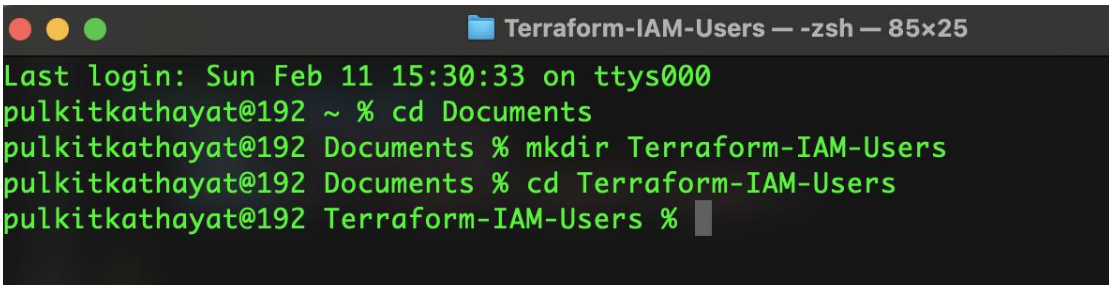
LAB–7

Creating Multiple IAM Users in Terraform

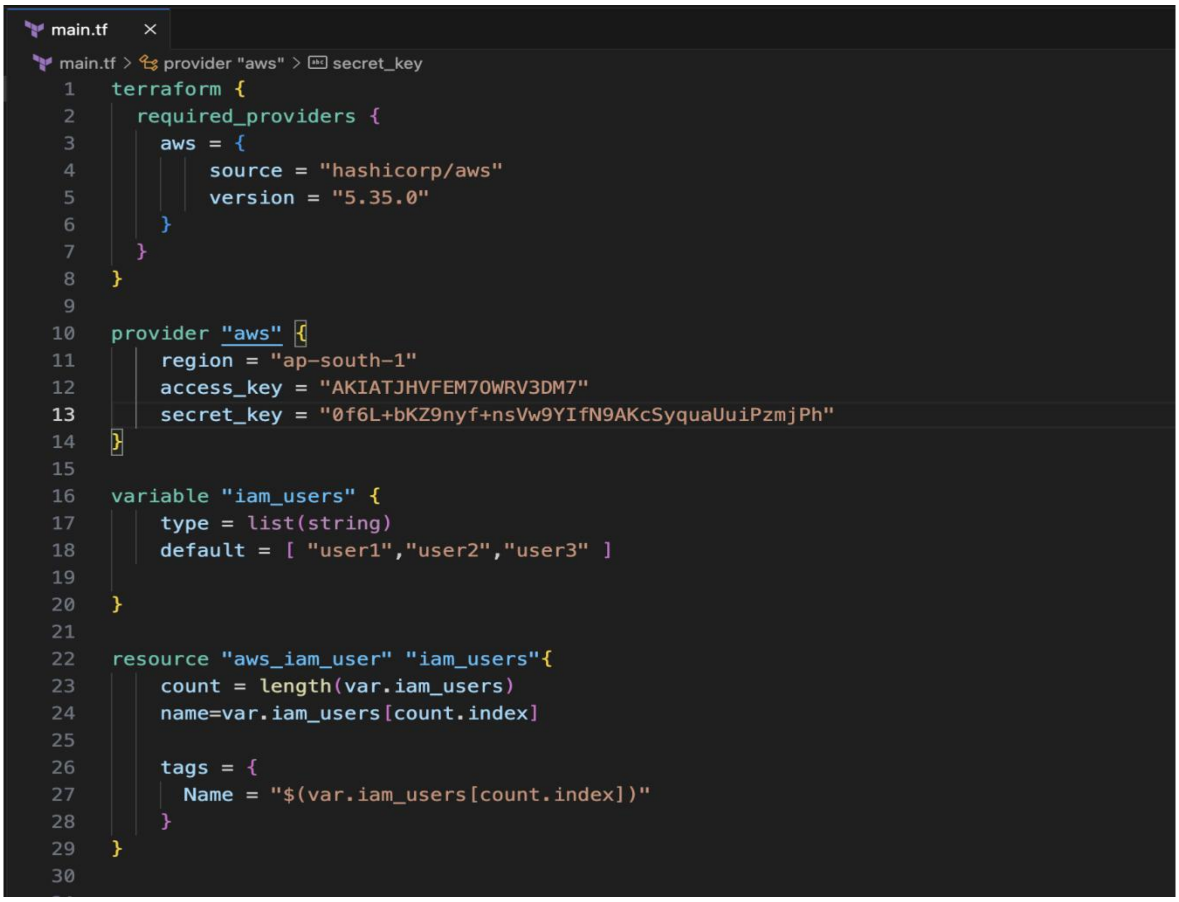
Step 1: Create a new Terraform IAM Users directory



Terraform-IAM-Users — -zsh — 85×25

```
Last login: Sun Feb 11 15:30:33 on ttys000
pulkitkathayat@192 ~ % cd Documents
pulkitkathayat@192 Documents % mkdir Terraform-IAM-Users
pulkitkathayat@192 Documents % cd Terraform-IAM-Users
pulkitkathayat@192 Terraform-IAM-Users %
```

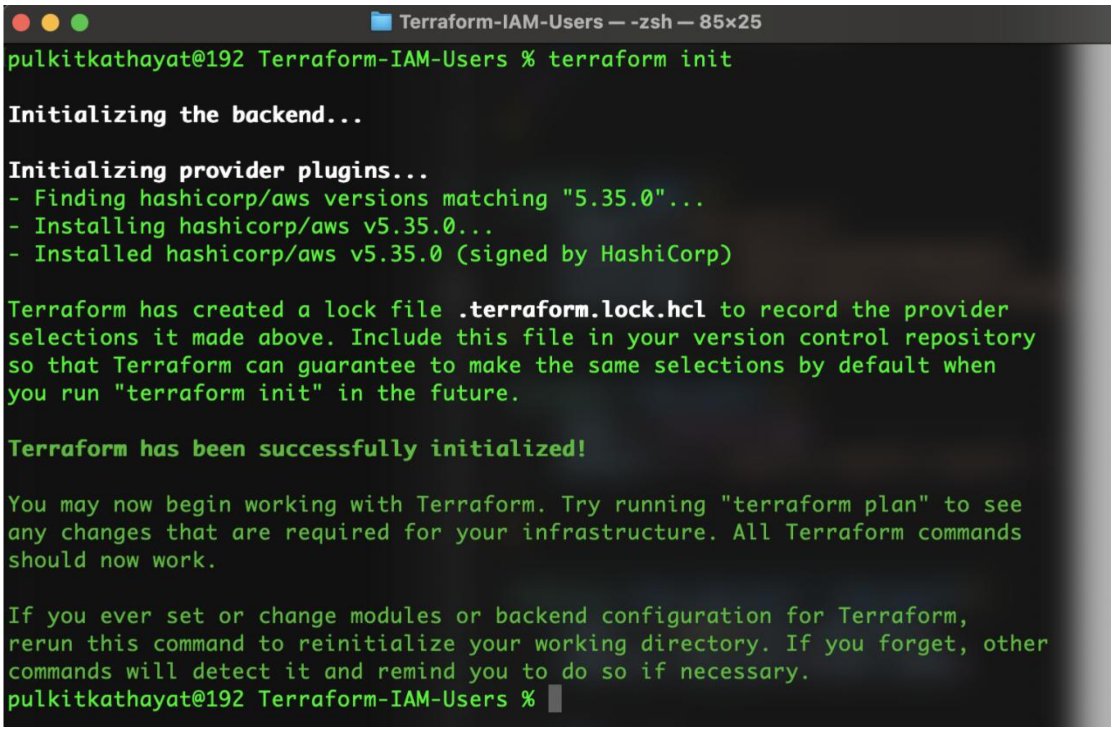
Step 2: Create a main.tf file



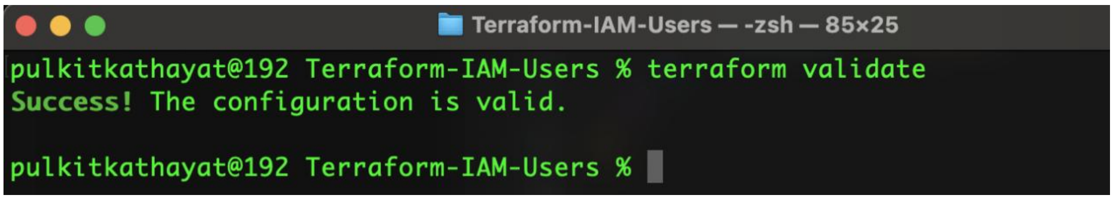
main.tf > provider "aws" > secret_key

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.35.0"
    }
  }
}

provider "aws" {
    region = "ap-south-1"
    access_key = "AKIATJHVFEM7OWRV3DM7"
    secret_key = "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"
}

variable "iam_users" {
    type = list(string)
    default = [ "user1","user2","user3" ]

}

resource "aws_iam_user" "iam_users"{
    count = length(var.iam_users)
    name=var.iam_users[count.index]

    tags = {
      Name = "$(var.iam_users[count.index])"
    }
}
```

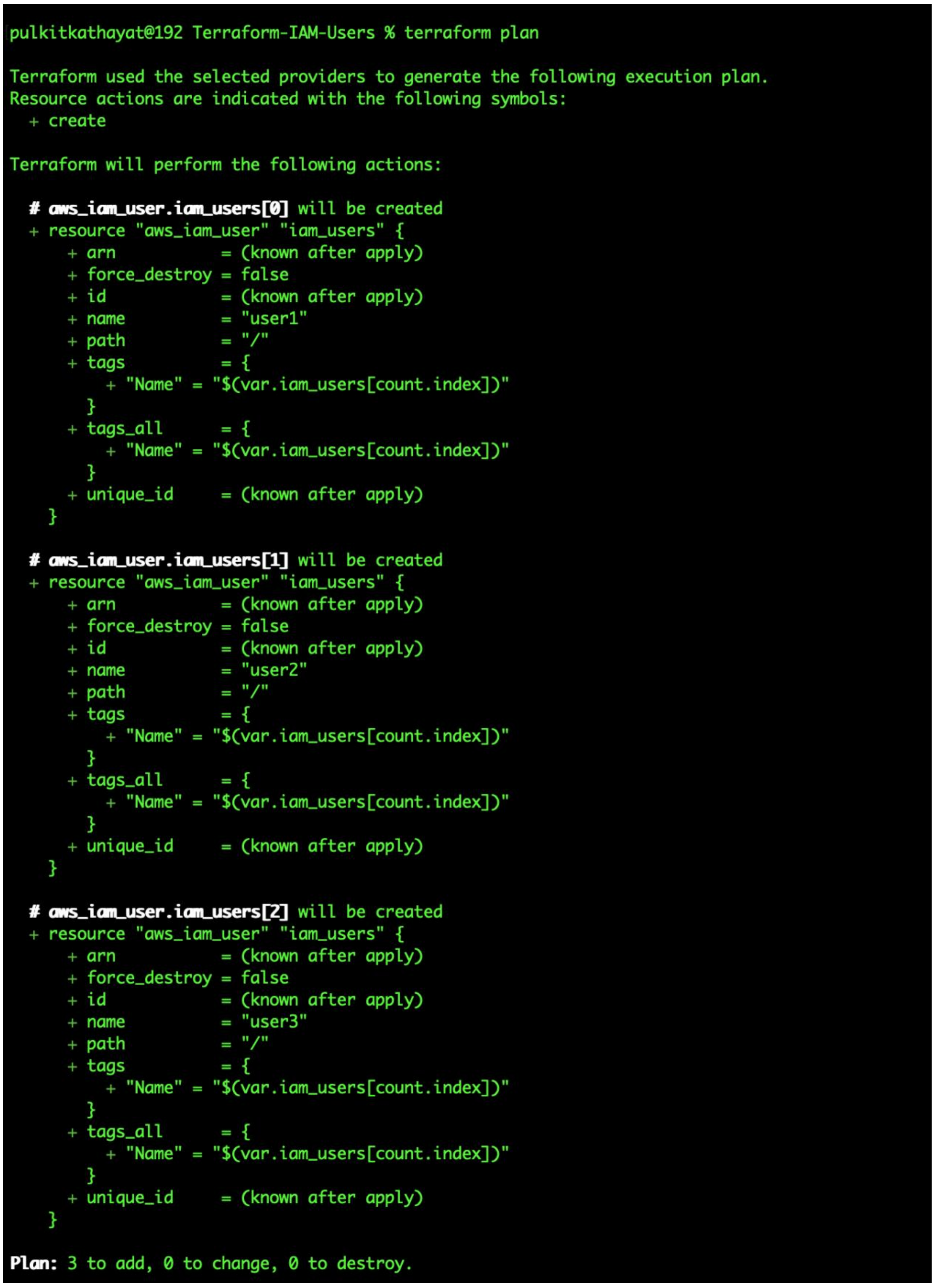
Step 3: Initialize and plan



```
pulkitkathayat@192 Terraform-IAM-Users % terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.35.0"...
- Installing hashicorp/aws v5.35.0...
- Installed hashicorp/aws v5.35.0 (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.
pulkitkathayat@192 Terraform-IAM-Users %
```

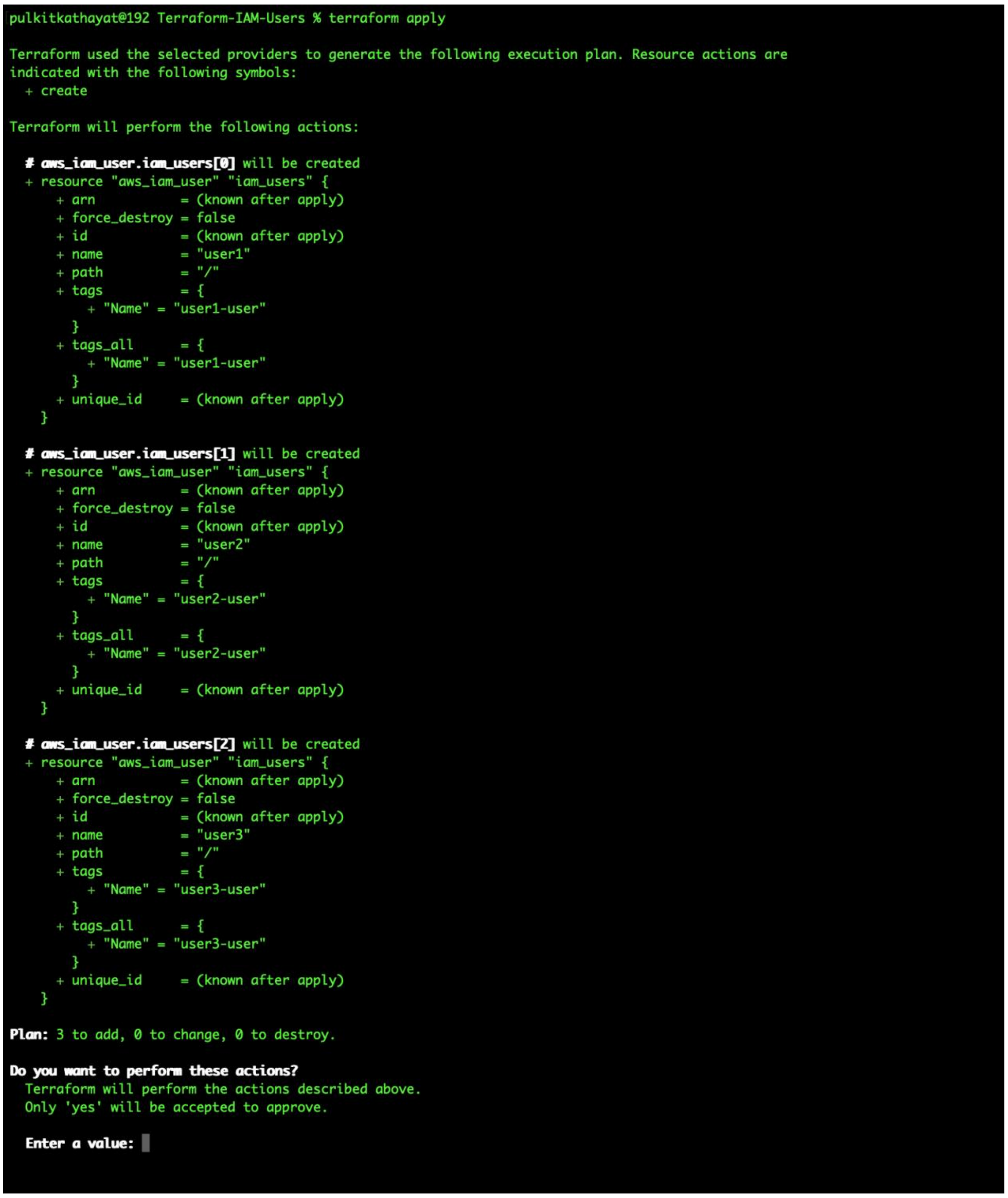


```
pulkitkathayat@192 Terraform-IAM-Users % terraform validate
Success! The configuration is valid.

pulkitkathayat@192 Terraform-IAM-Users %
```

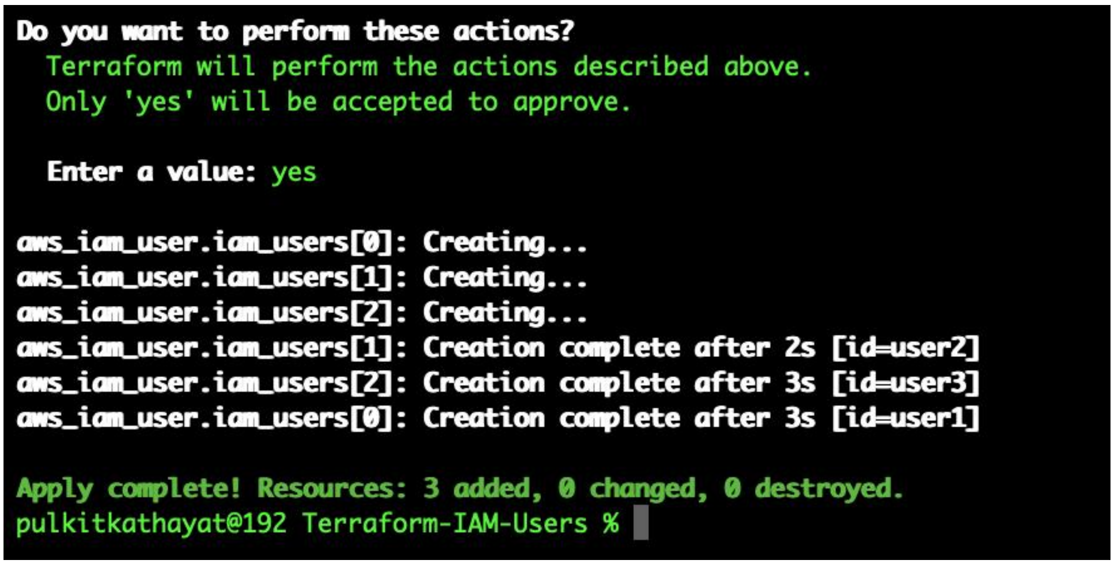


```
pulkitkathayat@192 Terraform-IAM-Users % terraform plan

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_iam_user.iam_users[0] will be created
  + resource "aws_iam_user" "iam_users" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "user1"
      + path          = "/"
      + tags          = {
          + "Name" = "$(var.iam_users[count.index])"
        }
      + tags_all      = {
          + "Name" = "$(var.iam_users[count.index])"
        }
      + unique_id     = (known after apply)
    }

  # aws_iam_user.iam_users[1] will be created
  + resource "aws_iam_user" "iam_users" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "user2"
      + path          = "/"
      + tags          = {
          + "Name" = "$(var.iam_users[count.index])"
        }
      + tags_all      = {
          + "Name" = "$(var.iam_users[count.index])"
        }
      + unique_id     = (known after apply)
    }

  # aws_iam_user.iam_users[2] will be created
  + resource "aws_iam_user" "iam_users" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "user3"
      + path          = "/"
      + tags          = {
          + "Name" = "$(var.iam_users[count.index])"
        }
      + tags_all      = {
          + "Name" = "$(var.iam_users[count.index])"
        }
      + unique_id     = (known after apply)
    }

Plan: 3 to add, 0 to change, 0 to destroy.
```

Step 4: Apply

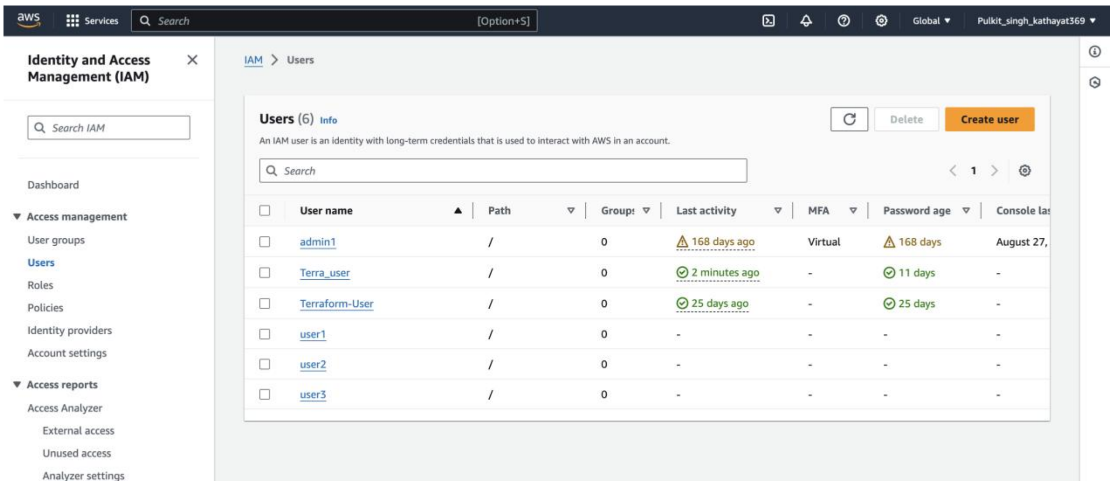


```
pulkitkathayat@192 Terraform-IAM-Users % terraform apply

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.iam_users[0] will be created
  + resource "aws_iam_user" "iam_users" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "user1"
      + path          = "/"
      + tags          = {
          + "Name" = "user1-user"
        }
      + tags_all      = {
          + "Name" = "user1-user"
        }
      + unique_id     = (known after apply)
    }

  # aws_iam_user.iam_users[1] will be created
  + resource "aws_iam_user" "iam_users" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "user2"
      + path          = "/"
      + tags          = {
          + "Name" = "user2-user"
        }
      + tags_all      = {
          + "Name" = "user2-user"
        }
      + unique_id     = (known after apply)
    }

  # aws_iam_user.iam_users[2] will be created
  + resource "aws_iam_user" "iam_users" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "user3"
      + path          = "/"
      + tags          = {
          + "Name" = "user3-user"
        }
      + tags_all      = {
          + "Name" = "user3-user"
        }
      + unique_id     = (known after apply)
    }

Plan: 3 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: 
```



```
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.iam_users[0]: Creating...
aws_iam_user.iam_users[1]: Creating...
aws_iam_user.iam_users[2]: Creating...
aws_iam_user.iam_users[1]: Creation complete after 2s [id=user2]
aws_iam_user.iam_users[2]: Creation complete after 3s [id=user3]
aws_iam_user.iam_users[0]: Creation complete after 3s [id=user1]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
pulkitkathayat@192 Terraform-IAM-Users %
```

Step 5: Verify Users in AWS Console



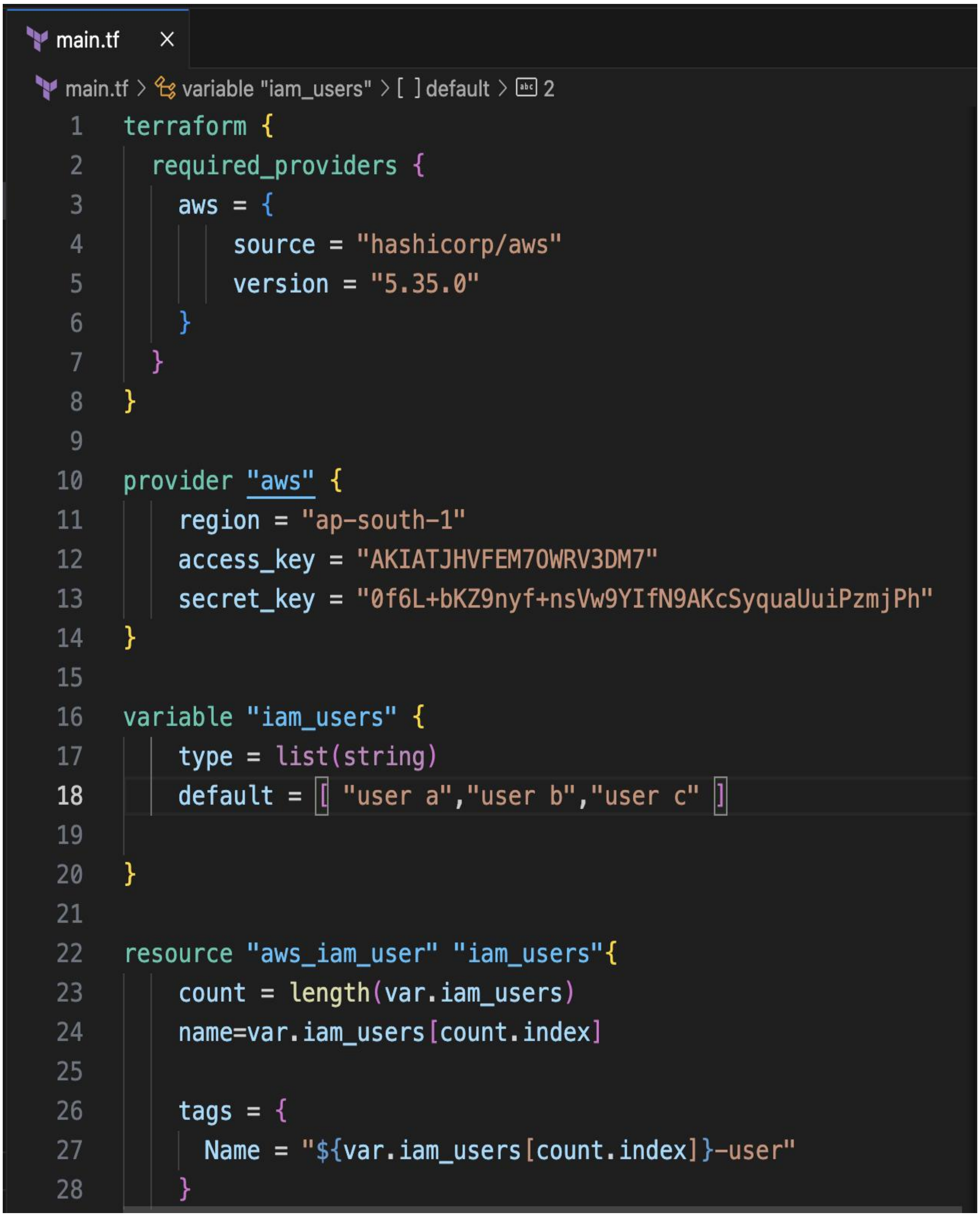
Users (6) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

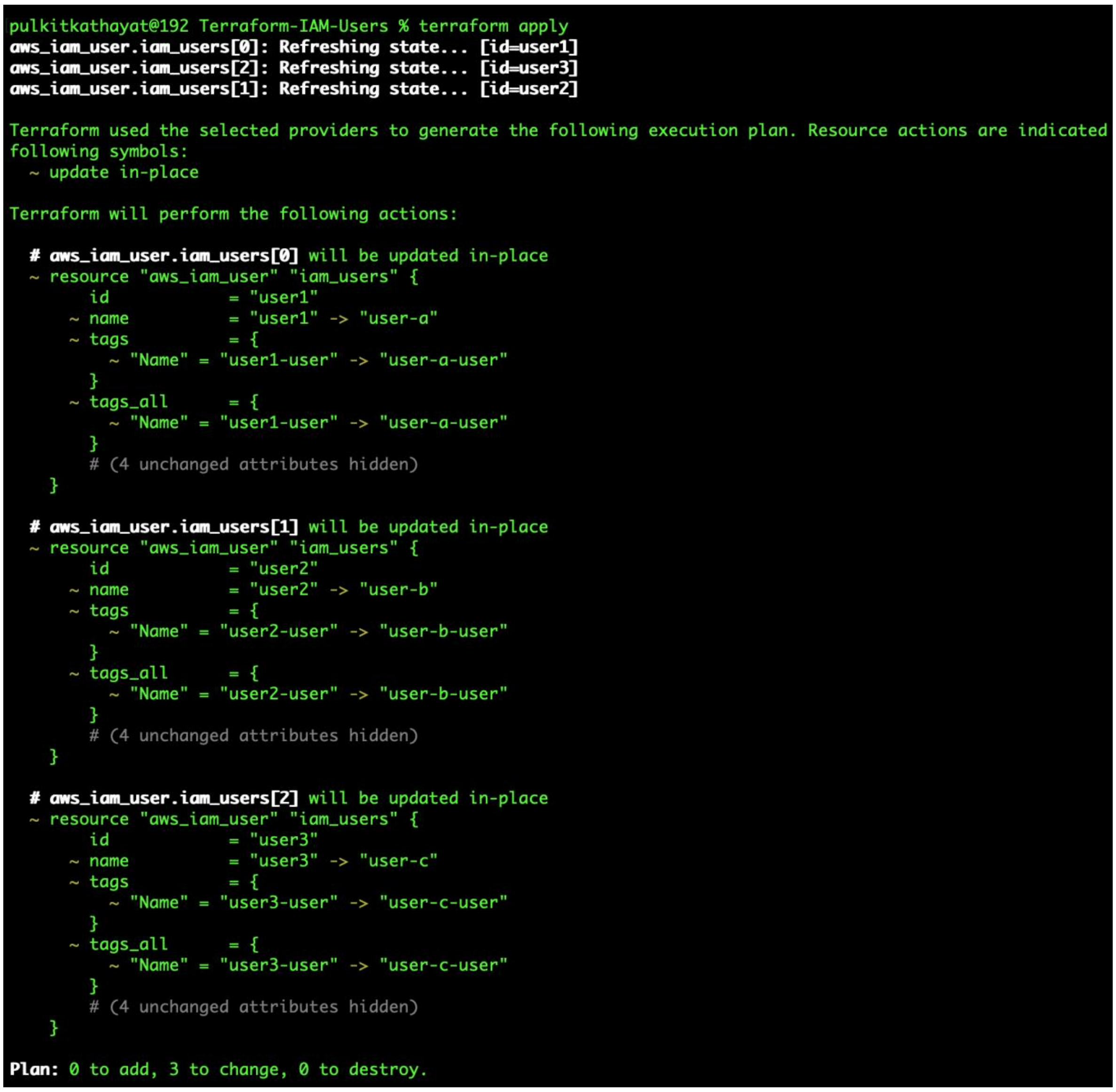
User name	Path	Group:	Last activity	MFA	Password age	Console las
admin1	/	0	⚠ 168 days ago	Virtual	⚠ 168 days	August 27,
Terra_user	/	0	✓ 2 minutes ago	-	✓ 11 days	-
Terraform-User	/	0	✓ 25 days ago	-	✓ 25 days	-
user1	/	0	-	-	-	-
user2	/	0	-	-	-	-
user3	/	0	-	-	-	-

Step 6: Add or remove IAM user

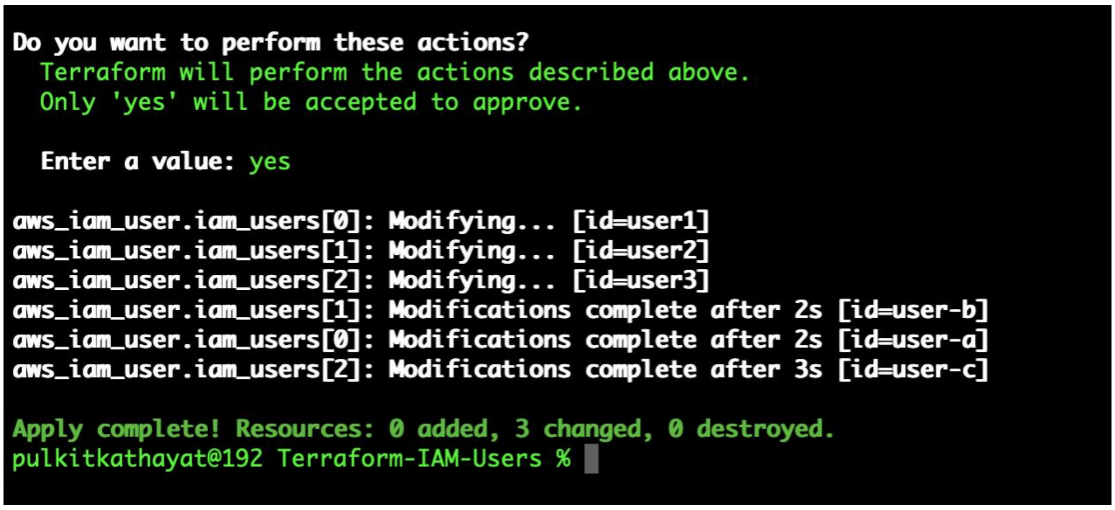
Modify the main.tf file to add or remove users the rerun terraform apply command to apply changes



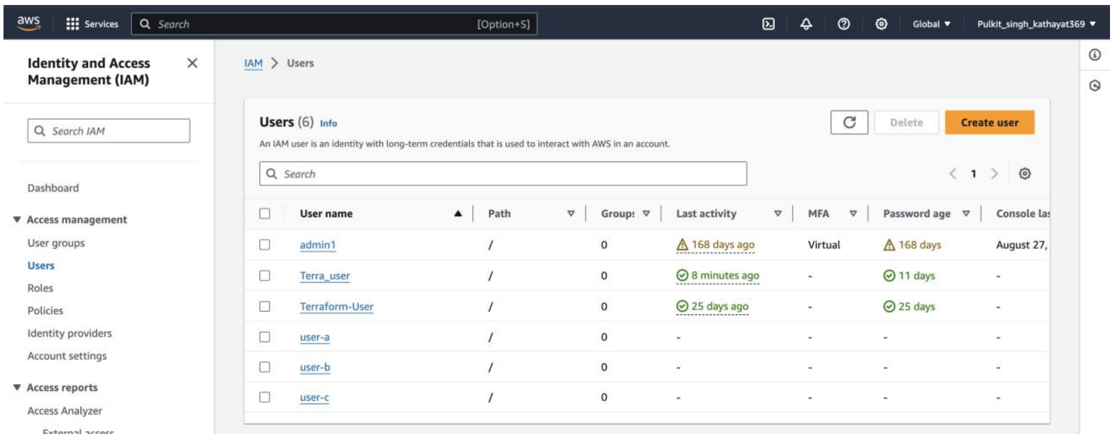
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.35.0"
    }
  }
}

provider "aws" {
    region = "ap-south-1"
    access_key = "AKIATJHVFEM7OWRV3DM7"
    secret_key = "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"
}

variable "iam_users" {
    type = list(string)
    default = [ "user a","user b","user c" ]

}

resource "aws_iam_user" "iam_users"{
    count = length(var.iam_users)
    name=var.iam_users[count.index]

    tags = {
      Name = "${var.iam_users[count.index]}-user"
    }
```



```
pulkitkathayat@192 Terraform-IAM-Users % terraform apply
aws_iam_user.iam_users[0]: Refreshing state... [id=user1]
aws_iam_user.iam_users[2]: Refreshing state... [id=user3]
aws_iam_user.iam_users[1]: Refreshing state... [id=user2]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated
following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_iam_user.iam_users[0] will be updated in-place
  ~ resource "aws_iam_user" "iam_users" {
        id            = "user1"
      ~ name          = "user1" -> "user-a"
      ~ tags          = {
          ~ "Name" = "user1-user" -> "user-a-user"
        }
      ~ tags_all      = {
          ~ "Name" = "user1-user" -> "user-a-user"
        }
        # (4 unchanged attributes hidden)
    }

  # aws_iam_user.iam_users[1] will be updated in-place
  ~ resource "aws_iam_user" "iam_users" {
        id            = "user2"
      ~ name          = "user2" -> "user-b"
      ~ tags          = {
          ~ "Name" = "user2-user" -> "user-b-user"
        }
      ~ tags_all      = {
          ~ "Name" = "user2-user" -> "user-b-user"
        }
        # (4 unchanged attributes hidden)
    }

  # aws_iam_user.iam_users[2] will be updated in-place
  ~ resource "aws_iam_user" "iam_users" {
        id            = "user3"
      ~ name          = "user3" -> "user-c"
      ~ tags          = {
          ~ "Name" = "user3-user" -> "user-c-user"
        }
      ~ tags_all      = {
          ~ "Name" = "user3-user" -> "user-c-user"
        }
        # (4 unchanged attributes hidden)
    }

Plan: 0 to add, 3 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.iam_users[0]: Modifying... [id=user1]
aws_iam_user.iam_users[1]: Modifying... [id=user2]
aws_iam_user.iam_users[2]: Modifying... [id=user3]
aws_iam_user.iam_users[1]: Modifications complete after 2s [id=user-b]
aws_iam_user.iam_users[0]: Modifications complete after 2s [id=user-a]
aws_iam_user.iam_users[2]: Modifications complete after 3s [id=user-c]

Apply complete! Resources: 0 added, 3 changed, 0 destroyed.
pulkitkathayat@192 Terraform-IAM-Users %
```

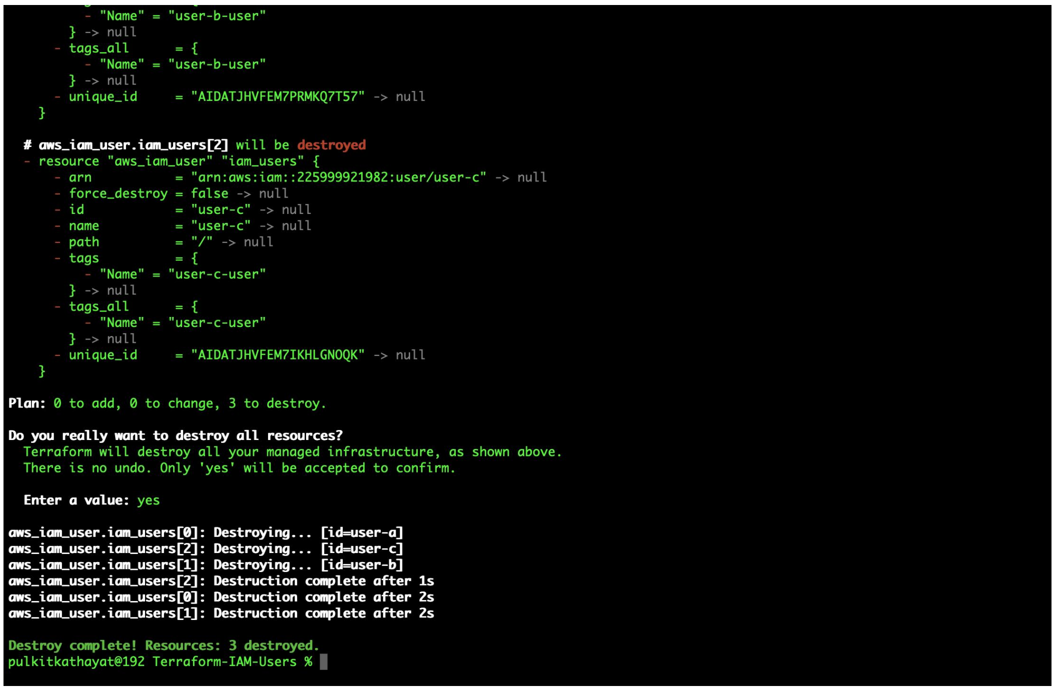


Users (6) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

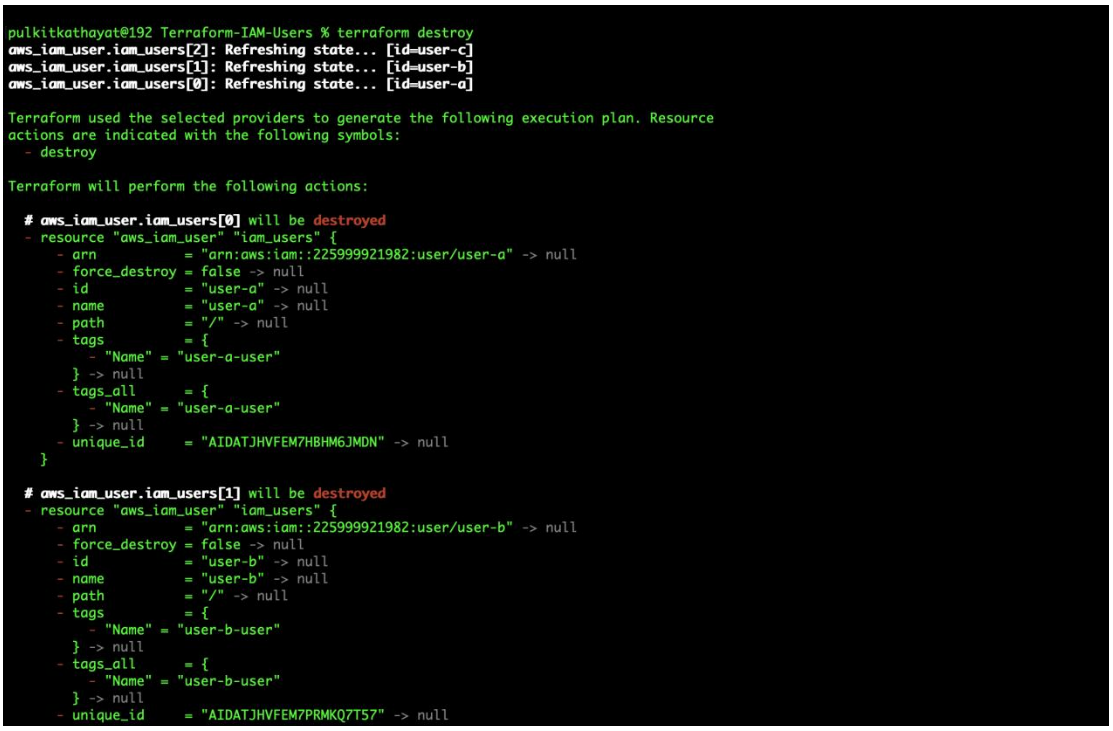
User name	Path	Groups	Last activity	MFA	Password age	Console las
admin1	/	0	168 days ago	Virtual	168 days	August 27,
Terra_user	/	0	8 minutes ago	-	11 days	-
Terraform-User	/	0	25 days ago	-	25 days	-
user-a	/	0	-	-	-	-
user-b	/	0	-	-	-	-
user-c	/	0	-	-	-	-

Step Clean Up

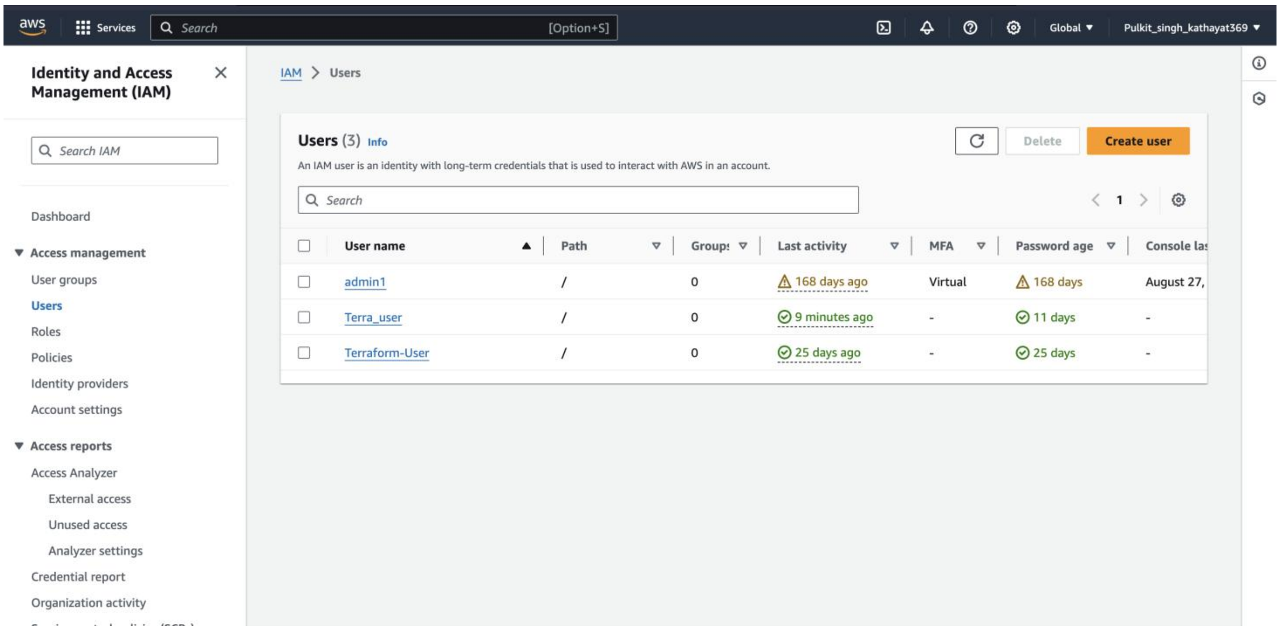


```
          - "Name" = "user-b-user"
        } -> null
      - tags_all         = {
          - "Name" = "user-b-user"
        } -> null
      - unique_id        = "AIDATJHVFEM7PRMKQ7T57" -> null
    }

  # aws_iam_user.iam_users[2] will be destroyed
  - resource "aws_iam_user" "iam_users" {
      - arn              = "arn:aws:iam::225999921982:user/user-c" -> null
      - force_destroy = false -> null
      - id               = "user-c" -> null
      - name             = "user-c" -> null
      - path             = "/" -> null
      - tags             = {
          - "Name" = "user-c-user"
        } -> null
      - tags_all         = {
          - "Name" = "user-c-user"
        } -> null
      - unique_id        = "AIDATJHVFEM7IKHLGNOQK" -> null
    }

Plan: 0 to add, 0 to change, 3 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.iam_users[0]: Destroying... [id=user-a]
aws_iam_user.iam_users[2]: Destroying... [id=user-c]
aws_iam_user.iam_users[1]: Destroying... [id=user-b]
aws_iam_user.iam_users[2]: Destruction complete after 1s
aws_iam_user.iam_users[0]: Destruction complete after 2s
aws_iam_user.iam_users[1]: Destruction complete after 2s

Destroy complete! Resources: 3 destroyed.
pulkitkathayat@192 Terraform-IAM-Users %
```

7:



```
pulkitkathayat@192 Terraform-IAM-Users % terraform destroy
aws_iam_user.iam_users[2]: Refreshing state... [id=user-c]
aws_iam_user.iam_users[1]: Refreshing state... [id=user-b]
aws_iam_user.iam_users[0]: Refreshing state... [id=user-a]

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.iam_users[0] will be destroyed
  - resource "aws_iam_user" "iam_users" {
      - arn           = "arn:aws:iam::225999921982:user/user-a" -> null
      - force_destroy = false -> null
      - id            = "user-a" -> null
      - name          = "user-a" -> null
      - path          = "/" -> null
      - tags          = {
          - "Name" = "user-a-user"
        } -> null
      - tags_all      = {
          - "Name" = "user-a-user"
        } -> null
      - unique_id     = "AIDATJHVFEM7HBHM6JMDN" -> null
    }

  # aws_iam_user.iam_users[1] will be destroyed
  - resource "aws_iam_user" "iam_users" {
      - arn           = "arn:aws:iam::225999921982:user/user-b" -> null
      - force_destroy = false -> null
      - id            = "user-b" -> null
      - name          = "user-b" -> null
      - path          = "/" -> null
      - tags          = {
          - "Name" = "user-b-user"
        } -> null
      - tags_all      = {
          - "Name" = "user-b-user"
        } -> null
      - unique_id     = "AIDATJHVFEM7PRMKQ7T57" -> null
```



Users (3) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password age	Console las
admin1	/	0	⚠ 168 days ago	Virtual	⚠ 168 days	August 27,
Terra_user	/	0	✓ 9 minutes ago	-	✓ 11 days	-
Terraform-User	/	0	✓ 25 days ago	-	✓ 25 days	-