

ASSIGNMENT 1

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

SUBMITTED BY:

NAME: Akash Chauhan

Roll No : R2142220223

SAP ID :500105955

BATCH: DevOps HB1

SUBMITTED TO:

Dr. Hitesh Sharma

Code :

```
terraform { Name = "VPN-Gateway"
    required_providers {
        aws = {
            source = "hashicorp/aws"
            version = "5.84.0"
        }
    }
}

provider "aws" {
    region  = "ap-south-1"
    access_key = "AKIAYHJANSHKDNELWGU3"
    secret_key =
"JQnbEelRdjsowW1AIR2aCsL2oOzsvejkXXGX+Nj"
}

resource "aws_vpc" "default" {
    cidr_block = "10.0.0.0/16"
    tags = {
        Name = "Simple-VPC"
    }
}

resource "aws_vpn_gateway" "vpn_gw" {
    vpn_id = aws_vpc.default.id
    tags = {
        Name = "VPN-Gateway"
    }
}

resource "aws_instance" "ec2_1" {
    ami      = "ami-0e35ddab05955cf57"
    instance_type = "t2.micro"
    tags = {
        Name = "EC2-Instance-1"
    }
}

resource "aws_instance" "ec2_2" {
    ami      = "ami-0e35ddab05955cf57"
    instance_type = "t2.micro"
    tags = {
        Name = "EC2-Instance-2"
    }
}

resource "aws_s3_bucket" "akabucket" {
    bucket = "my-demo-s3-bucketakash"
    tags = {
        Name = "Terraform-S3-Bucketakash"
    }
}
```

Terraform init :

```

PS C:\Users\2004a\OneDrive - UPES\SEMESTER 6\shareable\Labs\Provisioning\terraform_assig_1> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.84.0"...
- Installing hashicorp/aws v5.84.0...
- Installed hashicorp/aws v5.84.0 (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:\Users\2004a\OneDrive - UPES\SEMESTER 6\shareable\Labs\Provisioning\terraform_assig_1>

```

Terraform validate :

The screenshot shows the Visual Studio Code interface with the Terraform extension installed. The left sidebar displays the project structure with files like main.tf, vpc.tf, and various lab files (lab1 through lab12). The main editor area shows a snippet of Terraform code for creating an S3 bucket and an AWS VPC gateway. Below the editor is the terminal pane, which shows the command 'terraform validate' being run and the output: 'Success! The configuration is valid.'

```

● PS C:\Users\2004a\OneDrive - UPES\SEMESTER 6\shareable\Labs\Provisioning\terraform_assig_1> terraform validate
Success! The configuration is valid.

```

Terraform plan :

The screenshot shows the Visual Studio Code interface with the Terraform extension installed. The left sidebar displays the project structure with files like main.tf, vpc.tf, and various lab files (lab1 through lab12). The main editor area shows a snippet of Terraform code for creating an EC2 instance. Below the editor is the terminal pane, which shows the command 'terraform plan' being run and the output detailing the actions Terraform will perform.

```

● PS C:\Users\2004a\OneDrive - UPES\SEMESTER 6\shareable\Labs\Provisioning\terraform_assig_1> 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_1 will be created
+ resource "aws_instance" "ec2_1" {
    + ami                      = "ami-0e35ddab05955cf57"
    + 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
        + instance.state
            = (known after apply)
}

```

vpc.tf terraform

PROVISIONING

```

+ arn
+ cidr_block
+ default_network_acl_id
+ default_route_table_id
+ default_security_group_id
+ dhcp_options_id
+ enable_dns_hostnames
+ enable_dns_support
+ enable_network_address_usage_metrics
+ id
+ instance_tenancy
+ ipv6_association_id
+ ipv6_cidr_block
+ ipv6_cidr_block_network_border_group
+ main_route_table_id
+ owner_id
+ tags
      + "Name" = "Simple-VPC"
+ tags_all
      + "Name" = "Simple-VPC"
}

```

aws_vpn_gateway.vpn_gw will be created

```

resource "aws_vpn_gateway" "vpn_gw" {
+ amazon_side_asn = (known after apply)
+ arn
+ id
+ tags
      + "Name" = "VPN-Gateway"
+ tags_all
      + "Name" = "VPN-Gateway"
+ vpc_id
      - (known after apply)
}

```

Plan: 5 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\2004a\OneDrive - UPES\SEMESTER 6\shareable\Labs\Provisioning\terraform_assig_1>

Terraform apply

Plan: 5 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\2004a\OneDrive - UPES\SEMESTER 6\shareable\Labs\Provisioning\terraform_assig_1> **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.ec2_1 will be created

```

resource "aws_instance" "ec2_1" {
+ ami
+ arn
+ associate_public_ip_address
+ availability_zone
+ cpu_core_count
+ cpu_threads_per_core
+ disable_api_stop
      = (known after apply)
}

```

Plan: 5 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.default: Creating...
aws_s3_bucket.akabucket: Creating...
aws_instance.ec2_2: Creating...
aws_instance.ec2_1: Creating...
aws_vpc.default: Creation complete after 3s [id=vpc-07700e2a9282031d0]
aws_vpn_gateway.vpn_gw: Creating...
aws_s3_bucket.akabucket: Creation complete after 4s [id=my-demo-s3-bucketakash]
aws_instance.ec2_2: Still creating... [10s elapsed]
aws_instance.ec2_1: Still creating... [10s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [10s elapsed]
aws_instance.ec2_1: Creation complete after 14s [id=i-071f898160ff4ea23]
aws_instance.ec2_2: Creation complete after 14s [id=i-0ea72dee726b1ddeb]
aws_vpn_gateway.vpn_gw: Still creating... [20s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [30s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [40s elapsed]
aws_vpn_gateway.vpn_gw: Creation complete after 46s [id=vgw-0c59964fa56e11e76]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

PS C:\Users\2004a\OneDrive - UPES\SEMESTER 6\shareable\Labs\Provisioning\terraform_assig_1>

Relevant changes on AWS Dashboard :

1. EC2 Instances

Dashboard Location: EC2 > Instances

What to check:

- Two instances named **EC2-Instance-1** and **EC2-Instance-2**
- State should be **running**
- Instance type should be **t2.micro**
- AMI ID should match your input (ami-09e143e99e8fa74f9)
- ◆ **Why:** Confirms Terraform launched the instances properly using the correct image and type.

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar is collapsed, and the main content area displays the 'Instances' page. The title bar shows the URL as https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#instances&v=3;\$case=true%5C,client=false;\$regex=tags:false%5C,client=false. The top navigation bar includes links for YouTube, UPES, Apps, WELL, Entertainment, MISC, AI, Courses, Grok, Qwen, DeepSeek - Into the..., Workspace Home, and other favorites. The top right corner shows the region as Asia Pacific (Mumbai) and a user profile for Akash. The main content area has a header 'Instances (2) Info' with a search bar and filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. Below the header is a table listing two instances: EC2-Instance-1 and EC2-Instance-2. Both instances are shown as 'Running' with a green status icon. They are both t2.micro instances, currently initializing, and are located in the ap-south-1b availability zone with public IPv4 addresses ec2-13-232-207-218.ap... and ec2-13-203-206-215.ap... respectively. The table also lists their instance IDs (i-071f98160ff4ea23 and i-0ea72dee726b1ddeb) and their respective public IPs (13.232.2 and 13.203.2).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
EC2-Instance-1	i-071f98160ff4ea23	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-232-207-218.ap...	13.232.2
EC2-Instance-2	i-0ea72dee726b1ddeb	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-203-206-215.ap...	13.203.2

[Create VPC](#)[Launch EC2 Instances](#)

Note: Your Instances will launch in the Asia Pacific region.

Service

[View co](#)

Resources by Region

You are using the following Amazon VPC resources

[Refresh Resources](#)

VPCs ▶ See all regions	Asia Pacific 2	NAT Gateways ▶ See all regions	Asia Pacific 0
Subnets ▶ See all regions	Asia Pacific 3	VPC Peering Connections ▶ See all regions	Asia Pacific 0
Route Tables ▶ See all regions	Asia Pacific 2	Network ACLs ▶ See all regions	Asia Pacific 2
Internet Gateways ▶ See all regions	Asia Pacific 1	Security Groups ▶ See all regions	Asia Pacific 7
Egress-only Internet Gateways ▶ See all regions	Asia Pacific 0	Customer Gateways ▶ See all regions	Asia Pacific 0
DHCP option sets ▶ See all regions	Asia Pacific 1	Virtual Private Gateways ▶ See all regions	Asia Pacific 1
Endpoints ▶ See all regions	Asia Pacific 0	Site-to-Site VPN Connections ▶ See all regions	Asia Pacific 0
Instance Connect Endpoints ▶ See all regions	Asia Pacific 0	Running Instances ▶ See all regions	Asia Pacific 2
Endpoint Services ▶ See all regions	Asia Pacific 0		

Setting

[Block Pi](#)[Zones](#)[Console](#)

Addit

[VPC Do](#)[All VPC](#)[Forums](#)[Report :](#)

AWS |

AWS Ne help you Network connect trouble security

[Get star](#)

Site-t

Amazon resource those re industry

[Creat](#)

[Virtual Private Gateways](#)

▼ Only this region

United States (N. Virginia)	0
United States (Ohio)	0
United States (N. California)	0
United States (Oregon)	0
Africa (Cape Town)	Region not opted-in
Asia Pacific (Hong Kong)	Region not opted-in
Asia Pacific (Hyderabad)	Region not opted-in
Asia Pacific (Jakarta)	Region not opted-in
Asia Pacific (Malaysia)	Region not opted-in
Asia Pacific (Melbourne)	Region not opted-in
Asia Pacific (Mumbai)	1
Asia Pacific (Osaka)	0
Asia Pacific (Seoul)	0
Asia Pacific (Singapore)	0
Asia Pacific (Sydney)	0
Asia Pacific (Thailand)	Region not opted-in
Asia Pacific (Tokyo)	0
Canada (Central)	0
Canada (Calgary)	Region not opted-in
Europe (Frankfurt)	0
Europe (Ireland)	0

2. VPC (Virtual Private Cloud)

Dashboard Location: VPC > Your VPCs**What to check:**

- One VPC named **Simple-VPC**

- CIDR block should be **10.0.0.0/16**

◆ **Why:** Verifies Terraform created a custom network for your infrastructure, isolating it from other resources.

3. VPN Gateway

Dashboard Location: VPC > VPN Gateways

What to check:

- One VPN Gateway named **VPN-Gateway**
- Should show status as **attached** to the Simple-VPC or **available**

◆ **Why:** Confirms VPN Gateway was created. Although it won't function without a customer gateway, it's a placeholder for site-to-site VPN.

The screenshot shows two views of the AWS VPC console. The top view is a list titled 'Your VPCs (2) info' showing two entries: '-' and 'Simple-VPC'. The bottom view is a detailed view for the 'Simple-VPC' entry, showing its configuration including VPC ID, State, Block Public Access, and DNS hostnames.

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option
-	vpc-0f1d1af924d640f7a	Available	Off	172.31.0.0/16	-	dopt-09666
Simple-VPC	vpc-07700e2a9282031d0	Available	Off	10.0.0.0/16	-	dopt-09666

vpc-07700e2a9282031d0 / Simple-VPC

Details		Info	
VPC ID	vpc-07700e2a9282031d0	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-0cde5ed390f4e3ab	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
		Block Public Access	Off
		DHCP option set	dopt-09666bdf14a03a769
		IPv4 CIDR	10.0.0.0/16
		Route 53 Resolver DNS Firewall rule groups	-
		DNS hostnames	Disabled
		Main route table	rtb-01d7c48737ba89caf
		IPv6 pool	-
		Owner ID	565393068500

Resource map | CIDs | Flow logs | Tags | Integrations

4. S3 Bucket

Dashboard Location: S3

What to check:

- Bucket named **my-demo-s3-bucketakash**
- Region: **ap-south-1 (Mumbai)**
- Default permissions: **private**
- Check **Tags** to see:
 - Key: Name, Value: Terraform-S3-Bucketakash

- ◆ Why: Ensures the S3 bucket was created and is properly tagged. Used for storing files, logs, backups, etc.

Terraform destroy :

```

> lab7      aws_instance.ec2_2: Still creating... [10s elapsed]
> lab8      aws_instance.ec2_1: Still creating... [10s elapsed]
> lab9      aws_vpn_gateway.vpn_gw: Still creating... [10s elapsed]
> lab10     aws_instance.ec2_1: Creation complete after 14s [id=i-071f898160ff4ea23]
> lab11     aws_instance.ec2_2: Creation complete after 14s [id=i-0ea72dee726b1dde]
> lab11
  ↳ main.tf
  ↳ vpc.tf
> lab12
  ↳ Lab Exercise 1- Insta...
  ↳ Lab Exercise 2- Terr...
  ↳ Lab Exercise 3- Terr...
  ↳ Lab Exercise 4- Provi...
  ↳ Lab Exercise 5- Provi...
  ↳ Lab Exercise 6- Terr...
  ↳ Lab Exercise 7- Terr...
  ↳ Lab Exercise 8- Terr...
  ↳ Lab Exercise 9- Crea...
  ↳ Lab Exercise 10- Crea...
  ↳ Lab Exercise 11- Crea...
  ↳ Lab Exercise 12- Crea...
  ↳ terraform_assig_1
    > .terraform
    & .terraform.lock.hcl
> OUTLINE

aws_instance.ec2_2: Still creating... [10s elapsed]
aws_instance.ec2_1: Still creating... [10s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [10s elapsed]
aws_instance.ec2_1: Creation complete after 14s [id=i-071f898160ff4ea23]
aws_instance.ec2_2: Creation complete after 14s [id=i-0ea72dee726b1dde]
aws_vpn_gateway.vpn_gw: Still creating... [20s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [30s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [40s elapsed]
aws_vpn_gateway.vpn_gw: Creation complete after 46s [id=vgw-0c59964fa56e11e76]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
PS C:\Users\2004a\OneDrive - UPES\SEMESTER 6\shareable\labs\Provisioning\terrafrom_assig_1> terraform destroy
aws_vpc.default: Refreshing state... [id=vpc-07700e29282031d0]
aws_s3_bucket.akabucket: Refreshing state... [id=my-demo-s3-bucketakash]
aws_instance.ec2_1: Refreshing state... [id=i-0ea72dee726b1dde]
aws_instance.ec2_1: Refreshing state... [id=i-071f898160ff4ea23]
aws_vpn_gateway.vpn_gw: Refreshing state... [id=vgw-0c59964fa56e11e76]

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.ec2_1 will be destroyed
- resource "aws_instance" "ec2_1" {
  = "ami-0e35ddab0595cf57" -> null
  - ami
  - arn
  - associate_public_ip_address
  - availability_zone
  - cpu_core_count
  = "ap-south-1b" -> null
  = 1 -> null
}

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_vpn_gateway.vpn_gw: Destroying... [id=vgw-0c59964fa56e11e76]
aws_s3_bucket.akabucket: Destroying... [id=my-demo-s3-bucketakash]
aws_instance.ec2_1: Destroying... [id=i-071f898160ff4ea23]
aws_instance.ec2_2: Destroying... [id=i-0ea72dee726b1dde]
aws_vpn_gateway.vpn_gw: Destruction complete after 0s
aws_vpn_gateway.vpn_gw: Still destroying... [id=vgw-0c59964fa56e11e76, 10s elapsed]
aws_instance.ec2_2: Still destroying... [id=i-0ea72dee726b1dde, 10s elapsed]
aws_instance.ec2_1: Still destroying... [id=i-071f898160ff4ea23, 10s elapsed]
aws_vpn_gateway.vpn_gw: Still destroying... [id=vgw-0c59964fa56e11e76, 20s elapsed]
aws_instance.ec2_1: Still destroying... [id=i-0ea72dee726b1dde, 20s elapsed]
aws_vpn_gateway.vpn_gw: Still destroying... [id=vgw-0c59964fa56e11e76, 30s elapsed]
aws_instance.ec2_1: Still destroying... [id=i-071f898160ff4ea23, 30s elapsed]
aws_instance.ec2_2: Still destroying... [id=i-0ea72dee726b1dde, 30s elapsed]
aws_vpn_gateway.vpn_gw: Destruction complete after 35s
aws_vpc.default: Destroying... [id=vpc-07700e29282031d0]
aws_vpc.default: Destruction complete after 1s
aws_instance.ec2_2: Still destroying... [id=i-0ea72dee726b1dde, 40s elapsed]
aws_instance.ec2_1: Still destroying... [id=i-071f898160ff4ea23, 40s elapsed]
aws_instance.ec2_2: Destruction complete after 41s
aws_instance.ec2_1: Destruction complete after 41s

Destroy complete! Resources: 5 destroyed.
PS C:\Users\2004a\OneDrive - UPES\SEMESTER 6\shareable\labs\Provisioning\terrafrom_assig_1> & 'c:\Users\2004a\OneDrive - UPES\SEMESTER 6\shareable\labs\Provisioning\terrafrom_assig_1\lab8'

```