

## Use case

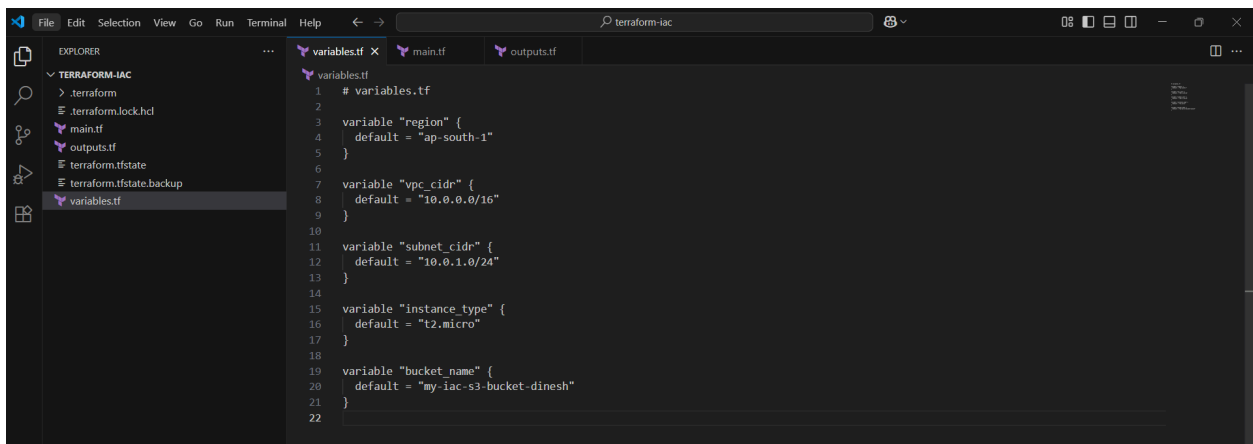
### Infrastructure as Code with Terraform

## Use case Description

Write Terraform scripts to deploy a VPC, EC2, and S3 with tagging and policies.

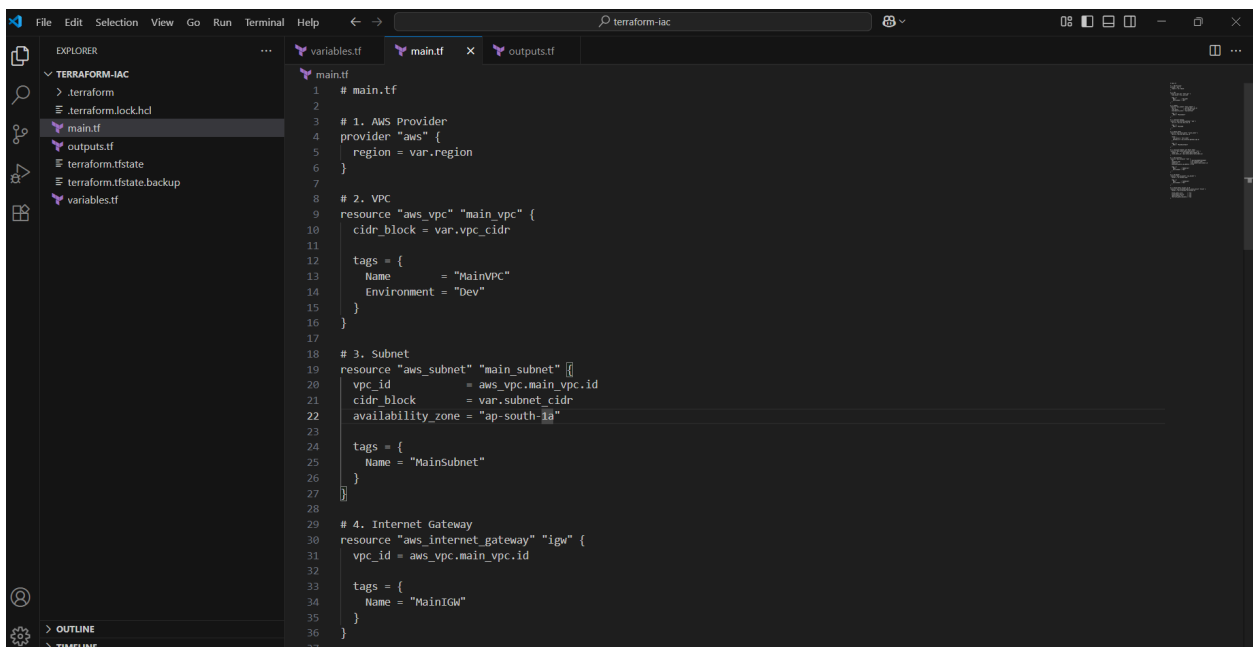
## Approach :

1. Install terraform and configure AWS credentials in powershell
2. Create the terraform files [variables.tf](#), [main.tf](#), [outputs.tf](#)
3. Initialize, Plan and apply from cmd



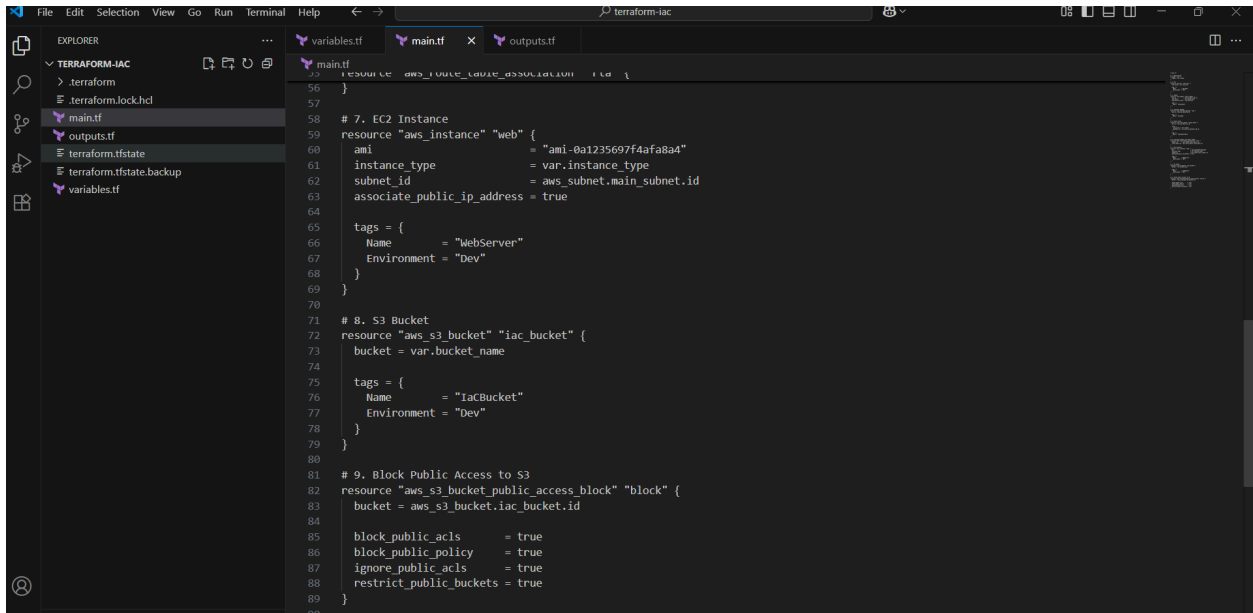
The screenshot shows the Visual Studio Code editor with the 'terraform-iac' project open. The Explorer sidebar on the left shows the project structure: .terraform, .terraform.lock.hcl, main.tf, outputs.tf, terraform.tfstate, terraform.tfstate.backup, and variables.tf. The main editor window displays the content of variables.tf, which defines several variables for the Terraform configuration.

```
1 # variables.tf
2
3 variable "region" {
4   default = "ap-south-1"
5 }
6
7 variable "vpc_cidr" {
8   default = "10.0.0.0/16"
9 }
10
11 variable "subnet_cidr" {
12   default = "10.0.1.0/24"
13 }
14
15 variable "instance_type" {
16   default = "t2.micro"
17 }
18
19 variable "bucket_name" {
20   default = "my-iac-s3-bucket-dinesh"
21 }
22
```

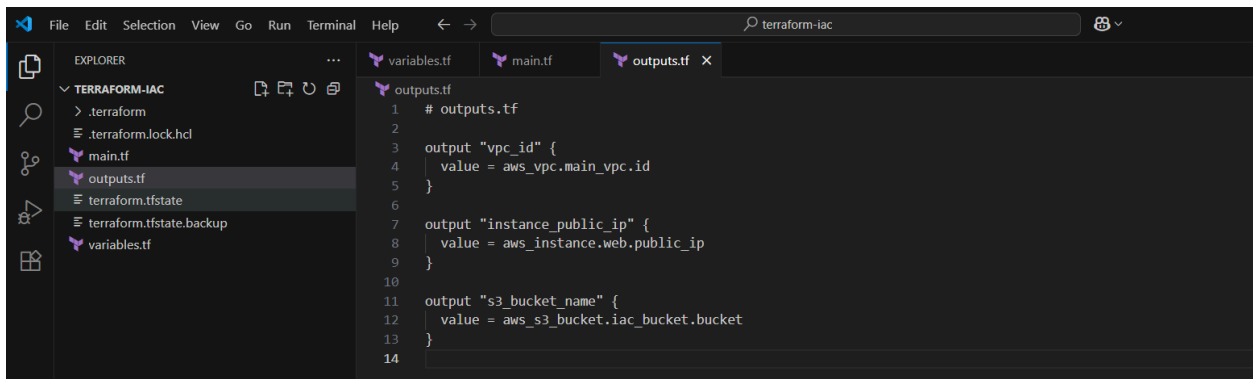


The screenshot shows the Visual Studio Code editor with the 'terraform-iac' project open. The Explorer sidebar on the left shows the project structure: .terraform, .terraform.lock.hcl, main.tf, outputs.tf, terraform.tfstate, terraform.tfstate.backup, and variables.tf. The main editor window displays the content of main.tf, which defines the Terraform configuration for an AWS VPC, Subnet, and Internet Gateway.

```
1 # main.tf
2
3 # 1. AWS Provider
4 provider "aws" {
5   region = var.region
6 }
7
8 # 2. VPC
9 resource "aws_vpc" "main_vpc" {
10   cidr_block = var.vpc_cidr
11
12   tags = {
13     Name       = "MainVPC"
14     Environment = "Dev"
15   }
16 }
17
18 # 3. Subnet
19 resource "aws_subnet" "main_subnet" {
20   vpc_id            = aws_vpc.main_vpc.id
21   cidr_block        = var.subnet_cidr
22   availability_zone = "ap-south-1a"
23
24   tags = {
25     Name = "MainSubnet"
26   }
27 }
28
29 # 4. Internet Gateway
30 resource "aws_internet_gateway" "igw" {
31   vpc_id = aws_vpc.main_vpc.id
32
33   tags = {
34     Name = "MainIGW"
35   }
36 }
37
```



```
56 }
57
58 # 7. EC2 Instance
59 resource "aws_instance" "web" {
60   ami           = "ami-0a1235697f4afa8a4"
61   instance_type = var.instance_type
62   subnet_id     = aws_subnet.main_subnet.id
63   associate_public_ip_address = true
64
65   tags = {
66     Name        = "WebServer"
67     Environment = "Dev"
68   }
69 }
70
71 # 8. S3 Bucket
72 resource "aws_s3_bucket" "iac_bucket" {
73   bucket = var.bucket_name
74
75   tags = {
76     Name        = "IaCBucket"
77     Environment = "Dev"
78   }
79 }
80
81 # 9. Block Public Access to S3
82 resource "aws_s3_bucket_public_access_block" "block" {
83   bucket = aws_s3_bucket.iac_bucket.id
84
85   block_public_acls       = true
86   block_public_policy     = true
87   ignore_public_acls     = true
88   restrict_public_buckets = true
89 }
```



```
1 # outputs.tf
2
3 output "vpc_id" {
4   value = aws_vpc.main_vpc.id
5 }
6
7 output "instance_public_ip" {
8   value = aws_instance.web.public_ip
9 }
10
11 output "s3_bucket_name" {
12   value = aws_s3_bucket.iac_bucket.bucket
13 }
14
```

```
PS C:\Users\dkarasu\desktop\AWSCloudPractitioner\UseCasesDocx\UC-23\terraform-iac> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.3.0...
- Installed hashicorp/aws v6.3.0 (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\dkarasu\desktop\AWScloudPractitioner\UseCasesDocx\UC-23\terraform-iac> 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.web will be created
+ resource "aws_instance" "web" {
  + ami              = "ami-0a1235697f4afa8a4"
  + arn              = (known after apply)
  + associate_public_ip_address = true
  + availability_zone = (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)
}
```

```
PS C:\Users\dkarasu\desktop\AWScloudPractitioner\UseCasesDocx\UC-23\terraform-iac> terraform apply
aws_vpc.main_vpc: Refreshing state... [id=vpc-0209a793591d0791d]
aws_s3_bucket.iac_bucket: Refreshing state... [id=my-iac-s3-bucket-dinesh]
aws_s3_bucket_public_access_block.block: Refreshing state... [id=my-iac-s3-bucket-dinesh]
aws_internet_gateway.igw: Refreshing state... [id=igw-0a9b1613330e2d04b]
aws_route_table.route_table: Refreshing state... [id=rtb-0764dfedb8ca1a185]

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.web will be created
+ resource "aws_instance" "web" {
  + ami              = "ami-0a1235697f4afa8a4"
  + arn              = (known after apply)
  + associate_public_ip_address = true
  + availability_zone = (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)
}
```

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

#### Outputs:

```
instance_public_ip = "13.232.234.121"
s3_bucket_name     = "my-iac-s3-bucket-dinesh"
vpc_id             = "vpc-0209a793591d0791d"
PS C:\Users\dkarasu\desktop\AWScloudPractitioner\UseCasesDocx\UC-23\terraform-iac> |
```

**VPC dashboard** < EC2 Global View [x] Filter by VPC [v]

**Virtual private cloud**

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections

**Security**

- Network ACLs
- Security groups

**PrivateLink and**

### vpc-0209a793591d0791d / MainVPC

**Details** [info]

<b>VPC ID</b> vpc-0209a793591d0791d	<b>State</b> Available	<b>Block Public Access</b> Off	<b>DNS hostnames</b> Disabled
<b>DNS resolution</b> Enabled	<b>Tenancy</b> default	<b>DHCP option set</b> dopt-0d2b28a8ce238d8e5	<b>Main route table</b> rtb-04e2b6cc6bf4db889
<b>Main network ACL</b> acl-009c4dd3a66b1e060	<b>Default VPC</b> No	<b>IPv4 CIDR</b> 10.0.0.0/16	<b>IPv6 pool</b> -
<b>IPv6 CIDR (Network border group)</b> -	<b>Network Address Usage metrics</b> Disabled	<b>Route S3 Resolver DNS Firewall rule groups</b> -	<b>Owner ID</b> 770424767712

**Resource map** | **CIDRs** | **Flow logs** | **Tags** | **Integrations**

**Tags**

Q Search tags

Key	Value
Name	MainVPC
Environment	Dev

**Manage tags** < 1 > [x]

**EC2** < Dashboard EC2 Global View [x] Events

**Instances**

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

**Images**

- AMIs
- AMI Catalog

**Elastic Block Store**

- Volumes
- Snapshots
- Lifecycle Manager

**Network & Security**

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

**Load Balancing**

- Load Balancers

### Instance summary for i-0dc18b3cae8efdb8 (WebServer)

Updated less than a minute ago

**Instance ID**  
i-0dc18b3cae8efdb8

**IPv6 address**  
-

**Hostname type**  
IP name: ip-10-0-1-18.ap-south-1.compute.internal

**Answer private resource DNS name**  
-

**Auto-assigned IP address**  
-

**IAM Role**  
-

**IMDSv2**  
Required

**Operator**  
-

**Public IPv4 address**  
-

**Instance state**  
Stopped

**Private IP DNS name (IPv4 only)**  
ip-10-0-1-18.ap-south-1.compute.internal

**Instance type**  
t2.micro

**VPC ID**  
vpc-0209a793591d0791d (MainVPC)

**Subnet ID**  
subnet-0e6b81de80a67f5a8 (MainSubnet)

**Instance ARN**  
arn:aws:ec2:ap-south-1:770424767712:instance/i-0dc18b3cae8efdb8

**Private IPv4 addresses**  
10.0.1.18

**Public DNS**  
-

**Elastic IP addresses**  
-

**AWS Compute Optimizer finding**  
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

**Auto Scaling Group name**  
-

**Managed**  
false

**Details** | **Status and alarms** | **Monitoring** | **Security** | **Networking** | **Storage** | **Tags**

**Tags**

Q Search tags

Key	Value
Environment	Dev
Name	WebServer

**Manage tags** < 1 > [x]

**Amazon S3** < General purpose buckets Directory buckets Table buckets Access Grants Access Points for general purpose buckets Access Points for directory buckets Object Lambda Access Points Multi-Region Access Points Batch Operations IAM Access Analyzer for S3

**Block Public Access settings for this account**

**Storage Lens**

- Dashboards
- Storage Lens groups
- AWS Organizations settings

**Feature spotlight** 11

**AWS Marketplace for S3**

### my-iac-s3-bucket-dinesh

**Objects** | **Properties** | **Permissions** | **Metrics** | **Management** | **Access Points**

**Bucket overview**

<b>AWS Region</b> Asia Pacific (Mumbai) ap-south-1	<b>Amazon Resource Name (ARN)</b> arn:aws:s3::my-iac-s3-bucket-dinesh	<b>Creation date</b> July 14, 2025, 22:14:28 (UTC+05:30)
---	--	---

**Bucket Versioning** [Edit]

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. | Learn more

**Bucket Versioning**  
Disabled

**Multi-factor authentication (MFA) delete**  
Disabled

**Tags (2)** [Edit]

You can use bucket tags to track storage costs and organize buckets. | Learn more

Key	Value
Environment	Dev
Name	iacBucket