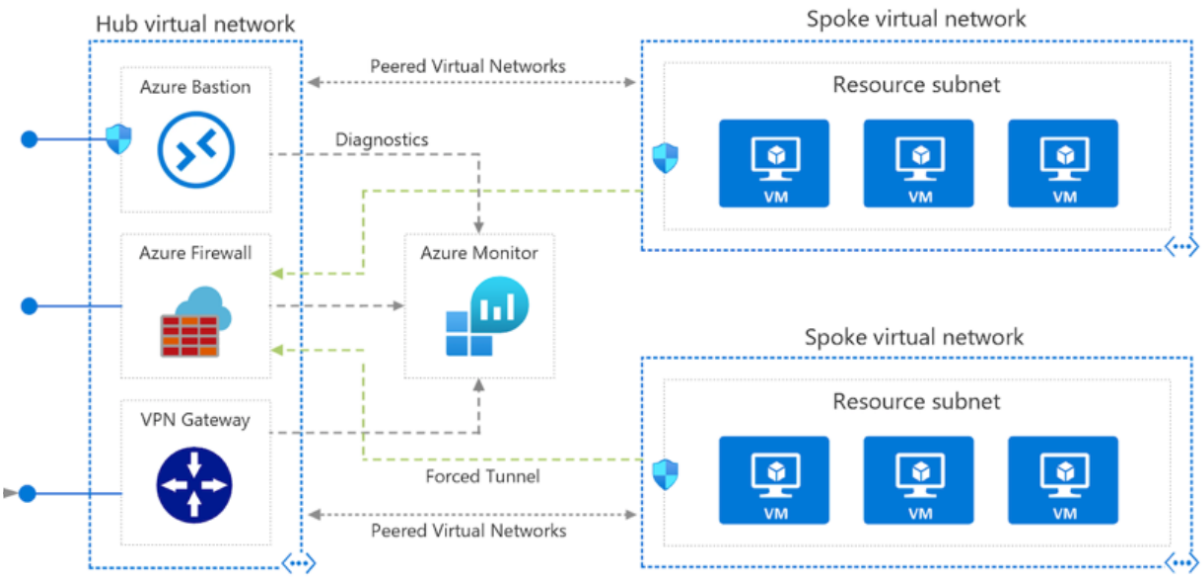
Chmurowisko

Szkolenie Terraform: Dzień 2

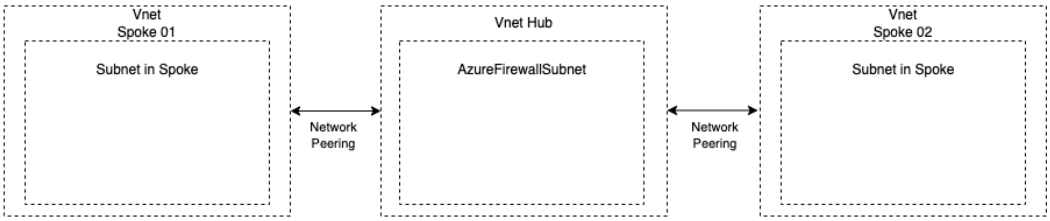
1. Zadanie: Budowa sieci Hub & Spoke

Przykładowa architektura rozwiązania zbudowanego w sieci Hub & Spoke widoczna jest na Rys. *1*.



Rys. 1 Przykładowa architektura sieci Hub & Spoke

Architektura sieciowa oraz zasoby tworzone w ramach zadania 1 są widoczne na Rys. 2.



Rys. 2 Architektura sieciowa Zad 1.

Opis zadania: Należy zbudować strukturę sieciową przedstawioną na Rys. 2.

Należy utworzyć trzy sieci virtualne (nie kolidujące adresacją) oraz w każdej z nich utworzyć 1 subnet po czym utworzyć peering między sieciami spoke a siecią hub zgodnie z rysunkiem.

Peering należy utworzyć w obie strony (w stronę sieci Spoke i w stronę sieci Hub). Podczas tworzenia peeringu należy nadać wartość „true" w parametrach: allow_virtual_network_access oraz allow_forwarded_traffic.

UWAGA: W podsieci w sieci VNet Hub, będzie tworzony Azure Firewall więc musi nazywać się „AzureFirewallSubnet" oraz posiadać maskę /26 np. 10.0.0.0/26.

Przykładowa adresacja sieci:

- Vnet Hub – 10.0.0.0/16
- Vnet Spoke 1 – 10.1.0.0/16
- Vnet Spoke 2 – 10.2.0.0/16
- Subnet in Hub – 10.0.0.0/26 (musi nazywać się „AzureFirewallSubnet”).
- Subnet in Spoke 1 – 10.1.0.0/24
- Subnet in Spoke 2 – 10.2.0.0/24

Zasoby które należy utworzyć:

- Sieć wirtualna Vnet Hub
- Sieć wirtualna Vnet Spoke 1
- Sieć wirtualna Vnet Spoke 2
- Subnet w sieci wirtualnej Vnet Hub
- Subnet w sieci wirtualnej Vnet Spoke 1
- Subnet w sieci wirtualnej Vnet Spoke 2
- Peering z sieci Vnet Hub do sieci Vnet Spoke 1
- Peering z sieci Vnet Spoke 1 do sieci Vnet Hub
- Peering z sieci Vnet Hub do sieci Vnet Spoke 2
- Peering z sieci Vnet Spoke 2 do sieci Vnet Hub

Przydatne linki:
Terraform dokumentacja sieci wirtualnych
https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/virtual_network

Terraform dokumentacja subnetów
https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/subnet

Terraform dokumentacja peeringu
https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/virtual_network_peering

Przykładowe rozwiązanie znajduje się w folderze „1-HubAndSpoke”

2. Zadanie: Ustawienia diagnostyczne

W tym zadaniu wykorzystaj kod utworzonych zasobów podczas zadań z dnia 2.

Na początku należy utworzyć Log Analytics Workspace.

Następnie należy włączyć wysyłanie danych diagnostycznych z np. Storage Account do Log Analytics Workspace wykorzystując zasób Diagnostic Settings. Przy pomocy data „monitor_diagnostic_categories" możesz pobrać dostępne kategorie metryk i logów dla danego zasobu np. Storage Account.

Przy tworzeniu Diagnostic Settings wykorzystaj blok dynamiczny do zdefiniowania kategorii logów. Blok dynamiczny może korzystać z zmiennej wykorzystaj tutaj mapę obiektów, gdzie zdefiniujesz w pojedynczym obiekcie nazwę kategorii oraz to czy jest ona włączona.

Przydatne linki:
Terraform dokumentacja diagnostic settings
https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/monitor_diagnostic_setting

Terraform dokumentacja data monitor categories
https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/data-sources/monitor_diagnostic_categories

Terraform dokumentacja dynamic blocks
https://www.terraform.io/language/expressions/dynamic-blocks

3. Zadanie: Wyrażenia i funkcje:

3.1. Należy utworzyć App Service Plan.

3.2. Należy utworzyć zmienną zawierającą 3 nazwy aplikacji (dowolne, ale nie mogą się powtarzać globalnie).

3.3. 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).

4. Zadanie: Wykorzystywanie gotowych modułów

Przygotuj nowy projekt terraform, skonfiguruj provider Azurerm i przygotuj data source pobierający Twoją grupę zasobów.

W następnym kroku utwórz sieć wirtualną i trzy podsieci z wykorzystaniem modułu z terraform registry: https://registry.terraform.io/modules/Azure/network/azurerm/latest

Po utworzeniu sieci, utwórz maszynę wirtualną (Ubuntu) wykorzystując moduł: https://registry.terraform.io/modules/Azure/compute/azurerm/latest. Podczas tworzenia wskaż id podsieci stworzonej w module 1 (sprawdź jakie wyjścia są w module tworzącym sieć).

Podczas tworzenia VM wykorzystaj autentykacji przy pomocy hasła i ustaw rozmiar „Standard_D2as_v5". Wyszukaj w dokumentacji jak ustawić te parametry.

Sprawdź utworzone moduły z poziomu portalu Azure.

5. Zadanie: Moduł VM

Przygotuj moduł, który tworzy maszynę wirtualną, interfejs sieciowy (wraz z opcjonalnym publicznym adresem ip). Dodaj odpowiedni alias dla providera azure w konfiguracji modułu np. „virtual-machine”.

Moduł powinien tworzyć takie zasoby:

- Publiczny adres ip (jeśli odpowiednia zmienna będzie przyjmować wartość true),
- Interfejs sieciowy
- Maszyna wirtualna linux (logowanie przy pomocy loginu i hasła)

Moduł powinien przyjmować takie wejścia (wraz z odpowiednim typowaniem):

- Use_public_ip
- Resource_group_name
- Location
- Admin_username
- Admin_password
- Virtual_machine_name
- Subnet_id
- Source_image (obiekt zawierający dane na temat wykorzystywanego obrazu systemu).

Na wyjściu z modułu ustaw:

- ID Maszyny wirtualnej
- Publiczny adres IP maszyny
- Prywatny adres IP maszyny

W przypadku zmiennej wejściowej przechowującej dane na temat systemu operacyjnego ustaw domyślną wartość z Ubuntu 16.04.

Jeśli jakieś zmienne nie są podane, przyjmij stałe wartości np. rozmiar maszyny.

Po przygotowaniu modułu, przetestuj go tworząc dwie maszyny wirtualne z jego wykorzystaniem. Tylko jedna maszyna powinna posiadać publiczny adres IP.