



**AWS using Terraform**

Parth Narendrabhai Patel (002933850)

CPS Informatics, Northeastern University

ITC 6345: System and Network Administration

Dr. Horia Humaila

October 26, 2023

## AWS and Terraform installation Check

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\parth> aws --version
aws-cli/2.13.26 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
PS C:\Users\parth> terraform --version
Terraform v1.6.1
on windows_386
```

## AWS account configuration and checking

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\parth> aws --version
aws-cli/2.13.26 Python/3.11.6 Windows/10 exe/AMD64 prompt/off
PS C:\Users\parth> terraform --version
Terraform v1.6.1
on windows_386

Your version of Terraform is out of date! The latest version
is 1.6.2. You can update by downloading from https://www.terraform.io/downloads.html
PS C:\Users\parth> aws configure
AWS Access Key ID [*****TW27]: AKI/*****7XTW27
AWS Secret Access Key [*****a70M]: VFhgMe,*****570:IIt7UgOC5a70M
Default region name [us-east-2]: us-east-2
Default output format [json]: json
PS C:\Users\parth> aws s3 ls
2023-10-18 07:31:08 testingconfigneu
```

## Creation and Access of the directory

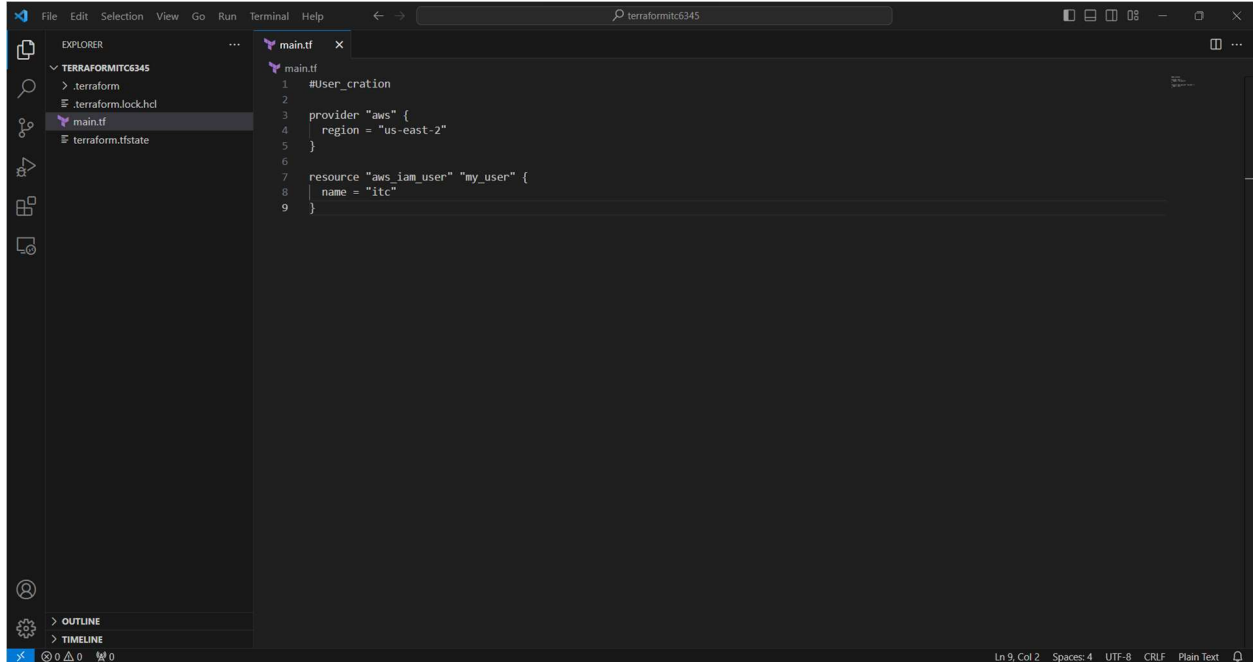
```
PS C:\Users\parth> mkdir terraformitc6345

Directory: C:\Users\parth

Mode                LastWriteTime         Length Name
----                -
d-----          10/25/2023   4:39 PM                terraformitc6345

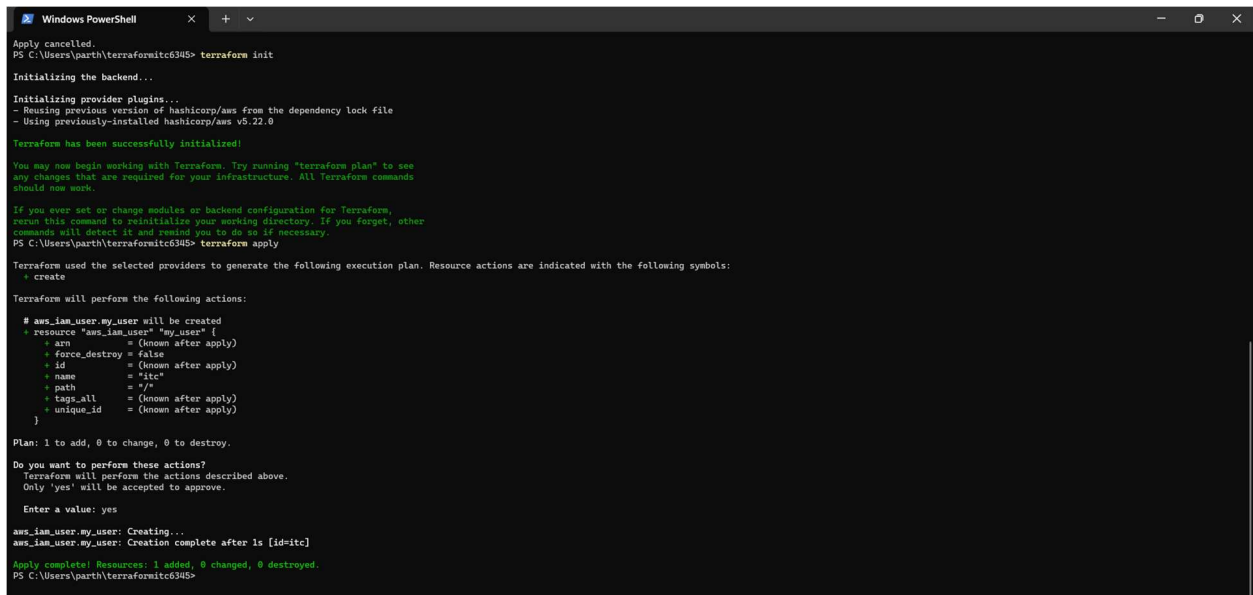
PS C:\Users\parth> cd terraformitc6345
PS C:\Users\parth\terraformitc6345>
```

## Creation of the IAM User in directory file named main.tf



```
1 #User_cration
2
3 provider "aws" {
4   region = "us-east-2"
5 }
6
7 resource "aws_iam_user" "my_user" {
8   name = "itc"
9 }
```

## Terraform Init and Terraform Apply



```
Apply cancelled.
PS C:\Users\parth\terraformitc6345> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.22.0

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:\Users\parth\terraformitc6345> 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_iam_user.my_user will be created
+ resource "aws_iam_user" "my_user" {
+   arn                = (known after apply)
+   force_destroy      = false
+   id                 = (known after apply)
+   name               = "itc"
+   path               = "/"
+   tags_all           = (known after apply)
+   unique_id          = (known after apply)
}

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

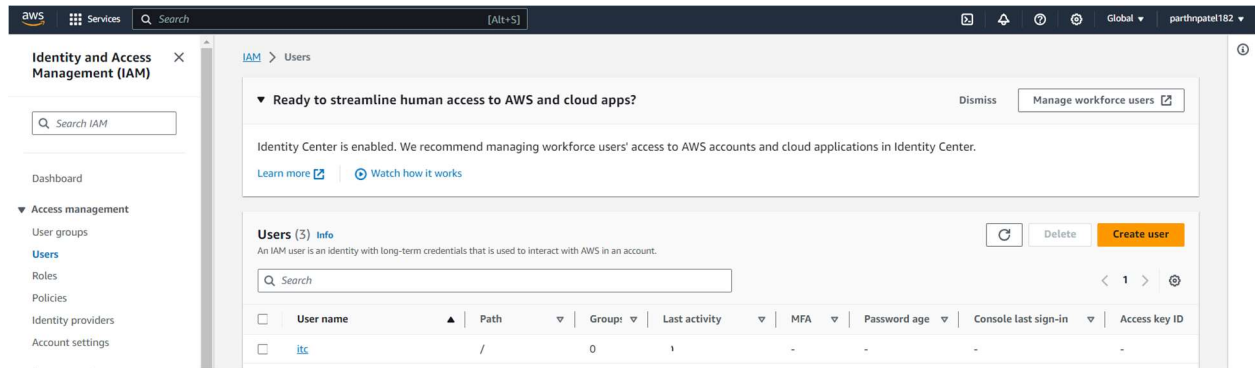
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

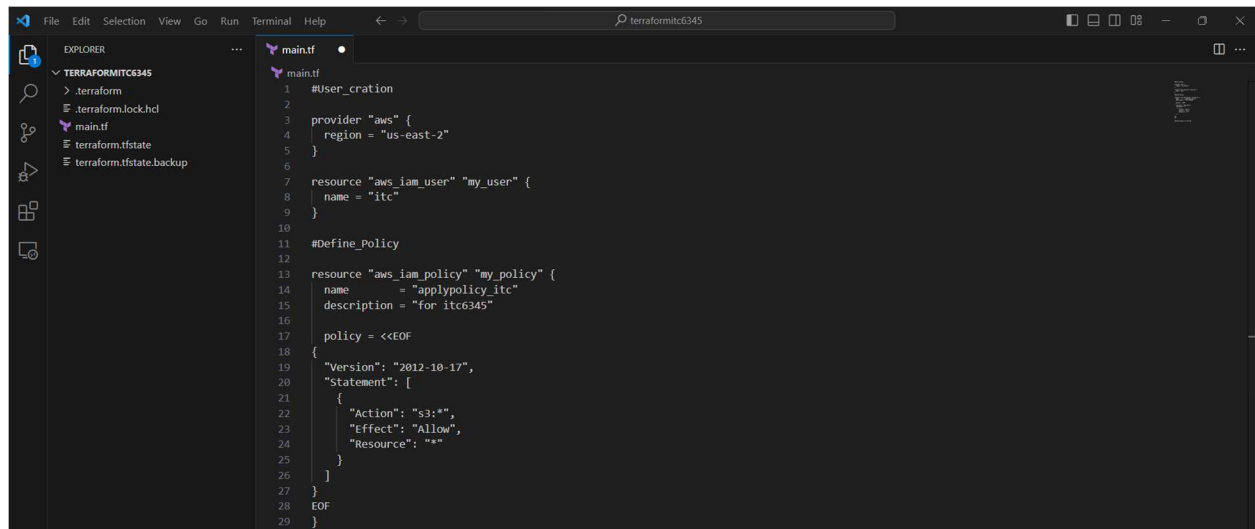
aws_iam_user.my_user: Creating...
aws_iam_user.my_user: Creation complete after 1s [id=itc]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\parth\terraformitc6345>
```

## AWS user created



## Policy code in main.tf



## Terraform init and apply for policy

```
Windows PowerShell
PS C:\Users\parth\terraform\itc6345> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.22.0

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:\Users\parth\terraform\itc6345> terraform apply
aws_iam_user.my_user: Refreshing state... [id=itc]

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_iam_policy.my_policy will be created
+ resource "aws_iam_policy" "my_policy" {
+   arn           = (known after apply)
+   description   = "for itc6345"
+   id            = (known after apply)
+   name          = "apolicy_its"
+   name_prefix   = (known after apply)
+   path          = "/"
+   policy        = jsonencode(
      {
        + Statement = [
          {
            + Action   = "s3:*"
            + Effect   = "Allow"
            + Resource = "*"
          },
        ]
        + Version = "2012-10-17"
      }
    )
+   policy_id     = (known after apply)
+   tags_all      = (known after apply)
}

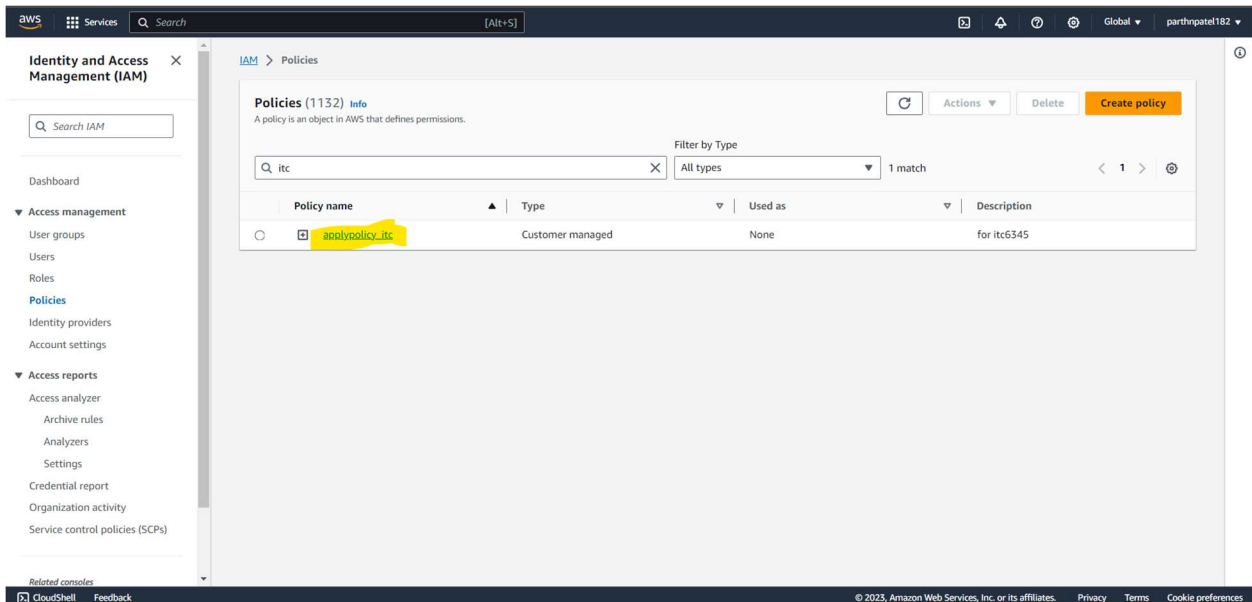
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

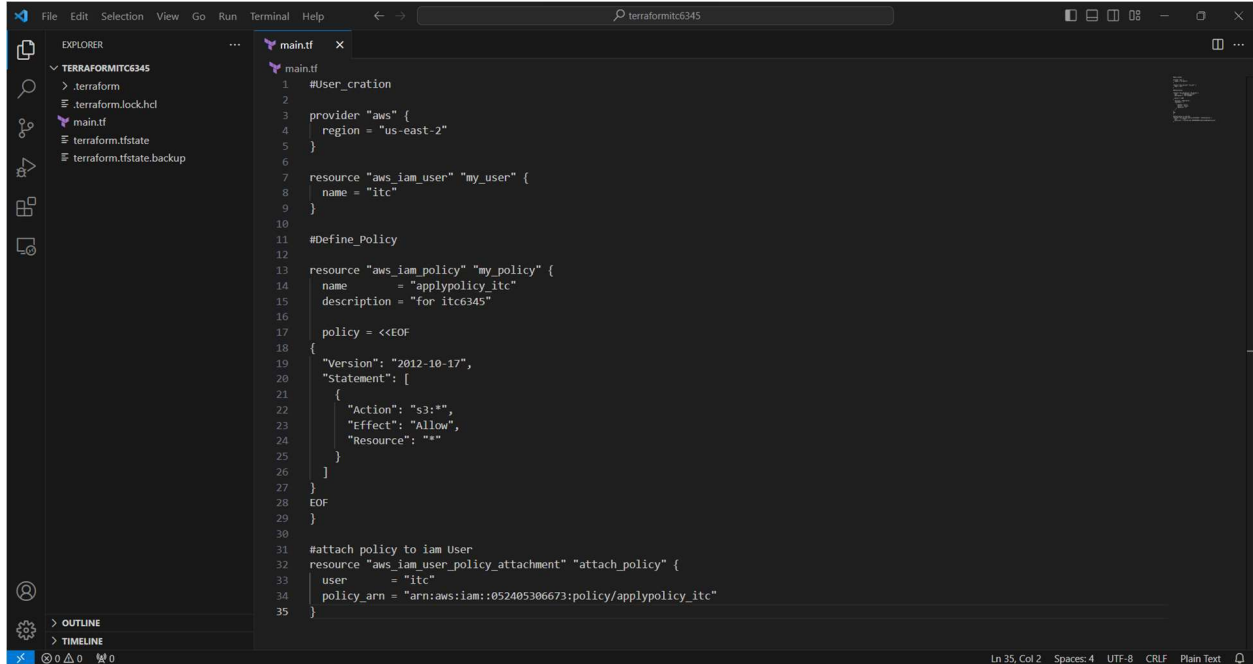
Enter a value: yes
aws_iam_policy.my_policy: Creating...
aws_iam_policy.my_policy: Creation complete after 0s [id=arn:aws:iam::052405306673:policy/apolicy_its]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\parth\terraform\itc6345>
```

## Policy Created



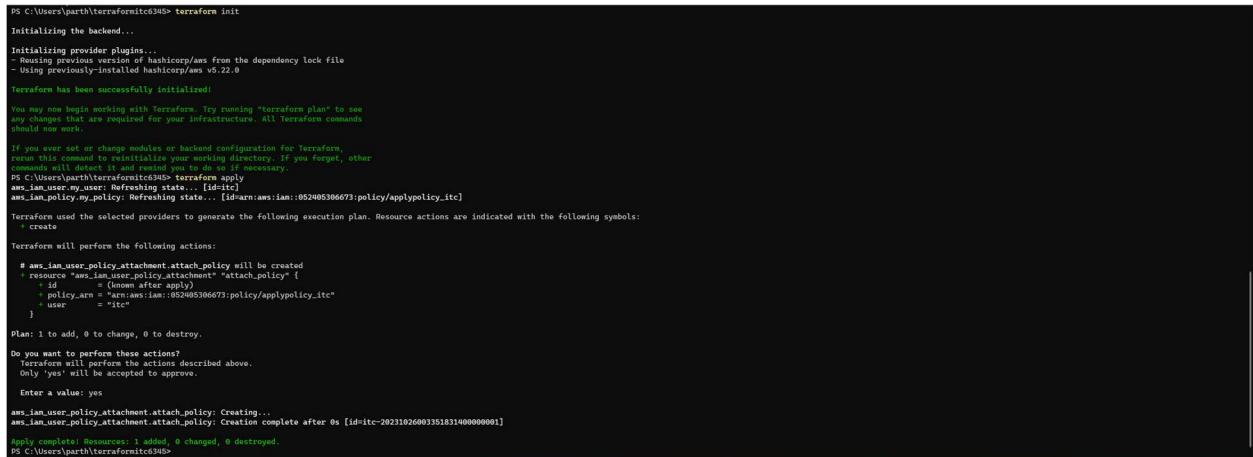
## Attachment policy to iam user code



```

1 #User_cration
2
3 provider "aws" {
4   region = "us-east-2"
5 }
6
7 resource "aws_iam_user" "my_user" {
8   name = "itc"
9 }
10
11 #Define_Policy
12
13 resource "aws_iam_policy" "my_policy" {
14   name = "applypolicy_itc"
15   description = "for itc6345"
16
17   policy = <<EOF
18 {
19   "Version": "2012-10-17",
20   "Statement": [
21     {
22       "Action": "s3:*",
23       "Effect": "Allow",
24       "Resource": "*"
25     }
26   ]
27 }
28 EOF
29 }
30
31 #attach policy to iam User
32 resource "aws_iam_user_policy_attachment" "attach_policy" {
33   user = "itc"
34   policy_arn = "arn:aws:iam::052405306673:policy/applypolicy_itc"
35 }
  
```

## terraform init and apply for the iam user



```

PS C:\Users\parth\terraformitc6345> terraform init
Initializing the backend...
Initializing provider plugins...
- Installing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.22.0

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,
run 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:\Users\parth\terraformitc6345> terraform apply
aws_iam_user.my_user: Refreshing state... [id=itc]
aws_iam_policy.my_policy: Refreshing state... [id=arn:aws:iam::052405306673:policy/applypolicy_itc]

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_iam_user_policy_attachment.attach_policy will be created
+ resource "aws_iam_user_policy_attachment" "attach_policy" {
+   id = (known after apply)
+   policy_arn = "arn:aws:iam::052405306673:policy/applypolicy_itc"
+   user = "itc"
+ }

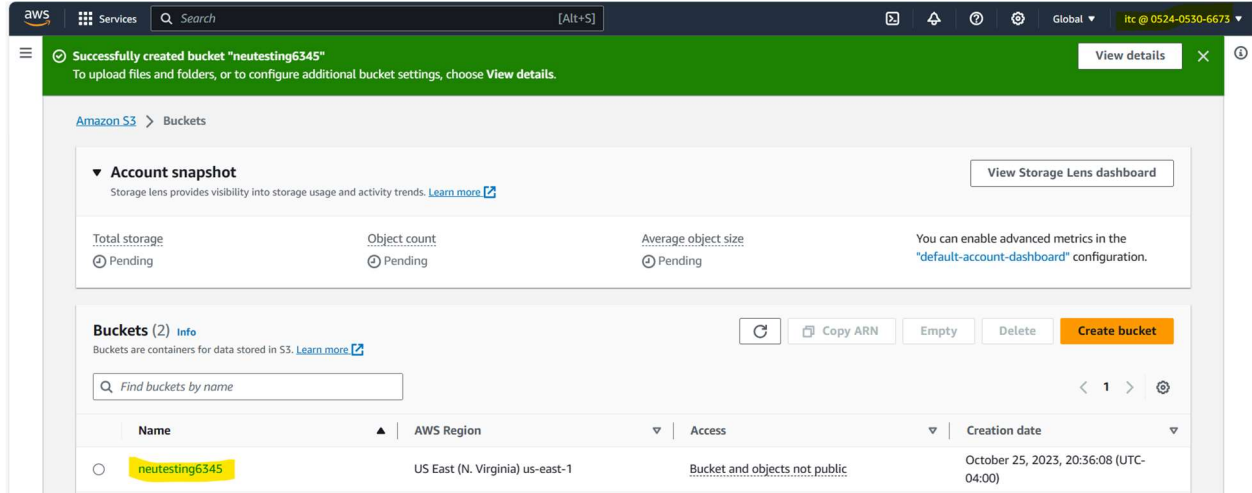
Plan: 1 to add, 0 to change, 0 to destroy.

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_iam_user_policy_attachment.attach_policy: Creating...
aws_iam_user_policy_attachment.attach_policy: Creation complete after 0s [id=itc-2023102600033518314000000001]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\parth\terraformitc6345>
  
```

## Able to crate s3 bucket using iam user



Now I am revoking the rights of all s3 bucket rights by commenting on the `attachment_policy` part of code then

## Terraform apply

```
PS C:\Users\parth\terraform\itc6345> terraform apply
aws_iam_user_policy_attachment.attach_policy: Refreshing state... [id=itc-20231026011223786600000001]
aws_iam_user_policy_attachment.detach_policy: Refreshing state... [id=itc-20231026011939428300000001]
aws_iam_policy.my_policy: Refreshing state... [id=arn:aws:iam::052405306673:policy/applypolicy_itc]
aws_iam_user.my_user: Refreshing state... [id=itc]

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_iam_user_policy_attachment.detach_policy will be destroyed
# (because aws_iam_user_policy_attachment.detach_policy is not in configuration)
- resource "aws_iam_user_policy_attachment" "detach_policy" {
  id          = "itc-20231026011939428300000001" -> null
  policy_arn  = "arn:aws:iam::052405306673:policy/applypolicy_itc" -> null
  user       = "itc" -> null
}

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

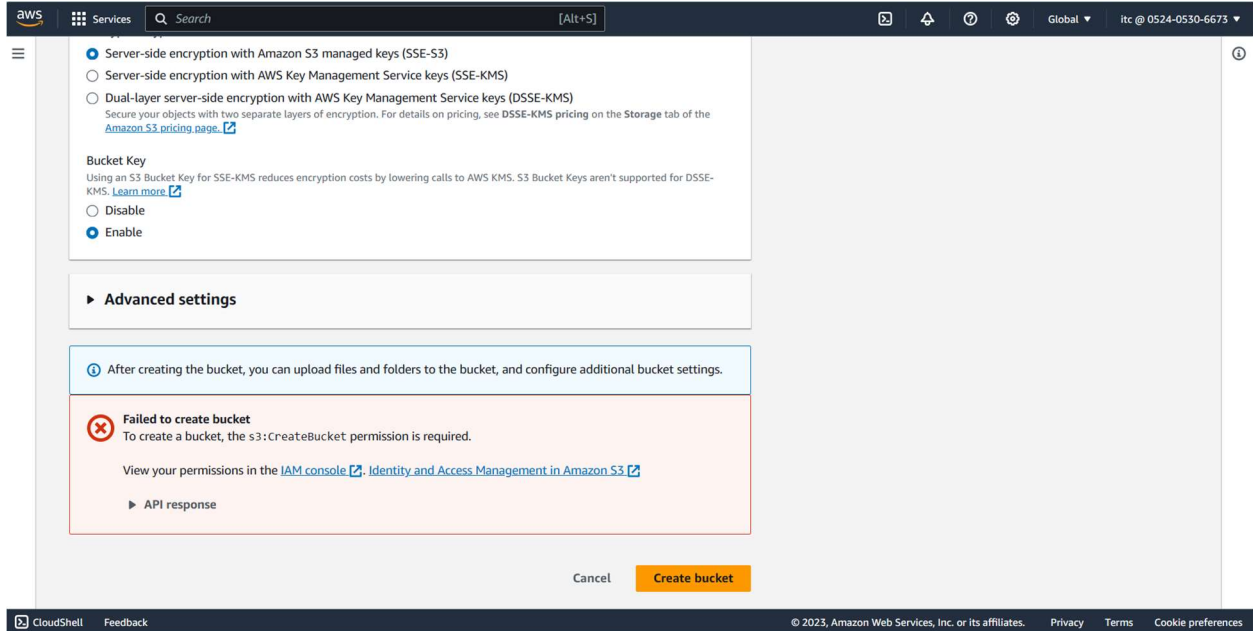
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

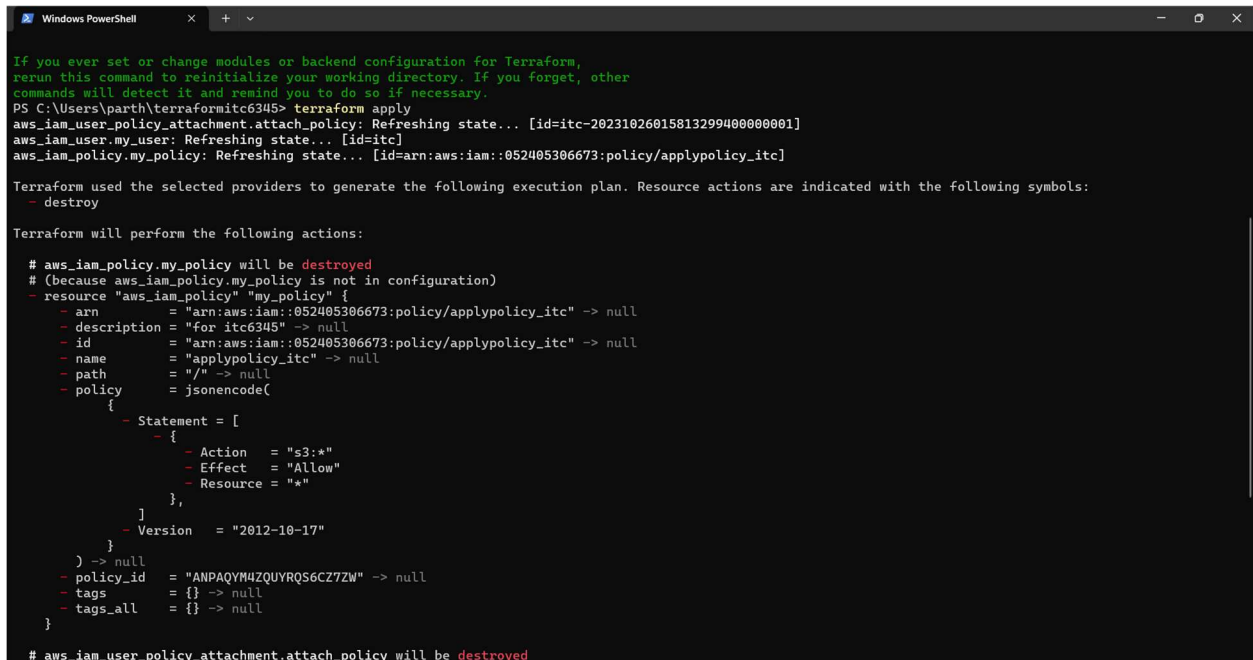
aws_iam_user_policy_attachment.detach_policy: Destroying... [id=itc-20231026011939428300000001]
aws_iam_user_policy_attachment.detach_policy: Destruction complete after 0s

Apply complete! Resources: 0 added, 0 changed, 1 destroyed.
PS C:\Users\parth\terraform\itc6345> |
```

## Not able to create s3 bucket

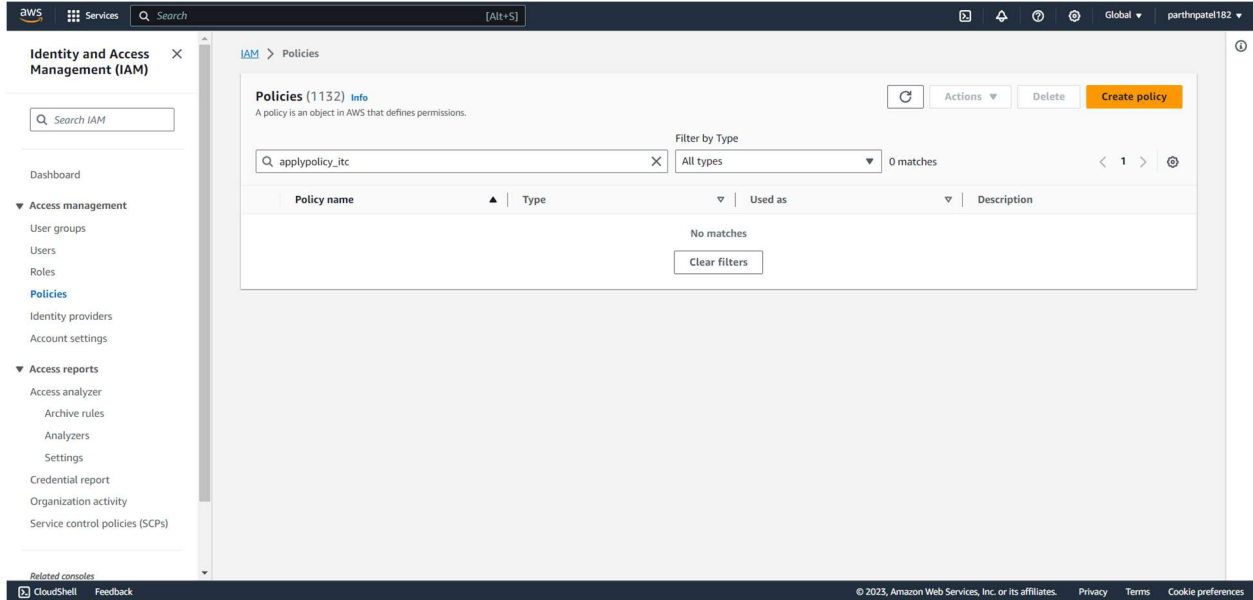


## Now by commenting apply\_policy. The do terraform apply to delete policy





## Policy destroyed



## Then I did the terraform.destroy

```
PS C:\Users\parth\terraformitc6345> terraform destroy
aws_iam_user.my_user: Refreshing state... [id=itc]

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_iam_user.my_user will be destroyed
- resource "aws_iam_user" "my_user" {
  - arn              = "arn:aws:iam::052405306673:user/itc" -> null
  - force_destroy    = false -> null
  - id               = "itc" -> null
  - name             = "itc" -> null
  - path             = "/" -> null
  - tags             = {} -> null
  - tags_all         = {} -> null
  - unique_id        = "AIDAQYM4ZQUYY65RYVGZ7" -> null
}

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

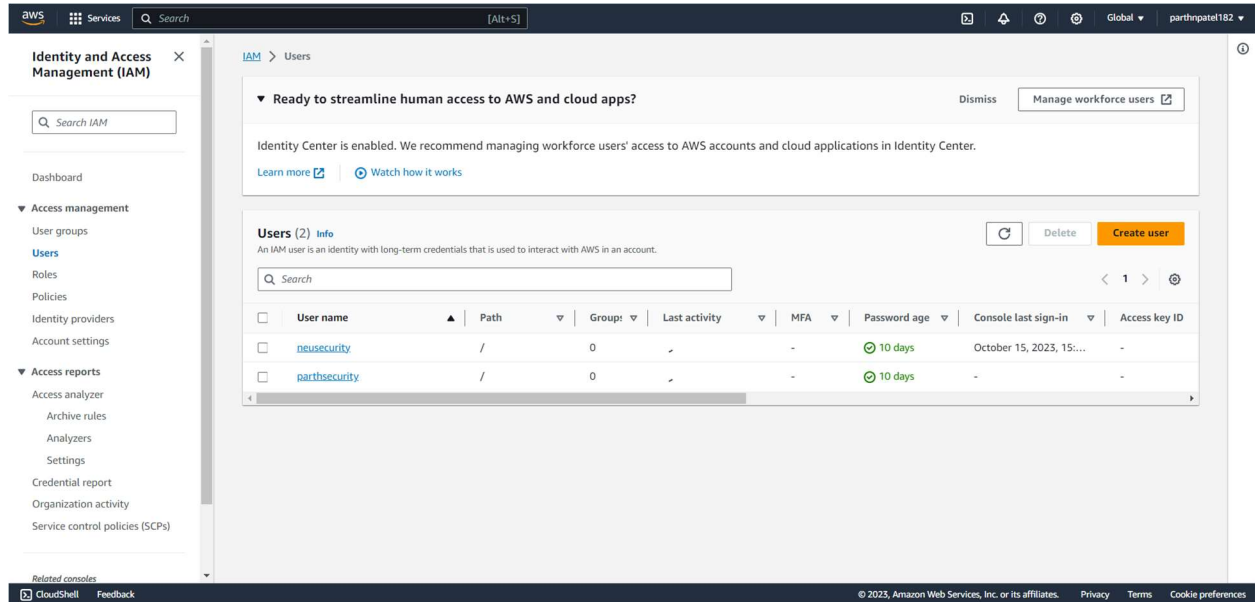
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_iam_user.my_user: Destroying... [id=itc]
aws_iam_user.my_user: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
PS C:\Users\parth\terraformitc6345>
```

## IAM user named itc got deleted



The screenshot shows the AWS IAM console interface. The left sidebar contains the 'Identity and Access Management (IAM)' section with a search bar and a list of navigation items: Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, Service control policies (SCPs)). The main content area is titled 'IAM > Users' and features a notification banner about streamlining human access to AWS and cloud apps. Below this, there's a section for 'Users (2)' with a search bar and a table listing the users. The table has columns for checkboxes, User name, Path, Groups, Last activity, MFA, Password age, Console last sign-in, and Access key ID. Two users are listed: 'newsecurity' and 'parthsecurity'. The 'parthsecurity' user is highlighted, and its 'Last activity' column shows a green checkmark and '10 days', indicating it has been deleted. The bottom of the console shows the 'Related consoles' section with 'CloudShell' and 'Feedback' links, and the footer contains copyright information and links for Privacy, Terms, and Cookie preferences.

	User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID
<input type="checkbox"/>	newsecurity	/	0	-	-	10 days	October 15, 2023, 15...	-
<input type="checkbox"/>	parthsecurity	/	0	-	-	10 days	-	-