

POLITECHNIKA WROCŁAWSKA

Wydział Informatyki i Telekomunikacji

Cyberbezpieczeństwo



PROJEKT

Aplikacja do integracji uwierzytelniania SSO (Smartbear)

Przemysław Jamonnt

Dawid Seweryn

Anton Urda

Cyryl Sznajder

Paweł Warcholiński

Projekt Zespołowy

Dr inż. Robert Czechowski

Wrocław, 2023

Spis treści

Wprowadzenie.....	3
Rozdział 1. Wstęp teoretyczny	3
Rozdział 2. Założenia projektowe	4
Planowanie projektu	4
Etap 1: Analiza wymagań projektu	4
Etap 2: Planowanie projektu.....	4
Etap 3: Implementacja projektu	5
Etap 4: Testowanie i wdrożenie projektu	5
Etap 5: Prezentacja projektu	5
Rozdział 3. Projekt „Amplify Application”	5
Rozpoczęcie prac technicznych	5
Tworzenia okna logowania w Amplify	6
Tworzenie pierwszych funkcji Lambda	7
Tworzenie pierwszych IAM	8
Tworzenie wspólnego konta AWS	8
Podłączenie Lambda do projektu Amplify	8
Rozdział 4. Badania i wnioski	9
Podsumowanie	9
Bibliografia	10

Wprowadzenie

Wspólnie z firmą Smartbear realizujemy projekt, którego celem jest stworzenie aplikacji do integracji uwierzytelniania SSO. Pierwszym krokiem jest podział obowiązków oraz ustalenie harmonogramu prac. Następnie decydujemy o środowiskach i technologiach, które wykorzystamy w projekcie, a członkowie zespołu przygotowują się do jego realizacji, zapoznając się ze środowiskami i zdobywając niezbędną wiedzę. W kolejnym etapie tworzymy odpowiednie środowiska, umożliwiające opracowanie projektu oraz śledzenie postępów prac, w tym zgodność z harmonogramem. Realizacja projektu obejmuje opracowanie Frontendu oraz Backendu, przy użyciu odpowiednich podzespołów, które pozwalają na osiągnięcie założonych celów. Na cotygodniowych spotkaniach omawiamy postępy prac oraz konsultujemy się z firmą w kwestii rozwiązań technicznych. Kolejnym etapem jest testowanie pierwszej wersji aplikacji, lokalizowanie błędów oraz wprowadzenie koniecznych poprawek. Finalnym krokiem jest złożenie gotowego projektu, w tym finalnej wersji aplikacji oraz dokumentacji projektowej.

Rozdział 1. Wstęp teoretyczny

Nasz zespół studentów pracuje nad stworzeniem aplikacji dla SSO (Single Sign-On) przy użyciu platformy AWS Amplify.

SSO to mechanizm uwierzytelniania, który umożliwia użytkownikom korzystanie z różnych usług i aplikacji za pomocą jednego zestawu poświadczeń. Oznacza to, że użytkownik może zalogować się raz, a następnie uzyskać dostęp do wielu usług bez potrzeby ponownego logowania.

AWS (Amazon Web Services) to platforma chmurowa, która umożliwia dostęp do różnych zasobów, takich jak serwery, bazy danych i narzędzia deweloperskie. Amplify to usługa AWS, która ułatwia tworzenie aplikacji mobilnych i webowych.

W ramach naszego projektu chcemy wykorzystać różne zasoby AWS, które będą przydatne dla naszej aplikacji SSO. Jednym z takich zasobów jest **AWS Cognito**, który pozwala na łatwe zarządzanie tożsamościami użytkowników, uwierzytelnianie i autoryzację.

Innym zasobem, który będzie przydatny dla naszej aplikacji, jest **AWS Lambda**, która umożliwia uruchamianie kodu bez konieczności zarządzania serwerami. Dzięki temu możemy skoncentrować się na tworzeniu funkcjonalności aplikacji, a nie na infrastrukturze.

Oprócz tego, chcemy wykorzystać **API Gateway**, usługę, która umożliwia tworzenie, publikowanie i zarządzanie API. To pozwoli nam na udostępnianie naszej aplikacji przez różne platformy i urządzenia.

Dzięki wykorzystaniu zasobów AWS i platformy Amplify, nasza aplikacja SSO będzie łatwa do skalowania, bezpieczna i łatwa do zarządzania.

Rozdział 2. Założenia projektowe

Nasz projekt ma na celu stworzenie aplikacji SSO opartej na platformie AWS Amplify. Główne cele, które chcemy osiągnąć to:

- Umożliwienie użytkownikom logowania się do różnych aplikacji za pomocą jednego zestawu poświadczeń.
- Zapewnienie bezpiecznego uwierzytelniania i autoryzacji użytkowników.
- Łatwość w zarządzaniu tożsamościami użytkowników i aplikacji.
- Możliwość łatwego dodawania nowych funkcji do systemu SSO.

W ramach projektu planujemy stworzyć prototyp naszej aplikacji, który będziemy mogli przetestować i udoskonalić na podstawie feedbacku od użytkowników (Pracownicy SmartBear, członkowie grupy studentów, chętni do testowania programowania). Głównym założeniem jest stworzenie aplikacji, która spełni wymagania użytkowników i zapewni im łatwość i wygodę w korzystaniu.

Planowanie projektu

Zespół podejmując się realizacji projektu stworzenia szablonu infrastruktury AWS oraz mikroaplikacji webowych, które umożliwią skonfigurowanie AWS Cognito i przetestowanie działania stworzonego połączenia bez udziału inżyniera DevOps, powinien przejść przez kilka etapów, takich jak analiza wymagań projektowych, planowanie projektu, implementacja projektu, testowanie i wdrożenie projektu, dokumentacja projektu oraz prezentacja projektu.

Etap 1: Analiza wymagań projektu

W ramach tego etapu zespół skupi się na zrozumieniu wymagań projektowych oraz wymagań stawianych przez firmę SmartBear. Wszyscy członkowie zespołu będą mieli dostęp do materiałów projektowych oraz do specyfikacji wymagań, aby lepiej zrozumieć zakres projektu. W wyniku tego etapu, zespół powinien być w stanie stworzyć plan projektu oraz określić zasoby i czas potrzebny do realizacji projektu.

Etap 2: Planowanie projektu

W ramach tego etapu zespół powinien stworzyć plan projektu, który uwzględnia kroki potrzebne do zrealizowania projektu, a także określenie zasobów, które będą potrzebne do realizacji projektu. W planie projektu powinny znaleźć się etapy, które będą wykonywane przez cały zespół, a także zadania, które będą przypisane poszczególnym członkom zespołu. Plan powinien także określać terminy, w których poszczególne etapy projektu mają zostać ukończone, aby projekt został zrealizowany na czas.

Etap 3: Implementacja projektu

W ramach tego etapu zespół rozpocznie pracę nad implementacją projektu. Każdy z członków zespołu będzie odpowiedzialny za wykonanie swojego zadania w ramach projektu zgodnie z planem projektu. Zespół będzie wykorzystywał różne narzędzia programistyczne, takie jak AWS Lambda, DynamoDB, CDK i Amplify.

Etap 4: Testowanie i wdrożenie projektu

W ramach tego etapu zespół przeprowadzi testy aplikacji, aby upewnić się, że działa ona zgodnie z wymaganiami projektowymi oraz oczekiwaniami firmy SmartBear. Po przeprowadzeniu testów aplikacja zostanie wdrożona, a zespół będzie monitorował jej działanie, aby zapewnić, że wszystko działa prawidłowo.

Etap 5: Prezentacja projektu

Ostatnim etapem projektu będzie prezentacja przedstawiająca realizację projektu oraz wyniki testów. Członkowie zespołu będą mieli okazję zaprezentować swoje rozwiązanie przed firmą SmartBear, aby zaprezentować jak wiele udało się osiągnąć w ramach projektu. Prezentacja będzie okazją do uzyskania informacji zwrotnej od pracowników firmy SmartBear oraz do ewentualnych poprawek i ulepszeń projektu

Rozdział 3. Projekt „Amplify Application”

Rozpoczęcie prac technicznych

Ten etap można nazwać pierwszą techniczną częścią naszego projektu. W ramach etapu "Rozpoczęcie prac technicznych" nasz zespół zacznie prace nad projektem polegającym na zapoznaniu się z usługami AWS, takimi jak Lambda, IAM, Amplify i API Gateway. Naszym pierwszym zadaniem będzie utworzenie konta AWS i skonfigurowanie dostępu do wymienionych usług, aby móc wdrożyć projekt na platformie AWS.

Kolejnym krokiem będzie zapoznanie się z dokumentacją każdej z usług, aby zdobyć wiedzę na temat ich działania i zrozumieć, jak najlepiej wykorzystać je w kontekście projektu. Po zrozumieniu, jak działają te usługi, będziemy mogli zacząć prace nad naszym projektem.

Aby przejść do dalszych kroków, musimy utworzyć repozytorium GitHub, na którym będziemy przechowywać nasz kod źródłowy projektu. Zadbamy o odpowiednie skonfigurowanie repozytorium, stosując najlepsze praktyki GitHub, takie jak korzystanie z gałęzi (branches), pull requestów (pull requests) i code review.

W kolejnym kroku połączymy kod źródłowy projektu z usługą Amplify. To pozwoli nam na łatwe wdrożenie aplikacji, a także automatyczne przeprowadzanie testów i wdrażanie zmian.

Będziemy korzystać z AWS API Gateway, aby udostępnić naszą aplikację użytkownikom i umożliwić im korzystanie z niej poprzez interfejs webowy.

W etapie "Rozpoczęcie prac technicznych" zrealizujemy również wiele innych zadań, takich jak konfiguracja usługi IAM, ustawienie uprawnień dostępu do usług, utworzenie infrastruktury mikro serwisowej w oparciu o AWS Lambda i DynamoDB, a także wiele innych.

Tworzenia okna logowania w Amplify

Główną ideą naszego projektu jest koncepcja logowania przez SSO, ale niezależnie od tego, jak ironicznie to brzmi na początku, logowanie do naszej aplikacji powinniśmy wdrożyć sami.

W pierwszej kolejności utworzymy interfejs użytkownika (UI) dla naszej aplikacji w oparciu o framework Amplify. Będziemy korzystać z gotowych komponentów, takich jak moduł uwierzytelniania, który pozwoli nam na łatwe i szybkie dodanie funkcjonalności logowania do naszej aplikacji.

Następnie, skonfigurujemy usługę uwierzytelniania w Amplify. Zdefiniujemy sposób uwierzytelniania użytkowników (np. poprzez e-mail i hasło), a także ustalimy polityki bezpieczeństwa i dostępności, takie jak wymagane minimalne hasło czy wymagane potwierdzenie adresu e-mail.

W kolejnym kroku połączymy nasze okno logowania z usługą uwierzytelniania, aby użytkownicy mogli bezpiecznie logować się do aplikacji i korzystać z jej funkcjonalności.

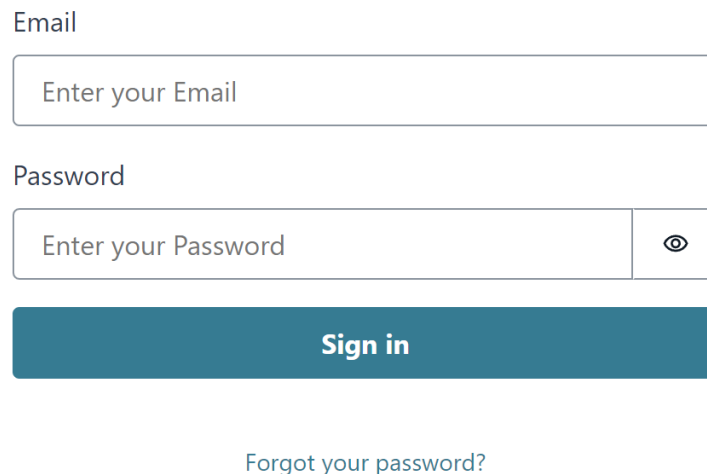
Wynik pracy nad tą częścią projektu:

The image shows a login form with the following elements:

- Two tabs at the top: "Sign In" (highlighted in blue) and "Create Account" (grey).
- An "Email" label above a text input field containing the placeholder "Enter your Email".
- A "Password" label above a text input field containing the placeholder "Enter your Password". To the right of the password field is an eye icon for toggling visibility.
- A large blue button labeled "Sign in" below the password field.
- A link labeled "Forgot your password?" below the "Sign in" button.

Rysunek 10 Okno logowania stworzone za pomocą AWS Amplify

Można zobaczyć, że istnieje opcja utworzenia nowego użytkownika z GUI, ale ponieważ nie jest to potrzebne, okno zostało zmodyfikowane tak, aby użytkownika na tym etapie można było stworzyć wyłącznie ręcznie w Cognito User Pool



The image shows a login form with the following elements:

- An "Email" label above a text input field containing the placeholder "Enter your Email".
- A "Password" label above a text input field containing the placeholder "Enter your Password". To the right of the password field is a toggle icon (an eye).
- A large blue button labeled "Sign in" below the password field.
- A link labeled "Forgot your password?" centered below the "Sign in" button.

Rysunek 2 Modyfikowane okno logowania stworzone za pomocą AWS Amplify

Tworzenie pierwszych funkcji Lambda

Rozpoczynamy etap tworzenia pierwszych funkcji Lambda dla naszego projektu. Zdecydowaliśmy, że będziemy korzystać z języka Python w wersji 3.8.

Pierwszą funkcją, którą będziemy tworzyć, jest `CreateUserPool.py`. Funkcja ta ma na celu stworzenie puli użytkowników w usłudze Cognito. Następnie, zaimplementujemy funkcję `StartNewProject.py`, będzie pełnić rolę głównej funkcji i będzie mógł aktywować inne Lambda.

Kolejną funkcją, którą stworzymy, jest `CreateUser.py`. Ta funkcja będzie służyć do dodawania nowych użytkowników do puli użytkowników w usłudze Cognito.

Po tym, stworzymy funkcję `AddUserToGroup.py`, która pozwoli na dodanie nowego użytkownika do grupy użytkowników w usłudze Cognito. Ostatnią funkcją, którą stworzymy, będzie `CreateGroup.py`, umożliwiającą utworzenie nowej grupy użytkowników w usłudze Cognito.

Wszystkie te funkcje będą wykorzystywane do zarządzania pulą użytkowników oraz grup użytkowników w usłudze Cognito. W tym celu wykorzystamy serwisy Lambda i IAM.

Lambda > Functions > Create function

Create function Info

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ **Author from scratch**
Start with a simple Hello World example.

☐ **Use a blueprint**
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**
Select a container image to deploy for your function.

Basic information

Function name Info
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture Info
Choose the instruction set architecture you want for your function code.
☒ **x86_64**
☐ arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Rysunek 3 Przykładowe tworzenie Lambdy

Na tym etapie nasze Lambdy są jeszcze oddzielone od głównego projektu i dołączą nieco później. Możemy podsumować, że teraz opracowujemy funkcjonalność tak, jak powinna wyglądać w przyszłości.

Tworzenie pierwszych IAM

W ramach projektu będziemy również tworzyć pierwsze uprawnienia IAM do naszych funkcji Lambda.

Najpierw utworzymy rolę IAM dla naszych funkcji, która pozwoli na dostęp do usługi Cognito oraz do usługi Amplify. Następnie, dla każdej z funkcji Lambda, utworzymy politykę IAM, która umożliwi jej wykonywanie operacji na usłudze Cognito i Amplify.

Polityki te będą zapewniać dostęp tylko do niezbędnych zasobów i operacji, aby ograniczyć ryzyko naruszenia bezpieczeństwa.

Tworzenie uprawnień IAM jest ważnym elementem w projektowaniu aplikacji w chmurze, ponieważ pozwala na kontrolowanie dostępu do zasobów i operacji w sposób zabezpieczający dane.

Tworzenie wspólnego konta AWS

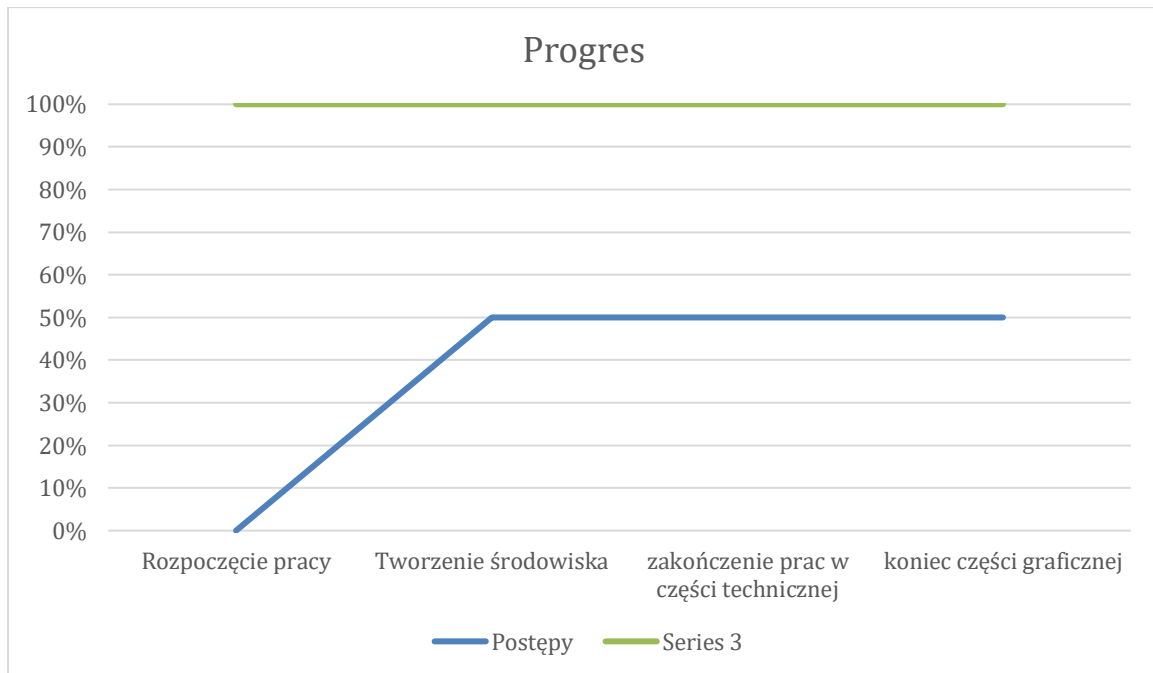
Ta część jest obecnie wdrażana przez zespół, zostanie dodana, jak tylko otrzymamy gotowe działające rozwiązanie.

Podłączenie Lambda do projektu Amplify

Ta część jest obecnie wdrażana przez zespół, zostanie dodana, jak tylko otrzymamy gotowe działające rozwiązanie.

Rozdział 4. Badania i wnioski

Sekcja zostanie wypełniona podczas fazy testowania programu



W chwili obecnej jesteśmy pomiędzy momentem powstania środowiska a zakończeniem prac technicznych. Mamy już utworzony oddzielnie backend, pozostaje tylko go podłączyć. Po podłączeniu zostanie tylko realizacja części graficznej.

Podsumowanie

Projekt nie został jeszcze ukończony.

Bibliografia

GitHub od zespołu:

<https://github.com/pwr-smartbear-AWS-SSO/amplify-app>

<https://aws.amazon.com/cognito/details/>

<https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/amplify.html>

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/apigateway.html>

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/cloudfront.html>

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/dynamodb.html>

<https://aws.amazon.com/amplify/authentication/?nc=sn&loc=3&dn=1>

<https://aws.amazon.com/amplify/features/?nc=sn&loc=3&dn=5>

