Plan Lab 1.3

Start working with terraform.tfvars to specify your variable values and then use `terraform plan` to display the actions that Terraform will take.

Overview

In this lab you will

- create a terraform.tfvars
- specify the value for location
- display a plan

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
}
```

- variables.tf

```
variable "resource_group_name" {
  description = "Name for the resource group"
  type        = string
  default     = "terraform-basics"
}

variable "location" {
  description = "Azure region"
  type        = string
  default     = "West Europe"
}
```

-

- main.tf

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

terraform.tfvars

The variables.tf file has been used to **declare** the variables, including their type and default values. A more complete configuration may add information such as a description and validation criteria.

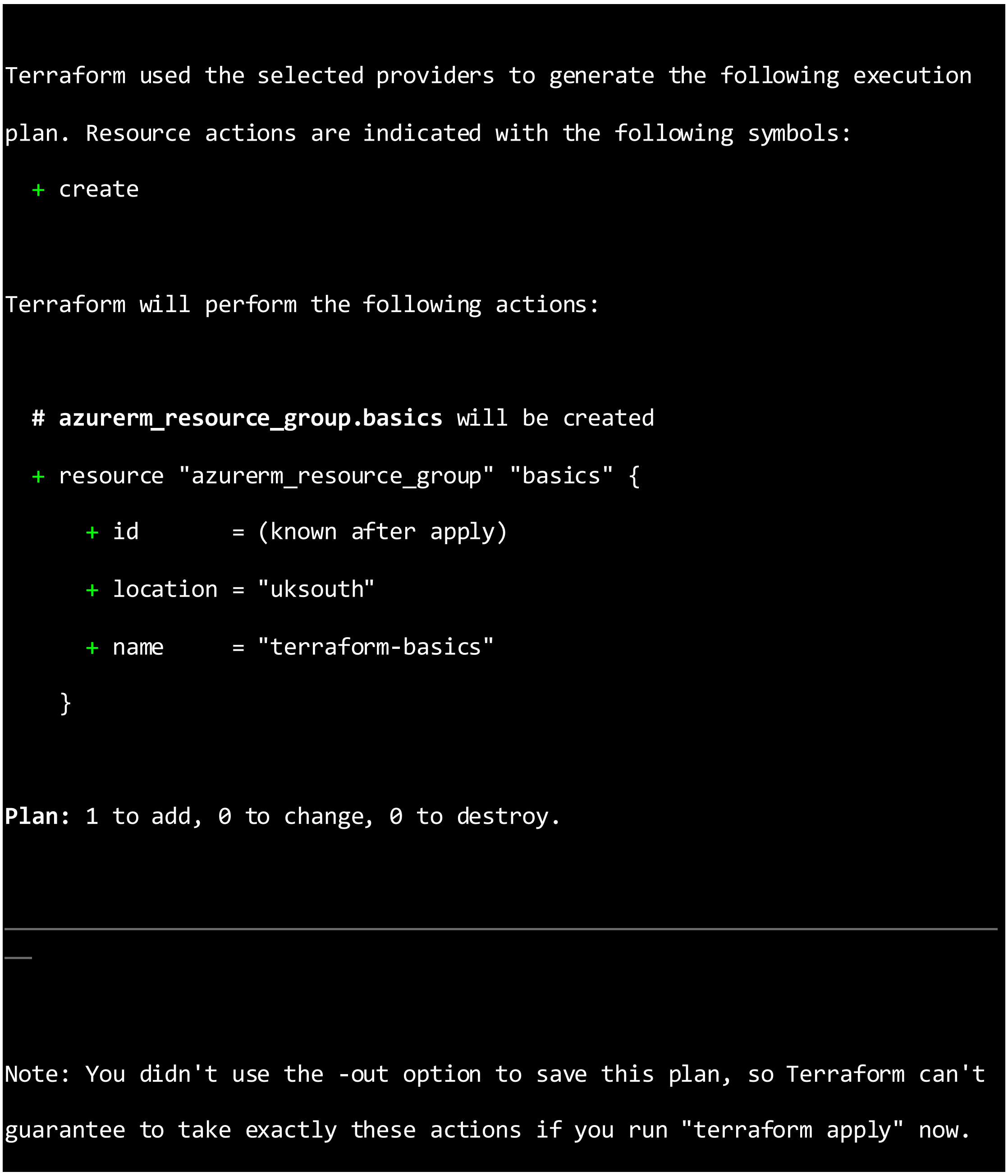
When planning a deployment then you usually **specify** the values for those variables using a terraform.tfvars file.

There are other ways of specifying variables, such as via command line arguments, environment variables and *.auto.tfvars files.

1. Create a terraform.tfvars file
2. `touch terraform.tfvars`
3. Click on the refresh icon in the editor's file explorer
4. Specify the location

```
terraform plan
```

Example output:



```
Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_resource_group.basics will be created
  + resource "azurerm_resource_group" "basics" {
      + id       = (known after apply)
      + location = "uksouth"
      + name     = "terraform-basics"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────────
─

Note: You didn't use the -out option to save this plan, so Terraform can't
guarantee to take exactly these actions if you run "terraform apply" now.
```

The output shows that the resource group will be created.

Summary

You have started using the terraform.tfvar and have displayed the output from `terraform plan`.

In the next lab we will finally create something and will then explore the state file.