

Lab Exercise 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:

```
LAB 10 > terraform-iam-users > main.tf > ...
1  terraform {
2
3  required_providers {
4
5    aws = {
6
7    source = "hashicorp/aws"
8
9    version = "5.68.0"
10
11  }
12
13  }
14
15  }
16
17
18
19  provider "aws" {
20
21    access_key = "AKIAQA3ENQ3PX24PN704"
22
23    secret_key = "2nMqr4ceDp9smT9D4aA5XkSLlynysIbAl33rD4f"
24
25    region = "ap-south-1"
26
27  }
```

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]}"  
  }  
}
```

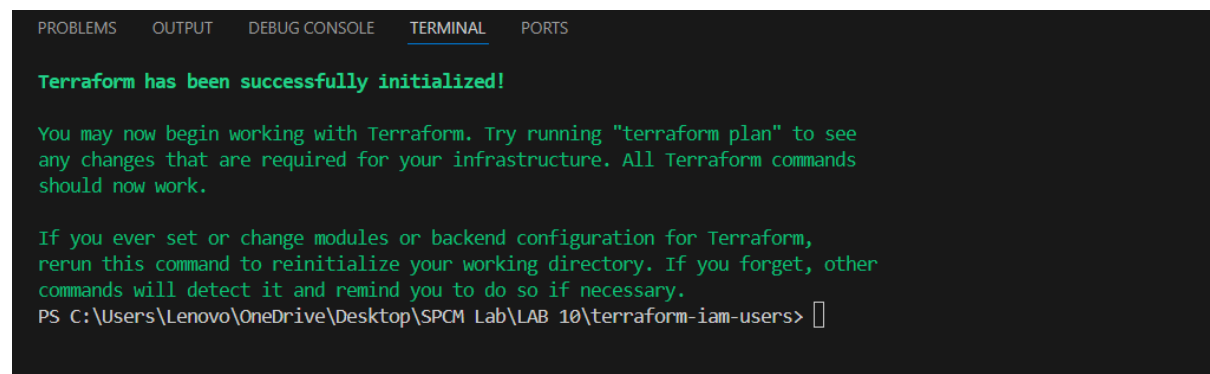
```
LAB 10 > terraform-iam-users > iam.tf > ...  
1  variable "iam_users" {  
2  
3    type    = list(string)  
4  
5    default = ["user1", "user2", "user3"]  
6  
7  }  
8  
9  
10  
11 resource "aws_iam_user" "iam_users" {  
12  
13   count = length(var.iam_users)  
14  
15   name = var.iam_users[count.index]  
16  
17  
18  
19   tags = {  
20  
21     Name = "${var.iam_users[count.index]}"  
22  
23   }  
24  
25 }
```

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:

```
terraform init
terraform apply
```

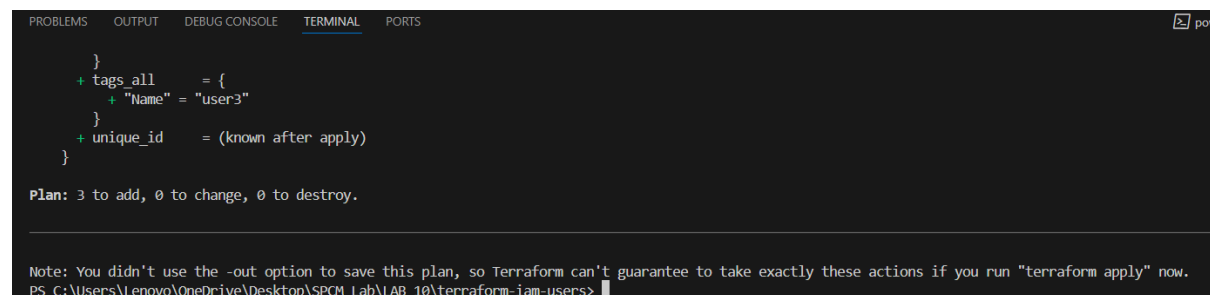


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

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 C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 10\terraform-iam-users> 
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  [x] pow

    }
+ tags_all      = {
+   + "Name" = "user3"
    }
+ unique_id     = (known after apply)
}

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
PS C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 10\terraform-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:

terraform apply

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Only 'yes' will be accepted to approve.

Enter a value: yes

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

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
PS C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 10\terraform-iam-users>
```

5. Clean Up:

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

terraform destroy

- Confirm the destruction by typing yes.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

There is no undo. Only 'yes' will be accepted to confirm.

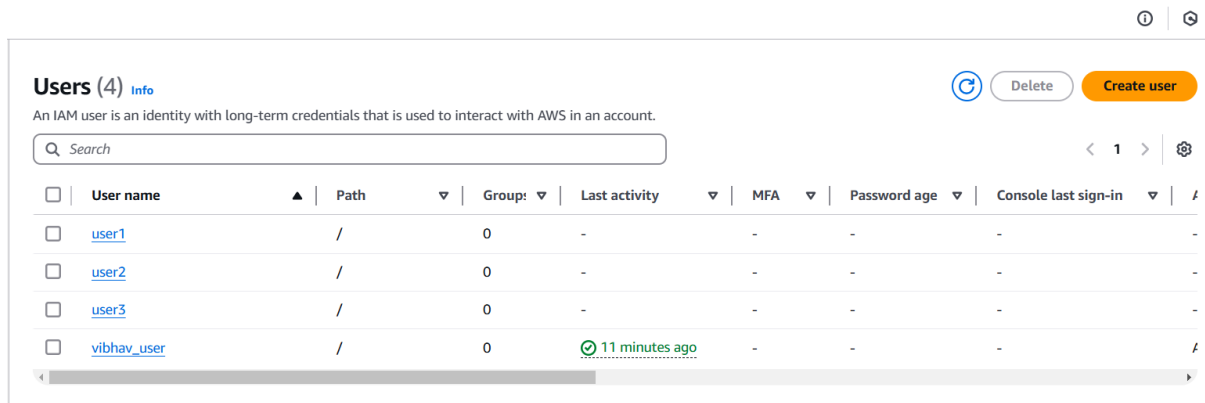
Enter a value: yes

aws_iam_user.iam_users[2]: Destroying... [id=user3]
aws_iam_user.iam_users[0]: Destroying... [id=user1]
aws_iam_user.iam_users[1]: Destroying... [id=user2]
aws_iam_user.iam_users[0]: Destruction complete after 2s
aws_iam_user.iam_users[1]: Destruction complete after 2s
aws_iam_user.iam_users[2]: Destruction complete after 2s

Destroy complete! Resources: 3 destroyed.
PS C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 10\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.



Users (4) [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.

| <input type="checkbox"/> | User name | Path | Group | Last activity | MFA | Password age | Console last sign-in | |
|--------------------------|-----------------------------|------|-------|-----------------------------|-----|--------------|----------------------|---|
| <input type="checkbox"/> | user1 | / | 0 | - | - | - | - | - |
| <input type="checkbox"/> | user2 | / | 0 | - | - | - | - | - |
| <input type="checkbox"/> | user3 | / | 0 | - | - | - | - | - |
| <input type="checkbox"/> | vibhav_user | / | 0 | 11 minutes ago | - | - | - | / |