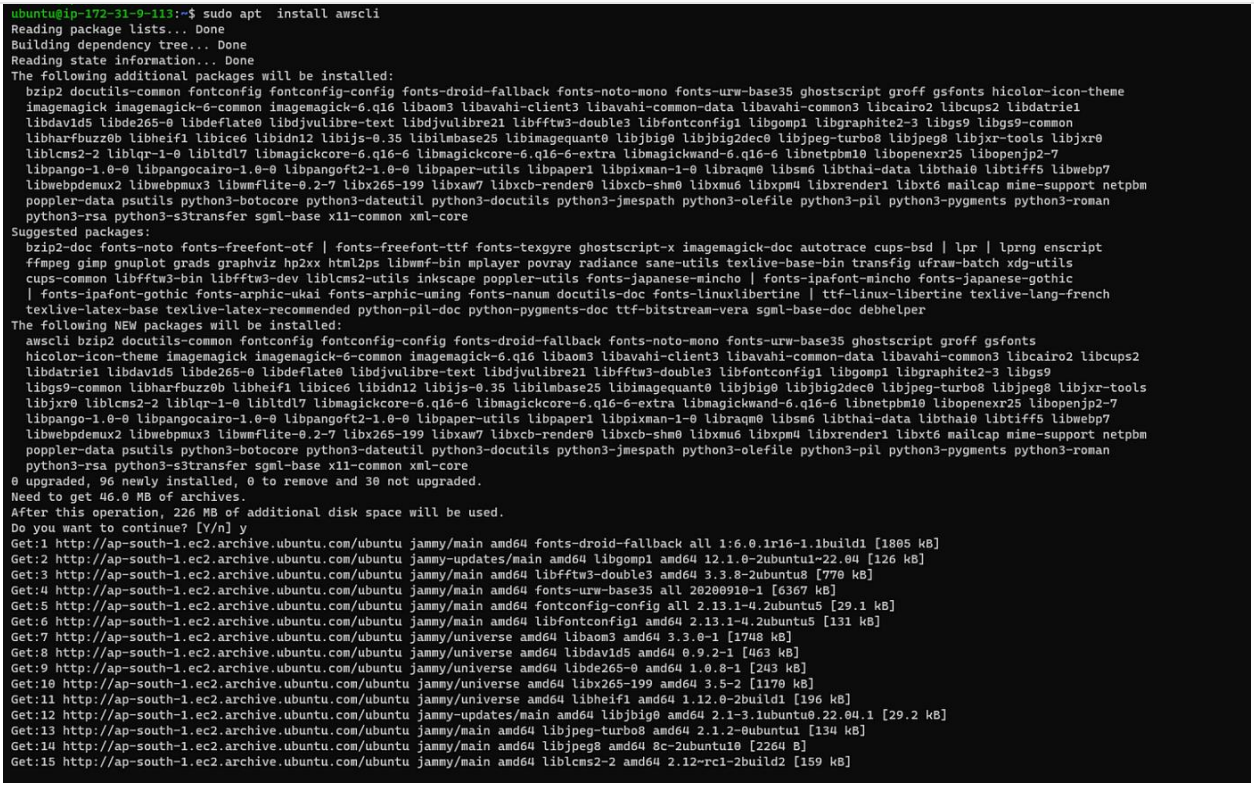
The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

1.Create an EC2 instance.

2.SSH into EC2 instance

3.Install AWS CLI

```
sudo apt install awscli
```



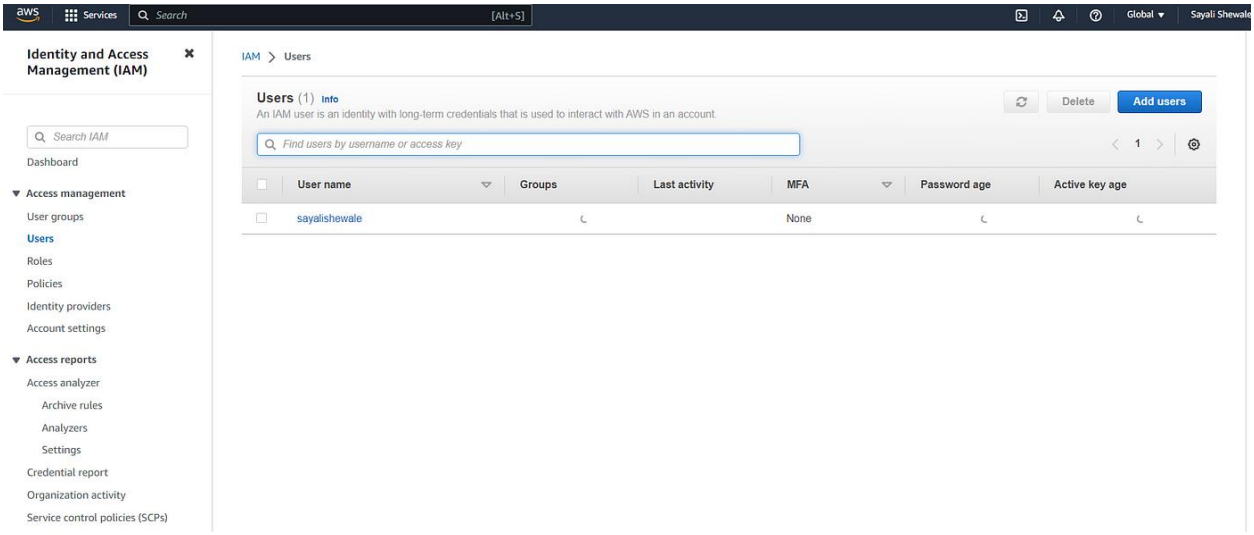
Check aws cli version

AWS IAM user

IAM (Identity Access Management) AWS Identity and Access Management (IAM) is a web service that helps you securely control

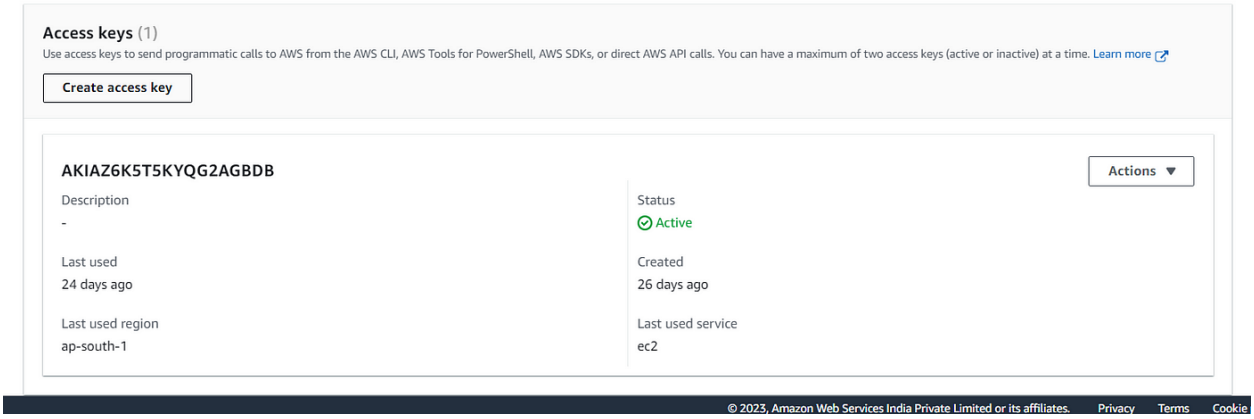
access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Create IAM user.

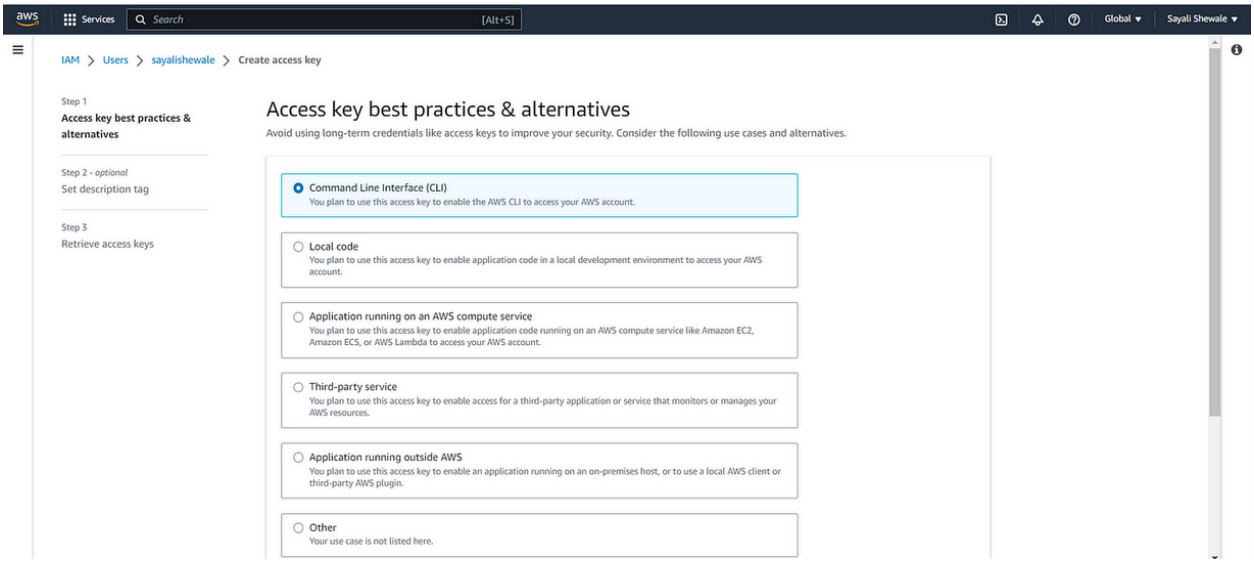


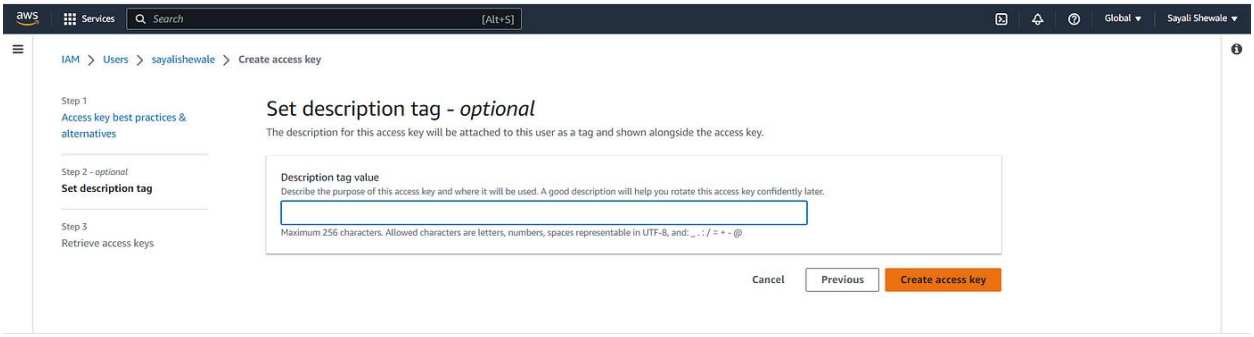
Create Access key for IAM user.

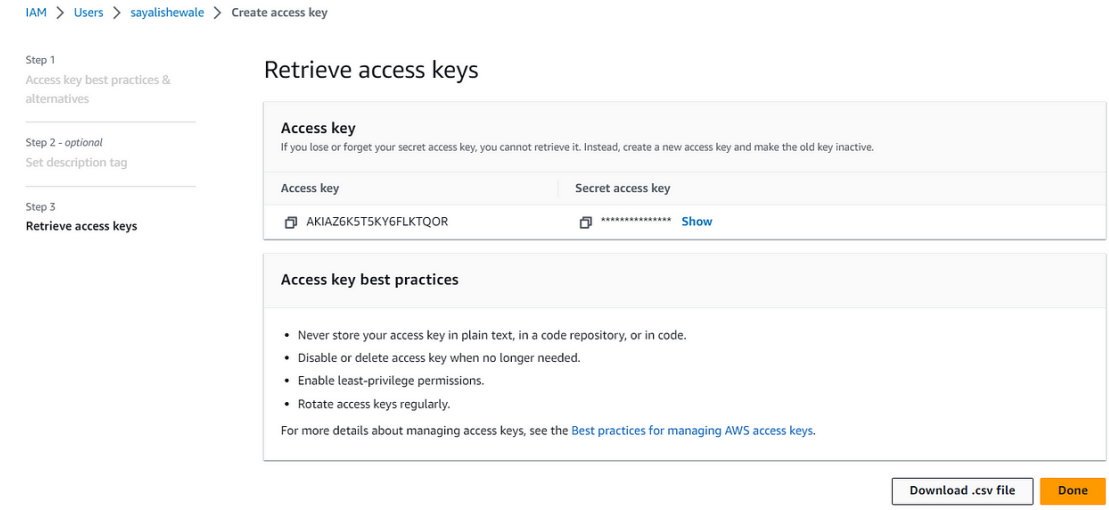
Click on ‘Create access key’



Select ‘command line interface’

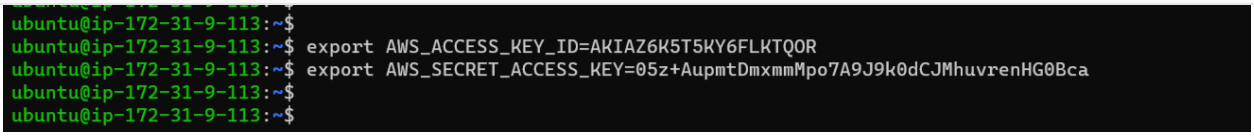






In order to connect your AWS account and Terraform, you need the access keys and secret access keys exported to your machine.

```
export AWS_ACCESS_KEY_ID=<access key>
export AWS_SECRET_ACCESS_KEY=<secret access key>
```



Install required providers

```
terraform {
 required_providers {
        aws = {
        source  = "hashicorp/aws"
        version = "~> 4.16"
}
}
        required_version = ">= 1.2.0"
}
```

The terraform block defines the version of Terraform that is required to execute this configuration. In this case, it specifies that the Terraform version must be >= 1.2.0.

The required_providers block declares the AWS provider and its version that Terraform will use for the resources defined in this configuration. In this case, it declares the AWS provider with the source hashicorp/aws and specifies that the version of the provider should be ~> 4.16, which means any version of the AWS provider greater than or equal to 4.16 and less than 5.0 will be acceptable.

Add the region where you want your instances to be

```
provider "aws" {
region = "ap-south-1"
}
```

```
ubuntu@ip-172-31-9-113:~$ cd terraform/
ubuntu@ip-172-31-9-113:~/terraform$ cd terraform-aws/
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$ sudo vi main.tf
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$ cat main.tf
terraform {
 required_providers {
        aws = {
        source  = "hashicorp/aws"
        version = "~> 4.16"
}
}
        required_version = ">= 1.2.0"
}

provider "aws" {
region = "ap-south-1"
}
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$
```

Task-01

Provision an AWS EC2 instance using Terraform

```
resource "aws_instance" "aws_ec2_demo" {
        count = 2
        ami = "ami-0f8ca728008ff5af4"
        instance_type = "t2.micro"
        tags = {
            Name = "TerraformTestInstance"
  }
}
```

The resource block has a resource type of “aws_instance” and a resource name of “aws_ec2_demo”. The count parameter is set to 2, which means that two instances will be created.

The ami parameter specifies the Amazon Machine Image (AMI) to use for the instances. In this case, the AMI ID is “ami-0f8ca728008ff5af4”.

The instance_type parameter specifies the type of instance to create. In this case, the instance type is “t2.micro”.

The tags parameter specifies metadata to attach to the instance, in this case, a tag named “Name” with the value “TerraformTestInstance”.

```
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$ sudo vi main.tf
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$ cat main.tf
terraform {
 required_providers {
        aws = {
        source  = "hashicorp/aws"
        version = "~> 4.16"
}
}
        required_version = ">= 1.2.0"
}

provider "aws" {
region = "ap-south-1"
}

resource "aws_instance" "aws_ec2_demo" {
        count = 2
        ami = "ami-0f8ca728008ff5af4"
        instance_type = "t2.micro"
        tags = {
            Name = "TerraformTestInstance"
  }
}
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$
```

first initialize the working directory with the necessary plugins and modules by executing terraform init

```
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 4.16"...
- Installing hashicorp/aws v4.59.0...
- Installed hashicorp/aws v4.59.0 (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-172-31-9-113:~/terraform/terraform-aws$
```

It will create an execution plan by analyzing the changes required to achieve the desired state of your infrastructure with terraform plan

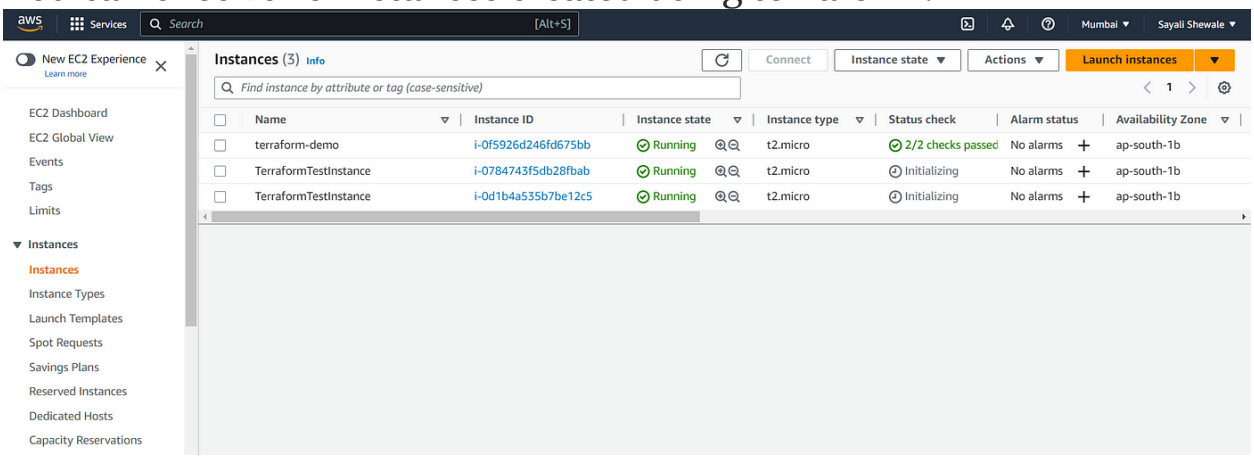
```
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$ terraform plan

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.aws_ec2_demo[0] will be created
  + resource "aws_instance" "aws_ec2_demo" {
      + ami                                  = "ami-0f8ca728008ff5af4"
      + 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_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)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + 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
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "TerraformTestInstance"
        }
      + tags_all                             = {
          + "Name" = "TerraformTestInstance"
        }
```

```
  # aws_instance.aws_ec2_demo[1] will be created
  + resource "aws_instance" "aws_ec2_demo" {
      + ami                                  = "ami-0f8ca728008ff5af4"
      + 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_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)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + 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
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "TerraformTestInstance"
        }
      + tags_all                             = {
          + "Name" = "TerraformTestInstance"
        }
      + 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)
    }

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

Finally, it will apply the changes to create or update resources as needed with terraform apply.

```
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$ 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.aws_ec2_demo[0] will be created
  + resource "aws_instance" "aws_ec2_demo" {
      + ami                                  = "ami-0f8ca728008ff5af4"
      + 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_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)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + 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
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "TerraformTestInstance"
        }
      + tags_all                             = {
          + "Name" = "TerraformTestInstance"
        }
      + tenancy                              = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + 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
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "TerraformTestInstance"
        }
      + tags_all                             = {
          + "Name" = "TerraformTestInstance"
        }
      + 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)
    }

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.aws_ec2_demo[0]: Creating...
aws_instance.aws_ec2_demo[1]: Creating...
aws_instance.aws_ec2_demo[0]: Still creating... [10s elapsed]
aws_instance.aws_ec2_demo[1]: Still creating... [10s elapsed]
aws_instance.aws_ec2_demo[0]: Still creating... [20s elapsed]
aws_instance.aws_ec2_demo[1]: Still creating... [20s elapsed]
aws_instance.aws_ec2_demo[0]: Still creating... [30s elapsed]
aws_instance.aws_ec2_demo[1]: Still creating... [30s elapsed]
aws_instance.aws_ec2_demo[0]: Creation complete after 31s [id=i-0784743f5db28fbab]
aws_instance.aws_ec2_demo[1]: Still creating... [40s elapsed]
aws_instance.aws_ec2_demo[1]: Creation complete after 41s [id=i-0d1b4a535b7be12c5]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-9-113:~/terraform/terraform-aws$
```

You can check two instances created using terraform.



Thank you for reading!