Terraform

Terraform is an Open Source infrastructure as a code software tool. With the help of Terraform you can provision infrastructure on cloud.

But before you start working with Terraform you must understand different types of variables provided by the terraform –

1. **string** - It will hold string values

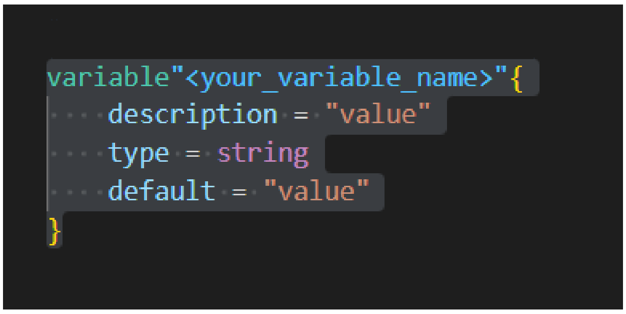
2. **number** - It will hold number values

3. **bool** - It will hold boolean value .i.e. true, false

4. **list** - It is used for the collection of values. Collection can be of type string, number

5. **map** - When you need to create key-value pair then need to use map variable type

1- Terraform variables are a way to store values that can be reused throughout your Terraform configuration.



```
variable"<your_variable_name>"{
    description = "value"
    type = string
    default = "value"
}
```

In description you can mention the full description of the variable.

Type can be string, number, bool, list and map.

2. Variables can be set in the command line when running Terraform commands using the -var flag.
3. Variables can also be set using a separate file, called a variable file, using the -var-file flag.
4. Variables can be accessed in Terraform configuration files using the var function, for example var.example_variable
5. Variables are useful for storing values that may change between environments, for example, different values for test and production environments.

We can see two types of variables in terraform:

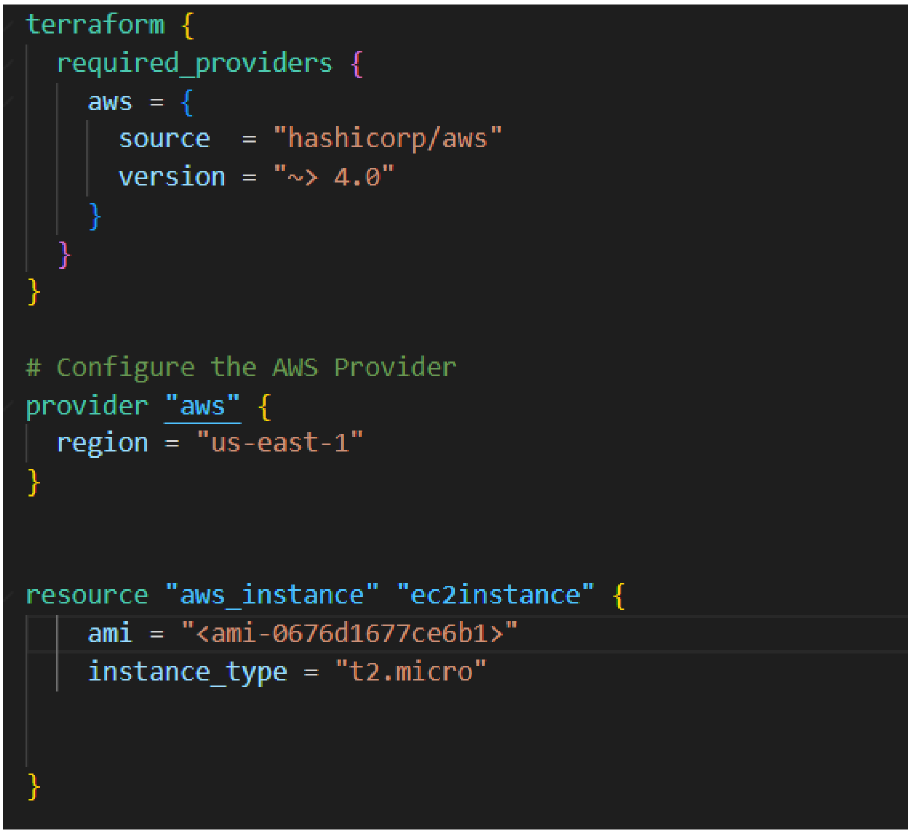
1- Simple 2- collection

1- Simple variables:
Are the one which hold only a single value such as string, number and bool

2- Collection Variables:
Such as list, map , set

Terraform Simple Variables:

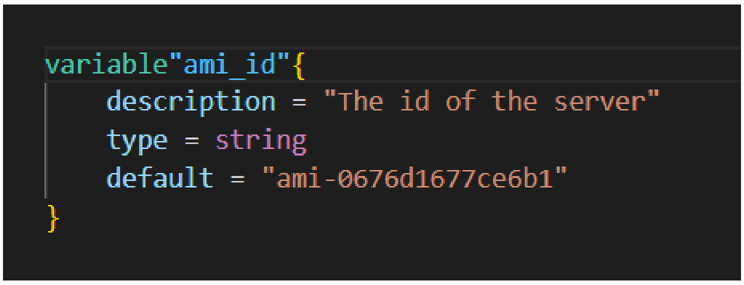
Here is the main.tf file



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.0"
    }
  }
}

# Configure the AWS Provider
provider "aws" {
  region = "us-east-1"
}


resource "aws_instance" "ec2instance" {
    ami = "<ami-0676d1677ce6b1>"
    instance_type = "t2.micro"


}
```

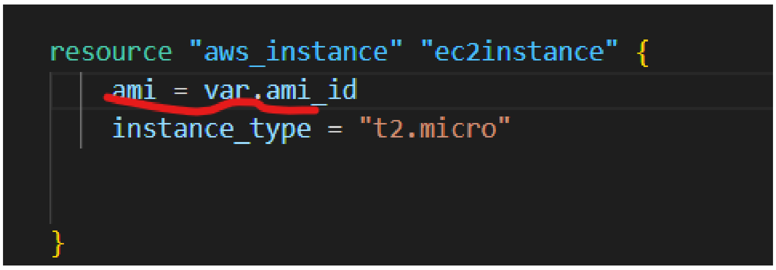
We are going to parameterized ‘ami’

So firstly we have to create a parameter in terraform file by defining variable block



```
variable"ami_id"{
    description = "The id of the server"
    type = string
    default = "ami-0676d1677ce6b1"
}
```

So let’s replace the hardcoded value of ami in main.tf with variables



```
resource "aws_instance" "ec2instance" {
    ami = var.ami_id
    instance_type = "t2.micro"


}
```

And now you can run the commands

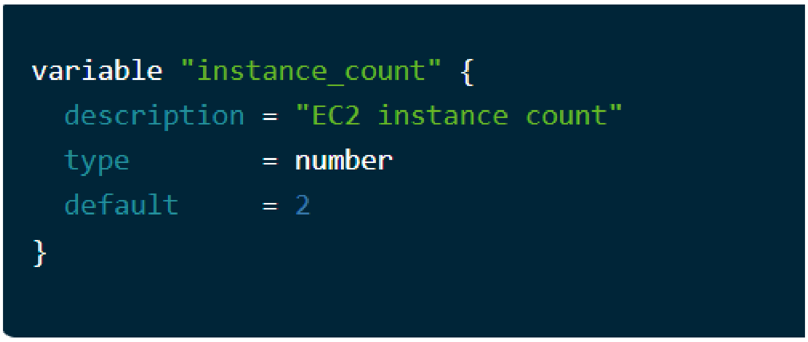
Terraform init: will initialize the working directory containing Terraform configuration files and install any required plugins

Terraform apply: will create the new resources

3- Number Variable type:

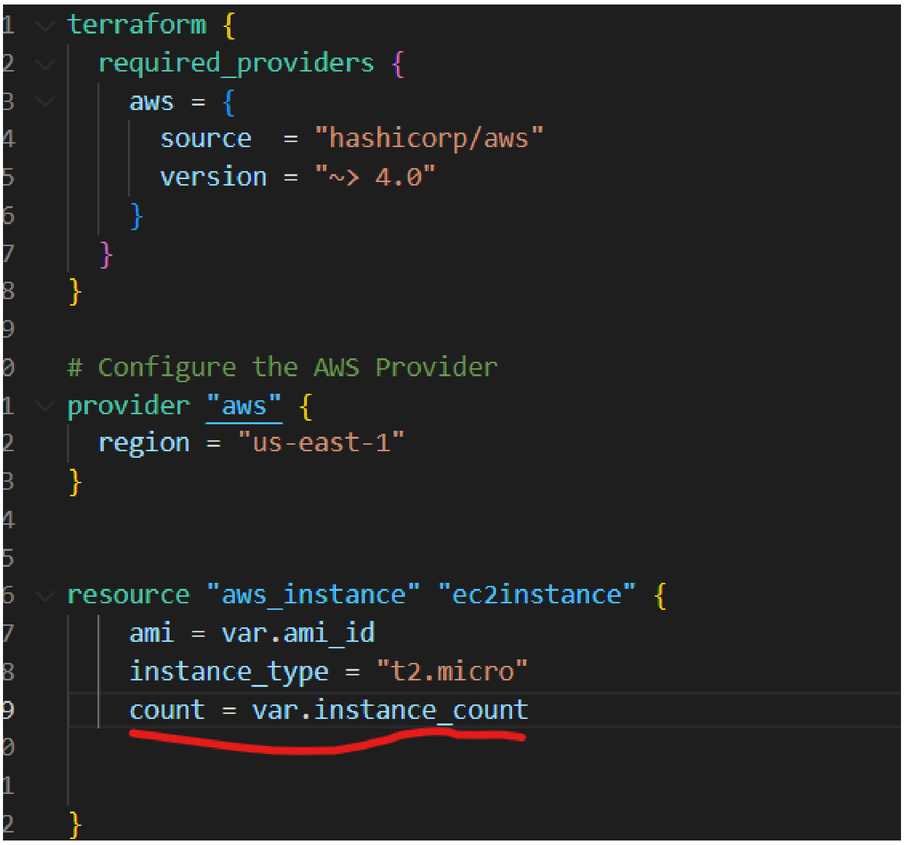
The next variable type we are going to take is ‘number’

For example, we are going to increase the `instance_count` of the `ec2_instances`.



```
variable "instance_count" {
  description = "EC2 instance count"
  type        = number
  default     = 2
}
```

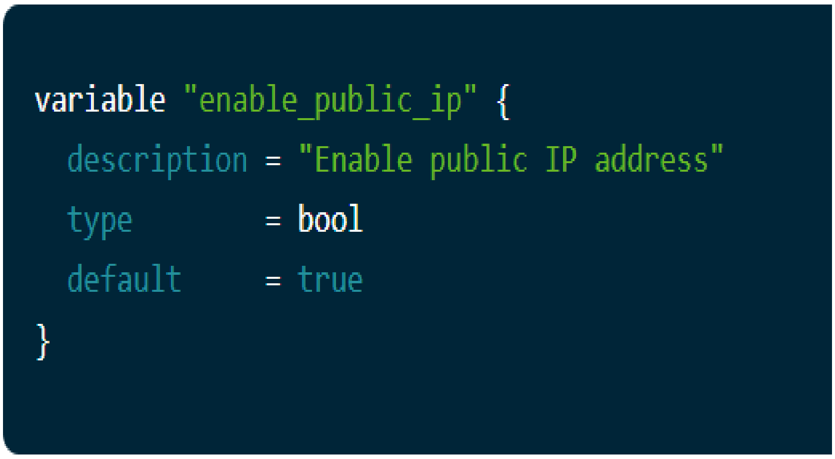
Final terraform file:



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.0"
    }
  }
}

# Configure the AWS Provider
provider "aws" {
  region = "us-east-1"
}


resource "aws_instance" "ec2instance" {
    ami = var.ami_id
    instance_type = "t2.micro"
    count = var.instance_count


}
```

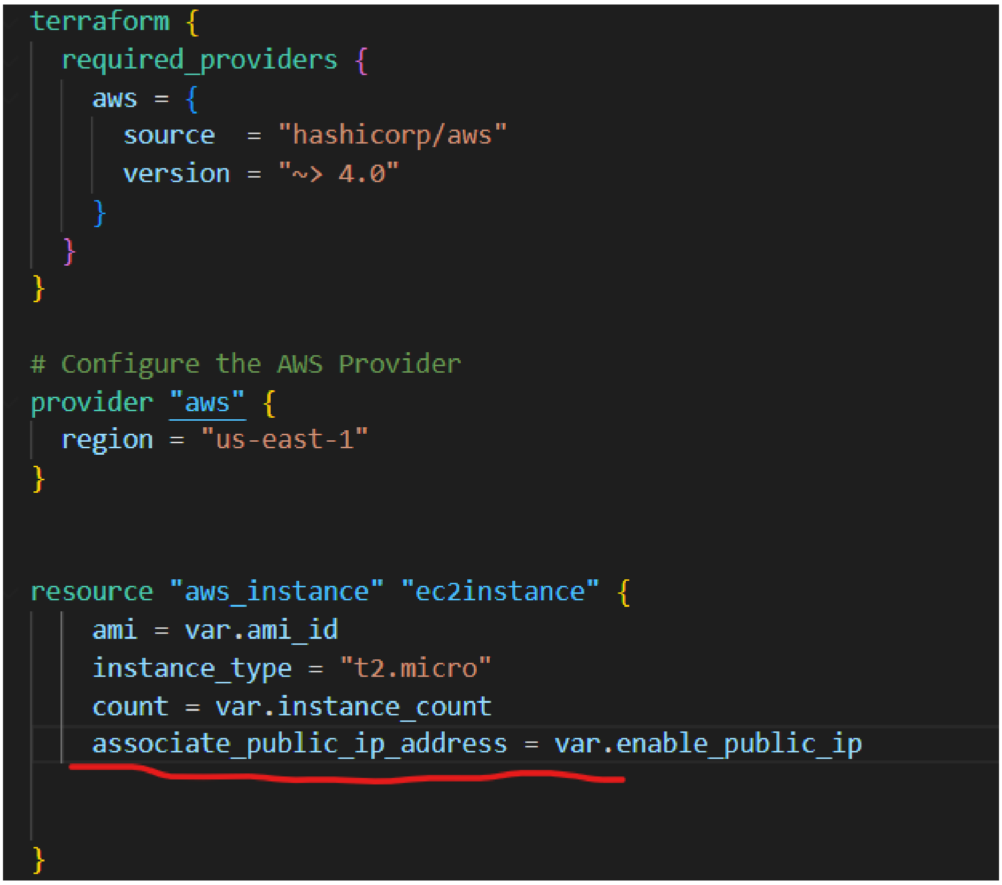
4- **Boolean variable type:**

The next variable type which we are going to discuss is `bool`.
The `bool` variable can be used to set `true` or `false` values inside your terraform file.



```
variable "enable_public_ip" {
  description = "Enable public IP address"
  type        = bool
  default     = true
}
```

Lets create a complete terraform file:



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.0"
    }
  }
}

# Configure the AWS Provider
provider "aws" {
  region = "us-east-1"
}


resource "aws_instance" "ec2instance" {
    ami = var.ami_id
    instance_type = "t2.micro"
    count = var.instance_count
    associate_public_ip_address = var.enable_public_ip


}
```