[Optional] Additional Terraform Configuration Example - Exercise

Introduction

You learned how you can create an EC2 instance using Terraform. In this optional exercise, you'll create a database instance in Terraform. The goal of this exercise is for you to practice using Terraform and its documentation.

Exercise Question

You are given the ID of a VPC created in the us-east-1 region. You will need to create a MySQL RDS database instance inside a subnet of the given VPC. The VPC contains two private subnets; the IDs of these subnets are not given to you.

Here are the specifications of the database:

- username: should be defined as a variable with a default value of "admin_user"
- password: should be defined as a variable. Its value is specified in the tfvars file.
- port number: 3306
- database instance class: db.t3.micro (this is to determine the memory and computation capacity of the database)
- allocated storage: 10 GiB

From this database instance, you want to return the database hostname, username, password and port number as output values.

Here are some useful links:

- Resource aws_db_instance , list of its arguments and attributes .
- Resource aws_db_subnet_group , list of its arguments and attributes .
- Data source aws_subnets , list of its attributes .

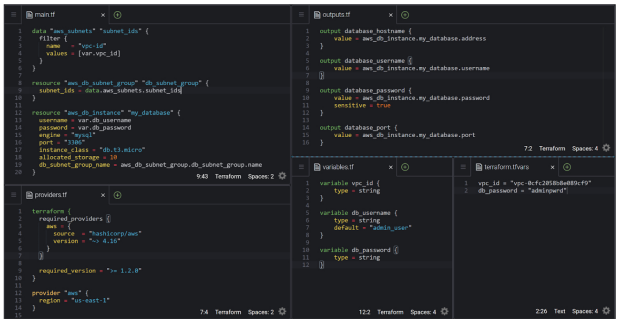
How should you replace the question marks in these configuration files?

Hints:

- If you look at the arguments of aws_db_instance, you will notice that you need to specify a name for a subnet group when you don't want the database to be launched in the default VPC. In Terraform, you can create the subnet group as another resource block in the main file and then use its attributes in the database resource block. The subnet group should consist of at least two subnet ids, and each subnet should reside in a different availability zone. This is an AWS requirement (for more info, check here). You can assume that the given VPC contains two subnets where each subnet belongs to a different availability zone.
- In the main file, the first data block (of type aws_subnets) gets you the subnet IDs of the given VPC.

Solutions:

When you're ready, check out the solution provided below



Mark as completed