

Lab Exercise 7– Creating Multiple IAM Users in Terraform

Objective:

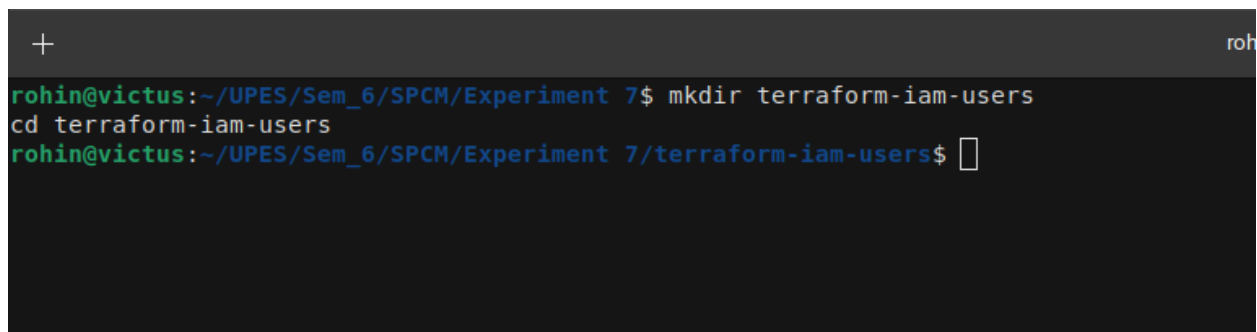
Learn how to use Terraform to create multiple IAM users with unique settings.

Prerequisites:

- Terraform installed on your machine.
- AWS CLI configured with the necessary credentials.

Steps:

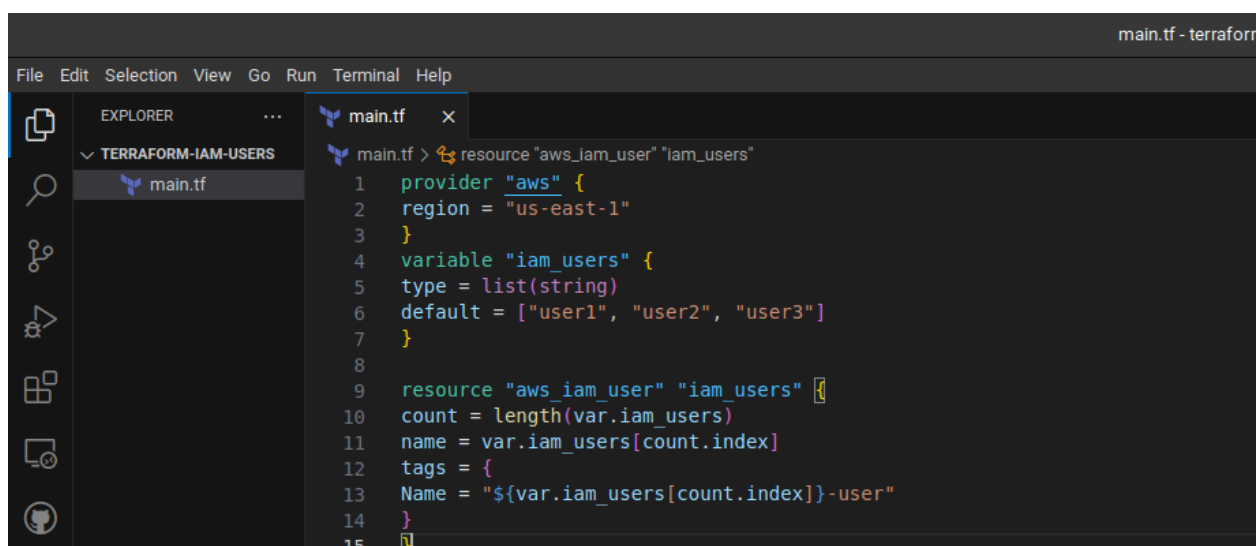
1. Create a Terraform Directory:

A terminal window with a dark background. The prompt is 'rohin@victus:~/UPES/Sem_6/SPCM/Experiment 7\$'. The user enters 'mkdir terraform-iam-users' and then 'cd terraform-iam-users'. The prompt changes to 'rohin@victus:~/UPES/Sem_6/SPCM/Experiment 7/terraform-iam-users\$' with a cursor at the end.

```
rohin@victus:~/UPES/Sem_6/SPCM/Experiment 7$ mkdir terraform-iam-users
rohin@victus:~/UPES/Sem_6/SPCM/Experiment 7/terraform-iam-users$
```

- Create Terraform Configuration Files:
- Create a file named main.tf:

main.tf

A screenshot of a code editor (VS Code) with a dark theme. The Explorer sidebar on the left shows a folder named 'TERRAFORM-IAM-USERS' containing a file 'main.tf'. The main editor area shows the content of 'main.tf'. The code defines an AWS provider, a list of user names, and a loop to create IAM users.

```
main.tf
1 provider "aws" {
2   region = "us-east-1"
3 }
4 variable "iam_users" {
5   type = list(string)
6   default = ["user1", "user2", "user3"]
7 }
8
9 resource "aws_iam_user" "iam_users"
10 count = length(var.iam_users)
11 name = var.iam_users[count.index]
12 tags = {
13   Name = "${var.iam_users[count.index]}-user"
14 }
15
```

In this configuration, we define a list variable `iam_users` containing the names of the IAM users we want to create. The `aws_iam_user` resource is then used in a loop to create users based on the values in the list.

2. Initialize and Apply:

Run the following Terraform commands to initialize and apply the configuration:

```
+ rohin@victus: ~/UPES/Sem_6/SPCM/Experiment 7/terraform-iam-users
rohin@victus:~/UPES/Sem_6/SPCM/Experiment 7/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" {
```

Terraform will prompt you to confirm the creation of IAM users. Type yes and press Enter.

3. Verify Users in AWS Console:

- Log in to the AWS Management Console and navigate to the IAM service.
- Verify that the IAM users with the specified names and tags have been created.

4. Update IAM Users:

- If you want to add or remove IAM users, modify the `iam_users` list in the `main.tf` File.
- Rerun the `terraform apply` command to apply the changes:

5. Clean Up:

- After testing, you can clean up the IAM users:
`terraform destroy`
- Confirm the destruction by typing yes.

6. Conclusion:

This lab exercise demonstrates how to create multiple IAM users in AWS using Terraform. The use of variables and loops allows you to easily manage and scale the creation of IAM users. Experiment with different user names and settings in the main.tf file to understand how Terraform provisions resources based on your configuration.