

Name - Vaibhav Navneet Jorvekar

- Create IAM user using terraform

- 1) Write code with vs code app in main.tf file



```
main.tf  X  profile.tf
C: > terraform > main.tf > ...
1  # Create the IAM User
2  resource "aws_iam_user" "this_iam" {
3      name = "vaibhav23"
4      path = "/"
5
6      tags = {
7          name = "vaibhav"
8      }
9  }
10
11 # Add the Access Key to User
12 resource "aws_iam_access_key" "this_iam_key" {
13     user = aws_iam_user.this_iam.name
14 }
15
16 # Create the Group
17 resource "aws_iam_group" "group-name" {
18     name = "jorvekar_Group"
19 }
20
21 # Add User & Group in Group Membership
22 resource "aws_iam_group_membership" "Group" {
23     name = "Terrafrom_Group-membership"
24     users = [
25         aws_iam_user.this_iam.name
26     ]
27     group = aws_iam_group.group-name.name
28 }
29
```

- 2) Write the access key and secret key in second file in profile.tf

```
main.tf  profile.tf  X
C: > terraform > profile.tf > provider "aws" > secret_key

1  provider "aws" {
2      region     = "us-west-1"
3      access_key = "AKIARVF2DVWRA76MY7BZ"
4      secret_key = "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"
5  }
```

- 3) Open terminal and initialize the file to command #terraform init

```
vaibhav@DESKTOP-P3NSBEM:~$ ls
vaibhav@DESKTOP-P3NSBEM:~$ cd /mnt
vaibhav@DESKTOP-P3NSBEM:/mnt$ ls
ls: cannot access 'DumpStack.log.tmp': Permission denied
ls: cannot access 'hiberfil.sys': Permission denied
ls: cannot access 'pagefile.sys': Permission denied
ls: cannot access 'swapfile.sys': Permission denied
vaibhav@DESKTOP-P3NSBEM:/mnt$ cd c
vaibhav@DESKTOP-P3NSBEM:/mnt/c$ ls
'AlwaysUse Icons'      'System Volume Information'
'$WINRE_BACKUP_PARTITION.MARKER'  'Program Files'
'Documents and Settings'  'Program Files (x86)'
DumpStack.log           'System Volume Information'
DumpStack.log.tmp       'Users'
'hiberfil.sys'          'Windows'
'pagefile.sys'          'cat-animals-bokeh-cute-wallpaper-preview.jpg'
'swapfile.sys'          'terraform_1.7.4_windows_386.zip'
vaibhav@DESKTOP-P3NSBEM:/mnt/c$ cd terraform
vaibhav@DESKTOP-P3NSBEM:/mnt/c/terraform$ ls
main.tf  profile.tf  terraform.tfstate
vaibhav@DESKTOP-P3NSBEM:/mnt/c/terraform$ tf apply -auto-approve
Command 'tf' not found, but can be installed with:
sudo apt install tf # version 1:4.0sl-21.1, or
sudo apt install tf5 # version 5.0beta8-10build1
vaibhav@DESKTOP-P3NSBEM:/mnt/c/terraform$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.38.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
```

#### 4) Apply the file in command #terraform apply

```
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.
vaibhav@DESKTOP-P3NSBEM:/mnt/c/terraform$ terraform apply -auto-approve
aws_iam_user.this_iam: Refreshing state... [id=vaibhav23]
aws_iam_group.group-name: Refreshing state... [id=jorvekar_Group]
aws_iam_group.membership.Group: Refreshing state... [id=Terrafrom_Group-membership]
aws_iam_access_key.this_iam_key: Refreshing state... [id=AKIARVF2DVWRJAG7YPVK]
```

**Note:** Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected this plan:

```
# aws_iam_user.this_iam has been deleted
- resource "aws_iam_user" "this_iam" {
  id      = "vaibhav23"
  - name   = "vaibhav23" -> null
  tags    = {
    "name" = "vaibhav"
  }
  # (5 unchanged attributes hidden)
}
```

Unless you have made equivalent changes to your configuration, or ignored the relevant attributes using `ignore_changes`, the following plan may include actions to undo or respond to these changes.

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

```
+ create
~ update in-place
```

Terraform will perform the following actions:

```
~ resource "aws_iam_group_membership" "Group" {
  id      = "Terrafrom_Group-membership"
  name    = "Terrafrom_Group-membership"
  ~ users = [
    + "vaibhav23",
  ]
  # (1 unchanged attribute hidden)
}

# aws_iam_user.this_iam will be created
+ resource "aws_iam_user" "this_iam" {
  + arn              = (known after apply)
  + force_destroy    = false
  + id               = (known after apply)
  + name             = "vaibhav23"
  + path             = "/"
  + tags             = {
    + "name" = "vaibhav"
  }
  + tags_all         = {
    + "name" = "vaibhav"
  }
  + unique_id        = (known after apply)
}
```

Plan: 2 to add, 1 to change, 0 to destroy.

aws\_iam\_user.this\_iam: Creating...

aws\_iam\_user.this\_iam: Creation complete after 2s [id=vaibhav23]

aws\_iam\_access\_key.this\_iam\_key: Creating...

aws\_iam\_group\_membership.Group: Modifying... [id=Terrafrom\_Group-membership]

aws\_iam\_access\_key.this\_iam\_key: Creation complete after 0s [id=AKIARVF2DVWRNG32QCHD]

aws\_iam\_group\_membership.Group: Modifications complete after 1s [id=Terrafrom\_Group-membership]

Apply complete! Resources: 2 added, 1 changed, 0 destroyed.

vaibhav@DESKTOP-P3NSBEM:/mnt/c/terraform\$ |