

Terraform.tfvars Files

Table of Content

- Terraform variable file - [variable.tf](#)
- Terraform tfvars file - terraform.tfvars
- Terraform multiple tfvars file
- Terraform setting variable using command line var

1. Terraform Variables

Create a `main.tf` file

```
provider "aws" {
    region      = "eu-central-1"
    access_key  = "AKIATQ37NXB2AYK7R6PQ"
    secret_key  = "SlYglQm2JNSej8EHdhPTiu5l5ZD36URsedf32kNT"
}

resource "aws_instance" "ec2_example" {

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

    tags = {
        Name = "Terraform EC2"
    }
}
```

Now create `variable.tf`

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

Here is the sequence of terraform commands which you need to run

```
terraform init
```

```
terraform plan
```

```
terraform apply
```

And finally, you can destroy the complete environment using -

```
terraform destroy
```

2. Terraform tfvars file - terraform.tfvars

Create a `main.tf` file

```
provider "aws" {  
    region      = "eu-central-1"  
    access_key  = "AKIATQ37NXB2AYK7R6PQ"  
    secret_key  = "SlYglQm2JNSej8EHdhPTiu5l5ZD36URsedf32kNT"  
}  
  
resource "aws_instance" "ec2_example" {  
  
    ami          = "ami-0767046d1677be5a0"  
    instance_type = var.instance_type  
  
    tags = {  
        Name = "Terraform EC2"  
    }  
}
```

Now create `variable.tf` but without any default values

```
variable "instance_type" {  
}
```

Finally create `terraform.tfvars`

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

Here is the sequence of terraform commands which you need to run

```
terraform init
```

```
terraform plan
```

```
terraform apply
```

And finally, you can destroy the complete environment using -

```
terraform destroy
```

3. Terraform multiple tfvars file

There can be situations where you need to create multiple tfvars files based on the environment like a stage, production.

So in such a scenario, you can create one tfvars file for each environment -

1. stage.tfvars
2. production.tfvars

First Create your main.tf

```
provider "aws" {
  region      = "eu-central-1"
  access_key  = "AKIATQ37NXB2AYK7R6PQ"
  secret_key  = "S1YglQm2JNSej8EHdhPTiu5l5ZD36URsedf32kNT"
}
resource "aws_instance" "ec2_example" {

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

  tags = {
    Name = var.environment_name
  }
}
```

Second, create your variable.tf

```
variable "instance_type" {
}

variable "environment_name" {
}
```

And third, create your `stage.tfvars` for staging

```
instance_type="t2.micro"

environment_name = "stage"
```

Apply the terraform configuration using the following command -

```
terraform init
```

```
terraform plan -var-file="stage.tfvars"
```

And then

```
terraform apply -var-file="stage.tfvars"
```

You can destroy the environment by running the following command

```
terraform destroy -var-file="stage.tfvars"
```

4. Terraform setting variable using command line var

Create a `main.tf` file

```
provider "aws" {  
    region      = "eu-central-1"  
    access_key  = "AKIATQ37NXB2AYK7R6PQ"  
    secret_key  = "SlYglQm2JNSej8EHdhPTiu5l5ZD36UsdfA99NyYT"  
}  
  
resource "aws_instance" "ec2_example" {  
  
    ami          = "ami-0767046d1677be5a0"  
    instance_type = var.instance_type  
  
    tags = {  
        Name = "Terraform EC2"  
    }  
}  
  
variable "instance_type" {  
}
```

Now run the following terraform command

```
terraform init
```

```
terraform plan -var="instance_type=t2.micro"
```

```
terraform apply -var="instance_type=t2.micro"
```

Finally, you destroy the EC2 environment using the following command -

```
terraform destroy -var="instance_type=t2.micro"
```