

Kierunek: IST

ZESPOŁOWE PRZEDSIĘWZIĘCIE INFORMATYCZNE

Scan and Store System do Zarządzania Inwentarzem z Użyciem Kodów Kreskowych i Kodów QR

Adam Klementowski
Adam Rudnicki
Adam Skowron
Wojciech Skrzypiec

Opiekun pracy: dr inż. Rafał Palak

SPIS TREŚCI

1	Wykaz symboli, oznaczeń i akronimów (opcja)	3
1.1	Symbole	3
1.2	Oznaczenia	3
1.3	Akronimy	3
2	Cel i zakres przedsięwzięcia	4
2.1	Cel	4
2.2	Zakres	4
3	Słownik pojęć (opcja)	5
4	Stan wiedzy w obszarze przedsięwzięcia (opcja)	6
5	Założenia wstępne	7
5.1	Dobór technologii	7
5.2	Ograniczenia	7
6	Specyfikacja wymagań na produkt programowy	8
6.1	Baza danych	8
6.2	Historyjki	8
7	Projekt produktu programowego	10
7.1	Baza danych	10
7.2	Architektura	10
7.3	Pozostałe schematy i diagramy	10
8	Implementacja (opcja)	12
8.1	Warstwa użytkownika - frontend	12
8.1.1	Struktura plików	12
8.1.2	Widoki	12
8.1.3	Routing	12
8.1.4	Testy	12
8.1.5	Konteneryzacja - Docker	12
8.2	Warstwa systemowa - backend	13
8.2.1	Struktura plików	13
8.2.2	Konfiguracja	13
8.2.3	Kontrolery	13
8.2.4	Inne	13
8.2.5	Testy	13
8.2.6	Konteneryzacja - Docker	13
8.3	Architektura chmurowa	14
8.3.1	Wstęp	14
8.3.2	Struktura plików	14
8.3.3	Wykorzystane zasoby	15
9	Demonstracja produktu programowego	16

1 WYKAZ SYMBOLI, OZNACZEŃ I AKRONIMÓW (OPCJA)

1.1 Symbole

1.2 Oznaczenia

1.3 Akronimy

- SNS - Scan and Store, Nazwa robocza projektu
- AWS - Amazon Web Services
- PWA - Progressive Web App

2 CEL I ZAKRES PRZEDSIĘWZIĘCIA

2.1 Cel

Celem przedsięwzięcia jest zbadanie dojrzałości technologii PWA na przykładzie systemu do zarządzania inwentarzem z użyciem kodów kreskowych i kodów QR.

2.2 Zakres

W fazie planowania projektu ustalono, iż aplikacja końcowa powinna korzystać z funkcjonalności oferowanych przez technologię PWA oraz umożliwiać zarządzanie stanem magazynu w podstawowym zakresie.

Z funkcjonalności PWA jakie powinny zostać zaimplementowane wybrano dostęp do kamery urządzenia w celu skanowania kodów kreskowych i kodów QR oraz dostęp do geolokacji.

3 SŁOWNIK POJĘĆ (OPCJA)

- framework - biblioteka narzędzi wspomagających implementację danego rozwiązania w danym języku
- API - interfejs programowy aplikacji (ang. application programming interface) - zestaw poleceń udostępnianych osobom korzystającym z danej aplikacji
- REST - reprezentacyjny transfer stanu (ang. representational state transfer) - zbiór zasad i wytycznych o tym jak budować API aplikacji webowych
- REST API - API trzymające się zasad REST

4 STAN WIEDZY W OBSZARZE PRZEDSIĘWZIĘCIA (OPCJA)

5 ZAŁOŻENIA WSTĘPNE

5.1 Dobór technologii

Najważniejszą technologią dla projektu jest **Progressive Web App** (PWA).

Aby wybrać framework do implementacji warstwy użytkownika (frontendu) zwrócono szczególną uwagę na dobre wsparcie dla technologii PWA i wybrano framework **Angular**.

Warstwę systemową (backend) zaimplementowano przy użyciu frameworku **Spring Boot**, który ma duże wsparcie dla zabezpieczeń (Spring Security) oraz prosty do zaimplementowania zestaw punktów końcowych (endpointów) jako REST API.

Jako bazę danych wybrano **PostgreSQL**.

Całość architektury aplikacji została wdrożona przy pomocy **Terraform HashiCorp** na serwery **Amazon Web Services** (AWS), co niskim kosztem pozwala na przetestowanie różnych rozwiązań bez konieczności samodzielnego utrzymania infrastruktury.

5.2 Ograniczenia

W fazie wstępnej planowania projektu ustalone zostały następujące założenia:

Klient

- Aplikacja wytwarzana jest dla jednego klienta
- Klient posiada wiele magazynów
- Klient chce być w stanie sprawdzić stan każdego z magazynów
- Klient chce dać swoim pracownikom możliwość dodawania artykułów do stanu magazynu
- Klient chce mieć wyłączny dostęp do dodawania i modyfikowania artykułów, magazynów i kategorii

Aplikacja

- Aplikacja musi wspierać technologię PWA
- Aplikacja musi być instalowalna na różnych urządzeniach
- Aplikacja musi wspierać skanowanie kodów kreskowych i kodów QR
- Aplikacja musi wspierać autoryzację użytkowników

Inne

- Infrastruktura musi być skalowalna

6 SPECYFIKACJA WYMAGAŃ NA PRODUKT PROGRAMOWY

6.1 Baza danych

- Magazyn posiada nazwę i adres: państwo, miasto, ulica, kod pocztowy oraz opcjonalnie numer domu
- Magazyn może posiadać opis
- Każdy magazyn może mieć artykuły
- Artykuły posiadają nazwę i kategorię
- Artykuły mogą posiadać zdjęcie, opis i kod EAN13
- Kategorie posiadają nazwę
- Kategorie mogą posiadać opis
- Kategorie mogą być przypisane do więcej niż jednego artykułu
- Jeden artykuł może mieć nie więcej niż jedną kategorię
- Artykuł może się znajdować w wielu magazynach w różnych ilościach

6.2 Historyjki

Użytkownik

- Jako użytkownik chcę mieć możliwość zalogowania się na swoje konto
- Jako użytkownik chcę mieć możliwość sprawdzenia listy przedmiotów
- Jako użytkownik chcę mieć możliwość sprawdzenia listy kategorii
- Jako użytkownik chcę mieć możliwość sprawdzenia listy magazynów
- Jako użytkownik chcę mieć możliwość sprawdzenia listy przedmiotów w magazynie
- Jako użytkownik chcę mieć możliwość dodania istniejącego przedmiotu do magazynu
- Jako użytkownik chcę mieć możliwość zwiększenia liczby przedmiotów na stanie magazynu
- Jako użytkownik chcę mieć możliwość zmniejszenia liczby przedmiotów na stanie magazynu
- Jako użytkownik chcę mieć możliwość skanowania kodów kreskowych
- Jako użytkownik chcę mieć możliwość skanowania kodów QR
- Jako użytkownik chcę mieć możliwość wyboru domyślnego magazynu

Administrator

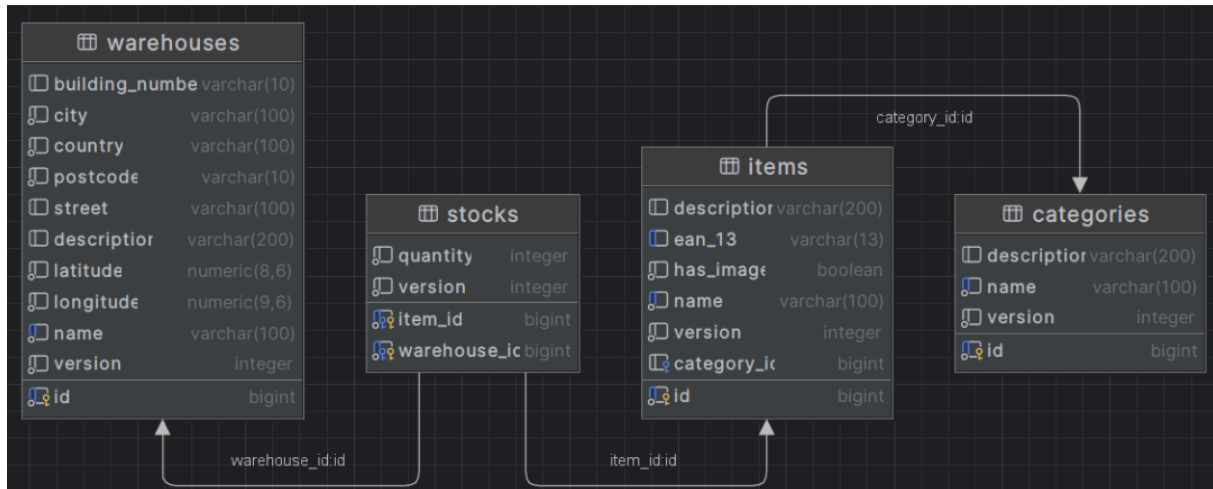
- Jako administrator chcę mieć możliwość dodania nowego przedmiotu
- Jako administrator chcę mieć możliwość dodania nowej kategorii
- Jako administrator chcę mieć możliwość dodania nowego magazynu
- Jako administrator chcę mieć możliwość dodania istniejącego przedmiotu do magazynu
- Jako administrator chcę mieć możliwość modyfikacji istniejącego przedmiotu
- Jako administrator chcę mieć możliwość modyfikacji istniejącej kategorii
- Jako administrator chcę mieć możliwość modyfikacji istniejącego magazynu
- Jako administrator chcę mieć możliwość zwiększenia liczby przedmiotów na stanie magazynu
- Jako administrator chcę mieć możliwość zmniejszenia liczby przedmiotów na stanie magazynu
- Jako administrator chcę mieć możliwość usunięcia istniejącego przedmiotu

- Jako administrator chcę mieć możliwość usunięcia istniejącej kategorii
- Jako administrator chcę mieć możliwość usunięcia istniejącego magazynu
- Jako administrator chcę mieć możliwość usunięcia istniejącego przedmiotu z magazynu

7 PROJEKT PRODUKTU PROGRAMOWEGO

7.1 Baza danych

Baza danych została zaimplementowana w PostgreSQL według diagramu na Rysunku 1.



Rysunek 1: Diagram fizyczny bazy danych

7.2 Architektura

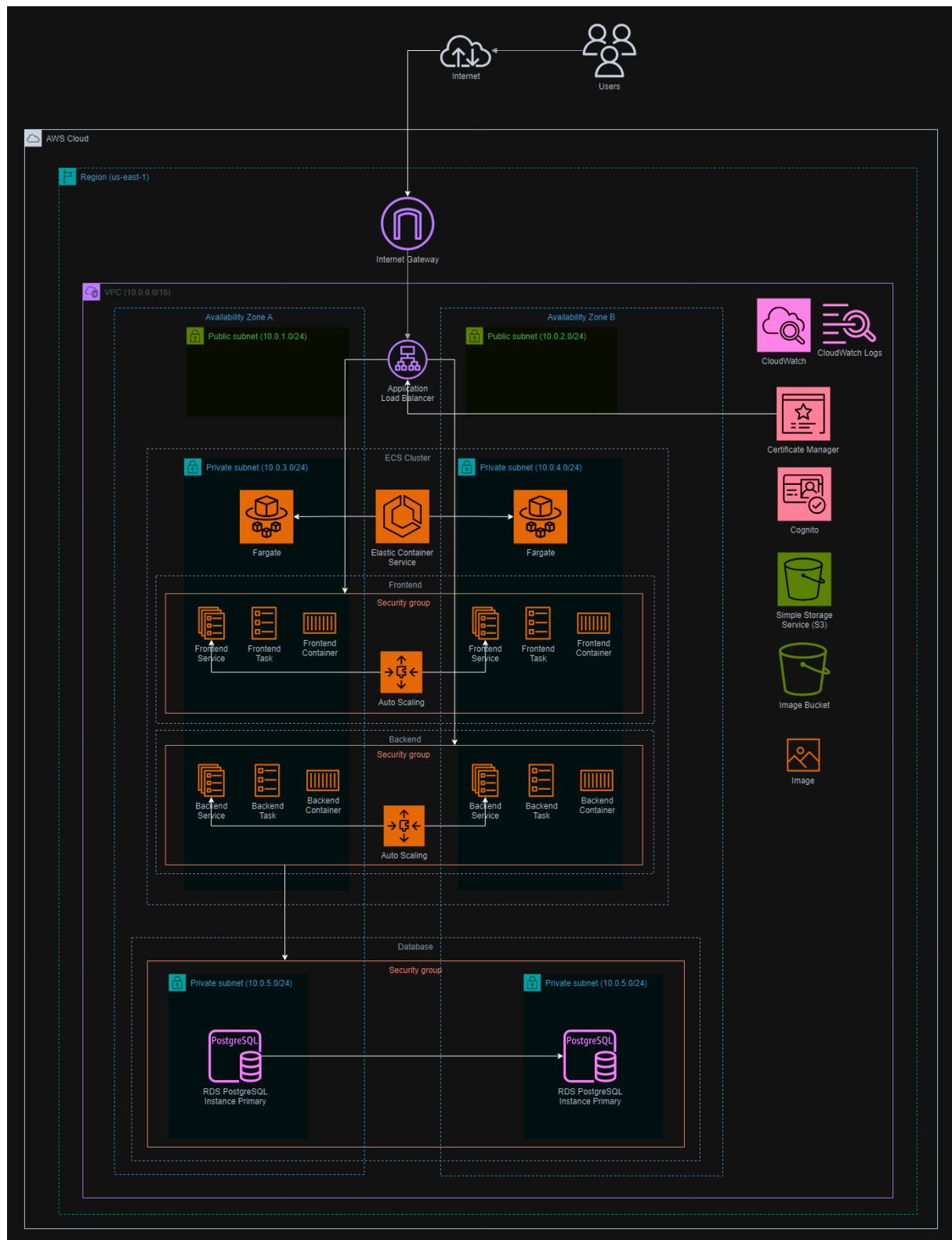
Architektura aplikacji została wdrożona przy pomocy narzędzia HashiCorp Terraform na platformie AWS. Schemat zawierający najważniejsze elementy architektury przedstawiony jest na Rysunku 2

7.3 Pozostałe schematy i diagramy

Diagram przypadków użycia

Diagram klas

Schemat przebiegu aplikacji



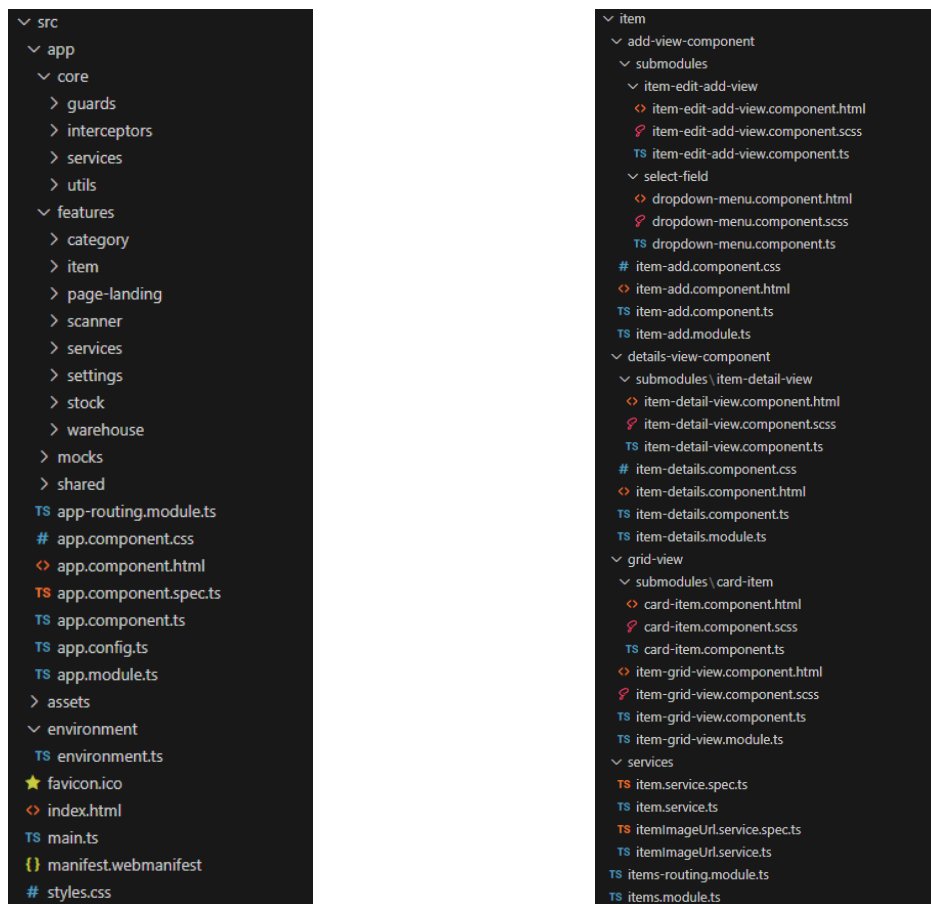
Rysunek 2: Diagram architektury AWS

8 IMPLEMENTACJA (OPCJA)

8.1 Warstwa użytkownika - frontend

8.1.1 Struktura plików

Warstwa użytkownika została zaimplementowana przy pomocy frameworku Angular [2] opartego o język TypeScript (TS). Strukturę plików przedstawia Rysunek 3.



(a) Ogólna struktura plików

(b) Dokładna struktura jednego z komponentów

Rysunek 3: Struktura plików frontendu

Aplikacja jest tworzona przy pomocy komponentów. Każdy komponent składa się ze skryptu TS, pliku stylu SCSS oraz pliku układu HTML. Dodatkowo niektóre komponenty korzystają z serwisów napisanych w języku TypeScript. Komponenty graficzne zazwyczaj mogą być wykorzystywane wiele razy i są one umieszczone w folderze *shared*. Komponenty jednokrotne są umieszczone w folderze *features* i zawierają one konkretne implementacje poszczególnych widoków.

8.1.2 Widoki

8.1.3 Routing

8.1.4 Testy

8.1.5 Konteneryzacja - Docker

8.2 Warstwa systemowa - backend

8.2.1 Struktura plików

8.2.2 Konfiguracja

8.2.3 Kontrolery

8.2.4 Inne

8.2.5 Testy

8.2.6 Konteneryzacja - Docker

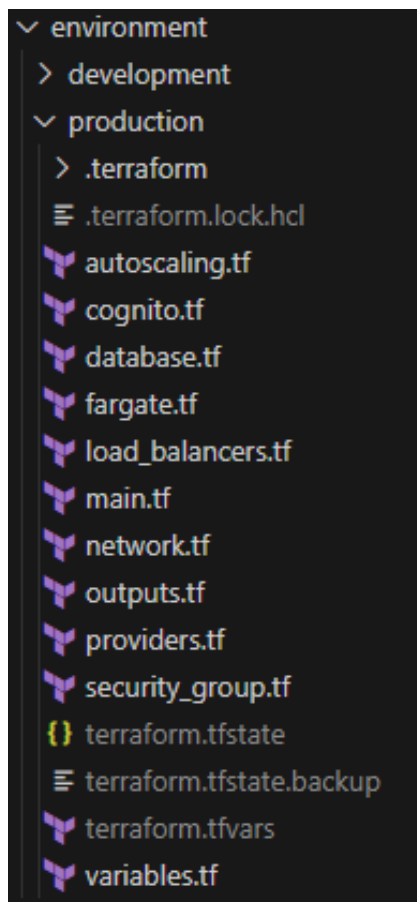
8.3 Architektura chmurowa

8.3.1 Wstęp

Architektura chmurowa została napisana w oprogramowaniu HashiCorp Terraform i wdrożona na serwery Amazon Web Services (AWS) [1]. HashiCorp Terraform (albo krótko Terraform) [3] to oprogramowanie Infrastructure as Code (IaC) pozwalające na skryptowe wdrażanie infrastruktury na serwery dostawcy.

8.3.2 Struktura plików

Struktura plików została podzielona ze względu na środowisko produkcyjne i testowe. Na Rysunku 4 przedstawione jest tylko środowisko produkcyjne ze względu na tą samą strukturę systemu plików. Z tego samego względu omówione zostanie tylko środowisko produkcyjne.



Rysunek 4: Struktura plików Terraform

Pliki .tf zawierają opis zasobów wdrażanych na serwery. Plik .tfstate przechowuje dokładny spis zasobów już wdrożonych wraz z ich unikalnymi identyfikatorami, sekretami, itd. Opis pozostałych plików:

- main.tf - wersje oprogramowania wykorzystywane w projekcie
- providers.tf - konfiguracja dostawcy AWS
- variables.tf - spis wszystkich zmiennych wraz z ich opisami i niektórymi wartościami domyślnymi
- terraform.tfvars - niewersjonowany plik zawierający wartości niektórych zmiennych; przede wszystkim przechowuje sekrety
- outputs.tf - zbiór przydatnych wartości wyjściowych, np. nazwa DNS Application Load Balancera
- network.tf - zasoby sieciowe: VPC, brama sieciowa, tablice routingu, podsieci
- security_groups.tf - grupy bezpieczeństwa (działają jak zapora ogniowa) konfiguruje przychodzący i wychodzący ruch sieciowy
- cognito.tf - dostawca tożsamości Amazon Cognito i jego konfiguracja

- `load_balancers.tf` - Application Load Balancer, zasoby nasłuchujące na portach 80, 443 i 8080 oraz grupy celów
- `fargate.tf` - definicje zadań wykonywanych przy pomocy AWS Fargate oraz serwisy które nimi zarządzają
- `autoscaling.tf` - zestaw reguł pozwalających na automatyczne zmniejszanie lub zwiększanie liczby działających instancji modułów (frontendu i backendu)
- `database.tf` - konfiguracja bazy danych RDS

8.3.3 Wykorzystane zasoby

- **Amazon Virtual Private Cloud (VPC)**
- **Amazon Cognito**
- **Amazon Relational Database Service (RDS)**
- **Amazon Application Load Balancer (ALB)**
- **Amazon Elastic Container Service (ECS)**
- **Amazon Certificate Manager (ACM)**
- **Amazon CloudWatch**
- **Amazon Simple Storage Service (S3)**
- **AWS Auto Scaling**

9 DEMONSTRACJA PRODUKTU PROGRAMOWEGO

LITERATURA

- [1] Matt Garman (CEO). Aws dokumentacja. <https://docs.aws.amazon.com>. Dostęp: 2024-11-30.
- [2] Google. Angular dokumentacja. <https://angular.dev>. Dostęp: 2024-11-30.
- [3] HashiCorp. Hashicorp terraform dokumentacja. <https://developer.hashicorp.com/terraform/docs>. Dostęp: 2024-11-30.