

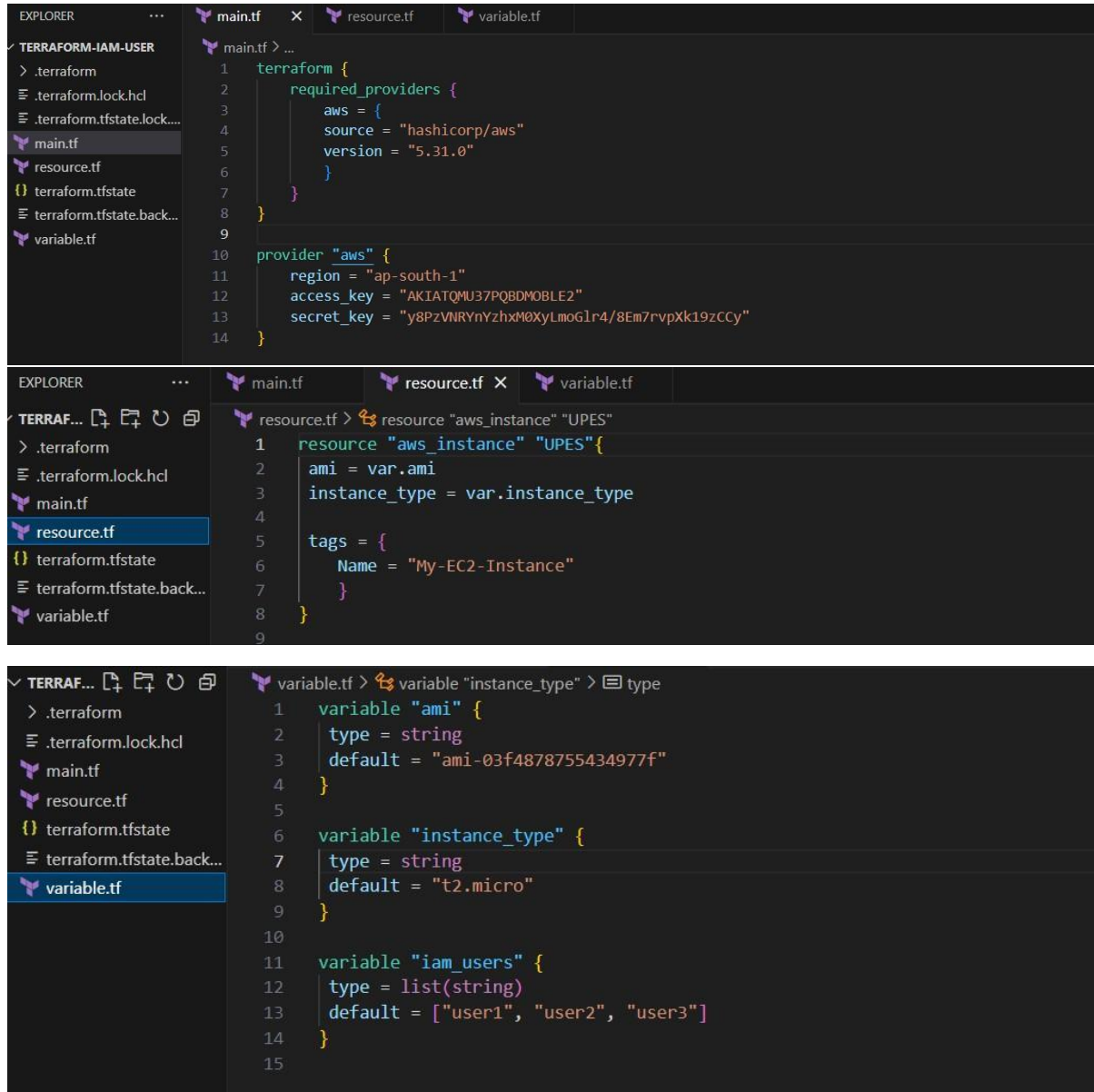
Lab Exercise 7– Creating Multiple IAM Users in Terraform

Anmol Ghai

500090959, R2142210118

B. Tech. CSE-DevOps – B1, 6th SEM

1. Create Terraform Configuration Files



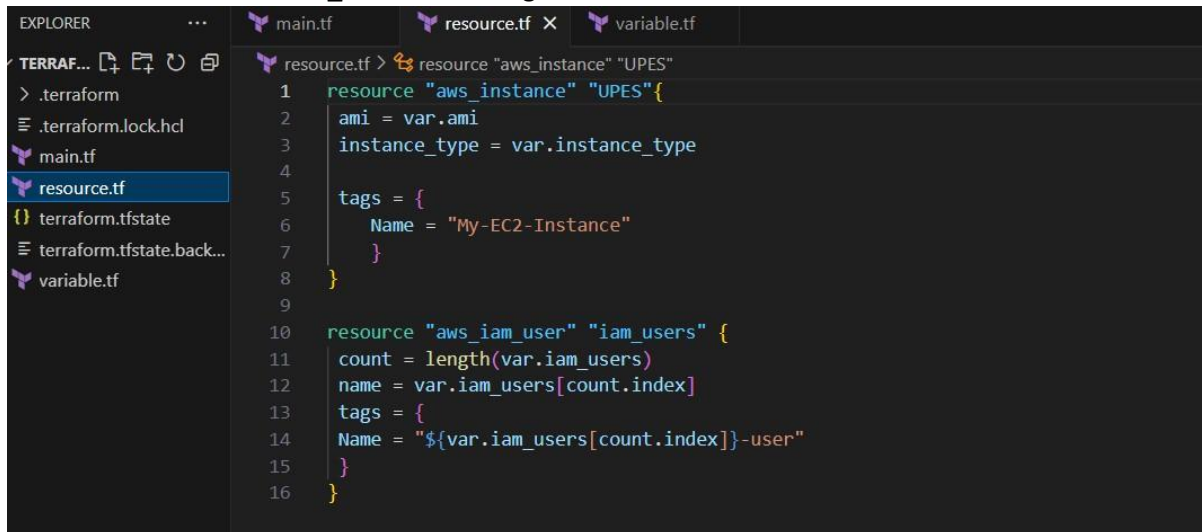
The image displays three screenshots of a code editor showing Terraform configuration files. The first screenshot shows the `main.tf` file with the `terraform` block and the `provider "aws"` configuration. The second screenshot shows the `resource.tf` file with the `resource "aws_instance" "UPES"` configuration. The third screenshot shows the `variable.tf` file with the `variable "ami"`, `variable "instance_type"`, and `variable "iam_users"` configurations.

```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "ap-south-1"
12   access_key = "AKIATQMU37PQBDMOBLE2"
13   secret_key = "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"
14 }
```

```
1 resource "aws_instance" "UPES"{
2   ami = var.ami
3   instance_type = var.instance_type
4
5   tags = {
6     Name = "My-EC2-Instance"
7   }
8 }
9
```

```
1 variable "ami" {
2   type = string
3   default = "ami-03f4878755434977f"
4 }
5
6 variable "instance_type" {
7   type = string
8   default = "t2.micro"
9 }
10
11 variable "iam_users" {
12   type = list(string)
13   default = ["user1", "user2", "user3"]
14 }
15
```

2. Define a list variable iam_users containing the names of the IAM users we want to create



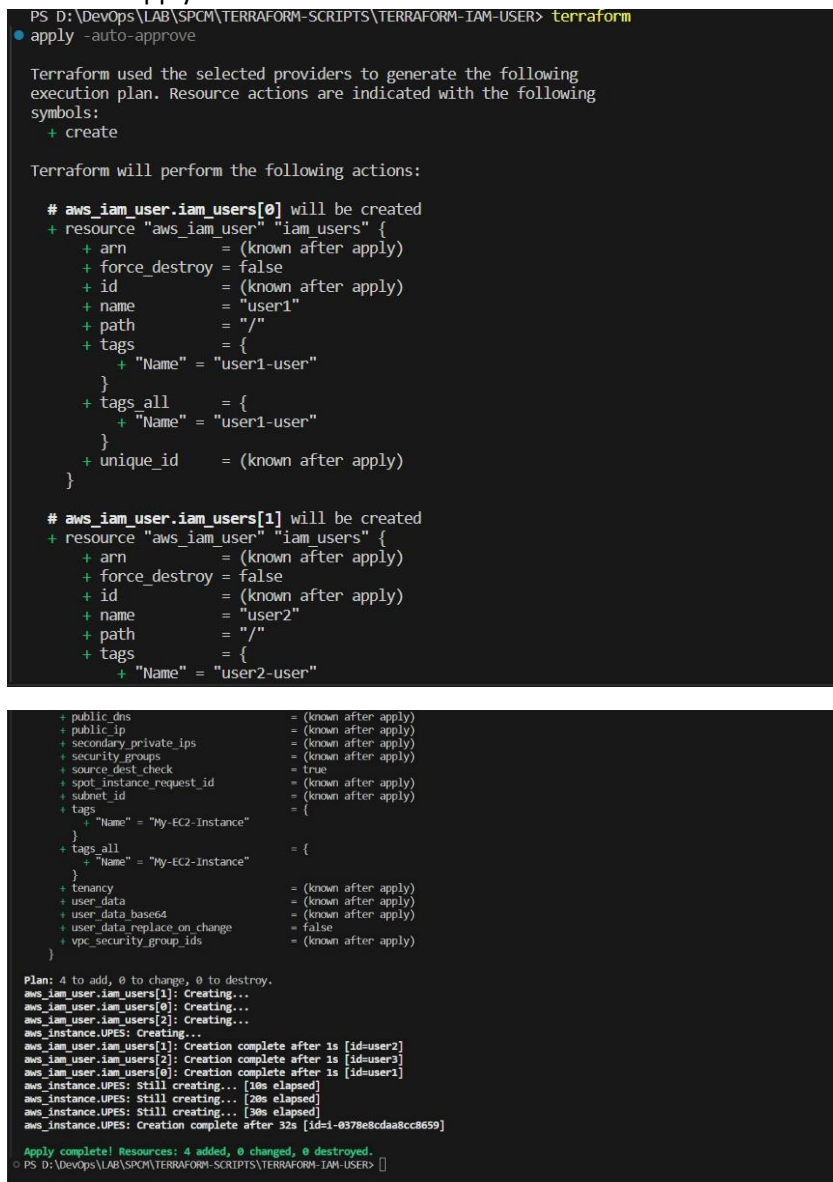
```
EXPLORER  ...  main.tf  resource.tf X  variable.tf

TERRAF...  .terraform  .terraform.lock.hcl  main.tf  resource.tf  {} terraform.tfstate  terraform.tfstate.back...  variable.tf

resource.tf  resource "aws_instance" "UPES"
1  resource "aws_instance" "UPES"{
2      ami = var.ami
3      instance_type = var.instance_type
4
5      tags = {
6          Name = "My-EC2-Instance"
7      }
8  }
9
10 resource "aws_iam_user" "iam_users" {
11     count = length(var.iam_users)
12     name = var.iam_users[count.index]
13     tags = {
14         Name = "${var.iam_users[count.index]}-user"
15     }
16 }
```

3. Initialize & Apply

terraform apply



```
PS D:\DevOps\LAB\SPCM\TERRAFORM-SCRIPTS\TERRAFORM-IAM-USER> terraform
● apply -auto-approve

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"
+   }
}

+ 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
+ spot_instance_request_id = (known after apply)
+ subnet_id            = (known after apply)
+ tags                 = {
+   "Name" = "My-EC2-Instance"
+ }
+ tags_all             = {
+   "Name" = "My-EC2-Instance"
+ }
+ 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: 4 to add, 0 to change, 0 to destroy.
aws_iam_user.iam_users[1]: Creation complete after 1s [id=user2]
aws_iam_user.iam_users[0]: Creation complete after 1s [id=user1]
aws_iam_user.iam_users[2]: Creation complete after 1s [id=user3]
aws_instance.UPES: Creation complete after 1s [id=i-0378e8cdaa8cc8659]
aws_instance.UPES: Still creating... [10s elapsed]
aws_instance.UPES: Still creating... [20s elapsed]
aws_instance.UPES: Still creating... [30s elapsed]
aws_instance.UPES: Creation complete after 32s [id=i-0378e8cdaa8cc8659]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
PS D:\DevOps\LAB\SPCM\TERRAFORM-SCRIPTS\TERRAFORM-IAM-USER>
```

3. Verify Users in AWS Console

Services

Search

[Alt+S]

Dashboard

Global View

Instances

EC2

EC2 Instance Types

1 Templates

Requests

15 Plans

Instances (1) Info

Refresh

Connect

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
	My-EC2-Instance	i-0378e8cd8aa8cc8659	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a

Users (3) Info

Refresh

Delete

Create user

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

Search

< 1 >

Settings

	User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in
	user1	/				-	-
	user2	/				-	-
	user3	/				-	-