**Wydział Nauk Ścisłych i Technicznych**

Wiktor Pink

326430

**Automatyzacja budowy infrastruktury w chmurze Azure przy użyciu narzędzi Ansible, Terraform i Bicep.**

PRACA DYPLOMOWA MAGISTERSKA

Promotor: dr Łukasz Strąk

Katowice, 2023 r.

Spis treści

[Wstęp 3](#_Toc148212550)

# Wstęp

Chmura obliczeniowa zrewolucjonizowała sposób, w jaki przedsiębiorstwa oraz zwykli użytkownicy podejmują działania w zakresie technologii informacyjnych. Pozwoliła ona na dunamiczne skalowanie zasobów, elastyczność w zarządzaniu infrastrukturą oraz przyspieszenie innowacji produktu. Projekty które stawiają na coraz większą elastyczność oraz skalowalność muszą efektywnie zarządzać swoją infrastrukturą. W tym kontekście „Infrastruktura jako kod” (ang. Infrastructure as Code) stało się kluczowym elementem w ekosystemie chmurowym, zapewniając powtarzalność jak spójność podczas wdrażania zasobów.

Niniejsza praca ma na celu zbadanie oraz przedstawienie mechanizmów, które pozwalają na automatyzację procesów zarządzania infrastrukturą w chmurze obliczeniowej Microsoft Azure. Microsoft Azure jest czołowym dostawcą usług chmurowych na całym świecie i tym samym oferuje szeroką gamę usług i rozwiązań dla wszystkich przedsiębiorstw. Skomplikowana architektura i ogromna liczba usług stanowią wyzwanie dla administratorów i inżynierów działajacych w obrębie rozwiązań chmurowych. Potrzeba narzędzi, które umożliwiają efektywne wdrażanie, monitorowanie i narządzanie tymi zasobami staje się niezwykle pomocna.

Wiele narzędzi jak i języków zostało stworzonych w celu maksymalizacji korzyści, które niesie ze sobą chmura, minimalizując tym samym złożność zarządzania. Trzy z tych narzędzi – Ansible, Terraform i Bicep – stanowią główny przedmiot badań tej pracy. Każde z tych narzędzi posiada unikalne cechy i podejście do problemu Infrastruktury jako kod a ich porównanie i analiza pozwoli na dokładne zrozumienie ich mocnych i słabych stron jak i pozwoli na określenie okolicznosci w których jedno narzędzie może być bardziej odpowiednie od innego.

# Problematyka

Chmury obliczeniowe na przełomie klilku ostatnich lat zaczęły odkrywać kluczową rolę w globalnej infrastrukturze IT. Przedsiębiorstwa różnych rozmiarów, zaczynając od start-upów a kończąc po wielkie korporacje, coraz częściej przenoszą swoje zasoby do chmur z wielu powodów, na przykład:

* **Skalowalnosć**: Microsoft Azure oferuje możliwość elastycznego dostosowywania zasobów do aktualnych potrzeb użytkownika, umożliwiając skalowanie w dół lub w górę w zależności od określonych wymagań
* **Koszty**: Chmura Azure dostarcza model, który oparty jest na płaceniu wyłącznie za faktyczne zużycie zasobów, co z pewnością w wielu przypadkach okaże się bardziej ekonomiczne w odróżnieniu od utrzymywania własnego centrum danych. Ponadto taki model pozwala uniknąć kosztów związnych z zakupem odpowiedniego sprzętu, prac konserwacyjnych czy też kosztów wynikających z modernizacji istniejącego
* **Bezpieczeństwo**: Dostawcy chmur stawiają na bardzo zaawansowane mechanizmy bezpieczeństwa takie jak szyfrowanie, monitoring oraz certyfikacje co zazwyczaj przewyższa możliwości indywidualnych firm.

Microsoft Azure jest czołowym dostawcą chmur obliczeniowych na świecie, obok takich produktów jak AWS stworzone przez firmę Amazon czy też Google Cloud Platform.

W tradycyjnych środowiskach IT, zarządzanie infrastrukturą wiąże się z ręczną konfiguracją i utrzymaniem sprzętu oraz oprogramowania. Procesy te są niezwykle czasohłonne

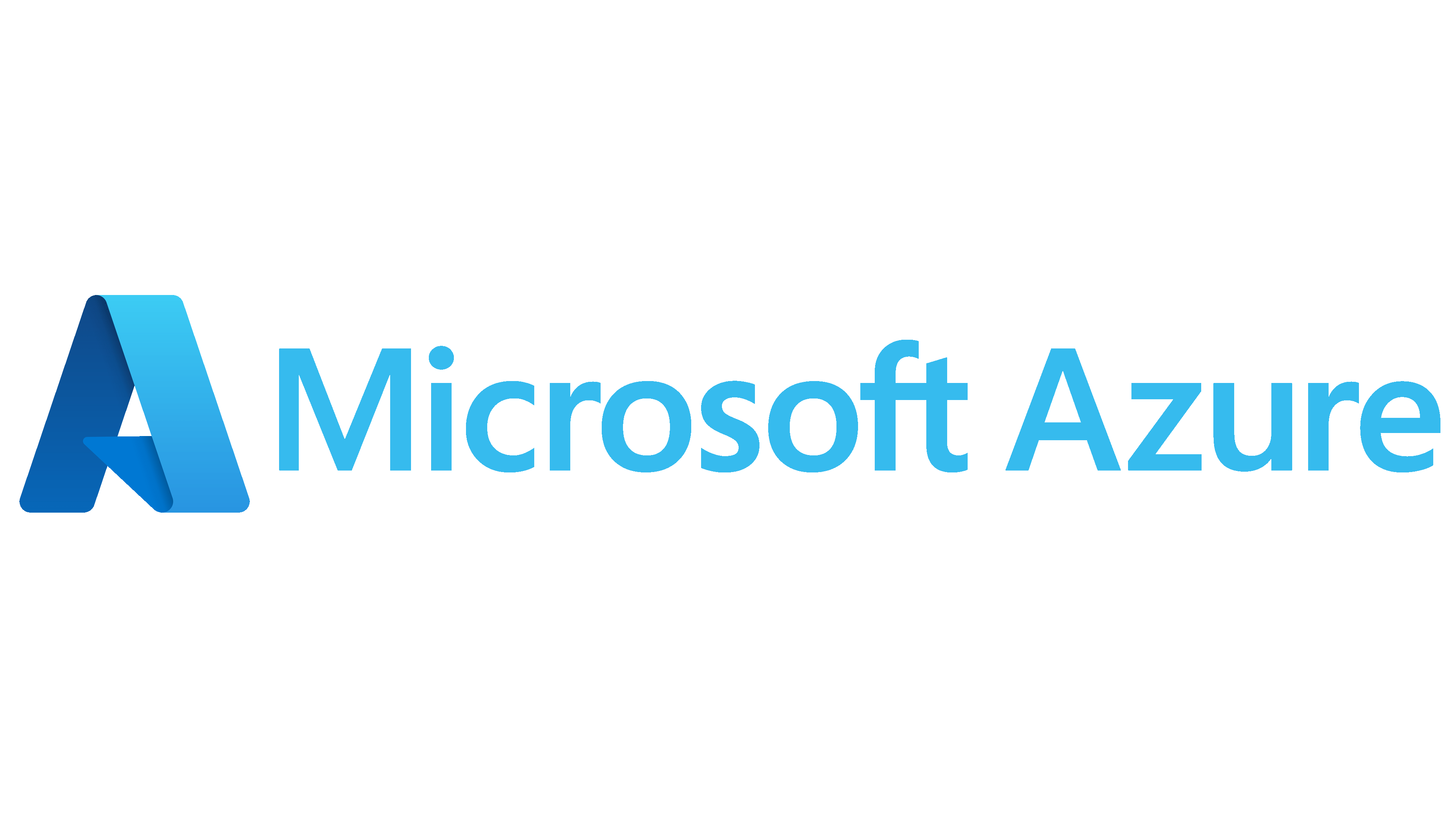
# 1. Problematyka V2

W dzisiejszym dynamicznie rozwijającym się środowisku technologicznym, chmura obliczeniowa stała się fundamentem dla wielu przedsiębiorstw dążących do cyfrowej transformacji. Chmura obliczeniowa oferuje nie tylko skalowalność oraz elastyczność ale także pozwala firmom na efektywne zarządzanie zasobami i kosztami. Jednakże wraz z adopcją tych technologii pojawiają się wyzwania. Najważniejszym z tych wyzwań jest skomplikowane zarządzanie narzędziami chmurowymi. Gdy przedsiębiorstwa rozwijają swoją infrastrukturę w chmurze stają się one bardziej złożone i zróżnicowane. Ręczne tworzenie, skalowanie, konfigurowanie czy pozbywanie się istniejących zasobów staje się nie tyle co czasochłonne ale i narażone na ludzkie błędy, które mogą prowadzić do problemów z bezpieczeństwem, niewłaściwym konfigurowaniem czy niepotrzebnymi kosztami.

Z czasem rośnie potrzeba standaryzacji jak i spójności w procesach wdrożeniowych. Przedsiębiorstwa muszą zapewnić, że ich zasoby są konfigurowane w sposób jednolity, niezależnie od skali złożoności projektu. Nie tylko ułatwia to zarządzanie ale także minimalizuje ryzyko błędów.

Microsoft Azure jest jedną z czołowych platform chmurowych, oferująca szeroką gamę usług, które mogą być wykorzystane do budowy złożonych rozwiązań. Narzędzia takie jak Ansible, Terraform i Bicep zostały stworzone po to aby ułątwić automatyzację procesów wdrażania w Azure. Narzędzia te mają swoje unikatowe cechy i zalety, ale wszystkie dążą do tego samego celu czyli, przekształcenia infrastruktury w kod, który jest łatwo powtarzalny, testowalny i można go wersjonować. Dzięki temu przedsiębiorstwa mogą budować, testować i wdrażać swoją infrastrukturę w sposób bardziej kontrolowany i zautomatyzowany.

# 1.2 Chmura Microsoft Azure



Rysunek 1.**Błąd! Użyj karty Narzędzia główne, aby zastosować 0 do tekstu, który ma się tutaj pojawić.**.1 Logo Microsoft Azure

Microsoft Azure jest jednym z czołowych dostawców chmur obliczeniowych na świecie, dostarczając szeroki wachlarz zintegrowanych usług chmurowych, które firmy z różnych branż wykorzystują do przyspieszenia ich procesów biznesowych, innowacji oraz adaptacji w dynamicznie zmieniającym się krajobrazie technologicznym. Azure, będący częścią rodziny produktów Microsoft, charakteryzuje się nie tylko zaawansowanymi technologicznie usługami, ale także głęboką integracją z innymi popularnymi produktami firmy.   
  
Główne założenia chmury Microsoft Azure:

* **Wszechstronność**: Azure oferuje szeroki wachlarz usług SaaS, PaaS oraz IaaS. Obejmuje to bazodanowe usługi, narzędzia do analizy danych, narzędzia do tworzenia i wdrażania aplikacji, oraz wiele innych. Firmy mogą budować złożone architektury, korzystając z jednego dostawcy co zdecydowanie upraszcza zarządzanie i integrację.
* **Bezpieczeństwo**: Azure wyposażony jest w zaawansowane mechanizmy bezpieczeństwa, takie jak zasady dostępu, szyfrowanie, monitoring oraz regularne audyty bezpieczeństwa. Cała infrastruktura jest zaprojektowana tak aby była jak najbardziej odporna na potencjalne ataki oraz tak aby ryzyko było jak najmniejsze.
* **Integracja**: Azure daje przewagę w postaci łatwej integracji z innymi produktami Microsoft, takim jak Office 365 czy nawet Visual Studio.
* **Elastycznosć i skalowalność**: Jednym z głównych atutów chmur obliczeniowych jest ich umiejętność do szybkiego dostosowywania się do potrzeb ich użytkownikw. Możliwość szybkiego zwiększania lub zmienjszania zasobów pozwala na wydajną optymalizację kosztów i wydajności, dostosowując się tym samym do zmieniających się potrzeb.

Microsoft Azure stale wprowadza innowacje, dostarczając nowe usługi i funkcje. W ciągu ostatnich lat, zintrodukowano narzędzia takie jak Azure Kubernetes Service, Azure DevOps i Azure AI, które umożliwiają tworzenie bardziej złożonych i skalowalnych aplikacji w chmurze. W kontekście automatyzacji, Azure dostarcza narzędzia takie jak Azure Resource Manager do zarządzania zasobami, czy też Azure Policy do egzekwowania zasad bezpieczeństwa na poziomie zasobów. Microsoft azure jest komercyjnym produktem a mimo to posiada głęboką integrację z narzędziami typu Open Source. Azure wspiera wiele technologii takich jak Linux, Java, Python.

# 1.3 Bicep



Rysunek 1.2 Logo Bicep

Bicep to deklaratywny język do definiowania i wdrażania infrastruktury w Azure jako kod. Deklaratywny język programowania pozwala skupić się na tym co program ma osiągnąć a nie na tym jak daną rzecz czy operację ma wykonać. Bicep został zaprojektowany przez firmę Microsoft jako ulepszona i bardziej czytelna alternatywa dla składni JSON która używana jest w szablonach Azure Resource Manager. Dzięki temu kod napisany w języku Bicep może być nawet o połowę krótszy w porównaniu do szablonu JSON. Nazwa tego języka jest związana z siłą oraz skutecznością jaką ten język oferuje a tym samym zapewniając prostotę i elegancję w projektowaniu rozwiązań chmurowych.

Główne cechy języka Bicep:

* **Czytelność:** Bicep oferuje składnię, która jest bardziej zwięzła i czyelna niż tradycyjne szablony z Azure Resource Manager. Sprawia to że kod napisany w tym języku jest łatwy w utrzymaniu i zrozumieniu.
* **Integracja z Microsoft Azure:** Język Bicep jest ściśle zintegrowany z chmurą Azure i całym jej ekosystemem, co pozowala na łatwą i sprawną interakcję z zasobami chmurowymi
* **Modularność:** Bicep umożliwia tworzenie modularnych i wielokrotnie używanych komponentów, co przyspiesza proces projektowania i wdrażania skomplikowanych rozwiązań w chmurze.
* **Kompilacja do Azure Resource Manager:** Kod napisany w języku bicep jest kompilowany bezpośrednio do JSON-a, który używany jest przez ARM(Azure Resource Manager), co zapewnia pełną zgodność z istniejącymi rozwiązaniami.
* **Orkiestracja:** Azure Resource Manager odpowiedzialny jest za organizację i wdrażanie współzależnych zasobów tak aby były tworzone w odpowiedniej kolejności. Jeśli jest to możliwe to zasoby wdrażane są równolegle, dzięki czemu tego typu wdrożenia zakończą się dużo szybciej niż wdrożenie szeregowe. Plik wdrażany jest za pomocą jednego polecenia dzięki czemu nie ma potrzeby wywoływania wielu poleceń imperatywnych.

Deklaratywna natura i składnia tego języka umożliwia na dokładne określenie wymagań infrastrukturalnych projektu, co pozwala na automatyczne tworzenie i zarządzanie zasobami chmurowymi. W odróżnieniu od szablonu JSON, Bicep jest dużo bardziej elastyczny i tym samym pozwala na wprowadzanie zmiennych, parametrów czy też danych wyjściowych, natomiast w przypadku szablonu JSON, wszystkie parametry, zmienne czy dane wyjściowe muszą być zadeklarowane w odpowiednich sekcjach szablonu

# 1.4 Ansible



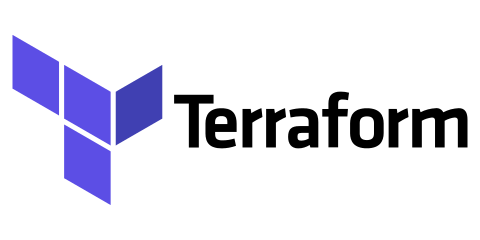
Rysunek 1.3 Logo Ansible

Ansible jest narzędziem stworzonym przez firmę Red Hat, swoją popularność zdobyło dzięki swojej prostocie oraz uniwersalności. W przeciwieństwie do innych rozwiązań, Ansible nie wymaga instalacji agenta na zarządzanych maszynach, co znacznie ułatwia jego wdrożenie i konfigurację. Narzędzie to wykorzystuje język oparty na języku YAML, który jest uniwersalnym językiem formalnym przeznaczonym do reprezentowania różnych danych w ustrukturyzowany sposób. Ansible opiera się na tymczasowych połączeniach zdalnych za pośrednictwem protokołu SSH lub na zdalnym zarządzaniu systemem Windows co umożliwia przykładowo na wykonanie programu PowerShell.

Główne cechy narzędzia Ansible:

* **Bezstanowość:** Ansible działa w sposób bezstanowy, oznacza to że operacje są powtarzalne bez wprowadzania dodatkowych zmian.
* **Modularność:** Ansible posiada bogaty zestaw modułów, które można dostosować przykładowo w konfiguracji serwerów do zarządzania sieciami czy też we wdrażaniu aplikacji.
* **Integracja z Chmurami:** Narzędzie Ansible oferuje moduły dedykowane dla głównych dostawców chmurowych, w tym Microsoft Azure, co ułatwia automatyzację zadań w środowiskach hybrydowych.

# 1.5 Terraform

****

Rysunek 1.4 Logo Terraform

Terraform jest to oprogramowanie stworzone przez firmę HashiCorp, i wyróżnia się jako jedno z czołowych narzędzi w zakresie Infrastruktury jako kod w ekosystemie chmurowym. Zapewnia rozwiązania umożliwiające zarządzanie chmurą w dynamicznie rozwijających się projektach biznesowych. Terraform pozwala na elastyczne i w pełni zautomatyzowane zarządzanie zasobami. Od momentu wejścia na rynek tego oprogramowania od razu zaczęło one zdobywać uznanie jako potężne narzędzie umożliwiające definiowanie i dostarczanie infrastruktury w sposób deklaratywny, przy użyciu języka opartego na tekstowym formacie HashiCorp Configuration language.  
  
Główne cechy oprogramowania Terraform:

* **Deklaratywność:** użytkownik określa konfigurację końcową, a nie kroki konieczne do jej osiągnięcia. Terraform automatycznie rozwiązuje zależności między zasobami.
* **Planowanie:** Terraform umożliwia użytkownikowi przeglądanie planowanych zmian przed ich wprowadzeniem, co zwiększa bezpieczeństwo i przewidywalność zamierzonych działań.
* **Rozszerzalność:** Oprogramowanie terraform możne interaktywować z wieloma platformami, w tym z platformą Azure. Dzieki szerokiej gamie dostawców, możliwe jest zarządzanie niemal dowolnymi zasobami.
* **Stan:** Terraform śledzi stan aktualnej infrastruktury, co pozwala na efektywne zarządzanie zmianami i detekcję różnic między stanem rzeczywistym a tym oczekiwanym.

# 1.6 Infrastruktura jako Kod

Infrastruktura jako kod w skrócie IaaC, to praktyka zarządzania i konfiguracji infrastruktury poprzez definicję maszynowo przetwarzalne w formie kodu, zamiast tradycyjnych procesów ręcznej konfiguracji lub interfejsów użytkownika. W praktyce oznacza to traktowanie infrastruktury tak samo jak oprogramowanie. Tego typu podejście pozwala na testowanie, kodowanie i przechowywanie w systemach kontrolowania wersji skryptów odpowiedzialnych za powstawanie odpowiednich zasobów. Obecnie podejście infrastruktury jako kod jest kluczowe do sprostania wyzwaniom wymagających automatyzację i organizację automatycznie powstających zasobów. Zastosowanie IaaC prowadzi do znacznego zwiększenia efektywności, elastyczności i niezawodności zarządzania infrastrukturą. Infrastrukura jako kod pozwala organizacjom na sprawne dostosowanie się do dynamicznego rozwoju technologii.

Główne korzyści Infrastruktury jako kod:

* **Automatyzacja:**  Redukcja ręcznych zadań i tym samym przyspieszenie procesu wdrożenia i minimalizacja ludzkich błędów.
* **Powtarzalnosć:** Środowiska stworzone poprzez wykorzystanie kodu do tworzenia infrastruktury są łatwe do replikacji, co przekłada się na łatwiejsze testowanie i skalowanie produktów implementowanych na tego typu środowiska.
* **Standaryzacja:** Wszystkie zasoby tworzone przez jeden skrypt są konfigurowane w jednolity sposób co przyczynia się do zwiększenia bezpieczeństwa.
* **Dokumentacja:** Kod wykorzystywany do budowy infrastruktury jest sam w sobie formą dokumentacji ponieważ pokazuje jak dokłądnie infrastruktura jest konfigurowana.

Infrastrukturę jako kod można wykorzystać w wielu zróżnicowanych przypadkach, przykładowo w przypadku kiedy środowisko musi dynamicznie dostosowywać się do rosnących wymagań, podczas przenoszenia zasobów między środowiskami lub chmurami. Jeśli infrastruktura uległa poważnej awarii dzięki IaaC można ją w bardzo krótkim czasie odtworzyć.

# 2 Podobne rozwiązania

Mimo, że Microsoft Azure jest wiodącym dostawcą rozwiązań chmurowych to na rynku istnieją inne rozwiązania, stanowiące bezpośrednią konkurencję dla Azure. Firma Amazon dostarcza swoją chmurę o nazwie AWS, która powstała przed chmurą Azure i jest jedną z najdłużej działających chmur w historii. Azure natomiast w odróżnieniu od AWS posiada dużo lepszą integrację z produktami firmy Microsoft. Kolejną firmą stanowiącą bezpośrednią konkurencję jest firma Google, która zaprezentowała swoją chmurę Google Cloud Platform, która skoncentrowana jest na mniejszych przedsiębiorstwach i oferowaniu usług związanych z dużymi danymi i analityką. Jeśli chodzi o narzędzia oferujące możliwości Infrastruktury jako kod to na rynku podobnym do Terraform, Ansible i bicep jest jeszcze narzędzie „Puppet”, które umożliwia automatyczne zarządzanie konfiguracją i pozwala na zdefiniowanie infrastruktury w deklaratywnym języku opisowym

# 2.1 Amazon Web Services (AWS)



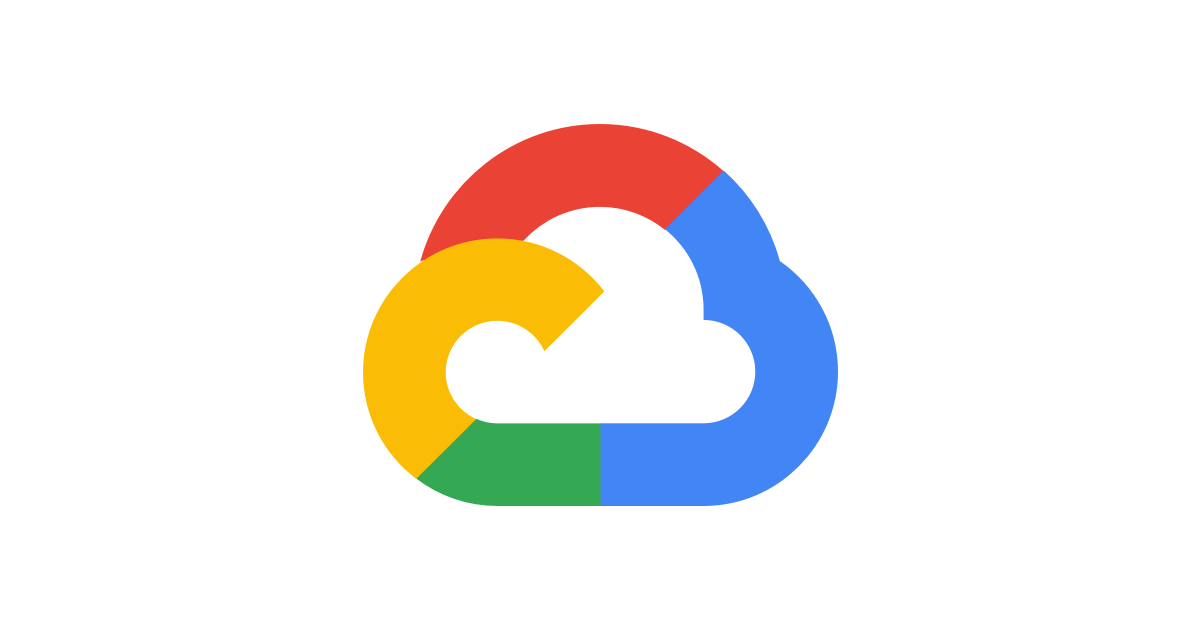
Rysunek 2.1 Logo AWS

Amazon Web Services to platforma chmurowa wyprodukowana przez firmę Amazon. Oferuje ona szeroką gamę usług infrastruktury internetowej. Została ona wprowadzona w 2006 roku i stała się najbardziej dominującą chmurą wsród wszystkich wtedy obecnych dostawców chmurowych. Jako pierwsza to właśnie chmura AWS zdefiniowała nowy model biznesowy „pay-as-you-go” i tym samym wyznaczyła nowy standard na całym rynku. Wspomniany wcześniej model biznesowy polega na płaceniu za kupione usługi tylko i wyłącznie wtedy kiedy te usługi są wykorzystywane. Pozwala to na uniknięcie kosztów początkowych a także ich przewidywanie, dzięki narzędziom do monitorowania i prognozowania kosztów. Użytkownik płaci za godziny w których przykładowo maszyny wirtualne działają, czy też gigabajty przechowywanych danych a także ilość wysyłanych zapytań do bazy.

**Cechy Amazon Web Services:**

* **Różnorodność usług:** AWS ma do zaoferowania ponad 200 kompletnych usług zaczynając od baz danych, przez analizę danych aż po uczenie maszynowe czy też sztuczną inteligencję
* **Skalowalnosć i elastycznosć:** usługi dostarczane przez AWS zostały zaprojektowane tak aby mogły automatycznie się skalować, co eliminuje konieczność inwestowania w nadmierną infrastrukturę
* **Globalna obecność:**  AWS posiada swoje centrale w 24 regionach geograficznych na całym świecie tym samym gwarantując niske opóźnienia i wysoką dostępność.

# 2.2 Google Cloud Platform

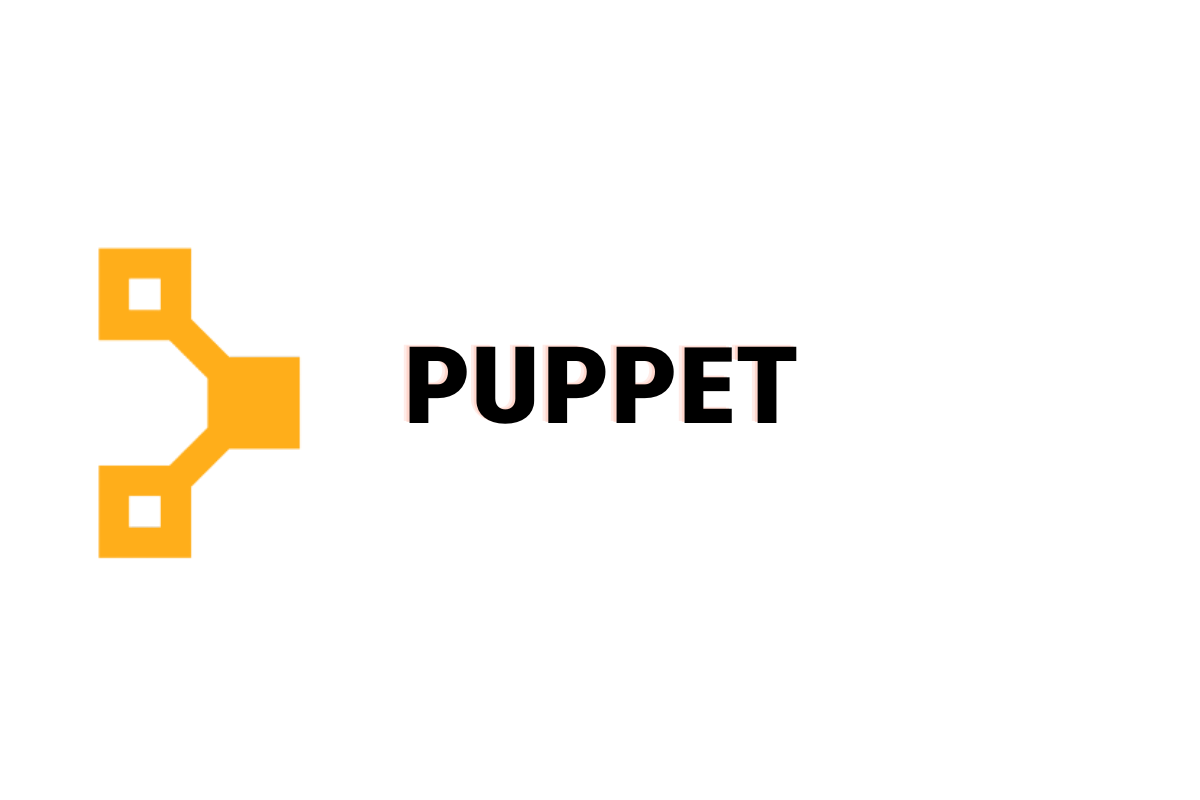


Google Cloud Platform jest zestawem usług chmurowych oferowanych przez Google i jest przeznaczony dla pojedynczych mniejszych projektów jak i do globalnych dużych przedsiębiorstw. Chmura od Google, dostarcza narzędzia dla deweloperów i przedsiębiorstw do budowy, wdrażania i skalowania aplikacji. Pozwala ona na integrację swoich rozwiązań w jednym środowisku z innymi produktami Google przykładowo G Suite. Google Cloud Platform pojawiło się na rynku w 2008 roku w odpowiedzi na szybko rosnący rynek usług chmurowych, doganiając tym samym Amazon i Microsoft i stając się tym samym jednym z głównych graczy na rynku usług chmurowych.

**Cechy Google Cloud Platform:**

* **Narzędzia dla deweloperów:** Google cloudoferuje narzędzia przeznaczone dla deweloperów, takie jak Firebase czy Kubernetes Engine, pomagające w tworzeniu, testowaniu i wdrażaniu aplikacji w chmurze
* **Integracja:** Google cloud zapewnia zintegrowane środowisko deweloperskie dzięki narzędziom takim jak Cloud Shell lub Cloud SDK
* **Uczenie maszynowe i Szuczna inteligencja:** Google Cloud Platform znany jest z narzędzi do uczenia maszynowego takiego jak TensorFlow oraz dostępu do API Vision, Speech czy Natural Language

# 2.3 Puppet



Puppet jest narzędziem do automatycznego zarządzania konfiguracją, które pozwala zdefiniować stan oprogramowania i infrastruktury w deklaratywnym języku opisowym. Działa wielu systemach operacyjnych i na wielu platformach chmurowych oferując tym samym wysoce skalowalne i wydajne rozwiązanie do zarządzania infrastrukturą od małych do bardzo dużych środowisk.

**Cechy narzędzia Puppet**

* **Deklaratywny język**: język zastosowany w narzędziu Puppet umożliwia na definiowanie złożonych stanów systemów w czytelny i zrozumiały sposób.
* **Skalowalność:** dzięki Puppet można zarządzać tysiącami maszyn równocześnie co sprawia że jest bardzo dobrym wyborem w dużych środowiskach.
* **Wsparcie dla wielu systemów operacyjnych:** Puppet obsługuje zarówno systemy Windows jak i Linux
* **Integracja z chmurą:** Puppet umożliwia działanie na wielu platformach oraz oferuje zintegrowane rozwiązania dla głównych dostawców chmur takich jak Azure, Amazon Web Services czy Google Cloud Platform.

Puppet poodbnie jak Terraform, Ansible i Bicep jest wykorzystywany do automatyzacji zarządzania infrastrukturą lecz z naciskiem na konfigurację systemu. W porównaniu do narzędzia Ansible które jest narzędziem „pull”, Puppet bazuje na agencie z systemem „push”, oznacza to że centralny serwer Puppet dystrybuuje konfigurację do zdalnych agentów działających na zarządzanych maszynach.