UPES

UNIVERSITY OF THE FUTURE

System Provisioning and Configuration Management Lab

Siddhant Singh

500095375

Batch - 3

Experiment 7

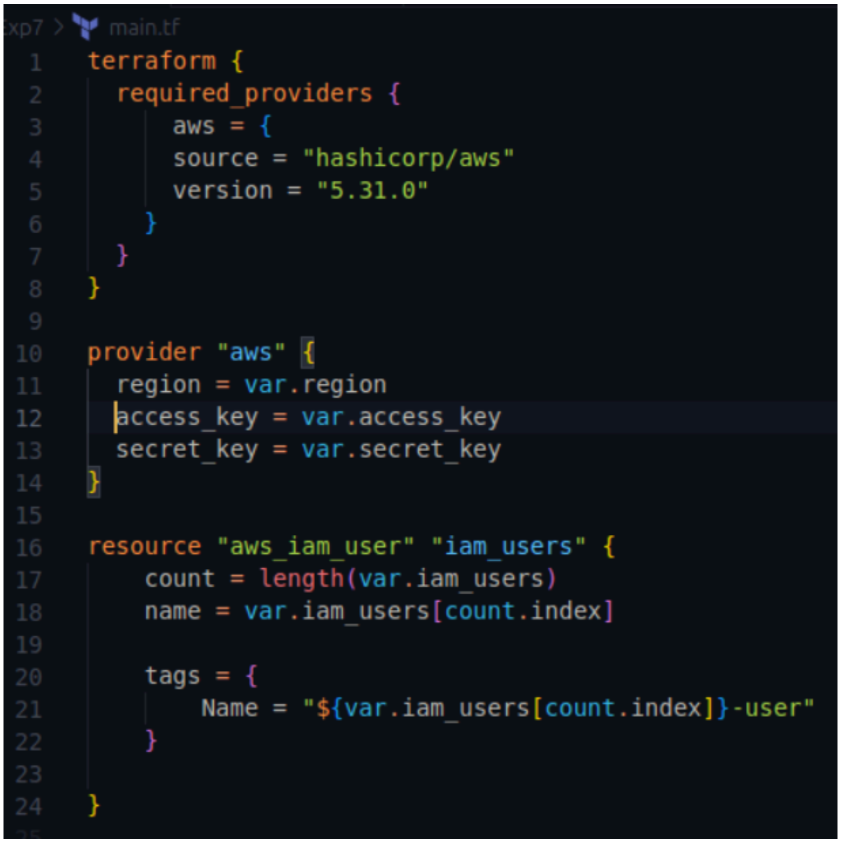
Creating Multiple IAM Users in Terraform

Aim

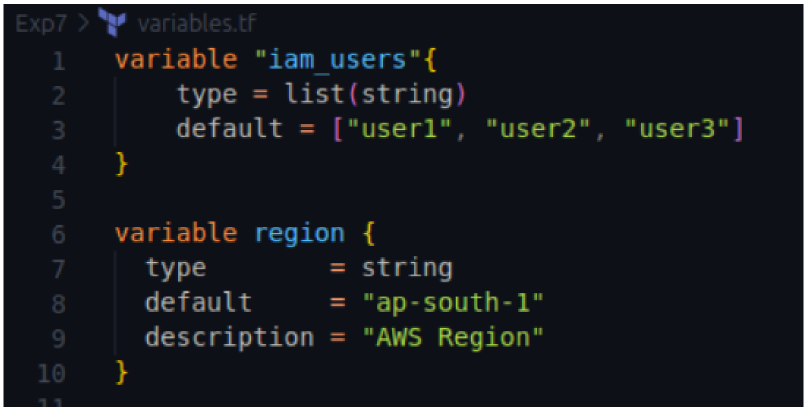
Learn how to use Terraform to create multiple IAM users with unique settings.

Steps

1. Create a main file & variables.tf file for EC2 Instance.

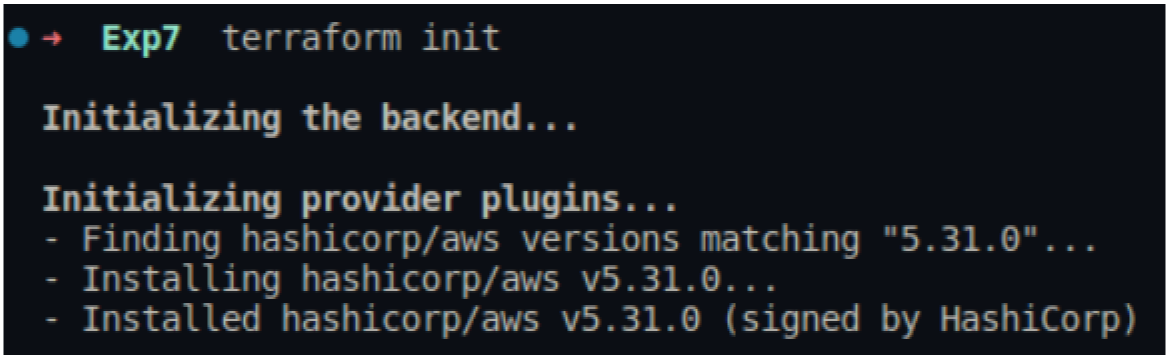


```
terraform {
  required_providers {
      aws = {
      source = "hashicorp/aws"
      version = "5.31.0"
    }
  }
}

provider "aws" {
  region = var.region
  access_key = var.access_key
  secret_key = var.secret_key
}

resource "aws_iam_user" "iam_users" {
    count = length(var.iam_users)
    name = var.iam_users[count.index]

    tags = {
        Name = "${var.iam_users[count.index]}-user"
    }

}
```

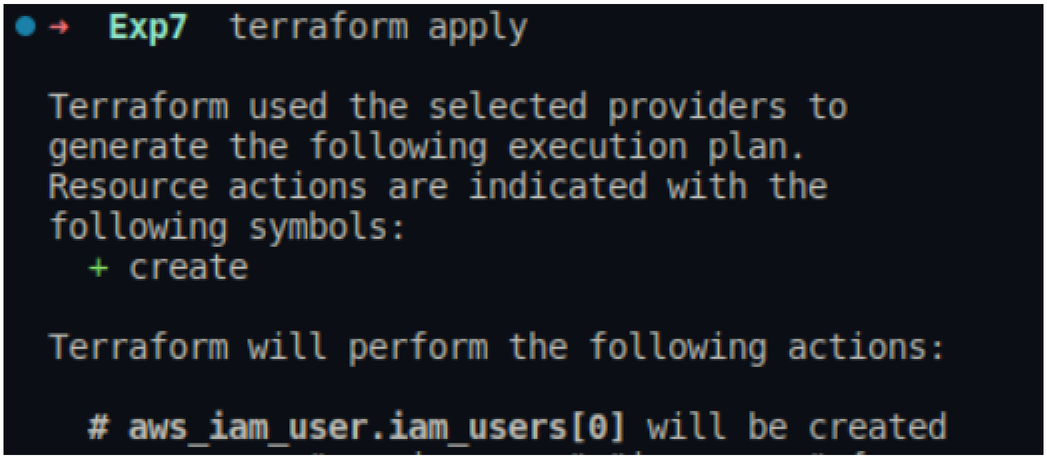


```
variable "iam_users"{
    type = list(string)
    default = ["user1", "user2", "user3"]
}

variable region {
  type        = string
  default     = "ap-south-1"
  description = "AWS Region"
}
```

2. Initialize & Apply the configuration

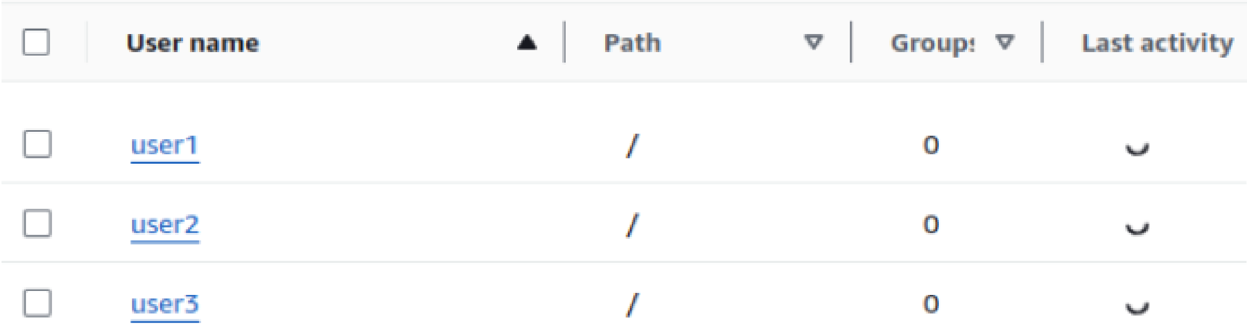


```
● ➜  Exp7  terraform init

  Initializing the backend...

  Initializing provider plugins...
  - Finding hashicorp/aws versions matching "5.31.0"...
  - Installing hashicorp/aws v5.31.0...
  - Installed hashicorp/aws v5.31.0 (signed by HashiCorp)
```



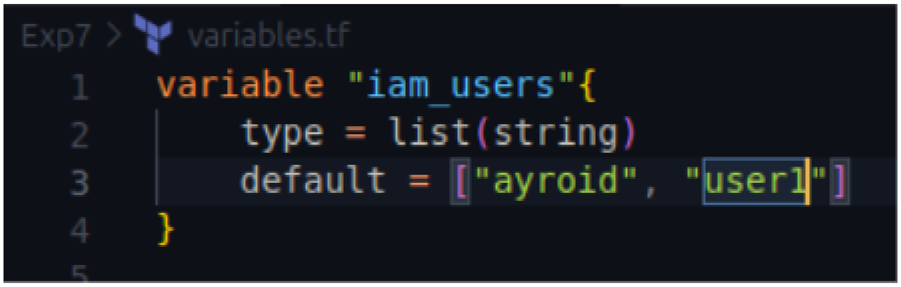
```
● ➜  Exp7  terraform apply

  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
```

3. Verify the IAM users on AWS

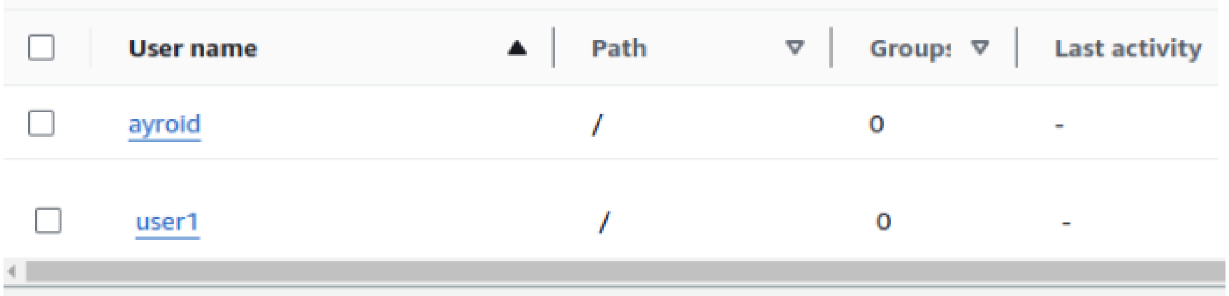


□	User name ▲	Path ▽	Group: ▽	Last activity
□	user1	/	0	◡
□	user2	/	0	◡
□	user3	/	0	◡

4. Update the list of users to update the count of IAM Users on AWS

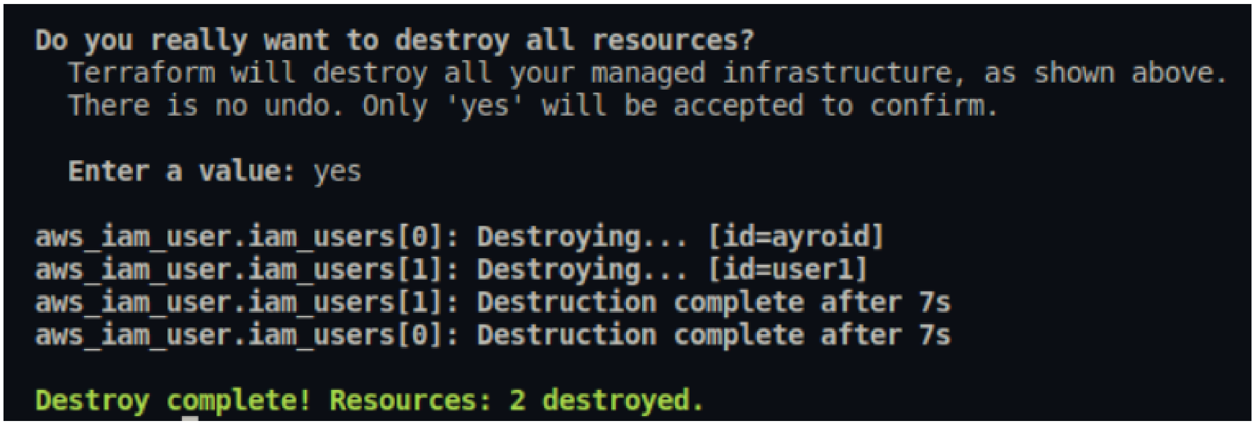


```
variable "iam_users"{
    type = list(string)
    default = ["ayroid", "user1"]
}
```



☐	User name ▲	Path ▽	Group: ▽	Last activity
☐	ayroid	/	0	-
☐	user1	/	0	-

5. Clean up resources



```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_iam_user.iam_users[0]: Destroying... [id=ayroid]
aws_iam_user.iam_users[1]: Destroying... [id=user1]
aws_iam_user.iam_users[1]: Destruction complete after 7s
aws_iam_user.iam_users[0]: Destruction complete after 7s

Destroy complete! Resources: 2 destroyed.
```