

## Terraform Variables

### What Is a Variable?

A variable is like a blank that you fill in. It lets you change values easily without editing the whole code.

Think of it like filling out a form:

“What region do you want to use?”

“How many servers do you want?”

Instead of hardcoding answers, we use variables to fill them in later.

### • Why Use Variables?

---

Let's say you write this in Terraform:

```
instance_type = "t2.micro"
```

But what if you want to change it to *t3.micro* later?

Instead of editing the file, you can define a variable:

```
instance_type = var.instance_type
```

Now, you just change it in one place — easy!

### How to Create a Variable

**Example:**

```
variable "region" {  
  description = "AWS region to create resources"  
  type        = string  
  default     = "us-west-1"
```

```
}
```

This means:

- 
- Name: region
  - Type: string (text)
  - Default: us-west-1
- 

## How to Use Variables in Terraform Code

---

You use them like this:

```
provider "aws" {  
    region = var. region  
}
```

Here, var. region means *“use the value from the variable named region.”*

### Types of Variables with Simple Examples.

#### 1. string: A word or sentence

```
variable "name" {  
    type = string  
    default = "John"  
}
```

Use when: you want to store a name, type, or region.

## 2. number: A number

```
variable "count" {  
    type = number  
    default = 3  
}
```

Use when: you want to launch 3 servers or set a port like 8080.

## 3. bool: Yes or No (true or false)

```
variable "enable_backup" {  
    type = bool  
    default = true  
}
```

Use when: you want to turn something on or off.

## 4. list(string): A list of items

```
variable "zones" {  
    type = list(string)  
    default = ["us-west-1a", "us-west-1b"]  
}
```

Use when: you have many items like zones, names, etc.

## 5. map(string): Key and value pairs

```
variable "tags" {  
    type = map(string)  
    default = {  
        Name = "MyServer"  
        Environment = "dev"  
    }  
}
```

Use when: you want to group values by a name, like labels.

## 6. object: A group of values with names

```
variable "db_info" {  
  type = object ({  
    username = string  
    password = string  
    port     = number  
  })  
  
  default = {  
    username = "admin"  
    password = "pass123"  
    port     = 3306  
  }  
}
```

Use when: you have grouped info, like database settings.

## 7. tuple: A list with different types

```
variable "example" {  
  type = tuple ([string, number, bool])  
  default = ["hello", 123, true]  
}  
  
variable "example" {  
  type = tuple ([string, number, bool])  
  default = ["hello", 123, true]  
}
```

Use when: you want fixed items like *[name, age, active]*.

---

## How to Set Variable Values

### Option 1: In a

#### **terraform.tfvars**

```
region = "us-east-1"  
name   = "MyServer"
```

Terraform uses this file automatically.

### Option 2: From the command line

```
terraform apply -var="region=us-east-1"
```

### Option 3: As an environment variable

```
export TF_VAR_region=us-east-1
```

---

## Tips

- Use description to explain your variable.
  - Use default when you want to give a standard value.
  - Keep your variables in a file like variables.tf.
  - Set values in terraform.tfvars to keep things clean.
-

---

## Summary Table

---

Type	Example	Meaning
string	"hello"	Text value
number	5, 100	A number
bool	true, false	Yes or no
list	["a", "b", "c"]	List of values
map	{key = "value"}	Key-value pairs
object	{name = "x", id = 1}	Group of named values
tuple	["a", 1, true]	List of different types

---