

Uniwersytet Mikołaja Kopernika
Wydział Matematyki i Informatyki

Klaudia Augustyńska

nr albumu: 265408

informatyka

Praca magisterska

Wykorzystanie Cloud Computing w aplikacjach mobilnych

Opiekun pracy dyplomowej
dr Błażej Zyglarski

Toruń 2018

Spis treści

Wstęp	5
Problem zastosowania chmury obliczeniowej w aplikacjach mobilnych	6
Cel pracy	7
Opis rozdziałów	7
1. Wprowadzenie	9
1.1. Charakterystyka Cloud Computingu	9
1.2. Ewolucja Cloud Computingu	12
1.2.1. Historia najnowsza. Współczesne wyzwania	15
1.3. Wymagania stawiane współczesnym chmurom	17
1.4. Dokąd zmierza rozwój CC.	17
2. Porównanie różnych podejść	19
2.1. Chmury publiczne	19
2.2. Chmury prywatne	19
3. Analiza wyboru platformy dla aplikacji mobilnej	21
3.1. Wytumaczenie czemu to jest typowy projekt	21
3.2. Wymagania co trzeba wziąć pod uwagę	21
3.3. Werdykt, kandydatka nr 1, 2 ,3	21
4. Opis wdrożenia aplikacji zgodnie z kandydatką nr 1	23
Spis rysunków	25
Spis tabel	27

Wstęp

Cloud Computing to model stanowiący podstawę dla jednych z najprężniej rozwijających się technologii informatycznych obecnych czasów. Jego wykorzystanie odnosi się do praktycznie wszystkich dziedzin informatyki, poczynając od administrowania infrastrukturą komputerową, przez tworzenie systemów informatycznych, po wsparcie badań naukowych oraz pracy przeciętnych użytkowników komputerów.

Technologie oparte o Cloud Computing mają swoje podwaliny w starszych technologiach, sięgających lat 70. ubiegłego wieku. Chodzi zatem o produkt ewolucji technologicznej, nie zaś odrębną nową technologię. Dzisiejszy dynamiczny rozwój Cloud Computingu zawdzięczamy m.in. szybkiemu łączu internetowemu, upowszechnieniu komputerów PC oraz mocy obliczeniowej umożliwiającej wirtualizację na poziomie wykorzystania sztucznej inteligencji do inteligentnego zarządzania infrastrukturą komputerową (ang. *autonomic computing*). Złożenie tych czynników umożliwiło urzeczywistnienie idei po raz pierwszy wymienionej w 1961 r. przez Johna McCarthy’ego, który pisał, że zasoby komputerowe staną się użytecznością publiczną – podobnie jak prąd, który pobieramy z sieci elektrycznej, nie zaś z własnego generatora prądu.[6, 5] Stąd też analizując temat Cloud Computingu niejednokrotnie można spotkać się ze zwrotem „X jako usługa” (ang. *X as a service*), ponieważ praktycznie wszystko, co może być związane z wykorzystaniem komputerów, można dostarczać jako usługę. [6]

Za chmurami obliczeniowymi stoją olbrzymie centra danych, łączące w sieć nawet tysiące komputerów. Wirtualizacja wykorzystywana w Cloud Computingu pozwala na automatyczną konfigurację nowych serwerów dołączanych do sieci oraz odpowiednie zachowanie sieci w przypadku gdy jakaś jej część przestanie działać. Dzięki takim narzędziom można dowolnie rozbudować sieć, a więc uzyskać dowolną przestrzeń dyskową i moc obliczeniową. Dlatego model chmury obliczeniowej jest nieodzownie powiązany z technikami wirtualizacji oraz technikami pracy na systemach rozproszonych.

Duża i zarazem bardzo znacząca część technologii związanych z Cloud

Computingiem dotyczy programowania. Chodzi nie tylko o usługi usprawniające proces wytwórczy oprogramowania, ale również o technologie i wzorce wspierające skalowalność oraz obsługę dużej ilości danych. Chmura pozwala zautomatyzować wiele zadań, z drugiej strony jej ogromne możliwości stanowią duże wyzwanie, gdyż mogą oznaczać całkowitą zmianę podejścia do wytwarzania oprogramowania, wyboru technologii oraz wzorców architektonicznych. Wiele firm powoli podejmuje to wyzwanie przez np. stopniową rezygnację z serwerów firmowych na rzecz serwerów wirtualnych dostępnych w chmurze, wdrażanie strategii CI/CD czy tworzenie nowych projektów w architekturze mikrousług.

Biorąc pod uwagę w jak szybkim czasie pojawiło się wiele istotnych narzędzi szybko znajdujących zastosowanie w biznesie, istnieje duże zapotrzebowanie na opracowanie tego tematu całościowo.

Problem zastosowania chmury obliczeniowej w aplikacjach mobilnych

Chmura obliczeniowa stanowi ważną część tzw. *networked society*, czyli obrazu do którego współcześnie dąży technologia oraz sposób korzystania z niej przez społeczeństwo. Jest to paradygmat w którym korzystanie z elektronicznych asystentów czy IoT (ang. *Internet of Things*) stanowi niezbędny element rzeczywistości.[5] W tym nowym obrazie świata chmura stanowi spoiwo, natomiast komunikacja z człowiekiem w dużej mierze odbywa się przy pomocy aplikacji mobilnych. Można spodziewać się rosnącego zapotrzebowania na aplikacje mobilne, których głównym zadaniem jest skuteczna komunikacja z chmurą.

Podczas wyboru technologii, na etapie analizy dostępnych rozwiązań okazuje się, że istnieje bardzo wiele podejść do tematu, z których duża część powstała w ciągu kilku ostatnich lat i od momentu publikacji zdobyła natychmiastową popularność. Dodatkowo wszystkie materiały, które pozwoliłyby to wszystko zrozumieć, występują w języku angielskim, najczęściej w kontekście wybranego specjalistycznego zastosowania.

Powyższy problem przyczynia się do braku zrozumienia technologii chmurowych jako całości. Powoduje to niską świadomość jak w pełni wykorzystać możliwości chmury w kontekście tworzenia typowego projektu informatycznego, jakim jest aplikacja mobilna.

Cel pracy

Celem pracy było rozstrzygnięcie jak na obecny stan wiedzy podejść do problemu realizacji typowego projektu aplikacji mobilnej w oparciu o Cloud Computing. W rozumieniu niniejszej pracy, typowa aplikacja pozwala na wprowadzanie i otrzymywanie danych od serwera za pomocą udostępnionego API. W rezultacie miała zostać wybrana konkretna technologia, która pod względem teoretycznym najlepiej odpowiadałaby stawianym wymaganiom.

Wybrana technologia miała być zastosowana w praktyce do stworzenia aplikacji na system Android, komunikującej się z API udostępnianym przez chmurę. Aplikacja miała umożliwiać zapisywanie wydatków w sposób uwzględniający fakt, że wiele wydatków ludzie dokonują nie tylko z myślą o sobie samych. System miał pozwolić założyć konto w serwisie, a następnie połączyć konta w grupy odpowiadające gospodarstwu domowemu. Kategorie wydatków miały pozwolić na uzgodnienie, kto ile płaci w danej kategorii (np. wydatki na higienę po 50%). Serwer miał być potrzebny do wspomaganie rozliczeń pomiędzy poszczególnymi osobami, dynamicznie naliczając kto ma ile do oddania. Ponadto miał wspomagać kontrolę własnych wydatków przez uwzględnienie, że koszty życia to nie tylko własne wydatki, ale również wydatki poniesione przez inne osoby prowadzące to samo gospodarstwo domowe. Ponieważ tradycyjne programy do zarządzania wydatkami nie posiadają wyżej wymienionych funkcjonalności, jak również zazwyczaj mogą działać bez serwera, to jest przykład aplikacji, która może nagle stać się popularna i powinna być w stanie wówczas obsłużyć większe obciążenie serwera.

Opis rozdziałów

Rozdział 1. – Wprowadzenie TODO. NAPISZĘ O TYCH ROZDZIAŁACH W CZASIE PRZESZŁYM, JAK RZECZYWIŚCIE BĘDĄ W CZASIE PRZESZŁYM

Rozdział 1.

Wprowadzenie

Niniejszy rozdział wyjaśnia w jaki sposób ewoluowały technologie, by dać się później poznać jako Cloud Computing, a także czego należy oczekiwać od współczesnej chmury. Na końcu rozdziału opisano dokąd zmierza rozwój przedstawionych technologii.

1.1. Charakterystyka Cloud Computingu

Cloud Computing to model zgodnie z którym wszelkie zasoby informatyczne (oprogramowanie, przestrzeń dyskowa, dostęp do bazy danych itp.) dostarczane są w formie usługi. Istotną cechą jest wysoka skalowalność udostępnianych rozwiązań. Po stronie klienta ma to wyglądać tak, jak gdyby posiadał dostęp do nieskończonej mocy obliczeniowej i niekończącej się przestrzeni dyskowej, natomiast po stronie usługodawcy podłączenie nowych serwerów w celu podtrzymania tej iluzji nie powinno stanowić problemu. [16]

Aby była możliwa tak wysoka elastyczność, fizyczne serwery są oddzielone warstwą abstrakcji, na której są widoczne jako pula zasobów takich jak przestrzeń dyskowa czy moc procesora. Każdy program czy serwer wirtualny działający w chmurze osadzany jest na wirtualnych zasobach; nie ma możliwości zdecydowania z którego konkretnego fizycznego zasobu chce się korzystać. Moc chmury obliczeniowej buduje się przez łączenie tanich, łatwo wymienialnych komponentów sprzętowych w potężne zasoby wirtualne. [6]

Poza rozwiązaniem problemu zarządzania ogromną ilością fizycznych serwerów, wirtualizacja zasobów pozwala na lepsze wykorzystanie sprzętu. W tradycyjnym modelu komputery muszą być przygotowane na wypadek gdyby zainstalowane na nich programy powodowały większe zużycie zasobów. Dzieje się tak nawet w przypadku komputerów PC – kupuje się specjalnie większe dyski i lepszy procesor, aby przydały się w przyszłości. W

ten sposób wykorzystuje się niewielką część możliwości pojedynczego komputera, ponieważ przez większość czasu potrzeba mu znacznie mniej zasobów niż fizycznie posiada. W przypadku wirtualnych zasobów, do fizycznej jednostki można dynamicznie przypisać zużycie powodowane przez wielu użytkowników, wiele wirtualnych systemów operacyjnych, w zgodzie z ustalonym algorytmem. Algorytm może definiować, że np. wszystkie komputery mają zostać obciążone po równo (round robin), czy że pojedynczy węzeł ma być wykorzystany w 100% [8]. Technikę tę nazywa się równoważeniem obciążenia (ang. *load balancing*).

Technika zrównoważonego obciążenia przynosi kilka ważnych korzyści będących istotnymi cechami chmur obliczeniowych. Dzięki niej usługi działające w chmurze mogą być uruchomione na różnych węzłach, w tylu instancjach, ile wymagane jest do prawidłowego obsłużenia ruchu. W przypadku awarii którejś z instancji użytkownicy usługi niczego nie odczuwają, gdyż zostają przekierowani na instancję co do działania której nie wykazano błędów. Wszystko to ułatwia tworzenie wysoce skalowalnego oprogramowania. Nie bez znaczenia jest także pozytywny wpływ na ekosystem, ponieważ chmury obliczeniowe oznaczają optymalne zużycie istniejącego sprzętu komputerowego, a więc nie trzeba produkować go więcej niż potrzeba ani niepotrzebnie zużywać energii elektrycznej.

Najbardziej spektakularne efekty wykorzystywania modelu chmury widać w przypadku dostawców posiadających największe centra danych na świecie, takich jak Amazon, Microsoft, Google czy IBM. Na kilkudziesięciu m^2 gromadzą zasoby komputerowe, których równocześnie mogą używać miliony osób na całym świecie. Na jednym fizycznym komputerze zasoby mogą być wykorzystywane przez wiele osób niewiedzących o sobie nawzajem (po angielsku tę właściwość określa się jako *multi-tenancy*). Chmury o takiej architekturze mają potencjał ucieleśnić ideę zgodnie z którą zasoby komputerowe mogą być dostarczane jako usługa użyteczności publicznej.

Wyżej wymieniony sposób myślenia o chmurze jest tym co wyraźnie odróżnia chmury publiczne od chmur prywatnych. Nic nie stoi na przeszkodzie, aby samemu stworzyć prywatną sieć opartą o model chmury. Wówczas formalnie jest to chmura, lecz nie zapewnia niektórych ważnych korzyści, głównie przez konieczność samodzielnego administrowania serwerami, mniejszą ilość potencjalnych użytkowników oraz mniejszy potencjał puli zasobów. Przykładowo, jeśli firma nadal musi martwić się o zarządzanie fizycznymi jednostkami komputerowymi, to nie będzie miała możliwości wychwalać modelu chmury za brak konieczności zatrudniania specjalistów zajmujących się

infrastrukturą komputerową.

Korzyści i wady

Do głównych korzyści wynikających ze stosowania modelu chmury należą:

- **rozwiązanie problemu skalowania aplikacji**

Wraz z rozwojem Web 2.0, dostępnością szybkiego łącza internetowego, zwiększaniem ilości urządzeń podłączonych do Internetu oraz ilości transmitowanych danych, istnieje rosnące zapotrzebowanie na aplikacje będące w stanie obsłużyć duży ruch. Dzięki chmurom uruchomienie wielu instancji aplikacji na różnych węzłach, dodatkowo w odmiennych wersjach, sprowadza się do opisu - ile, w jakiej wersji, w jakich proporcjach.[12] Podobnie ułatwione jest korzystanie z baz NoSQL.

- **rozwój Big Data**

Chmury są w stanie zapewnić odpowiednie środowisko do przechowywania oraz przetwarzania dużych zbiorów danych. Udostępnienie tego środowiska jako usługi znacząco redukuje koszt przechowywania i analizy tych zbiorów, co za tym idzie próg wejścia w ten temat staje się dużo niższy. Za tym idzie wydajniejsze odkrywanie wiedzy z danych, co może mieć wpływ m.in. na rozwój medycyny.

- **ochrona środowiska**

W modelu chmury sprzęt komputerowy zostaje wydajniej wykorzystany, w związku z tym jest mniejsze zapotrzebowanie na nowy sprzęt. Również dzięki pozostawieniu kosztownych operacji po stronie chmury, urządzenia klienckie łączące się z chmurą nie muszą być regularnie wymieniane na silniejsze.

- **lepszą koncentracją na wybranym zadaniu**

Model chmury zwalnia użytkowników z dodatkowych czynności przy realizacji danego zadania. Przykładowo, jeśli komuś jest potrzebny serwer, to z chmurą publiczną nie musi martwić się zakupem sprzętu, zapewnieniem odpowiedniego pomieszczenia czy dostępem do Internetu. Może się zamiast tego skupić na właściwym skonfigurowaniu systemu na serwerze wirtualnym.

- **płatność tylko za rzeczywiste zużycie**

Chmura umożliwia rozliczanie na podstawie czasu używania proce-

sora, ilości przesłanych megabajtów czy ilości danych przechowywanych w chmurze. W szczególności zmniejsza to początkowy koszt wdrażania czegokolwiek w chmurze.

- **gwarancja jakości usługi**

Każdy dostawca usług chmurowych udostępnia klientom SLA (ang. *Service-level agreements*), w którym zobowiązuje się do utrzymania określonego poziomu niezawodności usług (np. dostępność w 99,9% przypadków) oraz przewidywanego zachowania w przypadku niedotrzymania zapewnień (np. obniżki cen).

Ostatnią korzyścią, a jednocześnie kontrowersją, jest bezpieczeństwo danych w chmurach publicznych. Wysyłanie niejednokrotnie wrażliwych danych w nie do końca określone miejsce budzi obawy. Według autorów specjalistycznych wydawnictw[6, 16] są niesłuszne – pozostają zgodni co do opinii, że główni dostawcy usług chmurowych są w stanie zapewnić najwyższy poziom bezpieczeństwa, a fizyczne przechowywanie danych na terenie firmy daje tylko ułudę bezpieczeństwa.

Do potencjalnych wad rozwiązań bazujących na chmurach obliczeniowych należą:

- **problem z przenoszeniem aplikacji z jednej chmury na drugą**

Znaczna część usług oferowanych przez największych dostawców usług chmurowych jest dedykowana dla tworzonej przez nich chmury. Oznacza to, że jeśli użytkownik chce mieć możliwość zmiany dostawcy, powinien uwzględnić to przy wyborze narzędzi pracy. Więcej na ten temat można przeczytać w rozdziale 3.

- **problemy prawne**

Istnieje ryzyko rozbieżności pomiędzy prawem do którego dostosowywał się dostawca usług chmurowych, a prawem chroniącym dane w kraju, na terenie którego chce się korzystać z tych usług.

1.2. Ewolucja Cloud Computingu

Cloud Computing nie stanowi odrębnej, nowej technologii – jest to produkt ewolucji technologii rozwijanych od blisko 50 lat. Prześledzenie historii pozwala na zrozumienie, które koncepcje rzeczywiście wprowadzają nową jakość, a nie są jedynie od dawna istniejącą technologią, tyle że ubraną w ładnie brzmiące słowo.

Historia technologii chmurowych

Historia ma początek około lat 70., gdy używano wielkich superkomputerów zwanych **mainframe**. Można było z nich korzystać za pomocą terminali, określanych mianem „głupich” (ang. *dummy terminal*), ponieważ nie posiadały procesora i mogły być używane jedynie do operacji I/O. Jako że z serwera mógł korzystać jeden terminal naraz, serwer ustawiał je w kolejkę i musiały długo czekać na obsłużenie.

W latach 70. terminale zaczęły być wyposażone w mikroprocesory i być określane mianem „**inteligentnych terminali**” (ang. *intelligent terminal*). Mogły już partycypować przy uruchamianiu programów, co skróciło czas obsługi i dało początek modelowi klient-serwer.

Dalszy rozwój umożliwił rozwój mikroprocesorów, co dało początek **komputerom PC**. Komputery te mogły działać samodzielnie i były znacznie tańsze od mainframe-a.

Wynaleziono również LAN (ang. *local area network*) i WAN (ang. *wide area network*), co umożliwiło łączenie komputerów PC w sieci bez konieczności łączenia z mainframe-m. Powstały sieci **P2P** (ang. *Peer-To-Peer*).

We wczesnych latach 80. wynaleziono **systemy rozproszone**. Ponieważ były w stanie przetwarzać dane równolegle, zachwiało to poglądem iż większą moc obliczeniową należy uzyskiwać przez wynajdowanie silniejszych procesorów. Systemy rozproszone wymagały wzmożonej ilości operacji komunikacji, lecz występowało to w parze z rozwojem LAN osiągającym przepustowość 100 Mbps oraz WAN osiągającym 64 kbps.

Następnie powstała koncepcja **klastrów komputerowych**. Klaster polegał na połączeniu komputerów tego samego typu (homogenicznych) w sieć LAN i wyłonieniu wśród nich zarządcy, który będzie zajmował się przydzielaniem zadań pozostałym węzłom. Gdyby któryś z węzłów miał awarię, to inne mogły przejąć jego zadanie. Dało to początek idei **puli zasobów**. W klastrze niezawodność osiągało się przez nadmiarowość zasobów.

Główną wadą klastrów była konieczność powierzenia zarządzania klastrzem jednemu komputerowi. Stanowił on miejsce od niezawodności którego zależała niezawodność całej sieci (ang. *single point of failure*). Rozwiązaniem okazały się **gridy**, w których każdy węzeł posiadał równy priorytet. Klient mógł podłączyć się do dowolnego komputera w gridzie, a ponadto komputery te mogły być różnego typu (heterogeniczne). Wkrótce gridy przeniknęły ze świata naukowego do świata biznesu i zaczynały być łączone przez WAN.

Dotychczas gdy uruchamiano się program na wybranym węźle gridu, to znajdował się na tym węźle dopóki proces nie został zakończony. Stanowiło

to problem na drodze do skalowania w czasie rzeczywistym, gdzie na skutek zmienionych potrzeb byłoby dobrze mieć możliwość od nowa zaalokować zasoby bez zakłócania działającej usługi. Problem ten rozwiązała **wirtualizacja sprzętowa**, za pomocą której zadania były alokowane do maszyn wirtualnych. Technika ta umożliwiła **utility computing**, czyli model w którym zasoby komputerowe mogą być dostarczane jako usługa, co stanowi serce chmur obliczeniowych. Prekursorem została firma Salesforce.com, która udostępnia oprogramowanie w formie usługi internetowej od 1999 r.

W 2001 r. firma IBM zbudowała pierwszy **autonomiczny system** (ang. *autonomic computing*) – system, który potrafi sobą zarządzać bez interwencji człowieka. Dzięki wykorzystaniu sztucznej inteligencji do działania, eliminuje się ryzyko związane z błędem ludzkim oraz złożoność związaną z koniecznością doglądania przez człowieka złożonego systemu. IBM wytyczył następujące cechy modelu:

- automatyczna konfiguracja – konfiguracja tworzy się automatycznie na podstawie zapotrzebowania,
- automatyczna naprawa błędów – system sam wykrywa błędy i reaguje na nie,
- automatyczna optymalizacja – system sam dba o optymalne użycie zasobów,
- automatyczna ochrona – system wykrywa próby ataków i zapobiega im.

Wirtualizacje w tym systemie działają zgodnie z „pętlą adaptacyjną”: obserwuj, decyduj, zareaguj.[5]

Duże znaczenie miało także wynalezienie metody **SOA** (ang. service oriented architecture), czyli techniki wytwarzania oprogramowania przez wydzielanie niezależnie działających komponentów, składających się razem na większy system informatyczny. Ważna była również koncepcja **Web 2.0**, nazwana tak po raz pierwszy w 2002 r. roku. Model Web 2.0 spowodował zmiany w sposobie myślenia o Internecie, zgodnie z którymi nie powinien jedynie służyć do dostarczania statycznej treści, lecz może być użyteczny również do udostępniania treści tworzonej przez użytkowników Internetu.

Wymienione wyżej technologie zestawione razem dały podwaliny chmurom obliczeniowym.

Przyjmuje się, że termin „*cloud computing*” po raz pierwszy został użyty przez CEO firmy Google, Erica Schmidta, w trakcie konferencji w 2006 r.

Tego samego roku Amazon użył tej nazwy publikując pionierską w swoim gatunku usługę Elastic Compute Cloud (EC2), czyli możliwość wynajmu serwera wirtualnego w ich chmurze.

1.2.1. Historia najnowsza. Współczesne wyzwania

W 2003 roku firma Citrix stworzyła Xen - system operacyjny przeznaczony tylko do wirtualizacji innych systemów operacyjnych. Niedługo potem Xen został udostępniony jako oprogramowanie otwartoźródłowe. W marcu 2006 r. roku Amazon na podstawie Xena opracował swój produkt EC2. W ciągu 18 miesięcy zaczęło z niego korzystać ponad pół miliona osób.[16]

W kolejnych latach powstały kolejne chmury publiczne oraz nastąpił rozwój rozwiązań typu PaaS (ang. *platform as a service*). W czerwcu 2007 r. powstał Heroku, będący pionierem w tej kategorii. Niecały rok później Google udostępnił usługę Google App Engine. W lutym 2010 r. Microsoft udostępnił Windows Azure (w kwietniu 2014 przemianowaną na Microsoft Azure [15]).

Do tego momentu powstały już 3 największe chmury publiczne: Amazon AWS, Google Cloud Platform, Microsoft Azure. Początkowo miały zróżnicowaną ofertę. W literaturze[16] z 2011 r. porównywanie tych platform sprowadzało się do zakresu oferowanych przez nie usług. Z biegiem czasu różnice zatarły się – obecnie usługa publikowana przez jednego dostawcę po kilku miesiącach jest dostępna także u innych. Tabela 1.1 prezentuje porównanie tempa rozbudowy zakresu usług.

W tym samym czasie rozwijały się technologie przeznaczone do wystawiania własnych usług chmurowych. W 2010 r. powstała otwartoźródłowa platforma **OpenStack** służąca do świadczenia własnych usług typu IaaS. Rok później powstał **Cloud Foundry** – również otwartoźródłowy projekt, pozwalający na prowadzenie własnej platformy typu PaaS do tworzenia aplikacji w chmurze.

Cloud Foundry stanowił odpowiedź na główny problem chmur publicznych – tworzenie aplikacji na wybraną chmurę wiązało ją z mechanizmami specyficznymi dla tej chmury. Brakowało pomostu, który oddzieliłby warstwę tworzenia aplikacji w chmurze od wybranego usługodawcy, bez konieczności tworzenia serwerów wirtualnych na których trzeba instalować określone środowisko. [8]

Cloud Foundry jest niezależny od infrastruktury, może działać zarówno na OpenStacku jak i na serwerze wirtualnym w chmurze publicznej. Zapew-

Typ usługi	Dostawca		
	AWS	GCP	Azure
serwery wirtualne	2006 – Amazon EC2 [6]	2013 – Google Compute Engine [9]	2010 – Azure Virtual Machines [14]
platforma aplikacji	2011 – AWS Elastic Beanstalk [1]	2008 – Google App Engine [9]	2010 – Azure Cloud Services [14]
bazy SQL	2009 – Amazon Relational Database Service [3]	2011 – Google Cloud SQL [10]	2010 – Azure SQL Database [14]
bazy klucz-wartość	2007 – Amazon SimpleDb (w 2012 Amazon DynamoDB) [4]	2008 – Google Bigtable [7] (jako część App Engine)	2010 – Azure Table Storage [14] (2017 – Azure Cosmos DB)
serverless	2014 – AWS Lambda [17]	2016 – Google Cloud Functions [17]	2016 – Azure Functions [17]
Kubernetes	2018 – Amazon Elastic Container Service for Kubernetes [2]	2018 – Google Kubernetes Engine [11]	2017 – Azure Container Service [13]

Tablica 1.1: Porównanie historii udostępniania usług chmurowych wśród głównych dostawców

nia możliwość wyboru dowolnych technologii do wytwarzania oprogramowania. Do działania używa kontenerów zgodnych ze standardem OCI (Open Container Initiative). [8]

Konteneryzacja to metoda wirtualizacji na poziomie systemu operacyjnego, gdzie na działającym systemie operacyjnym można uruchomić wiele odizolowanych od siebie instancji przestrzeni użytkownika. Każda taka instancja to kontener. Kontener zapewnia odpowiednie środowisko do działania dla umieszczonej w nim aplikacji, np. środowisko uruchomieniowe .NET Core w odpowiedniej wersji. [8, 5]

Cloud Foundry zapewnia orkiestrację kontenerów oraz obsługę procesu CI/CD (ciągłej integracji i ciągłego dostarczania, ang. *continuous integra-*

tion / continuous delivery).[8] Stanowi narzędzie pozwalające z powodzeniem wdrażać strategię DevOps.

1.3. Wymagania stawiane współczesnym chmurom

1.4. Dokąd zmierza rozwój CC.

Rozdział 2.

Porównanie różnych podejść

2.1. Chmury publiczne

2.2. Chmury prywatne

Rozdział 3.

Analiza wyboru platformy dla aplikacji mobilnej

- 3.1. Wytumaczenie czemu to jest typowy projekt
- 3.2. Wymagania co trzeba wziąć pod uwagę
- 3.3. Werdykt, kandydatka nr 1, 2 ,3

Rozdział 4.

Opis wdrożenia aplikacji
zgodnie z kandydatką nr 1

Podsumowanie

Spis rysunków

Spis tablic

1.1. Porównanie historii udostępniania usług chmurowych wśród głównych dostawców	16
---	----

Bibliografia

- [1] *AWS Elastic Beanstalk – Wikipedia*. URL: https://en.wikipedia.org/wiki/AWS_Elastic_Beanstalk (term. wiz. 09.09.2018).
- [2] *Amazon EKS – Now Generally Available*. URL: <https://aws.amazon.com/blogs/aws/amazon-eks-now-generally-available/> (term. wiz. 09.09.2018).
- [3] *Amazon Relational Database Service*. URL: https://en.wikipedia.org/wiki/Amazon_Relational_Database_Service (term. wiz. 09.09.2018).
- [4] *Amazon SimpleDB – Wikipedia*. URL: https://en.wikipedia.org/wiki/Amazon_SimpleDB (term. wiz. 09.09.2018).
- [5] Nick Antonopoulos i Lee Gillam. *Cloud Computing. Principles, Systems and Applications*. 2nd. Cham, Switzerland: Springer International Publishing AG, 2017. ISBN: 978-3-319-54645-2.
- [6] Sandeep Bhowmik. *Cloud Computing*. 1st. New York, NY, USA: Cambridge University Press, 2017. ISBN: 1316638103, 9781316638101.
- [7] *BigTable – Wikipedia*. URL: <https://pl.wikipedia.org/wiki/BigTable> (term. wiz. 09.09.2018).
- [8] Rick Farmer, Rahul Jain i David Wu. *Cloud Foundry for Developers*. 1st. Birmingham, UK: Packt Publishing Ltd., November 2017. ISBN: 978-1-78839-144-3.
- [9] *Google Cloud Platform – Wikipedia*. URL: https://en.wikipedia.org/wiki/Google_Cloud_Platform (term. wiz. 09.09.2018).
- [10] *Google Cloud SQL: your database in the cloud*. URL: <http://googlecode.blogspot.com/2011/10/google-cloud-sql-your-database-in-cloud.html> (term. wiz. 09.09.2018).

- [11] *Google Kubernetes Engine 1.10 is generally available and ready for the enterprise.* URL: <https://cloudplatform.googleblog.com/2018/05/Google-Kubernetes-Engine-1-10-is-generally-available-and-ready-for-the-enterprise.html> (term. wiz. 09.09.2018).
- [12] Kelsey Hightower, Brendan Burns i Joe Beda. *Kubernetes: Up and Running*. 1st. O'Reilly Media, Inc., 2017. ISBN: 978-1-491-93567-5. URL: <https://books.google.pl/books?id=8QY0tAEACAAJ>.
- [13] *Introducing AKS.* URL: <https://azure.microsoft.com/en-us/blog/introducing-azure-container-service-aks-managed-kubernetes-and-azure-container-registry-geo-replication/> (term. wiz. 09.09.2018).
- [14] *Introducing Windows Azure.* URL: <https://azure.microsoft.com/en-us/blog/introducing-windows-azure/> (term. wiz. 09.09.2018).
- [15] Florian Klaffenbach, Jan-Henrik Damaschke i Oliver Michalski. *Implementing Azure Solutions*. 1st. Birmingham, UK: Packt Publishing Ltd., May 2017. ISBN: 978-1-78646-785-0.
- [16] Jothy Rosenberg i Arthur Mateos. *Chmura obliczeniowa. Rozwiązania dla biznesu*. Gliwice: Helion, 2011. ISBN: 978-83-246-4167-3.
- [17] Maddie Stigler. *Beginning Serverless Computing. Developing with Amazon Web Services, Microsoft Azure, and Google Cloud*. 1st. Apress, 2018. ISBN: 978-1-4842-3084-8.