Lab 4. Terraform data sources.

Background:

Here we learn how to use data sources to get data directly from third parties, rather than defining it ourselves.

Tasks:

1. Make a directory called ‘lab4’ underneath the terraform-labs directory.

2. Change into the directory.

3. Create the following files: *main.tf, resource.tf, variables.tf*

Here is the source code for the main.tf file:

```
provider "aws" {

    access_key  = "AKIAIZAHH7GJN6ASXVVA"
    secret_key = "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"
    region = "${var.region}"
}
```

Here is the source code for the variables.tf file.

```
variable "region" {
    default = "us-east-1"
}

variable "vpc_cidr" {
    default = "192.168.0.0/16"
}

variable "vpc_subnet_cidr" {
    type = "list"
    default =
["192.168.100.0/24","192.168.101.0/24","192.168.102.0/24"]
}

variable "ami_instance" {
    default = "ami-0ac019f4fcb7cb7e6"
}

variable "ami_instance_type" {
    default = "t2.micro"
}

data "aws_availability_zones" "aaz" {}
```

Here is the source code for the resource.tf file.

```
resource "aws_vpc" "main_vpc" {
    cidr_block = "${var.vpc_cidr}"
    instance_tenancy = "default"
    tags {
        Name = "Main"
        Location = "London"
    }
}

resource "aws_subnet" "vpc_subnets" {
    count = "${length(var.vpc_subnet_cidr)}"
    vpc_id = "${aws_vpc.main_vpc.id}"
    cidr_block = "${element(var.vpc_subnet_cidr,count.index)}"
    availability_zone = "$
{element(data.aws_availability_zones.aaz.names,count.$
    tags {
        Name = "subnet-${count.index+1}"
   }
}
```

Note: For each student, change the CIDR for the VPC and subnets to a unique value!

4. Run the following commands:

```
 > terraform init
```

Note that the '>' refers to the bash shell prompt and is not part of the command.

This command initializes the terraform directory structure.

7. Run the following command:

```
> terraform plan
```

This should print out what actions terraform will take.

8. Run the following command:

```
> terraform apply
```

Assuming that this works correctly, AWS create a VPC, three subnets located in three different availabilitgy zones, and an ami instance running on each subnet.

9.. Run the following:

```
> terraform destroy
```

This will now destroy the formerly created AWS vpc, and all subnets.