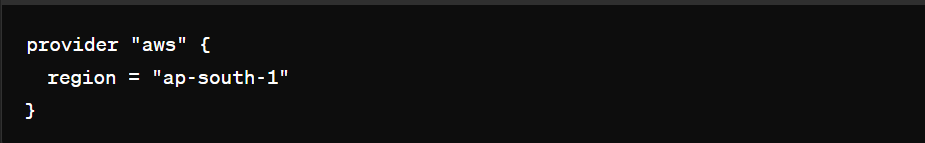
**Project: Secure Secrets Management and AWS Resource Provisioning with Terraform, AWS, and Vault Integration**

**Procedure:**

This Terraform configuration sets up providers for AWS and Vault. It also defines a data source to retrieve a secret from Vault and uses that secret to set tags on an AWS EC2 instance. Let's break down each part of the configuration:

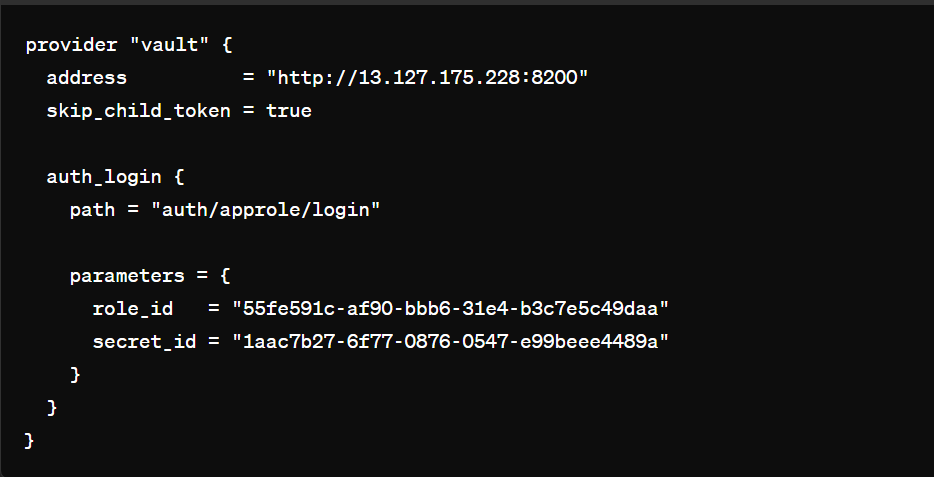
**1. AWS Provider:**

Specifies the AWS region as ‘ap-south-1.’



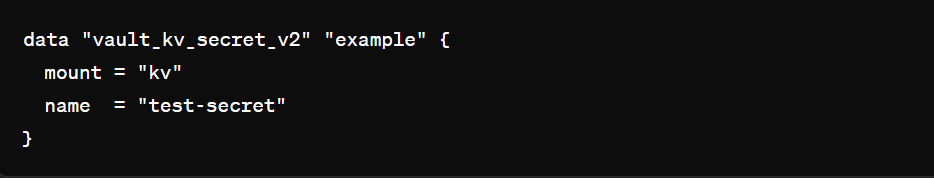
**2. Vault Provider:**

* Specifies the address of the Vault server (http://13.127.175.228:8200).
* Sets skip\_child\_token to true, which skips generating a new token when using Vault's response wrapping feature.
* Configures authentication using the AppRole method (auth/approle/login) with a specific role\_id and secret\_id.



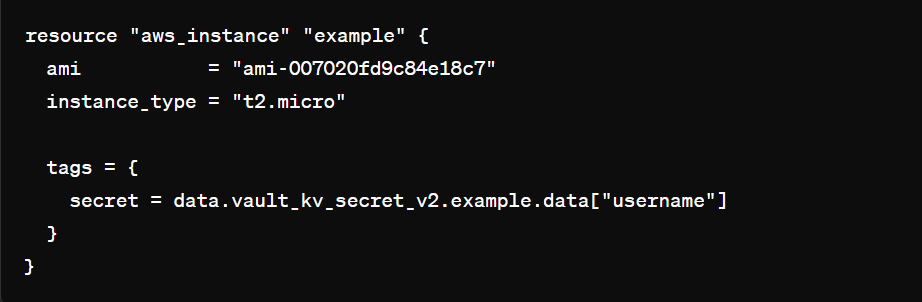
**3. Vault KV Secret Data Source:**

* Retrieves a secret named test-secret from the Vault KV secret engine mounted at ‘kv.’



**4. AWS Instance Resource:**

* Defines an AWS EC2 instance resource using a specific AMI (ami-007020fd9c84e18c7) and instance type (t2.micro).
* Sets tags on the instance, where the secret tag's value is taken from the Vault secret ‘username’



* In this configuration, the secret tag on the AWS instance will be set to the value of the username key retrieved from the test-secret stored in Vault's KV secret engine.

***Make sure that your Vault server is running and accessible at http://13.127.175.228:8200, and that the AppRole authentication method and credentials (role\_id and secret\_id) are correctly configured in your Vault setup. Also, ensure that the test-secret exists in the specified mount path (kv) in your Vault server.***