



System Provisioning & Configuration Management

ASSIGNMENT-1

SUBMITTED BY:

NAME: Dhruv Jain

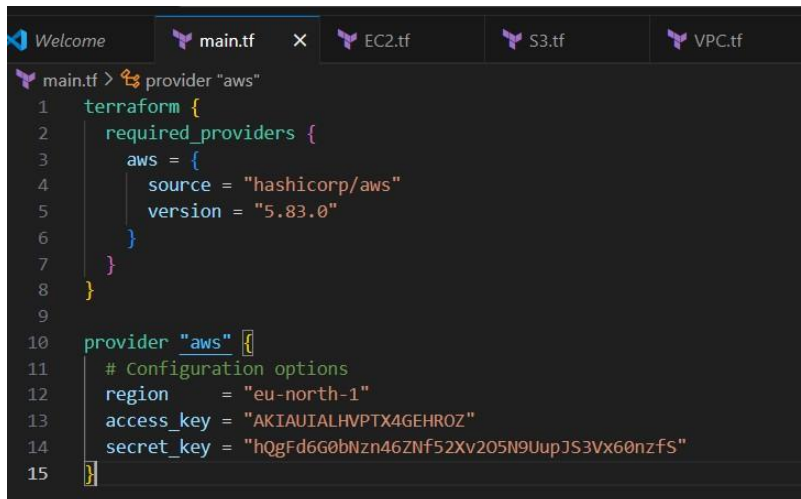
SAP ID: 500101737

BATCH: B1(NH)

ROLL NO: R2142220482

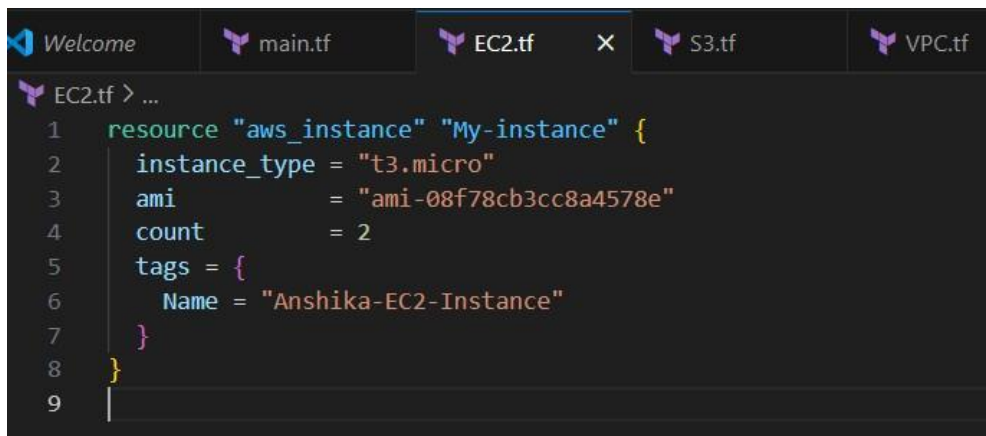
Write Terraform script to do perform following tasks on AWS cloud Platform

Step 1: Create Terraform Configuration File (main.tf):



```
main.tf > provider "aws"
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.83.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   # Configuration options
12   region = "eu-north-1"
13   access_key = "AKIAUIALHVPTX4GEHROZ"
14   secret_key = "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"
15 }
```

Step 2: To create two T3 Micro EC2 Instances create EC2.tf :



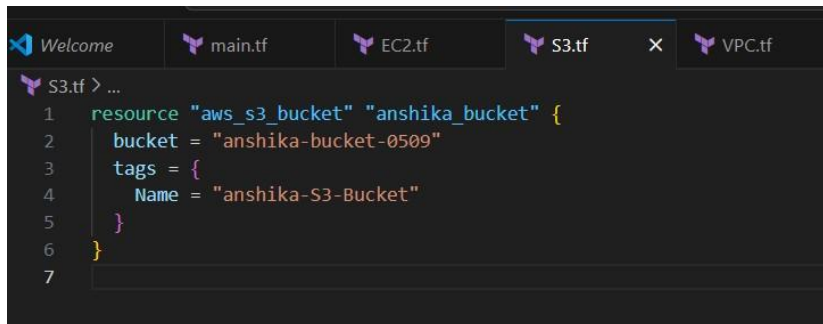
```
EC2.tf > ...
1 resource "aws_instance" "My-instance" {
2   instance_type = "t3.micro"
3   ami           = "ami-08f78cb3cc8a4578e"
4   count         = 2
5   tags = {
6     Name = "Anshika-EC2-Instance"
7   }
8 }
9
```

Step 3: To Create a VPC on AWS create VPC.tf :

```
Welcome  main.tf  EC2.tf  S3.tf  VPC.tf  X
VPC.tf > resource "aws_security_group" "gfg-sg"
1  resource "aws_vpc" "gfg-vpc" {
2    cidr_block = "10.0.0.0/16"
3  }
4
5  resource "aws_subnet" "gfg-subnet" {
6    vpc_id      = aws_vpc.gfg-vpc.id
7    cidr_block = "10.0.1.0/24"
8
9    tags = {
10     Name = "gfg-subnet"
11   }
12 }
13
14 resource "aws_internet_gateway" "gfg-gw" {
15   vpc_id = aws_vpc.gfg-vpc.id
16
17   tags = {
18     Name = "gfg-IG"
19   }
20 }
21
22 resource "aws_route_table" "gfg-rt" {
23   vpc_id = aws_vpc.gfg-vpc.id
24
25   route {
26     cidr_block = "0.0.0.0/0"
27     gateway_id = aws_internet_gateway.gfg-gw.id
28   }
29
30   tags = {
31     Name = "GFG-Route-Table"
32   }
33 }
34
35 resource "aws_route_table_association" "gfg-rta" {
36   subnet_id      = aws_subnet.gfg-subnet.id
37   route_table_id = aws_route_table.gfg-rt.id
38 }
```

```
Welcome  main.tf  EC2.tf  S3.tf  VPC.tf  X
VPC.tf > resource "aws_security_group" "gfg-sg"
22 resource "aws_route_table" "gfg-rt" {
33 }
34
35 resource "aws_route_table_association" "gfg-rta" {
36   subnet_id      = aws_subnet.gfg-subnet.id
37   route_table_id = aws_route_table.gfg-rt.id
38 }
39
40 resource "aws_security_group" "gfg-sg" {
41   name      = "my-gfg-sg"
42   vpc_id    = aws_vpc.gfg-vpc.id
43
44   ingress {
45     description = "TLS from VPC"
46     from_port   = 20
47     to_port     = 20
48     protocol    = "tcp"
49     cidr_blocks = ["0.0.0.0/0"]
50     ipv6_cidr_blocks = [ ":::/0" ]
51   }
52
53   egress {
54     from_port   = 0
55     to_port     = 0
56     protocol    = "-1"
57     cidr_blocks = ["0.0.0.0/0"]
58     ipv6_cidr_blocks = [ ":::/0" ]
59   }
60
61   tags = {
62     Name = "my-gfg-sg"
63   }
64 }
```

Step 4: To Create a S3 Bucket create S3.tf :

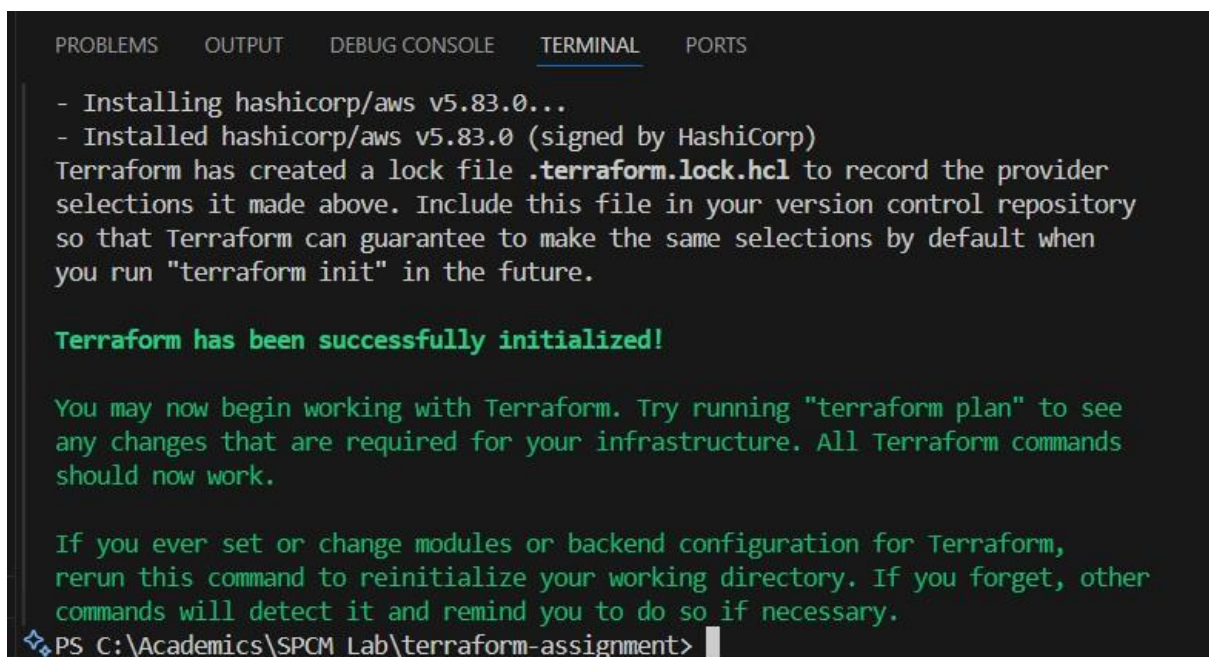
A screenshot of a code editor with a dark theme. At the top, there are tabs for 'Welcome', 'main.tf', 'EC2.tf', 'S3.tf' (which is active), and 'VPC.tf'. The 'S3.tf' tab shows a Terraform configuration for an S3 bucket. The code is as follows:

```
1 resource "aws_s3_bucket" "anshika_bucket" {  
2   bucket = "anshika-bucket-0509"  
3   tags = {  
4     Name = "anshika-S3-Bucket"  
5   }  
6 }  
7
```

Step 5: Initialize Terraform

Run the following command to initialize your Terraform working directory:

terraform init

A screenshot of a terminal window with a dark theme. The terminal shows the output of the 'terraform init' command. The output text is as follows:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
- Installing hashicorp/aws v5.83.0...  
- Installed hashicorp/aws v5.83.0 (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.  
  
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:\Academics\SPCM Lab\terraform-assignment>
```

Step 6: Review Plan

Run the following command to see what Terraform will do:

terraform plan

```
PS C:\Academics\SPCM Lab\terraform-assignment> 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.My-instance will be created
+ resource "aws_instance" "My-instance" {
  + ami                    = "ami-002f6e91abff6eb96"
  + 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)
  + enable_primary_ipv6    = (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)
  + vpc_id                 = (known after apply)
}
```

```
+ vpc_id                 = (known after apply)
}

# aws_vpc.gfg-vpc will be created
+ resource "aws_vpc" "gfg-vpc" {
  + arn                   = (known after apply)
  + cidr_block            = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id       = (known after apply)
  + enable_dns_hostnames   = (known after apply)
  + enable_dns_support     = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                     = (known after apply)
  + instance_tenancy       = "default"
  + ipv6_association_id    = (known after apply)
  + ipv6_cidr_block        = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id    = (known after apply)
  + owner_id               = (known after apply)
  + tags_all               = (known after apply)
}
```

Plan: 8 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:\Academics\SPCM Lab\terraform-assignment>
```

Step 7: Apply Changes:

Apply the changes to create the AWS resources:

terraform apply

```
PS C:\Academics\SPCM Lab\terraform-assignment> 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_instance.My-instance will be created
+ resource "aws_instance" "My-instance" {
  + ami                    = "ami-002f6e91abff6eb96"
  + 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)
  + enable_primary_ipv6    = (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)
}
```

It will ask for approval before creating, enter “yes” to continue.

```
}

# aws_vpc.gfg-vpc will be created
+ resource "aws_vpc" "gfg-vpc" {
  + arn                    = (known after apply)
  + cidr_block             = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id       = (known after apply)
  + enable_dns_hostnames  = (known after apply)
  + enable_dns_support    = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                    = (known after apply)
  + instance_tenancy      = "default"
  + ipv6_association_id   = (known after apply)
  + ipv6_cidr_block       = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id   = (known after apply)
  + owner_id              = (known after apply)
  + tags_all              = (known after apply)
}

Plan: 8 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
```

After approval, changes will be made


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

aws_instance.My-instance[0]: Creating...
aws_instance.My-instance[1]: Creating...
aws_instance.My-instance[1]: Still creating... [10s elapsed]
aws_instance.My-instance[0]: Still creating... [10s elapsed]
aws_instance.My-instance[1]: Creation complete after 16s [id=i-0a5d3439ad1f765b8]
aws_instance.My-instance[0]: Creation complete after 16s [id=i-012fdb724f77fc6fc]

aws_instance.My-instance[0]: Still creating... [10s elapsed]
aws_instance.My-instance[1]: Creation complete after 16s [id=i-0a5d3439ad1f765b8]
aws_instance.My-instance[0]: Creation complete after 16s [id=i-012fdb724f77fc6fc]

aws_instance.My-instance[1]: Creation complete after 16s [id=i-0a5d3439ad1f765b8]
aws_instance.My-instance[0]: Creation complete after 16s [id=i-012fdb724f77fc6fc]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Academics\SPCM Lab\terraform-assignment>
```

Verification

Creation of 2 EC2 instances:

<input type="checkbox"/>	Anshika-EC2-I...	i-0a5d3439ad1f765b8	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-51-20
<input type="checkbox"/>	Anshika-EC2-I...	i-012fdb724f77fc6fc	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-13-5'

Creation of VPC:

[Alt+S]							
Your VPCs (2) Info							
Find VPCs by attribute or tag							
<input type="checkbox"/>	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	
<input type="checkbox"/>	-	vpc-07fd53de924f834a0	Available	Off	10.0.0.0/16	-	
<input type="checkbox"/>	-	vpc-070899dfe309b16ce	Available	Off	172.31.0.0/16	-	

Creation of S3:

[Alt+S]							
Account snapshot - updated every 24 hours All AWS Regions							
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more							
View Storage Lens dashboard							
General purpose buckets Directory buckets							
General purpose buckets (1) Info All AWS Regions							
Buckets are containers for data stored in S3.							
Find buckets by name							
<input type="radio"/>	Name	AWS Region	IAM Access Analyzer	Creation date			
<input type="radio"/>	anshika-bucket-0509	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	April 24, 2025, 00:35:29 (UTC+05:30)			

Step 8: Cleanup Resources

When done experimenting, run the following command to destroy the created resources: *terraform destroy*

```
PS C:\Academics\SPCM Lab\terraform-assignment> terraform destroy
aws_vpc.gfg-vpc: Refreshing state... [id=vpc-07fd53de924f834a0]
aws_s3_bucket.anshika_bucket: Refreshing state... [id=anshika-bucket-0509]
aws_instance.My-instance[0]: Refreshing state... [id=i-012fdb724f77fc6fc]
aws_instance.My-instance[1]: Refreshing state... [id=i-0a5d3439ad1f765b8]
aws_internet_gateway.gfg-gw: Refreshing state... [id=igw-07d5fbc94abfed331]
aws_subnet.gfg-subnet: Refreshing state... [id=subnet-02542b6965874ece1]
aws_security_group.gfg-sg: Refreshing state... [id=sg-0405ea9df5fc1fe16]
aws_route_table.gfg-rt: Refreshing state... [id=rtb-05c84076d1e5d6ade]
aws_route_table_association.gfg-rta: Refreshing state... [id=rtbassoc-0b91adda59b05caf4]

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_instance.My-instance[0] will be destroyed
- resource "aws_instance" "My-instance" {
```

It will ask for approval before destroying, enter “yes” to continue.

```
- enable_dns_support           = true -> null
- enable_network_address_usage_metrics = false -> null
- id                           = "vpc-07fd53de924f834a0" -> null
- instance_tenancy             = "default" -> null
- ipv6_netmask_length          = 0 -> null
- main_route_table_id          = "rtb-01d909b31e6203bcd" -> null
- owner_id                    = "292081347559" -> null
- tags                         = {} -> null
- tags_all                     = {} -> null
# (4 unchanged attributes hidden)
}

Plan: 0 to add, 0 to change, 9 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
```

After approval, it starts destroying

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

aws_internet_gateway.gfg-gw: Destruction complete after 0s
aws_vpc.gfg-vpc: Destroying... [id=vpc-07fd53de924f834a0]
aws_vpc.gfg-vpc: Destruction complete after 1s
aws_instance.My-instance[0]: Still destroying... [id=i-012fdb724f77fc6fc, 10s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-0a5d3439ad1f765b8, 10s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-012fdb724f77fc6fc, 20s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-0a5d3439ad1f765b8, 20s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-012fdb724f77fc6fc, 30s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-0a5d3439ad1f765b8, 30s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-012fdb724f77fc6fc, 40s elapsed]
aws_instance.My-instance[1]: Still destroying... [id=i-0a5d3439ad1f765b8, 40s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-012fdb724f77fc6fc, 40s elapsed]
aws_instance.My-instance[1]: Destruction complete after 42s
aws_instance.My-instance[0]: Still destroying... [id=i-012fdb724f77fc6fc, 50s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-012fdb724f77fc6fc, 1m0s elapsed]
aws_instance.My-instance[0]: Still destroying... [id=i-012fdb724f77fc6fc, 1m10s elapsed]
aws_instance.My-instance[0]: Destruction complete after 1m13s

Destroy complete! Resources: 9 destroyed.
PS C:\Academics\SPCM Lab\terraform-assignment>
```