



System Provisioning and Configuration Management

Submitted by:

NAME: Tanishqua Tellakula

SAP ID: 500084269

Roll number: R120220007

Batch: B1(Hons)

EXPERIMENT 10

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:

```
mkdir terraform-iam-users
```

```
cd terraform-iam-users
```

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

```
PS D:\Terraform-Demo\terraform-ec2-for-each> cd ..
PS D:\Terraform-Demo> mkdir terraform-iam-users

Directory: D:\Terraform-Demo
Mode                LastWriteTime         Length Name
----                -

```

```
PS D:\Terraform-Demo> cd terraform-iam-users
PS D:\Terraform-Demo\terraform-iam-users> |
```

iam.tf

```
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]}"
```

```
}
```

```
}
```

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.

```
terraform-iam-users > cat .tf 2 ...
1  variable "iam_users" {
2    type    = list(string)
3    default = ["user1", "user2", "user3"]
4  }
5
6  resource "aws_iam_user" "iam_users" {
7    count = length(var.iam_users)
8    name = var.iam_users[count.index]
9
10   tags = {
11     Name = "${var.iam_users[count.index]}"
12   }
13 }
```

2. Initialize and Apply:

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

```
terraform init
```

```
terraform apply
```

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

```
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.

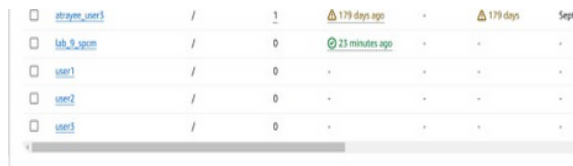
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS D:\Terraform-Demo\terraform-iam-users>
```

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

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
PS D:\Terraform-Demo\terraform-iam-users>
```

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.



Name	Path	Created	Access Key ID	Secret Access Key	Groups
atraym_user1	/	179 days ago			
lab_9_user	/	23 minutes ago			
user1	/				
user2	/				
user3	/				

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:

```
terraform apply
```

5. Clean Up:

- After testing, you can clean up the IAM users:

```
terraform destroy
```

- Confirm the destruction by typing yes.

```
Enter a value: yes
aws_iam_user.iam_users[1]: Destroying... [id=user2]
aws_iam_user.iam_users[0]: Destroying... [id=user1]
aws_iam_user.iam_users[2]: Destroying... [id=user3]
aws_iam_user.iam_users[0]: Destruction complete after 1s
aws_iam_user.iam_users[1]: Destruction complete after 1s
aws_iam_user.iam_users[2]: Destruction complete after 1s
Destroy complete! Resources: 3 destroyed.
PS D:\Terraform-Demo\terraform-iam-users>
```

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.