

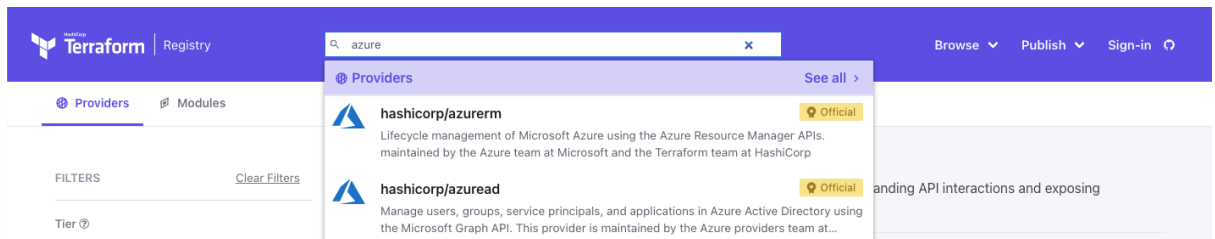
Szkolenie

Terraform: Dzień 1



1. Terraform Provider

- 1.1. W dokumentacji Terraform, można znaleźć informacje na temat dostępnych providerów: <https://registry.terraform.io/browse/providers>
- 1.2. Podczas tego kursu będziemy pracować z providerami Azure (azurerm) oraz Azure Active Directory (azuread). Ich dokumentację można łatwo znaleźć wpisując w wyszukiwarkę hasło „azure”. Rekomendujemy zostawić otworzoną dokumentację dla obu providerów podczas pracy z kodem.



- 1.3. Stwórz folder do wykorzystania w dalszych krokach tego ćwiczenia.
- 1.4. Należy utworzyć plik „providers.tf” i zarejestrować oba wyżej wymienione providery zgodnie z dokumentacją w najnowszej wersji. ID subskrypcji oraz ID tenanta Azure AD można znaleźć w portalu Azure. W dokumentacji providerów znajduje się instrukcja jak prawidłowo się zalogować, aby móc korzystać z providera. Do edytowania plików polecany jest program Visual Studio Code z odpowiednimi rozszerzeniami Terraform.

UWAGA: zwróć uwagę na ID subskrypcji oraz ID tenanta Azure AD. Pamiętaj, że masz uprawnienia tylko w wybranej subskrypcji i nie widzisz wszystkich.

Przykładowa konfiguracja:

```
1 terraform {
2   required_providers {
3     azurerm = {
4       source = "hashicorp/azurerm"
5       version = "~> 3.27.0"
6     }
7     azuread = {
8       source = "hashicorp/azuread"
9       version = "~> 2.29.0"
10    }
11  }
12 }
13
14 # Configure the Microsoft Azure Provider
15 provider "azurerm" {
16   features {}
17   subscription_id = "79283b62-f23b-4420-9ae7-1ac41de00335"
18 }
19
20 # Configure the Azure Active Directory Provider
21 provider "azuread" {
22   tenant_id = "3a81269f-0731-42d7-9911-a8e9202fa750"
23 }
```

- 1.5. Dokonaj inicjalizacji providerów za pomocą komendy *terraform init*.
- 1.6. Przy pomocy providera Azure (azurerm) należy utworzyć zasób Storage Account należący do gotowej grupy zasobów (widocznej w portalu) oraz wypisać data source dotyczący własnego użytkownika (wypisać przy pomocy output). Metodę tworzenia zasobów można znaleźć w dokumentacji bazując na nazwie zasobu np. „Storage Account” lub „User”. Należy również utworzyć data source reprezentujący grupę zasobów znajdującą się w Azure.

Podczas wszystkich zadań w czasie szkolenia, będziemy korzystać z tej grupy zasobów i stosować referencję do niej z wykorzystaniem data source.

Dokumentacja Storage Account:

https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/storage_account

Dokumentacja User:

<https://registry.terraform.io/providers/hashicorp/azuread/latest/docs/data-sources/user>

2. Terraform Stan:

- 2.1. Po utworzeniu zasobów w poprzednim zadaniu należy przejrzeć plik *terraform.tfstate* i zwrócić uwagę na utworzony zasób oraz output.
- 2.2. Wykonać ponownie *terraform apply* i zobaczyć jak wygląda wynik wywołania komendy bez zmian w plikach terraform. *
- 2.3. Zmienić ręcznie wartości dla output - nie powinno się ręcznie wykonywać zmian w pliku stanu terraform, jest to jedynie wykonywane w celu lepszego zapoznania się z działaniem stanu. *
- 2.4. Ponownie wykonać *terraform apply* i porównać wynik wywołania komendy z wynikiem bez wprowadzania zmian. *
- 2.5. Należy zmienić w pliku stanu nazwę utworzonego Storage Account. *
- 2.6. Ponownie wykonać komendę *terraform apply* i porównać wynik komendy z wynikiem bez wprowadzanie zmian. *
- 2.7. Należy zmienić w pliku stanu ID utworzonego Storage Account. *
- 2.8. Ponownie wykonać komendę *terraform apply* i porównać wynik komendy z wynikiem bez wprowadzanie zmian. *

* - zadania dla chętnych

3. Terraform Importowanie:

3.1. W nowym katalogu utwórz plik „providers.tf” zarejestruj i skonfiguruj provider „azurerm”. Przygotuj plik „data.tf” i umieść w nim przy pomocy data source reprezentację przypisanej do Ciebie grupy zasobów znajdującej się w Azure.

3.2. Utworzyć zasób Storage Account w portalu Azure.

Dokumentacja storage account:

https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/storage_account

3.3. Utworzyć reprezentację storage account w pliku Terraform (kod).

3.4. Zaimportować utworzony Storage Account (*terraform import*).

3.5. Sprawdzić zmiany w pliku stanu po zaimportowaniu zasobu.

3.6. Wpisać komendę *terraform plan* w celu sprawdzenia zgodności opisanego zasobu z znajdującym się w chmurze

4. Zmienne na różne sposoby:

W zadaniu będzie tworzona maszyna wirtualna wykorzystująca logowanie przy pomocy nazwy użytkownika oraz hasła – niektóre zmienne muszą być traktowane w inny sposób zapewniający więcej bezpieczeństwa.

- 4.1. W nowym katalogu utwórz plik „providers.tf” zarejestruj i skonfiguruj provider „azurerm”. Przygotuj plik „data.tf” i umieść w nim przy pomocy data source reprezentację przypisaną do Ciebie grupy zasobów znajdującej się w Azure.
- 4.2. Zapoznaj się z dokumentacją do tworzenia maszyn wirtualnych Linux providera azurerm: https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/linux_virtual_machine
- 4.3. Utworzenie pliku ze zmiennymi „variables.tf” zawierającego takie zmienne jak:
 - ID Subskrypcji
 - Nazwa użytkownika – do maszyny wirtualnej
 - Hasło użytkownika – do maszyny wirtualnej
 - Tablicę tagów wykorzystywaną do utworzonych zasobów
- 4.4. Utworzenie maszyny wirtualnej Linux wykorzystującej SKU „Standard_B1s”
- 4.5. Należy wykorzystać zmienne zapisując wyłącznie ich definicje w pliku.
- 4.6. Należy wykorzystać zmienne zapisując ich definicje w pliku tfvars automatycznie wczytywanym przez terraform np. nazwa „terraform.tfvars”.
- 4.7. Należy zapisać zmienne „Nazwa użytkownika” oraz „Hasło użytkownika” przy pomocy zmiennych środowiskowych.

Pomocne komendy:

```
export TF_VAR_username=(the username)
export TF_VAR_password=(the password)
```

5. Wyrażenia i funkcje:

- 5.1. W nowym katalogu utwórz plik „providers.tf” zarejestruj i skonfiguruj provider „azurerm”. Przygotuj plik „data.tf” i umieść w nim przy pomocy data source reprezentację przypisaną do Ciebie grupy zasobów znajdującą się w Azure.
- 5.2. Należy utworzyć App Service Plan.
- 5.3. Należy utworzyć zmienną zawierającą 3 nazwy aplikacji (dowolne, ale nie mogą się powtarzać globalnie).
- 5.4. Utworzyć 3 razy app service wykorzystując nazwy aplikacji w zmiennej, należy dokonać tego przy pomocy wyrażenia *for_each*. Dla jednej wybranej nazwy wersja *dotnet_framework* powinna różnić się względem innych, można to osiągnąć przy pomocy wyrażenia warunkowego. Tworzony zasób powinien zawierać tag *creation_date*, którego wartość jest równa czasowi wykonywania skryptu (funkcje *formatdate* oraz *timestamp*). Należy również ignorować zmiany w tagach oraz parametrze *app_settings* (terraform lifecycle).

6. Przechowywanie stanu w Azure Blob Storage:

- 6.1. W nowym katalogu utwórz plik „providers.tf” zarejestruj i skonfiguruj provider „azurerm”. Przygotuj plik „data.tf” i umieść w nim przy pomocy data source reprezentację przypisaną do Ciebie grupy zasobów znajdującej się w Azure.
- 6.2. Należy utworzyć Azure Blob Storage (nie powinno być zarządzane przez terraform).
- 6.3. W Azure Blob Storage należy utworzyć kontener.
- 6.4. Należy wykonać komendę „terraform destroy” dla istniejącej infrastruktury.
- 6.5. Należy utworzyć provider zawierający backend z opisanym dostępem do Azure Blob Storage.
- 6.6. Utworzyć infrastrukturę korzystając z providera przechowującego stan w Azure Blob Storage

Pomocne linki:

<https://learn.microsoft.com/en-us/azure/developer/terraform/store-state-in-azure-storage?tabs=azure-cli>

<https://www.terraform.io/language/settings/backends/configuration>