

Terraform variables

Table of Content

- Types of Terraform Variables
- Terraform Variables - string, number, bool
- Terraform Variables - list, set, map

Terraform input variables provides an easy and more flexible way to define values so that we can change the configuration of the terraform as per our need.

What are Input Variables in Terraform?

Answer - A value defined by the user which we can use inside terraform file for more customization.

```
variable "<YOUR_VARIABLE_NAME>" {  
  description    = "Instance type t2.micro" ← Meaning full description  
  type           = string ← Ex - string, number, bool, list, set, map..  
  default        = "t2.micro" ← variable default value  
}
```

Types of Terraform Variables

There are two types of variables in Terraform -

- **Simple values**
- **Collection Variable**

1.1 Simple Values variables

As the name suggests *Simple Values* variables are which hold only a single value. Here are the types of *Simple Value* variables -

1. string
2. number
3. bool

1.2 Collection Variable

In the collection variable, it consists of -

1. List
2. Map
3. Set

2. Terraform Variables - string, number, bool

Let's take a simple example in which we are going to set up an EC2 instance on AWS.

So to create an EC2 instance we need two things -

1. provider
2. resource

Here is the `main.tf` which we are going to parameterized using terraform variables.

```

provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "ec2_example" {

  ami          = "ami-0767046d1677be5a0"
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform EC2"
  }
}

```

2.1 string variable type - We are going parameterized *instance_type* = "t2.micro"

The first rule to create a parameter in terraform file is by defining a **variable block**

Example -

```

variable "instance_type" {
  description = "Instance type t2.micro"
  type        = string
  default     = "t2.micro"
}

```

description: Small or short description about the purpose of the variable

type: What type of variable it is going to be ex - string, bool, number ...

default: What would be the default value of the variable

Let's replace the hardcoded value of *instance_type* with variable

```

instance_type = var.instance_type

```

Here is our final terraform file after replacing the hardcoded value of a variable -

```

provider "aws" {
    region = "us-east-1"
}

resource "aws_instance" "ec2_example" {

    ami = "ami-0767046d1677be5a0"
    instance_type = var.instance_type

    tags = {
        Name = "Terraform EC2"
    }
}

variable "instance_type" {
    description = "Instance type t2.micro"
    type = string
    default = "t2.micro"
}

```

And now you can apply your terraform configuration

```
terraform apply
```

2.2 number variable type - We are going parameterized *instance_count = 2*

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`.

Let's create the variable first -

```

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

```

Here is the final terraform file with instance count -

```

provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "ec2_example" {

  ami          = "ami-0767046d1677be5a0"
  instance_type = "t2.micro"
  count = var.instance_count

  tags = {
    Name = "Terraform EC2"
  }
}

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

```

2.3 boolean variable type - We are going parameterized *enable_vpn_gateway = false*

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.

Here is an example to create your `bool` variable -

```

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

```

Let's create a complete terraform file with `bool` variable -

```

provider "aws" {
  region      = "eu-central-1"
  access_key  = "AKIATQ37NXB2AYK7R6PQ"
  secret_key  = "SlYglQm2JNSej8EHdhPTiu5l5ZD36URs3ed2NyYT"
}

resource "aws_instance" "ec2_example" {

  ami          = "ami-0767046d1677be5a0"
  instance_type = "t2.micro"
  count        = 1
  associate_public_ip_address = var.enable_public_ip

  tags = {
    Name = "Terraform EC2"
  }

}

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

```

3. Terraform Variables - *list, set, map*

When it comes to collection input variables then we are talking about -

1. List
2. Map
3. Set

3.1 List variable type

As the name suggests we are going to define a `list` that will contain more than one element in it.

Let's define our first List variable -

Here is the list of IAM users

```

variable "user_names" {
  description = "IAM usernames"
  type        = list(string)
  default     = ["user1", "user2", "user3s"]
}

```

Here is our final terraform file with List variables -

```

provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "ec2_example" {

  ami          = "ami-0767046d1677be5a0"
  instance_type = "t2.micro"
  count = 1

  tags = {
    Name = "Terraform EC2"
  }
}

resource "aws_iam_user" "example" {
  count = length(var.user_names)
  name  = var.user_names[count.index]
}

variable "user_names" {
  description = "IAM usernames"
  type        = list(string)
  default     = ["user1", "user2", "user3s"]
}

```

3.2 Map variable type

Terraform also supports the `map` variable type where you can define the key-value pair.

Let's take an example where we need to define project and environment, so we can use the `map` variable to achieve that.

Here is an example of `map` variable -

```

variable "project_environment" {
  description = "project name and environment"
  type        = map(string)
  default     = {
    project      = "project-alpha",
    environment  = "dev"
  }
}

```

Let's create a Terraform file

```
provider "aws" {  
  region = "us-east-1"  
}  
resource "aws_instance" "ec2_example" {  
  
  ami = "ami-0767046d1677be5a0"  
  instance_type = "t2.micro"  
  
  tags = var.project_environment  
}  
  
variable "project_environment" {  
  description = "project name and environment"  
  type        = map(string)  
  default     = {  
    project      = "project-alpha",  
    environment = "dev"  
  }  
}
```