wydział elektrotechniki elektroniki informatyki i automatyki

AMG.net
A Bull Group Company

PROJEKTOWANIE SYSTEMÓW OBIEKTOWYCH I ROZPROSZONYCH

LABORATORIUM 4

Simple Storage System (S3)

wersja 1.1

przygotował:
Radosław Adamus

.

HISTORIA WERSJI

DATA	WERSJA	AUTOR	OPIS
07.03.2014	0.1	Radosław Adamus	Pierwsza robocza wersja dokumentu
28.03.2014	1.0beta1	Radosław Adamus	Opisane laboratorium, brak linku do projektu
30.03.2014	1.0	Radosław Adamus	Pierwsza oficjalna wersja opisu
31.03.2014	1.1	Radosław Adamus	Poprawki edycyjne i uzupełnienia opisu

Cel:

Celem laboratorium jest:
1. Zapoznanie się i praktyczne wykorzystanie architektury aplikacji webowej wykorzystującej system S3 jako mechanizm trwałego przechowywania danych.

Wymagania wstępne:

1. Posiadanie konta na platformie Github.
2. Skonfigurowane konto AWS
3. Rozumienie znaczenia metod komunikatu żądania (request) protokołu HTTP w kontekście architektury REST.

Narzędzia:

Git, nodeJS, edytor programistyczny (np. Notepad++).

Reguły wykonywania ćwiczeń laboratoryjnych:

1. Po ukończeniu laboratorium należy wyłączyć wszystkie działające instancje EC2 i/lub wyzerować ustawienia dotyczące docelowej, minimalnej i maksymalnej liczby instancji w usłudze ASG.

2. Otrzymane dane autoryzacyjne (hasła oraz klucze dostępu) są danymi wrażliwymi i muszą być chronione. W szczególności nie można dodawać do repozytorium kontroli wersji oraz pozwolić na wysłanie na usługi hostujące repozytoria kontroli wersji kodu źródłowego (GitHub) plików zawierających konfigurację autoryzacji dostępu do API AWS.

3. Zmiany zatwierdzane w repozytorium powinny mieć znaczące komentarze.

Opis laboratorium:

1. Informacje podstawowe

Usługa S3

Amazon S3 (Simple Storage Service) to usługa udostępniająca “internetowy dysk” pozwalający na wysoko skalowalne oraz niezawodne przechowywanie dowolnych danych (obiektów). Pobieranie i zapisywanie danych odbywa się za pośrednictwem prostego interfejsu usługi internetowej S3.
Dane przechowywane w usłudze S3 są składowane w tzw. kubełkach (ang. buckets) - będących odpowiednikami dysków. Służą one przede wszystkim organizowaniu przestrzeni nazw usługi S3, identyfikacji konta właściciela oraz stanowią podstawę kontroli dostępu do danych. Kubełki są przechowane w wybranym regionie (Region) i mogą dodatkowo wersjonować przechowywane obiekty.
Podstawowa jednostka danych przechowywana w usłudze S3 nosi nazwę obiektu. Na obiekt składają się faktyczne dane oraz ich opis (metadane). Dane, zapisane w ramach obiektu dane są niepodzielne i nieprzezroczyste z punktu widzenia usługi. Natomiast metadane to zbiór par

klucz-wartość. Każdy obiekt zawiera domyślny zestaw metadanych które mogą być uzupełnianie przez dowolne wartości dodawane w trakcie tworzenia obiektu. Każdy obiekt posiada również unikatowy identyfikator. Składa się on z adresu kubełka, klucza (unikatowej nazwy) obiektu w obrębie kontenera oraz identyfikatora wersji (opcjonalnie). Każdy obiekt w kontenerze posiada dokładnie jeden, unikatowy klucz. Np.:

http://lab4-weeia.s3.amazonaws.com/test/chaos.txt

lub

https://s3-us-west-2.amazonaws.com/lab4-weeia/test/chaos.txt

jest identyfikatorem obiektu. Składa się on z:

1. Identyfikatora kubełka (bucket):

   http://lab4-weeia.s3.amazonaws.com/ - w tzw. “virtual-hosted” style (niezależny od regionu)

   lub

   https://s3-us-west-2.amazonaws.com/lab4-weeia/ - w tzw “path-style” (uwzględniający endpoint specyficzny dla regionu).

2. Klucza obiektu:

   test/chaos.txt

Blokowanie i transakcje

Usługa S3 nie zapewnia blokowania obiektów, oraz atomowych transakcji, dlatego współbieżna aktualizacja tego samego klucza oraz atomowa aktualizacja wielu kluczy musi być, jeżeli jest to wymagane, obsługiwana na poziomie aplikacji.

Kontrola dostępu

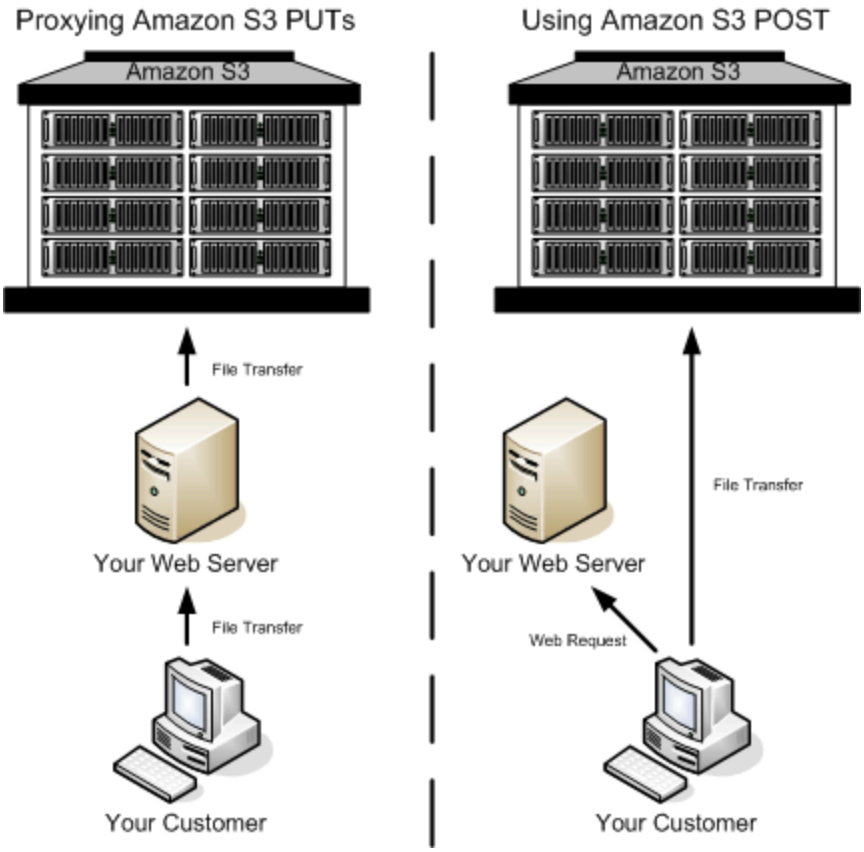
Kontrola dostępu do zasobów przechowywanych w S3 może być konfigurowana tak na poziomie kontenera jak i poszczególnych obiektów. Określanie praw (np. do zapisu) może uwzględniać nie tylko użytkowników i kontenery ale również sieć z której wysyłane jest żądanie, czas w którym zostało zgłoszone oraz aplikację, która była jego źródłem.

Dostęp programowy

Programowy dostęp do usługi S3 możliwy jest za pośrednictwem interfejsu REST. Interfejs może być wykorzystany bezpośrednio lub za pośrednictwem AWS-SDK dla wybranego języka programowania. W przypadku komunikacji wymagającej autoryzacji, wywołujący musi, wraz z żądaniem, przesłać sygnaturę, której wartość identyfikuje wysyłającego. Sygnatura generowana jest na podstawie kluczy dostępu. W zależności od sposobu interakcji z serwisem, sygnatura generowana jest automatycznie (AWS API) lub musi być wyliczona przez aplikację (REST API).

2. Przesyłanie plików na do usługi S3 bezpośrednio z przeglądarki

Usługa S3, poprzez obsługę metody POST komunikatu żądania HTTP pozwala na przesyłanie danych do kontenera z pominięciem serwera webowego. Pozwala to na minimalizację opóźnień oraz zmniejszenie obciążenia serwera webowego (co może mieć pozytywny wpływ na koszty). Różnica w architekturze pomiędzy podejście klasycznym a wykorzystującym metodę POST przedstawiona jest na rysunku 1.



Rysunek 1: Przesyłanie danych do usługi S3 z i bez pośrednictwa serwera webowego (źródło http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingHTTPPOST.html).

Ponieważ metoda POST jest bezpośrednio dostępna za pośrednictwem formularzy HTML, przesyłanie pliku nie wymaga żadnych specjalnych zabiegów poza odpowiednim przygotowaniem struktury dokumentu HTML. Dane przesyłane w ramach żądania do usługi S3 muszą, poza danymi identyfikującymi miejsce składowania pliku (bucket, key), posiadać również informacje pozwalające na uwierzytelnienie operacji. Na informacje te składają się: klucz dostępu (access key id), dokument "policy" oraz sygnatura. Dokument "policy" umożliwia weryfikację przez usługę, czy dane w formularzu nie zostały złośliwie zmodyfikowane, a sygnatura stanowi podpis dokumentu "policy". Do jej wygenerowania wymagane jest z wykorzystanie drugiego klucza z pary (secret access key). Szczegóły dotyczące zasad budowania formularza HTTP dla potrzeb usługi dostępne są w HTTPPOSTForms oraz Browser Uploads to S3 using HTML POST Forms.

3. Zadania

Zadania wykorzystują jako podstawę projekt w repozytorium GitHub o identyfikatorze: https://github.com/amgnet-weeia/awslab4 .
Repozytorium należy skopiować (fork) na swoje konto. Rozwiązania (**pamiętając o regułach bezpieczeństwa dotyczących przechowywania kluczy**) powinny być umieszczane w tym repozytorium.

W obecnej wersji projektu mapowanie ścieżki na akcję odbywa się poprzez dodanie wpisu w pliku *actions.json* oraz utworzenie pliku skryptu w folderze actions/. Skrypt powinien eksportować funkcję, o parametrach request i callback, pod nazwą *action*, np.:

```
exports.action = function(request, callback) {
      callback(null, "Hello" + request.params.name);
}
```

1. Wysyłanie plików za pośrednictwem formularza

Uzupełnij aplikację w taki sposób, aby wyświetlał odpowiednio skonfigurowany formularz za pomocą którego będzie możliwe zapisywanie plików w usłudze S3. W obecnej wersji aplikacja wyświetla formularz, jednak brak jest konfiguracji wymaganej przy wywoływaniu usługi. Wstępna konfiguracja zapisana jest w pliku *policy.json* (należy ją uzupełnić podając nazwę kubełka, prefix klucza oraz adres przekierowania po poprawnym zapisaniu danych).
Do rozwiązania zadania możesz wykorzystać mechanizmy dostępne w skryptach *s3post.js* oraz *helpers.js*.
Zmodyfikuj aplikację w taki sposób, aby razem z plikiem w metadanych obiektu zapisywane było Twoje imię i nazwisko oraz adres komputera na którym wygenerowany został formularz.

2. Wyliczenie wartości skrótu

Poprawne zapisanie dokumentu na S3 powinno wywołać akcję pobrania zapisanego dokumentu i wyliczenie dla niego wartości skrótów (http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/S3.html#getObject-property).
Do wyliczenia wartości skrótów można wykorzystać funkcje skryptu helpers.js (patrz implementacja skryptu digest.js). Rezultat, wyświetlany w przeglądarce powinien prezentować obliczone skróty, wraz z kluczem pliku oraz jego metadanymi.

3. Instalacja aplikacji na instancji EC2

Zainstaluj aplikację na instancji EC2. Zastanów się w jaki sposób zautomatyzować proces wdrażania aplikacji na EC2.

4. Rekonfiguracja aplikacji (opcjonalnie)

Podstawowe ustawienia dokumentu "policy" oraz formularza, aplikacja pobiera z pliku *policy.json*. Przenieś ten plik do usługi S3 i zaimplementuj możliwość pobierania tej konfiguracji podczas startu aplikacji.

Pytania/zadania uzupełniające

1. Usługa Amazon S3 zapewnia spójność typu 'eventual consistency' dla wszystkich typów żądań w regionach US. Dla pozostałych regionów zapewniana jest spójność typu 'read-after-write' dla żądania PUT wstawiającego nowy obiekt oraz 'eventual consistency' dla żądań nadpisujących (PUT, DELETE). Co to oznacza?
2. Jaki interfejs udostępnia usługa S3?

Materiały:

1. Wprowadzenie do S3: http://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html

2. Wersjonowanie obiektów w S3:
http://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectVersioning.html
3. Koszty S3 http://aws.amazon.com/s3/pricing/
4. Struktura formularz HTTP POST dla wysyłania plików do usługi S3:
http://docs.aws.amazon.com/AmazonS3/latest/dev/HTTPPOSTForms.html
5. Wskazówki dotyczące budowy formularzy HTTP POST dla S3
http://aws.amazon.com/articles/1434
6. Usługa S3 w Java Script SDK
http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/S3.html