

Wyższa Szkoła Informatyki Stosowanej i Zarządzania

pod auspicjami Polskiej Akademii Nauk



WYDZIAŁ INFORMATYKI

Kierunek INFORMATYKA

Studia I stopnia (dyplom inżyniera)



PRACA DYPLOMOWA

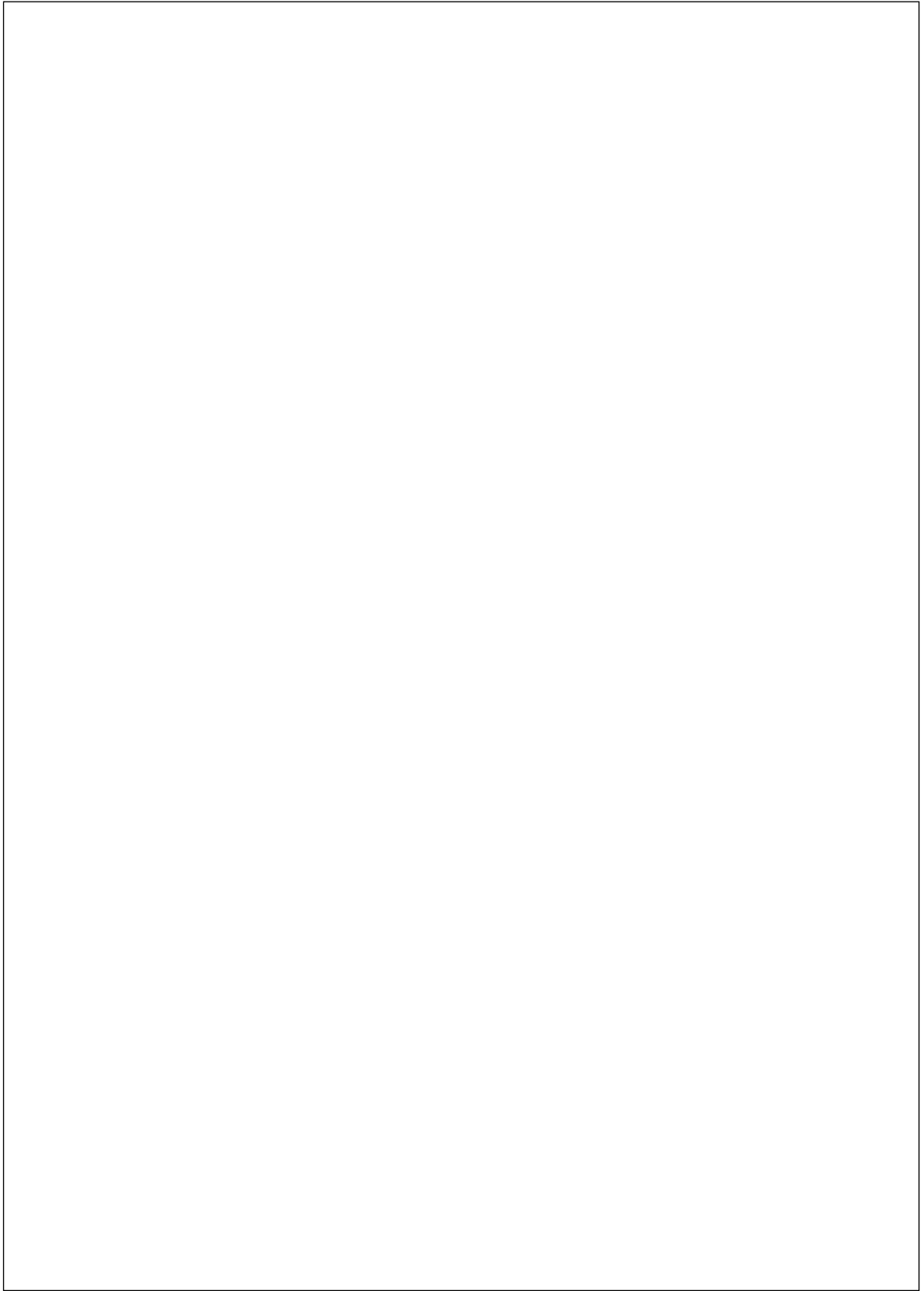
Igor Nowicki

Projekt, implementacja i zautomatyzowane
wdrożenie systemu do wizualizacji danych w
chmurze Azure

Promotor pracy:

Dr inż. Jarosław Sikorski

WARSZAWA, rok akademicki 2022/2023





Wyższa Szkoła Informatyki Stosowanej i Zarządzania

pod auspicjami Polskiej Akademii Nauk
ul. Nowelska 6, 01-447 Warszawa

Wydział Informatyki i Grafiki

Kierunek: **Informatyka**

OŚWIADCZENIE AUTORA PRACY

Temat pracy: *Projekt, implementacja i zautomatyzowane wdrożenie systemu do wizualizacji danych w chmurze Azure.*

Świadom(a) odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami prawa, w szczególności z ustawą z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. z 1994 r. Nr 24, poz. 83 – tekst pierwotny i Dz. U. z 2000 r. Nr 80, poz. 904 – tekst jednolity, z późn. zm.). Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w szkole wyższej. Oświadczam także, że wersja pracy składana w dziekanacie, pliki elektroniczne pracy zawierają identyczną treść. Oświadczam również, że wszystkie narzędzia informatyczne zastosowane do wykonania niniejszej pracy wykorzystałem(am) zgodnie z obowiązującymi przepisami prawa w zakresie ochrony własności intelektualnej i przemysłowej. Wyrażam zgodę na przetwarzanie następujących danych osobowych: imiona, nazwisko, nr albumu, temat pracy dyplomowej oraz oceny na dyplomie, ukończony kierunek oraz rok, system i rodzaj studiów, adres do korespondencji (w tym e-mail i telefon kontaktowy) dla celów badania losu absolwenta zgodnie z Rozporządzeniem Parlamentu Europejskiego i Rady Unii Europejskiej 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (RODO).

Wyrażam zgodę na publikację pracy.

Wyrażam zgodę na przetwarzanie następujących danych osobowych: imiona, nazwisko, nr albumu, ukończony kierunek oraz rok, system i rodzaj studiów, adres do korespondencji (w tym e-mail i telefon kontaktowy) dla celów promocji uczelni zgodnie z Rozporządzeniem Parlamentu Europejskiego i Rady Unii Europejskiej 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (RODO).

Wyrażam zgodę na przetwarzanie następujących danych osobowych: tematu pracy dyplomowej oraz oceny na dyplomie dla celów promocji uczelni zgodnie z Rozporządzeniem Parlamentu Europejskiego i Rady Unii Europejskiej 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (RODO).

Wyrażam zgodę na przetwarzanie następujących danych osobowych: aktualnego miejsca pracy, miejsca pracy po ukończeniu studiów oraz obecnie zajmowanego stanowiska dla celów promocji uczelni zgodnie z Rozporządzeniem Parlamentu Europejskiego i Rady Unii Europejskiej 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (RODO).

Wyrażam zgodę na przesyłanie mi informacji o ofertach uczelni i ankiet dla celów promocji uczelni zgodnie z Rozporządzeniem Parlamentu Europejskiego i Rady Unii Europejskiej 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (RODO).

Wyrażam zgodę na przetwarzanie następujących danych osobowych: imiona, nazwisko, nr albumu, ukończony kierunek, specjalność oraz rok, system i rodzaj studiów, wynik ukończenia, miejsce urodzenia w celu wydrukowania dyplomu na zlecenie WSISiZ przez drukarnię ZPW Bogucin k/Poznań z siedzibą w Kobylnicy, ul. Gnieźnieńska 127 zgodnie z Rozporządzeniem Parlamentu Europejskiego i Rady Unii Europejskiej 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (RODO).

19/04/2023

(data)

Igor Nowicki, 18608

(student)

.....
(podpis studenta)

Niniejszy dokument został wygenerowany na podstawie danych zgromadzonych w systemie UBI.

Spis treści

Wstęp	9
1. Chmura obliczeniowa.....	10
1.1. Wstęp.....	10
1.2. Definicje	12
1.3. Rodzaje usług	13
2. Chmura Azure	14
2.1. Wstęp i historia.....	14
2.2. Model tworzenia zasobów	16
2.4. Virtual Machine	18
2.5. App Service	19
2.6. Azure SQL.....	20
2.8. Power BI.....	22
3. Platforma ASP.NET Core	23
3.1. Wstęp i historia.....	23
3.2. Architektura MVC	25
3.3. Razor	25
4. Skrypty DevOps.....	26
4.1. Wstęp.....	26
4.2. DevOps.....	27
4.3. PowerShell	27
4.4. Moduł PowerShell Az oraz Azure CLI	28
4.5. Terraform.....	28
5. Opis pracy	29
5.1. Wstęp.....	29
5.2. Zarys opisu projektu	29
6. Zbieranie danych.....	30
6.1. Wstęp.....	30
6.2. Opis wskaźników.....	30
6.3. Opis struktury bazy danych.....	31
7. Raport Power BI	33
7.1. Wstęp.....	33
7.2. Opis stron.....	33
7.3. Podsumowanie	34
8. Witryna ASP.NET Core MVC.....	35

8.1.	Wstęp.....	35
8.2.	Opis aplikacji	36
8.3.	Biblioteka programistyczna	36
8.4.	Panel administratora	37
8.5.	Wykorzystane narzędzia	38
8.6.	Integracja z usługą Power BI	38
9.	Skrypty wdrożeniowe PowerShell.....	39
9.1.	Wstęp.....	39
9.2.	Logowanie do usługi Azure	39
9.3.	Przesyłanie raportu do Power BI App	39
9.4.	Tworzenie infrastruktury z Terraform	40
9.5.	Ładowanie kopii zapasowej do przestrzeni magazynowej	40
9.6.	Ładowanie bazy danych z kopii zapasowej.....	40
9.7.	Kasowanie przestrzeni magazynowej	40
9.8.	Zmiana źródła danych w raporcie Power BI.....	40
9.9.	Tworzenie paczki binariów aplikacji	41
9.10.	Ładowanie paczki do serwisu aplikacji.....	42
9.11.	Podsumowanie	42
	Zakończenie	43
	Dyskusja	43
	Podsumowanie	44
	Podziękowania.....	44
	Bibliografia.....	45
	Spis ilustracji	45
	Informacje dodatkowe	46
	Temat	46
	Cel	46
	Zakres.....	46
	Temat w języku angielskim	46
	Streszczenie	46
	Streszczenie w języku angielskim.....	46

Coraz więcej osób i instytucji dostrzega potrzebę korzystania z nowoczesnych technologii do przetwarzania danych. Sprawne zarządzanie informacją stało się kluczowe do umiejętnego prowadzenia działalności. Rozwój Internetu, który miał swój początek w latach 60. XX wieku, znacząco wpłynął na oczekiwania wobec organizacji. Współcześnie oczekuje się, by co najmniej część usług była dostarczana zdalnie.

Pojawia się pytanie, w jaki sposób tworzyć oraz jak wdrażać usługi internetowe? Postęp technologii doprowadził do stanu, w którym samo przechowywanie i przetwarzanie danych stało się usługą świadczoną przez zewnętrzny podmiot. Przestrzeń i moc obliczeniowa są dostarczane wedle zapotrzebowania, skalowane na żądanie, z opłatami za ich używanie naliczanymi zgodnie z rzeczywistym zużyciem.

Celem pracy jest przedstawienie procesu implementacji i wdrożenia przykładowego serwisu internetowego. W programie zostały wykorzystane technologie dostarczane w ramach Microsoft Azure, jednego z największych dostawców usług chmury obliczeniowej. Niniejszy artykuł prezentuje użyte narzędzia oraz efekt ich zastosowania w projekcie.

Rozdział pierwszy omawia pojęcie chmury obliczeniowej. W rozdziale drugim przedstawiono popularne technologie wykorzystywane przy projektowaniu rozwiązań chmurowych. Rozdział trzeci prezentuje platformę ASP.NET jako narzędzie do tworzenia serwisów aplikacji. W rozdziale czwartym omówiono praktyki DevOps, program Terraform oraz interpreter PowerShell w kontekście narzędzi do tworzenia skryptów automatycznego wdrażania.

W rozdziale piątym przedstawiono zarys wykonanego zadania oraz architekturę systemu. W rozdziale szóstym znajdują się informacje na temat metodyki zbierania i obróbki danych oraz zarys struktury bazy danych. Rozdział siódmy zawiera relację z tworzenia raportu w programie Power BI Desktop. W rozdziale ósmym został zawarty opis struktury serwisu internetowego w technologii ASP.NET. Rozdział dziewiąty dotyczy przygotowania skryptów PowerShell do automatycznej publikacji witryny w chmurze Azure.

W ostatnim, nienumerowanym rozdziale przeprowadzono dyskusję na temat wykonanej pracy oraz przedstawiono wnioski odnośnie użytych technologii. Rozważania dotyczą zasadności użycia określonych narzędzi i propozycji alternatywnych dróg projektu. Omówiono potencjalne zastosowania pracy pod kątem ponownego użycia fragmentów kodu źródłowego w innych przedsięwzięciach.

Motywacją do podjęcia opisanego powyżej tematu pracy jest rosnąca popularność chmur obliczeniowych. Usługi w modelu Platform as a Service zostały użyte ze względu na prostotę projektu i brak zapotrzebowania na dodatkową konfigurację programu. Istotna ilość czasu została poświęcona na przygotowanie skryptów wdrożeniowych ze względu na popularność praktyk DevOps i metodyki Agile. Temat strony internetowej i użytych danych z bazy danych World Bank został wybrany po lekturze książki Factfulness autorstwa Hansa Roslinga.

1. Chmura obliczeniowa

1.1. Wstęp

Każda strona internetowa potrzebuje komputera, na którym może być zlokalizowana. Maszyna przechowująca i udostępniająca witrynę internetową nazywana jest serwerem, natomiast sam proces określany jest jako hosting (ang. *host*, gospodarz). Serwer obsługuje komunikację z odwiedzającymi witrynę, korzystając przy tym z zestawu programów – repozytorium plików, baz danych oraz aplikacji internetowych. Zależnie od zapotrzebowań i przewidywanego obciążenia, maszyna musi być wyposażona w odpowiednie zasoby, m.in. moc obliczeniową, pamięć operacyjną i przestrzeń dyskową.

Organizacja może tworzyć serwer na wiele sposobów. Jedno z najwcześniejszych rozwiązań to hostowanie lokalne – przechowywanie maszyny w budynku należącym do podmiotu. Pozwala to na pełną kontrolę nad danymi i działaniem komputera. Urządzenie wewnątrz firmy oznacza konieczność połączenia z Internetem, by korzystać z oferowanych przez niego usług. Właściciel ma bezpośrednią kontrolę nad danymi, co jest istotne podczas przechowywania informacji wrażliwych. Koszt zakupu maszyny jest ponoszony raz. Późniejsze koszty są związane jedynie z konserwacją i opłatami za prąd.

Takie podejście wymaga zatrudnienia osoby zajmującej się konfiguracją, naprawą i rozwojem serwera. Ponoszone są także koszty związane z wymianą sprzętu. Komputer świadczący usługi użytkownikom zewnętrznym wymaga stałego połączenia z Internetem – jednak nawet najlepsi dostawcy usług internetowych zastrzegają możliwość okresowego braku dostępu do sieci. Konieczne jest również zapewnienie stałego zasilania, co wymaga zabezpieczenia się przed potencjalnymi problemami związanymi z dostawami prądu z elektrowni.

Serwer może doświadczać sezonowej zmienności natężenia ruchu. Przykładem są sklepy internetowe obserwujące zwiększoną liczbę klientów w okresie świątecznym. Dostosowanie zasobów serwera staje się wtedy konieczne do poprawnego działania usługi. Mogą być wtedy potrzebne nowe podzespoły albo nawet cała jednostka komputerowa, co wiąże się z dodatkowymi kosztami. Mimo to ryzyko przeciążenia nadal istnieje, gdy planowane obciążenie zostanie przekroczone.

Innym rozwiązaniem jest hosting dedykowany, czyli wykupienie dostępu do serwera u zewnętrznego dostawcy. Pozwala to na zredukowanie kosztów konserwacji i ryzyka związanego z dostępnością. Odbiorca usług zawiera umowę i uiszcza zryczałtowaną opłatę za dostęp. W ramach umowy gwarantowany jest odpowiednio wysoki czas dostępności, pakiet usług (np. dostęp do bazy danych, serwera plików, odpowiedniego środowiska uruchomieniowego, etc.) oraz graniczna przepustowość. Podmiot łączy się zdalnie z maszyną, przygotowując ją do świadczenia usług odwiedzającym stronę internetową. Gdy skalowanie usługi staje się konieczne, straty są minimalizowane – nie trzeba kupować nowego komputera, lecz jedynie zmodyfikować umowę dotyczącą korzystania z maszyny o innych parametrach. Konserwacja sprzętu, bezpieczeństwo oraz aktualizacja oprogramowania są również obowiązkami przeniesionymi na dostawcę.

Hosting dedykowany wymaga konfiguracji nowych maszyn, co jest czasochłonnym procesem. W przypadku dużej awarii u dostawcy, odbiorca usług traci dostęp do maszyny. Dostawca usług jest zazwyczaj zlokalizowany w jednym miejscu – jeśli świadczona usługa jest popularna na całym świecie, oznacza to dłuższy czas odpowiedzi do bardziej oddalonych użytkowników. Model kosztowy ogranicza możliwość eksperymentowania z maszynami – niemożliwe jest wynajęcie serwera na krótszy okres, np. w celach testowych – maszyna musi być dzierżawiona na co najmniej miesiąc.

Chmura obliczeniowa jest rozwiązaniem podobnym do hostingu dedykowanego, jednak wynajmowany serwer jest tworzony na czas świadczenia usługi i usuwany po zakończeniu działania. Maszyna zostaje powołana jako zasób wirtualny w środowisku całego centrum obliczeniowego, składającego się z setek komputerów. W ten sposób parametry serwera mogą być dowolnie zmieniane, poprzez przydzielanie mu mniejszych bądź większych zasobów na dany moment.

Każdy potrzebny komponent – maszyna wirtualna, baza danych, serwer plików, środowisko uruchomieniowe aplikacji bądź nawet przestrzeń robocza wykonywania pojedynczej funkcji – może być tworzony w ramach chmury obliczeniowej jako osobny zasób, opłacany wedle zużycia. Większość zasobów może być skalowana w czasie rzeczywistym – jeśli w trakcie działania serwera okaże się, że brakuje przestrzeni dyskowej, podmiot może zażądać dodanie kolejnego dysku, można również utworzyć skrypt automatycznie przyznający dodatkową przestrzeń w zależności od zapotrzebowań.

Podmiot korzystający z publicznej chmury obliczeniowej może tworzyć instancje serwera w różnych lokalizacjach geograficznych, skracając czas dostępu użytkownikom w różnych regionach świata. Dodatkowo, możliwa jest redundancja pomiędzy różnymi centrami danych – nawet podczas dużych awarii czy kataklizmów dane powinny pozostać nienaruszone, o ile zdecydowano się na używanie wielu centrów danych do hostowania usług.

Przejęcie odpowiedzialności za przechowywanie witryny internetowej przez zewnętrzną organizację oznacza, że podmiot korzystający z usług będzie zależna od awarii po stronie dostawcy^{1 2}. Dodatkowo, właściciele chmur obliczeniowych zastrzegają sobie prawo do nagłej odmowy świadczenia usług wybranym klientom³.

Model automatycznego skalowania zasobów może być w niektórych przypadkach również traktowany jako wada. Przy nieodpowiednio skonfigurowanej usłudze, działanie aplikacji może doprowadzić do bardzo wysokich kosztów. Przypadkiem może być błąd algorytmu prowadzącego do niekontrolowanego wzrostu liczby zapytań, co prowadzi do zużycia dużej ilości zasobów chmurowych w krótkim czasie⁴. Problematyka ta nie dotyczy hostingu lokalnego czy dedykowanego – aplikacja może dalej wykonywać dużą liczbę zapytań, jednak spowoduje to zużycie wszystkich dostępnych zasobów maszyny, co zakończy się odmową działania i przymusem restartu komputera.

Podsumowując, proces hostowania witryny internetowej może być przeprowadzony z użyciem różnych narzędzi, z których każde ma swoje wady i zalety. W dalszych częściach pracy omówię szczególne cechy używania chmury obliczeniowej Microsoft Azure.

¹ <https://techhq.com/2023/01/azure-outage-disconnects-thousands-from-outlook-and-teams-around-the-world/>

² <https://www.forbes.com/sites/kellyclay/2012/12/24/amazon-aws-takes-down-netflix-on-christmas-eve/>

³ <https://time.com/5929888/amazon-parler-aws/>

⁴ <https://blog.tomilkieway.com/72k-1/>

1.2. Definicje

Nie istnieje jedna definicja chmury obliczeniowej. Autorzy prac z dziedziny przedstawiają następujące opisy:

- **Chmura jest systemem rozproszonym** – według (Buyya, 2011): *chmura to rodzaj równoległego i rozproszonego systemu składającego się z kolekcji połączonych ze sobą i zwirtualizowanych komputerów, które są dynamicznie dostarczane i prezentowane jako jeden lub więcej zunifikowanych zasobów obliczeniowych w oparciu o umowy o poziomie usług, ustalonych w drodze negocjacji między dostawcą usług i konsumentami.*
- **Chmura umożliwia dostęp na żądanie do puli zasobów** – według (Mell, 2011): *chmura obliczeniowa to model umożliwiający powszechny, wygodny i na żądanie dostęp przez sieć do współdzielonej puli zasobów konfigurowalnych zasobów obliczeniowych (np. sieci, serwerów, pamięci masowej, aplikacji i usług), które mogą być szybko udostępniane i zwalniane przy minimalnym wysiłku związanym z zarządzaniem bądź działaniem ze strony dostawcy usług.*
- **Chmura umożliwia skalowanie zasobów** – według (Erl, 2013): *chmura obliczeniowa jest specjalistyczną formą obliczeń rozproszonych, która wprowadza modele wykorzystania zdalnie udostępnianych, skalowalnych i mierzalnych zasobów.*

Wśród pożądanych cech chmury obliczeniowej wyróżnianych przez (Buyya, 2011) znajdują się:

- **Samoobsługowość** (Self Service) – żądanie utworzenia nowych zasobów powinno wymagać minimalnych nakładów pracy ludzkiej, możliwie żadnej. Polecenie tworzenia zasobów powinno być przetwarzane w krótkim czasie, rzędu minut.
- **Naliczanie zgodne z rzeczywistym użyciem** (Per Usage Metering and Billing) – dostawca usług chmurowych oferuje obciążanie opłatami zgodne z rzeczywistym zużyciem zasobów, naliczanym proporcjonalnie do czasu używania zasobu bądź liczby zużytych jednostek obliczeń.
- **Elastyczność** (Elasticity) – chmura obliczeniowa powinna dawać złudzenie możliwości nieskończonego skalowania na żądanie. Dodatkowo, oczekiwane jest, że dodatkowe zasoby będą alokowane i zwalniane automatycznie, w miarę zapotrzebowania.

(Mell, 2011) proponuje dodatkowo cechy:

- **Wielopodmiotowość/łączenie zasobów** (Multitenancy/Resource pooling) – chmura obliczeniowa powinna umożliwiać niezależne przydzielanie zasobów wielu podmiotom jednocześnie. Zasoby powinny być od siebie izolowane. Użytkownicy nie powinni mieć w żaden sposób możliwości obserwacji wzajemnych działań.
- **Szeroki dostęp przez sieć** (Broad network access) – zarządzanie zasobami powinno być możliwe z poziomu szerokiego zakresu zróżnicowanych urządzeń. Interfejs do zarządzania zasobami chmury powinien być ustandaryzowany, pozwalając na prostą komunikację z dowolnym środowiskiem.

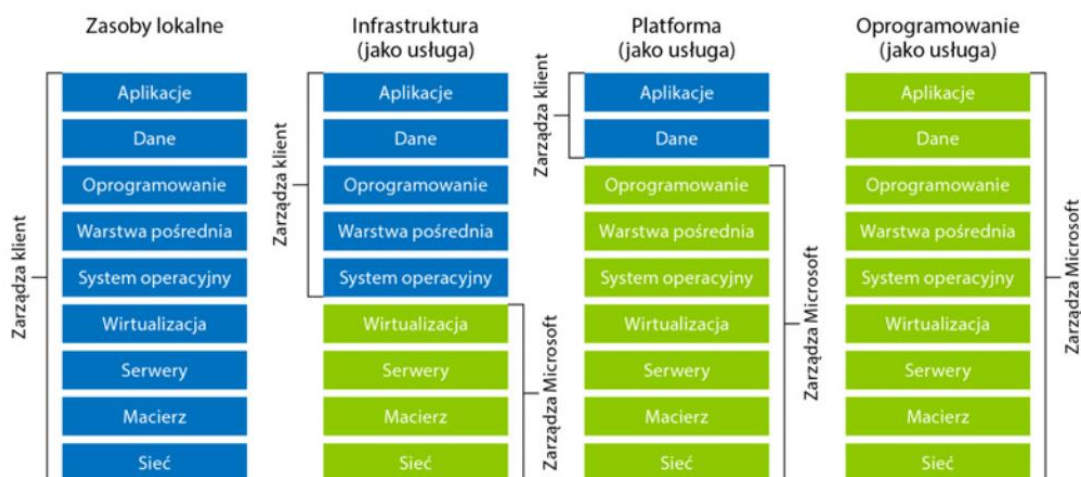
Natomiast (Erl, 2013) postuluje wymagalność dodatkowej cechy:

- **Odporność** (Resilience) – zasoby powinny móc być tak konfigurowane, by na wypadek awarii jednego bądź więcej komputera (lub nawet centrum danych), inne maszyny mogły przejmować na siebie dodatkowe obciążenie. Osoba korzystająca z usług nie powinna być w żaden sposób w stanie zaobserwować awarii sprzętu. Oznacza to, że każdy zasób chmury obliczeniowej przynajmniej potencjalnie powinien mieć możliwość redundancji – działania na kilku maszynach (a nawet na kilku centrach danych) jednocześnie.

1.3. Rodzaje usług

W architekturze chmurowej wyróżnia się trzy główne rodzaje usług: Infrastruktura jako usługa (IaaS), Platforma jako usługa (PaaS) oraz Oprogramowanie jako usługa (SaaS) (Toroman, 2020). Każdy z tych modeli oferuje różny poziom rozdzielania odpowiedzialności pomiędzy klienta a dostawcę usług chmurowych. Schemat podziału przedstawiono na rysunku 1.

- **Infrastruktura jako Usługa** (Infrastructure as a Service, IaaS) – dostawca usług chmurowych oferuje zasób, który musi zostać skonfigurowany przed użyciem. Przykładem jest maszyna wirtualna – symulowane środowisko systemu operacyjnego, w którym podmiot instaluje środowisko uruchomieniowe, definiuje role, użytkowników oraz dostarcza oprogramowanie i dane wymagane do działania programu. Do dostawcy usług należy udostępnienie zasobów w postaci przestrzeni masowej, pamięci operacyjnej, mocy obliczeniowej i połączenia z siecią do zasobu.
- **Platforma jako Usługa** (Platform as a Service, PaaS) – dostawca usług chmurowych oferuje zasób, na którym znajduje się gotowe, skonfigurowane środowisko. Przykładem jest serwis aplikacji (App Service) – wyodrębnione środowiska, w ramach których działa pojedynczy program, bez dostępu do systemu operacyjnego. Do użytkownika należy dostarczenie aplikacji oraz danych, na których aplikacja ma działać.
- **Oprogramowanie jako Usługa** (Software as a Service, SaaS) – dostawca usług chmurowych oferuje gotowe środowisko z działającym oprogramowaniem. Przykładem jest pakiet Office 365, dostarczający typowe funkcjonalności biurowe bądź usługa OneDrive, pozwalająca na przechowywanie plików w chmurze, przy minimalnej konfiguracji. Do użytkownika należy zdefiniowanie hasła. Dostawca usług dba o wszystkie pozostałe potrzeby.



Rysunek 1. Diagram przedstawiający hierarchię rodzajów usług, z wyszczególnieniem podziału odpowiedzialności pomiędzy klienta i dostawcę usług chmurowych. Źródło: (Toroman, 2020)

2. Chmura Azure

2.1. Wstęp i historia

Portal Microsoft Azure oferuje dziś (08.08.2023) ponad 200 usług w chmurze⁵, przynosząc firmie Microsoft zysk prawie 20 mld dolarów rocznie, co stanowi większą kwotę, niż zyski ze sprzedaży pakietu Office oraz systemu Windows łącznie⁶.

W 2005 roku, Ray Ozzie, po objęciu stanowiska dyrektora technicznego (Chief technology officer, CTO) w Microsoft, wystosował notatkę do pracowników firmy, podkreślając konieczność transformacji rodzaju usług świadczonych przez Microsoft. Ozzie kładł duży nacisk na potrzebę zmian, zauważając, że firma dokonuje zwrotu w działaniu mniej więcej w odstępach pięciu lat. W dokumencie został zaproponowany szereg nowych rodzajów usług, wraz z obserwacją, że wiele nowo zakładanych startupów koncentruje się na wykorzystaniu możliwości oferowanych przez Internet, co było propozycją kierunku rozwoju firmy Microsoft. Zaproponowano stworzenie szeregu usług i funkcjonalności – utworzenie „platformy usług” (ang. services platform) kierowanej do startupów działających w Internecie, w celu rozwiązania problemu skalowania w rozwijających się organizacjach. Ponadto, Ray Ozzie zaproponował model płynnego (ang. seamless) dostępu do usług, gdzie zalogowany użytkownik posiada dostęp z różnych maszyn do plików osobistych, rozrywki, kontaktów, oraz do przestrzeni roboczej⁷.

W roku 2006 został rozpoczęty projekt Red Dog. Przedsięwzięcie miało początkowo być aplikacją wewnętrzną, w celu rozwiązania problemu poświęcania dużej ilości czasu inżynierów firmy na konfigurację i zarządzanie maszynami (fizycznymi i wirtualnymi), jak również problemu współdzielenia niewykorzystanych zasobów (mocy obliczeniowej, przestrzeni masowej, etc.) wewnątrz organizacji. Ponieważ nie istniała wspólna platforma ani standardowy zestaw narzędzi, to założeniem projektu Red Dog było utworzenie środowiska pozwalającego na współdzielenie zasobów obliczeniowych^{8 9}.

W tym samym czasie, Amazon stworzył platformę chmury obliczeniowej Amazon Web Services (AWS), oferując usługę przestrzeni magazynowej Simple Storage Service (S3), oraz usługę tworzenia maszyn wirtualnych w ramach Elastic Compute Cloud (EC2)¹⁰. Firma Google zapoczątkowała analogiczną platformę, Google Cloud, w 2008 roku¹¹. Projekt Red Dog został przeorientowany w kierunku utworzenia systemu zarówno wewnątrz, w ramach działalności firmy Microsoft, jak i podmiotów zewnętrznych. W październiku roku 2008 nastąpiła pierwsza prezentacja systemu Windows Azure¹², w postaci zamkniętej bety wybranej grupie użytkowników. W 2010 roku Windows Azure został udostępniony jako usługa publiczna¹³.

⁵ <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/>

⁶ <https://www.nasdaq.com/articles/these-2-revenue-streams-account-for-60-of-microsofts-top-line>

⁷ <https://www.cnet.com/tech/services-and-software/ozzie-memo-internet-services-disruption/>

⁸ <https://www.oreilly.com/library/view/programming-windows-azure/9781449383176/ch01.html>

⁹ Programming Windows Azure: Programming the Microsoft Cloud, by Sriram Krishnan, O'Reilly Media, 2010.

¹⁰ <https://www.awsgeek.com/AWS-History/>

¹¹ <https://www.pluralsight.com/resources/blog/cloud/history-google-cloud-platform>

¹² <https://devblogs.microsoft.com/cesardelatorre/pdc-2008-disclosures-the-cloud-windows-azure-services-platform/>

¹³ <https://techcommunity.microsoft.com/t5/educator-developer-blog/the-history-of-microsoft-azure/ba-p/3574204>

Początkowo Azure udostępnił usługi PaaS, zapewniając środowisko uruchomieniowe aplikacji .NET, ale już nie maszyny wirtualne w modelu IaaS. Organizacje używające działających rozwiązań nie miały możliwości przenieść swojego środowiska na nową platformę – rozwiązaniem było jedynie tworzenie aplikacji bezpośrednio pod Windows Azure¹⁴.

W 2012 roku Azure dodał usługę maszyn wirtualnych, oferując – obok Windows Server – maszyny z dystrybucjami Linux, takich jak CentOS, SUSE oraz Ubuntu. Dodano środowiska Node.js, Java, PHP oraz Python¹⁵.

W roku 2014 nazwa Windows Azure została zmieniona na Microsoft Azure¹⁶. Nastąpiła zmiana modelu zarządzania zasobami z ASM (Azure Service Management) na ARM (Azure Resource Management). Opublikowano narzędzia do zarządzania olbrzymimi zbiorami danych, Big Data – Azure HDInsight, Azure Data Lake Store, Azure Data Lake Analytics, a także programy do analizy danych, m.in. aplikacja Power BI¹⁷.

W 2015 roku Microsoft przejęła Revolution Analytics, firmę zajmującą się rozwojem narzędzi do analizy danych z wykorzystaniem języka R. W 2016 roku Revolution R przemianowano na Microsoft R, a w chmurze Azure opublikowano usługę Microsoft R Server¹⁸. Uruchomiono wersję zapoznawczą usługi Azure Machine Learning.

W 2017 roku Azure dodał do swoich usług możliwość hostowania kontenerów Docker, rozszerzając oferowane możliwości do hostowania środowisk uruchomieniowych¹⁹. W roku 2018 opublikowano usługę Azure Kubernetes Service (AKS), pozwalającą na orkiestrację (automatyczne zarządzanie i koordynację) aplikacjami w kontenerach²⁰. Została wprowadzona usługa IoT Hub, pozwalająca na zarządzanie urządzeniami w Internecie Rzeczy²¹.

W latach 2020 – 2021 udostępniono usługi Azure Space – możliwość przetwarzania danych satelitarnych, komunikacji z satelitami okołoziemskimi oraz wykorzystywanie stacji naziemnych. W ofercie znalazła się również możliwość zdjęć satelitarnych z funkcją patrzenia „przez” chmury, z użyciem technologii SpaceEye²². Przykład zdjęć przedstawiono na rysunku 3.

W roku 2023 Microsoft opublikował usługę Azure OpenAI Service, w skład której wchodzi ChatGPT²³, generowanie obrazów, kodu źródłowego oraz tekstów w modelach językowych²⁴.

Microsoft Azure systematycznie rozwija swoją ofertę o usługi z bardzo zróżnicowanej puli dziedzin technologicznych, tworząc platformę o bardzo szerokim zakresie zastosowań. Zaczynając od środowisk uruchomieniowych, przestrzeni magazynowych i maszyn wirtualnych, poprzez narzędzia do analizy danych, a kończąc na przetwarzaniu danych kosmicznych i zaawansowanej sztucznej inteligencji, Microsoft udostępnia szeroką gamę usług, dostępną szerokiemu gronu odbiorców.

¹⁴ <https://www.forbes.com/sites/janakirammsv/2020/02/03/a-look-back-at-ten-years-of-microsoft-azure/?sh=7b4e98b44929>

¹⁵ <https://azure.microsoft.com/de-de/blog/infrastructure-as-a-service-series-virtual-machines-and-windows/>

¹⁶ <https://azure.microsoft.com/en-us/blog/upcoming-name-change-for-windows-azure/>

¹⁷ <https://powerbi.microsoft.com/en-sg/blog/happy-first-birthday-to-power-bi/>

¹⁸ <https://blog.revolutionanalytics.com/2016/01/microsoft-r-open.html>

¹⁹ <https://azure.microsoft.com/en-us/blog/microsoft-docker-investing-in-the-future-of-your-applications/>

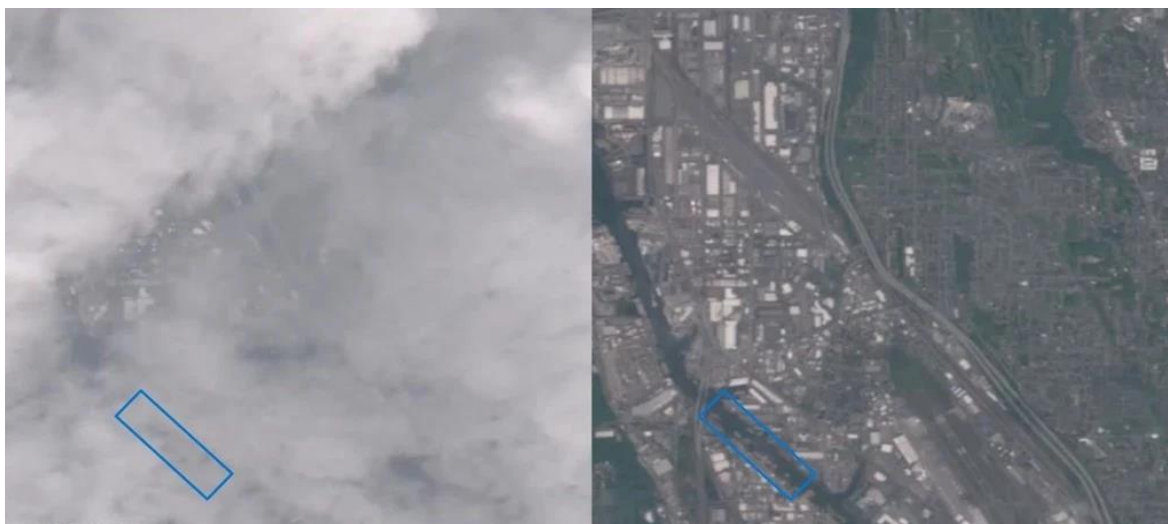
²⁰ <https://azure.microsoft.com/en-us/blog/kubernetes-on-azure/>

²¹ <https://azure.microsoft.com/en-us/blog/introducing-iot-hub-device-streams-in-public-preview/>

²² <https://azure.microsoft.com/en-us/blog/new-satellite-connectivity-and-geospatial-capabilities-with-azure-space/>

²³ <https://azure.microsoft.com/en-us/blog/chatgpt-is-now-available-in-azure-openai-service/>

²⁴ <https://azure.microsoft.com/en-us/blog/explore-the-benefits-of-azure-openai-service-with-microsoft-learn/>



Rysunek 2. Przykład zdjęcia z usługi Azure Orbital, przed (lewy obraz) i po (prawy obraz) obróbce przez SpaceEye. Źródło: <https://azure.microsoft.com/en-us/blog/new-satellite-connectivity-and-geospatial-capabilities-with-azure-space/>

2.2. Model tworzenia zasobów

Każda usługa świadczona przez Azure jest określana jako zasób – przykładem będą maszyny wirtualne, magazynowanie danych, serwis aplikacji lub równoważnik obciążenia. Niektóre usługi potrzebują innych zasobów do działania – np. każda maszyna wirtualna potrzebuje przestrzeni masowej, sieci wirtualnej, interfejsu sieciowego i grupy zabezpieczeń.

Przez pierwsze lata działania platformy Windows Azure, każdy zasób w chmurze istniał niezależnie. W 2014 roku, Microsoft dodał usługę Menadżera Zasobów (Resource Manager), wprowadzając grupy zasobów. Wszystkie dotychczasowe zasoby zostały dodane zbiorczo do grupy domyślnej. Każdy nowy tworzony zasób musiał być przypisany do nowej lub już istniejącej grupy zasobów²⁵.

Microsoft udostępnia wiele sposobów zarządzania zasobami, zgodnie z zasadą szerokiego dostępu – zróżnicowane metody komunikacji z chmurą pozwalają na integrację Azure z licznymi rodzajami urządzeń i procesów. Poniżej przedstawiono kilka możliwości:

- **Portal Azure** umożliwia sterowanie zasobami chmury z poziomu wizualnego interfejsu strony internetowej – w ten sposób możliwe jest ręczne zarządzanie zasobami z poziomu każdego urządzenia posiadającego przeglądarkę. Azure udostępnia również metody automatyzacji zadań z poziomu interfejsu graficznego. Dostępne są m.in. Azure Automation²⁶ oraz Azure DevOps²⁷.
- **Konsola PowerShell wraz z biblioteką Az**, dostępna również z poziomu portalu Azure poprzez usługę Cloud Shell²⁸, pozwala na zarządzanie chmurą Azure z poziomu użytkownika terminala. Umożliwia to tworzenie, przeglądanie, modyfikowanie i zwalnianie zasobów z poziomu systemu bez interfejsu graficznego. Biblioteka Az pozwala również na tworzenie skryptów PowerShell automatyzujących operacje w chmurze²⁹.

²⁵ <https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/deployment-models>

²⁶ <https://learn.microsoft.com/en-us/azure/automation/overview>

²⁷ <https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops>

²⁸ <https://learn.microsoft.com/en-us/azure/cloud-shell/overview>

²⁹ <https://learn.microsoft.com/en-us/powershell/azure/new-azureps-module-az?view=azps-10.1.0>

- **Azure CLI** jest multiplatformowym narzędziem w linii komend do zarządzania zasobami Azure, napisanym w języku Python³⁰. Według pierwotnych założeń Azure CLI miało zastępować PowerShell.Az w systemach operacyjnych innych niż Windows, jednak od wersji 6, PowerShell stał się multiplatformowy³¹.
- **Azure REST API** pozwala na komunikację z chmurą Azure poprzez protokół HTTP, z wykorzystaniem architektury Reprezentacyjnego Transferu Stanu (Representational State Transfer, REST). Interfejs jest docelowo skierowany do developerów tworzących osobne narzędzia do komunikacji z chmurą, takie jak Terraform. Wymiana danych może zachodzić poprzez każde urządzenie posiadające kartę sieciową³².

Poza powyżej przedstawionymi, możliwa jest również obsługa zasobów chmury z poziomu środowisk programistycznych dostarczanych przez Microsoft, takich jak Visual Studio³³ oraz Visual Studio Code³⁴. Oferowane są również oficjalne biblioteki Microsoftu do popularnych języków programowania³⁵.

2.3. Usługa Storage

W kontekście informatyki, termin *chmura* w mowie potocznej jest kojarzony z usługą przechowywania i szybkiego dostępu do danych z dowolnego urządzenia³⁶. Na rynku istnieje wiele komercyjnych rozwiązań z dziedziny zapisu plików w chmurze kierowanych do klientów indywidualnych i do firm. Spośród najpopularniejszych można wymienić usługę Microsoft OneDrive, Google Drive, Dropbox oraz iCloud³⁷.

Usługa Magazynu (ang. Storage) w chmurze Azure służy do przechowywania i udostępniania plików. Ponieważ większość usług Azure wymaga do działania pamięci masowej, Magazyn jest często wykorzystywany jako narzędzie towarzyszące innym usługom (Toroman, 2020).

Magazyn może być wykorzystywany w różnych formach, dopasowanych do potencjalnych zapotrzebowań i przewidywanych struktur danych. Wśród dostępnych usług można wymienić:

- **Azure Files** – przechowywanie plików współdzielonych pomiędzy wieloma maszynami wirtualnymi bądź maszynami użytkowników. Przykład: współdzielona baza dokumentów w projekcie firmowym.
- **Azure Blobs** – przechowywanie dużych plików binarnych pozbawionych struktury (Big Large Binary Objects, BLOB) z wymogiem dużej dostępności. Przykład: galeria zdjęć na stronie internetowej.
- **Azure Disks** – przechowywanie danych bez wymogu dostępu z zewnątrz. Przykład: pliki niezbędne do działania programu na maszynie wirtualnej.
- **Azure Queries** – przechowuje pliki związane z komunikacją asynchroniczną w kolejkach. Przykład: strona internetowa do przetwarzania dużych plików.

³⁰ <https://learn.microsoft.com/en-us/cli/azure/what-is-azure-cli>

³¹ <https://4sysops.com/archives/azure-powershell-vs-azure-cli/>

³² <https://learn.microsoft.com/en-us/rest/api/azure/>

³³ <https://learn.microsoft.com/en-us/visualstudio/azure/overview-azure-integration?view=vs-2022>

³⁴ <https://learn.microsoft.com/en-us/dotnet/azure/configure-vs-code>

³⁵ <https://learn.microsoft.com/en-us/azure/communication-services/concepts/sdk-options>

³⁶ <https://sjp.pwn.pl/sjp/chmura;2448319.html>

³⁷ <https://zapier.com/blog/best-cloud-storage-apps/>

Przechowując dane w usłudze Magazynu, klient wybiera poziom nadmiarowości (Redundancy) w przechowywaniu plików:

- **Magazyn lokalnie nadmiarowy (Locally-redundant storage, LRS)** – domyślna forma nadmiarowości, wykorzystująca kopiowanie plików pomiędzy trzy maszyny w lokalnym centrum danych. Choć prawdopodobieństwo utraty danych maleje, gdy dojdzie do uszkodzenia maszyn, metoda ta nie zabezpiecza przed kompletną awarią centrum danych. Usługa jest rekomendowana jedynie do wykorzystania przy mało wartościowych danych.
- **Magazyn strefowo nadmiarowy (Zone-redundant storage, ZRS)** – synchronizuje dane pomiędzy trzema centrami danych w jednej lokalizacji geograficznej. Zabezpiecza przed awarią danych, jednak nie zwiększa dostępności plików. Usługa jest zalecana przy ważnych plikach o małej dostępności, na przykład kopiach zapasowych.
- **Magazyn geograficznie nadmiarowy (Geo-redundant storage, GRS)** – synchronizuje dane pomiędzy centrami danych w trzech różnych regionach geograficznych. Usługa jest wykorzystywana przy wymogu wysokiej dostępności plików (np. serwis filmów, zdjęć).
- **Magazyn geograficznie i strefowo nadmiarowy (Geo-zone-redundant storage, GZRS)** – połączenie nadmiarowości geograficznej i strefowej. Usługa jest zalecana w scenariuszach krytycznych, kiedy jednocześnie jakakolwiek utrata danych jest nie do zaakceptowania, przy jednoczesnym wymogu wysokiej dostępności (np. stale aktualizowane wyniki badań z całego świata)³⁸.

2.4. Virtual Machine

Usługa maszyn wirtualnych jest jedną z podstawowych funkcji w ofercie większości chmur obliczeniowych. Tworzone jest środowisko wirtualne o zadanych parametrach i wybranej konfiguracji systemu operacyjnego. Pracę można zacząć w ciągu kilku minut od wysłania polecenia utworzenia instancji³⁹.

Działanie maszyny wirtualnej jest nieodróżnialne od działania fizycznego komputera z opcją zdalnego dostępu. Z tego względu, takie środowisko stanowi domyślną formę przenoszenia działającego wcześniej rozwiązania z fizycznego komputera do chmury podczas migracji. Innym zastosowaniem maszyn wirtualnych może być też potrzeba wykorzystania starych, niewspieranych już programów jako części rozwiązania przenoszonego w infrastrukturę chmury.

Maszyny wirtualne są również wykorzystywane jako tymczasowe środowiska, np. do testów aplikacji w domyślnym środowisku świeżo zainstalowanego systemu operacyjnego, bądź jako tymczasowe maszyny o dużej mocy obliczeniowej (np. w przypadku konieczności wykonania zbyt dużej na komputer osobisty ilości obliczeń statystycznych).

Azure umożliwia skalowanie pionowe – zmianę zasobów przydzielonych maszynie, np. liczby procesorów, przypisanej pamięci RAM lub pamięci trwałej. Taka operacja wymaga restartu instancji, jednak może być wykonana bez przymusu tworzenia nowej maszyny wirtualnej.

Obok skalowania pionowego, istnieje możliwość skalowania poziomego – zwielokrotniania istniejącej maszyny wirtualnej, by, poprzez równoważnik obciążenia (ang. load balancer), ruch był kierowany równo pomiędzy wszystkie instancje tego samego procesu. Operacja zwielokrotniania również wymaga chwilowego przestoju (w celu zachowania spójności danych), jednak tworzenie nowych instancji na

³⁸ <https://learn.microsoft.com/en-us/azure/storage/common/storage-introduction>

³⁹ <https://learn.microsoft.com/en-us/azure/virtual-machines/overview>

podstawie wcześniejszej kopii zapasowej bądź konfiguracji pozwala na zrównoważenie obciążenia bez zatrzymywania działania witryny internetowej.

Połączenie z maszyną wirtualną w chmurze Azure odbywa się poprzez Protokół Zdalnego Pulpitu (Remote Desktop Protocol, RDP) w systemie Windows, lub przez Protokół Bezpiecznej Powłoki (Secure Shell Protocol, SSH) przy użyciu dystrybucji Linux. W obydwu przypadkach konieczne jest przydzielenie maszynie wirtualnej publicznego adresu IP. Możliwe jest również skorzystanie z usługi Bastion, pozwalającej na połączenie zdalne poprzez protokół SSH lub RDP bez używania publicznego adresu IP.

2.5. App Service

Konfiguracja serwera pod działanie aplikacji internetowej jest pracochłonnym zajęciem. Coraz częściej programiści korzystają z usług w modelu Platform as a Service, gdzie środowisko uruchomieniowe jest już skonfigurowane i gotowe by zapewniać działanie oprogramowania⁴⁰.

Najpierw należy utworzyć plan działania usługi aplikacji – App Service Plan – określając system operacyjny w którym będzie działało środowisko – Linux lub Windows, region, rodzaj przydzielonych zasobów obliczeniowych i związanych z nimi kosztów, oraz nadmiarowość.

Następnie, definiowana jest usługa aplikacji – App Service – w ramach wcześniej opisanego planu. Obok możliwości wyboru środowisk uruchomieniowych .NET, Go, Java, Node.js, PHP oraz Python, użytkownik może skorzystać z kontenerów Docker, np. do uruchomienia aplikacji napisanej w języku bez wspieranego środowiska uruchomieniowego (np. Rust bądź C++), albo do uruchomienia aplikacji z dobrze zdefiniowanym, ujednoliconym środowiskiem działania.

Po skonfigurowaniu Usługi Aplikacji, podmiot przesyła pliki aplikacji do przestrzeni magazynowej związanej z usługą – może to zrobić poprzez połączenie FTP/S⁴¹, lokalne repozytorium git⁴², poprzez wysłanie paczki zip z użyciem narzędzi PowerShell bądź Azure CLI⁴³, bądź też przez wysłanie projektu do przestrzeni magazynowej znajdującej się w serwisie OneDrive lub Dropbox⁴⁴.

Usługa Aplikacji pozwala na automatyczną integrację z projektem GitHub. Wybierając opcję **Ciągłe wdrażanie** (ang. Continuous deployment) należy się zalogować na konto GitHub i wybrać projekt, który następnie zostanie zmodyfikowany, by przy aktualizacjach repozytorium zmiany były wdrażane na chmurze automatycznie.

W przypadku projektów .NET oferowana jest możliwość **Publikowania na chmurze Azure** z poziomu środowiska programistycznego Visual Studio. Taka operacja pozwala na szybką weryfikację działania aplikacji internetowej w środowisku chmury obliczeniowej.

W ramach Usługi Aplikacji można skorzystać z usługi **Wglądu w Aplikację** (ang. Application Insights). Narzędzie pozwala na uzyskiwanie kompleksowych statystyk działania, wykorzystania, wydajności oraz podglądu popularności oferowanych funkcji programu⁴⁵.

Usługa Aplikacji stanowi alternatywę dla hostowania na maszynie wirtualnej, oferując natywne środowisko uruchomieniowe bez konieczności dodatkowej konfiguracji programu.

⁴⁰ <https://learn.microsoft.com/en-us/azure/app-service/>

⁴¹ <https://learn.microsoft.com/en-us/azure/app-service/deploy-ftp>

⁴² <https://learn.microsoft.com/en-us/azure/app-service/deploy-local-git>

⁴³ <https://learn.microsoft.com/en-us/azure/app-service/deploy-run-package>

⁴⁴ <https://learn.microsoft.com/en-us/azure/app-service/deploy-content-sync>

⁴⁵ <https://learn.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview>

2.6. Azure SQL

Baza danych to ogólne określenie systemu służącego do organizacji i przechowywania danych zgodnie z określonymi regułami. Relacyjna baza danych to typ bazy danych, która przechowuje i organizuje punkty danych ze zdefiniowanymi relacjami w celu szybkiego dostępu⁴⁶. Dane są zawarte w postaci rekordów w tabelach z kolumnami o ustalonych typach i zakresach wartości, z możliwością szybkiego zwrócenia z użyciem zapytań o szczególnych warunkach przeszukiwania. W przypadku bardziej skomplikowanych struktur danych, np. danych hierarchicznych, wartości mogą być magazynowane w wielu tabelach, z przechowywaniem relacji pomiędzy rekordami.

Azure SQL jest usługą relacyjnej bazy danych w modelu Platform as a Service oferowaną w chmurze Azure. W odróżnieniu od bazy danych SQL hostowanej na maszynie wirtualnej, za przygotowanie środowiska, bezpieczeństwo i aktualizacje oprogramowania odpowiada firma Microsoft. Podmiot jest odpowiedzialny za konfigurację dostępu użytkowników oraz za przygotowanie struktury bazy danych.

Tworzenie bazy danych Azure SQL wymaga w pierwszej kolejności utworzenia serwera Azure SQL. Serwer musi mieć adres, który jest unikatowy w skali całej chmury obliczeniowej – jeśli zostanie wybrany już zajęty identyfikator, Azure informuje o błędzie. Ustalany jest również dostęp do konta administratora – możliwa jest autentykacja poprzez podanie parametrów logowania oraz poprzez konto Azure Active Directory.

W ramach konfiguracji usługi, podmiot ustala przeznaczenie bazy danych, od którego zależy poziom opłat oraz różnice w działaniu usługi. Dostępne możliwości to:

- **Przeznaczenie ogólne (general purpose)** – poziom usług przeznaczony do standardowych zastosowań, zoptymalizowany pod kątem budżetu. Wspiera rozmiary danych do 4 TB oraz zapewnia opóźnienia nie większe niż 5 – 10 ms⁴⁷.
- **Hiperskala (hyperscale)** – zaprojektowany do większości zastosowań biznesowych. Oferuje szybkie skalowanie pionowe, dużą prędkość zapisu i odczytu kopii zapasowych, oraz wsparcie danych do rozmiaru 100 TB⁴⁸.
- **Krytyczne dla działania firmy (business critical)** – poziom usług przeznaczony do aplikacji w modelu Przetwarzania Transakcji Online (Online Transaction Processing, OLTP), o wysokiej szybkości transakcji i niskich opóźnieniach operacji wejścia/wyjścia. Model OLTP jest kategorią aplikacji zorientowanych na przetwarzanie dużej liczby transakcji zachodzących jednocześnie – np. wiadomości w komunikatorze, zakupów w sklepie internetowym bądź dokonywania operacji bankowych⁴⁹. Azure SQL na poziomie usług Business Critical oferuje wysoką odporność na awarie poprzez wykorzystywanie izolowanych replik baz danych⁵⁰.

Azure SQL oferuje **dynamiczne maskowanie danych**, pozwalając administratorowi wybierać kolumny do maskowania i definiować odpowiednie zestawy reguł. Umożliwia to szeroki dostęp do danych dla pracowników, jednocześnie minimalizując ryzyko ujawnienia wrażliwych informacji.

⁴⁶ <https://azure.microsoft.com/pl-pl/resources/cloud-computing-dictionary/what-is-a-relational-database/>

⁴⁷ <https://learn.microsoft.com/en-us/azure/azure-sql/database/service-tiers-sql-database-vcare>

⁴⁸ <https://learn.microsoft.com/en-us/azure/azure-sql/database/service-tier-hyperscale>

⁴⁹ <https://www.oracle.com/database/what-is-oltp/>

⁵⁰ <https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview>

2.7. Azure Active Directory

Usługa Azure Active Directory (AAD) udostępnia zestaw narzędzi do uwierzytelniania, przechowywania tożsamości, zarządzania uprawnieniami dostępu do zasobów, oraz definiowania ról w modelu Kontroli Dostępu Opartej na Rolach (Role – Based Access Control, RBAC) (Toroman, 2020).

Poprzez AAD administrator odpowiedzialny za subskrypcje może zarządzać Katalogami Azure AD (ang. Azure AD directory), skupiającymi w sobie członków grupy, usługi oraz przypisane uprawnienia i role. Każdy katalog ma przypisaną sobie domenę w formie **nazwadomeny.onmicrosoft.com**. Użytkownicy utworzeni w ramach katalogu AAD mogą korzystać z wykupionych usług Microsoftu (np. Office 365) oraz wykonywać działania w chmurze Azure korzystając z subskrypcji udostępnionych przez administratora domeny⁵¹.

Azure Active Directory oferuje funkcję Rejestrowania Aplikacji (ang. App Registration), która jest kluczowa do zapewnienia komunikacji programów z innymi usługami chmury Azure. W ramach nowej rejestracji określana jest nazwa, wspierany zakres kont (do wyboru konta z zakresu organizacji, spoza organizacji, oraz konta osobiste w ramach Microsoftu). Po utworzeniu rejestracji konieczne jest dodanie **Uprawnień interfejsu API** (ang. API permissions), by określić, w jakim zakresie aplikacja ma mieć dostęp do usług Azure.

Po przypisaniu uprawnień, w zakładce **Certyfikaty i klucze tajne** (ang. Certificates & secrets), podmiot może wygenerować pary identyfikatorów i sekretów klienta. Każda para może mieć ustalony czas wygasania, z domyślnym rekomendowanym terminem 180 dni. Wartość klucza tajnego przestaje być do wglądu z poziomu Azure AD po pierwszym wyświetleniu.

Klucz tajny jest wykorzystywany w aplikacji przy użyciu protokołu OAuth 2.0. Wykonywane jest żądanie o dostęp do zasobu poprzez wysłanie identyfikatora klienta w postaci GUID oraz klucza tajnego. Te dane są kierowane pod odpowiednią ścieżkę w domenie microsoftonline.com, unikalnej dla każdego katalogu AAD. W tym kontekście, katalog AAD jest określany jako dzierżawca (ang. tenant). Przy poprawnej autentykacji oraz autoryzacji, zwracany jest token JWT (JSON Web Token). Obiekt stanowi jednoznaczny autoryzację do wykonywania przyznanych działań⁵².

Usługa Azure Active Directory pozwala na sprawne organizowanie kont i aplikacji w chmurze Azure. Serwis AAD jest wykorzystywany zarówno przez administratora organizacji do zarządzania dostępem oraz uprawnieniami grupy, jak i przez indywidualnego użytkownika, konfigurującym komunikację pomiędzy aplikacjami w chmurze.

⁵¹ <https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/whatis>

⁵² <https://learn.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-client-creds-grant-flow>

2.8. Power BI

Power BI jest komercyjnym zestawem narzędzi pozwalającym na przygotowywanie i udostępnianie analiz i interaktywnych raportów, głównie w zastosowaniach biznesowych⁵³. Usługa pozwala na pobieranie danych ze zróżnicowanych źródeł, od baz danych Access, SQL Server, Azure SQL lub MySQL, poprzez zapytania do usług online takich jak Google Big Data aż do plikowych baz danych, jak XML, JSON, CSV lub nawet poprzez parsowanie danych z tabel z plików PDF.

Microsoft udostępnia następujące narzędzia i usługi:

- **Power BI Desktop** – aplikacja okienkowa w systemie Windows do tworzenia i obróbki danych i raportów⁵⁴.
- **Power BI Service** – usługa chmurowa w modelu Software as a Service pozwalająca na tworzenie i udostępnianie raportów w sieci. Ponieważ serwis współpracuje z **Power BI Desktop**, może być wykorzystywany jako docelowe miejsce składowania raportów utworzonych w aplikacji okienkowej⁵⁵.
- **Power BI Embedded** – usługa w ramach Azure pozwalająca na osadzanie wykresów Power BI w zewnętrznych aplikacjach⁵⁶.

Istotną częścią programu jest edytor Power Query, pozwalający na obróbkę i pielęgnację danych do ostatecznego wykresu. Z poziomu graficznego edytora, twórca raportu może szybko wybrać konieczne zależności do zaprezentowania na wykresie. Wewnątrz programu możliwe jest również korzystanie z funkcjonalności języków R oraz Python, będących powszechnymi narzędziami w analizie danych.

Aplikacja umożliwia automatyczne grupowanie danych w raporcie końcowym, pozwalając na wyświetlanie wykresów ograniczanych wyłącznie do interesujących informacji. Możliwa jest również automatyczna aktualizacja posiadanych danych, poprzez aktywne połączenie ze źródłem danych. Użytkownicy wyświetlający raport mogą podejmować decyzje na podstawie najświeższych informacji.

⁵³ <https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>

⁵⁴ <https://learn.microsoft.com/en-us/power-bi/fundamentals/desktop-what-is-desktop>

⁵⁵ <https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-service-overview>

⁵⁶ <https://learn.microsoft.com/en-us/power-bi/developer/embedded/embedded-analytics-power-bi>

3. Platforma ASP.NET Core

3.1. Wstęp i historia

W roku 1996 Microsoft wydał pierwszą wersję Active Server Pages (ASP), będącym językiem skryptowym stron internetowych, wykonywanym po stronie serwera. Podobnie do PHP, ASP służył do dynamicznego generowania stron internetowych na podstawie skryptów `.asp`. Pliki składały się z kombinacji HTML i skryptów w językach VBScript, JScript lub PerlScript (do wyboru przez programistę, deklarowane na początku dokumentu). Serwer, po otrzymaniu żądania wyświetlenia strony, używał interpretera ASP do przygotowania strony internetowej na podstawie skryptów `.asp` oraz bieżących danych. Wygenerowana strona była wysyłana w odpowiedzi użytkownikowi.

ASP został zastąpiony w 2002 roku przez ASP.NET, będącego częścią frameworka .NET Framework. Językami obsługującymi ASP.NET były C# oraz VB.NET (do wyboru przez programistę). Tym razem witryna internetowa była już kompilowana przed wysłaniem na serwer. Nastąpił również podział kodu źródłowego, na pliki `.aspx` zawierające układ strony oraz pliki `.cs` (lub `.vb`), zawierające resztę funkcjonalności programu webowego. Istotną częścią ASP.NET była możliwość korzystania z pakietu Web Forms, oferującego zestaw formularzy do stron internetowych analogicznych do formatek Win Forms w środowisku okienkowym⁵⁷.

W roku 2009 miała swój debiut nowa wersja ASP.NET w wersji 3.5, implementująca narzędzie ASP.NET MVC 1.0. W odróżnieniu od Web Forms, ASP.NET MVC korzystał ze wzorca architektonicznego Model–Widok–Kontroler (Model–View–Controller, MVC), mającego na celu dodatkowo rozdzielić odpowiedzialności w projekcie⁵⁸.

W roku 2010 opublikowano ASP.NET MVC 3, wraz z systemem składni Razor, będący językiem szablonów html, zastępującym format stron `.aspx`. Razor implementował znacznie bardziej zwięzły format używania zmiennych oraz instrukcji warunkowych. Również system składni Razor odchodził od modelu Web Forms, zastępując go standardowym zestawem formatek dostarczanych przez HTML⁵⁹.

W roku 2012 wdrożono .NET Framework 4.0, wraz z ASP.NET MVC 4.0. Jedną z głównych zmian było dodanie ASP.NET Web API, umożliwiającego komunikację z witryną internetową poprzez zapytania i odpowiedzi HTTP, z pominięciem warstwy wizualnej⁶⁰. Format Web API umożliwiał rozdział aplikacji internetowej na warstwę witryny (ang. frontend) oraz zaplecza (ang. backend), gdzie witryna stanowiła aplikację działającą po stronie klienta, natomiast zaplecze działało po stronie serwera. Dodano również funkcjonalność SignalR, umożliwiającą komunikację asynchroniczną pomiędzy klientem a serwerem w czasie rzeczywistym⁶¹.

W roku 2016 wydano międzyplatformowy framework o publicznie dostępnym kodzie źródłowym (ang. open-source, otwarte źródło) .NET Core 1.0, który po raz pierwszy był dostępny także na inne platformy niż Windows. Przygotowana wersja ASP.NET 5.0 (przemianowana na ASP.NET Core 1.0) wraz z MVC 6.0 (przemianowanym na ASP.NET MVC Core 1.0) pozwalała na budowanie aplikacji webowych możliwych do uruchamiania w środowisku Linux oraz Docker, zarówno na maszynach wirtualnych, jak i serwisach aplikacyjnych⁶².

⁵⁷ <https://www.dotnetcurry.com/aspnet/1492/aspnet-history-part-1>

⁵⁸ <https://weblogs.asp.net/scottgu/asp-net-mvc-1-0-release-candidate-now-available>

⁵⁹ <https://weblogs.asp.net/scottgu/introducing-asp-net-mvc-3-preview-1>

⁶⁰ <https://learn.microsoft.com/en-us/aspnet/whitepapers/mvc4-release-notes>

⁶¹ <https://learn.microsoft.com/en-us/aspnet/signalr/overview/older-versions/tutorial-getting-started-with-signalr-and-mvc-4>

⁶² <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core>

Zostało również dodane narzędzie CLI, `dotnet`, pozwalające na sprawne zarządzanie projektem z poziomu wiersza poleceń⁶³.

W roku 2019 opublikowano .NET Core 3.0, wraz ASP.NET Core 3.0, zawierającym szereg nowych funkcjonalności. Udostępniono system szablonów Blazor, oferujący podobne zastosowania do Razor, jednak z możliwością wykonywania skryptów C# po stronie klienta. Wdrożono również gRPC, implementację standardu do komunikacji zdalnej (Remote Procedure Call, RPC), szeroko używanego w architekturze mikroservisów⁶⁴. Sam .NET Core dodawał wsparcie frameworków Windows Forms oraz WPF (wcześniej dostępnych jedynie z poziomu .NET Framework), co jednak nie umożliwiało uruchamianie aplikacji okienkowych z poziomu systemów Linux⁶⁵.

W roku 2020 był wydany .NET 5, wraz z ASP.NET Core 5.0 (wersja 4 została pominięta, w celu zachowania rozróżnienia od .NET Framework 4)⁶⁶. Dodane zostały funkcjonalności języka C# w wersji 9.0. Nowy framework był reklamowany jako zunifikowany system do wielu zastosowań, wśród których były wymienione tworzenie aplikacji okienkowych, webowych, chmurowych, mobilnych, gier, Internetu rzeczy oraz wykorzystania uczenia maszynowego⁶⁷. Diagram poglądowy przedstawiono na rysunku 3.

.NET – A unified platform



Rysunek 3. Diagram reklamujący możliwości .NET 5 jako zunifikowanej platformy ogólnego zastosowania. Źródło: <https://devblogs.microsoft.com/dotnet/introducing-net-5/>

W roku 2021 ogłoszono wydanie .NET 6.0. Nowy framework dodawał wsparcie języka C# w wersji 10.0, funkcje obsługujące protokół HTTP/3, wprowadzone były liczne poprawki przyspieszające działanie programów. Dodany został również nowy system .NET MAUI, międzyplatformowy framework do tworzenia aplikacji graficznych na systemy Windows 10/11, macOS oraz iOS i Android. Systemy Linux dalej nie miały wsparcia w obsłudze programów okienkowych, jakkolwiek były używane jako platformy do wykonywania aplikacji linii komend i webowych⁶⁸.

W 2022 roku ogłoszono wydanie .NET 7, implementującym cechy języka C# 11. Nowa wersja frameworka była reklamowana jako lepiej zoptymalizowana pod kątem prędkości działania programów, dodany

⁶³ <https://devblogs.microsoft.com/dotnet/announcing-net-core-rc2/>

⁶⁴ <https://www.codeproject.com/Articles/5164467/Getting-Started-with-gRPC-Client-and-Server-using>

⁶⁵ <https://devblogs.microsoft.com/dotnet/announcing-net-core-3-0/>

⁶⁶ <https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-5>

⁶⁷ <https://devblogs.microsoft.com/dotnet/introducing-net-5/>

⁶⁸ <https://devblogs.microsoft.com/dotnet/announcing-net-6/>

został również szereg usprawnień do istniejących bibliotek. Ogłoszono rozwinięte wsparcie tworzenia oprogramowania na systemach Ubuntu Linux, jak również na procesorach z rodziny ARM⁶⁹.

Pod koniec 2023 roku ma zostać wydany .NET 8, wprowadzający szereg usprawnień do bieżących bibliotek. Ma zostać dodana obsługa typów 128-bitowych (jak Int128 oraz UInt128), poprawiona integracja z Dockerem oraz szereg implementacji nowych funkcji kryptograficznych⁷⁰.

W ciągu ostatnich 20 lat, framework .NET przeszedł drogę od biblioteki funkcji do rozwoju aplikacji w systemie Windows, aż do uniwersalnego systemu tworzenia zróżnicowanych produktów całego szeregu platform.

3.2. Architektura MVC

Architektura MVC podąża za zasadą **separacji odpowiedzialności** (ang. separation of concerns), stanowiącej, że komponenty aplikacji powinny być podzielone na sekcje, z których każda zajmuje się innym problemem⁷¹. W strukturze MVC prowadzony jest rozdział pomiędzy wyglądem a funkcjonalnością strony, według następującego schematu:

- **Widok** – pliki szablonów związane z wyglądem i graficznym interfejsem użytkownika na stronie internetowej.
- **Kontroler** – zestaw klas odpowiedzialnych za:
 - przetwarzanie zapytań do strony internetowej,
 - uzyskiwaniem danych z modelu,
 - przygotowywanie odpowiedzi w postaci wyrenderowanych plików na podstawie szablonów z Widoku.
- **Model** – zestaw klas służący do reprezentacji danych oraz implementacji logiki biznesowej związanej z problemem⁷².

Tak przeprowadzony podział obowiązków pozwala na implementowanie zarówno monolitycznych aplikacji, w których renderowanie strony internetowej zachodzi po stronie serwera (jak w przypadku ASP.NET MVC z szablonami Razor), jak również wdrażanie rozwiązań z podziałem na witrynę i zaplecze (frontend/backend), gdzie po stronie aplikacji ASP.NET implementowany jest kontroler obsługujący i wysyłający komunikaty HTTP z danymi w postaci JSON.

3.3. Razor

Razor jest systemem znaczników pozwalającym na osadzanie kodu napisanego w C# na stronach internetowych, używanym w projektach ASP.NET MVC, w Razor Pages oraz aplikacjach Blazor. Składnia języka składa się z kodu HTML, kodu C# oraz znaczników Razor⁷³. Przy generowaniu odpowiedzi na zapytanie, kontroler ASP.NET parsuje plik Razor, wykonuje cały kod w C# i generuje końcowy dokument HTML wysyłany w odpowiedzi. Silnik renderujący strony Razor wykorzystuje dane przesyłane z kontrolera pod postacią zmiennej @Model, do którego komponentów odwołuje się poprzez standardową składnię C#. Możliwe jest również użycie dowolnej funkcji zdefiniowanej wewnątrz projektu bądź w bibliotekach. (Freeman, 2013)

⁶⁹ <https://devblogs.microsoft.com/dotnet/announcing-dotnet-7/>

⁷⁰ <https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-8#windows-presentation-foundation>

⁷¹ <https://medium.com/@evon.dong3/key-ideas-about-separation-of-concerns-f971bdb8bd6b>

⁷² <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-controller>

⁷³ <https://learn.microsoft.com/en-us/aspnet/core/mvc/views/razor>

4. Skrypty DevOps

4.1. Wstęp

Wytwarzanie oprogramowania jest procesem żmudnym i wieloetapowym, z często zmieniającymi się potrzebami użytkowników. Historyczny zestaw praktyk, określany zbiorczo jako Waterfall, opiera się na trzech głównych zasadach: niskim zaangażowaniu klienta, silnej dokumentacji i sekwencyjnych etapach rozwoju projektu⁷⁴. Klient zamawiający oprogramowanie przedstawia szereg oczekiwań wobec programu, z których tworzony jest Dokument Specyfikacji Wymagań (ang. Software Requirements Specification, SRS).

Dokument Specyfikacji Wymagań jest oficjalnym oświadczeniem opisującym listę oczekiwań wobec wdrażanego systemu. Zawiera szczegółową specyfikację wymagań interesantów, opisywanych zazwyczaj za pomocą języka naturalnego. Każda żądana funkcjonalność jest przedstawiana w postaci jednoznacznej, jasno sformułowanej specyfikacji określającej działanie końcowego produktu, jednak niedotykającej szczegółów implementacyjnych (Sommerville, 2016).

Model Waterfall jest często krytykowany przez zbyt intensywne skupianie się na dokumentacji, kosztem komunikacji z klientem, który często nie jest w stanie dokładnie określić wymagań projektu przed procesem tworzenia oprogramowania. Dodatkowo, Waterfall zamyka się na zmianę trajektorii działania w trakcie tworzenia projektu, poprzez założenie, że fazy planowania i wdrażania są oddzielnymi, następującymi po sobie procesami. Metodyka Agile powstała jako bezpośrednia alternatywa do procesu Waterfall⁷⁵.

W 2001 roku powstał Manifest Zwinnego Wytwarzania Oprogramowania (ang. The Agile Manifesto):

Odkrywamy nowe metody programowania dzięki praktyce w programowaniu i wspieraniu w nim innych. W wyniku naszej pracy, zaczęliśmy bardziej cenić:

- *Ludzi i interakcje od procesów i narzędzi*
- *Działające oprogramowanie od szczegółowej dokumentacji*
- *Współpracę z klientem od negocjacji umów*
- *Reagowanie na zmiany od realizacji założonego planu.*

Oznacza to, że elementy wypisane po prawej są wartościowe, ale większą wartość mają dla nas te, które wypisano po lewej⁷⁶.

Manifest skupiał w sobie idee stojące za powstającymi nowymi metodykami wytwarzania oprogramowania, zbiorczo określanymi jako **Zwinne** (ang. Agile). Wśród założeń jest, między innymi, gotowość na przygotowywanie nowych wersji produktu w krótkich odcinkach czasu.

⁷⁴ <https://hygger.io/blog/waterfall-process-model-principles/>

⁷⁵ <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/>

⁷⁶ <https://agilemanifesto.org/iso/pl/manifesto.html>

4.2. DevOps

Pojęcie Operacji Developerskich (ang. Developer Operations, DevOps) skupia się na kombinacji wytwarzania oprogramowania z procesem jego wdrażania i zarządzania. Ciągła konieczność rozwoju oprogramowania – ze względu na zmieniające się środowisko technologiczne, zapotrzebowanie na nowe cechy produktu, znajduwane błędy w istniejącej aplikacji – sprawiają, że wdrożenia są coraz częstszym procesem. Z tego względu jest stawiany duży nacisk na automatyzację wszystkich procesów towarzyszących wdrożeniom, tj. testowaniu, wdrażaniu i monitorowaniu produktu⁷⁷.

Jedną z implementacji założeń filozofii DevOps jest proces ciągłej integracji i ciągłego dostarczania (ang. Continuous Integration / Continuous Delivery, CI/CD), związany z utworzonym środowiskiem i zestawem skryptów uruchomieniowych wywoływanych automatycznie podczas aktualizacji repozytorium kodu źródłowego. Przy każdym wysłaniu zmian, serwer kompiluje kod wykonywalny, przeprowadza szereg testów sprawdzających poprawność działania aplikacji i wdraża aplikację na serwer testowy. Jeśli nie zostanie wykryty błąd na żadnym z etapów, nowa wersja projektu jest oznaczana jako poprawna.

Nowoczesne chmury obliczeniowe dostarczają szereg narzędzi usprawniających operacje monitorowania i wdrażania – przykładem jest zestaw usług Azure DevOps⁷⁸ w ramach chmury Azure, narzędzia AWS CodeBuild⁷⁹, CodePipeline⁸⁰, CodeDeploy⁸¹ oraz CloudFormation⁸² w ramach Amazon Web Services, bądź Google Cloud Build⁸³ i Google Cloud Deployment Manager⁸⁴ w ramach Google Cloud Platform.

4.3. PowerShell

PowerShell to interpreter poleceń oraz język skryptowy wydany przez Microsoft w 2006 roku⁸⁵. Początkowo tylko na system Windows (pod nazwą Windows PowerShell), w roku 2016 projekt został udostępniony na GitHub na licencji open-source, wraz z wydaniem na systemy Linux oraz macOS. Wydana zostaje wersja PowerShell Core, zbudowana i wykorzystująca funkcje .NET Core Common Language Runtime, w odróżnieniu od Windows PowerShell, korzystającej z ówczesnej wersji .NET CLR⁸⁶.

W roku 2020, wraz z ogłoszeniem PowerShell w wersji 7.0, zostało zarzucone rozróżnienie na PowerShell oraz PowerShell Core – nowa wersja, zbudowana na .NET 5, była natywnie multiplatformowa⁸⁷.

PowerShell umożliwia zarządzanie systemem Windows z poziomu linii komend, zastępującym funkcyjnie program Command Prompt. Komunikacja z PowerShell zachodzi poprzez bezpośrednie komendy do konsoli programu, możliwe jest również przygotowywanie skryptów wykonywalnych (wyróżnianych rozszerzeniem .ps1). Dzięki integracji z platformą .NET, PowerShell umożliwia wywoływanie metod zdefiniowanych w skryptach C#, ma również dostęp do funkcji biblioteki .NET.

⁷⁷ <https://about.gitlab.com/topics/devops/>

⁷⁸ <https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops>

⁷⁹ <https://docs.aws.amazon.com/codebuild/latest/userguide/welcome.html>

⁸⁰ <https://docs.aws.amazon.com/codepipeline/>

⁸¹ <https://docs.aws.amazon.com/codepipeline/latest/userguide/action-reference-CodeDeploy.html>

⁸² <https://docs.aws.amazon.com/codepipeline/latest/userguide/action-reference-CloudFormation.html>

⁸³ <https://cloud.google.com/build/docs>

⁸⁴ <https://cloud.google.com/deployment-manager/docs>

⁸⁵ <https://devblogs.microsoft.com/powershell/windows-powershell-m Monad-has-arrived/>

⁸⁶ <https://devblogs.microsoft.com/powershell/powershell-on-linux-and-open-source-2/>

⁸⁷ <https://devblogs.microsoft.com/powershell/announcing-powershell-7-0/>

Język skryptowy PowerShell jest szeroko wykorzystywany przy zarządzaniu i automatyzacji poleceń w systemie Windows oraz chmurze Azure, umożliwia również działania technologii innych dostawców, jak AWS, VMWare oraz Google Cloud⁸⁸.

4.4. Moduł PowerShell Az oraz Azure CLI

Biblioteka Az jest zbiorem funkcji pozwalających na zarządzanie zasobami Azure z poziomu linii komend i skryptów PowerShell. Az może być wykorzystywany do działań w Azure Cloud Shell, z komputerów osobistych z systemami Windows, macOS oraz Linux, kontenerów Docker oraz do wykonywania skryptów CI/CD w usłudze GitHub Actions.

Moduł PowerShell Az pozwala nie tylko na tworzenie i zwalnianie zasobów w chmurze Azure, ale też skalowanie, monitorowanie oraz przesyłanie plików z komputera lokalnego do zasobów, np. w celu wdrożenia aplikacji webowej na App Service lub przywrócenia danych z kopii zapasowej w bazie danych Azure SQL⁸⁹.

Azure Command – Line Interface jest międzyplatformowym programem linii komend napisanym w Pythonie. Podobnie do PowerShell Az, program CLI pozwala na wykonywanie działań na chmurze Azure z poziomu wiersza poleceń.

Program Azure CLI został utworzony w 2012 roku jako międzyplatformowe narzędzie do obsługi chmury Azure z linii komend. W roku 2016 została wydana wersja 2.0, napisana w języku Python. Azure CLI jest zaprojektowany, by współpracować z innymi programami w konsoli Bash, będącym domyślnym wierszem poleceń na systemach Linux i macOS⁹⁰.

4.5. Terraform

Terraform to narzędzie open-source typu Infrastruktura jako Kod (Infrastructure as a Code, IaC), służące do automatycznego tworzenia i zwalniania zasobów w chmurze⁹¹. Poprzez zestaw definicji w języku HCL określany jest pożądany układ zasobów i ich konfiguracja. Podczas wdrażania Terraform ustala zestaw kroków do osiągnięcia stanu docelowego z poziomu stanu zastanego. Inaczej niż moduły PowerShell Az lub Azure CLI, Terraform ma własność idempotentności: wielokrotne zastosowanie działania przynosi skutek identyczny jak jednokrotne⁹².

Terraform jest uniwersalnym narzędziem projektowanym do wdrażania rozwiązań na wielu platformach chmurowych – m.in. Azure, AWS, GCP oraz Terraform Cloud. Umożliwia on również wdrożenia na chmurach prywatnych, korzystających z rozwiązań takich VMware vSphere lub OpenStack⁹³.

⁸⁸ <https://learn.microsoft.com/en-us/powershell/scripting/overview>

⁸⁹ <https://learn.microsoft.com/en-us/powershell/azure/what-is-azure-powershell>

⁹⁰ <https://learn.microsoft.com/en-us/cli/azure/get-started-with-azure-cli>

⁹¹ <https://learn.microsoft.com/en-us/azure/developer/terraform/overview>

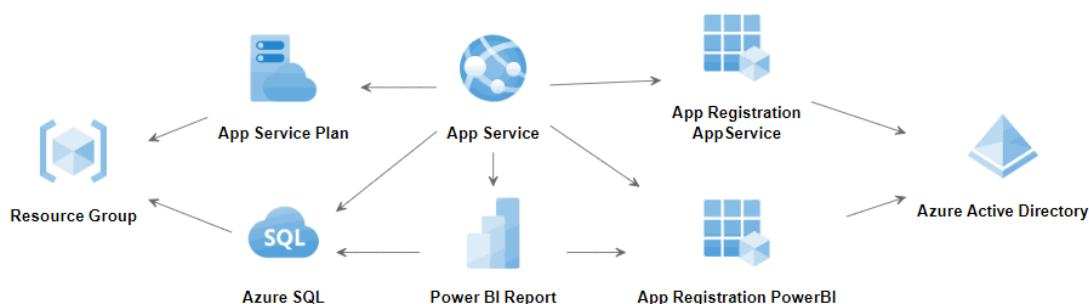
⁹² <https://aws.plainenglish.io/what-is-idempotency-in-terraform-and-ansible-ebc2ef2e4234>

⁹³ <https://cloudify.co/what-is-terraform-automation/>

5. Opis pracy

5.1. Wstęp

Witryna internetowa jest hostowana w usłudze App Service, z konfiguracją definiowaną w App Service Plan. Dodatkowo, witryna korzysta z raportu Power BI udostępnianego przez usługę Power BI App, z usługi Azure SQL oraz z Azure Active Directory. Raport Power BI również korzysta z bazy danych Azure SQL. Diagram zależności usług przedstawiono na rysunku 4.



Rysunek 4. Diagram zależności usług w infrastrukturze witryny internetowej. Opracowanie własne.

5.2. Zarys opisu projektu

Opis projektu składa się z czterech rozdziałów. W rozdziale szóstym przedstawiono, jak korzystając ze skryptów Python, PowerShell oraz z bazy danych SQL Server, wykonano operacje zbierania danych z usługi DataBank strony WorldBank. Tak utworzoną bazę danych wyeksportowano do postaci zgodnej z Azure SQL.

Rozdział siódmy przedstawia sposób przygotowania prezentacji na stronach raportu używając programu Power BI Desktop, wykorzystując zgromadzone dane. Wykresy przedstawiały zmiany Światowych Wskaźników Rozwoju na przełomie ostatnich 60 lat.

Rozdział ósmy opisuje, jak używając środowiska Visual Studio 2022, utworzono witrynę internetową z wykorzystaniem frameworka ASP.NET Core MVC w .NET 6.0. Witryna była tworzona w celu zbierania informacji o świadomości na zmian na świecie, wraz z prezentowaniem przekonujących statystyk na temat rozwoju – analogicznie do strony gapminder.com.

W rozdziale dziewiątym przedstawiono opis tworzenia zestawu skryptów wdrożeniowych w języku PowerShell, pozwalających na automatyczne utworzenie witryny internetowej w chmurze Azure. Skorzystano z narzędzia Terraform w celu konfiguracji infrastruktury, z biblioteki PowerShell Az umożliwiającej przesłanie plików do zasobów Azure oraz z bibliotek i interfejsu REST API pozwalających na zarządzanie usługą Power BI.

6. Zbieranie danych

6.1. Wstęp

Założenia pracy skupiały się na prezentacji danych udostępnianych przez witrynę World Bank⁹⁴. Serwis udostępnia, m.in. Wskaźniki Rozwoju Świata (ang. World Development Indicators, WDI) na przestrzeni lat 1960 – 2021, opracowane na podstawie oficjalnie uznanych źródeł międzynarodowych. WDI prezentują najbardziej aktualne i dokładne dane dotyczące rozwoju globalnego oraz walki z ubóstwem⁹⁵.

6.2. Opis wskaźników

Spośród danych wyróżniono 6 wskaźników rozwoju (opisy tłumaczone bezpośrednio z informacji udostępnianych przez World Bank):

- **GNI per capita (constant 2015 US\$)** – Dochód narodowy brutto, DNB (dawniej PNB) to suma wartości dodanej przez wszystkich producentów będących rezydentami plus wszelkie podatki od produktów (pomniejszone o subsydia) nieuwzględnione w wycenie produkcji plus wpływy netto dochodu pierwotnego (wynagrodzenia pracowników i dochody z tytułu własności) z zagranicy. Dane w cenach stałych z 2015 r., wyrażone w dolarach amerykańskich⁹⁶.
- **Poverty headcount ratio at \$2.15 a day (2017 PPP) (% of population)** – Wskaźnik zagrożenia ubóstwem przy 2,15 USD dziennie to odsetek ludności żyjącej za mniej niż 2,15 USD dziennie w cenach skorygowanych o siłę nabywczą z 2017 r. W wyniku zmian w kursach wymiany PPP, wskaźniki ubóstwa w poszczególnych krajach nie mogą być porównywane ze wskaźnikami ubóstwa zgłoszonymi we wcześniejszych edycjach⁹⁷.
- **Population, total** – Całkowita liczba ludności opiera się na faktycznej definicji populacji, która obejmuje wszystkich mieszkańców bez względu na status prawny lub obywatelstwo. Podane wartości są szacunkami z połowy roku⁹⁸.
- **Life expectancy at birth, total (years)** – Oczekiwana długość życia w chwili narodzin wskazuje liczbę lat, jaką żyłby noworodek, gdyby dominujące wzorce umieralności w chwili jego narodzin pozostały takie same przez całe jego życie⁹⁹.
- **Mortality rate, infant (per 1,000 live births)** – Współczynnik umieralności niemowląt to liczba niemowląt umierających przed ukończeniem pierwszego roku życia na 1000 żywych urodzeń w danym roku¹⁰⁰.
- **Fertility rate, total (births per woman)** – Całkowity współczynnik dzietności reprezentuje liczbę dzieci, które kobieta urodziłaby, osiągając koniec okresu rozrodczego i rodząc dzieci z intensywnością obserwowaną wśród kobiet w danym roku¹⁰¹.

⁹⁴ <https://databank.worldbank.org>

⁹⁵ <https://datatopics.worldbank.org/world-development-indicators/>

⁹⁶ <https://databank.worldbank.org/metadataglossary/world-development-indicators/series/NY.GNP.PCAP.KD>

⁹⁷ <https://databank.worldbank.org/metadataglossary/world-development-indicators/series/SI.POV.DDAY>

⁹⁸ <https://databank.worldbank.org/metadataglossary/world-development-indicators/series/SP.POP.TOTL>

⁹⁹ <https://databank.worldbank.org/metadataglossary/world-development-indicators/series/SP.DYN.LE00.IN>

¹⁰⁰ <https://databank.worldbank.org/metadataglossary/world-development-indicators/series/SP.DYN.IMRT.IN>

¹⁰¹ <https://databank.worldbank.org/metadataglossary/world-development-indicators/series/SP.DYN.TFRT.IN>

6.3. Opis struktury bazy danych

Ponieważ strona internetowa nie udostępnia kopii zapasowej bazy danych SQL, wymagane były dodatkowe działania w celu uzyskania ustrukturalizowanych danych w formie tabel w SQL Server. World Bank udostępnia API pozwalające na wysyłanie zapytań na temat wskaźników¹⁰². Pobieranie danych odbywało się poprzez tworzenie zapytań w skryptach w języku PowerShell, a następnie zapis do tymczasowych plików JSON. Obróbka i zapis danych do bazy danych zostały wykonane z użyciem skryptów w języku Python.

Dane były przechowywane w bazie danych SQL Server, wyeksportowanej później do Azure SQL. Baza danych była zarządzana poprzez środowisko SQL Server Management Studio, SSMS.

Utworzona została baza danych WORLDBANK wraz z tabelami:

- **INDICATORS** – zebrane wskaźniki rozwoju,
- **INDICATOR_DESCRIPTIONS** – opisy wskaźników, wraz z kodami identyfikującymi,
- **REGIONS** – opisy regionów, w języku polskim i angielskim.

Tabela **INDICATORS** zawiera następujące kolumny:

- **INDICATOR_ID** – identyfikator wskaźnika (w postaci kolejnych liczb całkowitych dodatnich),
- **REGION_CODE** – dwuliterowy kod regionu,
- **YEAR** – rok, którego dotyczą pomiary, w zakresie 1960-2021,
- **GNI_PER_CAPITA_CONSTANT_2015_USUSD** – dochód narodowy brutto w danym roku,
- **POVERTY_HEADCOUNT_RATIO_...** – wskaźnik zagrożenia ubóstwem,
- **POPULATION_TOTAL** – całkowita liczba ludności,
- **LIFE_EXPECTANCY_AT_BIRTH_TOTAL_YEARS** – oczekiwana długość życia przy narodzinach,
- **MORTALITY_RATE_INFANT_PER_1000_LIVE_BIRTHS** – współczynnik śmiertelności niemowląt,
- **FERTILITY_RATE_TOTAL_BIRTHS_PER_WOMAN** – współczynnik dzietności.

Tabela **INDICATOR_DESCRIPTIONS** zawiera następujące kolumny:

- **INDICATOR_ID** – identyfikator wskaźnika (w postaci kodu wskaźnika, używany przez World Bank),
- **INDICATOR_COLUMN** – nazwa kolumny wykorzystanej w tabeli **INDICATORS**,
- **INDICATOR_DESCRIPTION** – opis wskaźnika w języku angielskim, pobrany z oficjalnych opisów World Bank.

¹⁰² <https://datahelpdesk.worldbank.org/knowledgebase/articles/889386>

Tabela **REGIONS** zawiera następujące kolumny:

- **REGION_CODE** – dwuliterowy kod regionu,
- **REGION_ISO_CODE** – trójliterowy kod regionu zgodny z ISO 3166-1 alpha-3,
- **REGION_NAME_EN** – angielska nazwa regionu, pobrana z World Bank,
- **REGION_IS_COUNTRY** – czy region powinien być traktowany jako państwo,
- **INCOME_GROUP_EN** – grupa zarobkowa regionu (w języku angielskim),
- **SPECIAL_NOTES** – dodatkowy komentarz World Bank (w języku angielskim),
- **COUNTRY_REGION_EN** – wyróżniony obszar geograficzny państwa (w języku angielskim),
- **REGION_NAME_PL** – polska nazwa regionu/kraju, według Głównego Urzędu Statystycznego¹⁰³,
- **INCOME_GROUP_PL** – grupa zarobkowa regionu (w języku polskim, tłumaczenie własne),
- **COUNTRY_REGION_PL** – wyróżniony obszar geograficzny państwa (w języku polskim, tłumaczenie własne).

Dodatkowo, zostały utworzone cztery tabele do przechowywania danych z witryny internetowej.

Tabela **QUESTIONS** służy do przechowywania pytań ankiet. Tabela zawiera kolumny:

- **QUESTION_ID** – identyfikator pytania,
- **CONTENT** – tekst pytania.

Tabela **ANSWERS** posiada informacje na temat możliwych odpowiedzi w ankietach oraz komentarze do odpowiedzi. Kolumny:

- **ANSWER_ID** – identyfikator odpowiedzi,
- **QUESTION_ID** – klucz obcy identyfikatora pytania,
- **CONTENT** – tekst odpowiedzi,
- **ANSWER_TYPE** – rodzaj odpowiedzi. Wartości: poprawne, niepoprawne, bardzo niepoprawne,
- **COMMENT** – komentarz do odpowiedzi.

Tabela **NARRATIVES** zawiera linki i opisy do raportów w Power BI. Kolumny:

- **NARRATIVE_ID** – identyfikator opisu (narracji),
- **REPORT_PAGE_ID** – ustandaryzowana nazwa strony raportu,
- **TITLE** – nadany tytuł strony raportu,
- **DESCRIPTION** – opis strony raportu.

Tabela **QUESTION_RESULTS** zawiera zebrane odpowiedzi odwiedzających stronę, do celów statystyk. Kolumny:

- **QUESTION_RESULT_ID** – identyfikator wyniku,
- **QUESTION_ID** – klucz obcy, identyfikator pytania,
- **ANSWER_ID** – klucz obcy, identyfikator odpowiedzi,
- **IP_ADDRESS** – adres IP odwiedzającego stronę,
- **RESULT_TIMESTAMP** – znacznik czasu, domyślnie ustawiany na czas tworzenia rekordu.

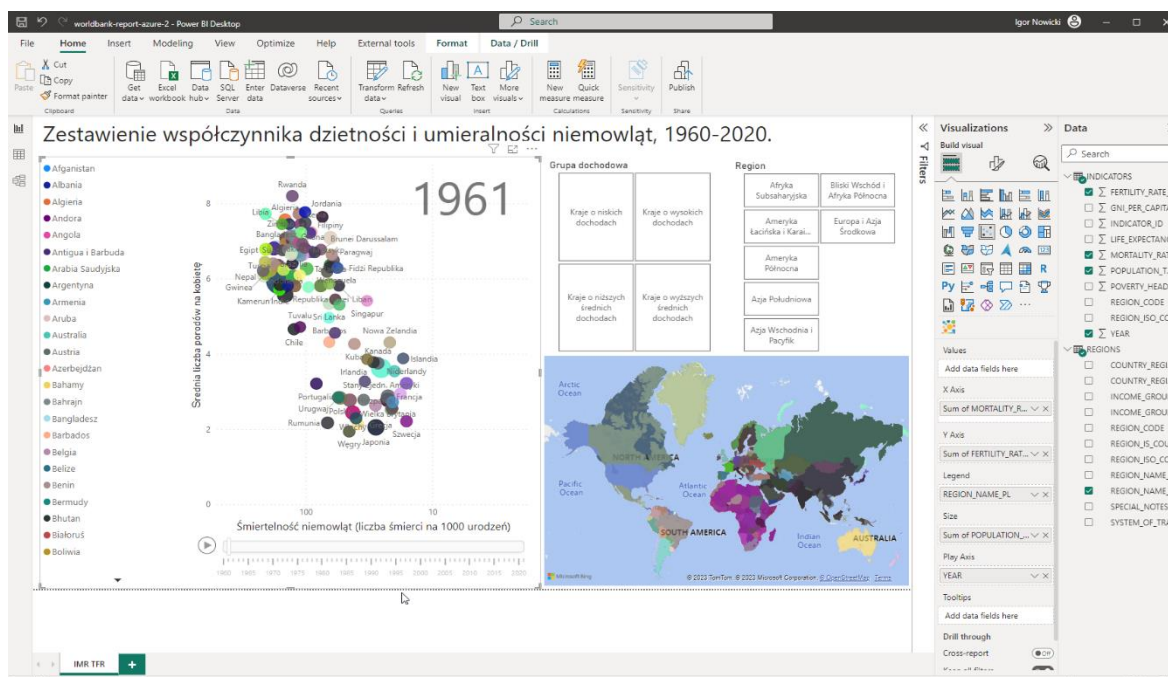
Baza danych SQL Server została wyeksportowana do kopii zapasowej bacpac zgodnej z formatem Azure SQL poprzez operację Tasks → Deploy Database to Azure SQL Database. Kopia zapasowa bazy danych znajduje się w pliku /report/WORLDBANK.bacpac w załączonym archiwum.

¹⁰³ <https://stat.gov.pl/badania-statystyczne/sprawozdawczosc/intrastat/alfabetyczny-wykaz-krajow/>

7. Raport Power BI

7.1. Wstęp

Do utworzenia prezentacji danych został użyty program Power BI Desktop. Dane zostały pobrane z wykorzystaniem lokalnej bazy danych SQL Server, z użyciem nazwy serwera i nazwy bazy danych w postaci parametrów. Takie działanie pozwalało na zmianę adresu i nazwy bazy danych na etapie automatycznego wdrażania. Opcja Data Connectivity została skonfigurowana jako Import, z uwagi na statyczną naturę danych – przy wyborze DirectQuery, dane byłyby pobierane z bazy danych przy każdym wyświetleniu wykresu. Na rysunku 5 zaprezentowano przykładowy zrzut ekranu z aplikacji.



Rysunek 5. Przykładowy zrzut ekranu z tworzonej prezentacji w Power BI Desktop. Opracowanie własne.

7.2. Opis stron

Przy tworzeniu kolejnych stron raportu, wykorzystywane były filtry **Grupy dochodowej** oraz **Regionu**, aby dodatkowo zróżnicować prezentowane kraje. Poniżej jest przedstawiony opis utworzonych wykresów:

- **Fertility 1** – porównanie bieżących (na rok 2020) współczynników dzietności w wybranych krajach oraz wykresy opisujące zmianę współczynnika dzietności na przełomie lat 1960-2020.
- **Fertility 2** – wykresy prezentujące zależność pomiędzy współczynnikiem dzietności a dochodem narodowym per capita (GNI), z możliwością filtrowania po grupie dochodowej i regionie geograficznym.
- **Life Length 1** – wykresy przedstawiające średnią długość życia w wybranych krajach, na przełomie lat 1960-2020.
- **Life Length 2** – wykresy przedstawiające zależność średniej długości życia od dochodu narodowego per capita, z możliwością filtrowania po grupie dochodowej i regionie geograficznym.

- **Life Length 3** – porównanie średniej długości życia w latach 1960 i 2020 we wszystkich krajach świata, z możliwością filtrowania.
- **Extreme Poverty 1** – wykres zmian i porównanie wskaźników skrajnego ubóstwa w latach 1980 – 2020.
- **Infant Mortality Rate 1** – wykres zmian współczynnika umieralności niemowląt w wybranych krajach w latach 1980 – 2020.
- **Infant Mortality Rate 2** – wykres zmian współczynnika umieralności niemowląt we wszystkich krajach świata w latach 1980 – 2020.
- **IMF TFR** – animowany wykres prezentujący zmiany współczynnika dzietności i śmiertelności niemowląt w latach 1960-2020.

7.3. Podsumowanie

Tak utworzony raport został wysłany oraz opublikowany w usłudze Power BI App, skąd będzie dalej używany na stronie internetowej, opisanej w kolejnym podrozdziale. Pliki raportu oraz używanej bazy danych znajdują się w załączonych plikach w katalogu /report.

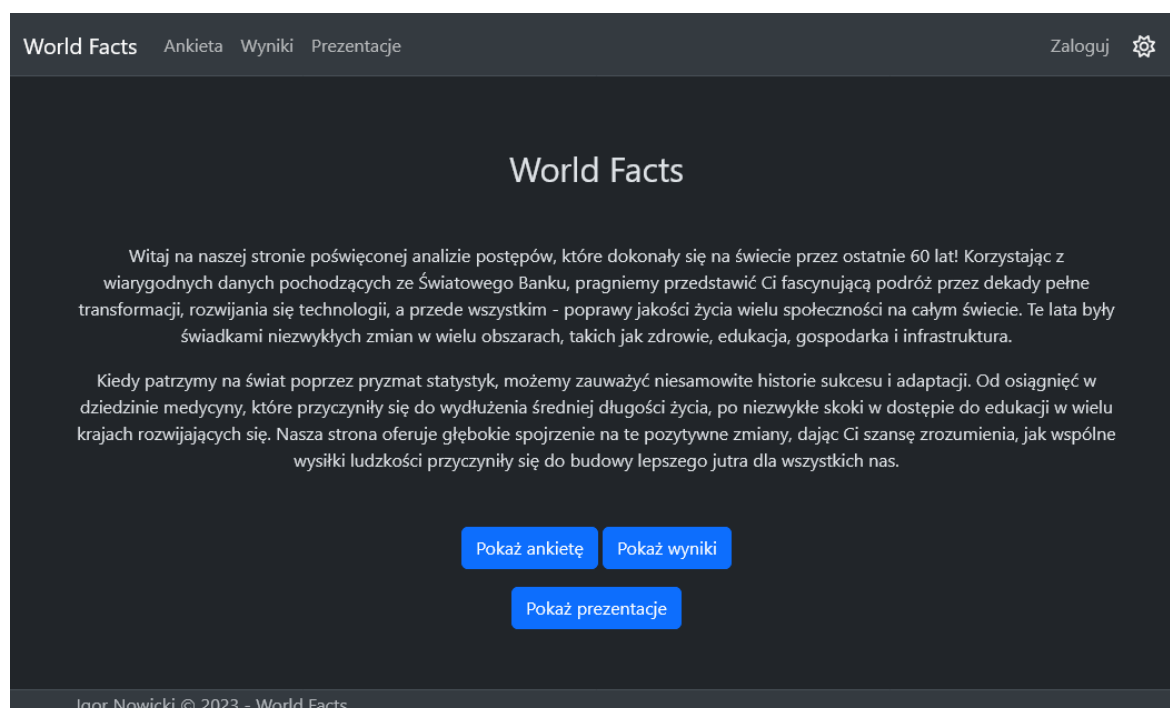
8. Witryna ASP.NET Core MVC

8.1. Wstęp

Witryna World Facts ma na celu przedstawianie ankiet oraz wyświetlanie interaktywnych prezentacji. Przedstawione dane dotyczyły wskaźników światowego rozwoju i zostały bardziej szczegółowo omówione w rozdziale dotyczącym zbierania danych.

Strona internetowa została przygotowana z użyciem środowiska Visual Studio 2022. Rozwiązanie napisano w języku C# z wykorzystaniem frameworka ASP.NET Core MVC 6.0. Witryna jest zaprojektowana w architekturze monolitu. Zostało pominięte rozdzielanie aplikacji na witrynę oraz zaplecze, komunikujące się ze sobą poprzez interfejs WebAPI. Program tworzy kod HTML po stronie serwera, wysyłając następnie użytkownikowi gotową stronę w odpowiedzi na zapytanie. Do renderowania stron został wykorzystany język szablonów Razor.

Strona startowa witryny została zaprezentowana na rysunku 6.



Rysunek 6. Strona startowa witryny World Facts. Opracowanie własne.

Aplikacja wykorzystuje usługę Power BI do wyświetlania prezentacji, natomiast dane są przechowywane w usłudze Azure SQL. Do autoryzacji administratora użyto Azure Active Directory. Sama aplikacja jest hostowana z użyciem usługi Azure App Service.

Rozwiązanie składało się z trzech projektów:

WorldFacts.App – główna aplikacja internetowa, będąca punktem dostępu do systemu. Zostały w niej umieszczone również pliki specyficzne we frameworku ASP.NET Core MVC: zestaw kontrolerów, modeli oraz widoków.

WorldFacts.Library – biblioteka programistyczna przechowująca narzędzia do pobierania i zarządzania danymi z bazy danych i z serwisu Power BI App.

WorldFacts.Tests – projekt przechowujący testy jednostkowe napisane z użyciem bibliotek NUnit oraz Moq. Pozwalały one na automatyczne sprawdzanie poprawności działania narzędzi z biblioteki programistycznej względem przygotowanych atrap (ang. mockups).

8.2. Opis aplikacji

Projekt **WorldFacts.App** korzysta z następujących modułów – kontrolerów odpowiedzialnych za odbieranie, przetwarzanie i odpowiadanie na zapytania:

- **HomeController** – moduł odpowiedzialny za przetwarzanie zapytań o stronę startową, znajdującą się pod główną ścieżką strony. Wyświetla stronę tytułową wraz z listą linków do podstron.
- **AdminController** – moduł używany do odsyłania użytkownika zalogowanego jako administrator do kontrolerów **QuestionsAdminController** oraz **NarrativesAdminController**.
- **QuestionsController** – moduł ankiet. Pytania do ankiet są pobierane z bazy danych z wykorzystaniem **QuestionService**. Pytania są następnie wyświetlane przeglądającemu stronę. Odpowiedzi są zapisywane w bazie danych, w celu przedstawiania statystyk.
- **QuestionsAdminController** – moduł administracyjny ankiet. Administrator strony loguje się do panelu poprzez Azure Active Directory, gdzie może zarządzać pytaniami przedstawianymi w ankiecie. Zarządzanie pytaniami odbywa się z użyciem serwisu **QuestionService**.
- **QuestionsResultsController** – moduł służący do zbierania oraz wyświetlania statystyk odpowiedzi oraz wyników ankiet.
- **ReportController** – moduł odpowiedzialny za wyświetlanie stron raportu wraz z opisem. Same raporty są przechowywane w usłudze Power BI App, natomiast opisy są zebrane w bazie danych. Dostęp tych ostatnich odbywa się poprzez serwis **NarrativeService**. Dostęp do raportu uzyskiwany jest z użyciem **PowerBiService**.
- **NarrativesAdminController** – moduł odpowiedzialny za zarządzanie opisami raportów. Z poziomu panelu administracyjnego można zmieniać treść oraz dodawać i usuwać wyświetlane strony raportu wraz z opisami.

Projekt aplikacji znajduje się w załączonym archiwum, pod ścieżką `/app/WorldFacts.App`.

8.3. Biblioteka programistyczna

Projekt **WorldFacts.Library** przechowuje niezależne od frameworka ASP.NET Core MVC serwisy używane przez kontrolery. Rozwiązanie zostało zaplanowane tak, by biblioteka programistyczna mogła być wykorzystana niezależnie w innych rozwiązaniach, niekoniecznie koncentrujących się na tworzeniu witryny internetowej. Poniżej zaprezentowane są moduły utworzone w ramach biblioteki:

- **QuestionService** – usługa pozwalająca na dostęp i zarządzanie pytaniami do ankiety, zlokalizowanych w tabelach **QUESTIONS** oraz **ANSWERS** w bazie danych.
- **NarrativeService** – serwis odpowiadający za zarządzanie opisami raportów. Opisy są zlokalizowane w tabeli **NARRATIVES** w bazie danych.
- **PowerBiService** – moduł odpowiadający za komunikację z serwisem Power BI App, w celu uzyskania tokena autoryzacyjnego dostępu do raportu. Przy tworzeniu serwisu został wykorzystany kod źródłowy prezentowany na stronie Gunnar Peipman – Programming Blog¹⁰⁴.

¹⁰⁴ <https://gunnarpeipman.com/aspnet-core-power-bi-embedded/>

- **QuestionResultService** – serwis odpowiedzialny za dostęp i aktualizację do danych odpowiedzi użytkowników na pytania, przechowywane w tabeli **QUESTION_RESULTS**. Usługa zwraca podsumowania na temat udzielanych odpowiedzi na pytania w ankiecie.

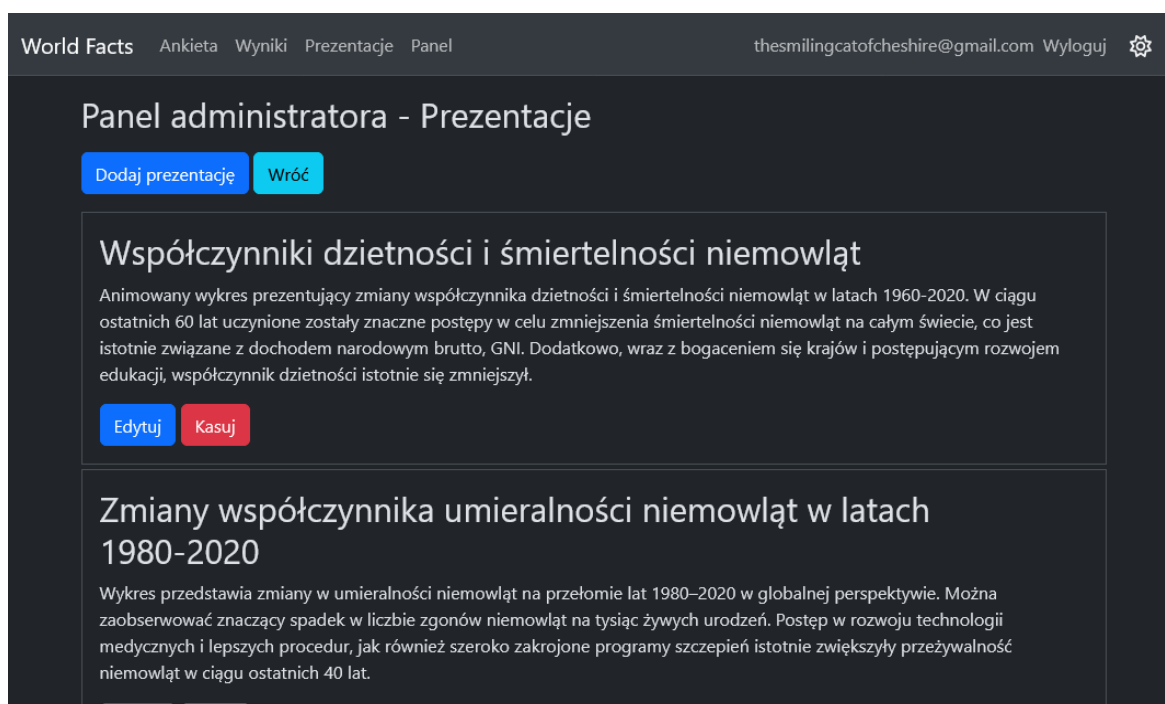
Projekt biblioteki znajduje się w załączonym archiwum, pod ścieżką /app/WorldFacts.Library.

Serwisy były testowane w ramach projektu WorldFacts.Tests z użyciem bibliotek NUnit oraz Moq – poprzez tworzenie odpowiednich atrap komponentów wchodzących w interakcje z testowanymi modułami, możliwe było zadanie określonych warunków brzegowych działania systemu. Projekt testów znajduje się pod ścieżką /app/WorldFacts.Tests.

8.4. Panel administratora

Administrator strony internetowej jest uwierzytelniany z wykorzystaniem okna logowania Azure Active Directory, oraz autoryzowany w trakcie używania kontrolerów QuestionsAdminController oraz NarrativesAdminController. Mechanizmy autentykacji i autoryzacji zostały pobrane z bibliotek Microsoft.Identity, i ich implementacja jest poza zakresem projektu strony internetowej. W ten sposób została zmniejszona możliwość nieautoryzowanego dostępu z winy błędnej implementacji algorytmów uwierzytelniania przez twórcę projektu.

Przykładowy zrzut ekranu z panelu administratora został zaprezentowany na rysunku 7.



Rysunek 7. Panel administracyjny witryny World Facts. Strona implementuje podstawową funkcjonalność systemu zarządzania treścią. Opracowanie własne.

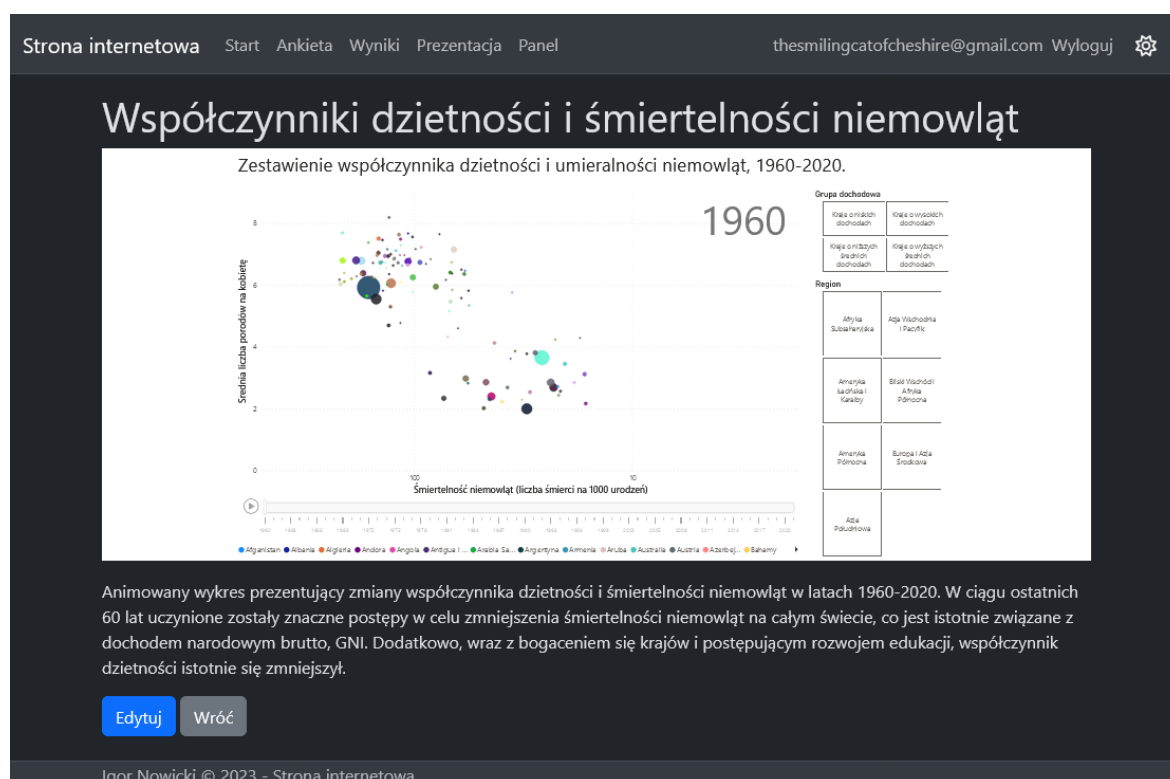
8.5. Wykorzystane narzędzia

Dostęp do bazy danych odbywał się poprzez narzędzia biblioteki Entity Framework Core w wersji 7. Biblioteka dostarcza możliwość tworzenia automatycznych powiązań w projekcie C# ze strukturami bazy danych. Została utworzona klasa `AppDbContext`, będącej mediatorem pomiędzy bazą danych a reprezentacjami danych w projekcie. Implementacja wykorzystuje podejście „Database First” (najpierw baza danych) – w pierwszej kolejności zostały zaprojektowane struktury bazodanowe, do których później został utworzony kontekst i reprezentacje struktur w projekcie C#.

Aplikacja korzysta z mechanizmu wstrzykiwania zależności (dependency injection, DI). Przy rozruchu programu, określane są używane moduły w ramach konkretnych interfejsów oraz zakres ich działania. Przy tworzeniu nowego obiektu danego modułu, usługa DI automatycznie przydziela wymagane serwisy do użycia w ramach serwisu, bez wymogu jawnej deklaracji. W ramach mechanizmu wstrzykiwania zależności można również ustalać zakresy działania, tj. w jakich warunkach są tworzone nowe obiekty danego modułu.

8.6. Integracja z usługą Power BI

Witryna korzysta z raportów publikowanych w usłudze Power BI. Aby móc prezentować raport, wymagany jest mechanizm automatycznej autoryzacji i zapytań w serwisie Power BI App. Raport był prezentowany w postaci elementu `iframe` na stronie internetowej – efektywnie witryna integrowała się z częścią innej strony internetowej. Przykład integracji został zaprezentowany na rysunku 8.



Rysunek 8. Witryna internetowa, pokazująca prezentację udostępnianą przez usługę Power BI App. Opracowanie własne.

9. Skrypty wdrożeniowe PowerShell

9.1. Wstęp

Tworzenie infrastruktury witryny zostało zautomatyzowane poprzez zestaw skryptów PowerShell. Język skryptowy PowerShell umożliwia sprawną automatyzację zadań; dzięki zróżnicowanym bibliotekom możliwa jest kompletna integracja z chmurą Azure.

Samo utworzenie infrastruktury było przeprowadzane z użyciem narzędzia Terraform – popularnego narzędzia umożliwiającego automatyczne tworzenie i modyfikację konfiguracji usług chmurowych. Terraform przyjmuje deklarację oczekiwanego stanu usług, po czym operuje na koncie chmurowym, aby uzyskać stan zgodny z wprowadzoną.

Poniżej przedstawiono kolejne skrypty wdrożeniowe – skrypty zostały podzielone na mniejsze etapy ze względu na wymóg przejrzystości oraz sporadyczną potrzebę ponawiania niektórych kroków.

Skrypty korzystały z konfiguracji z plików:

- `config.json` – ogólnie dostępne dane konfiguracyjne, jak nazwa bazy danych, lokalizacja pliku raportu, kopii zapasowej bazy danych oraz pożądanego lokalizacji zasobów chmury.
- `secrets.json` – informacje poufne, które nie powinny znaleźć się w publicznym repozytorium – sekretne klucze dostępu do chmury, parametry logowania do usług, identyfikatory dzierżaw etc.

9.2. Logowanie do usługi Azure

Skrypt przeprowadza autoryzację w chmurze Azure. Logowanie odbywa się poprzez wyświetlenie okna logowania użytkownika. Panel jest wywoływany poprzez komendę `Connect-AzAccount` z biblioteki `Az.Accounts`. Źródło znajduje się w pliku `00-login-azure.ps1`.

9.3. Przesyłanie raportu do Power BI App

Operacja wysłania raportu do usługi Power BI. Znajdywana jest bądź tworzona przestrzeń robocza o zadanej nazwie, po czym pobierany jest jej identyfikator. Następnie tworzony jest nowy raport na podstawie przedstawionego pliku i identyfikatora przestrzeni roboczej.

Do komunikacji z usługą Power BI App zostały wykorzystane funkcje z biblioteki `MicrosoftPowerBIMgmt`:

- `Get-PowerBIWorkspace` – pobiera identyfikator przestrzeni roboczej,
- `New-PowerBIWorkspace` – tworzy nową przestrzeń roboczą,
- `New-PowerBIReport` – wysyła raport z pliku do przestrzeni roboczej,
- `Get-PowerBIReport` – pobiera dane raportu.

Skrypt zapisuje identyfikatory przestrzeni roboczej oraz utworzonego raportu w pliku `report.json` do użytku przez kolejne skrypty. Źródło znajduje się w pliku `01-report-upload.ps1`.

9.4. Tworzenie infrastruktury z Terraform

W tym skrypcie wywoływany jest program Terraform, wykorzystując konfigurację z zestawu plików konfiguracyjnych z katalogu /deploy/terraform. Tworzone były serwer Azure SQL, konfiguracja Azure Active Directory oraz App Service. W konfiguracji zmiennych środowiskowych App Service umieszczane były dane połączenia z Active Directory, dane połączenia z bazą danych oraz dane raportu Power BI. Zbierane były dane adresu App Service oraz serwera Azure SQL. Do działania programu Terraform używany był utworzony wcześniej klucz App Registration z Azure Active Directory. Źródło znajduje się w pliku 02-create-infrastructure.ps1.

9.5. Ładowanie kopii zapasowej do przestrzeni magazynowej

Aby załadować bazę danych z kopii zapasowej, należy wczytać plik .bacpac do usługi Azure Storage, po czym załadować go na serwer Azure SQL. W tym skrypcie tworzona jest przestrzeń magazynowa oraz wysyłany jest plik kopii zapasowej na serwer. W pliku storage.json zapisywane są informacje na temat nazwy pliku oraz lokalizacji. Plik z kopią zapasową bazy danych ma rozszerzenie .bacpac. Źródło znajduje się w pliku 03-upload-backup.ps1.

9.6. Ładowanie bazy danych z kopii zapasowej

W tym skrypcie używane są funkcje z biblioteki Az.Sql:

- New-AzSqlDatabaseImport – tworzy bazę danych z załadowanej kopii zapasowej,
- Get-AzSqlDatabaseImportExportStatus – pobiera status tworzenia bazy danych.

Proces tworzenia nowej bazy danych jest najbardziej czasochłonny, zajmuje ok. 10 minut. Ponieważ kolejne elementy skryptu wymagają posiadania utworzonej bazy danych, wykorzystywana jest funkcja Get-AzSqlDatabaseImportExportStatus z biblioteki Az.Sql do sprawdzania statusu tworzenia. Skrypt jest kończony po uzyskaniu potwierdzenia od chmury Azure, że baza danych została utworzona. Źródło znajduje się w pliku 04-load-backup-from-storage-to-database.ps1.

9.7. Kasowanie przestrzeni magazynowej

Usuwana jest utworzona wcześniej przestrzeń magazynowa z kopią zapasową bazy danych, z użyciem funkcji Remove-AzStorageAccount z biblioteki Az.Storage.

Źródło znajduje się w pliku 05-remove-storage.ps1.

9.8. Zmiana źródła danych w raporcie Power BI

Wykonywane jest połączenie do aplikacji Power BI App w celu przekierowania połączenia ze źródłem danych z lokalnej bazy danych (nieдоступnej na serwerze) na utworzoną bazę danych Azure SQL. Skrypt łączy się z usługą w celu uzyskania identyfikatorów źródła danych, następnie wysyła żądanie autoryzacji użytkownika do serwisu logowania Microsoft Online. Aby ta operacja się powiodła, musi być utworzony

dodatkowy użytkownik w ramach dzierżawy (niemożliwe jest używanie konta administratora dzierżawy). Po uzyskaniu tokena dostępu, możliwe jest wysłanie żądania do API Power BI App o zmianę źródła danych raportu.

Używane były funkcje:

- `Connect-PowerBIServiceAccount` – logowanie do serwisu Power BI App. Wywołanie metody jest niezbędne do poprawnej autentykacji przy kolejnych metodach. Funkcja pochodzi z biblioteki `MicrosoftPowerBIMgmt.Profile`.
- `Invoke-PowerBIRestMethod` – wywołuje odpowiednią funkcję interfejsu REST API. Zostały użyte metody:
 - `Datasets: Update Parameters` – funkcja nadpisująca sparametryzowane wartości wewnątrz raportu. Narzędzie używane do zmiany parametrów połączenia z bazą danych¹⁰⁵.
 - `Datasets: Get Datasources In Group` – wyświetla listę źródeł danych w ramach przestrzeni roboczej oraz źródła danych. Używane do pobrania identyfikatora źródła danych¹⁰⁶.
 - `Datasource: Update Datasources In Group` – narzędzie aktualizujące źródło danych w ramach przestrzeni roboczej. Wykorzystywane do autoryzacji połączenia z nowym źródłem danych, w tym wypadku – bazą danych Azure SQL¹⁰⁷.

Funkcje `Connect-PowerBIServiceAccount` oraz `Invoke-PowerBIRestMethod` należą do biblioteki `MicrosoftPowerBIMgmt`¹⁰⁸.

Źródło skryptu znajduje się w pliku `06-report-set-database.ps1`.

9.9. Tworzenie paczki binariów aplikacji

W tym skrypcie jest budowana i pakowana aplikacja webowa. Aplikacja jest tworzona z użyciem narzędzia `dotnet`. Program jest kompilowany do środowiska 64-bitowego Linux, ze względu na wybrany serwis aplikacji. Skompilowane binaria są pakowane i przenoszone do folderu skryptów. Źródło znajduje się w pliku `07-build-app.ps1`.

¹⁰⁵ <https://learn.microsoft.com/en-us/rest/api/power-bi/datasets/update-parameters>

¹⁰⁶ <https://learn.microsoft.com/en-us/rest/api/power-bi/datasets/get-datasources-in-group>

¹⁰⁷ <https://learn.microsoft.com/en-us/rest/api/power-bi/datasets/update-datasources-in-group>

¹⁰⁸ <https://learn.microsoft.com/en-us/powershell/module/microsoftpowerbimgmt.profile/invoke-powerbirestmethod>

9.10. Ładowanie paczki do serwisu aplikacji

W tym skrypcie wykorzystywana jest funkcja `Publish-AzWebApp` z biblioteki `Az.Websites`¹⁰⁹, w celu wysłania spakowanych binariów do serwisu aplikacji. Jeśli wdrożenie przebiegło pomyślnie, wyświetlany jest adres utworzonej strony internetowej oraz powiadomienie o poprawnym zakończeniu operacji. Źródło znajduje się w pliku `08-upload-app.ps1`.

9.11. Podsumowanie

Kod źródłowy skryptów wdrożeniowych jest przechowywany w folderze `deploy`. Szczegółowy opis plików znajduje się w pliku `README.md`. Wraz z pozostałymi folderami (`report`, `app`), pliki te są wystarczająco do przeprowadzenia pełnego wdrożenia witryny internetowej i raportu Power BI na chmurze Azure.

¹⁰⁹ <https://learn.microsoft.com/pl-pl/powershell/module/az.websites/publish-azwebapp>

Zakończenie

Dyskusja

Założeniem niniejszej pracy było utworzenie aplikacji internetowej w środowisku chmury Azure, z wykorzystaniem usług Power BI do prezentacji danych oraz przygotowanie skryptów automatycznego wdrażania całego systemu. Do utworzenia aplikacji internetowej został wykorzystany ASP.NET Core MVC w .NET 6.0, do raportów – Power BI Desktop, wraz z Power BI App, natomiast w procesie wdrażania użyte były PowerShell oraz Terraform.

Podczas końcowej fazy pisania pracy, po przygotowaniu skryptów PowerShell oraz Terraform, dostrzeżono możliwość wykorzystania narzędzia Ansible jako praktycznego środka do automatyzacji operacji wdrożeniowych. Przy kolejnych podobnych zadaniach w przyszłości, rozważane będzie skorzystanie z wspomnianego oprogramowania.

Narzędzie Terraform jest niezwykle użyteczne i uniwersalne, jednak być może zbędne przy wdrażaniu tak prostej infrastruktury w środowisku chmury Azure. Rozważano użycie zwyczajnych plików szablonów ARM, jednak zdecydowano na wykorzystanie Terraform w połączeniu z funkcjami PowerShell Az.

Można zadać pytanie, dlaczego nie została użyta usługa Azure DevOps, reklamowana jako pomocne narzędzie usprawniające automatyzację procesów. Zdecydowano, że w ramach pracy należy poznać i wykorzystać uniwersalne narzędzia, odpowiednie do stosowania na różnych platformach chmurowych, takich jak Terraform wraz z dodatkowymi skryptami PowerShell. Z podobnego powodu nie użyto usługi Azure Automation.

Planując wdrożenie w środowisku Linux, można zastosować narzędzia Azure CLI oraz skrypty Bash, uzupełniając je o dodatkowe funkcje języka Python. Ze względu na użycie środowiska Windows 10 do tworzenia projektu zdecydowano, że najbardziej dogodnymi narzędziami będą PowerShell oraz biblioteka Az.

Witryna internetowa była pisana z użyciem pakietu ASP.NET Core MVC w .NET 6.0. Postanowiono nie używać .NET 7.0 ze względu na dłuższe wsparcie wersji 6.0¹¹⁰. Przy tworzeniu projektu w późniejszym terminie zostałby wykorzystany .NET 8.0.

Można poddać dyskusji decyzję wykorzystania szablonów Razor w ASP.NET Core MVC. Alternatywną drogą byłoby przeprowadzenie podziału aplikacji na witrynę i zaplecze. Projekt ASP.NET zostałby utworzony jako WebAPI, natomiast witryna internetowa byłaby napisana we frameworku Angular. Innym wyborem byłby projekt Razor Pages, pozwalający na pominięcie kontrolerów w projekcie.

Technologia Power BI pozwala na tworzenie szybkich wykresów z użyciem metodyki "przeciągnij i upuść" (ang. drag and drop). Wykresy są widowiskowe i przyciągają uwagę możliwością interakcji z danymi. Jednak ze względu na wysoki koszt samego oprogramowania oraz nauki obsługi proponowane jest rozważenie alternatywnych narzędzi.

Można poddać dyskusji wybór danych poddanych analizie w ramach strony internetowej. Wskaźniki są poprawne i wspierane przez poważną instytucję, jednak zostały wybrane selektywnie. Aby zachować równowagę, w pracy powinny również pojawić się wskaźniki pesymistyczne, takie jak zmiany globalnej temperatury czy zanieczyszczenia środowiska.

¹¹⁰ <https://dotnet.microsoft.com/en-us/platform/support/policy/dotnet-core>

Podsumowanie

Niniejsza praca miała na celu zgromadzenie informacji na temat technologii przy tworzeniu witryny internetowej w chmurze, opracowanie przykładowej aplikacji webowej oraz utworzenie skryptów automatyzujących proces wdrożenia. Utworzony projekt spełnia te założenia.

Rozprawa składała się ze wstępu przedstawiającego opis sekcji dokumentów oraz wstępu do chmury obliczeniowej, omawiającego ogólny koncept platformy, proponowanych oczekiwań od portali chmurowych, oraz kategorie usług oferowanych we współczesnych rozwiązaniach. Następnie, przedstawiona została historia chmury Azure wraz z opisem wybranego zestawu usług dostarczanych w ramach platformy. Omówione zostały również narzędzia wykorzystane przy tworzeniu prezentacji, witryny internetowej oraz skryptów wdrożeniowych.

Kolejna część pracy skupiała się na opisie działań podjętych przy tworzeniu pracy – zbieraniu danych, tworzeniu prezentacji, tworzeniu witryny internetowej oraz pisaniu skryptów wdrożeniowych. Został przedstawiony ogólny przegląd, bez szczegółowych instrukcji na temat zastosowanych rozwiązań.

Zakończenie pracy składało się z dyskusji, gdzie omówiono możliwe alternatywne decyzje do podjęcia podczas tworzenia projektu, wraz z powodami, dlaczego zdecydowano się na przyjęcie konkretnych rozwiązań. W podsumowaniu raz jeszcze przeprowadzono opis całej pracy.

Podziękowania

Autor pragnie podziękować firmie FRAKTAL, w osobach Mirosława Lisa i Roberta Koźluka, za wsparcie i finansowanie studiów na uczelni WIT. Kierowane są również podziękowania rodzinie i przyjaciołom.

Bibliografia

- Buyya, B. G. (2011). *Cloud computing, Principles and Paradigms*. John Wiley & Sons, Inc.
- Erl, T. R. (2013). *Cloud computing: concepts, technology & architecture*. Pearson Education.
- Freeman, A. (2013). *Pro ASP.NET MVC 5*. Apress.
- Mell, P. M. (2011). *The NIST Definition of Cloud Computing*. National Institute of Standards and Technology.
- Sommerville, I. (2016). *Software Engineering, Tenth edition*. Pearson Education Limited.
- Toroman, M. (2020). *Chmura Azure, Praktyczne wprowadzenie dla administratora*. Helion SA.

Spis ilustracji

<i>Rysunek 1. Diagram przedstawiający hierarchię rodzajów usług.....</i>	<i>13</i>
<i>Rysunek 2. Przykład zdjęcia z usługi Azure Orbital</i>	<i>16</i>
<i>Rysunek 3. Diagram reklamujący możliwości .NET 5.....</i>	<i>24</i>
<i>Rysunek 4. Diagram zależności usług w infrastrukturze witryny internetowej.....</i>	<i>29</i>
<i>Rysunek 5. Przykładowy zrzut ekranu z tworzonej prezentacji w Power BI Desktop</i>	<i>33</i>
<i>Rysunek 6. Strona startowa witryny World Facts</i>	<i>35</i>
<i>Rysunek 7. Panel administracyjny witryny World Facts.....</i>	<i>37</i>
<i>Rysunek 8. Witryna internetowa, pokazująca prezentację udostępnianą przez usługę Power BI App.....</i>	<i>38</i>

Informacje dodatkowe

Temat

Projekt, implementacja i zautomatyzowane wdrożenie systemu do wizualizacji danych w chmurze Azure

Cel

Celem pracy jest zaprojektowanie i zaimplementowanie platformy do wizualizacji danych społeczno-gospodarczych z publicznych baz. System będzie dostępny w chmurze Azure, a jego wdrożenie zostanie zautomatyzowane z wykorzystaniem narzędzi DevOps.

Zakres

Praca obejmuje projekt, implementację i automatyczne wdrożenie systemu do wizualizacji i analizy danych społeczno-ekonomicznych w chmurze Azure. Analiza danych będzie realizowana przy użyciu narzędzia Power BI, natomiast aplikacja internetowa zostanie przygotowana z użyciem frameworka ASP.NET Core w języku C#. Dane będą przechowywane w bazie danych SQL Server, z użyciem usługi Azure SQL Database. Do automatycznego wdrażania zostaną wykorzystane skrypty PowerShell wraz z biblioteką Azure PowerShell oraz narzędziem Terraform.

Temat w języku angielskim

Design, implementation and automated deployment of a data visualisation system in the Azure cloud

Streszczenie

Praca omawia narzędzia i działania użyte przy tworzeniu aplikacji internetowej, której celem jest prezentacja danych porównawczych. Opisywane są założenia chmur obliczeniowych oraz główne usługi oferowane przez chmurę Azure, narzędzia wykorzystywane przy zbieraniu, przetwarzaniu i prezentowaniu danych oraz programy i usługi używane do przygotowania skryptów automatycznego wdrożenia przykładowego serwisu w chmurze. Autor opisuje działania zastosowane przy obróbce danych, tworzeniu raportu, implementacji aplikacji internetowej oraz tworzeniu skryptów automatycznego wdrażania.

Streszczenie w języku angielskim

Thesis discusses the tools and activities used in the development of a web application aimed at presenting comparative data. Author describes concepts of cloud computing and the main services offered by the Azure cloud, the tools used in collecting, processing and presenting data, as well as the programs and services used to prepare scripts for the automatic deployment of a sample service in the cloud. The author describes the steps used to process data, create a report, implement a web application and create automatic deployment scripts.