Wprowadzenie

Ta książka pomoże Ci rozpocząć pracę z AWS Lambda oraz narzędziem Serverless Application Model (SAM). Lambda jest usługą dostępną na platformie Amazon Web Services do uruchamiania kodu, który jest sterowany zdarzeniami, a SAM to rozwiązanie otwartoźródłowe (open source), które w znaczący sposób upraszcza konfigurację i wdrażanie oparte na wyżej wymienionej usłudze. Wspólnie pozwalają na tworzenie w łatwy sposób automatycznie skalujących się API oraz serwisów zaprojektowanych pod wdrożenia klasy serverless. W kolejnych rozdziałach dowiesz się, jak:

- stworzyć aplikacje, które w pełni wykorzystają potencjał architektury serverless;
- zbudować automatycznie skalujące się API;
- obsługiwać wykonywanie zadań w tle oraz różne modele obsługi zdarzeń;
- skonfigurować potok wdrożeniowy, tak aby praca w zespole była efektywna;
- testować oraz rozwiązywać problemy w aplikacjach zaprojektowanych na platformę AWS Lambda;
- sprawdzać i monitorować rozwiązania klasy serverless.

Książka została przygotowana na podstawie moich własnych doświadczeń zdobytych w trakcie budowy narzędzia MindMup, służącego do przygotowań map myśli w formie zespołowej pracy zdalnej, które było jednym z pierwszych rozwiązań opartych na usłudze AWS Lambda. MindMup zmigrował na architekturę serverless z klasycznego rozwiązania opartego na hostingu w trakcie roku 2016, tak aby skorzystać z mechanizmów skalowania na żądanie. Dzięki temu zredukowaliśmy nasze koszty operacyjne o około dwie trzecie, zwiększając w tym samym czasie możliwości aplikacji, przyspieszając proces wytwarzania oprogramowania oraz obniżenie czasu wprowadzenia na rynek. Wnioski z tej migracji zostały zebrane w postaci jednego z pierwszych opracowań naukowych na ten temat oraz były podstawą do przygotowania wystąpień na wielu konferencjach na całym świecie.

Kiedy piszę te słowa w styczniu 2020 roku, mając prawie czteroletnie doświadczenie w pracy z systemem klasy serverless, bardzo łatwo jest zauważyć pojawiające się wzorce architektoniczne oraz wybrać te eksperymenty, które przerodziły się w dobre pomysły, i wyeliminować te, które prowadziły donikąd. Ta książka ma Ci pomóc przejść tę drogę szybciej, tak aby skorzystać z dobrych praktyk oraz uniknąć pułapek.

Kolejne rozdziały pokażą, jak zbudować taką aplikację w praktyce. Rozpoczniemy od prostego statycznego API i iteracyjnie rozbudujemy ją do pełnej i prawdziwej aplikacji służącej do skalowania obrazów, gotowej do przyjęcia milionów użytkowników, ze wszystkimi usługami wspierającymi wdrożenie i zarządzanie takim produktem. Przygotowany system będzie przypominał to, z czym wielu z Was pracuje na co dzień. W ten sposób przygotujemy solidny fundament pod ewentualne przyszłe wykorzystanie, włącznie z kopiowaniem fragmentów na początek. W trakcie tworzenia i wdrażania różnych części aplikacji dowiesz się, jak działa AWS Lambda i usługi pokrewne, poznasz także wskazówki, techniki i narzędzia, które pomagają w tworzeniu rozwiązań klasy serverless.

Omawiany przykład wyewoluował z kilkunastu otwartych warsztatów i wystąpień konferencyjnych przeze mnie przeprowadzonych. Wraz z kolegami wykorzystywaliśmy przykłady zamieszczone w tej książce podczas nauki setek programistów, jak budować aplikacje w architekturze serverless. Dzięki temu przygotowane przykłady zostały sprawdzone w praktyce i usprawniono korzystanie z uwag oraz doświadczeń płynących z warsztatów.

Dlaczego AWS SAM?

Kiedy rozpoczęliśmy migrację MindMup do architektury serverless, SAM nie był jeszcze dostępny. Oznaczało to, że musieliśmy przygotować podobne narzędzie własnoręcznie. Gdybyśmy podchodzili teraz do tego problemu, wykorzystanie SAM byłoby oczywistym wyborem.

Jeśli interesujesz się tym tematem, to wiesz o tym, że SAM nie jest ani jedynym, ani najpopularniejszym, ani także nie najlepszym czy najłatwiejszym narzędziem w użyciu (biorąc pod uwagę, że piszę te słowa w czerwcu 2019). Jednakże SAM ma dwie znaczące zalety: jest ono efektywne, jeśli chodzi o pracę z nim, oraz jest ono wspierane i przygotowywane przez AWS. Szczególnie to ostatnie jest szalenie ważne, ponieważ oznacza solidne komercyjne wsparcie i jest bezpiecznym wyborem na przyszłość.

Użytkownicy tego narzędzia są beneficjentami wolności związanej z faktem, że jest to rozwiązanie o otwartym kodzie źródłowym, oraz ze świetną integracją z platformą dostawcy, która jest charakterystyczna dla komercyjnych rozwiązań, o zamkniętym kodzie. W tym przypadku kod jest dostępny na platformie Github, więc każdy może go przeanalizować i ocenić. Dzięki tym dwóm faktom dookoła narzędzia powstała znacząca społeczność, która chętnie pomaga w sytuacji ewentualnych problemów i wyzwań. Dodatkowo większość kontrybutorów to pracownicy AWS. Z racji tego, że cały ekosystem serverless szybko się zmienia, fakt, że autorzy tego narzędzia mają bezpośredni dostęp do produktów i usług na AWS oraz zespołów, które za nimi stoją, jest nie do przecenienia – przykładem jest chociażby to, jak szybko to narzędzie się rozwija.

Wykorzystanie AWS SAM powinno być relatywnie łatwe dla większości organizacji, ponieważ jest ono zbudowane de facto na standardzie dla wdrożeń na platformie AWS o nazwie CloudFormation (powiemy więcej o tym narzędziu w rozdziale 3). Dla firm, które już teraz używają CloudFormation, wykorzystanie AWS SAM oznacza tylko drobne zmiany w procesie przygotowania szablonów w tym narzędziu. Nie ma potrzeby przygotowywania nowego procesu czy nauki kompletnie nowego narzędzia. Dodatkowo, SAM w znacznym stopniu redukuje powtarzalność w celu przygotowania infrastruktury klasy serverless, co ułatwia pracę w szczególności początkującym użytkownikom CloudFormation.

Grupa docelowa

To jest podręcznik techniczny i w związku z tym uważam, że będzie pomocny dla dwóch grup programistów oraz architektów oprogramowania:

- tych bez wcześniejszego doświadczenia w zakresie aplikacji typu serverless, ale zainteresowanych nauką na temat nowych podejść związanych z architekturą aplikacji korzystających z chmury;
- tych pracujących z AWS Lambda, ale korzystających z innych narzędzi i chcących poznać AWS SAM oraz podejście nazwane przez inżynierów Amazona Serverless Application Model.

Drogi Czytelniku, nie potrzebujesz żadnej wiedzy na temat chmury, aby zrozumieć przedstawione przykłady, ale będziesz musiał umieć przeanalizować i przeczytać kod w języku JavaScript.

W momencie, gdy piszę te słowa, AWS Lambda pozwala na uruchamianie kodu napisanego na sześciu platformach: OpenJDK (dla języka Java i pokrewnych), .NET Core (dla języka C# i pokrewnych), Go, Ruby, Python oraz JavaScript (z wykorzystaniem Node.js). Dodatkowo, jest możliwe przygotowanie własnego środowiska uruchomieniowego w celu wsparcia innych platform oraz języków programowania.

Wybór języka programowania, w którym będziesz programować swoje funkcje, nie jest szalenie ważny, ale aby zachować spójność i uczynić tę książkę łatwiejszą do przyswajania wiedzy, wybrałem tylko jeden z nich. Wybór padł na JavaScript, ponieważ jest to język najbliższy uniwersalnemu dla

wszystkich programistów w roku 2020. Wszystkie dobre praktyki przedstawione w książce mogą być z łatwością zaaplikowane na każdej z pozostałych wspieranych platform, a różnice, jakie będą występowały między platformami, zostały wyjaśnione w treści książki. Kompletny i działający kod źródłowy wszystkich przykładów oraz przykłady w innych językach programowania możesz pobrać z oficjalnej strony: https://runningserverless.com.

Jak czytać tę książkę?

Pierwszy rozdział niniejszej książki to szybkie wprowadzenie w podstawowe pojęcia aplikacji typu serverless oraz omówienie wad i zalet, jakie wynikają z wykorzystania tego podejścia. Jeśli jest to dla Ciebie kompletnie nowe, zachęcam do przeczytania i potraktowania tego jako zwiastun tego, czego możesz spodziewać się później. Jeżeli jednak ta tematyka jest Ci bliska i posiadasz doświadczenie w tym zakresie, możesz śmiało pominąć tę część książki.

Drugi rozdział wyjaśnia, jak skonfigurować narzędzie SAM, tak aby móc tworzyć i testować funkcje Lambda na lokalnej maszynie. Zgodnie z tą instrukcją przygotowujemy również takie środowisko, aby na bieżąco wypróbowywać przykłady znajdujące się w książce. Jeśli nie planujesz korzystać z przykładów, możesz śmiało pominąć ten rozdział.

Rozdziały od drugiego włącznie są kompletnym przewodnikiem, jak zbudować wysoce skalowalną aplikację, która w pełni korzysta z chmury. To ważne, aby czytać rozdziały 2–12 w zadanej kolejności. Na początku nauczysz się, jak wykonać najprostsze wdrożenie i jak przygotować projekt, tak aby mógł nad nim pracować cały zespół. W trakcie rozbudowy naszego przykładu dowiesz się o wysokopoziomowych aspektach, takich jak architektura aplikacji, jak dobrać właściwe usługi dostępne w portfolio AWS oraz jak wykorzystać potencjał, który drzemie w serverless. W każdym rozdziale znajdziesz ważne wskazówki i dlatego polecam, żeby nie omijać żadnej sekcji, nawet jeśli znasz ten konkretny temat.

Ostatni rozdział wyjaśnia wzorce architektoniczne oraz modele wdrożeniowe dla typowych przykładów aplikacji klasy serverless. Z racji tego, że tematyka ta jest jeszcze relatywnie młoda i trudno mówić o zestawie złotych rad oraz dobrych praktykach to ten rozdział pomoże Ci wykorzystać pełen potencjał AWS Lambda i pokrewnych usług, jeśli to Ty staniesz przed wyzwaniem zaprojektowania systemu z wykorzystaniem tej architektury.

Kod źródłowy

Aby skupić się na najważniejszych elementach i niepotrzebnie nie rozdmuchiwać przykładów, starałem się przygotować okrojone do niezbędnego minimum fragmenty kodu źródłowego. Każda osoba, która ma doświadczenie z językiem JavaScript, będzie w stanie odtworzyć samodzielnie kompletną aplikację, bazując na przygotowanych przykładach, ale przygotowany kod jest zoptymalizowany, tak aby go łatwo zrozumieć i przeczytać, a nie kopiować.

Pełny kod aplikacji i wszystkich przykładów jest dostępny na stronie https://runningserverless.com. Najważniejsze listingi posiadają w tytule ścieżkę do pliku. Na przykład poniższy fragment można znaleźć w kompletnej paczce z kodem źródłowym w liniach 126 i 127 znajdujących się w pliku template-with-dlq.yaml, wewnątrz katalogu ch9 (rozdział 9).

ch9/template-with-dlq.yaml

```
126 NotifyAdmins:
127   Type: AWS::SNS::Topic
```

Przykłady komend

Narzędzia AWS wywoływane z linii komend bardzo często wymagają długiej listy parametrów, która przekracza długość linii na małym ekranie lub stronie książki. Jeśli faktycznie zabraknie miejsca na taką linijkę, w kolejnym wierszu znajdziesz małą strzałkę, która podpowie Ci, że należy traktować całość jako jeden ciąg. Oto przykład:

```
sam deploy --template-file output.yaml --stack-name sam-test-1 --capabilities CAPABILITY_IAM
↪  --profile sam-test-profile --region us-east-1
```

Aby wizualnie rozróżnić wywołania komend od jej wyniku, same komendy, które należy wywołać, nie będą miały numerów linii. Rezultaty zaś będą taki numer posiadały. Kolejny listing pokazuje, jak wygląda wywołanie komendy:

```
aws sts get-caller-identity
```

Następny listing pokazuje rezultat poprzedniej komendy, wraz z numerami linii dołączonymi do wydruku:

```
1 $ aws sts get-caller-identity
2 {
3     "UserId": "111111111111",
4     "Account": "222222222222",
5     "Arn": "arn:aws:iam:1111111111:root"
6 }
```

Wskazówki

Ważne miejsca i wskazówki będą jasno oznaczone w tekście i będą zaprezentowane za pomocą następującego obramowania:

Kod źródłowy

Nie kopiuj kodu źródłowego zamieszczonego w książce. Pobierz całą paczkę ze strony: https://runningserverless.com.

Będę korzystał z identycznego mechanizmu do wyjaśnienia kluczowych pojęć oraz ważnych informacji wartych zapamiętania, w szczególności jeśli będą wymagały specjalnej uwagi lub przeczą intuicji.