

Lab Exercise 10– Creating Mu Terraform

Objective:

Learn how to use Terraform to create multiple IAM

Prerequisites:

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

Steps:

1. Create a Terraform Directory:

mkdir terraform-iam-users

cd terraform-iam-users

```
(base) → terraform-ec2-for-each-lab9 git:
(base) → MyLab git:(main) ✗ mkdir terra
(base) → MyLab git:(main) ✗ cd terrafor
(base) → terraform-iam-users-lab10 git:
(base) → terraform-iam-users-lab10 git:
```

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

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]}"  
  }  
}
```

```

se) » terraform-iam-users-lab1
variable iam_users
variable "iam_users"
    type = [Ujjwaluser1]
    default ["Ujjwaluse1, 'Ujjwalu

resource "aws_iam_user" "iam_user"
    count = length(var.iam_user)
    {
        tags = "$var.iam_users(count
    }

```

configuration, we define a list variable iam, AM users we want to create. The aws_iam, create users based on the values in the list.

2. Initialize and Apply:

Run the following Terraform commands to initialize and

```
terraform init
```

```
terraform apply
```

```
(base) → terraform-iam-users-lab10 git:(main) × terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.68.0"...
- Installing hashicorp/aws v5.68.0...
- Installed hashicorp/aws v5.68.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the
selections it made above. Include this file in your version control
so that Terraform can guarantee to make the same selections if you
you run "terraform init" in the future.
```

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running `terraform apply` to make any changes that are required for your infrastructure. In future runs, these changes 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 want to automatically update the backend configuration to use the Terraform CLI, the commands will detect it and remind you to do so if necessary.

```
(base) → terraform-iam-users-lab10 git:(main) × terraform init
```

Terraform will prompt you to confirm the creation of IAM users. Press **Enter**.

3. Verify Users in AWS Console:

- Log in to the AWS Management Console and navigate to IAM > Users.
- Verify that the IAM users with the specified names are created.

4. Update IAM Users:

- If you want to add or remove IAM users, modify the `users.tf` file.
- Rerun the `terraform apply` command to apply the changes.


```
terraform apply
aws_iam_users0: Creating...
aws_iam_users1: Creating...
aws_iam_users2: Creation complete
aws_ujjwaluser1: Creation complete
aws_ujjwaluser3
Apply complete! Resources: 3 added
(base) > terraform-lam-users-lab10
```

Users (3)

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

<input type="checkbox"/>	User name ▾	Group ▾	Last activity ▾	MFA ▾
<input type="checkbox"/>	ujjwaluser1	/	-	/
<input type="checkbox"/>	ujjwaluser2	/	-	/

6. Conclusion:

This lab exercise demonstrates how to create multiple IAM users with Terraform. The use of variables and loops allows you to create multiple IAM users. Experiment with different use cases by modifying the main.tf file to understand how Terraform provision configuration.