

Format Lab 1.1

Create a variables.tf and main.tf. Use `terraform fmt` to automatically format the files.

Overview

In this lab you will

- create a variables.tf and a main.tf
- use terraform fmt to tidy them up

Starting point

Your files should currently look like this:

- provider.tf
- terraform {
- required_providers {
- azurerm = {
- source = "hashicorp/azurerm"
- version = "~>3.1"
- }
- }
- }
-
- provider "azurerm" {
- features {}
-
- storage_use_azuread = true
- }

Open the editor

1. Open the [Cloud Shell](#)

Authenticate and check you are in the right subscription.

2. Change to your working directory
3. `cd ~/terraform-basics`
4. Create empty variables.tf and main.tf files
5. `touch variables.tf main.tf`

The touch command updates the modified timestamp on linux files. If the file does not exists then it also creates files.

6. Open the Monaco editor

7. `code .`

Using `code .` opens the current working directory.

main.tf

Create the initial main.tf.

1. Copy the code block

```
2. resource azurerm_resource_group basics {  
3.   name = var.name  
4.   location="West Europe"  
5. }
```

6. Select the main.tf file in the editor's file browser

7. Paste

8. Save the file

terraform fmt

The format used within Terraform files is called HashiCorp Configuration Language or HCL for short. The formatting of these files is very standardised and has been hardcoded into the terraform binary itself. (It is based on the principles of gofmt, which is not that surprising as terraform is written in Go.)

1. Run `terraform fmt`
2. `terraform fmt`

The command will find all valid *.tf files in the current working directory and will format them correctly. The command outputs the filenames of any modified files.

```
main.tf
```

3. Reselect the main.tf

Click on the main.tf filename in the editor's file browser.

The file should now have indentation, alignment and the resource type and name are now quoted correctly.

Example:

```
resource "azurerm_resource_group" "basics" {  
  name      = var.name  
  location = "West Europe"  
}
```

Create a variables.tf

1. Copy and paste into variables.tf
2.

```
variable "resource_group_name" {
```
3.

```
  description = "Name for the resource group"
```
4.

```
  type        = "string"
```
5.

```
  default     = "terraform-basics"
```
6.

```
}
```
7. Save the file
8. Rerun `terraform fmt` and reselect the file in the editor.
9. `terraform fmt`

The up cursor key recalls previous commands.

You'll see the value for the type argument, string does not need quotes.

Variables can be strings, booleans, numbers or more complex types. We'll cover [variables](#) in more depth later.

Summary

We have reached the end of the lab. You have created a couple of files and used `terraform fmt` to fix common formatting issues.

Move onto the next lab and we'll customise and validate the files.

s
