TERRAFORM

Infrastructure as a code

Presentation By: Madhusudhan

What is terraform?

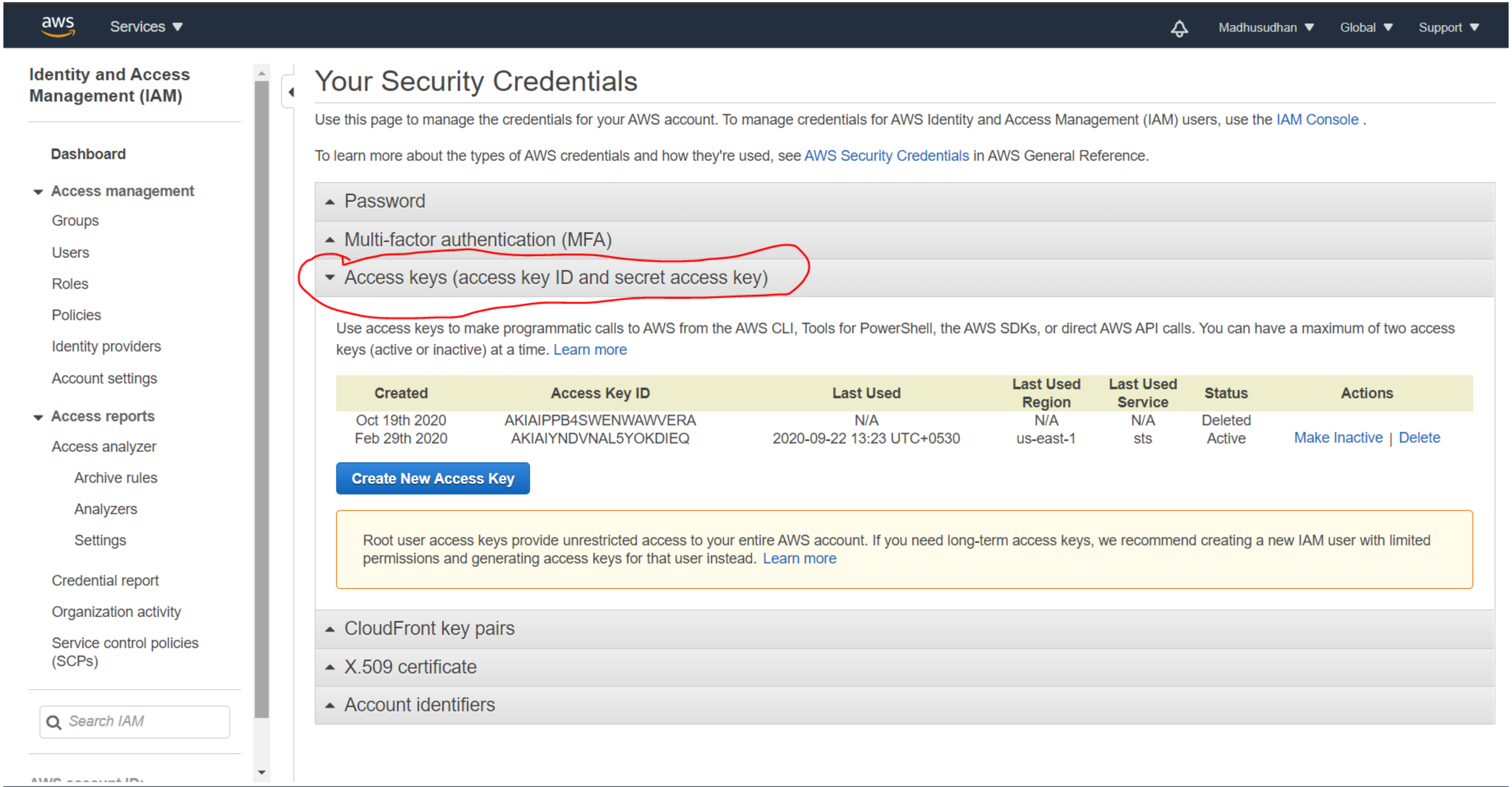
- **Terraform** is a tool for building, changing, and versioning infrastructure safely and efficiently. **Terraform** can manage existing and popular service providers as well as custom in-house solutions. Configuration files describe to **Terraform** the components needed to run a single **application** or your entire datacenter.

How to install Terraform?

- Please follow this video
- https://www.youtube.com/watch?v=R3fohgDHCYg&ab_channel=AutomationwithScripting

- Terraform AWS Scripts - https://registry.terraform.io/providers/hashicorp/aws/latest/docs
- https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/s3_bucket

Generate the Access key by going to IAM



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console .

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

Password

Multi-factor authentication (MFA)

Access keys (access key ID and secret access key)

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, the AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Oct 19th 2020	AKIAIPPB4SWENWAWVERA	N/A	N/A	N/A	Deleted	
Feb 29th 2020	AKIAIYNDVNAL5YOKDIEQ	2020-09-22 13:23 UTC+0530	us-east-1	sts	Active	Make Inactive | Delete

Create New Access Key

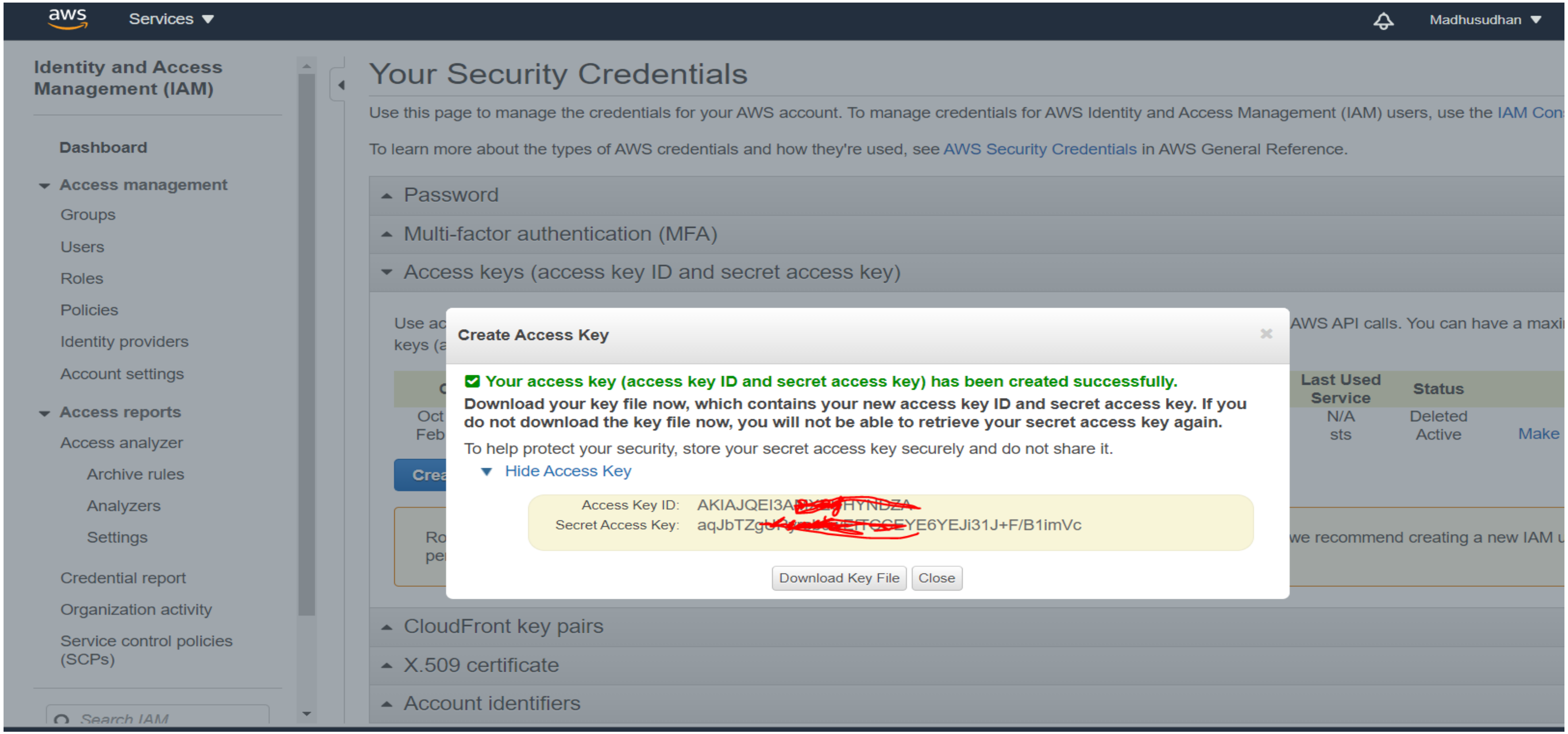
Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead. Learn more

CloudFront key pairs

X.509 certificate

Account identifiers

Access And Secret key – Copy both these keys



Identity and Access Management (IAM)

- Dashboard
- Access management
  - Groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
- Access reports
  - Access analyzer
    - Archive rules
    - Analyzers
    - Settings
  - Credential report
  - Organization activity
  - Service control policies (SCPs)

Search IAM

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Con

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

- Password
- Multi-factor authentication (MFA)
- Access keys (access key ID and secret access key)

Use ac ... AWS API calls. You can have a maxi
keys (a

C	Last Used Service	Status	
Oct	N/A	Deleted	
Feb	sts	Active	Make

Crea

Ro ... we recommend creating a new IAM u
pe

Create Access Key

✓ Your access key (access key ID and secret access key) has been created successfully.

Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.

To help protect your security, store your secret access key securely and do not share it.

▼ Hide Access Key

Access Key ID:	AKIAJQEI3A[redacted]
Secret Access Key:	aqJbTZg[redacted]YE6YEJi31J+F/B1imVc

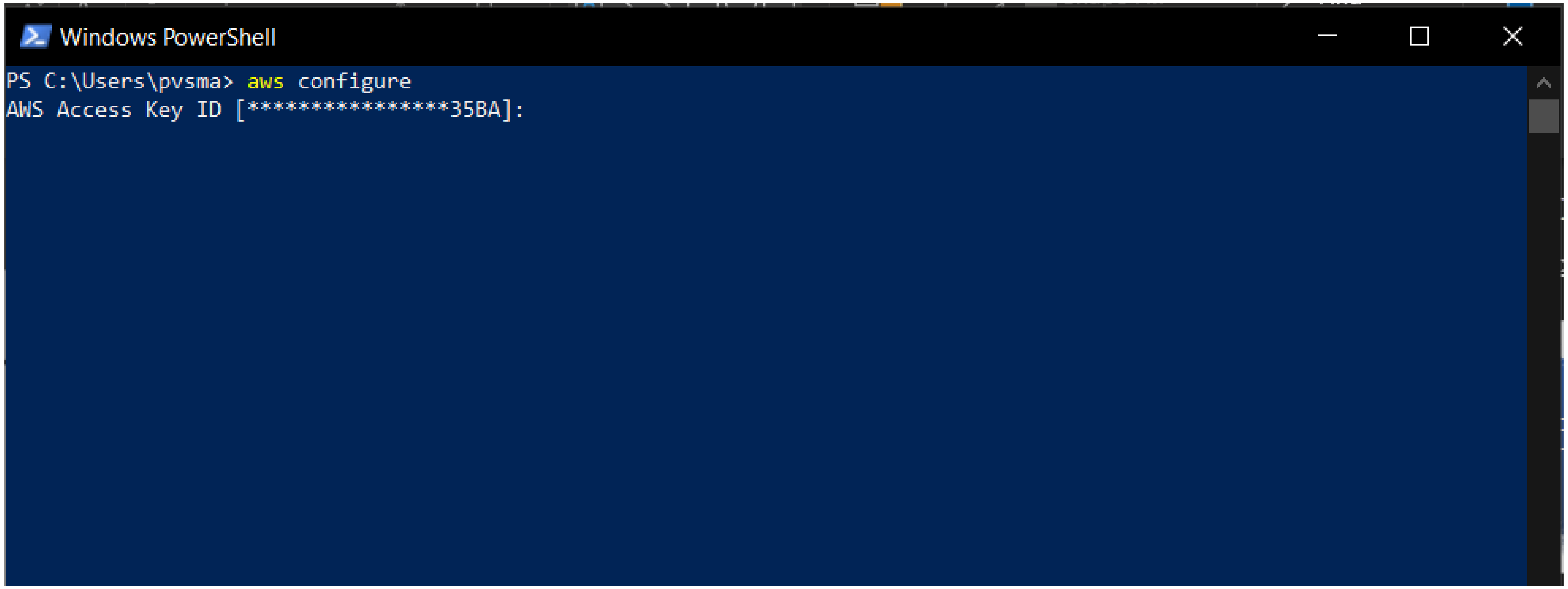
Download Key File | Close

- CloudFront key pairs
- X.509 certificate
- Account identifiers

Configure AWS in the PowerShell?

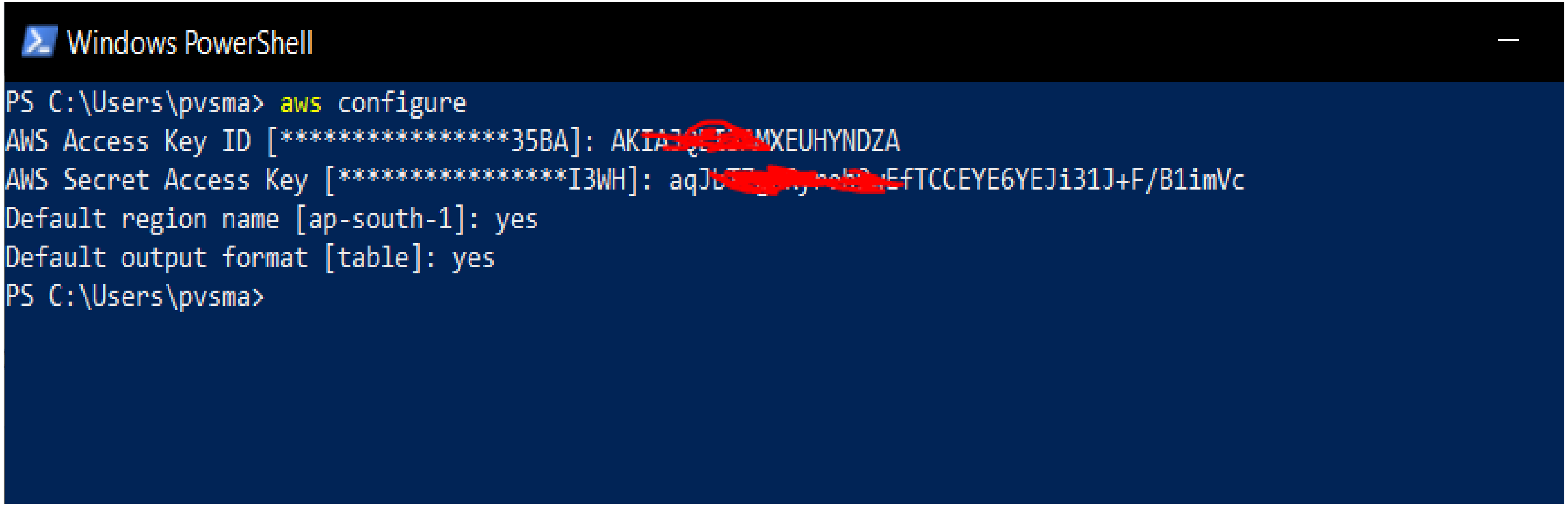
Cmd: aws configure

Enter the Access key and Secret key(copy & paste)



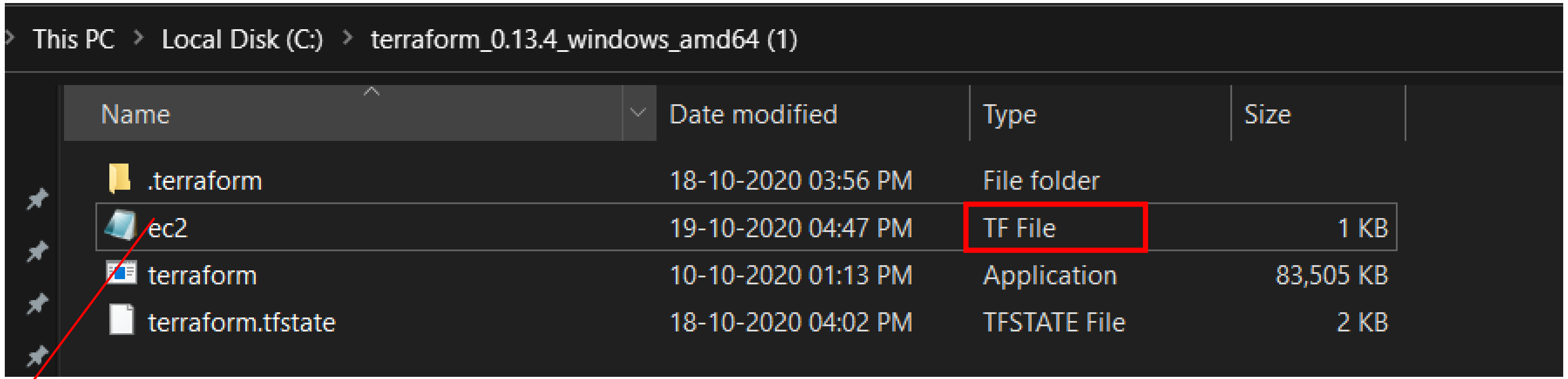
```
PS C:\Users\pvsma> aws configure
AWS Access Key ID [****************35BA]:
```

After configuring



```
PS C:\Users\pvsma> aws configure
AWS Access Key ID [****************35BA]: AKI[redacted]MXEUHYNDZA
AWS Secret Access Key [****************I3WH]: aqJ[redacted]EfTCCEYE6YEJi31J+F/B1imVc
Default region name [ap-south-1]: yes
Default output format [table]: yes
PS C:\Users\pvsma>
```

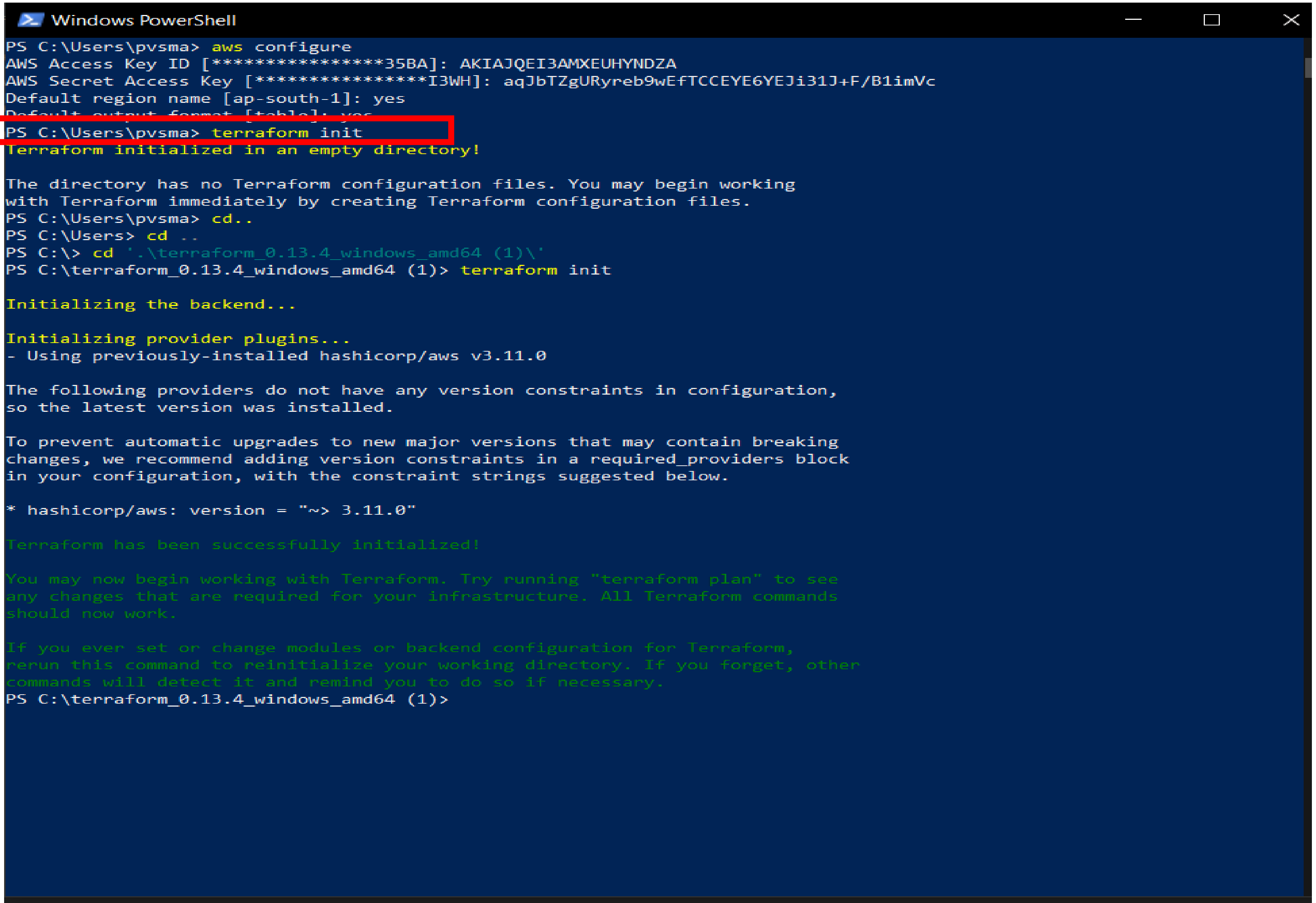
Go to the terraform folder and create a note pad file with extension .tf(tf)


This PC > Local Disk (C:) > terraform_0.13.4_windows_amd64 (1)

Name	Date modified	Type	Size
.terraform	18-10-2020 03:56 PM	File folder	
ec2	19-10-2020 04:47 PM	TF File	1 KB
terraform	10-10-2020 01:13 PM	Application	83,505 KB
terraform.tfstate	18-10-2020 04:02 PM	TFSTATE File	2 KB

```
# Configure the AWS Provider
provider "aws" {
  region = "ap-south-1"
}

# Create a VPC
resource "aws_vpc" "example" {
  cidr_block = "10.0.0.0/16"
}
```

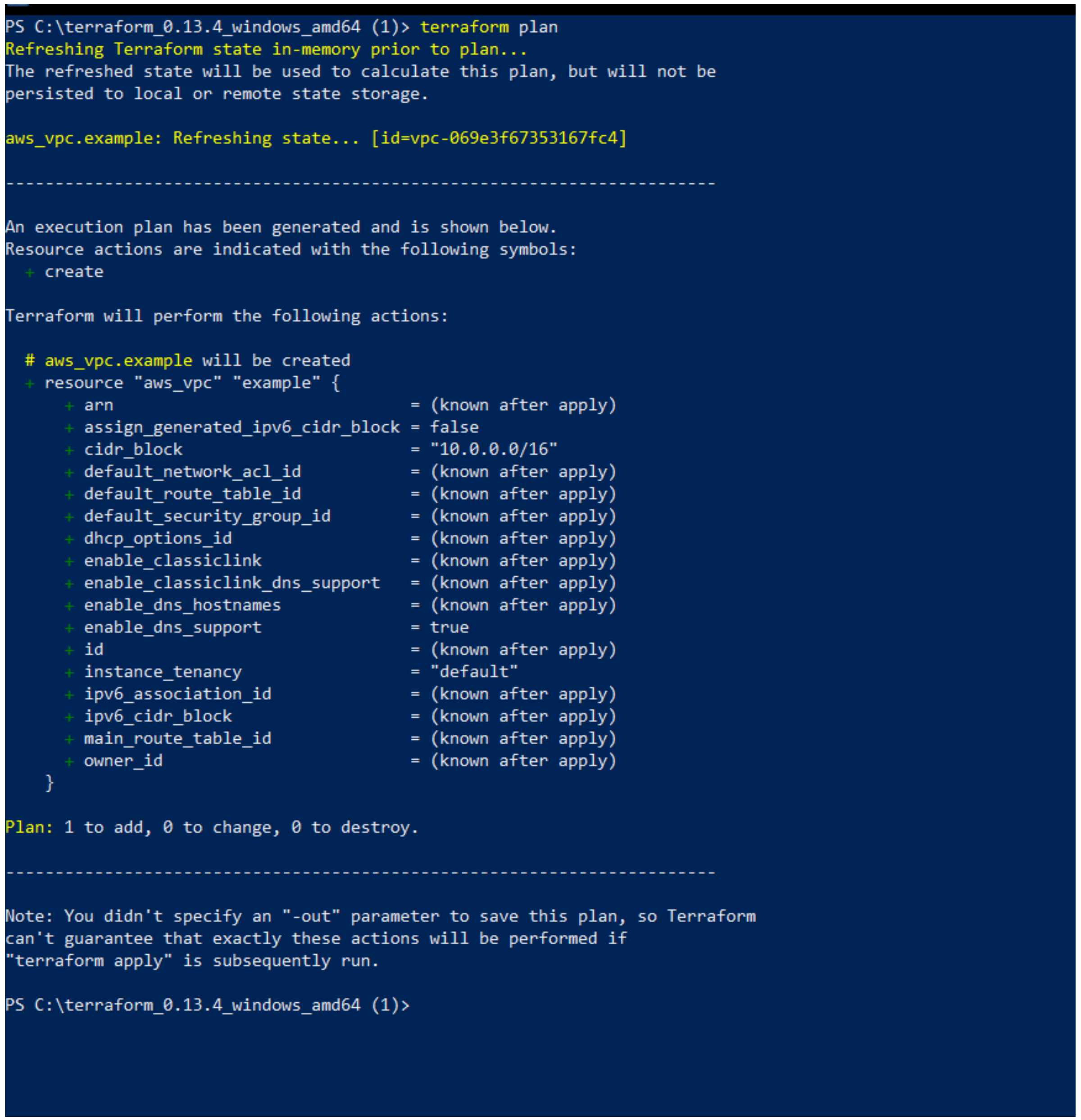
Step 1 – **Terraform init** (ensure before init you are in the right terraform directory)



Windows PowerShell

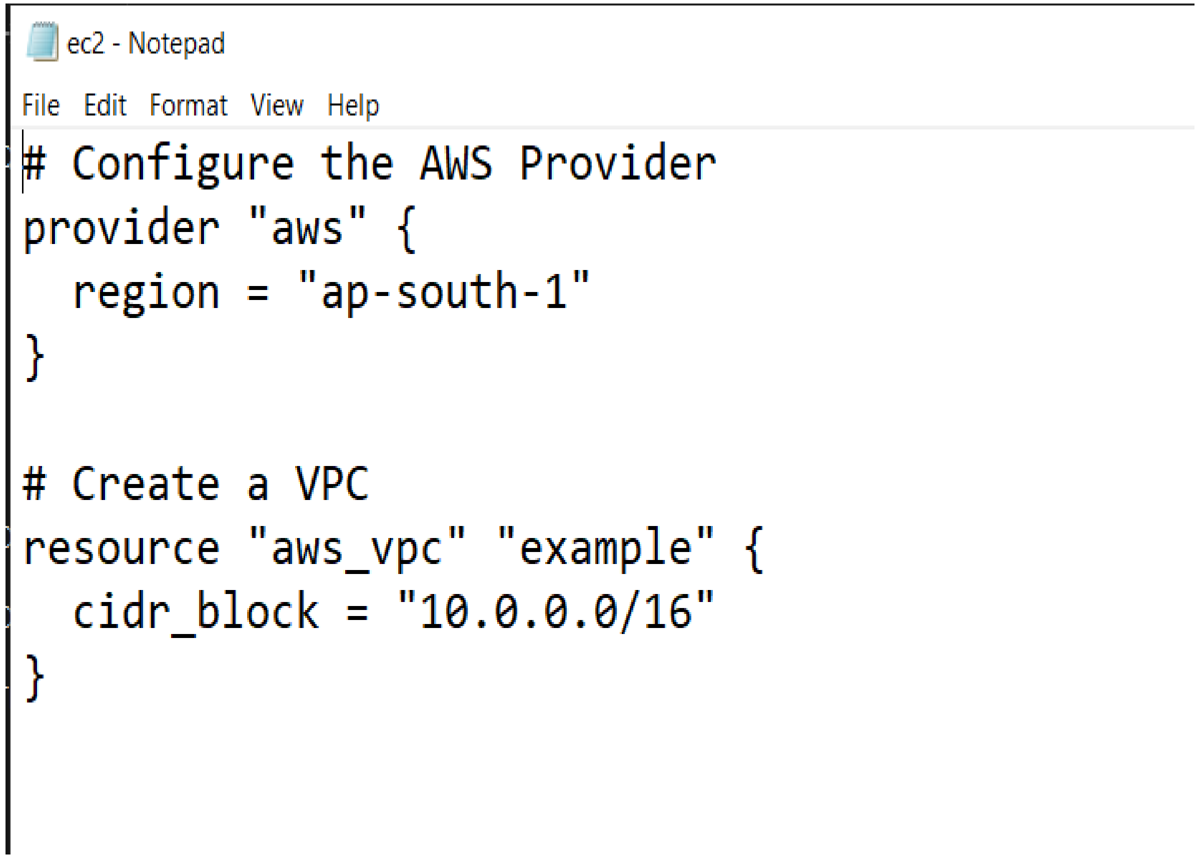
```
PS C:\Users\pvsma> aws configure
AWS Access Key ID [****************35BA]: AKIAJQEI3AMXEUHYNDZA
AWS Secret Access Key [****************I3WH]: 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
Default region name [ap-south-1]: yes
Default output format [table]: yes
PS C:\Users\pvsma> terraform init
Terraform initialized in an empty directory!

The directory has no Terraform configuration files. You may begin working
with Terraform immediately by creating Terraform configuration files.
PS C:\Users\pvsma> cd..
PS C:\Users> cd ..
PS C:\> cd '.\terraform_0.13.4_windows_amd64 (1)\'
PS C:\terraform_0.13.4_windows_amd64 (1)> terraform init

Initializing the backend...

Initializing provider plugins...
- Using previously-installed hashicorp/aws v3.11.0

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, we recommend adding version constraints in a required_providers block
in your configuration, with the constraint strings suggested below.

* hashicorp/aws: version = "~> 3.11.0"

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.
PS C:\terraform_0.13.4_windows_amd64 (1)>
```

Step 2 – **Terraform Plan**



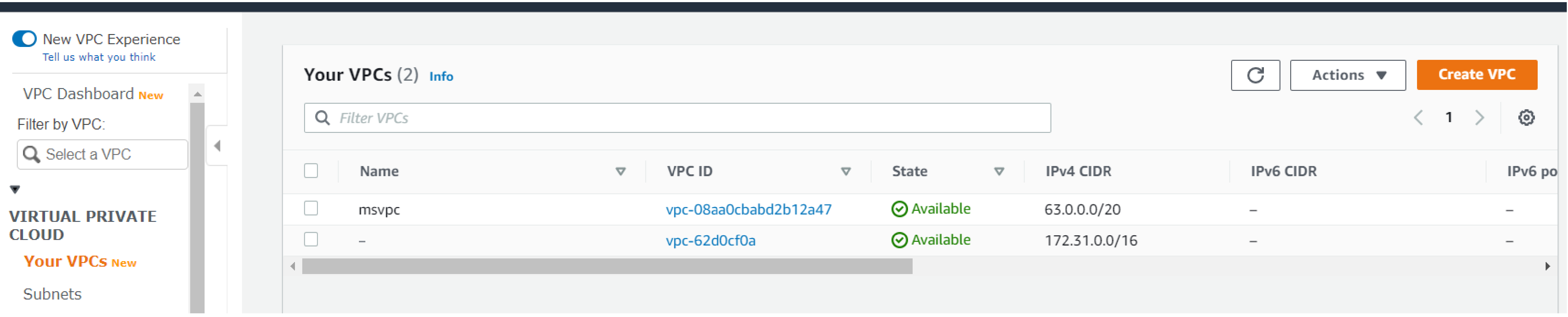
```
PS C:\terraform_0.13.4_windows_amd64 (1)> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

aws_vpc.example: Refreshing state... [id=vpc-069e3f67353167fc4]

------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_vpc.example will be created
  + resource "aws_vpc" "example" {
      + arn                              = (known after apply)
      + assign_generated_ipv6_cidr_block = false
      + 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_classiclink               = (known after apply)
      + enable_classiclink_dns_support   = (known after apply)
      + enable_dns_hostnames             = (known after apply)
      + enable_dns_support               = true
      + id                               = (known after apply)
      + instance_tenancy                 = "default"
      + ipv6_association_id              = (known after apply)
      + ipv6_cidr_block                  = (known after apply)
      + main_route_table_id              = (known after apply)
      + owner_id                         = (known after apply)
    }

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

------------------------------------------------------------------------

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that these actions will be performed if
"terraform apply" is subsequently run.

PS C:\terraform_0.13.4_windows_amd64 (1)>
```

Sample script to launch a VPC



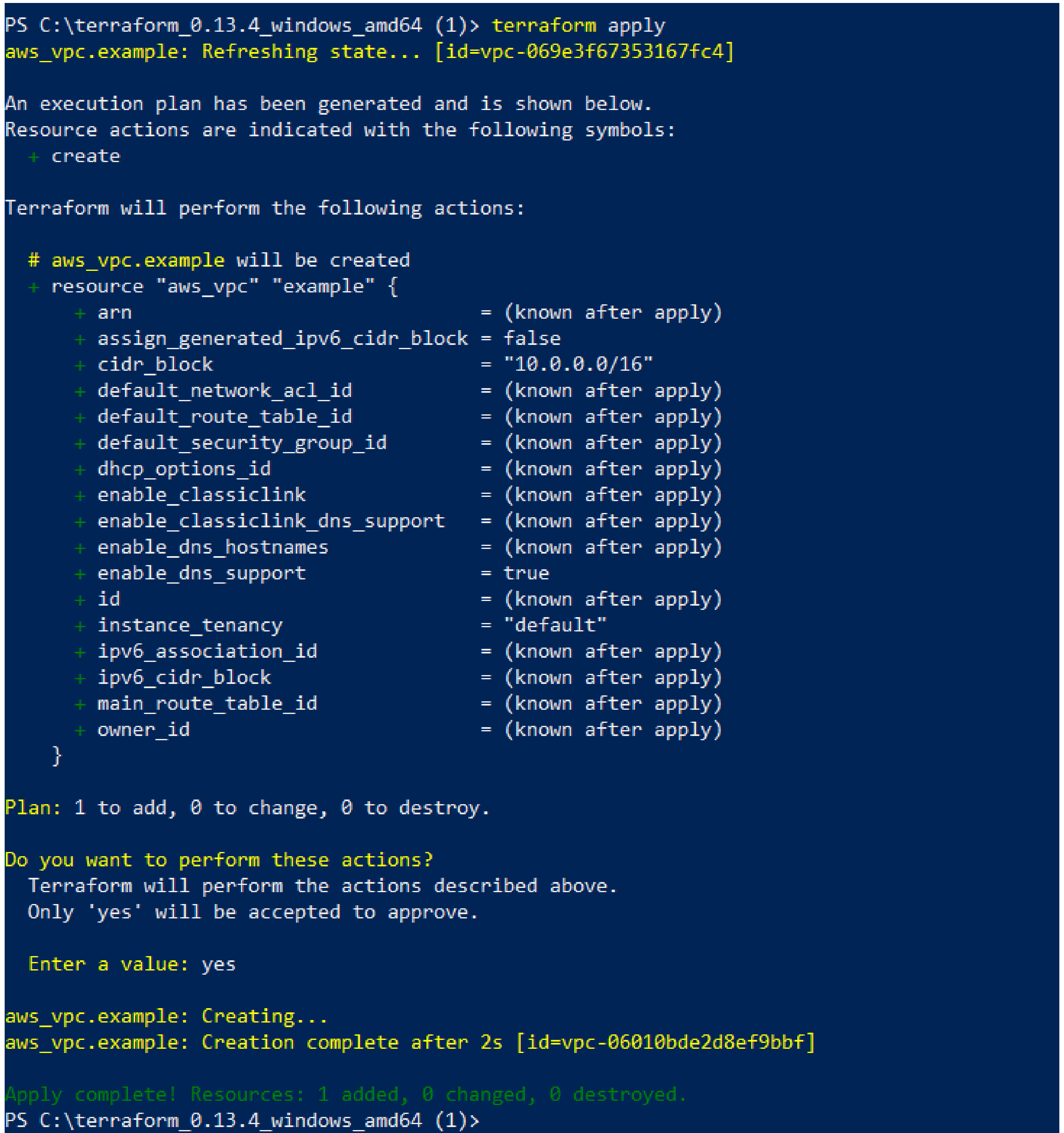
```
# Configure the AWS Provider
provider "aws" {
  region = "ap-south-1"
}

# Create a VPC
resource "aws_vpc" "example" {
  cidr_block = "10.0.0.0/16"
}
```

Before running the “**terraform apply**” my AWS VPC just has 2 VPC created with CIDR block

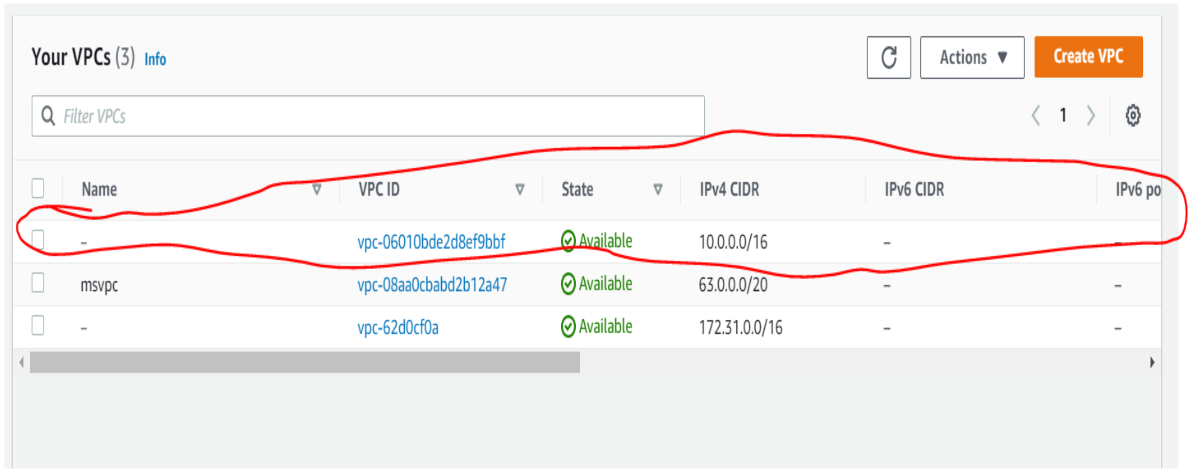


	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	IPv6 po
	msvpc	vpc-08aa0cbabd2b12a47	Available	63.0.0.0/20	–	–
	–	vpc-62d0cf0a	Available	172.31.0.0/16	–	–

Step 3 – Terraform apply



```
PS C:\terraform_0.13.4_windows_amd64 (1)> terraform apply
aws_vpc.example: Refreshing state... [id=vpc-069e3f67353167fc4]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_vpc.example will be created
  + resource "aws_vpc" "example" {
      + arn                              = (known after apply)
      + assign_generated_ipv6_cidr_block = false
      + 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_classiclink               = (known after apply)
      + enable_classiclink_dns_support   = (known after apply)
      + enable_dns_hostnames             = (known after apply)
      + enable_dns_support               = true
      + id                               = (known after apply)
      + instance_tenancy                 = "default"
      + ipv6_association_id              = (known after apply)
      + ipv6_cidr_block                  = (known after apply)
      + main_route_table_id              = (known after apply)
      + owner_id                         = (known after apply)
    }

Plan: 1 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_vpc.example: Creating...
aws_vpc.example: Creation complete after 2s [id=vpc-06010bde2d8ef9bbf]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\terraform_0.13.4_windows_amd64 (1)>
```

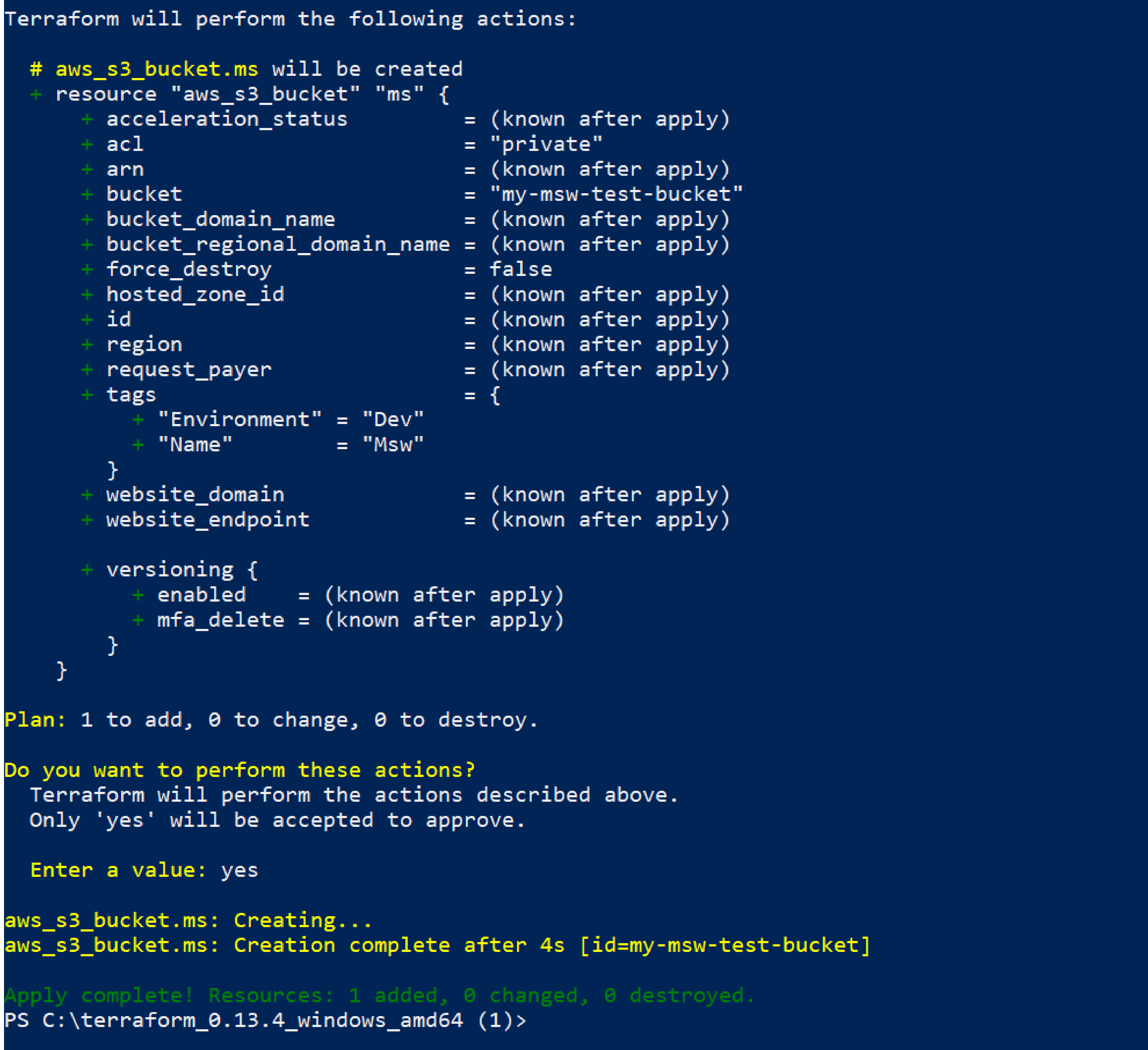


Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	IPv6 po
-	vpc-06010bde2d8ef9bbf	Available	10.0.0.0/16	-	-
msvpc	vpc-08aa0cbabd2b12a47	Available	63.0.0.0/20	-	-
-	vpc-62d0cf0a	Available	172.31.0.0/16	-	-

Script for creating a Bucket

```
resource "aws_s3_bucket" "ms" {
bucket = "my-msw-test-bucket"
acl    = "private"

 tags = {
Name        = "Msw"
Environment = "Dev"
}
}
```

Powershell Output



```
Terraform will perform the following actions:

  # aws_s3_bucket.ms will be created
  + resource "aws_s3_bucket" "ms" {
      + acceleration_status         = (known after apply)
      + acl                         = "private"
      + arn                         = (known after apply)
      + bucket                      = "my-msw-test-bucket"
      + bucket_domain_name          = (known after apply)
      + bucket_regional_domain_name = (known after apply)
      + force_destroy               = false
      + hosted_zone_id              = (known after apply)
      + id                          = (known after apply)
      + region                      = (known after apply)
      + request_payer               = (known after apply)
      + tags                        = {
          + "Environment" = "Dev"
          + "Name"        = "Msw"
        }
      + website_domain              = (known after apply)
      + website_endpoint            = (known after apply)

      + versioning {
          + enabled    = (known after apply)
          + mfa_delete = (known after apply)
        }
    }

Plan: 1 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_s3_bucket.ms: Creating...
aws_s3_bucket.ms: Creation complete after 4s [id=my-msw-test-bucket]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\terraform_0.13.4_windows_amd64 (1)>
```