

Projektowanie aplikacji w chmurze obliczeniowej

Michał Baliński
IT Manager, AMG.net



Politechnika
Łódzka



Wykorzystuj zalety chmur obliczeniowych

- Samoobsługa na żądanie (*ang. on-demand self-service*)
- Elastyczność i skalowalność (*ang. elasticity and scalability*)
- Globalna infrastruktura (*ang. global infrastructure*)
- Automatyzacja (*ang. automation, Infrastructure as Code*)
- Mierzalność usług (*ang. measured service usage*)
- Płatność za zużycie (*ang. pay per use*)

Dobre praktyki w chmurach obliczeniowych

- Projektuj z myślą o problemach (*ang. design for failure*)
- Luźne powiązania (*ang. loosely coupling*)
- Zabezpieczaj każdy komponent (*ang. secure each component*)
- Nie bój się ograniczeń (*ang. don't fear constraints*)
- Myśl równolegle (*ang. think parallel*)
- Bezstanowość (*ang. stateless*)
- Dobieraj technologie i usługi do potrzeb (*ang. choose right technologies*)

11 Dostępność (niezawodność)

Miara dostępności (ang. availability) – jedna z podstawowych miar określenia stopnia odporności systemu. Pojęcie dostępności oznacza czas bezawaryjnego działania usługi w stosunku do całości czasu, w którym usługa ta powinna być klientom świadczona.

$$A = \frac{T_m}{T_m + T_d} \begin{cases} A - \text{dostępność} \\ T_m - \text{średni czas do wystąpienia usterki} \\ T_d - \text{średni czas naprawy} \end{cases}$$

Źródło: [http://pl.wikipedia.org/wiki/Dostępność_\(niezawodność\)](http://pl.wikipedia.org/wiki/Dostępność_(niezawodność))

Klasy dostępności

Typ systemu	Czas niedostępności (w roku)	Czas niedostępności (w miesiącu)	Dostępność	Klasa dostępności
Unmanaged	36,5 dni	72 godzin	90%	1
Managed	3,65 dni	7,2 godzin	99%	2
Well Managed	8,76 godzin	43,8 minut	99,9%	3
Fault Tolerant	52,56 minut	4,32 minut	99,99%	4
High-Availability	5,26 minut	25,9 sekund	99,999%	5
Very-High-Availability	31,5 sekund	2,59 sekund	99,9999%	6
Ultra-Availability	3,15 sekund	0,259 sekund	99,99999%	7

SLA (ang. Service Level Agreement)

oznacza umowę utrzymania ustalonego poziomu jakości usług między klientem a usługodawcą. Potocznie przyjmuje się, że SLA oznacza klasę dostępności usług.

System wysokiej dostępności (*ang. HA – High Availability*)

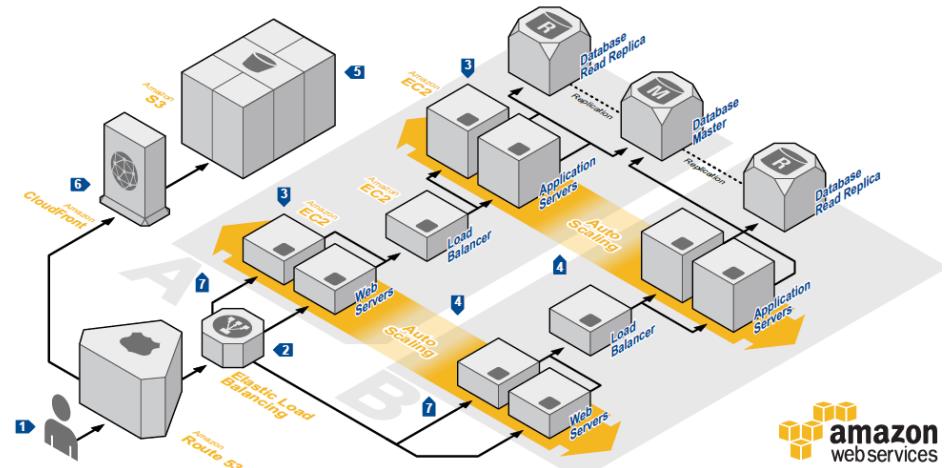
- system informatyczny charakteryzujący się odpowiednio dostosowywaną: niezawodnością, dostępnością i wydajnością; do specyficznych zwykle krytycznych zastosowań.
- zwyczajowo poprzez HA rozumie się systemy i praktyki zapewniające dostępność powyżej klasy 3 (99,9%)
- HA – zestaw praktyk i podejście do projektowania systemów aby zapewnić odpowiednią klasę dostępności

- Redundancja
 - Active – active
 - Active – passive
- Replikacja danych
- Odporność na awarie (*ang. fault tolerance, graceful degradation*)
 - Zdolność do częściowego działania pomimo awarii
- Wzorzec „bezpiecznika” (*ang. circuit breaker*)
 - Odłączanie elementów systemu w celu ochrony przed awarią
- Eliminowanie SPoF (*ang. Single Point of Failure*)
- Automatyzacja reagowania na awarie
- Testy, testy, testy ...

Projektuj z myślą o problemach (*ang. Design for failure*)

“Everything fails, all the time”

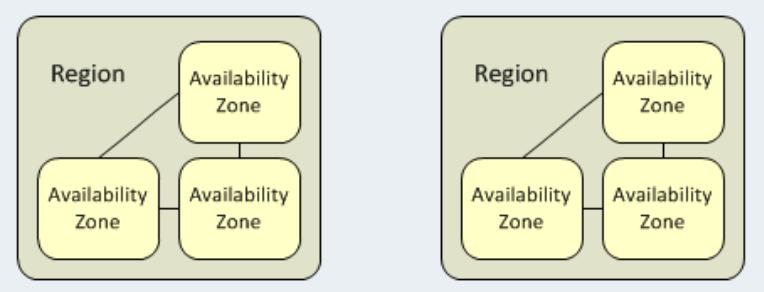
Werner Vogels, CTO Amazon.com



- Skalowalność (*ang. scalability*)
 - zapewnienie coraz wydajniejszej pracy w miarę zwiększania zapotrzebowania na moc obliczeniową
- Modele skalowania:
 - Wertykalny (silniejsze maszyny)
 - Horyzontalny (więcej maszyn)



Regiony i strefy dostępności w chmurze



Globalne (HA)

 Route 53
Scalable Domain Name System

 CloudTrail
User Activity and Change Tracking

 IAM
Secure AWS Access Control

 CloudFront
Global Content Delivery Network

Regionalne (HA)

 S3
Scalable Storage in the Cloud

 DynamoDB
Predictable and Scalable NoSQL Data Store

 SQS
Message Queue Service

 SNS
Push Notification Service

 SES
Email Sending Service

Strefowe (HA – trzeba zapewnić poprzez architekturę)

 EC2
Virtual Servers in the Cloud

 ElasticCache
In-Memory Cache

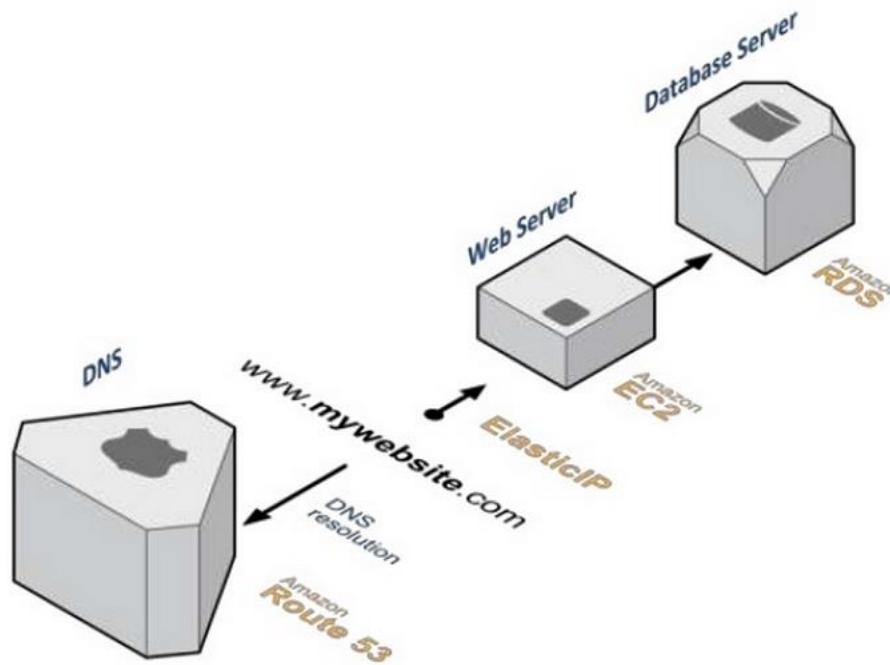
 VPC
Isolated Cloud Resources

 RDS
Managed Relational Database Service

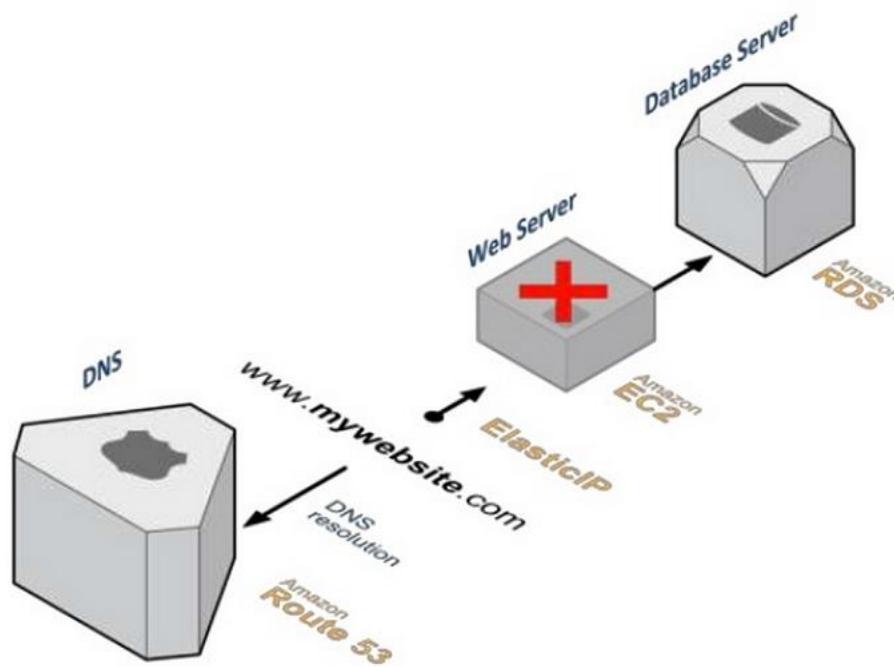
” Aplikacja webowa

Wysoko dostępna i skalowalna

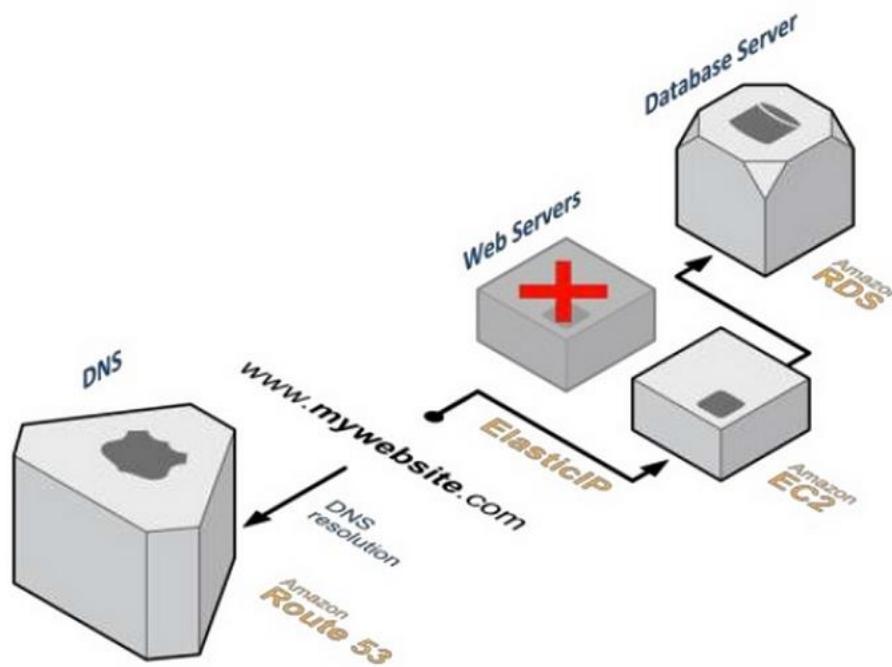
Wyjściowa aplikacja webowa



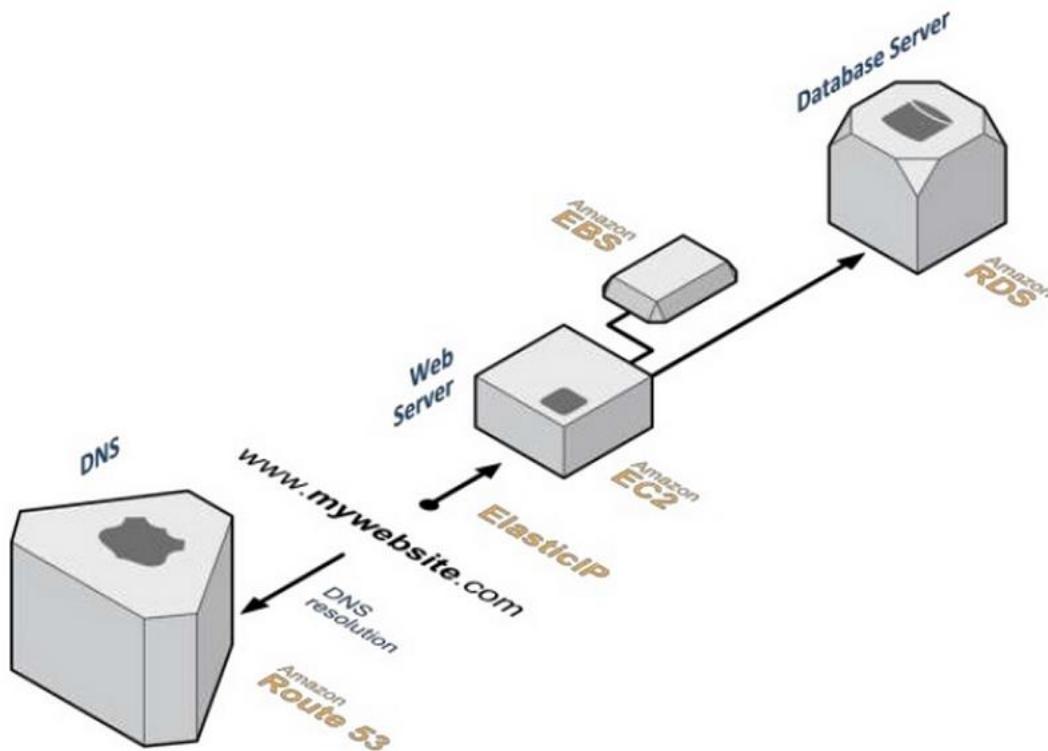
Awaria pojedynczego serwera



Zastąpienie serwera

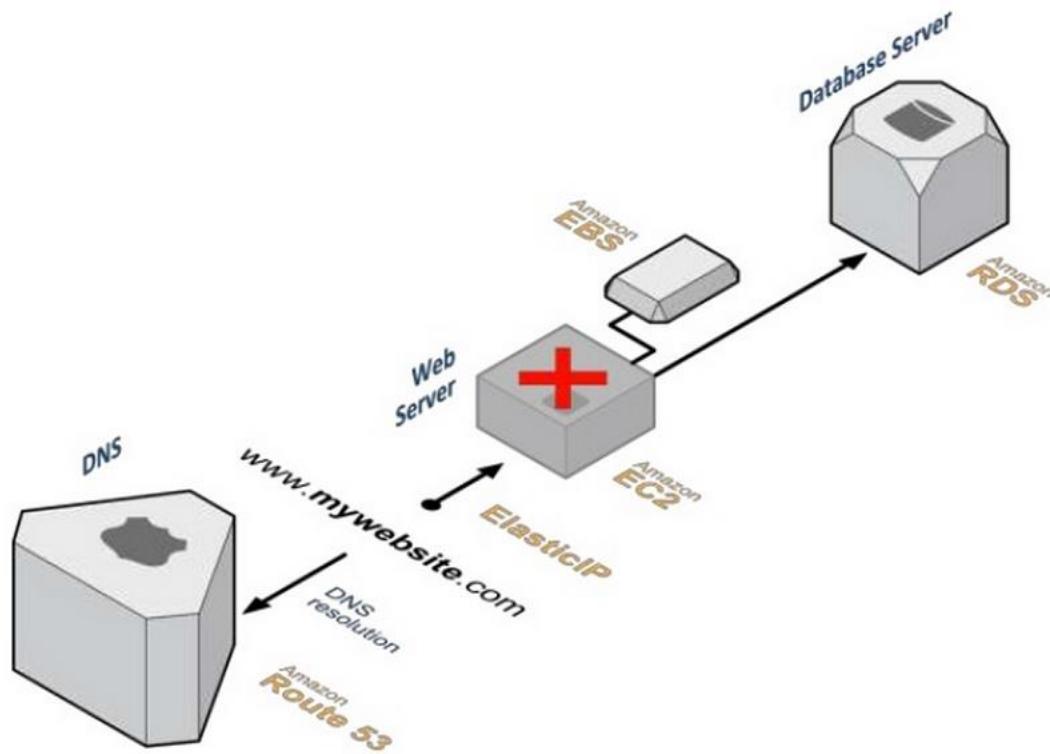


Problem: utrata danych na dyskach serwera który uległ awarii

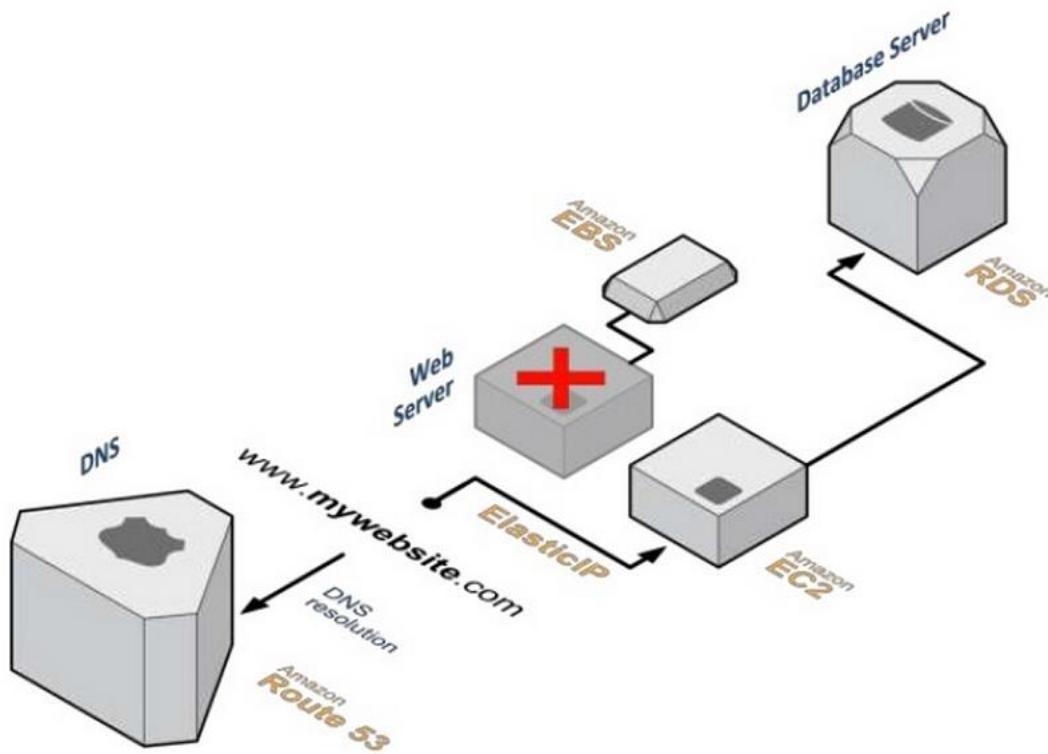


Rozwiązanie: zewnętrzne urządzenie blokowe

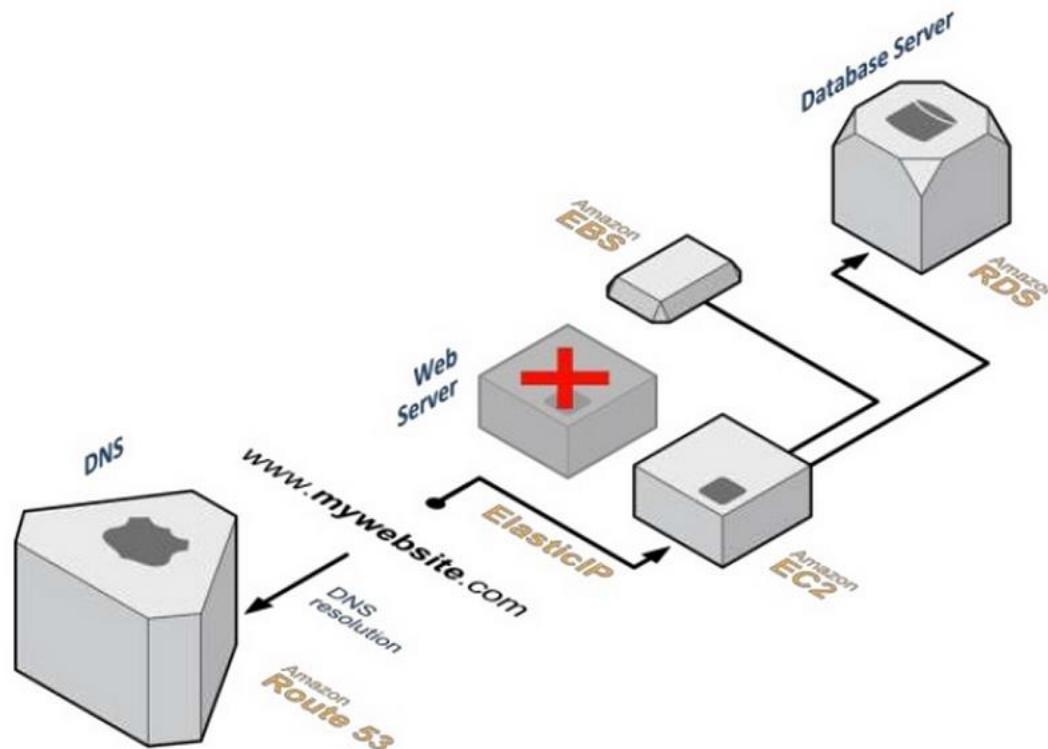
Awaria serwera



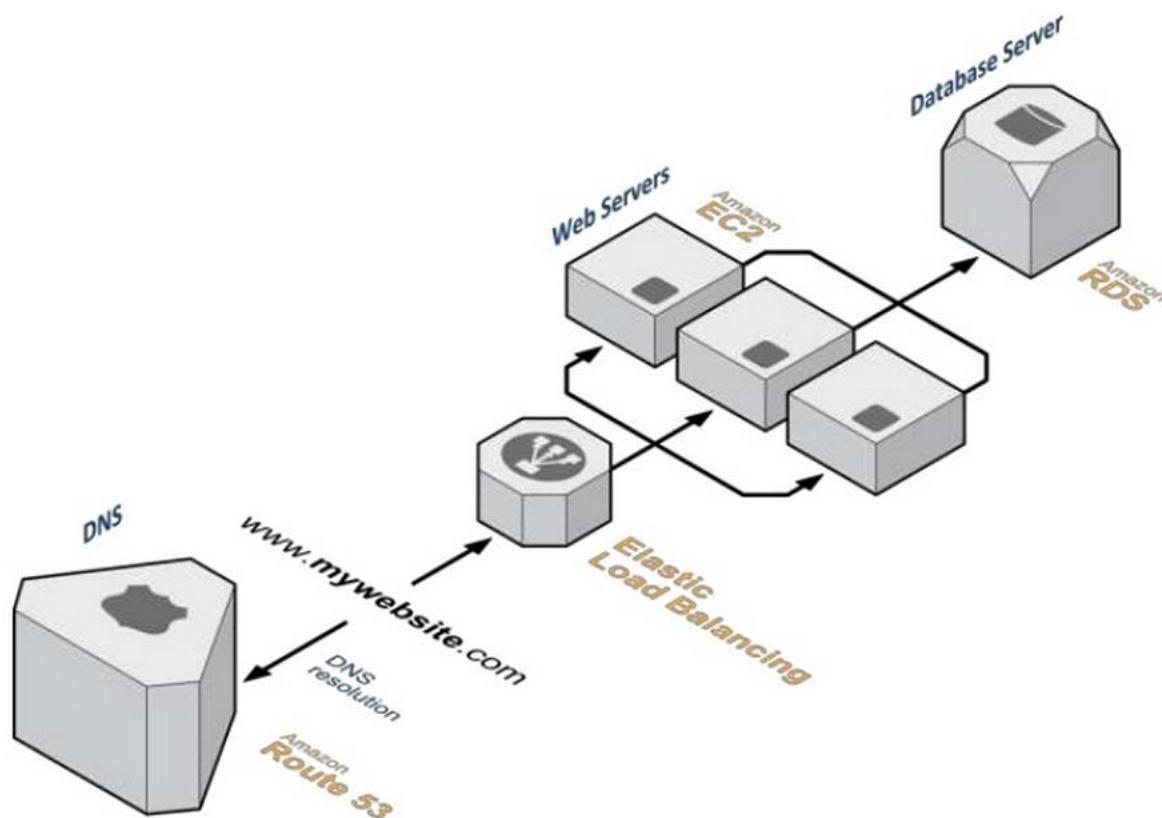
Zastąpienie serwera nowym



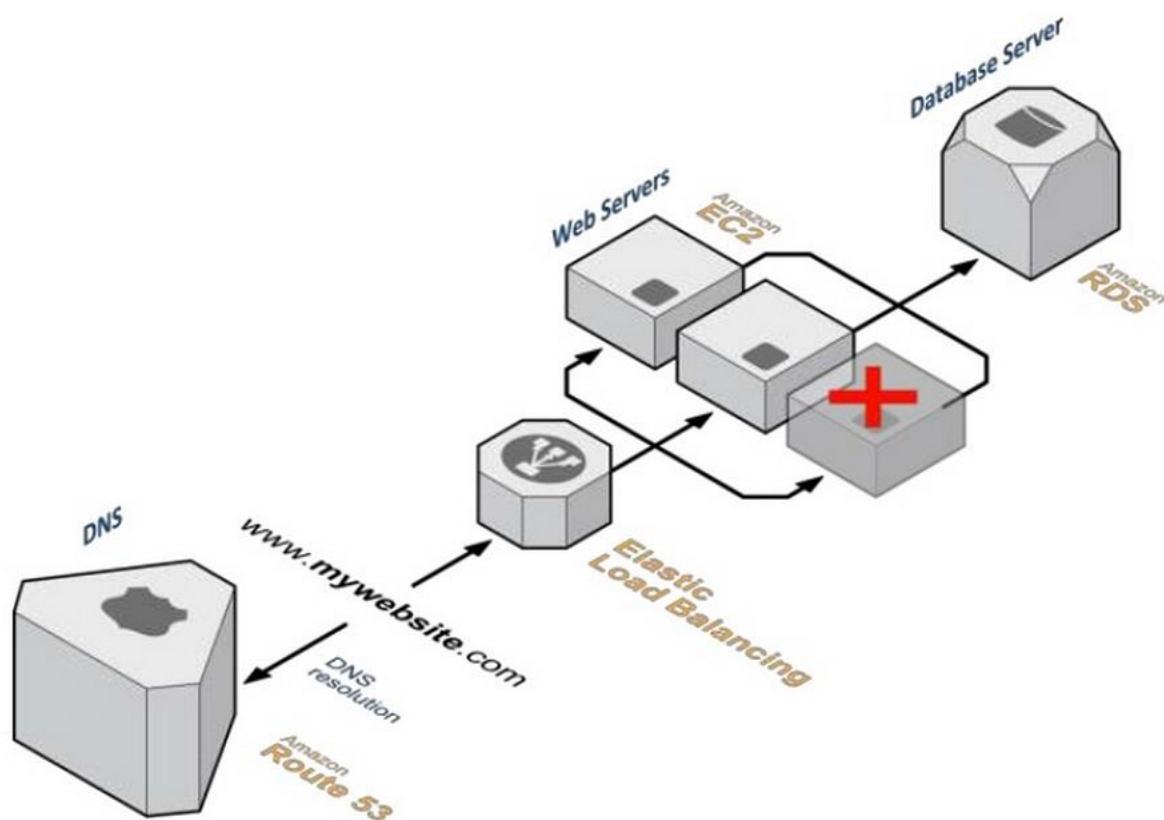
Przepięcie zasobu danych blokowych



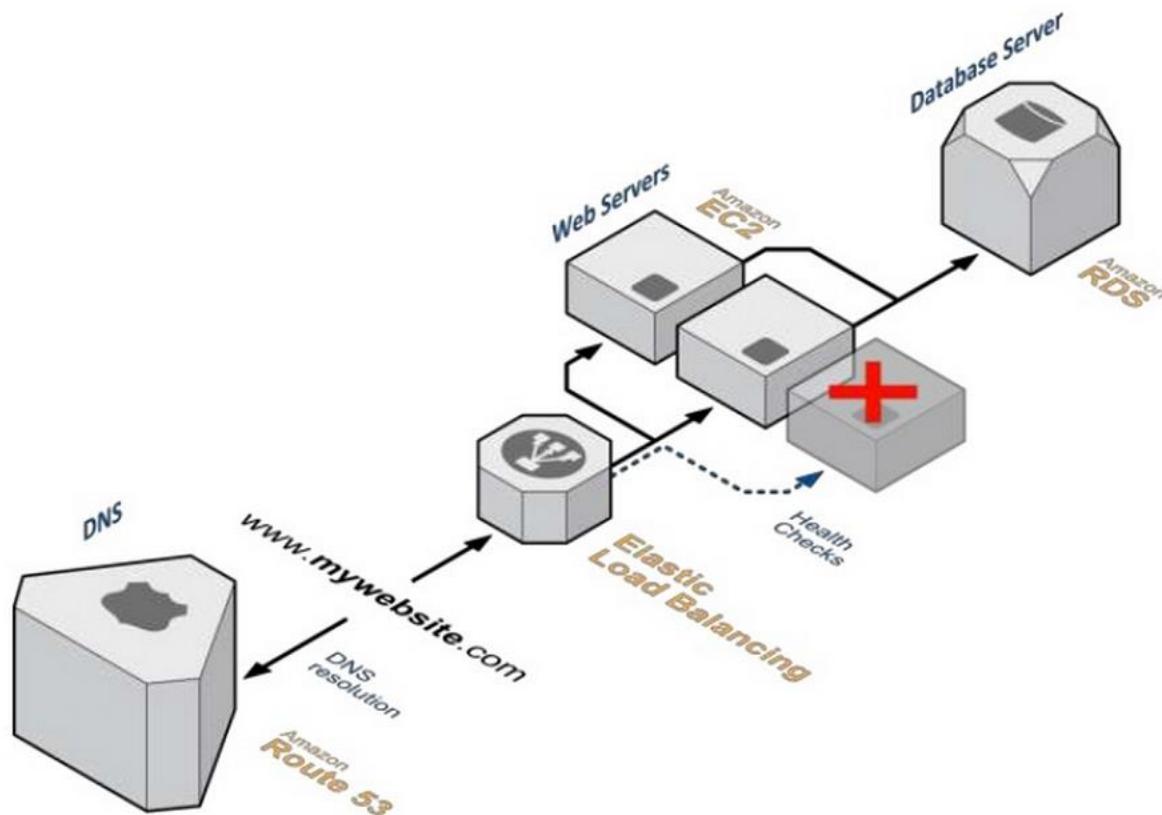
Problem: czas uruchomienia nowej instancji (~kilka minut)

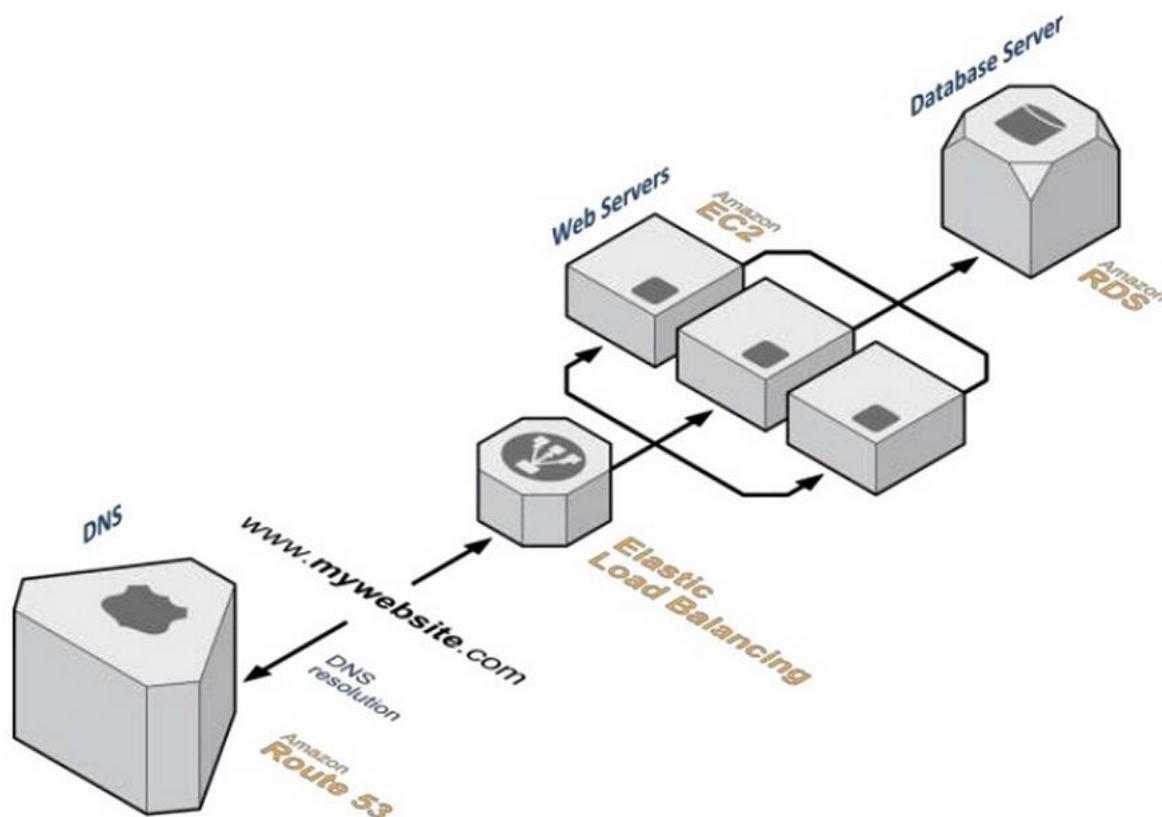


Rozwiązanie: wiele aktywnych instancji + load balancer

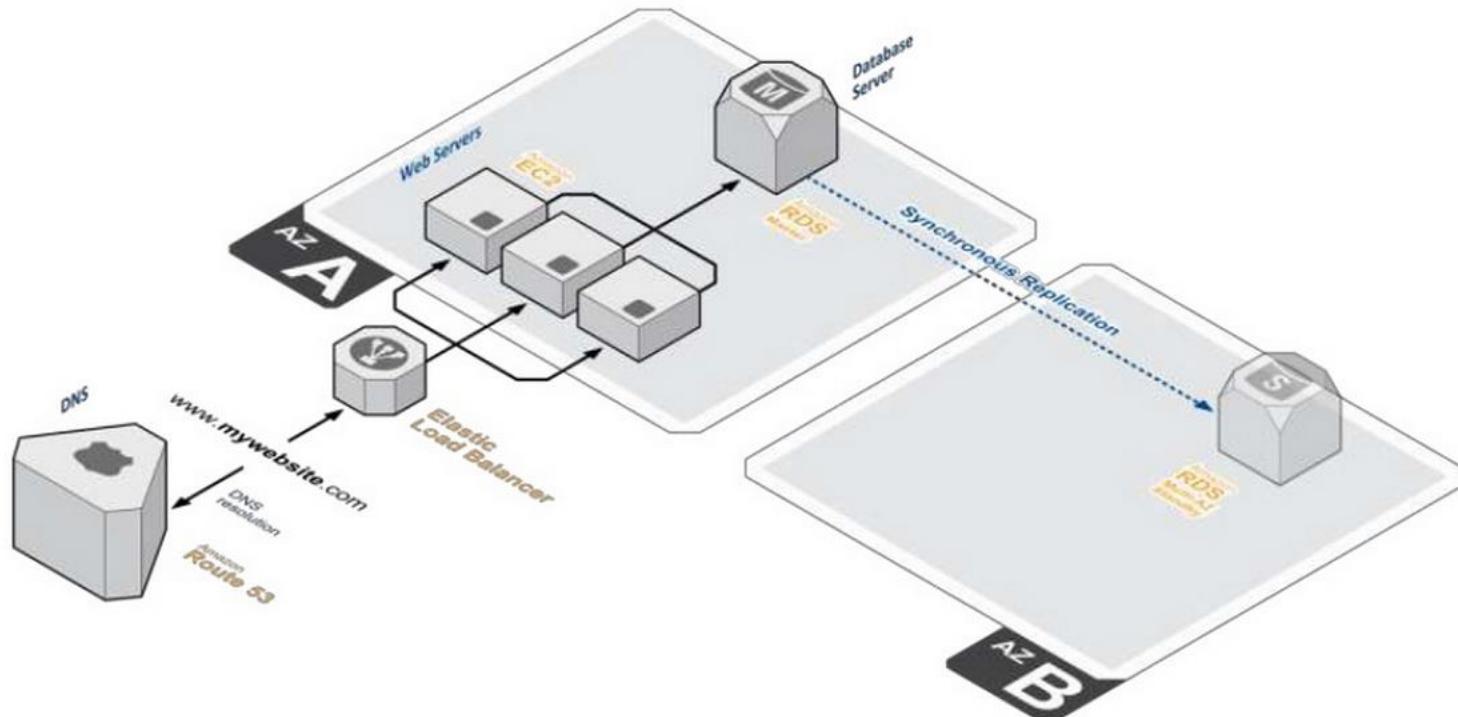


Wypięcie instancji z LB



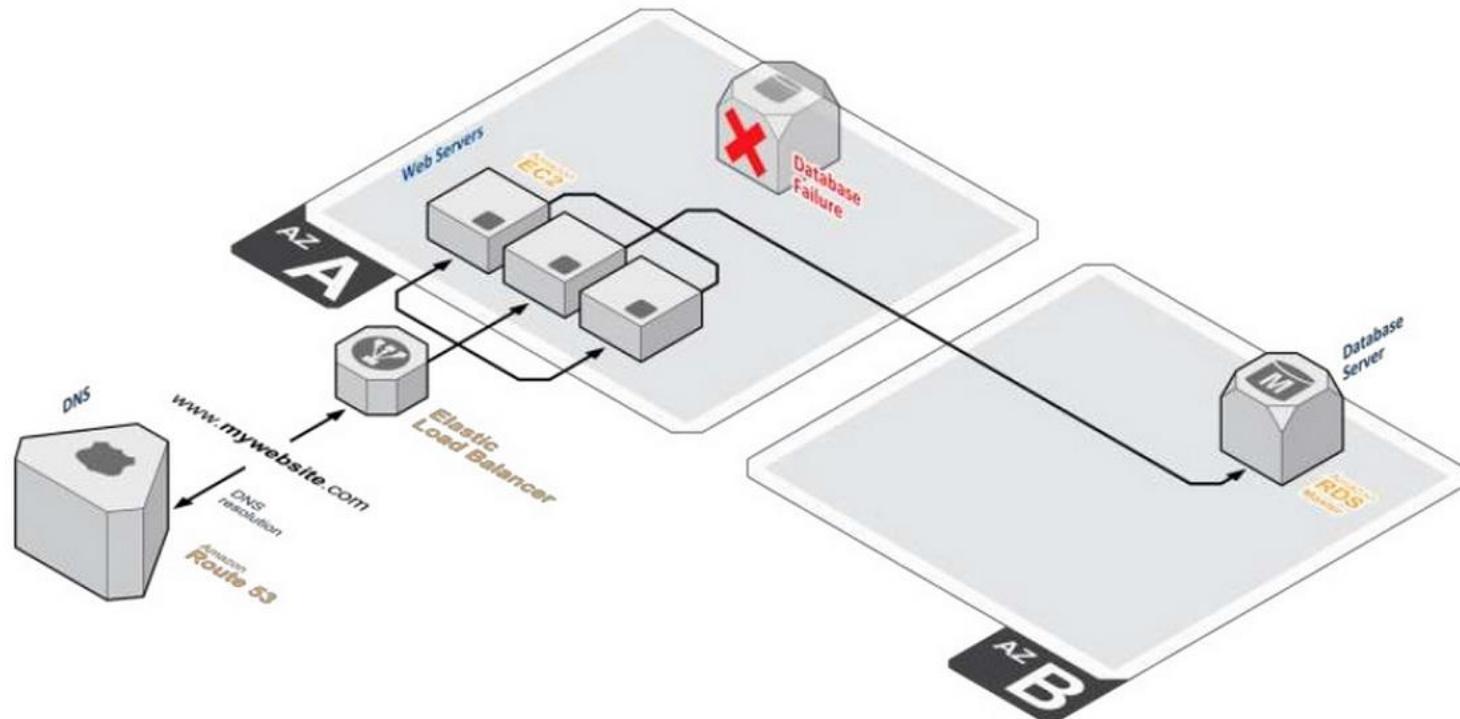


Problem: baza danych – Single Point of Failure

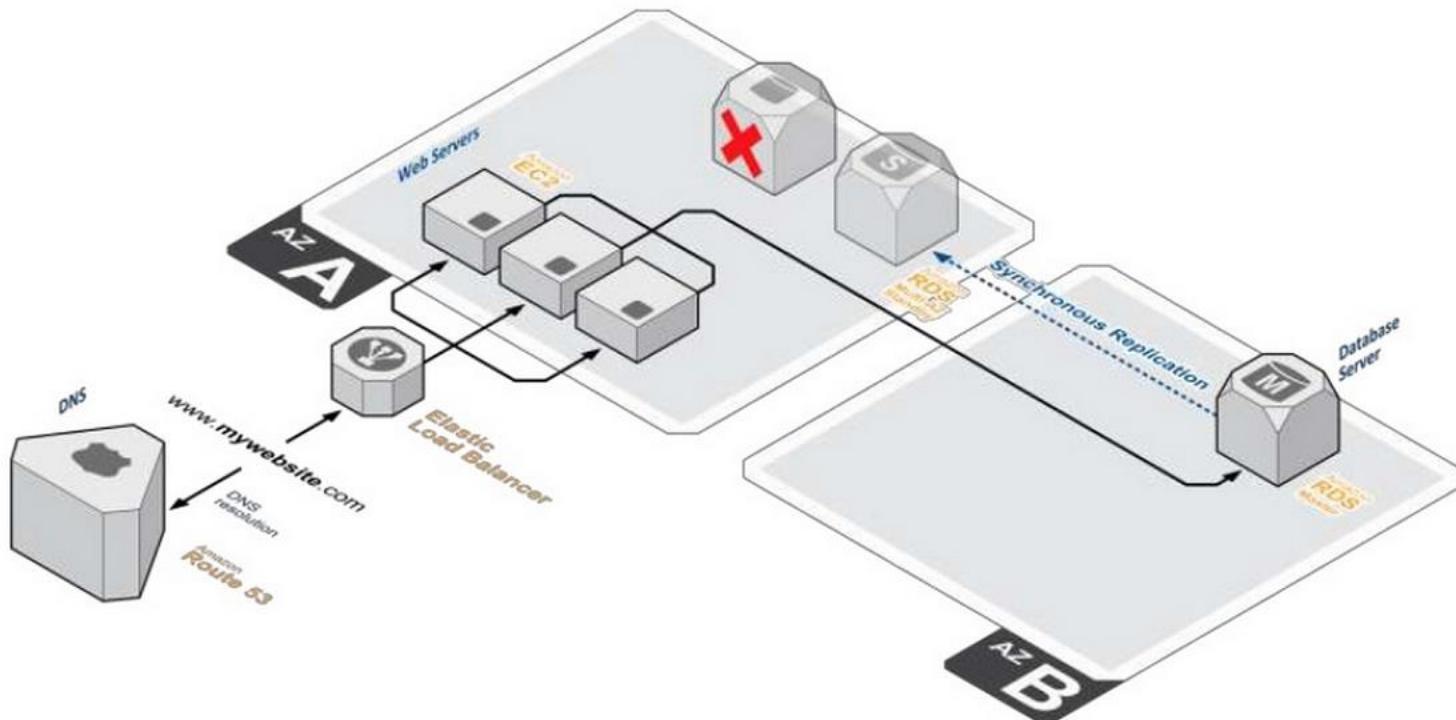


Rozwiązanie: replikacja danych master-slave (active-passive)

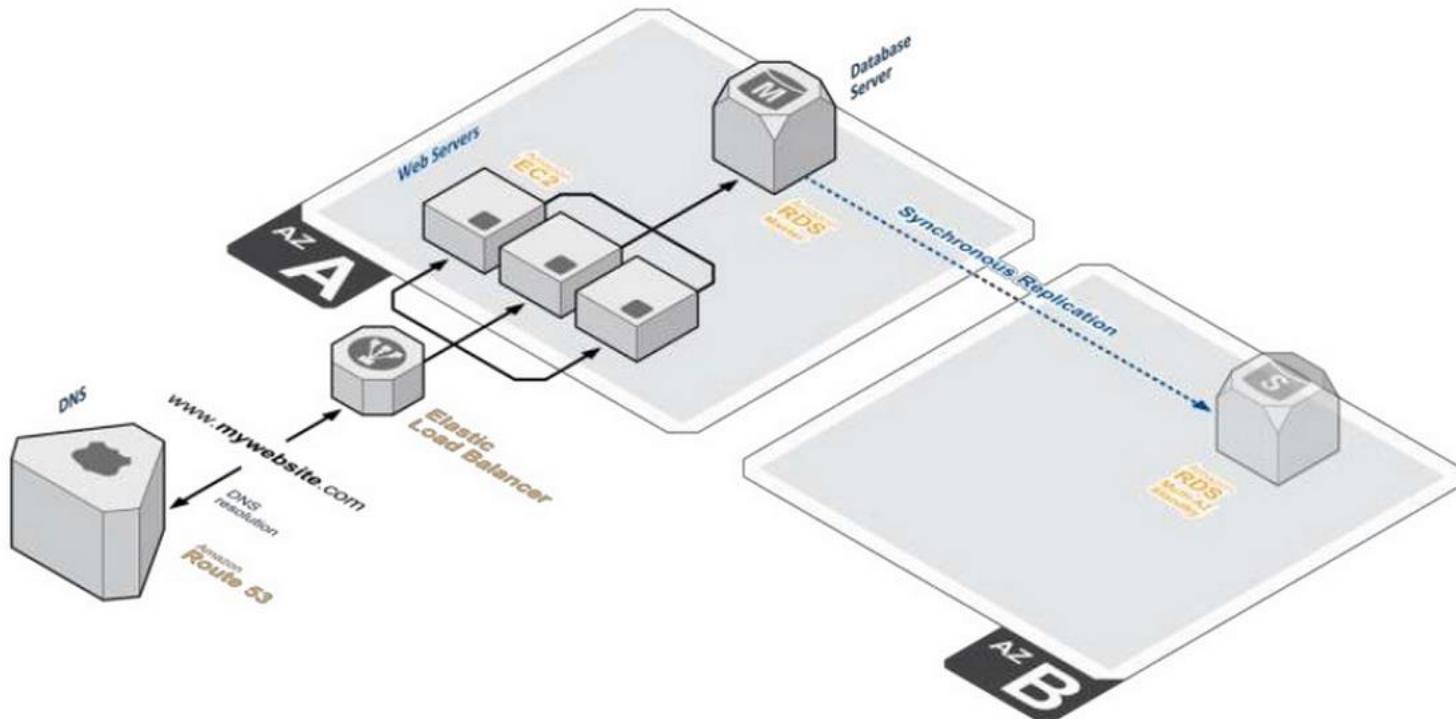
Awaria i przepięcie bazy



Zamiana master-slave

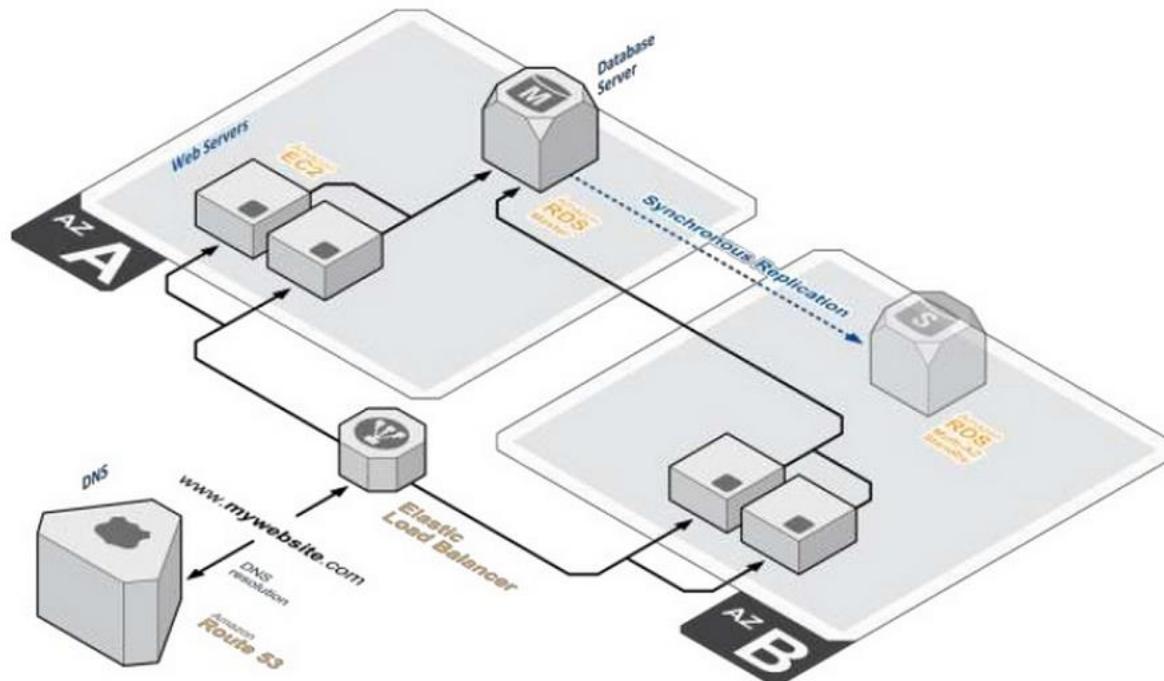


Baza danych Multi-AZ

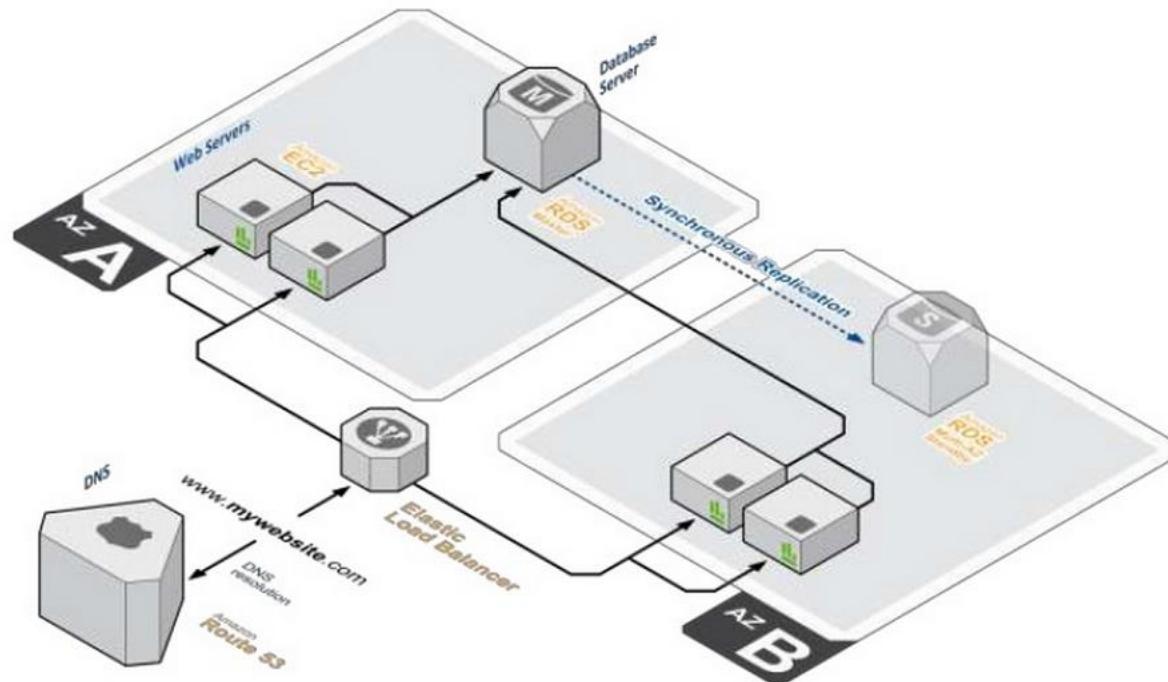


Problem: podatność na awarię całej strefy dostępności (AZ-A)

Instancje w wielu AZ

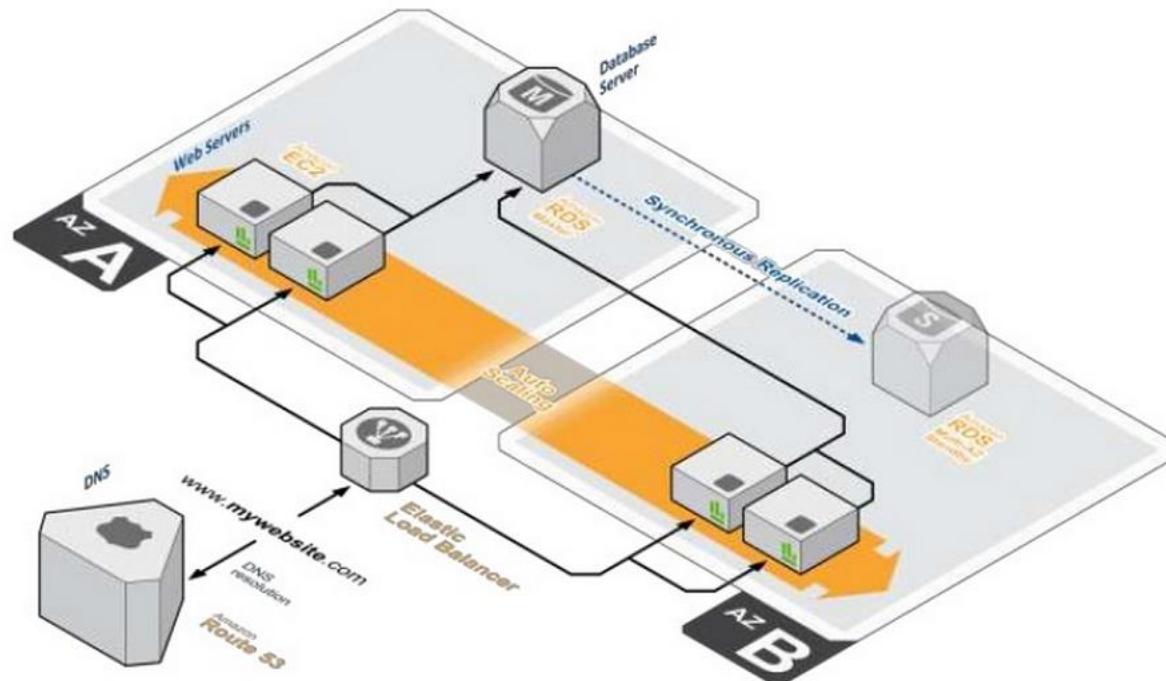


Rozwiązanie: redundancja serwerów w dwóch strefach dostępności



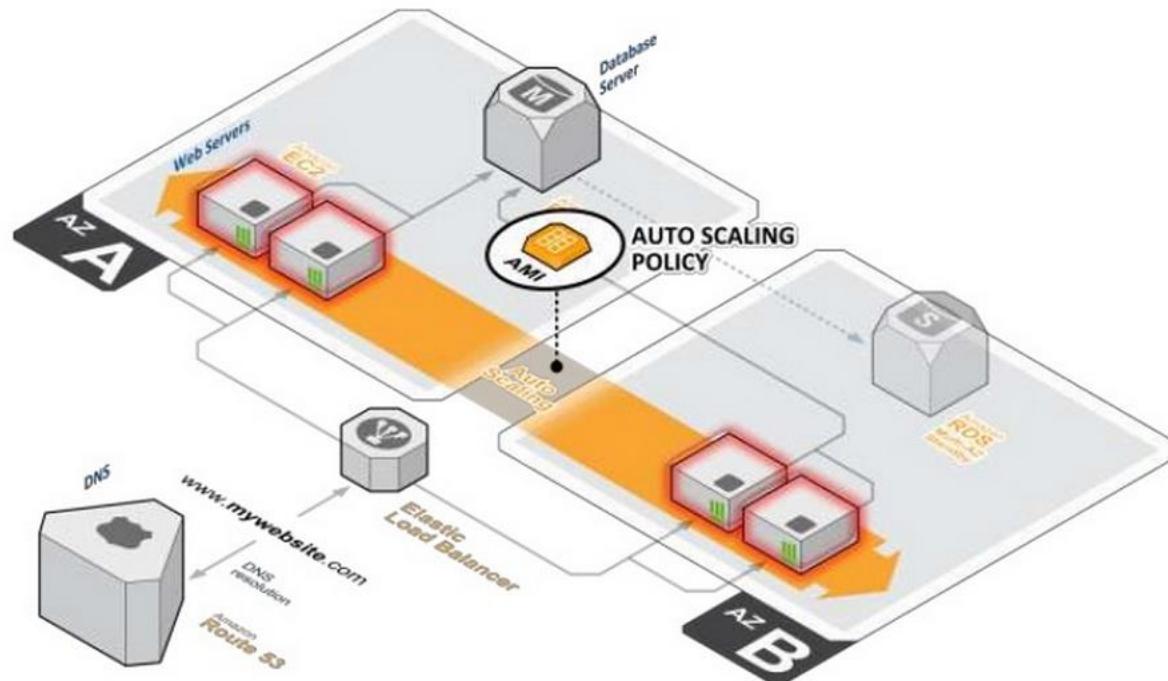
Problem: wysokie obciążenie serwerów

Automatyczne skalowanie

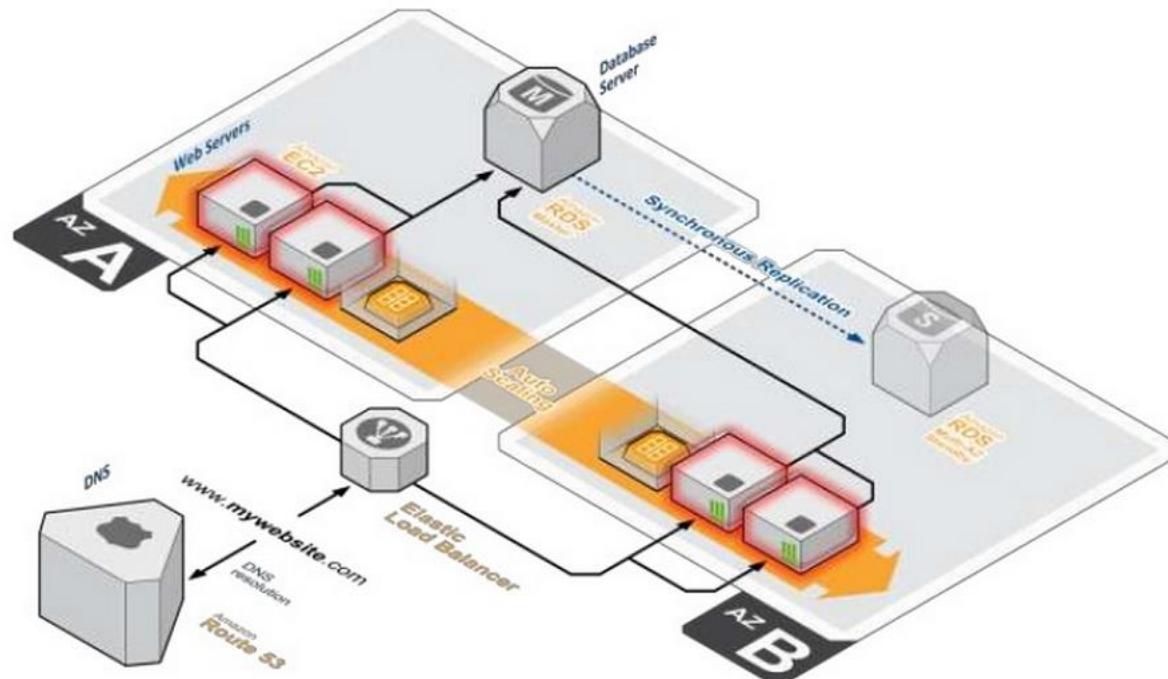


Rozwiązanie: automatyczne skalowanie

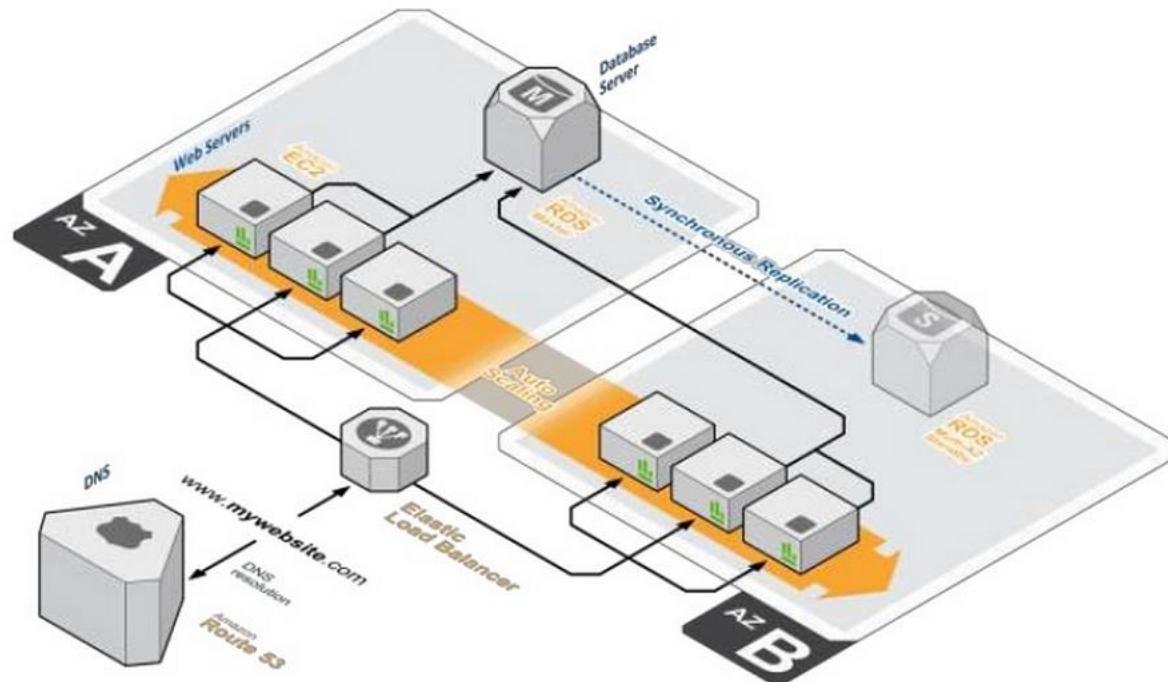
Scale out



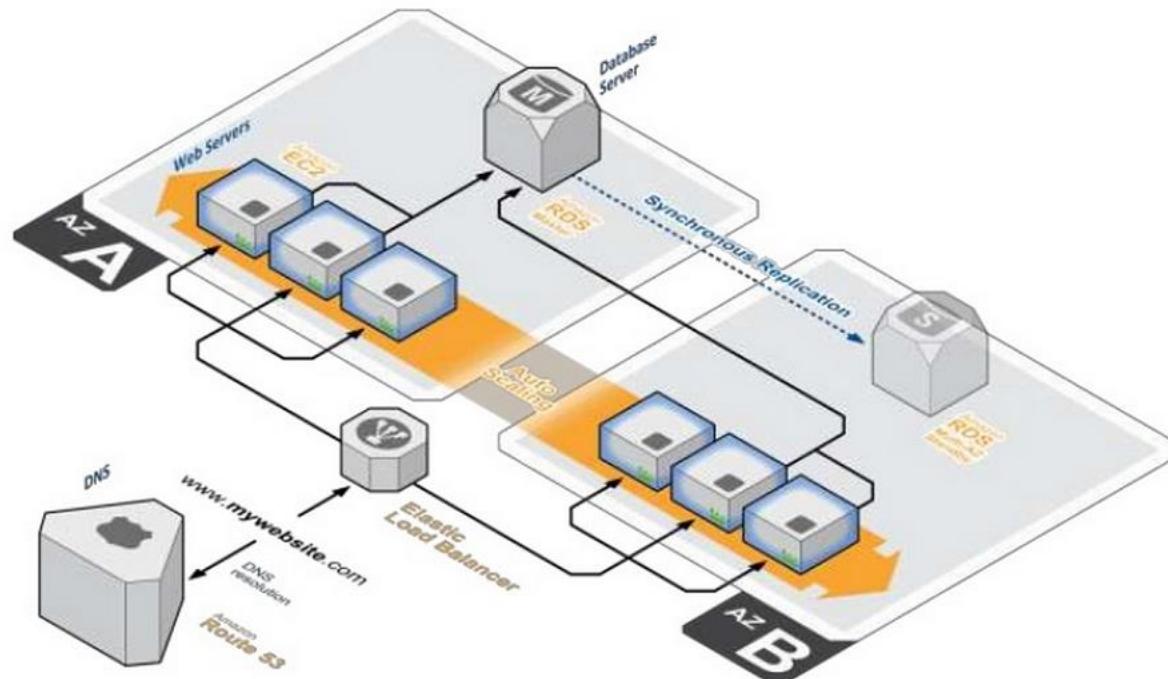
Scale out



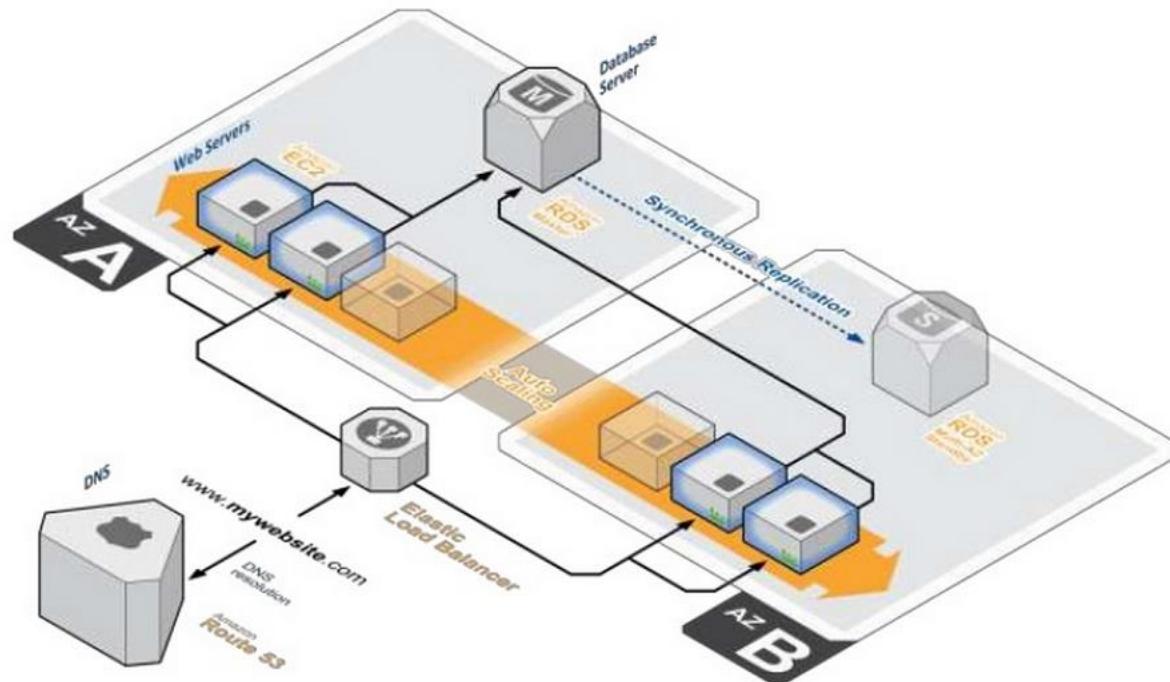
Scaled out

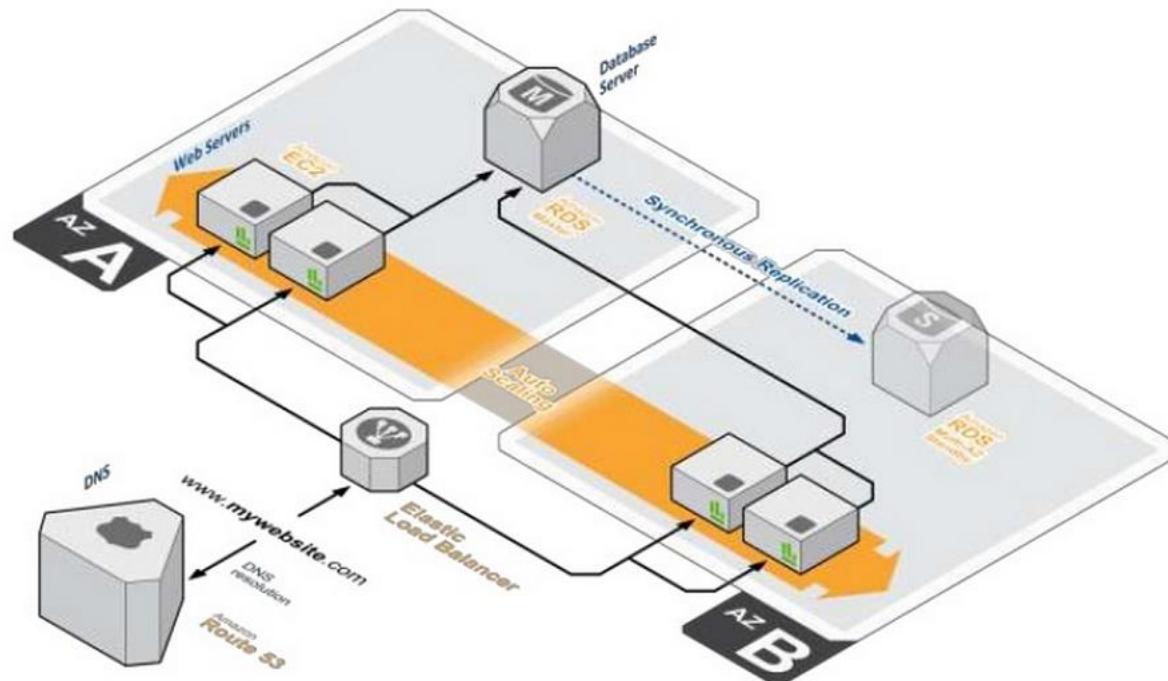


Cool down

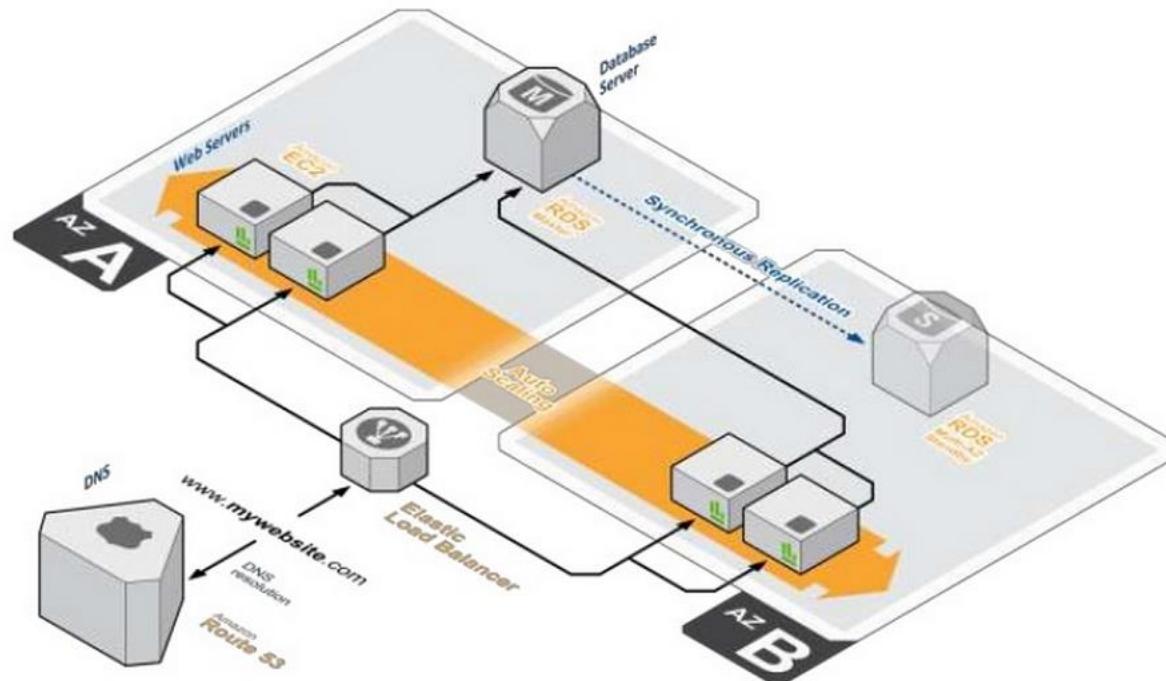


Scale in

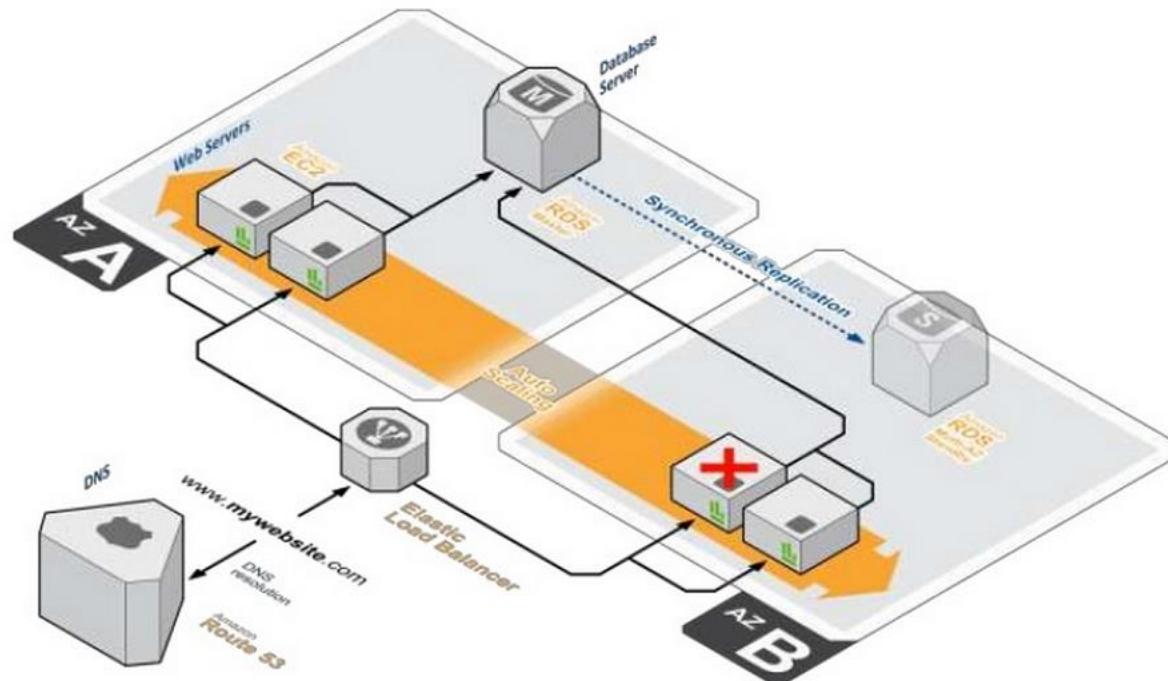




Stała ilość instancji

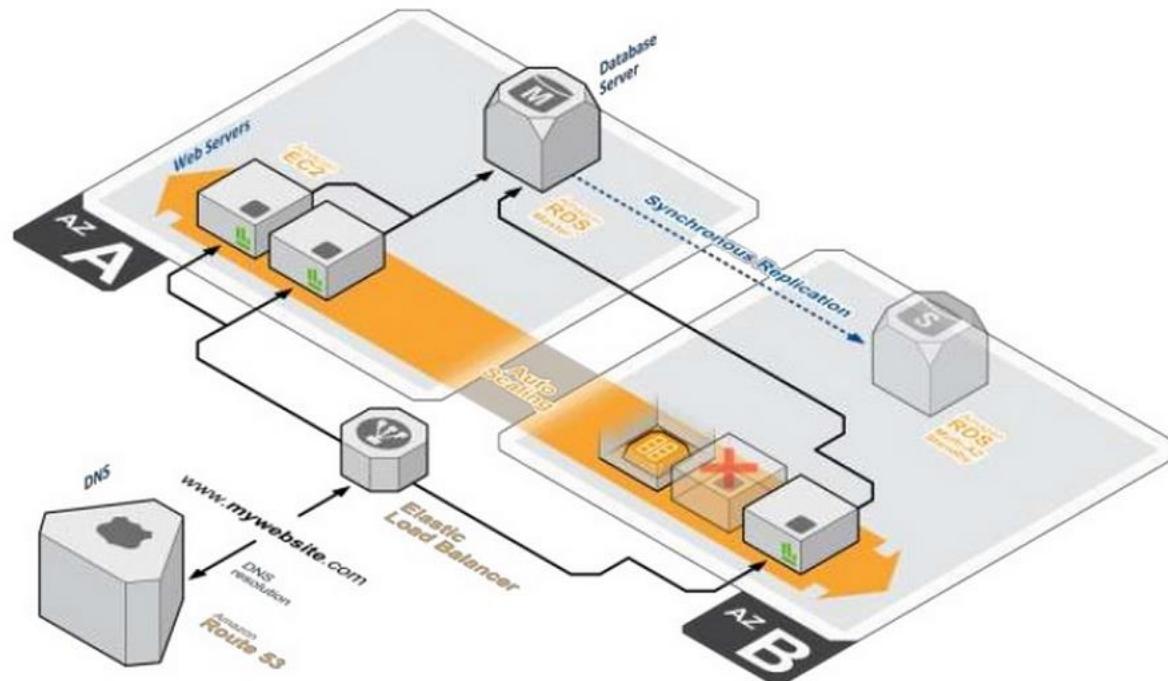


Problem: konieczność utrzymania stałej ilości instancji

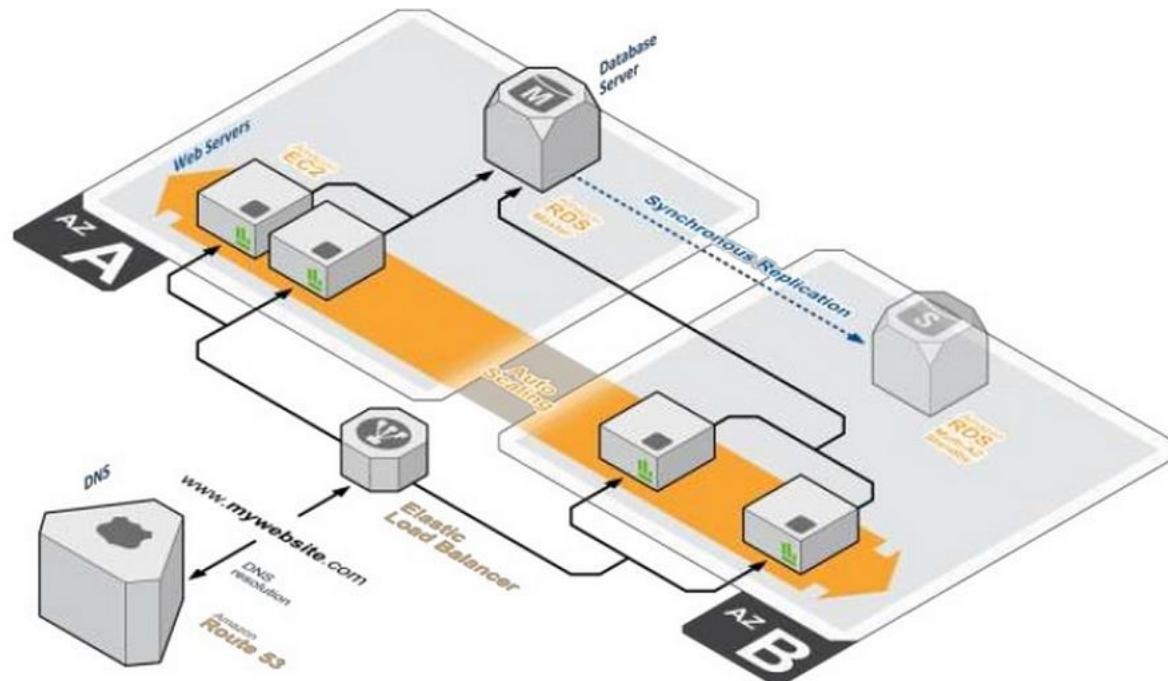


Rozwiążanie: automatyczne skalowanie

Zastąpienie instancji



Operacyjne działanie



Gdzie trzymać stan sesji HTTP w aplikacjach webowych?

Gdzie trzymać stan sesji HTTP w aplikacjach webowych?

Na serwerze

- LB wspierają „sticky sessions”
- Ale nie jest to HA
- Dynamika chmury sprawia, że jest to podejście nieakceptowalne

Na serwerach

- Synchronizowanie stanu sesji pomiędzy wieloma serwerami

W cache pamięciowym (*np. AWS ElastiCache*)

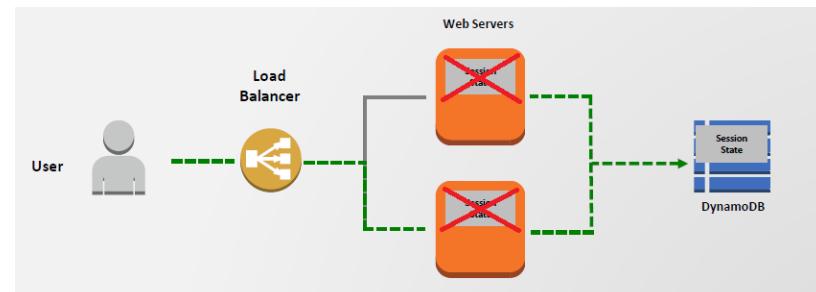
- Może być współdzielone międzyinstancjami (ale tylko w jednej AZ)

W bazie SQL (*np. AWS RDS*)

- Chociaż nie jest to naturalne użycie bazy danych

W bazie NoSQL (*np. AWS DynamoDB*)

- Najlepsza wydajność



Problem:

- konieczność zapewnienia szybkiego transferu danych od/do użytkowników na całym świecie

Problem:

- konieczność zapewnienia szybkiego transferu danych od/do użytkowników na całym świecie

Rozwiązanie: geo proximity

- dla danych statycznych użycie CDN (Content Delivery Network) np. *AWS CloudFront*
- dla danych dynamicznych rozproszenie geograficzne infrastruktury i użycie DNS z opcją LBR (Latency Based Routing) np. *AWS Route53*

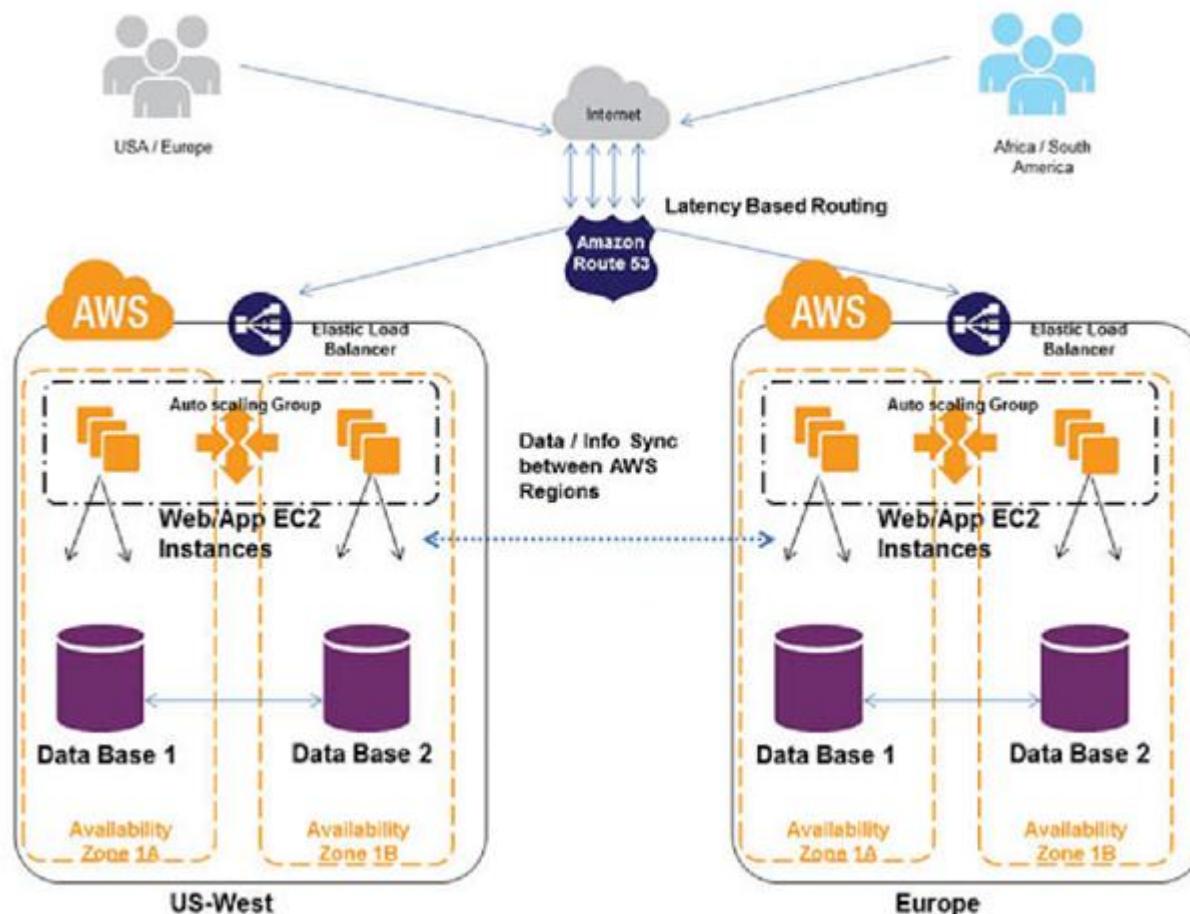
AWS CloudFront edge locations



- - edge locations (CDN)



DNS Route53 LBR

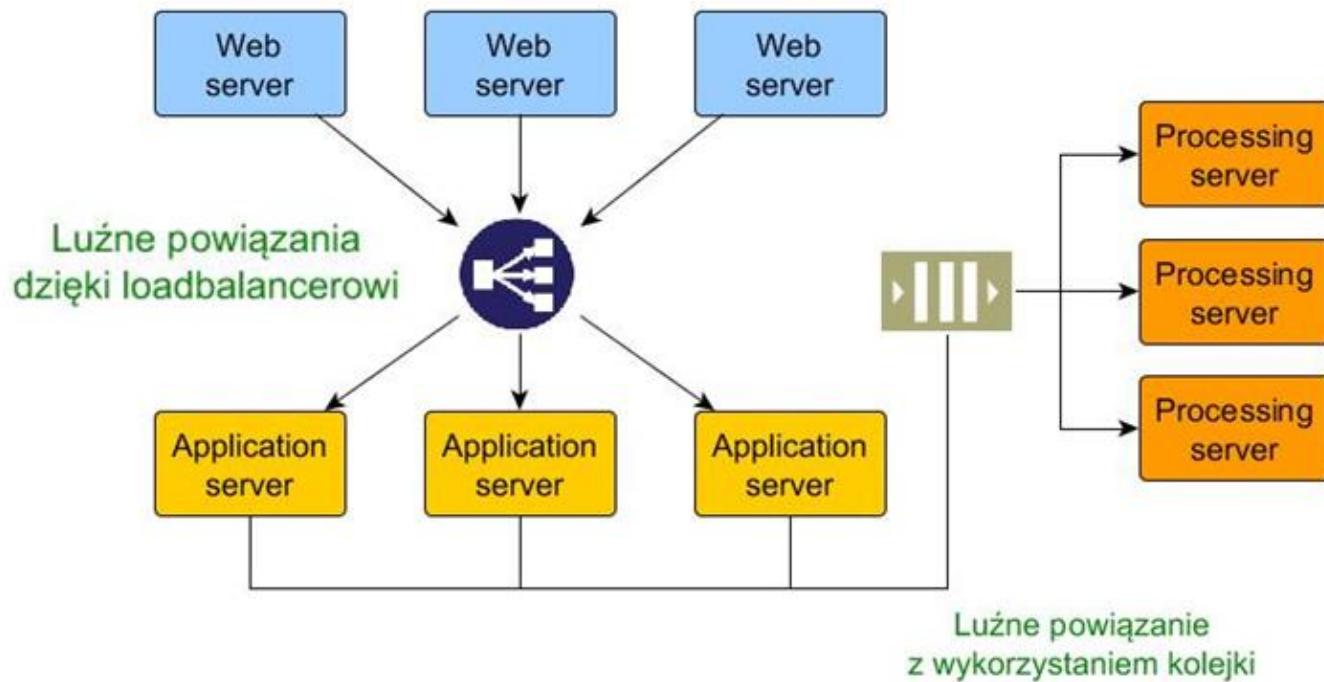


” Przetwarzanie asynchroniczne i wsadowe

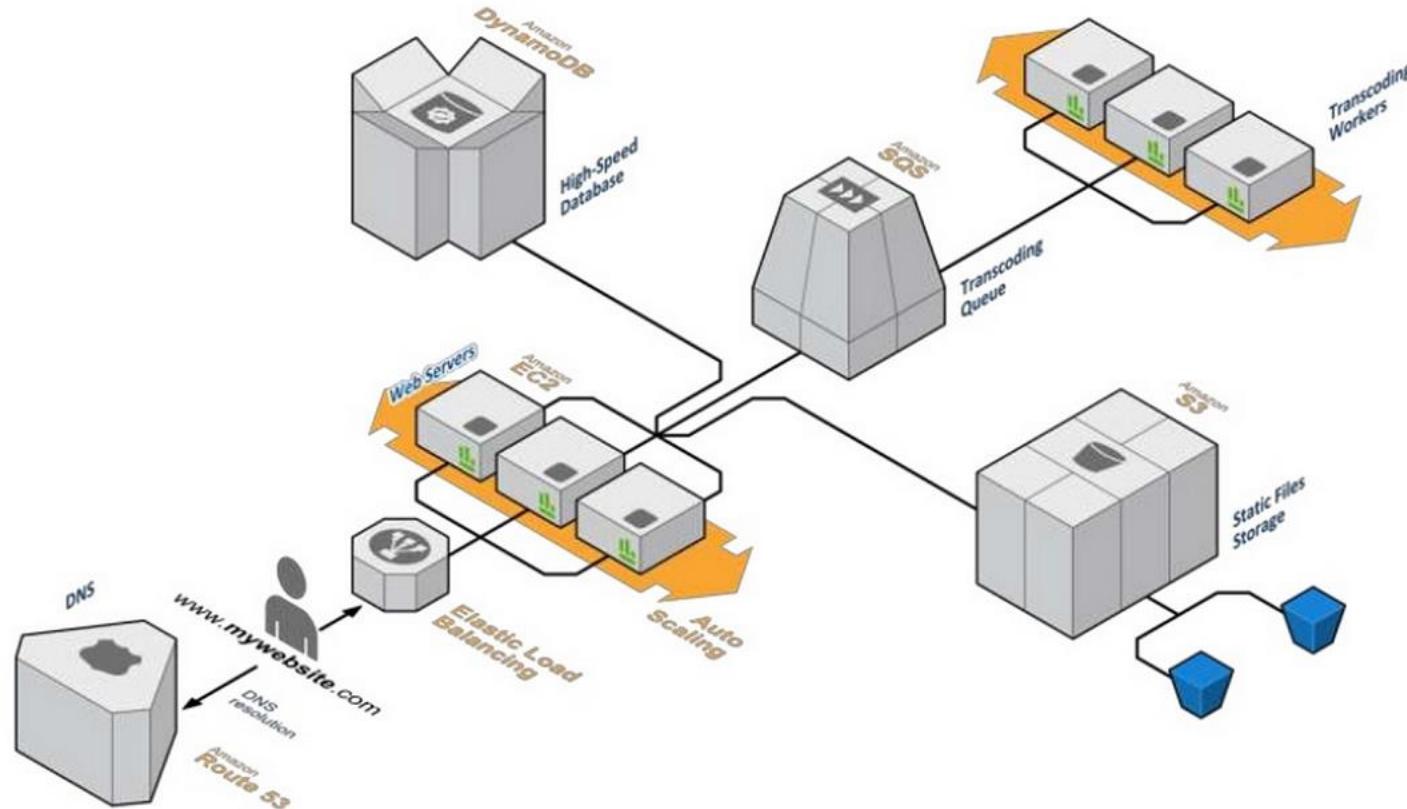
Backend, batch processing, asynchronous messages



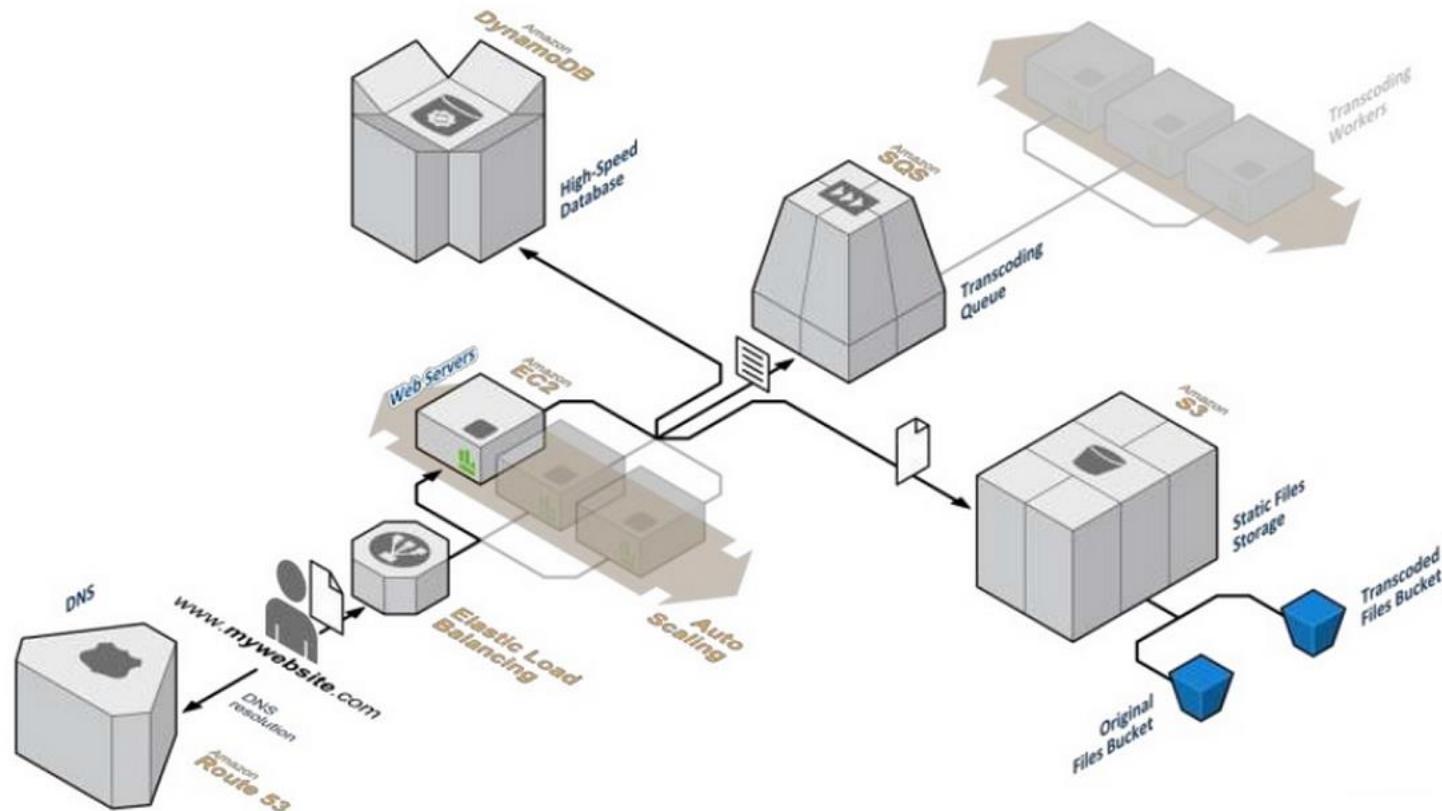
Preferuj luźne powiązania (loose coupling)



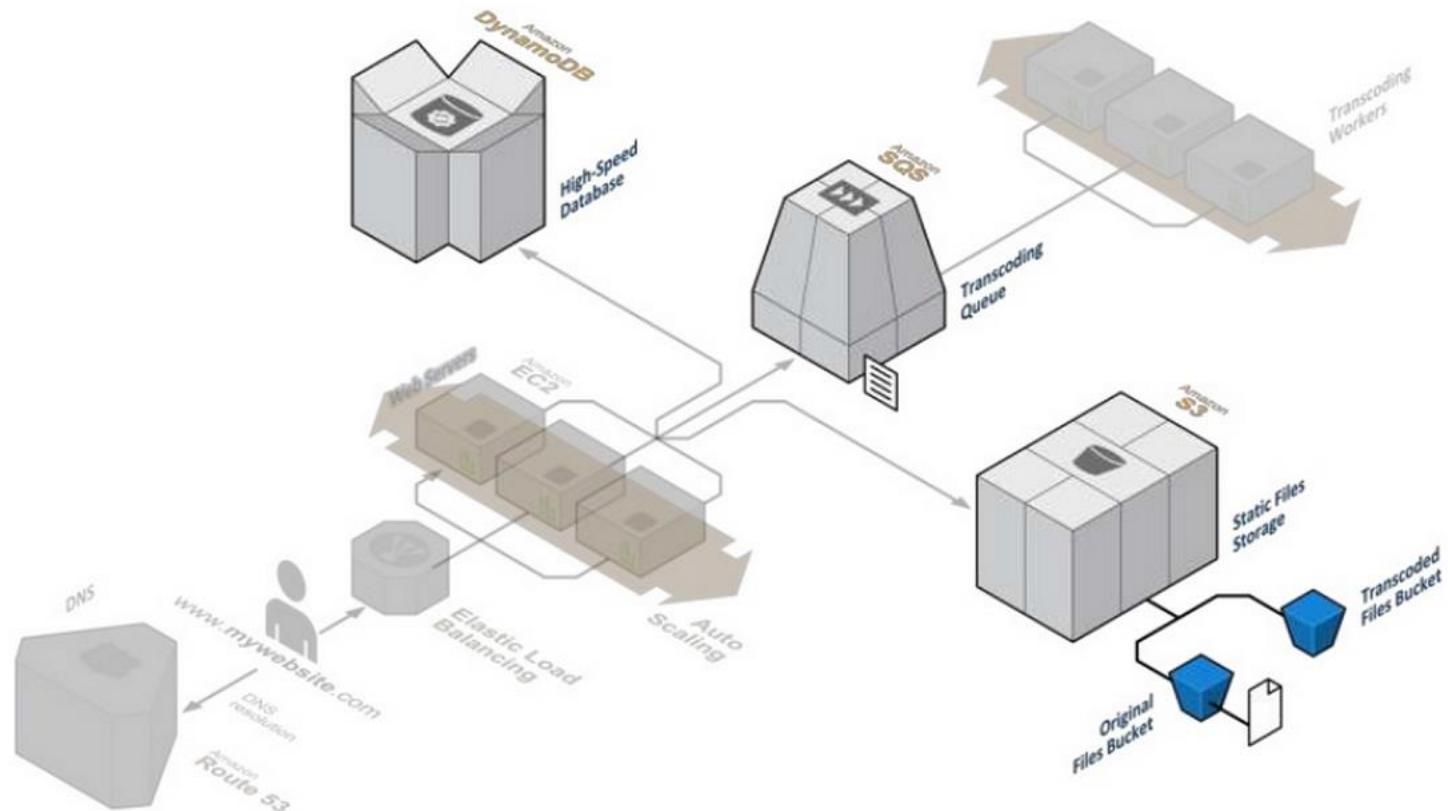
Przetwarzanie asynchroniczne



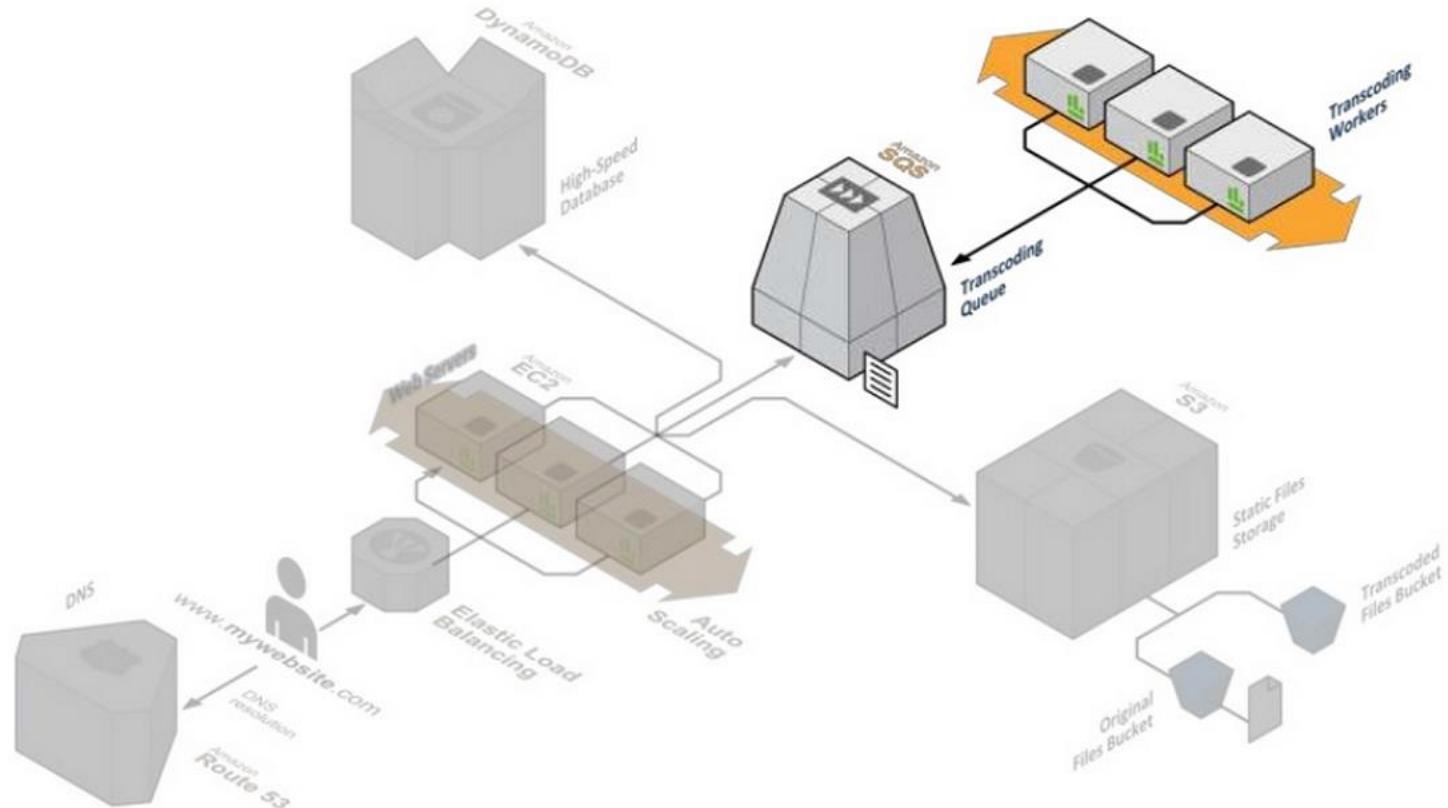
Przetwarzanie asynchroniczne



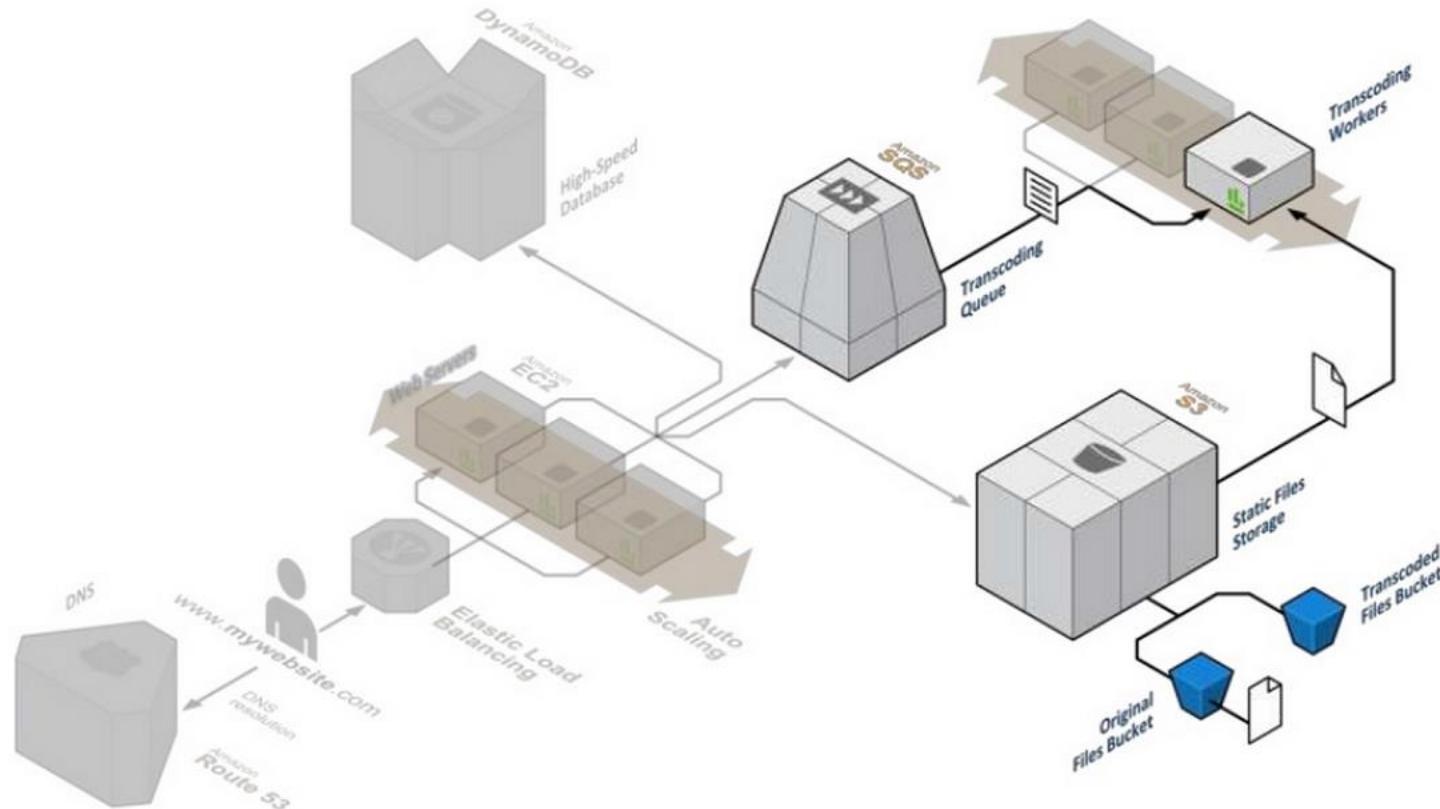
Przetwarzanie asynchroniczne



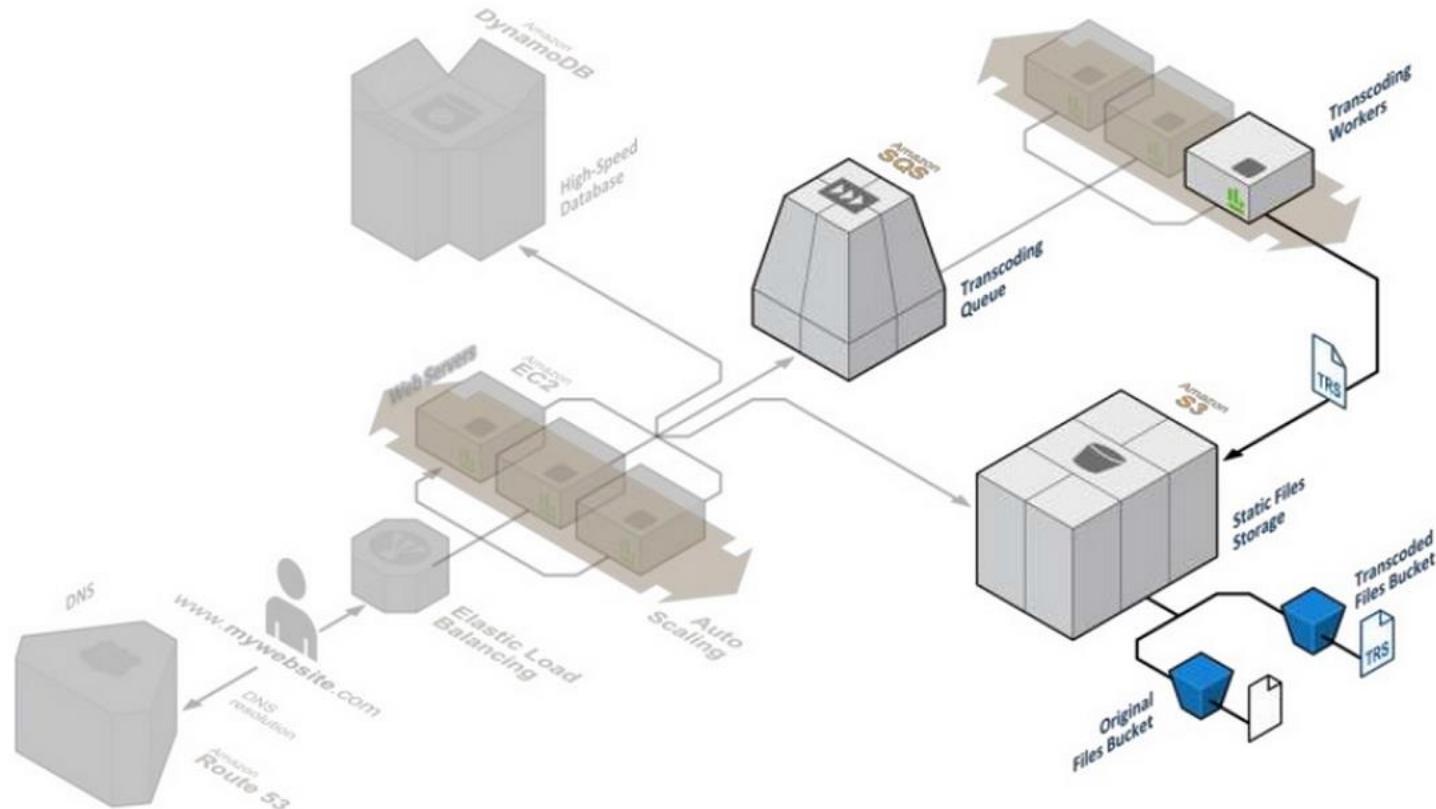
Przetwarzanie asynchroniczne

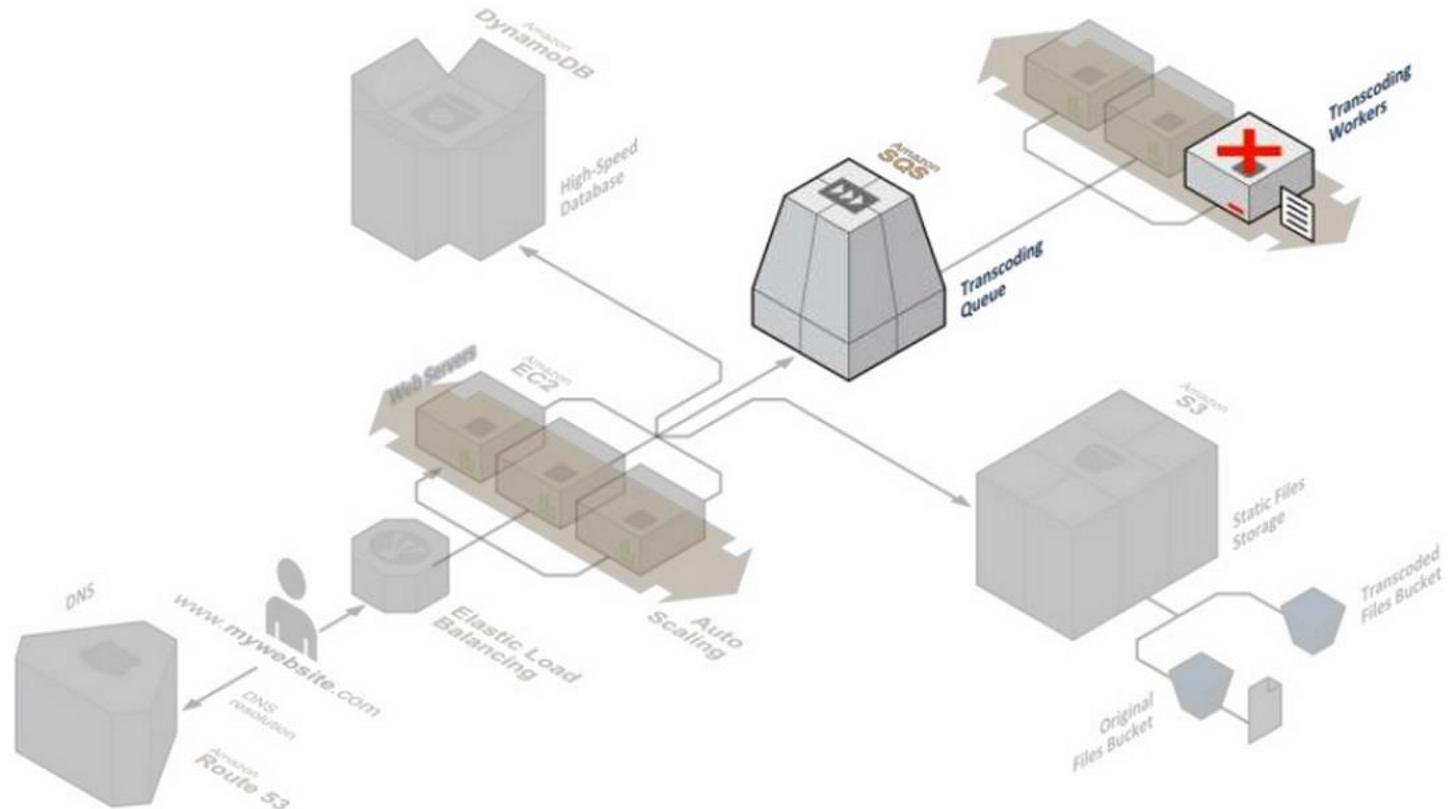


Przetwarzanie asynchroniczne



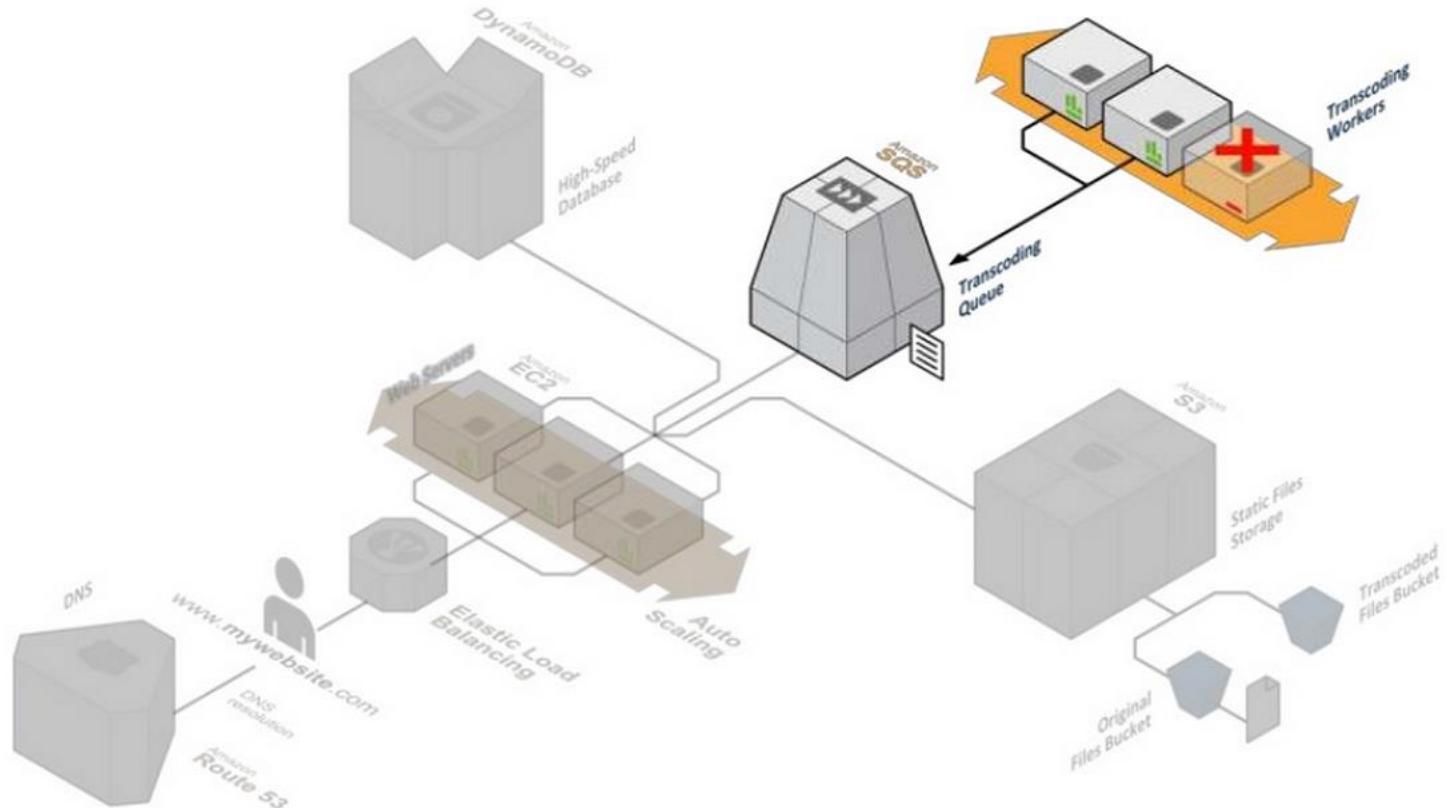
Przetwarzanie asynchroniczne





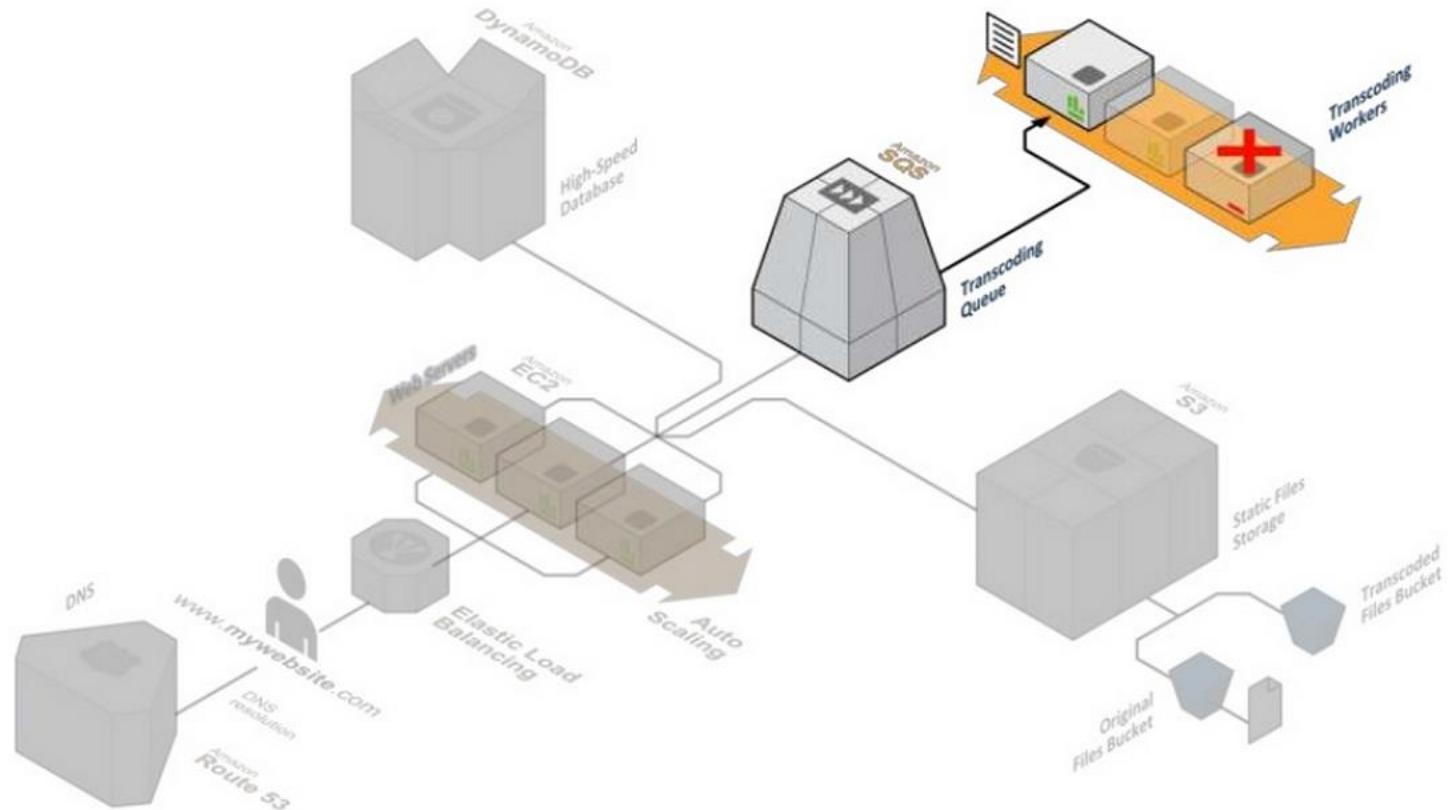
Problem: awaria serwera który pobrał z kolejki komunikat do przetworzenia

Ponowienie komunikatu

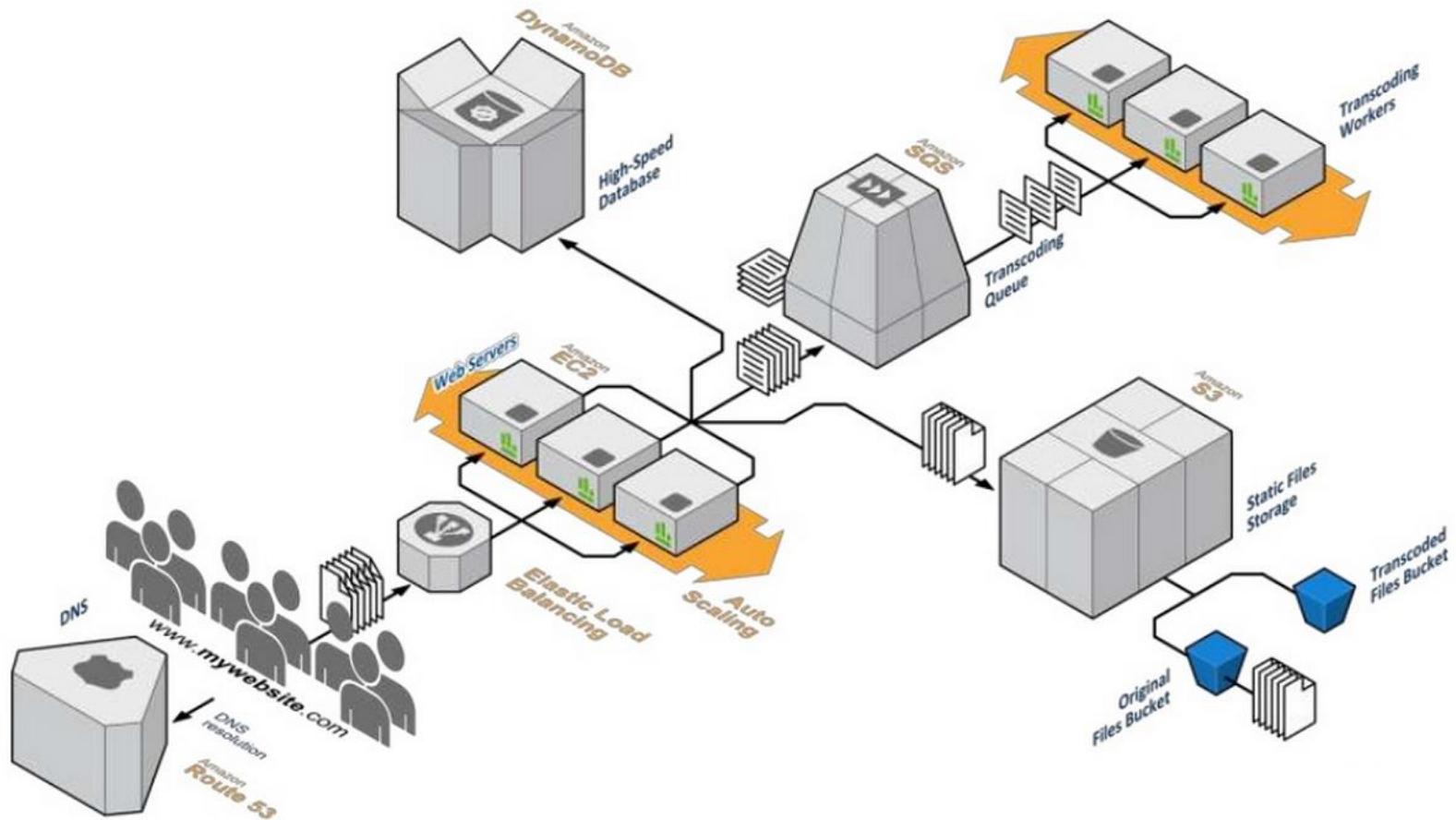


Rozwiążanie: visibility timeout, ponowienie (retry)

Przetworzenie ponowionego komunikatu

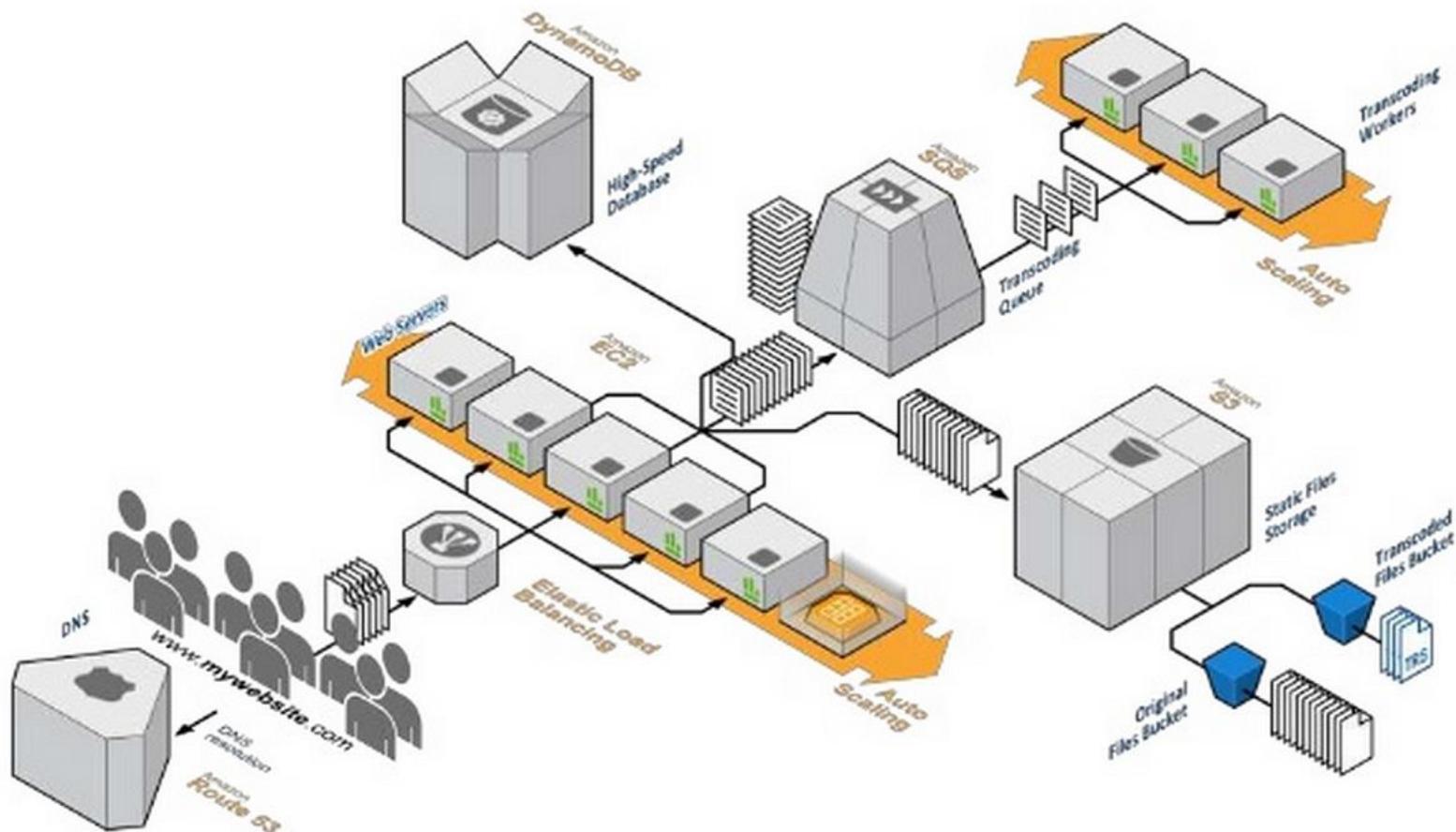


Wzrost obciążenia



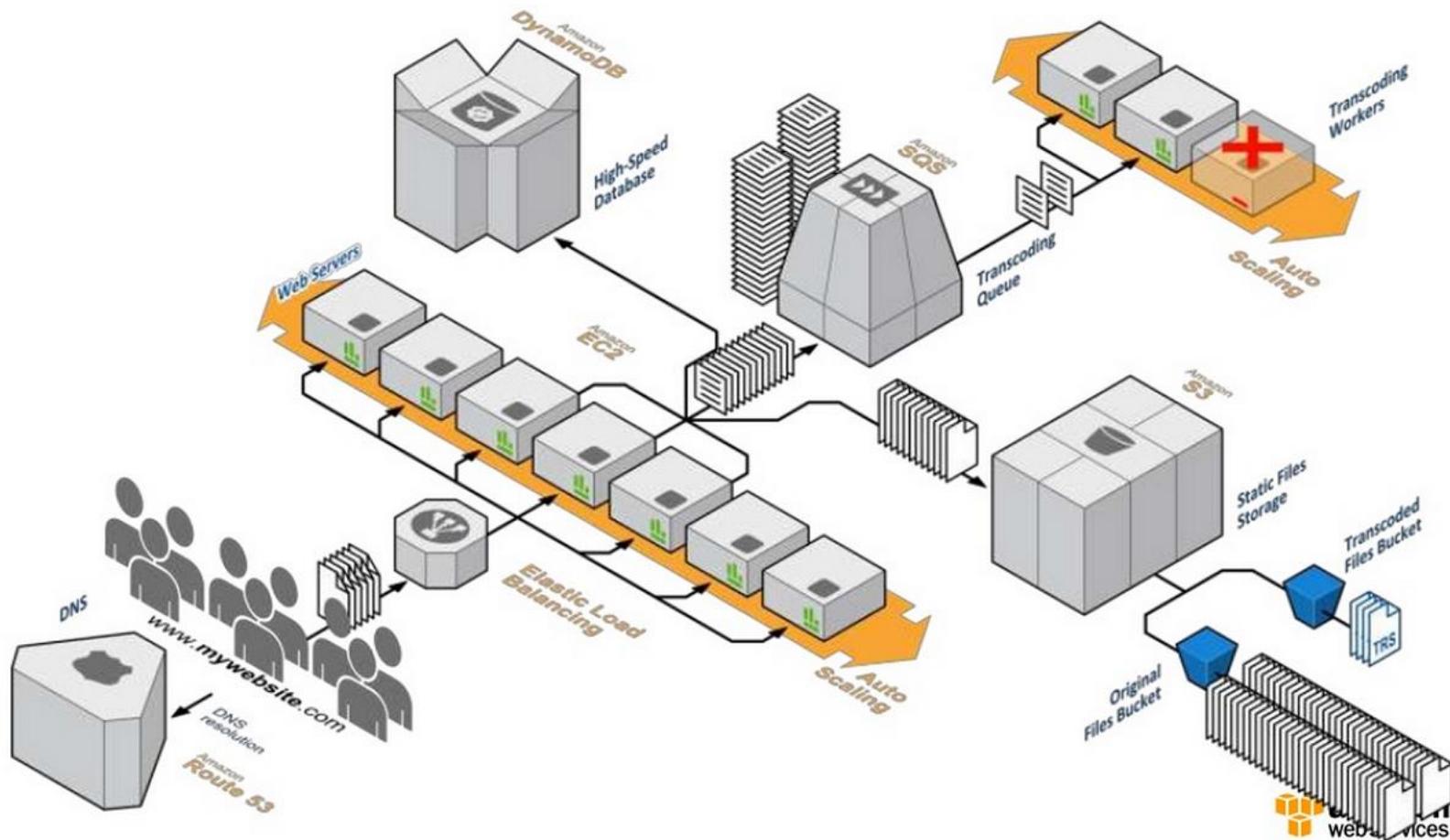
Problem: chwilowy wzrost obciążenia

Kolejkowanie komunikatów

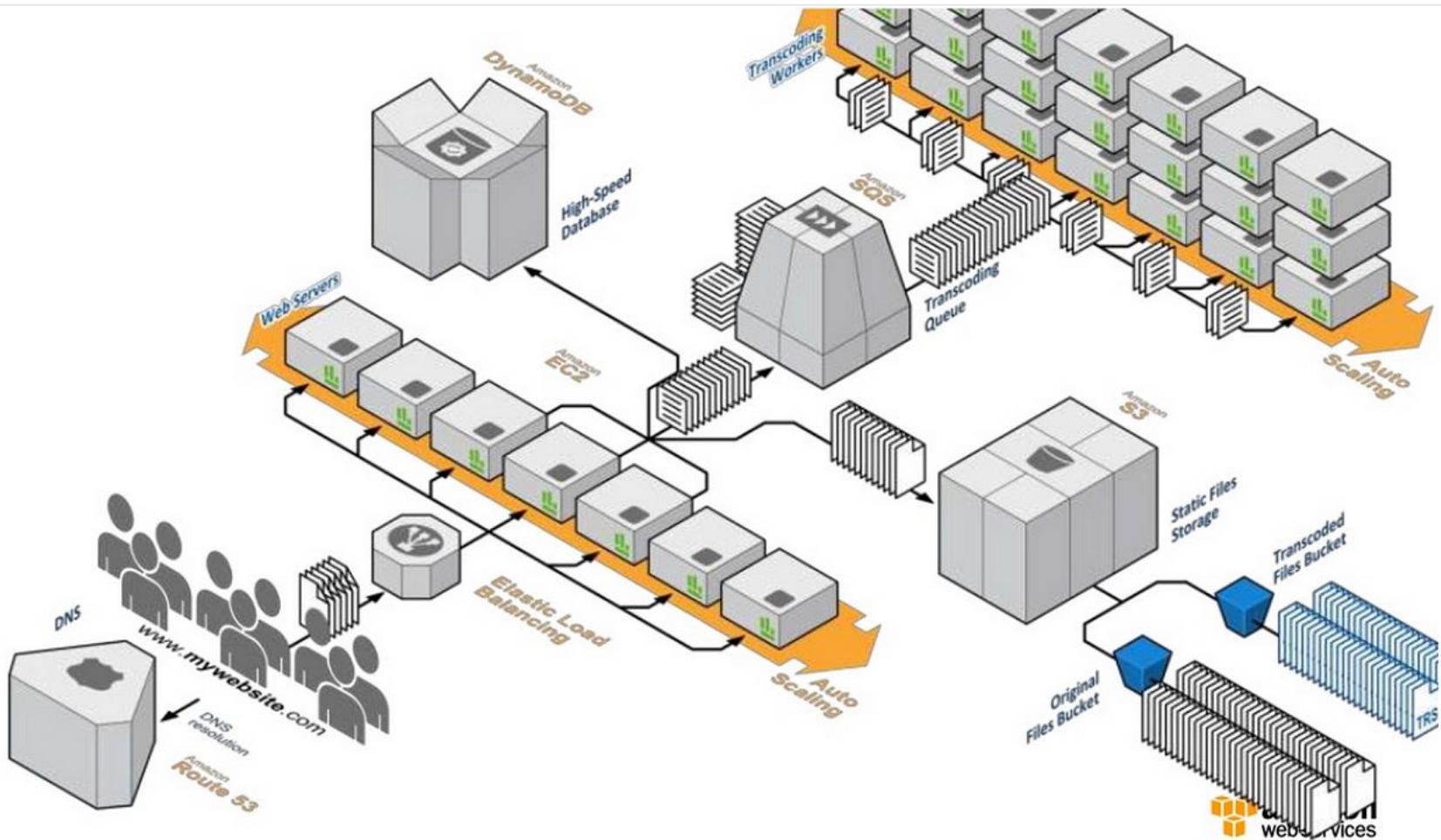


Rozwiązanie: kolejkowanie komunikatów

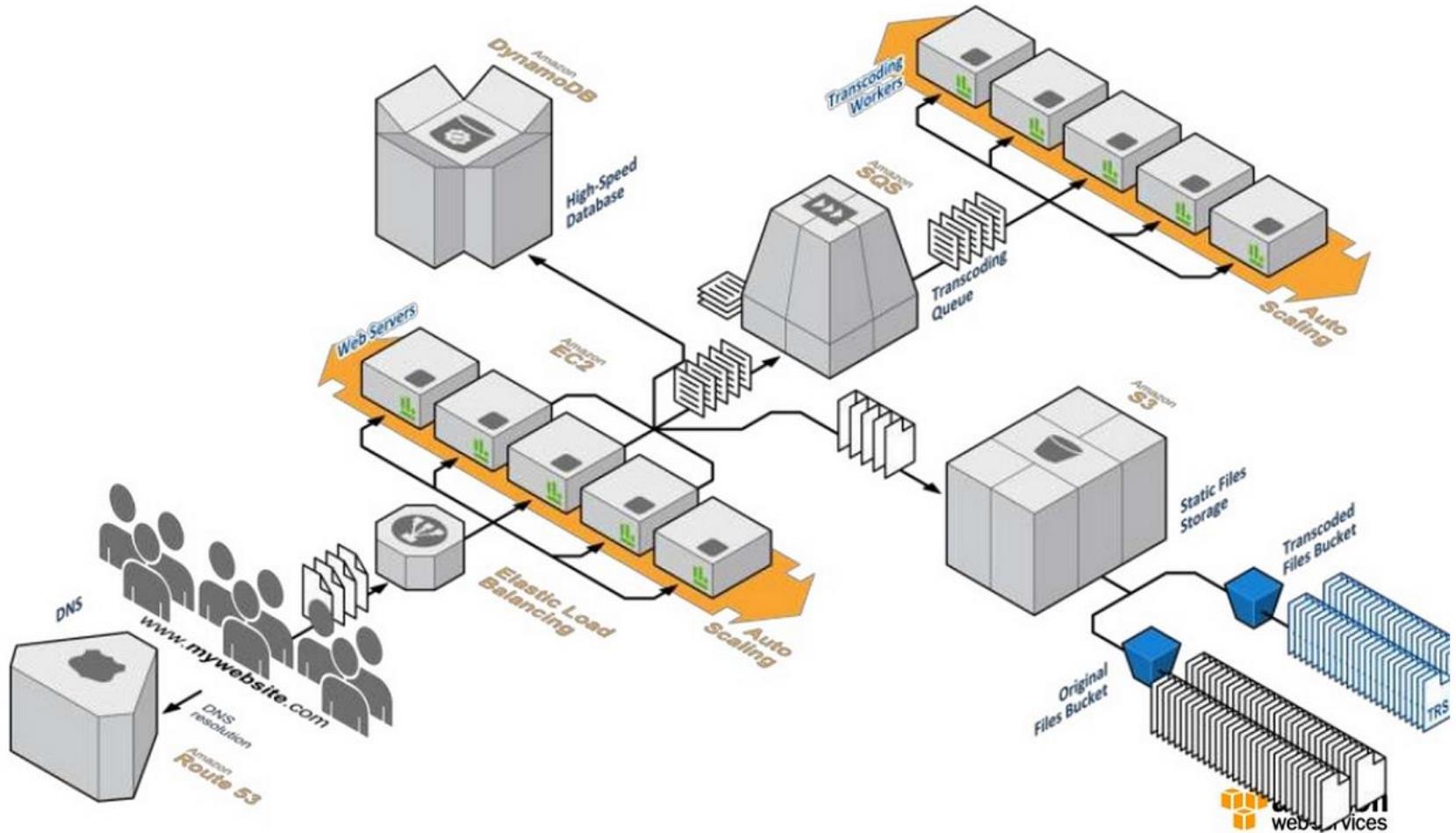
Kolejkowanie komunikatów



Wyskalowanie infrastruktury (scale out)



Scale in



” Odtwarzanie po awarii

ang. disaster recovery

11 Disaster recovery (odtwarzanie awaryjne)

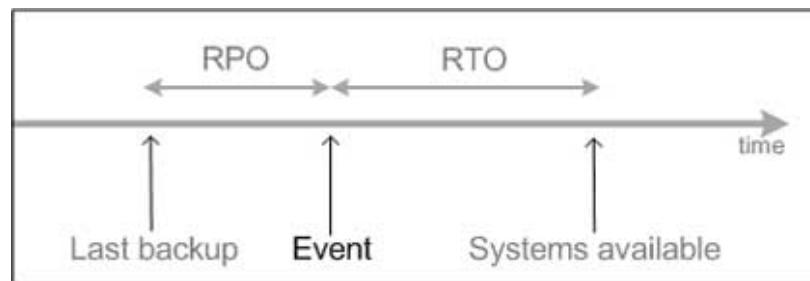
to procesy, polityki i procedury związane z wznowieniem lub utrzymywaniem infrastruktury teleinformatycznej, krytycznej dla organizacji, po wystąpieniu katastrofy naturalnej lub wywołanej przez człowieka

RTO (Recovery Time Objective)

określa maksymalny czas po awarii potrzebny do przywrócenia działania aplikacji, systemów i procesów biznesowych. Określając parametr RTO, należy doprowadzić do kompromisu między potencjalnymi stratami a kosztami rozwiązania umożliwiającego jak najszybsze odtworzenie stanu sprzed awarii.

RPO (Recovery Point Objective)

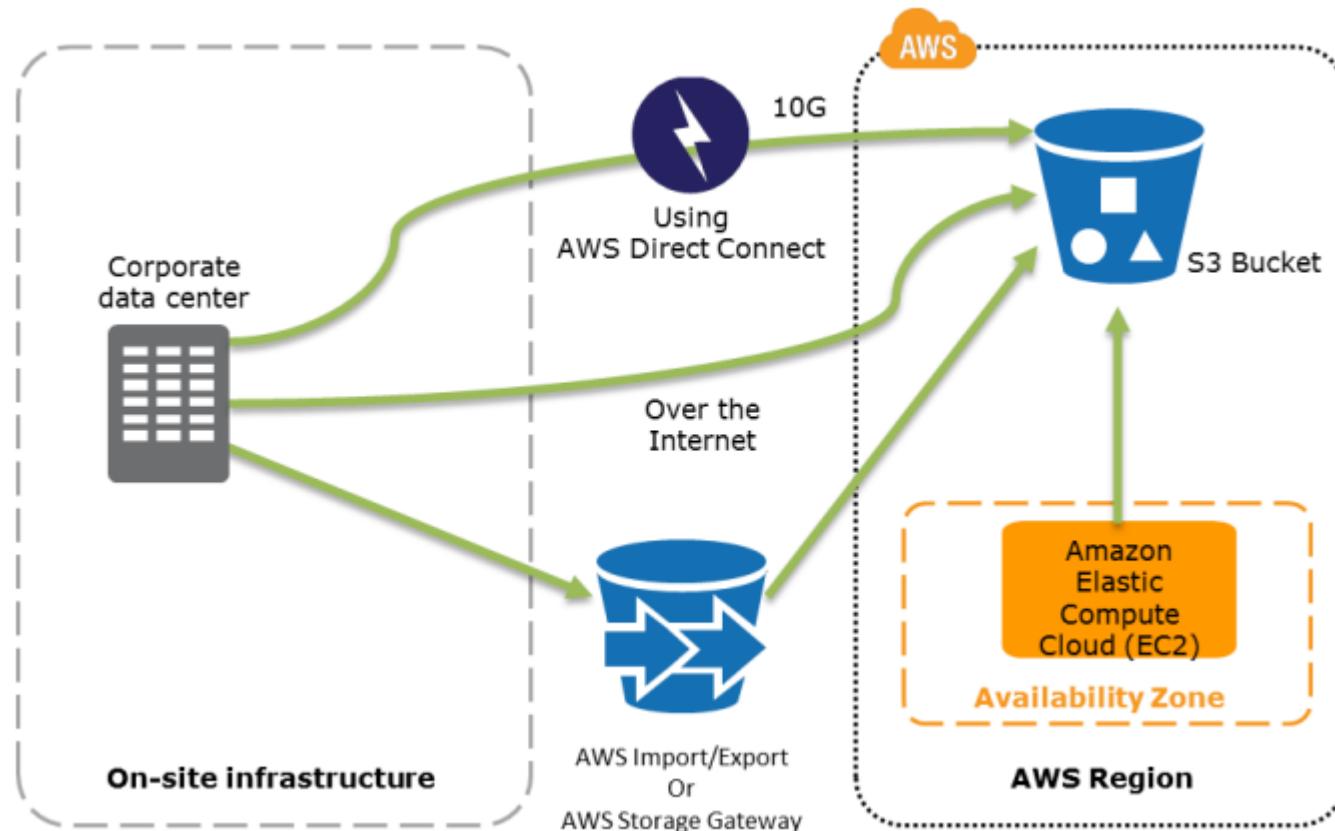
określa moment w przeszłości, w którym po raz ostatni została wykonana kopia danych i do którego momentu działalności będzie można wrócić. Czas ten jest różny i zależy głównie od charakteru działalności - jednym wystarczy kopia sprzed tygodnia, inni zaś potrzebują danych sprzed kilkunastu sekund.



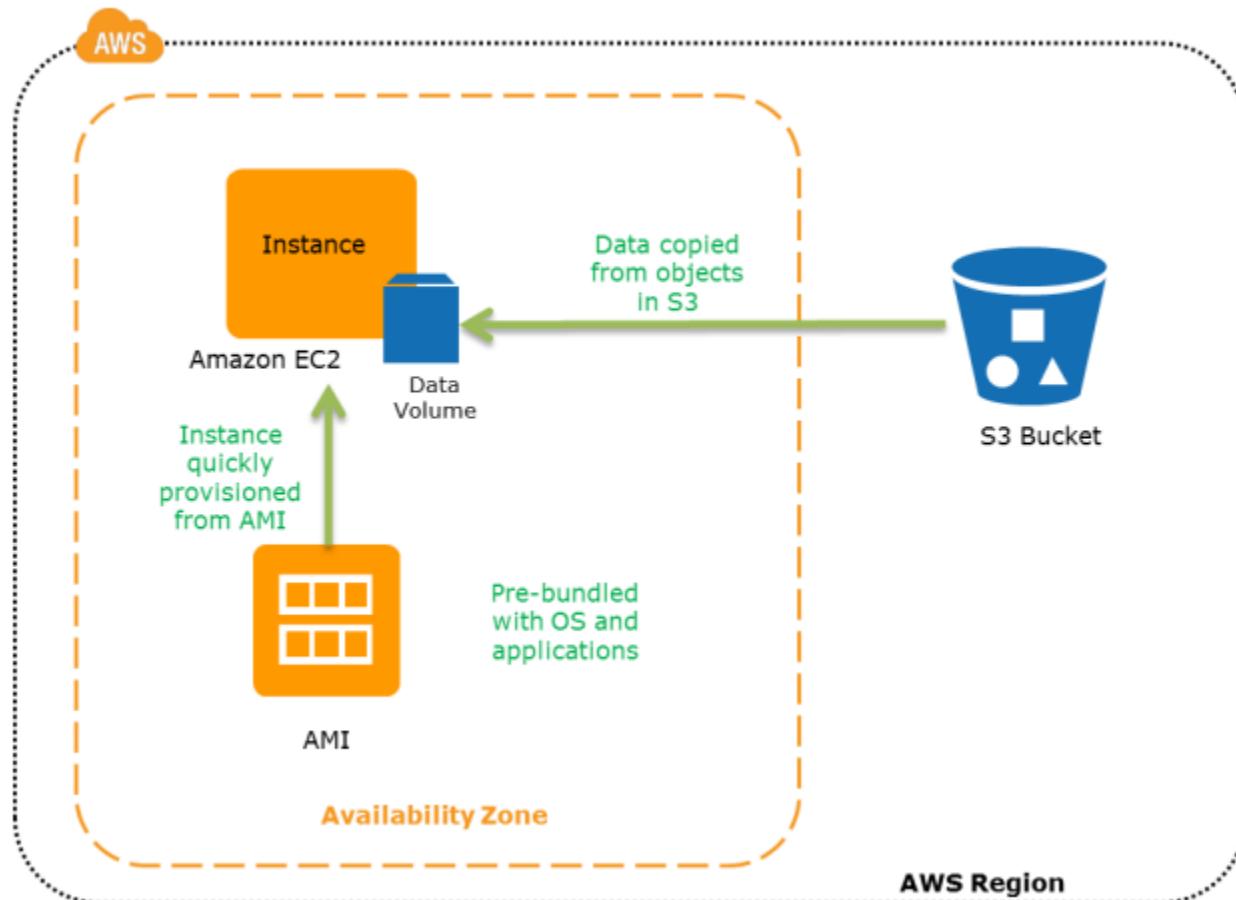
- Kopie bezpieczeństwa i odtwarzanie
(ang. backup and restore)
- Płomyk kontrolny
(ang. pilot light)
- Rezerwa aktywna
(ang. warm standby)
- Aktywny ośrodek zapasowy
(ang. multi-site active-active)



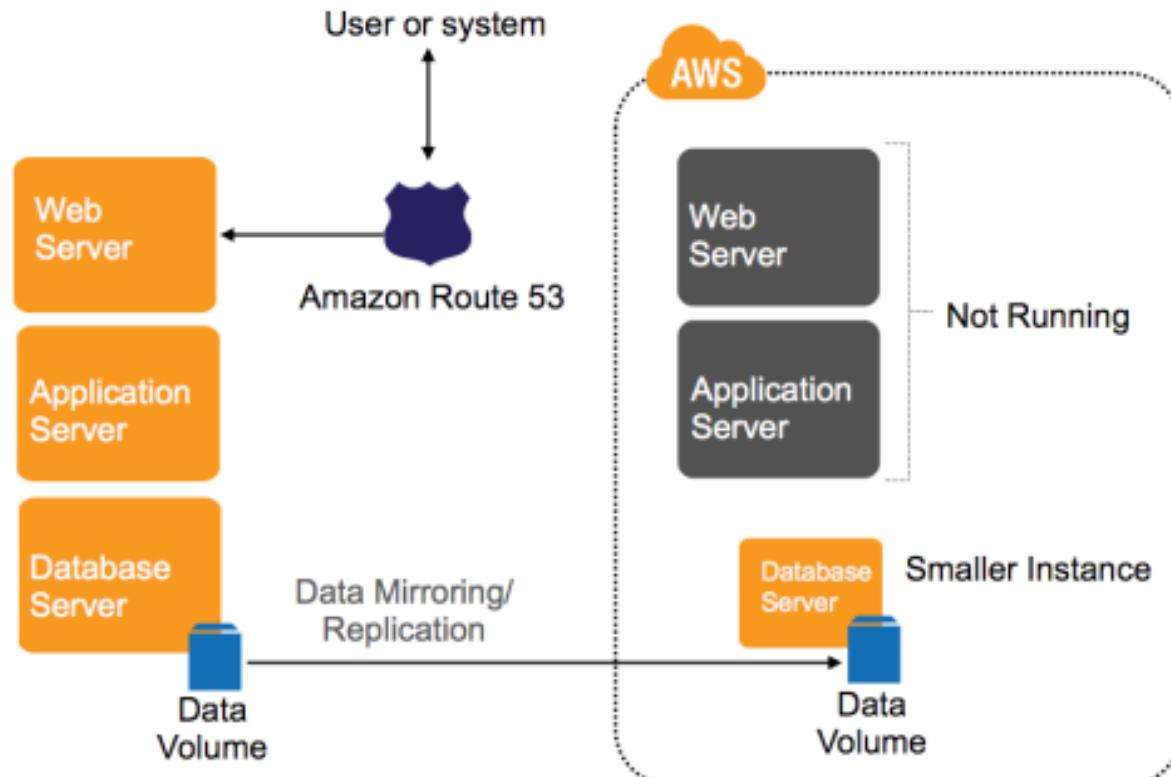
Przygotowanie



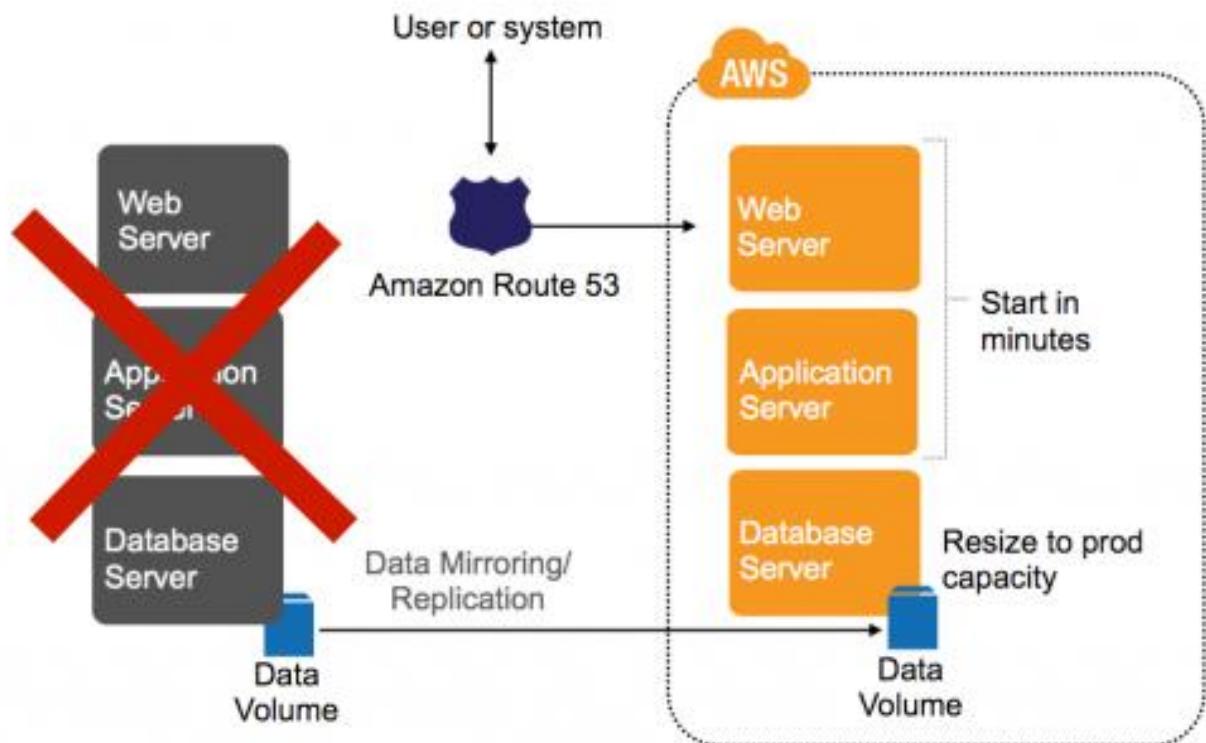
Odtwarzanie



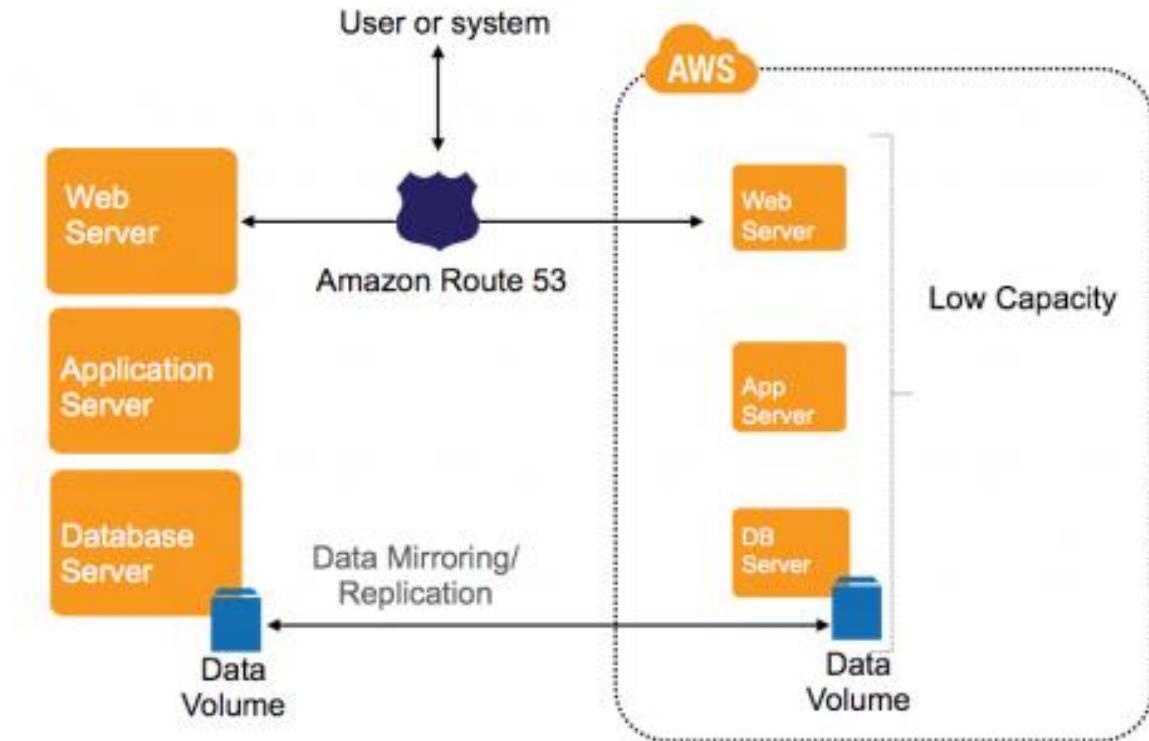
Przygotowanie



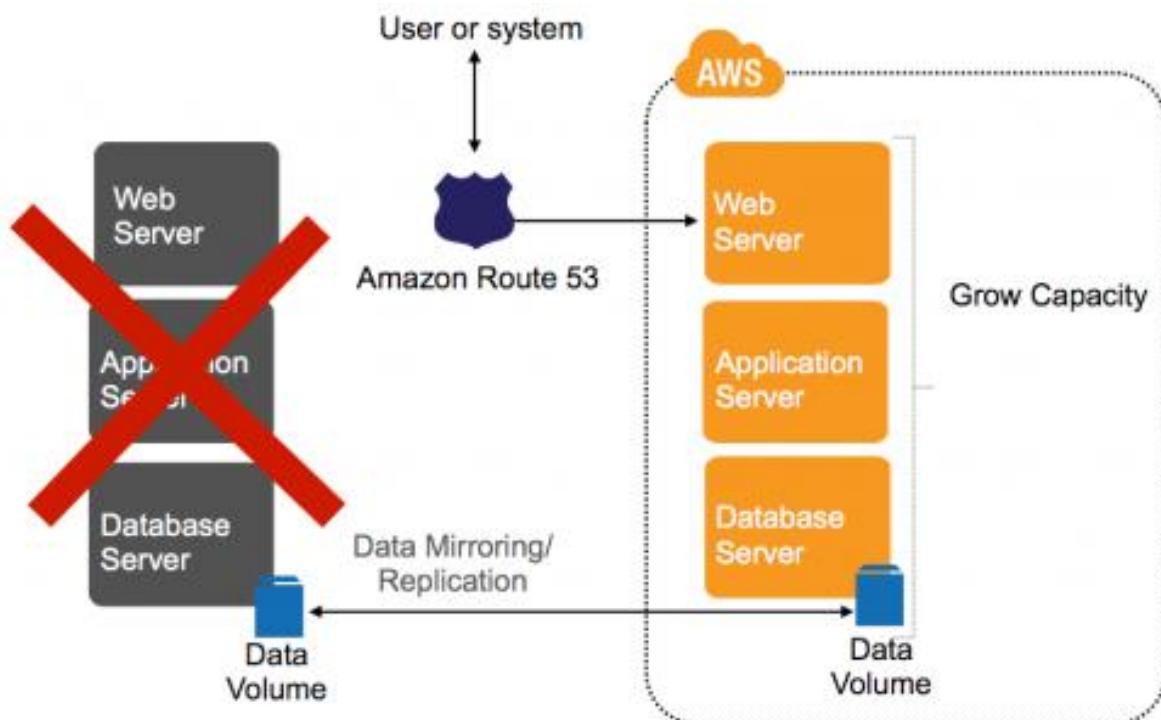
Odtwarzanie



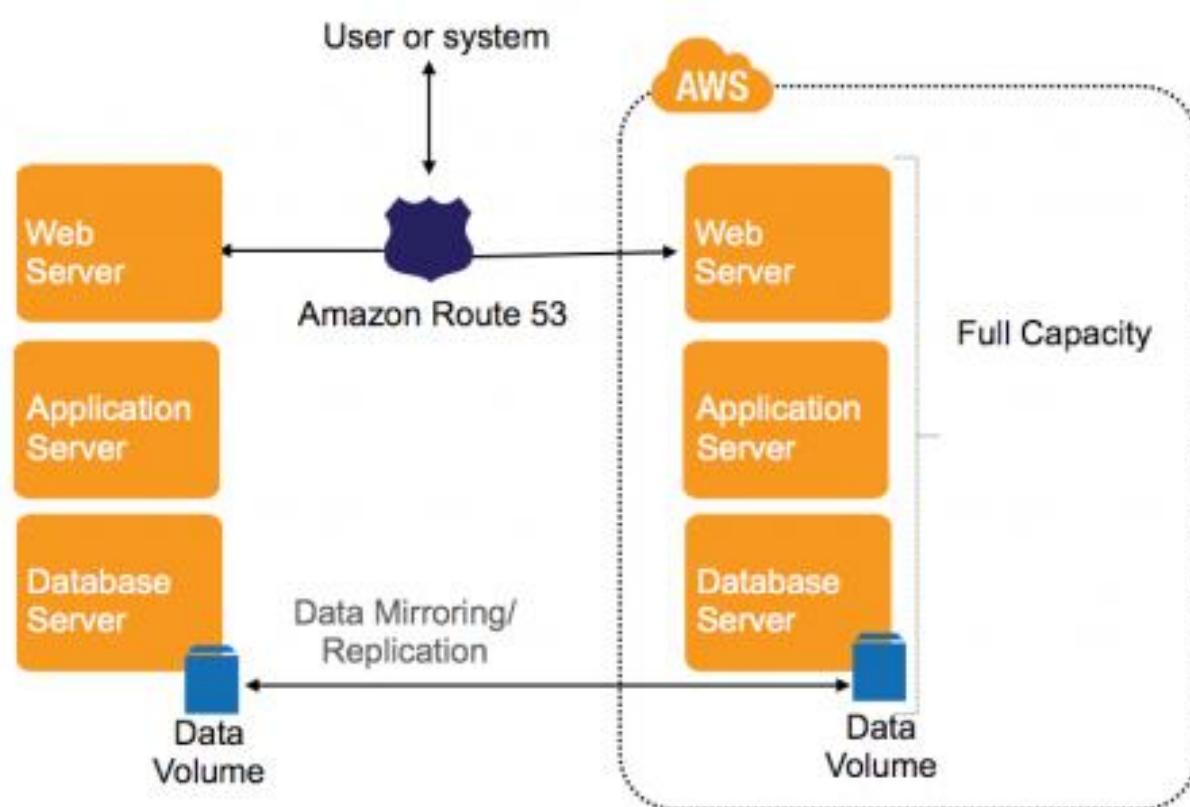
Przygotowanie



Odtwarzanie

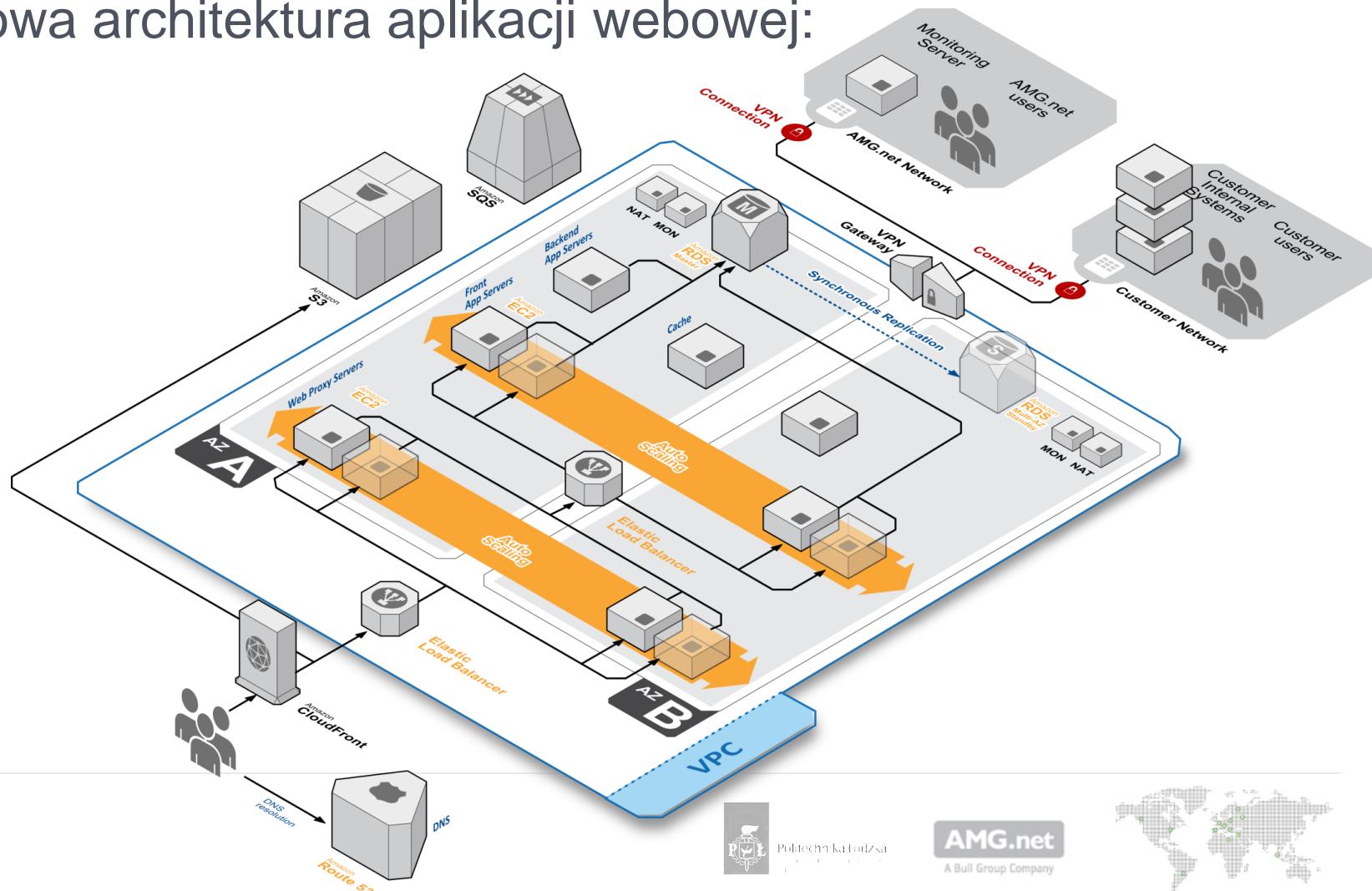


Operacyjne działanie



Przykłady rzeczywistej architektury:

- <http://awssofa.info/>
- typowa architektura aplikacji webowej:

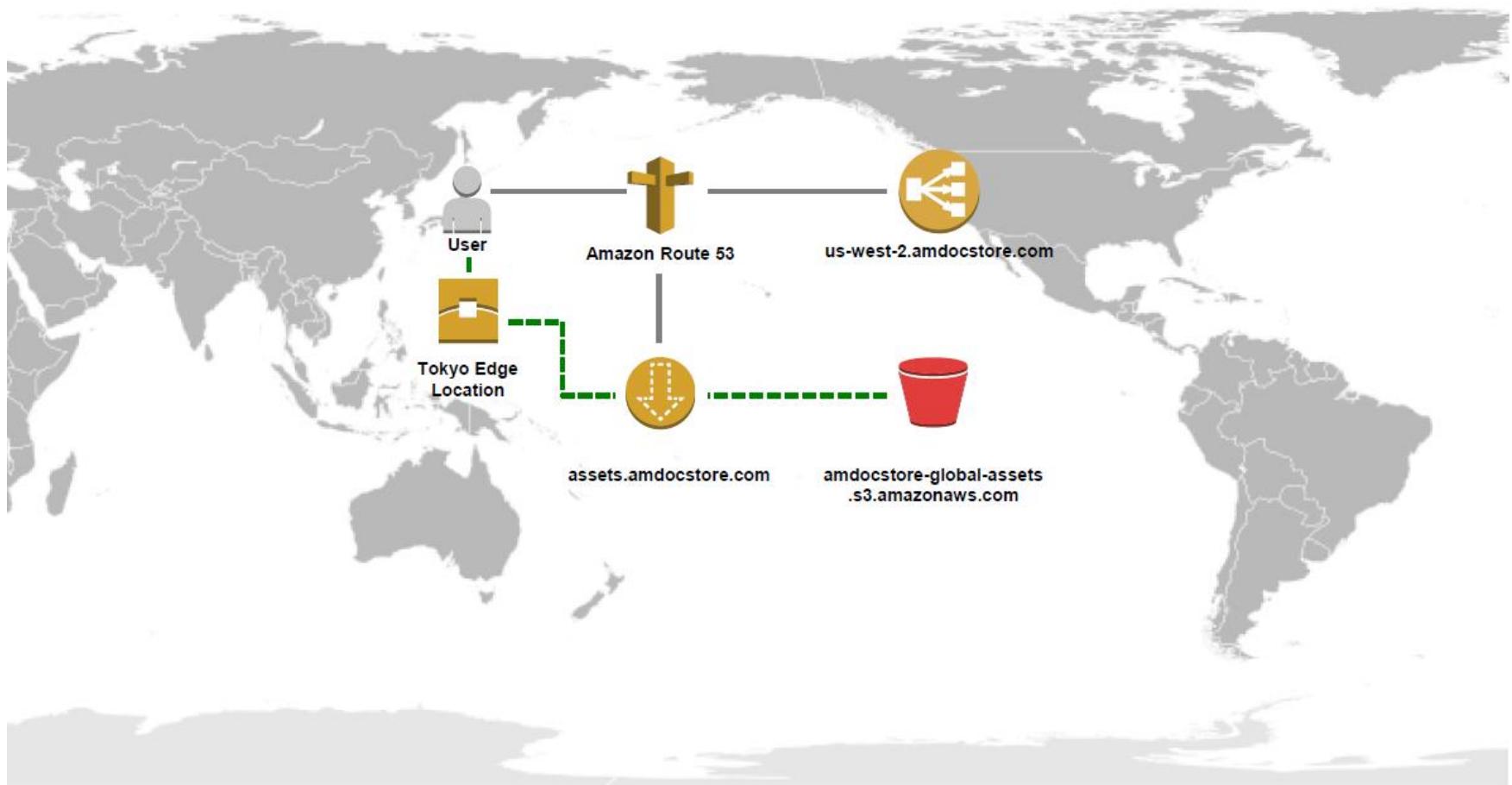


Rozwiązańe klasy DropBox (<https://www.dropbox.com/>)

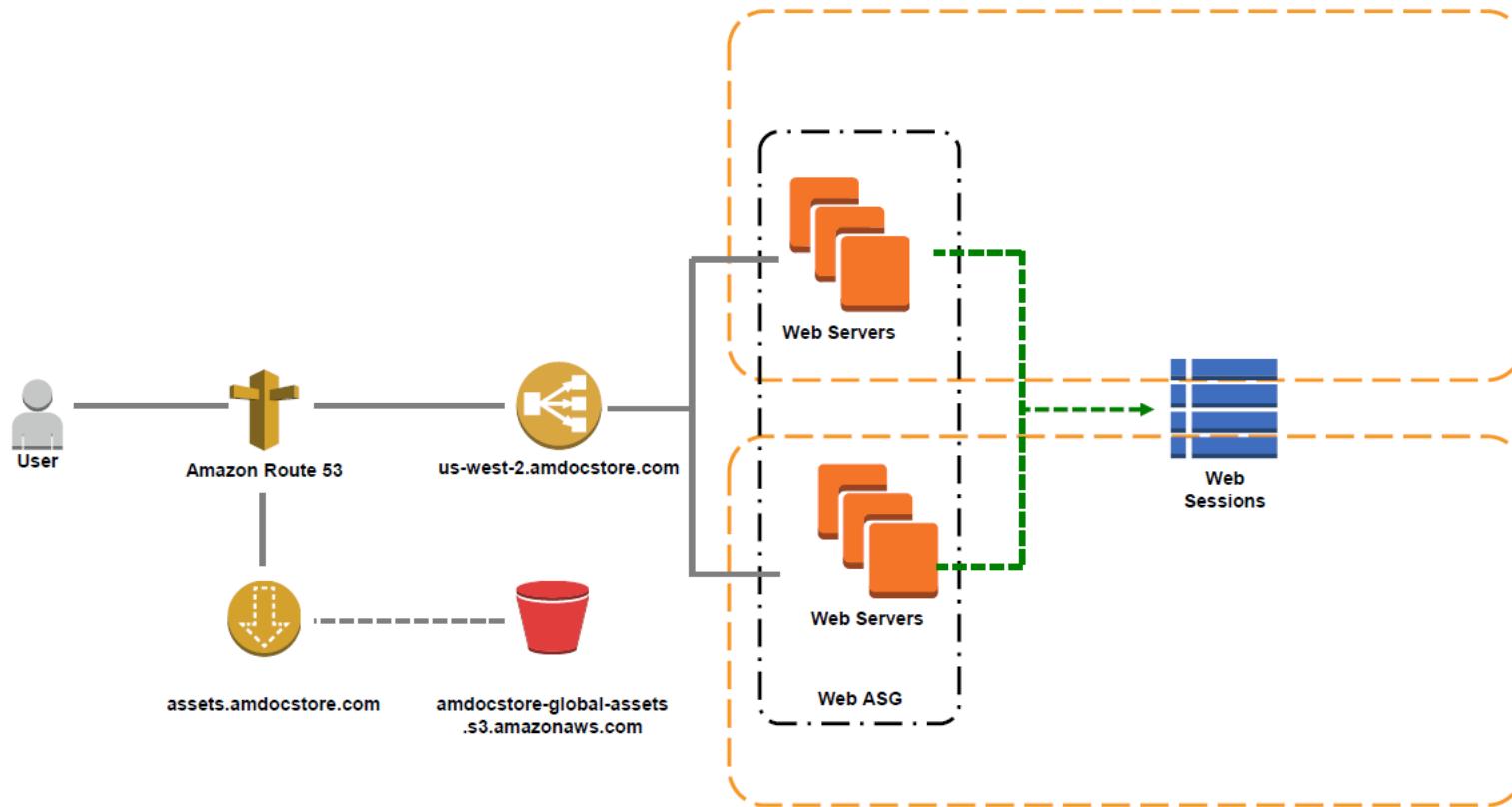
- Możliwość uploadu dokumentów i zarządzania nimi
- Dostęp z dowolnego miejsca przez internet (przeglądarka, telefon)
- Wspierane formaty:
 - PDF, dokumenty MS Office, obrazki, filmy i inne
- Treści tekstowe ekstrahowane (np. OCR), indeksowane i możliwe do przeszukiwania
- Dwa typy kont
 - Darmowe: 5 GB, wyszukiwanie tylko po nazwie i tagach, dostęp do plików tylko dla właściciela
 - Płatne: 20 GB, wyszukiwanie po treści, możliwość udostępniania plików innym
- Dwa interfejsy: WWW + aplikacja mobilna
- API
- Scalable + HA + worldwide

DocStore – LBR DNS

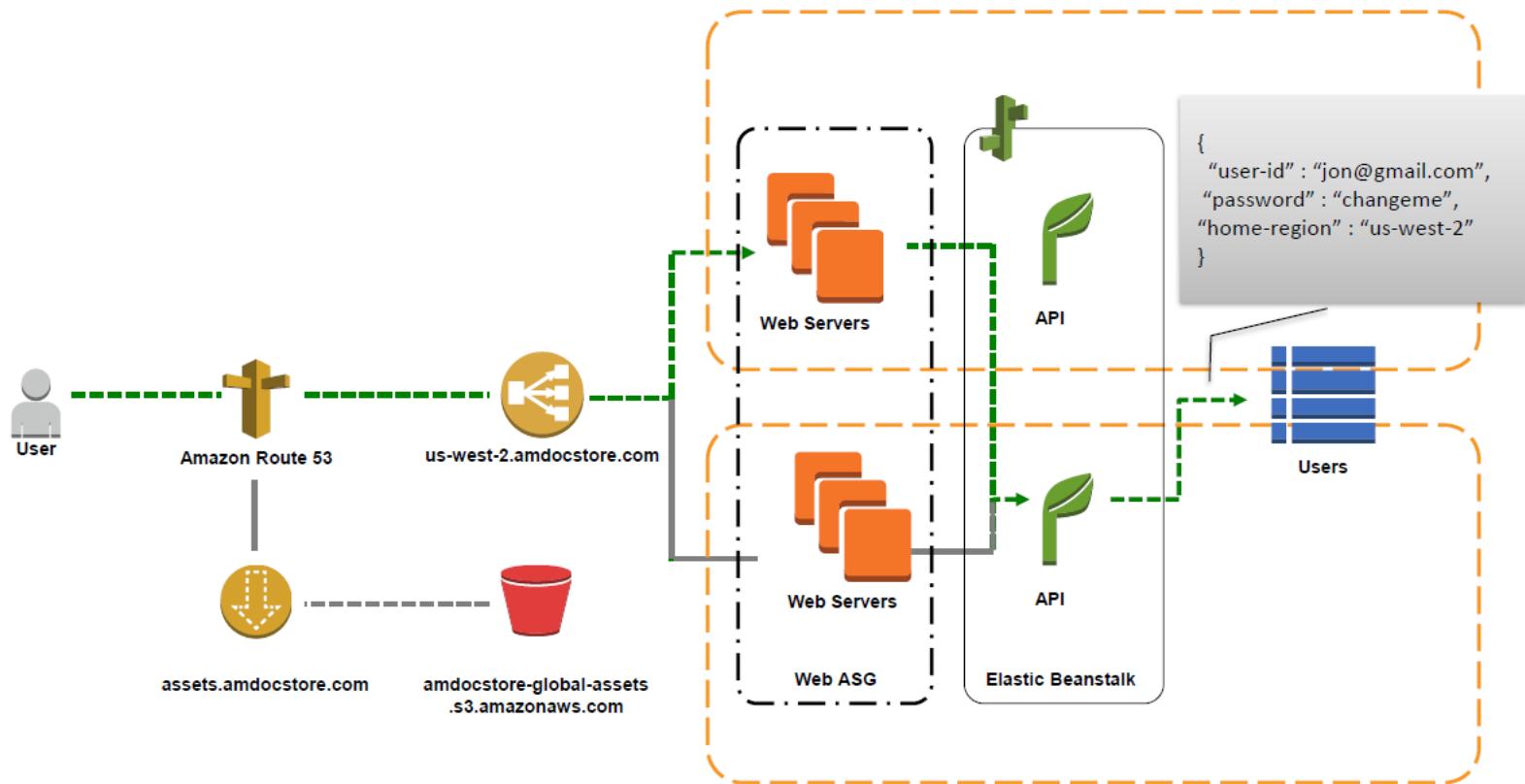




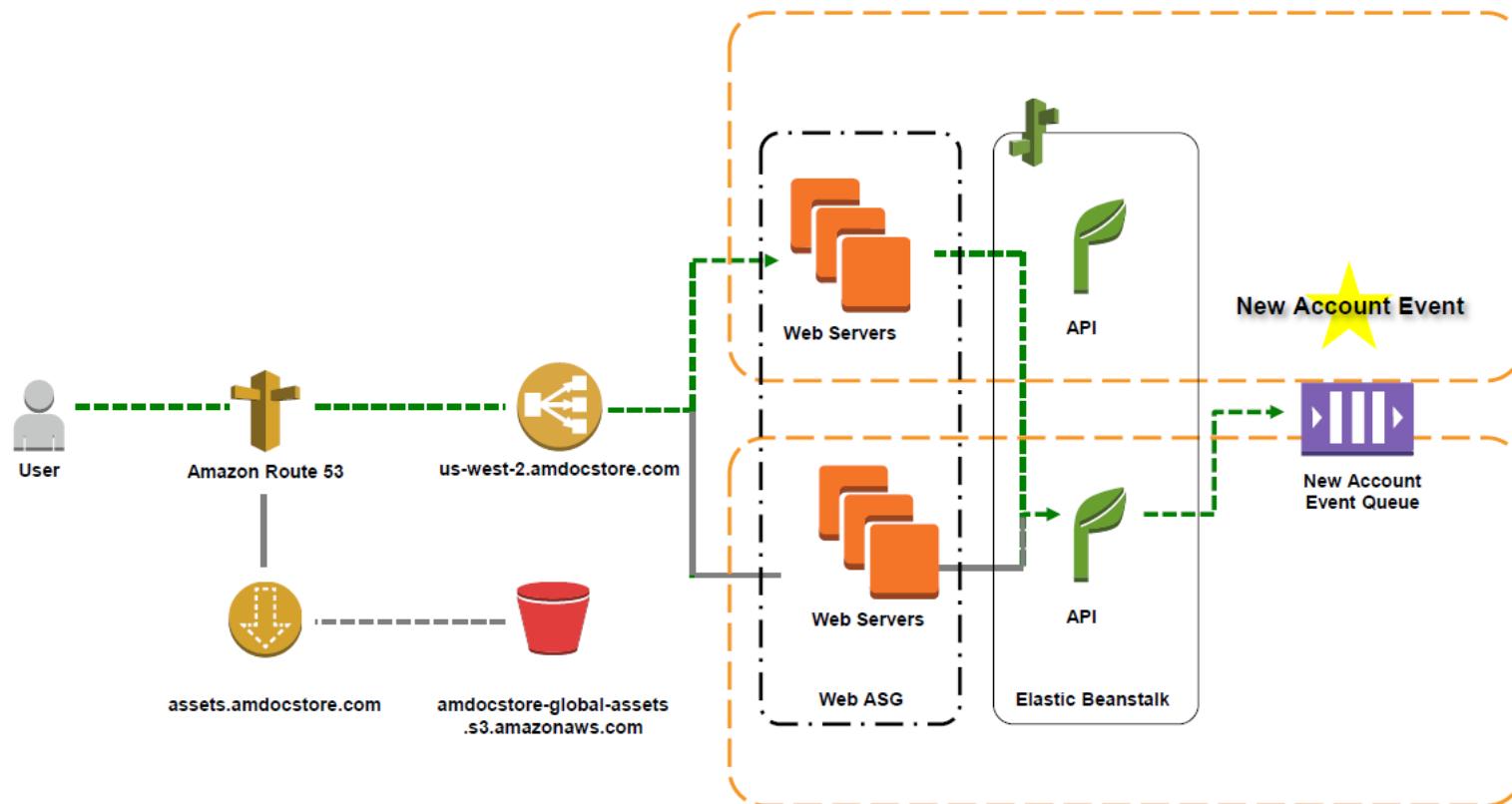
DocStore – web tier



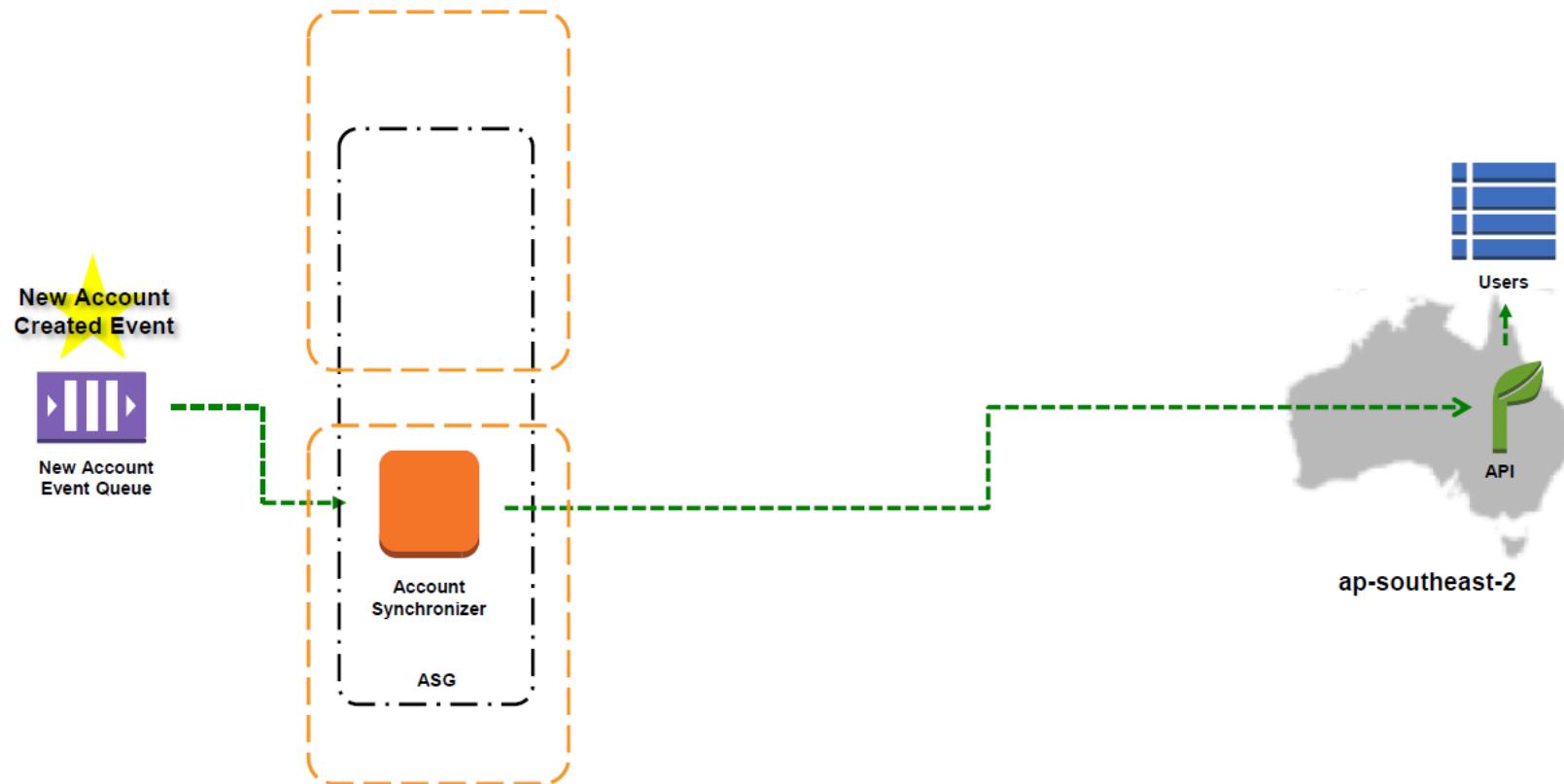
DocStore – API tier



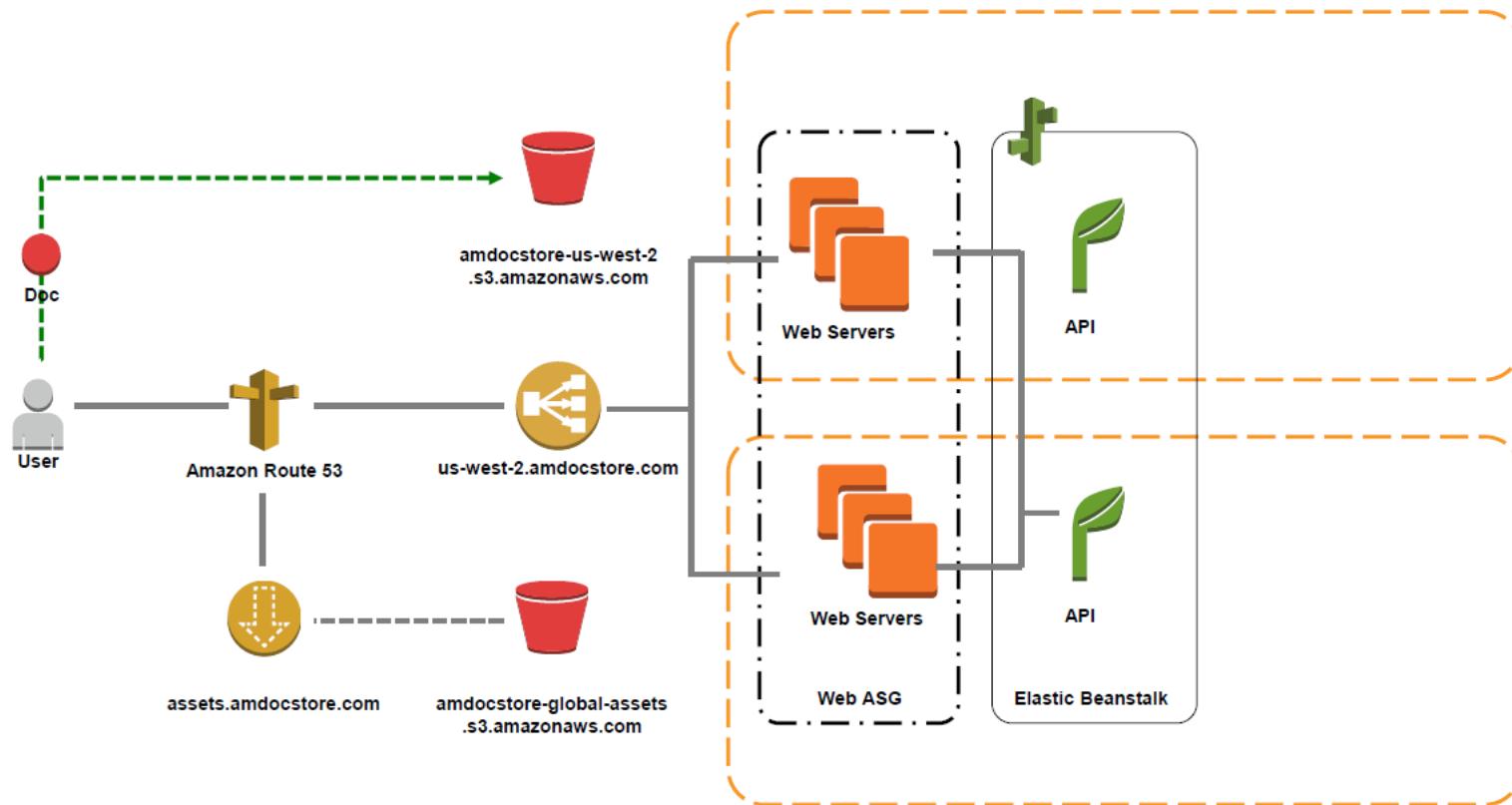
DocStore – async sync



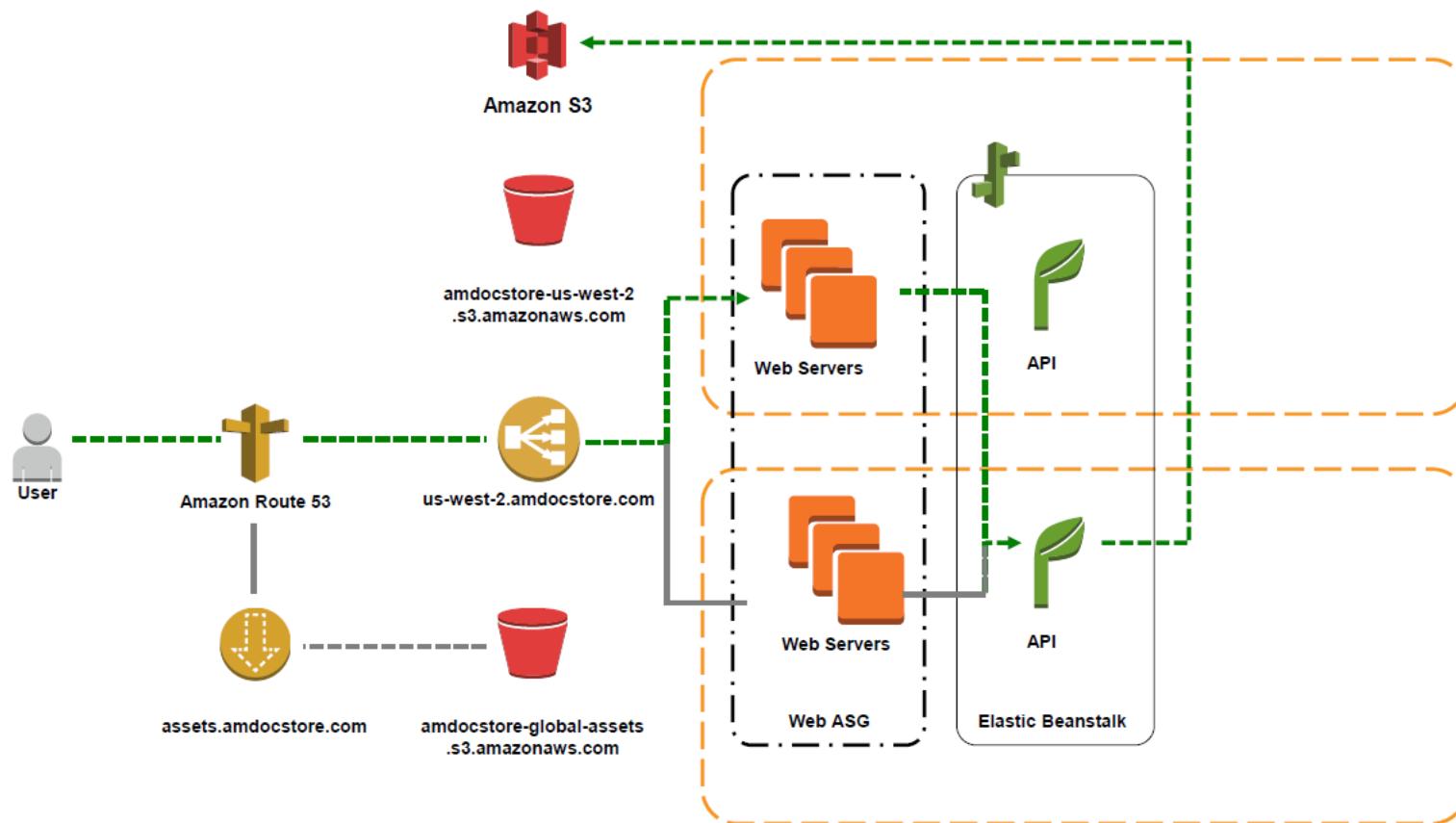
DocStore – async sync



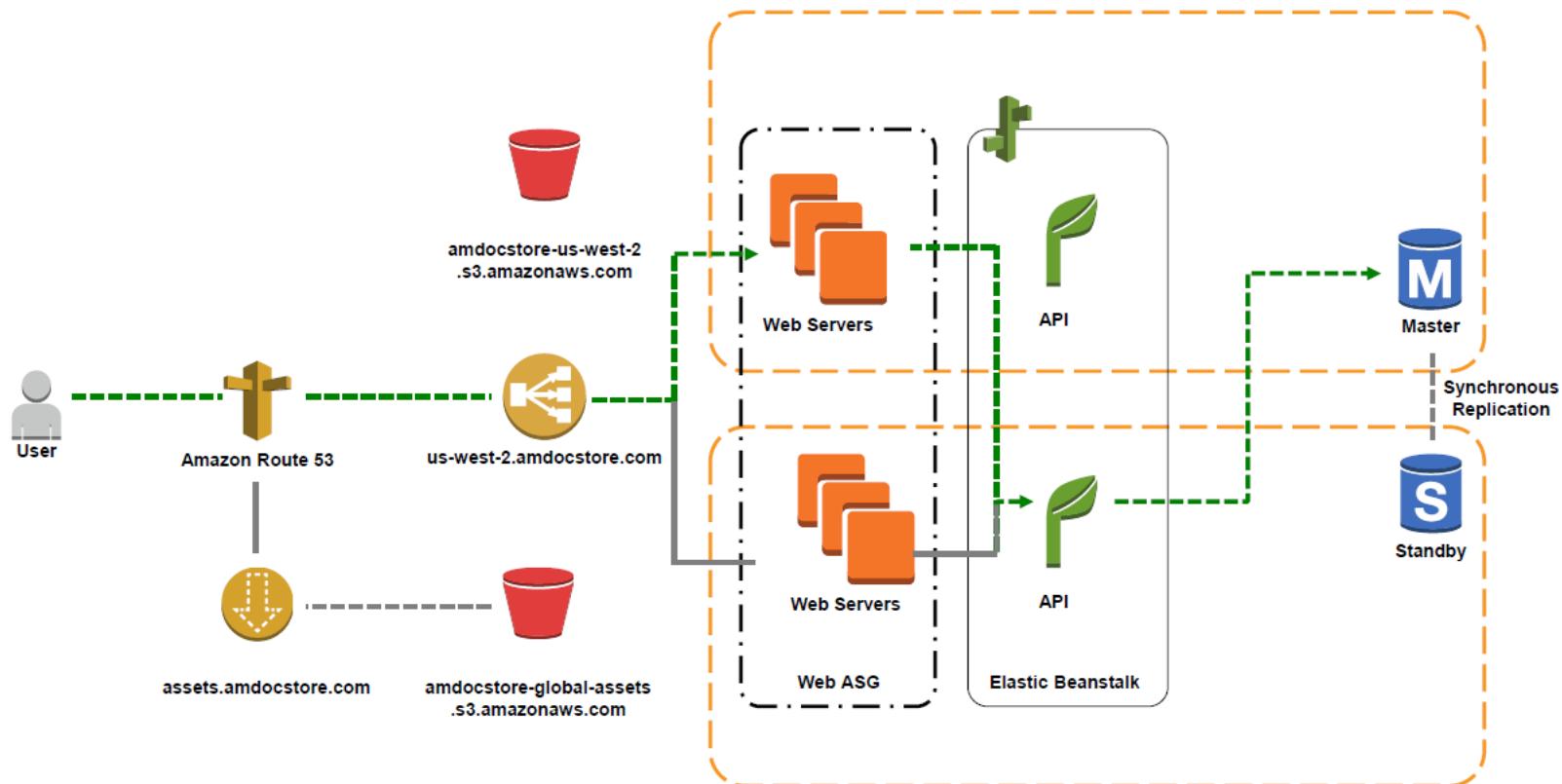
DocStore – direct S3 upload



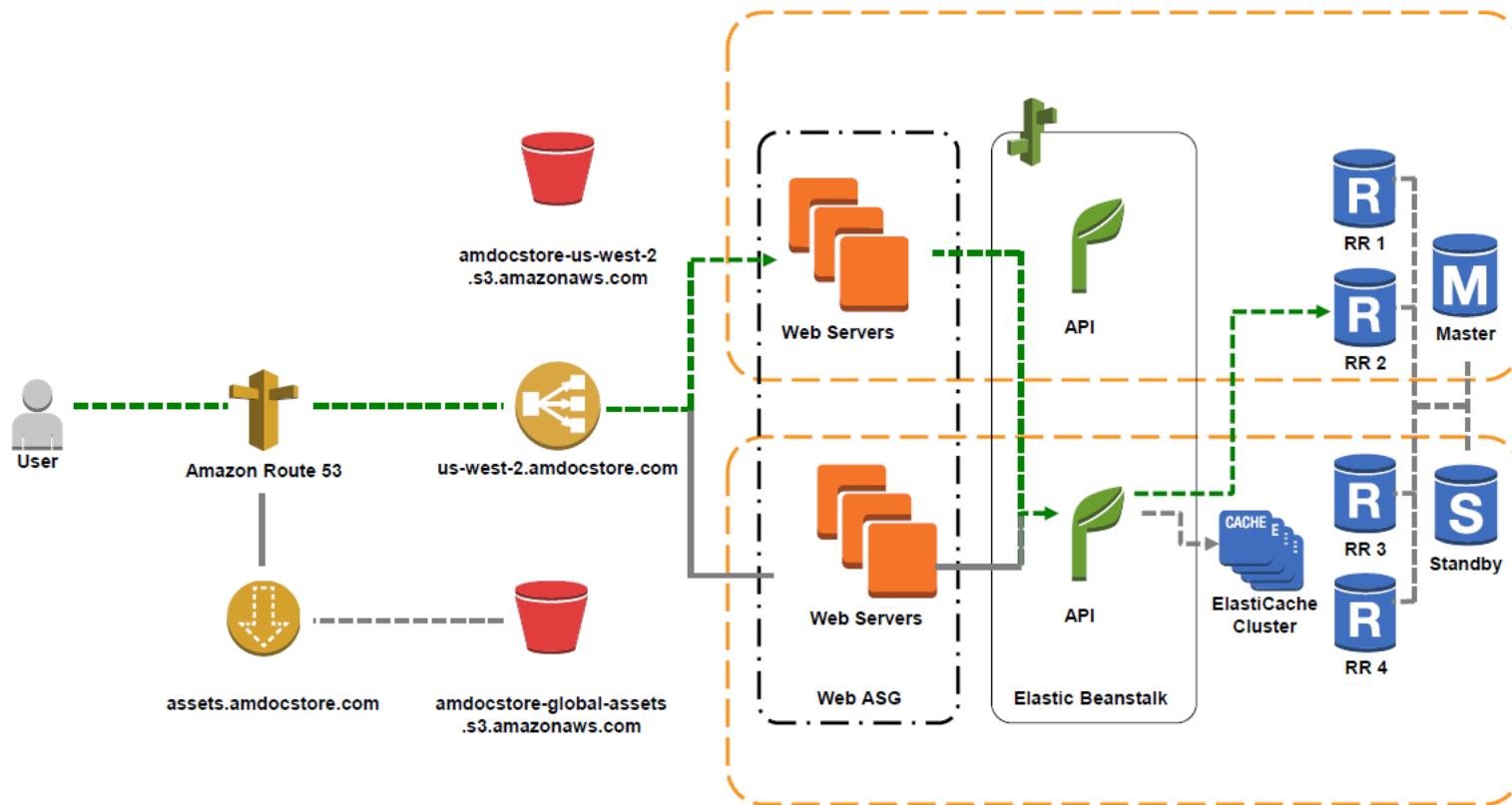
DocStore – obtain metadata from S3



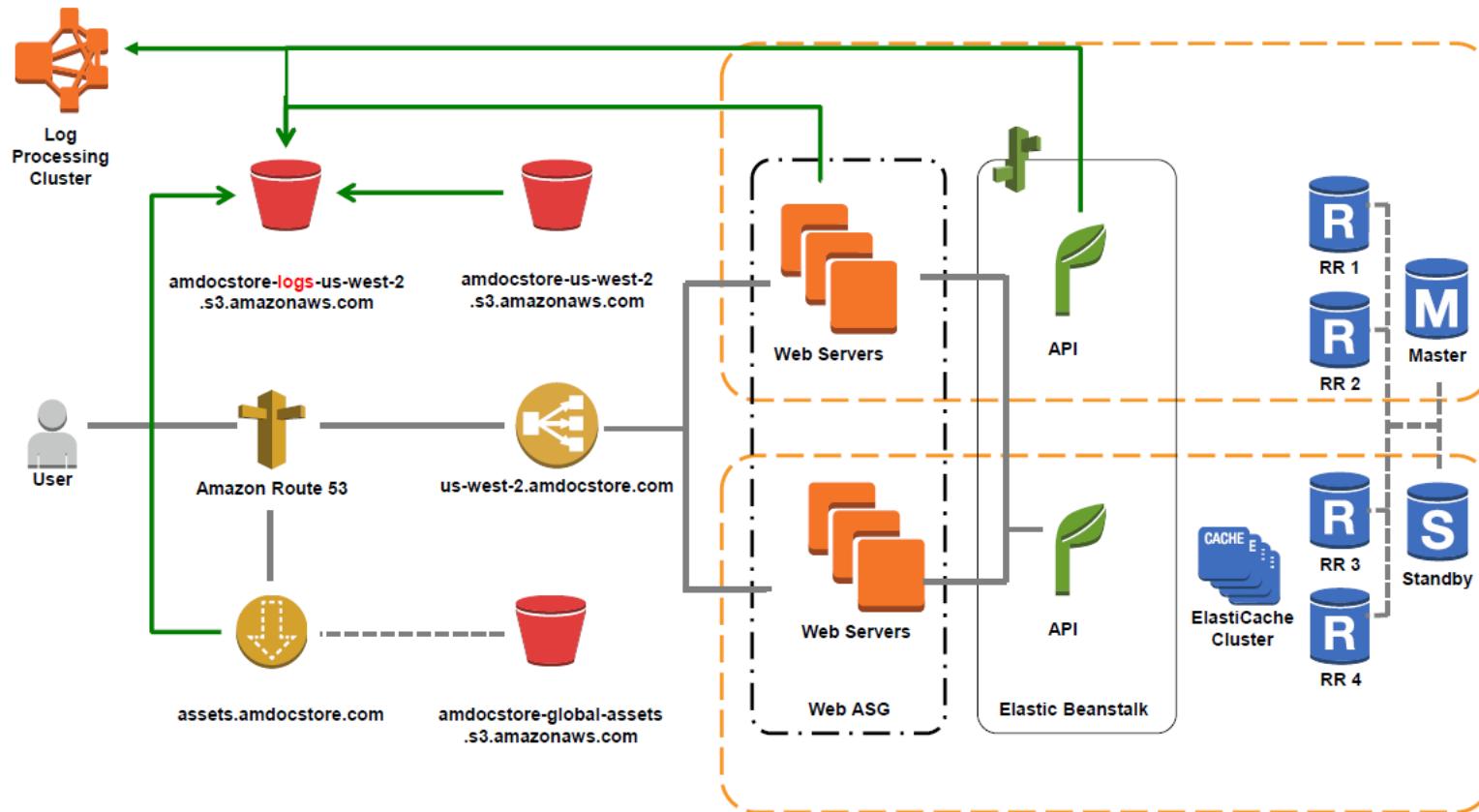
DocStore – save metadata in DB



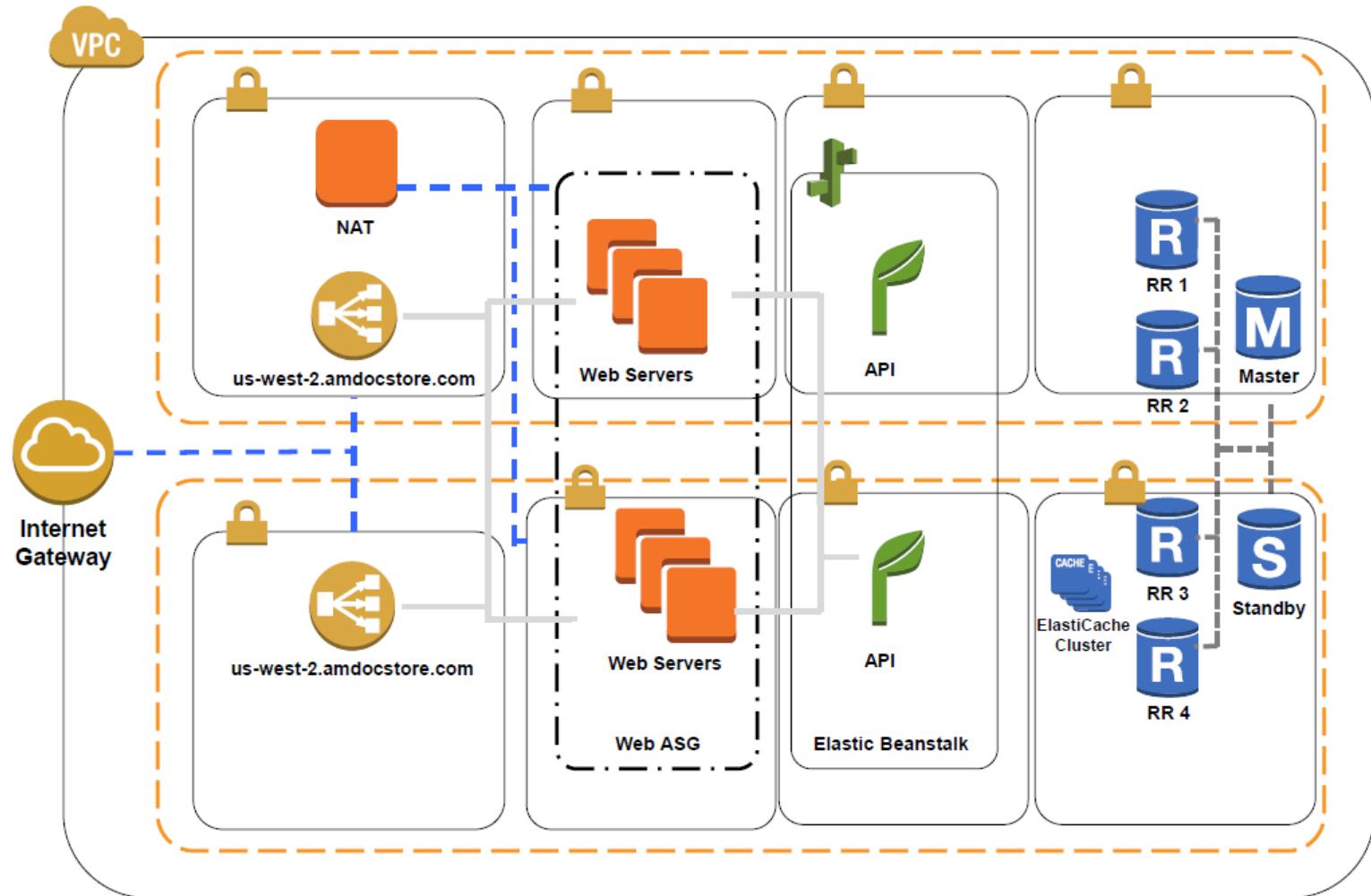
DocStore – scalable DB reads



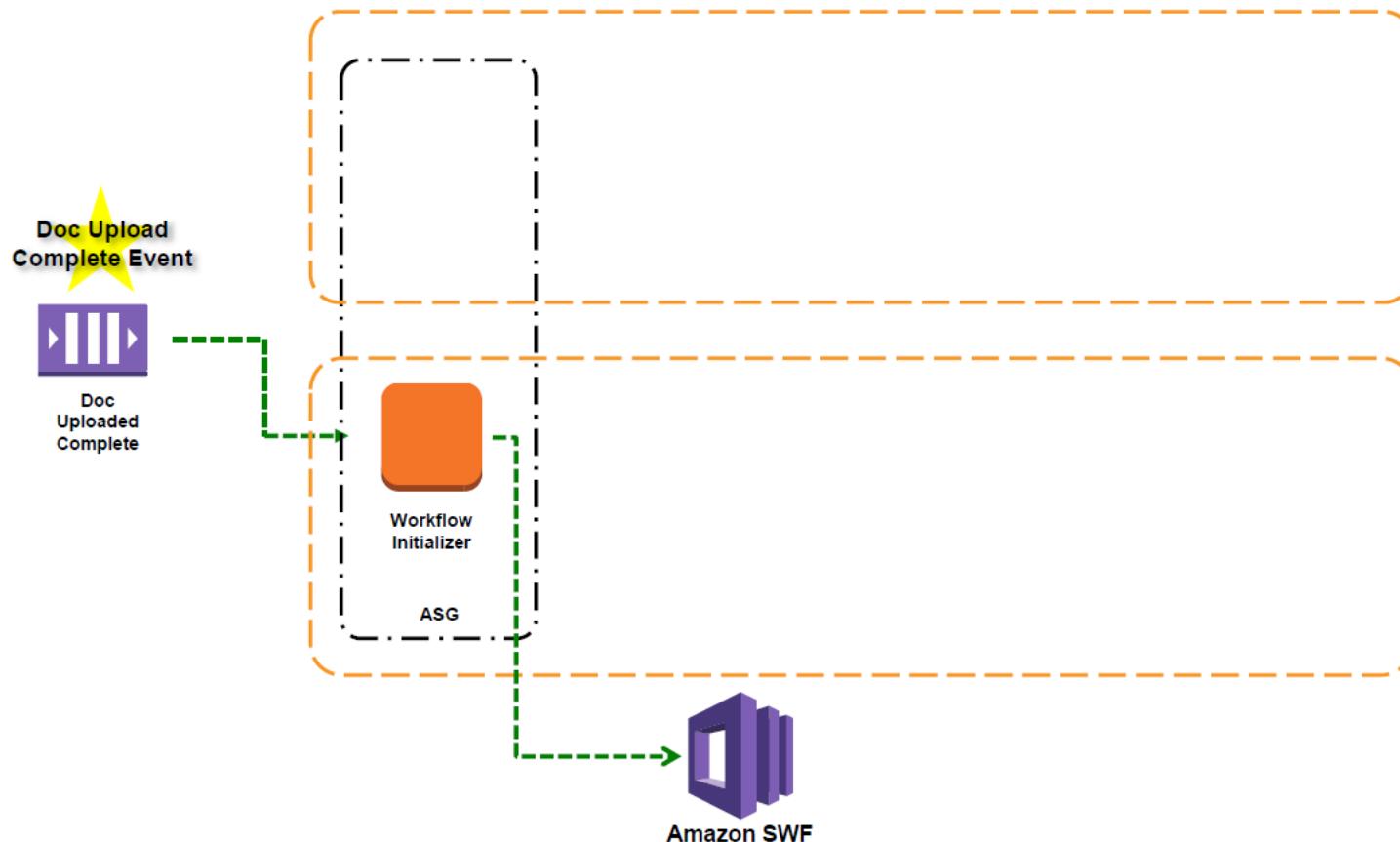
DocStore – logi aplikacyjne



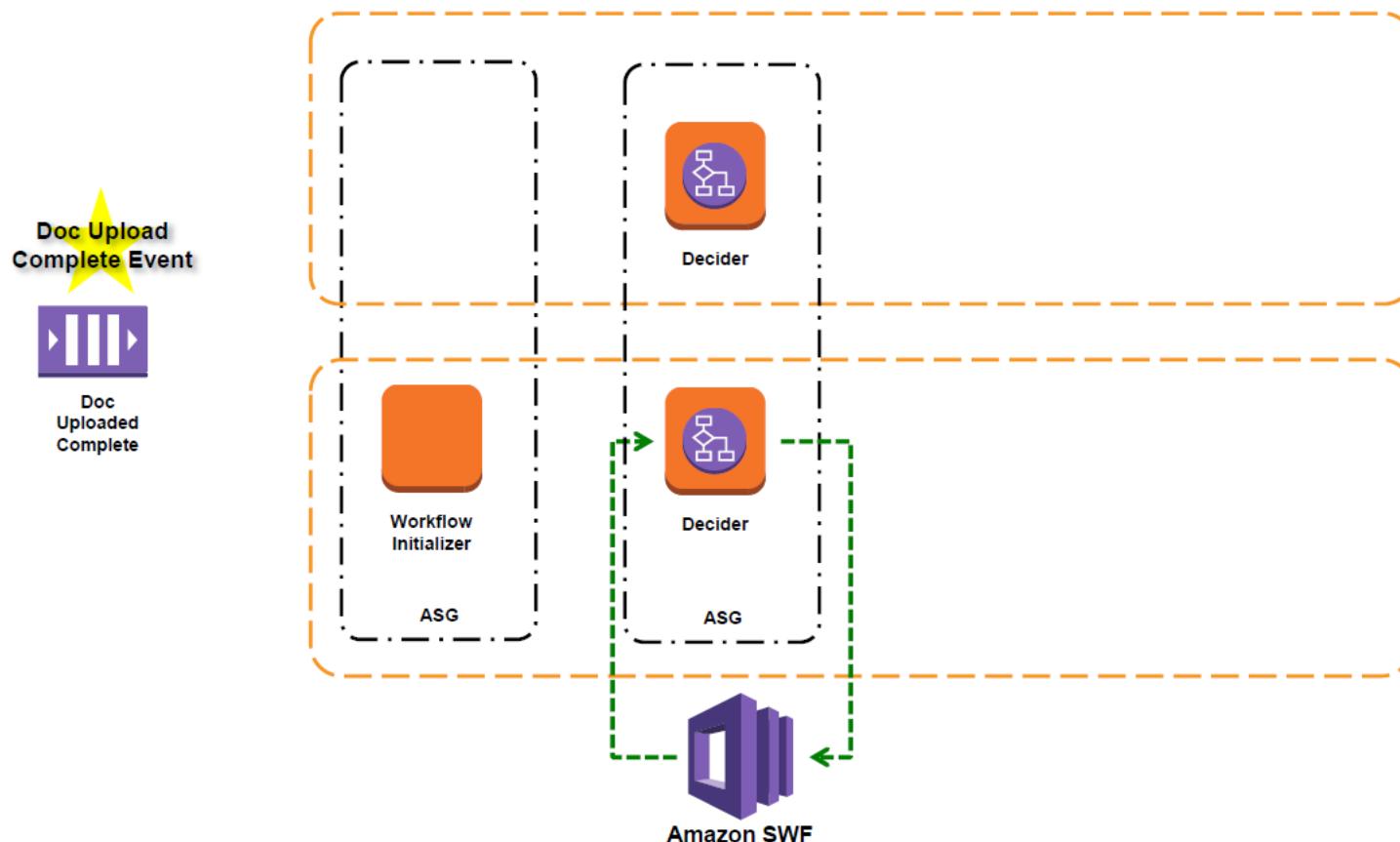
DocStore - VPC



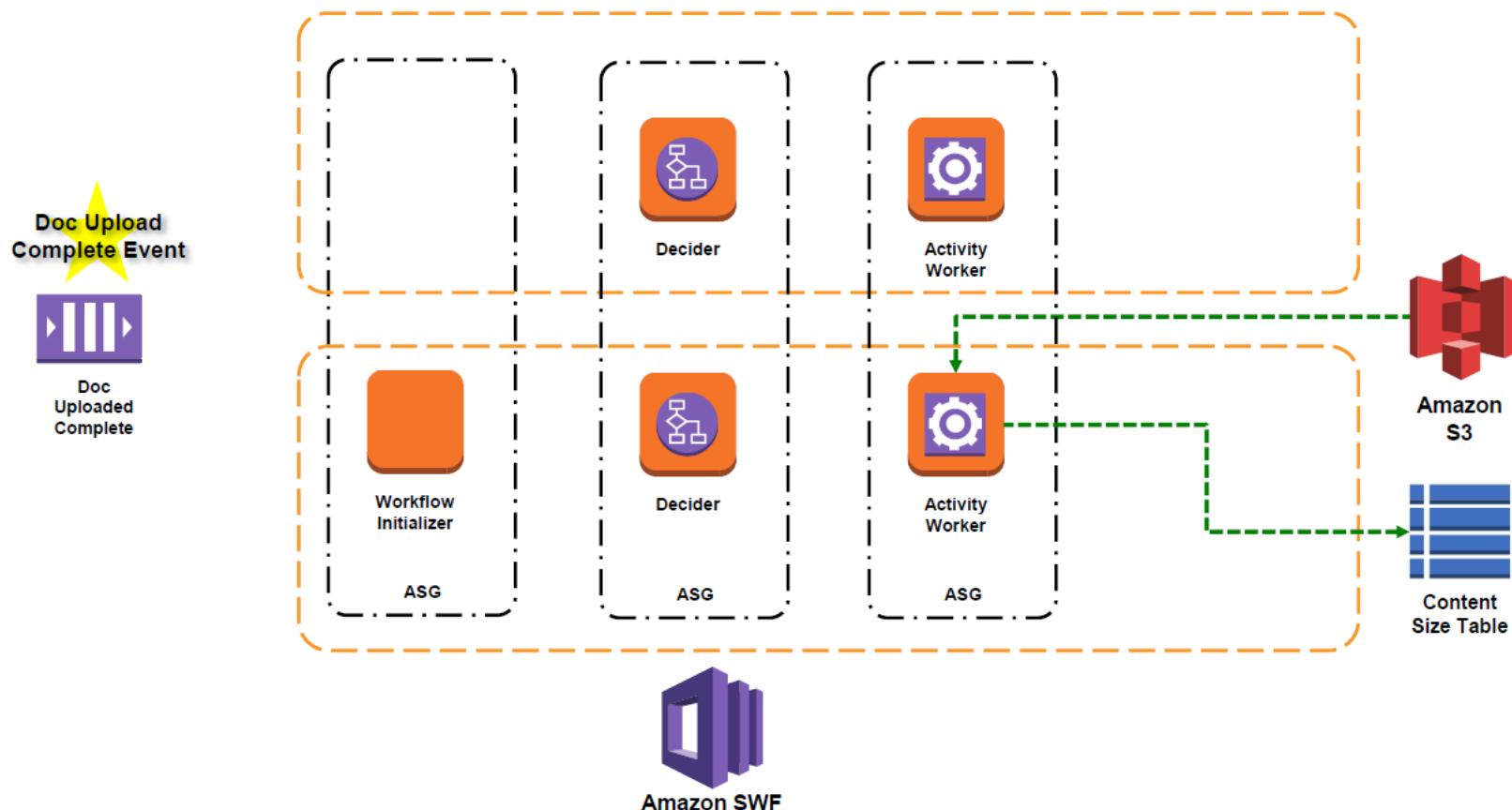
DocStore – async post processing



DocStore – async post processing



DocStore – async post processing

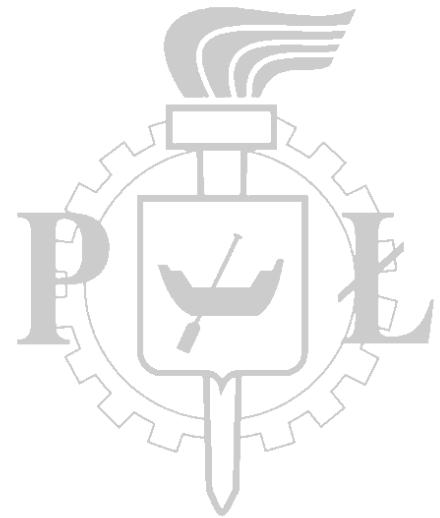


- Wykorzystanie globalnej infrastruktury (CDN, LBR DNS, Regiony)
- Wykorzystanie Multi-AZ
- Wykorzystanie bezpośrednie Object Store (S3)
- Wykorzystanie baz danych NoSQL (DynamoDB) i SQL (RDS)
- Synchronizacja minimalnej ilości danych między regionami – tylko w celu odpowiedniego przekierowania po treści
- Asynchroniczne procesowanie w tle
- Automatyczne skalowanie
 - EC2 autoscaling
 - Cache, read replicas

- Jinesh Varia:
Architecting for the Cloud: Best Practices
- John Hildebrandt
Architecting for High Availability
- Harish Ganesan
Overcoming Outages in AWS : HA Architectures
- AWS Architecture Center
<http://aws.amazon.com/architecture/>
- Jeff Barr, Attila Narin, and Jinesh Varia
Building Fault-Tolerant Applications on AWS



wydział
elektrotechniki
elektroniki
informatyki
i automatyki



Dziękujemy za uwagę