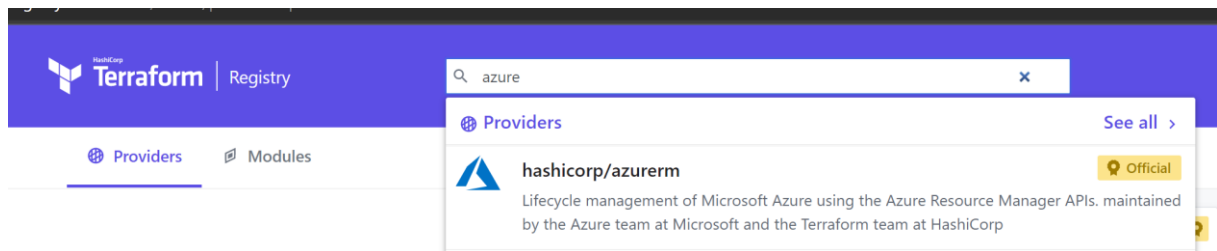


## 1. Terraform Providery:

1) Początkowo należy wejść na stronę i zapoznać się z dostępnymi providerami

<https://registry.terraform.io/browse/providers>

2) W wyszukiwarce należy wpisać nazwę azure oraz bliżej zapoznać się z providerami Azure oraz Azure Active Directory. Dobrze jest zostawić 2 otwarte okna przeglądarki jedno na każdą z dokumentacji.



3) Należy utworzyć folder wykorzystywany w dalszych krokach ćwiczeń

4) Należy utworzyć plik „providers.tf” i zarejestrować oba wyżej wymienione providery zgodnie z dokumentacją w najnowszej wersji. Aby utworzyć plik należy początkowo przejść do portalu Azure i znaleźć w nim ID subskrypcji i tenanta. 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:

```
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~> 2.77.0"
    }
    azuread = {
      source  = "hashicorp/azuread"
      version = "~> 2.3.0"
    }
  }
}

# Configure the Microsoft Azure Provider
provider "azurerm" {
  features {}
  subscription_id = "72ae933f-dab5-4a35-861e-1db14caba401"
```

```
}  
  
# Configure the Azure Active Directory Provider  
provider "azuread" {  
  tenant_id = "16dd9b56-84a8-4b26-b800-741b14debe2f"  
}
```

5) Po prawidłowej inicjalizacji providerów, należy utworzyć przy pomocy providera Azure zasób Storage Account należący do gotowej grupy zasobów (widocznej w portalu) oraz wypisać Data Sources dotyczący własnego użytkownika. 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.

Początkowo należy skorzystać z nazwy grupy zasobów oraz lokalizacji dostępnej w Portalu Azure.

Dokumentacja Storage Account:

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

- 1) Po utworzeniu zasobów w poprzednim zadaniu należy przejrzeć plik terraform.tfstate w szczególności zwrócić uwagę na utworzony zasób i output.
- 2) Wykonać ponownie terraform apply i zobaczyć jak wygląda wynik wywołania komendy bez zmian w plikach terraform.
- 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.
- 4) Ponownie wykonać terraform apply i porównać wynik wywołania komendy z wynikiem bez wprowadzania zmian.
- 5) Należy zmienić w pliku stanu nazwę utworzonego Storage Account
- 6) Ponownie wykonać komendę terraform apply i porównać wynik komendy z wynikiem bez wprowadzanie zmian
- 7) Należy zmienić w pliku stanu ID utworzonego Storage Account

- 8) Ponownie wykonać komendę terraform apply i porównać wynik komendy z wynikiem bez wprowadzanie zmian

### 3. Terraform Importowanie:

- 1) Utworzyć zasób Storage Account w portalu Azure

Dokumentacja storage account:

[https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/storage\\_account](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/storage_account)

- 2) Utworzyć reprezentację storage account w pliku Terraform (kod)
- 3) Zaimportować utworzony Storage Account (polecenie terraform import)
- 4) Sprawdzić zmiany w pliku stanu po zaimportowaniu zasobu
- 5) 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.

- 1) Zapoznanie się z dokumentacją do tworzenia maszyn wirtualnych Linux providera azurerm

[https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/linux\\_virtual\\_machine](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/linux_virtual_machine)

- 2) Utworzenie pliku z 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
- 3) Utworzenie maszyny wirtualnej Linux wykorzystującej SKU “Standard\_B1s”
- 4) Należy wykorzystać zmienne zapisując wyłącznie ich definicje w pliku
- 5) Należy wykorzystać zmienne zapisując ich definicje w pliku tfvars automatycznie wczytywanym przez terraform np. nazwa „terraform.tfvars”

- 6) 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:

- 1) Należy utworzyć App Service Plan
- 2) Należy utworzyć zmienną zawierającą 3 nazwy aplikacji (dowolne ale nie mogą się powtarzać globalnie)
- 3) 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 wyrażeń warunkowych. 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:

- 1) Należy utworzyć Azure Blob Storage (nie powinno być zarządzane przez terraform)
- 2) W Azure Blob Storage należy utworzyć kontener
- 3) Należy wykonać komendę „`terraform destroy`” dla istniejącej infrastruktury
- 4) Należy utworzyć provider zawierający backend z opisanym dostępem do Azure Blob Storage
- 5) Utworzyć infrastrukturę korzystając z providera przechowującego stan w Azure Blob Storage

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