**Task Overview:**

In this task, you will design and provision a detailed infrastructure setup using Terraform. This includes creating four backend servers, two frontend servers, setting up networking components such as security groups, internet gateway, VPC, subnet, and an S3 bucket. Additionally, you will create an IAM user named "BOB" with the description "CTO," and finally, you will initialize a Git repository, commit your Terraform configurations, and share the Git URL.

**Instructions:**

**Define Variables:**

Create a variables.tf file to define variables for your infrastructure setup. Include variables for instance types, availability zones, region, CIDR blocks, S3 bucket name, IAM user name, and any other parameters you may need.

Create a terraform.tfvars file to set the values for these variables. Make sure to include sensitive information like access keys and secret keys in a separate .gitignore file to avoid accidental

Exposure.

**Set Up Networking Components:**

Define a Virtual Private Cloud (VPC) using Terraform and specify the CIDR block.

Create a public subnet within the VPC that spans at least two availability zones.

Set up an internet gateway and attach it to the VPC to enable internet access for resources within the VPC.

Configure a security group to control inbound and outbound traffic for your servers. Allow HTTP/HTTPS traffic for frontend servers and SSH access for backend servers.

**Provision Backend Servers: (use tags)**

Define Terraform resources for the backend servers. Use the specified instance types and number of instances.

Associate the backend servers with the previously created subnet and security group.

Ensure that each backend server has appropriate tags for identification and organization.

**Provision FrontEnd Servers: (use tags)**

Define Terraform resources for the frontend servers. Use the specified instance types and number of instances.

Associate the frontend servers with the same subnet and security group as the backend servers.

Apply appropriate tags to distinguish frontend servers from backend servers.

**Set Up S3 Bucket:**

Create an S3 bucket using Terraform with a unique name.

Implement any necessary bucket policies or access controls as per your requirements.

Document the purpose and configuration of the S3 bucket in your documentation.

**Create IAM User:**

Define a Terraform resource for creating an IAM user named "BOB."

Set the description of the user to "CTO."

Ensure that appropriate permissions are assigned to the IAM user based on your organization's policies.

Document the IAM user creation process and the assigned permissions in your documentation.

**Initialize Git and Push to Git:**

Initialize a Git repository in your project directory using git init.

Add all Terraform configuration files to the repository using git add ..

Commit the changes to the repository using git commit -m "Initial commit".

Create a remote repository on a Git hosting service like GitHub, GitLab, or Bitbucket.

Add the remote repository URL using git remote add origin <remote\_repository\_url>.

Push the commits to the remote repository using git push -u origin master.

**Deliverables:**

Terraform configuration files (\*.tf) for creating the specified infrastructure components.

variables.tf file defining the necessary variables.

terraform.tfvars file setting the values for the variables.

Documentation detailing the infrastructure design, including networking setup, server provisioning, S3 bucket configuration, and IAM user creation.

Initialized Git repository containing all Terraform configurations.

Git repository URL for sharing.

**Submission Guidelines:**

Submit all Terraform configuration files, documentation, and Git repository URL through the provided platform or repository.

Ensure that your documentation is comprehensive and well-structured, providing clear explanations of each infrastructure component and its configuration.

**Evaluation Criteria:**

Correctness and completeness of the Terraform configurations.

Proper utilization of variables and adherence to best practices in Terraform usage.

Effective setup of networking components and server provisioning.

Proper configuration of S3 bucket and IAM user creation.

Successful initialization and configuration of Git repository.

Clarity and quality of documentation, including detailed explanations and organization.

USEFUL LINK:

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance>

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/network_interface>

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/security_group>

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/vpc>

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/iam_user>

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/subnet>

<https://github.com/josephifekwe97/Terrafom-Clean-Project.git>