

[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?

main.tf

```

1 data "aws_subnets" "subnet_ids" {
2   filter {
3     name   = "vpc-id"
4     values = [var.vpc_id]
5   }
6 }
7
8 resource "aws_db_subnet_group" "db_subnet_group" {
9   subnet_ids = ?
10 }
11
12 resource "aws_db_instance" "my_database" {
13   ?
14   ?
15   ?
16   ?
17   ?
18 }
19
20 ?
21 ?
22 ?

```

outputs.tf

```

1 output database_hostname {
2   ?
3 }
4
5 output database_username {
6   ?
7 }
8
9 output database_password {
10  ?
11  sensitive = true
12 }
13
14 output database_port {
15  ?
16 }
17

```

providers.tf

```

1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "~> 4.14"
6     }
7   }
8   required_version = "~> 1.2.0"
9 }
10
11 provider "aws" {
12   region = "us-east-1"
13 }
14

```

variables.tf

```

1 variable vpc_id {
2   ?
3 }
4
5 variable db_username {
6   ?
7 }
8
9 variable db_password {
10  ?
11 }
12

```

terraform.tfvars

```

1 vpc_id = "vpc-b0fc2058bde0b0c9f"
2 db_password = "admin123456"

```

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

main.tf

```

1 data "aws_subnets" "subnet_ids" {
2   filter {
3     name   = "vpc-id"
4     values = [var.vpc_id]
5   }
6 }
7
8 resource "aws_db_subnet_group" "db_subnet_group" {
9   subnet_ids = data.aws_subnets.subnet_ids
10 }
11
12 resource "aws_db_instance" "my_database" {
13   username = var.db_username
14   password = var.db_password
15   engine   = "mysql"
16   port     = 3306
17   instance_class = "db.t3.micro"
18   allocated_storage = 10
19   db_subnet_group_name = aws_db_subnet_group.db_subnet_group.name
20 }
21

```

outputs.tf

```

1 output database_hostname {
2   value = aws_db_instance.my_database.address
3 }
4
5 output database_username {
6   value = aws_db_instance.my_database.username
7 }
8
9 output database_password {
10  value = aws_db_instance.my_database.password
11  sensitive = true
12 }
13
14 output database_port {
15  value = aws_db_instance.my_database.port
16 }
17

```

providers.tf

```

1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "~> 4.14"
6     }
7   }
8   required_version = "~> 1.2.0"
9 }
10
11 provider "aws" {
12   region = "us-east-1"
13 }
14

```

variables.tf

```

1 variable vpc_id {
2   type = string
3 }
4
5 variable db_username {
6   type = string
7   default = "admin_user"
8 }
9
10 variable db_password {
11  type = string
12 }
13

```

terraform.tfvars

```

1 vpc_id = "vpc-b0fc2058bde0b0c9f"
2 db_password = "admin123456"

```

Mark as completed

👍 Like 👎 Dislike 📧 Report an issue

