Terraform, can you keep a secret?

with Azure Database for postgresql

Meetup OWASP Paris

21/10/2020

There is a better way

Azure Database for Postgresql

- DBaaS (DataBase as a Service)
- Base de données compatible avec Postgresql
- Ressources:
  - Postgresql server (un conteneur logique)
  - Postgresql database
- Directement accessible via internet
  - hostname: ServerName.postgres.database.azure.com
  - port: TCP/5432

There is a better way

Configuration d’un serveur PSQL avec Terraform

```
resource "azurerm_postgresql_server" "main" {
    name                = var.name
    location            = var.azure_location
    resource_group_name = var.resource_group_name

    # …

    administrator_login          = "adm1n157r470r"
    administrator_login_password = "p4ssw0rD"

    version                      = "11"
    ssl_enforcement_enabled          = true
    ssl_minimal_tls_version_enforced = "TLS1_2"
}
```

Database Connection

using PSQL internal account

- Download certificate

```
$ curl -O --location https://www.digicert.com/CACerts/BaltimoreCyberTrustRoot.crt.pem
```

- Define Settings

```
$ export SERVER_NAME=psql-server-XXXX--dev-YYYY
$ export USER_NAME=adm1n157r470r
$ export PGPASSWORD=**************
```

- Connection

```
$ psql "host=${SERVER_NAME}.postgres.database.azure.com \
        sslmode=verify-full \
        sslrootcert=BaltimoreCyberTrustRoot.crt.pem \
        user=${USER_NAME}@${SERVER_NAME} \
        dbname=postgres"
```

There is a better way

Security issue: hard-coded credential (CWE-798)

Solutions:

- Conserver le secret dans un fichier chiffré (SOPS ou Ansible Vault)

```
terraform apply -var-file=<(sops -d secret.tfvars.json)
```

- Conserver le secret dans la CI/CD (Gitlab, Azure DevOps, …)
- Conserver le secret dans un coffre (key Vault, Hashicorp Vault, …)
- Générer un mot de passe aléatoire

There is a better way

Use random password

```
resource "random_password" "admin" {
 length           = 30
 special          = true
}
```

Sensitive value

```
# module.azure_database_postgresql.azurerm_postgresql_server.main will be updated in-place
~ resource "azurerm_postgresql_server" "main" {
      administrator_login                = "adm1n157r470r"

    ~ administrator_login_password       = (sensitive value)

      auto_grow_enabled                  = false
      backup_retention_days              = 7
      create_mode                        = "Default"
      fqdn                               = "psql-server-hw--dev-785.postgres.database.azure.com"
      geo_redundant_backup_enabled       = false
      id                                 =
  }
```

Security issue: credential leak

terraform.tfstate

```
 "resources": [
  {
   "module": "module.azure_database_postgresql",
   "mode": "managed",
   "type": "azurerm_postgresql_server",
   "name": "main",
   "provider": "provider[\"registry.terraform.io/hashicorp/azurerm\"]",
   "instances": [
    {
     "schema_version": 0,
     "attributes": {

      "administrator_login": "adm1n157r470r",
      "administrator_login_password": "#F@IR_fdqsfMz1[9Jhn:04{i-HJ]dJ[i",

      "auto_grow_enabled": false,
      "backup_retention_days": 7,
      "create_mode": "Default",
      "creation_source_server_id": null,
      "fqdn": "psql-server-XXX-dev-YYYY.postgres.database.azure.com",

…
```

There is a better way

Security issue: credential leak

Solutions

Solutions:

- Mettre en place une rotation des secrets
  - non supporté par Azure database for postgresql
- Protéger le fichier State
  - remote backend
  - Chiffrement en transport (TLS1.2) et en stockage
  - Controle d'accès

There is a better way

Security issue: Generic Account

Enable authentication via Azure Active Directory

```
resource azurerm_postgresql_active_directory_administrator main {
   server_name         = azurerm_postgresql_server.main.name
   resource_group_name = var.resource_group_name
   login               = var.psql_server_administrator_name
   tenant_id           = var.azure_tenant_id
   object_id           = var.psql_server_administrator_id
}
```

Les rôles disponibles par défaut: **azure_ad_admin** et **azure_ad_user**

Les comptes utilisateurs internes dans Postgresql sont toujours actifs !

There is a better way

Database Connection

using Azure Active Directory account

- Download certificate

```
$ curl -O --location https://www.digicert.com/CACerts/BaltimoreCyberTrustRoot.crt.pem
```

- Define Settings

```
$ export SERVER_NAME=psql-server-XXXX--dev-YYYY
$ export PGPASSWORD=$(az account get-access-token \
         --resource-type oss-rdbms | jq .accessToken | tr -d '"')
$ export USER_NAME=$(az account list | jq -r '.[] | .user | .name')
```

- Connection

```
 $ psql "host=${SERVER_NAME}.postgres.database.azure.com \
         sslmode=verify-full \
         sslrootcert=BaltimoreCyberTrustRoot.crt.pem \
         user=${USER_NAME}@${SERVER_NAME} \
         dbname=postgres"
```

There is a Better Way