# Azure Key Vault Module 🔑

This Terraform module deploys an **Azure Key Vault** resource, primarily designed to securely store secrets and SSH keys.

## Features

* Deploys an Azure Key Vault with a dynamically generated name suffix (`kv-secure-xxxxxx`).
* Configures **Soft Delete** with a 7-day retention period.
* Sets an initial **Access Policy** for a specified principal (`object_id`), granting comprehensive permissions for secret management.

## Prerequisites

To use this module, you must have:

1. An Azure Resource Group.
2. Your Azure **Tenant ID**.
3. The **Object ID** (GUID) of the Azure AD user, group, or service principal that requires initial access to the Key Vault secrets.

## Usage

To include this module in your Terraform configuration, add a block similar to the following, replacing the variable values with your specific details:

```terraform
module "secure_key_vault" {
  source             = "./modules/key_vault" # Adjust path as necessary
  location           = "East US"
  resource_group     = "rg-example-vault-01"
  tenant_id          = data.azurerm_client_config.current.tenant_id
  object_id          = azuread_service_principal.my_app.object_id # Example: Object ID of a Service Principal

  # Note: The actual Key Vault name will be dynamically generated as "kv-secure-xxxxxx"
}
```

# Azure Key Vault Security Configuration Module 🔒

This Terraform module manages **secrets** and **access policies** for an **existing** Azure Key Vault. It's designed to separate the creation of the vault itself from the management of its sensitive contents and granular access controls.

## Features

* **Secret Injection:** Creates a specified secret (e.g., a database password) within the target Key Vault.
* **Granular Access Policy:** Grants "Get" access to a specific principal, typically a Virtual Machine's Managed Identity, allowing the application running on the VM to retrieve the secret.

## Prerequisites

To use this module, you must have:

1. An **existing** Azure Key Vault deployed (and its ID).
2. The **Tenant ID** of the Key Vault.
3. The **Object ID** of the Managed Identity (or other principal) that needs access to the secrets.
4. The secret value (e.g., database password) you intend to store.

## Usage

Reference this module in your root configuration, ensuring you pass the required outputs from the Key Vault creation module and the sensitive data.

```terraform
module "kv_security_config" {
  source = "./modules/security" # Adjust path as necessary

  # Required inputs from the Key Vault module output
  key_vault_id = module.key_vault.key_vault_id
  tenant_id    = data.azurerm_client_config.current.tenant_id

  # Secret value (should be passed securely, e.g., from an input variable or local file)
  database_password_secret_value = var.app_db_password

  # Object ID for the principal that needs read access (e.g., a VM's Managed Identity)
  private_vm_object_id = azurerm_managed_identity.vm_identity.principal_id
}
```