

Infrastructure as Code

Konfiguracja, graf zależności, plan i state

mgr inż. Jakub Woźniak

Zarządzanie Systemami Rozproszonymi · Semestr zimowy 2026/27

Element	Przykład	Rola
Konfiguracja	name = „orders”	Intencja zespołu
State	Adres → zdalne ID	Powiązanie obiektu z deklaracją
API dostawcy	Bieżąca kolejka i właściwości	Odczyt rzeczywistości przez provider

Konfiguracja + state + odczyt API → Plan → Apply

Terraform — State

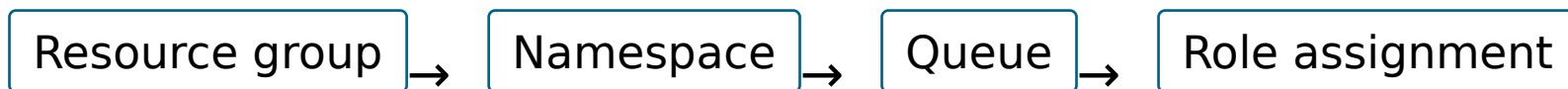
```
terraform {  
  required_providers {  
    azurearm = {  
      source  = "hashicorp/azurearm"  
      version = "~> 4.0"  
    }  
  }  
}  
provider "azurearm" {  
  features {}  
}
```

- Ograniczenie dopuszcza serię 4.x; plik lock wybiera konkretną wersję providera.
- Uwierzytelnienie i subskrypcja dostarczone przez środowisko.

Terraform — Dependency lock file

```
resource "azurerm_servicebus_queue" "orders" {  
  name                = "orders"  
  namespace_id        = azurerm_servicebus_namespace.main.id  
  max_delivery_count = 10  
}  
  
output "queue_id" {  
  value = azurerm_servicebus_queue.orders.id  
}
```

- Fragment wymaga istniejącej deklaracji namespace main.
- Adres w konfiguracji: azurerm_servicebus_queue.orders.
- Zdalne ID: identyfikator kolejki zwrócony przez Azure.



- Odwołanie do `.id` tworzy zależność; kolejność tekstu w pliku jej nie zastępuje.
- Niezależne zasoby mogą być obsługiwane równolegle.
- Kolejka utworzona przed nadaniem roli nie oznacza natychmiastowej gotowości całej aplikacji.

Terraform — Resource dependencies

```
terraform init
terraform fmt -check
terraform validate
terraform plan -out=reviewed.tfplan
terraform show reviewed.tfplan
terraform apply reviewed.tfplan
```

- `validate`: poprawność konfiguracji, nie gwarancja dostępności zasobów w chmurze.
- Zapisany plan: konkretny przedmiot przeglądu; może zawierać dane wrażliwe.

Terraform — plan

```
# Skrócony, przykładowy wynik planu
~ resource "azurerm_servicebus_queue" "orders" {
  name          = "orders"
  ~ max_delivery_count = 10 -> 5
}
```

Plan: 0 to add, 1 to change, 0 to destroy.

- ~: proponowana aktualizacja bez zastąpienia w tym planie.
- Mniej prób dostarczenia → wcześniejsze przeniesienie problematycznej wiadomości do DLQ.
- Brak destroy nie oznacza braku wpływu na zachowanie usługi.

```
azurerm_servicebus_queue.orders  
-> /subscriptions/.../namespaces/shop/queues/orders
```

- Usunięcie lokalnego state nie usuwa automatycznie kolejki w chmurze.
- Utrata mapowania utrudnia zarządzanie istniejącym zasobem; możliwy konflikt tworzenia.
- Odtworzenie state lub import wymaga sprawdzenia zgodności z konfiguracją.

Terraform — State purpose

```
# Zmiana etykiety bloku: orders -> incoming
resource "azurerm_servicebus_queue" "incoming" {
  name          = "orders"
  namespace_id = azurerm_servicebus_namespace.main.id
}
moved {
  from = azurerm_servicebus_queue.orders
  to   = azurerm_servicebus_queue.incoming
}
```

- Nazwa zdalnej kolejki pozostaje orders.
- moved zachowuje powiązanie; pozostałe właściwości także muszą być zgodne.

```
# Skróć planu dla zmiany właściwości name
-/+ resource "azurerm_servicebus_queue" "orders" {
  ~ name = "orders" -> "orders-v2" # forces replacement
}
```

Plan: 1 to add, 0 to change, 1 to destroy.

- W tym zasobie provider wymaga zastąpienia przy zmianie name.
- moved nie migruje wiadomości między dwiema kolejkami.

AzureRM — Service Bus Queue: name

```
resource "azurerm_role_assignment" "receiver" {  
  scope          = azurerm_servicebus_queue.orders.id  
  role_definition_name = "Azure Service Bus Data Receiver"  
  principal_id    = var.worker_principal_id  
}
```

- Fragment: tożsamość workera utworzona osobno; input to jej principal ID.
- Zakres na konkretnej kolejce; brak hasła w deklaracji.
- Nadanie roli wymaga uprawnienia operatora IaC do zarządzania dostępem.

Microsoft — Service Bus managed identities

Queue: utworzona



Rola: błąd uprawnień



Apply przerwany

- Część działań może zostać wykonana przed błędem.
- Sprawdzić stan zdalny i state; poprawić przyczynę, wygenerować nowy plan.
- Nie zakładać automatycznego cofnięcia wszystkich zasobów.

Terraform — apply

Zmiana A	Zmiana B
Etykieta bloku orders → incoming	Właściwość name: orders → orders-v2
Zdalna nazwa bez zmian	Zdalna nazwa inna
Cel: czytelniejszy kod	Cel: nowa kolejka produkcyjna

Przegląd

Gdzie użyć moved? Kiedy potrzebna migracja wiadomości i odbiorców? Jakie linie planu wymagają zatrzymania zmiany?

- Terraform — State
- Terraform — plan
- Terraform — Dependency lock file
- Terraform — Refactor modules
- AzureRM — Service Bus Queue
- Terraform — apply

Dokumentacja sprawdzona: 26.09.2026. Dane przypadków: przykłady dydaktyczne.