

Lab Exercise 7– Creating Multiple IAM Users in Terraform

```
main.tf U X
main.tf > resource "aws_iam_user" "iam_users"
1 provider "aws" {
2   access_key = "AKIAI7C83847ZP7WBW5Q"
3   secret_key = "4wkyZm80YHh11111,cXx0EeqRzjay62NwYlDJHzrdF"
4   region = "us-east-1"
5 }
6 variable "iam_users" {
7   type = list(string)
8   default = ["user1", "user2", "user3"]
9 }
10 resource "aws_iam_user" "iam_users" {
11   count = length(var.iam_users)
12   name = var.iam_users[count.index]
13   tags = {
14     Name = "${var.iam_users[count.index]}-user"
15   }
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Initializing provider plugins...

- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.35.0...
- Installed hashicorp/aws v5.35.0 (signed by HashiCorp)

Terraform has created a lock file `.terraform.lock.hcl` to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when 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.

```
PS E:\SPCM Lab\terraform-iam-users> 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
+ resource "aws_iam_user" "iam_users" {
  + access_key = (known after apply)
  + secret_key = (known after apply)
  + region = (known after apply)
  + name = (known after apply)
  + tags = {
    Name = (known after apply)
  }
}
```

```
aws_iam_user.iam_users[2]: Creating...
aws_iam_user.iam_users[0]: Creating...
aws_iam_user.iam_users[1]: Creating...
aws_iam_user.iam_users[1]: Creation complete after 3s [id=user2]
aws_iam_user.iam_users[2]: Creation complete after 3s [id=user3]
aws_iam_user.iam_users[0]: Creation complete after 3s [id=user1]
```

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

PS E:\SPCM Lab\terraform-iam-users>

[IAM](#) > Users

Users (3) [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.

< 1 > ⚙

<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		-	-	-

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

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

Destroy complete! Resources: 3 destroyed.

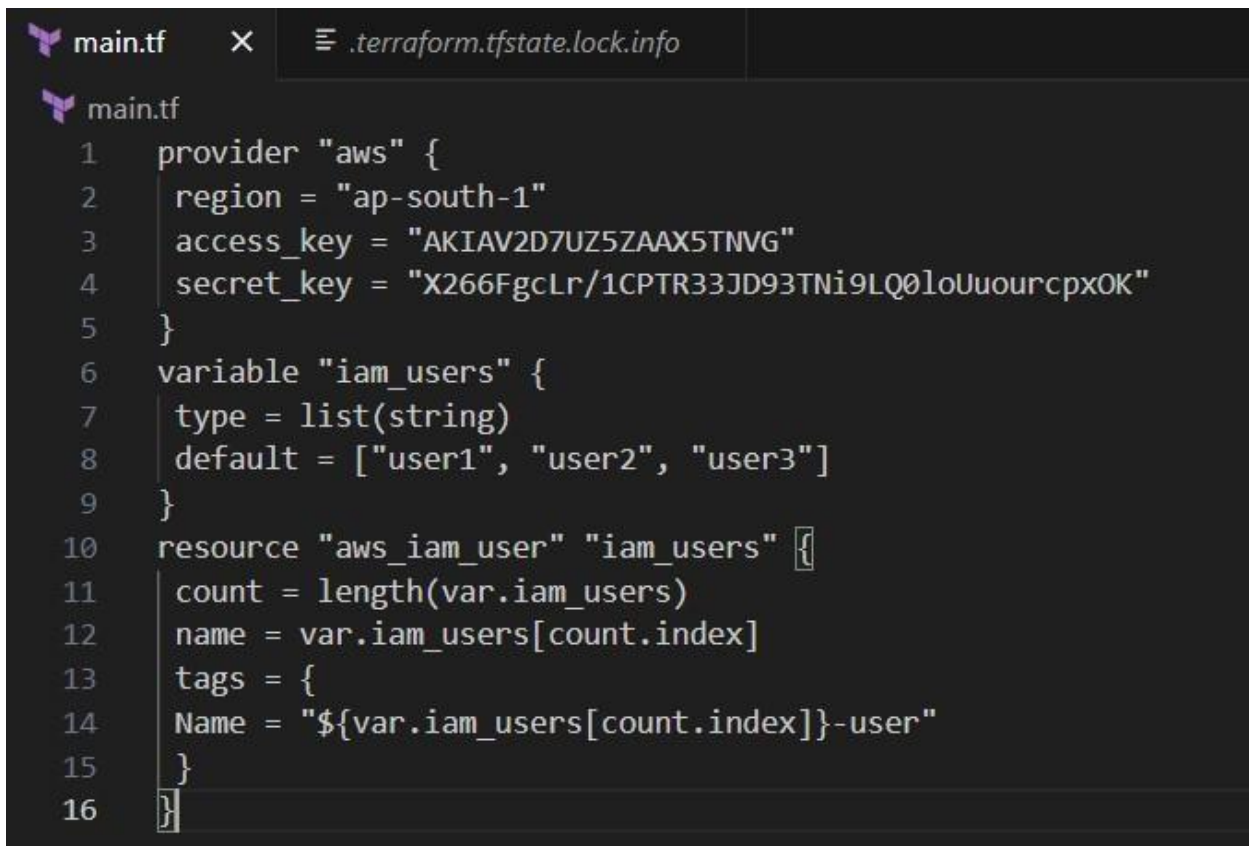
PS E:\SPCM Lab\terraform-iam-users>

Lab-8: Creating a VPC in Terraform

Step 1: Create a Terraform Directory

```
C:\Users\hp>mkdir terraform-vpc
```

```
C:\Users\hp>cd terraform-vpc
```



```
main.tf x .terraform.tfstate.lock.info
main.tf
1 provider "aws" {
2     region = "ap-south-1"
3     access_key = "AKIAV2D7UZ5ZAAX5TNVG"
4     secret_key = "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"
5 }
6 variable "iam_users" {
7     type = list(string)
8     default = ["user1", "user2", "user3"]
9 }
10 resource "aws_iam_user" "iam_users" {
11     count = length(var.iam_users)
12     name = var.iam_users[count.index]
13     tags = {
14         Name = "${var.iam_users[count.index]}-user"
15     }
16 }
```



```

+ owner_id                = (known after apply)
+ tags                    = {
  + "Name" = "MyVPC"
}
+ tags_all                = {
  + "Name" = "MyVPC"
}
}

```

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

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.my_vpc: Creating...

aws_vpc.my_vpc: Still creating... [10s elapsed]

aws_vpc.my_vpc: Creation complete after 18s [id=vpc-0ccb810e9b0816eb0]

aws_subnet.my_subnet[0]: Creating...

aws_subnet.my_subnet[1]: Creating...

aws_subnet.my_subnet[0]: Still creating... [10s elapsed]

aws_subnet.my_subnet[1]: Still creating... [10s elapsed]

aws_subnet.my_subnet[0]: Creation complete after 15s [id=subnet-0d23ee873bde8cbec]

aws_subnet.my_subnet[1]: Creation complete after 15s [id=subnet-0159de7521427b1d6]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Step 3: Verify the Resources

Your VPCs (1) Info							
<input type="text" value="Search"/> Actions Create VPC							
<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o	
<input type="checkbox"/>	-	vpc-00f6a10d530fa5fa6	Available	172.31.0.0/16	-	dopt-0c	

Step 4: Clean up

```
C:\Users\hnp\terrafrom-vpc>terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-0ccb810e9b0816eb0]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-0159de7521427b1d6]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-0d23ee873bde8cbec]

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

Terraform will perform the following actions:

  # aws_subnet.my_subnet[0] will be destroyed
  - resource "aws_subnet" "my_subnet" {
    - arn                                = "arn:aws:ec2:us-east-1:399699660658:subnet/subnet-0d23ee873bde8
cbec" -> null
    - assign_ipv6_address_on_creation    = false -> null
    - availability_zone                  = "us-east-1a" -> null
    - availability_zone_id                = "use1-az2" -> null
    - cidr_block                          = "10.0.1.0/24" -> null
    - enable_dns64                        = false -> null
    - enable_lni_at_device_index          = 0 -> null
    - enable_resource_name_dns_a_record_on_launch = false -> null
    - enable_resource_name_dns_aaaa_record_on_launch = false -> null
    - id                                  = "subnet-0d23ee873bde8cbec" -> null
    - ipv6_native                         = false -> null
    - map_customer_owned_ip_on_launch     = false -> null
    - map_public_ip_on_launch             = true -> null
    - owner_id                            = "399699660658" -> null
    - private_dns_hostname_type_on_launch = "ip-name" -> null
    - tags                                = {
  }
```

```
    - instance_tenancy                  = "default" -> null
    - ipv6_netmask_length                 = 0 -> null
    - main_route_table_id                 = "rtb-0a25ebb03b7bf05c3" -> null
    - owner_id                            = "399699660658" -> null
    - tags                                = {
      - "Name" = "MyVPC"
    } -> null
    - tags_all                            = {
      - "Name" = "MyVPC"
    } -> null
  }
```

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

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

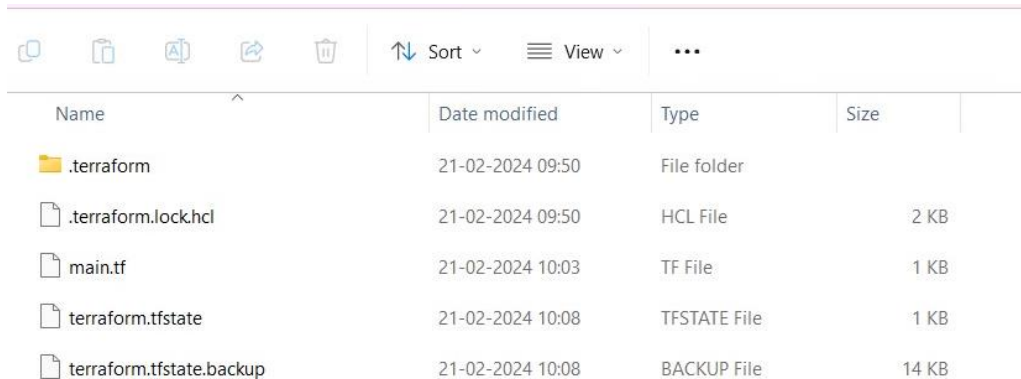
```
aws_subnet.my_subnet[1]: Destroying... [id=subnet-0159de7521427b1d6]
aws_subnet.my_subnet[0]: Destroying... [id=subnet-0d23ee873bde8cbec]
aws_subnet.my_subnet[0]: Destruction complete after 2s
aws_subnet.my_subnet[1]: Destruction complete after 3s
aws_vpc.my_vpc: Destroying... [id=vpc-0ccb810e9b0816eb0]
aws_vpc.my_vpc: Destruction complete after 3s
```

Destroy complete! Resources: 3 destroyed.

Lab Exercise 9– Creating Multiple EC2 Instances with for each in Terraform

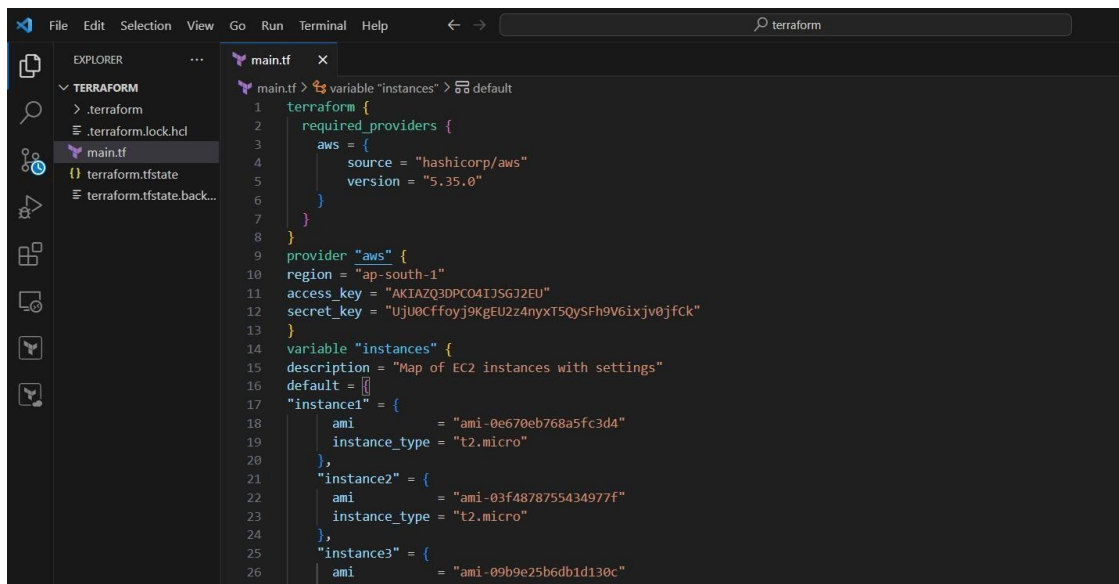
1. Create a Terraform Directory:

- Create Terraform Configuration Files:

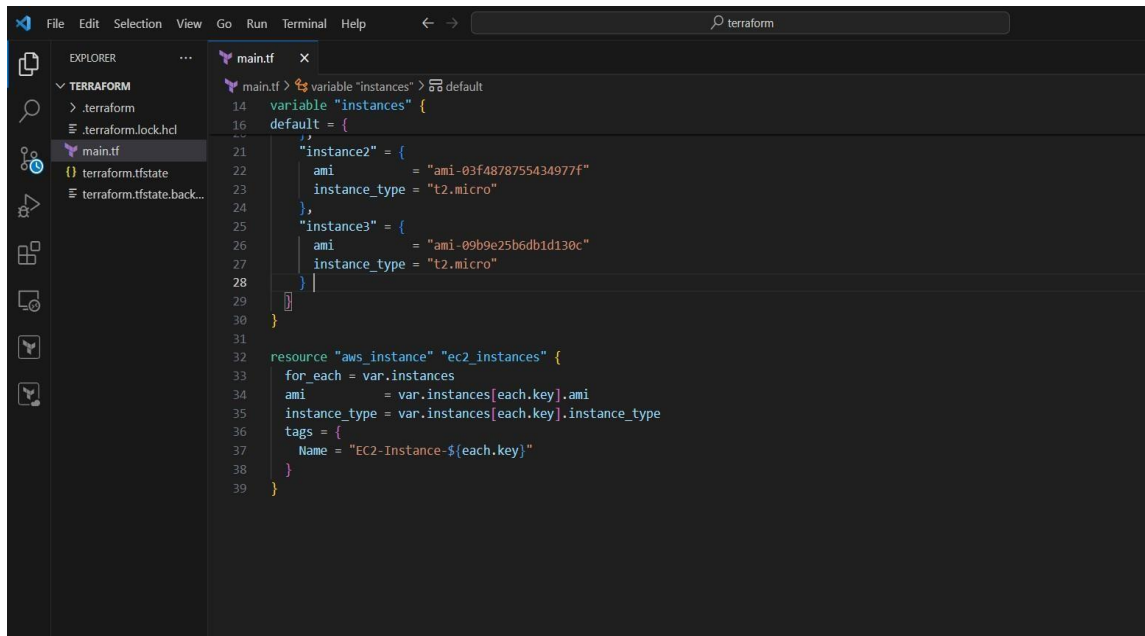


Name	Date modified	Type	Size
.terraform	21-02-2024 09:50	File folder	
.terraform.lock.hcl	21-02-2024 09:50	HCL File	2 KB
main.tf	21-02-2024 10:03	TF File	1 KB
terraform.tfstate	21-02-2024 10:08	TFSTATE File	1 KB
terraform.tfstate.backup	21-02-2024 10:08	BACKUP File	14 KB

- Create a file named main.tf:

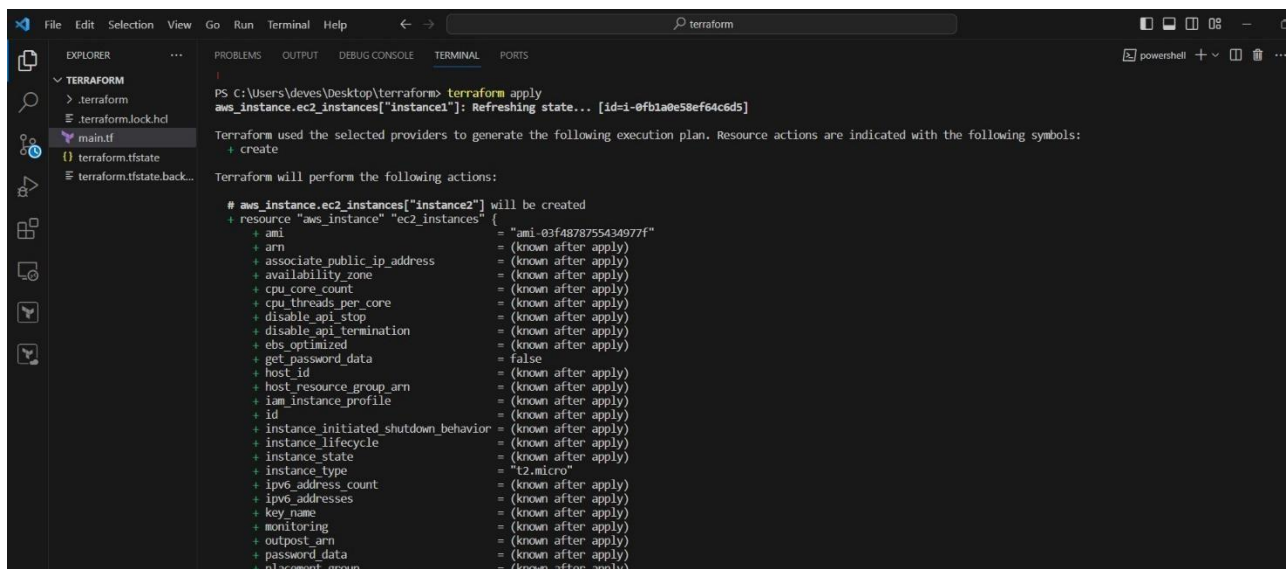


```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.35.0"
6     }
7   }
8 }
9 provider "aws" {
10  region = "ap-south-1"
11  access_key = "AKIAZQ3DPC04IJS6J2EU"
12  secret_key = "UjU0Cffoyj9KgEU2z4nyXT5QysFh9V6ixjv0jfcK"
13 }
14 variable "instances" {
15  description = "Map of EC2 instances with settings"
16  default = {
17    "instance1" = {
18      ami = "ami-0e670eb768a5fc3d4"
19      instance_type = "t2.micro"
20    },
21    "instance2" = {
22      ami = "ami-03f4878755434977f"
23      instance_type = "t2.micro"
24    },
25    "instance3" = {
26      ami = "ami-09b9e25b6db1d130c"
```

```
14 variable "instances" {
15     default = {
16         "instance2" = {
17             ami           = "ami-03f4878755434977f"
18             instance_type = "t2.micro"
19         },
20         "instance3" = {
21             ami           = "ami-09b9e25b6db1d130c"
22             instance_type = "t2.micro"
23         }
24     }
25 }
26
27 resource "aws_instance" "ec2_instances" {
28     for_each = var.instances
29     ami      = var.instances[each.key].ami
30     instance_type = var.instances[each.key].instance_type
31     tags = {
32         Name = "EC2-Instance-${each.key}"
33     }
34 }
```

2. Initialize and Apply:

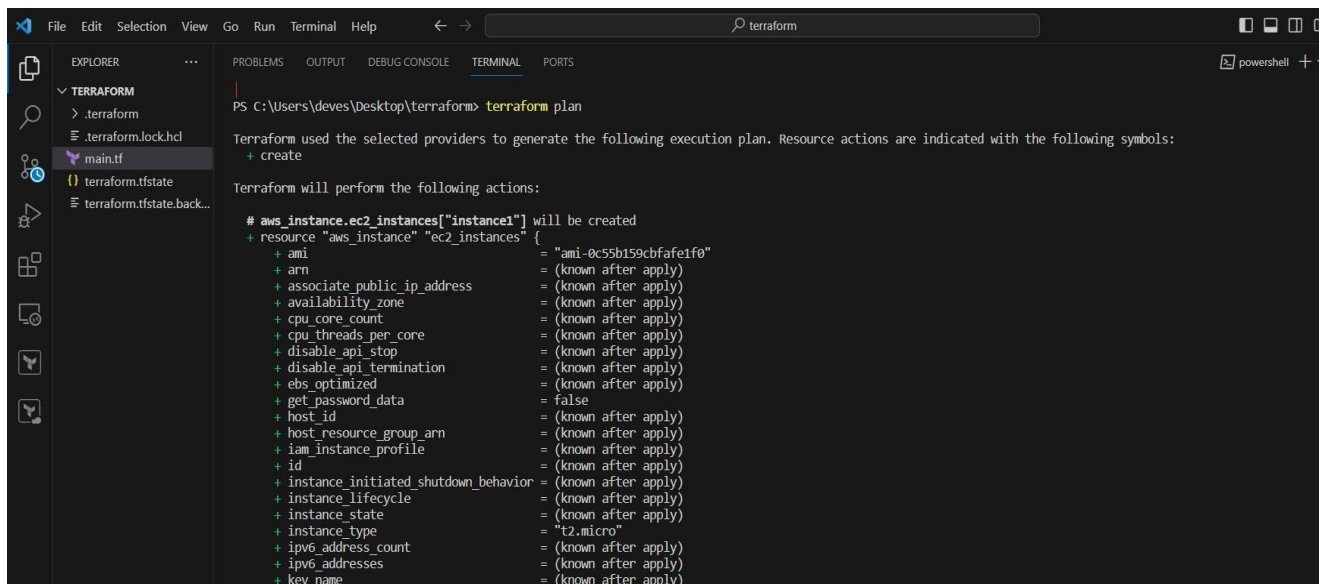


```
PS C:\Users\deves\Desktop\terraform> terraform apply
aws_instance.ec2_instances["instance1"]: Refreshing state... [id=i-0fb1a0e58ef64c6d5]

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_instance.ec2_instances["instance2"] will be created
+ resource "aws_instance" "ec2_instances" {
  + ami                  = "ami-03f4878755434977f"
  + arn                  = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone     = (known after apply)
  + cpu_core_count        = (known after apply)
  + cpu_threads_per_core  = (known after apply)
  + disable_api_stop      = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized         = (known after apply)
  + get_password_data     = false
  + host_id               = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile  = (known after apply)
  + id                    = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle    = (known after apply)
  + instance_state        = (known after apply)
  + instance_type         = "t2.micro"
  + ipv6_address_count    = (known after apply)
  + ipv6_addresses        = (known after apply)
  + key_name              = (known after apply)
  + monitoring            = (known after apply)
  + outpost_arn           = (known after apply)
  + password_data         = (known after apply)
  + placement_group       = (known after apply)
}
```

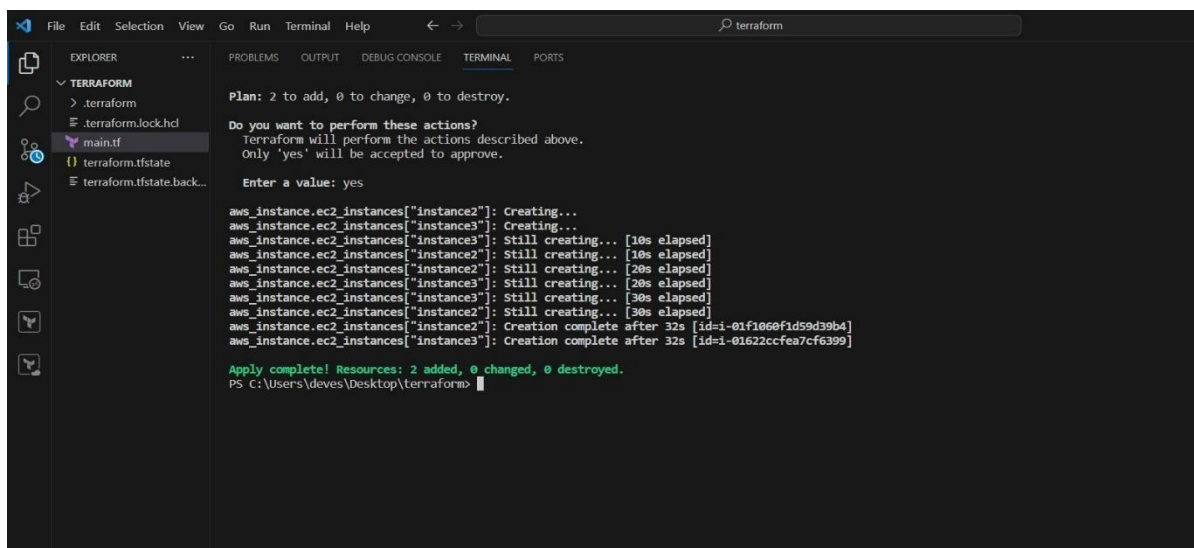
The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying a project named 'TERRAFORM' containing files like '.terraform', 'terraform.lock.hcl', 'main.tf', 'terraform.tfstate', and 'terraform.tfstate.back...'. The Terminal pane on the right shows the output of the 'terraform plan' command. The output indicates that Terraform will create an AWS EC2 instance named 'instance1' with various configuration parameters. The terminal text is as follows:

```
PS C:\Users\deves\Desktop\terraform> terraform plan

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_instance.ec2_instances["instance1"] will be created
+ resource "aws_instance" "ec2_instances" {
  + ami              = "ami-0c55b159cbfafe1f0"
  + arn              = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone = (known after apply)
  + cpu_core_count   = (known after apply)
  + cpu_threads_per_core = (known after apply)
  + disable_api_stop  = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized     = (known after apply)
  + get_password_data = false
  + host_id           = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile = (known after apply)
  + id                = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle = (known after apply)
  + instance_state     = (known after apply)
  + instance_type       = "t2.micro"
  + ipv6_address_count  = (known after apply)
  + ipv6_addresses      = (known after apply)
  + key_name            = (known after apply)
```



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying the same project. The Terminal pane on the right shows the output of the 'terraform apply' command. The output indicates that Terraform will create two AWS EC2 instances named 'instance2' and 'instance3'. The terminal text is as follows:

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

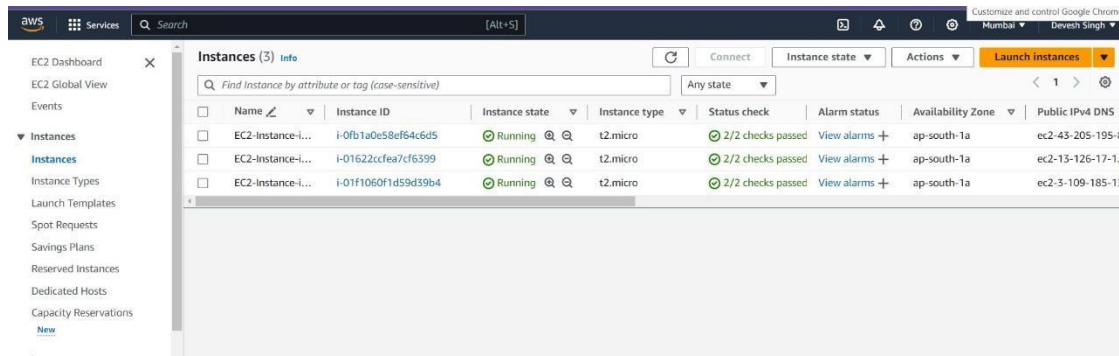
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.ec2_instances["instance2"]: Creating...
aws_instance.ec2_instances["instance3"]: Creating...
aws_instance.ec2_instances["instance3"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance2"]: Still creating... [10s elapsed]
aws_instance.ec2_instances["instance2"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance3"]: Still creating... [20s elapsed]
aws_instance.ec2_instances["instance3"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance2"]: Still creating... [30s elapsed]
aws_instance.ec2_instances["instance2"]: Creation complete after 32s [id=i-01f1060f1d59d39b4]
aws_instance.ec2_instances["instance3"]: Creation complete after 32s [id=i-01622ccfea7cf6399]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\deves\Desktop\terraform>
```

4. Update Instance Configuration:



5. Clean Up:

