



JAWS-UG
AWS User Group - Japan

AWSクラウドデザインパターン (CDP) を極める



玉川憲の自己紹介

❖ 所属

- ❖ 技術統括部長 / エバンジェリスト(初号機)
- ❖ アマゾンデータサービスジャパン株式会社

❖ ID

- ❖ @KenTamagawa

❖ 好きなAWSサービス

- ❖ Amazon S3

❖ 好きなCDP

- ❖ Server Swapping



インフラストラクチャは
ソフトウェアになった



プログラマブル！

```
$options = array('KeyName' => "User's Keys",  
                'InstanceType' => "m1.small");
```

```
$res = $ec2->run_instances("ami-48aa4921",  
                           1, 1, $options);
```

AWSのサービス

お客様のアプリケーション

ライブラリ & SDKs

Java, PHP, .NET,
Python, Ruby

Web インターフェース

Management Console

IDE プラグイン

Eclipse
Visual Studio

デプロイと自動化

AWS Elastic Beanstalk
AWS CloudFormation

認証 & 請求

AWS IAM
Identity Federation
Consolidated Billing

モニタリング

Amazon CloudWatch

スケーリング

Auto Scale

ネットワーク&ルーティング

Amazon VPC
Amazon Elastic LB
Amazon Route 53
AWS Direct Connect

コンテンツ配信

Amazon
CloudFront

メッセージング

Amazon SNS
Amazon SQS

分散処理

Elastic
MapReduce

メール配信

Amazon SES

コンピュータ処理

Amazon EC2

ストレージ

Amazon S3
Amazon EBS

データベース

Amazon RDS
Amazon SimpleDB
Amazon DynamoDB
Amazon ElastiCache

AWS のグローバルなインフラ

Amazon Appstoreで日本アプリ登録開始！



Amazon Web Services ブログ

開発者、ITプロフェッショナル向けにAmazon Web Services(AWS)が提供するアマゾンクラウド(Amazon EC2, S3, RDS, EMR他)の公式ブログです。最新情報をエバンジェリスト



最近の記事

アプリ開発事業者のみなさまへ - 日本でのアプリ登録スタート!

【AWS発表】Amazon RDS で無料使用枠が利用可能に!

【AWS発表】Amazon RDS for Oracle でマイクロインスタンスが利用可能に!

【AWS発表】Amazon RDS で SQL Server 2012 が選択可能に!

NASDAQ OMX FinCloud - 金融業界向けクラウドソリューションの登場

【AWS発表】Amazon RDS でハイパフォーマンスなプロビジョンドIOPSストレージを利用可能に(東京リージョンはもう少しお待ちください)

【AWS発表】クラスター GPU インスタンスがEUリージョンとVPC内で利用可能に

【AWS発表】Amazon VPCに2つの新機能追加 - BGPをサポートしないデバイスでもVPCに接続可能に

【AWS発表】Amazon RDSでOracle Data Pumpが利用可能に。データのインポート、エクスポートをより高速

« [【AWS発表】Amazon RDS で無料使用枠が利用可能に!](#) | [メイン](#)

アプリ開発事業者のみなさまへ - 日本でのアプリ登録スタート!

すでに米国の[Amazon Appstore](#)では、大変ご好評いただいております。非常に多くのお客様にスマートフォンやタブレット用のAndroidアプリを導入、ご利用いただいております。

アマゾンでは、アメリカでのアプリの配信に加え、最近英国、イタリア、フランス、ドイツ、スペインにApp配信を拡大しました。このグローバル展開の一環として、**日本でも、年内に日本で提供いただく予定のアプリの登録受付を開始いたしました。**アプリの配信について詳しくは、[Amazon Mobile App Distribution Portal](#)でご確認下さい。[日本語版FAQはこちら](#)。

日本での配信開始により、開発者のみなさまには、新たなマーケットをご提供し、売り上げの拡大の機会をご提供いたします。デジタルコンテンツや定期購読などの**アプリ内課金**もご利用いただくことができ、**『本日限定 無料アプリ』**のようなプロモーションをご利用いただくことで、[Amazon Appstore](#)内における露出を高めるサポートをさせていただきます。

Getting started

Sign up today to offer your apps to millions of client. Learn about making your apps available on [Amazon Appstore](#) and by downloading the [Development Resources](#)

The process is simple:

1. Download the [Amazon Mobile App SDK](#).
2. Create a developer account.
3. Submit your mobile apps for review.
4. Sell your approved apps on [amazon.com](#).

Create an Account

プロセッシング、ストレージ、コンテンツ配信サービスが必要なゲームにとって、AWSは非常に適したサービスです。さらに詳しく知りたい方は[AWS game hostingページ](#)をご覧ください。[KingNet](#)や[Playfish](#)、[Gumi](#)、[Onoko](#)に[Cmune](#)といった企業がゲームをホストし、どのような模のゲームであっても経済的で質の高い経験を提供するために、どのようにAWSを活用いただいているのか、詳しく紹介しております。

堀内康弘 ([Facebook](#), [Twitter](#))

新世代の アーキテクテイング が必要



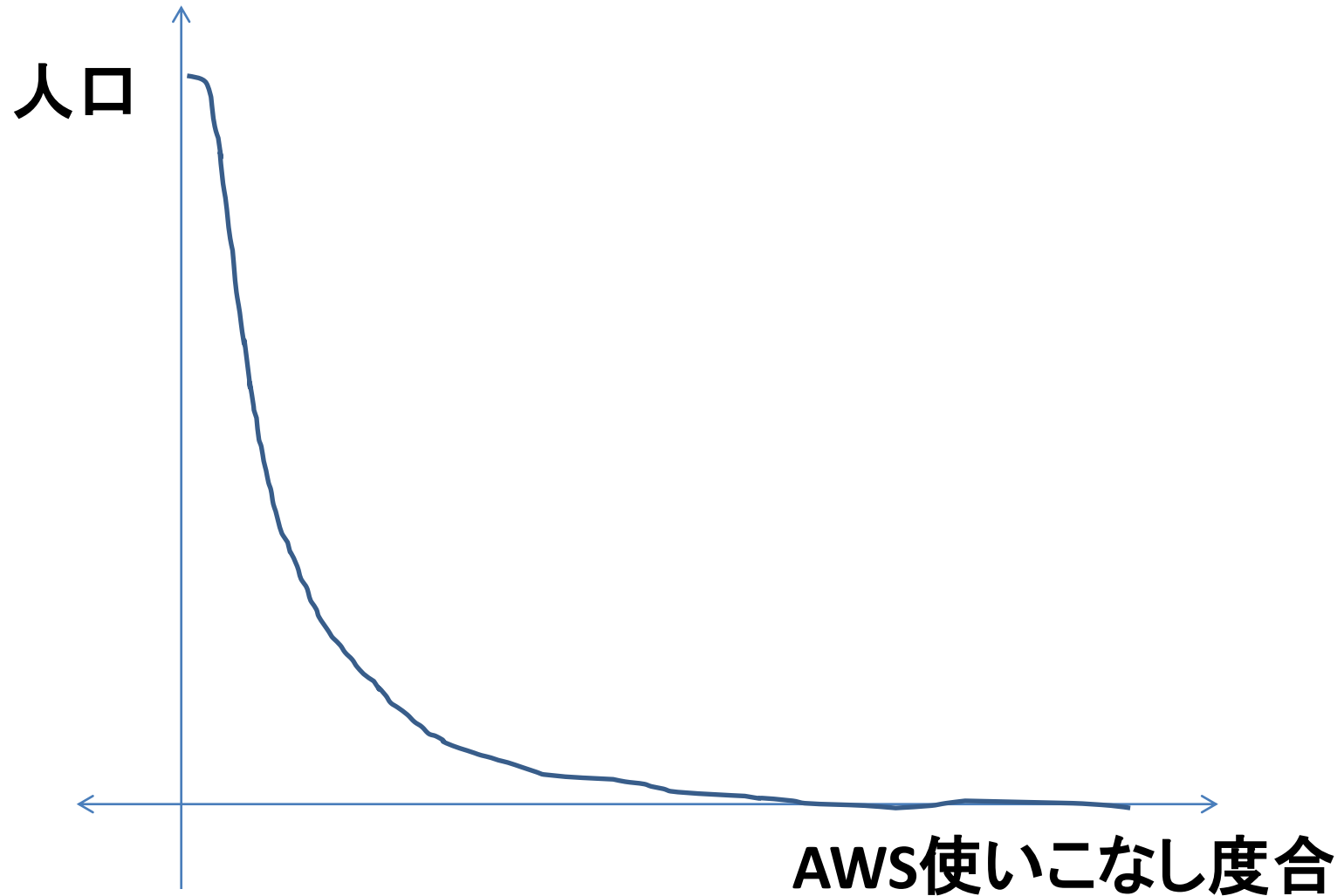
AWS使いは、こんなこと良く言う



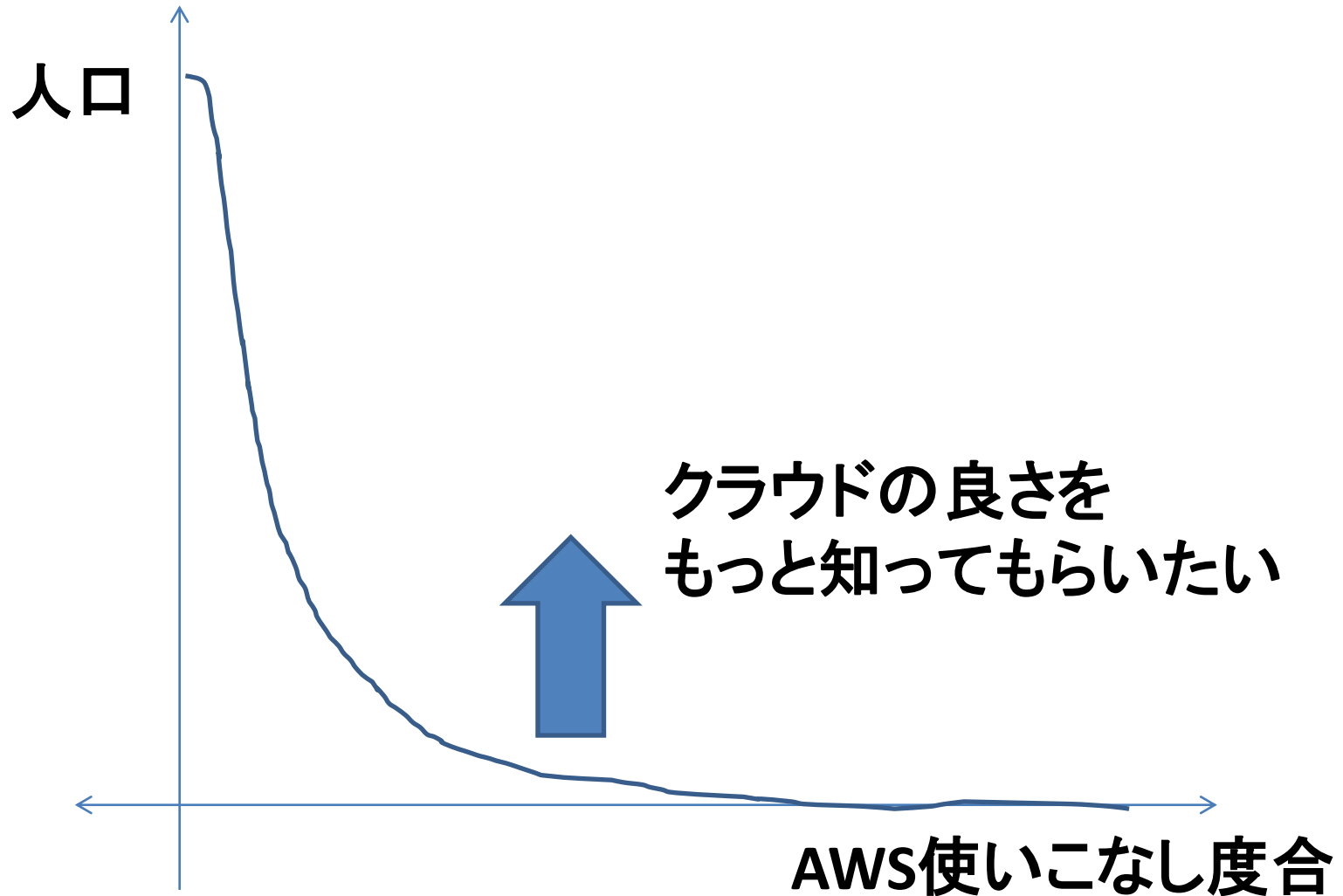
- ❖ 「万が一障害おきても、EIP付け替えればいいや。EBSスワップして一瞬で復帰って、すごい楽じゃないですか？」
- ❖ 「EC2の前にELBを立てて、マルチAZに振り分けよう。最悪マルチリージョンして、LBRすれば？」
- ❖ 「EC2にNFS入れて、エフアーメラルディスクにrsyncすれば同期も楽でしょ？」

ネタが細かすぎて 伝わらない 選手権

AWS使いこなし度合と人数分布



AWS使いこなし度合と人数分布



AWSクラウドを
より伝わりやすくしたい

既存ノウハウを伝えたい
(暗黙知→形式知化)

そこで
AWSクラウドデザインパターン
[beta]

(略して、CDP)

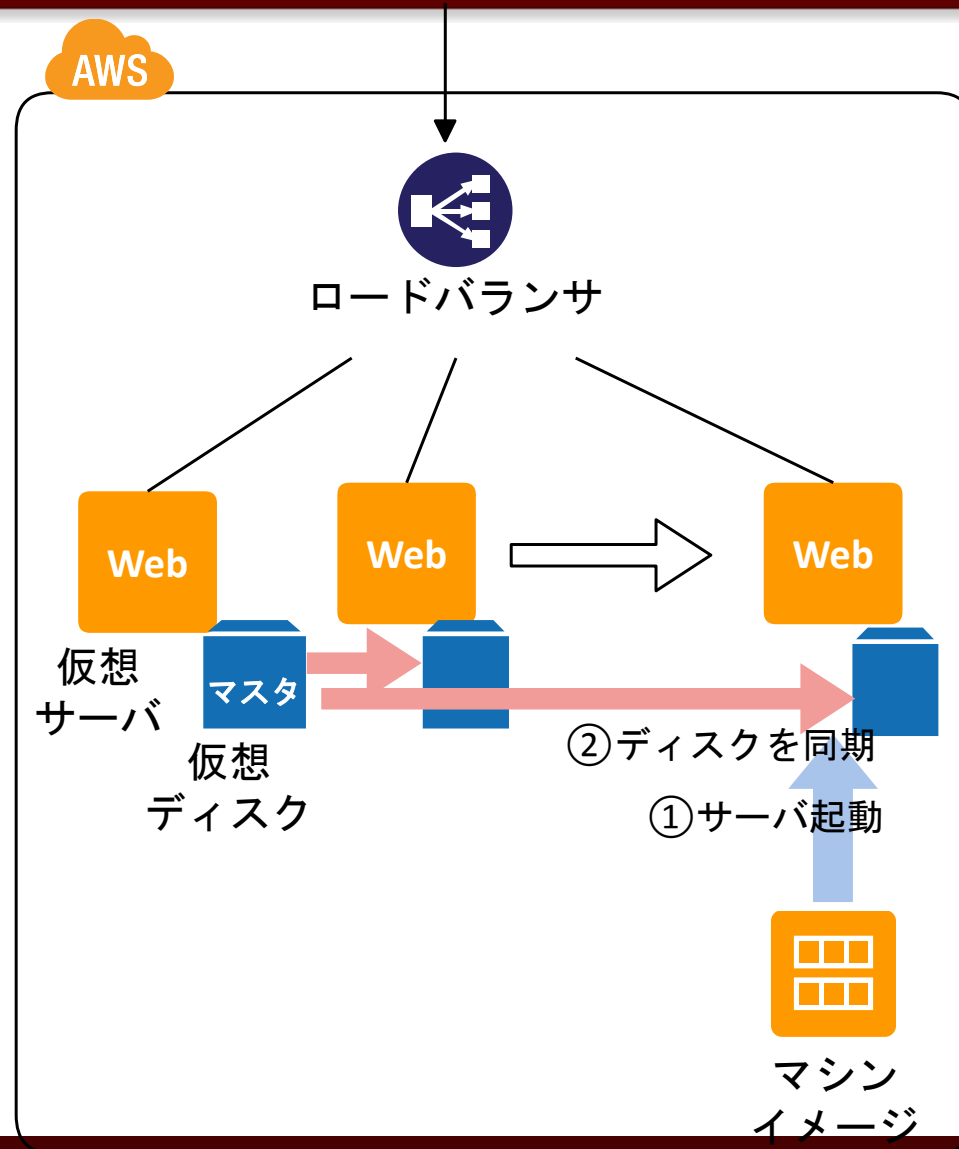
AWS クラウドデザインパターンとは



❖ AWSクラウドを使ったシステムアーキテクチャ設計を行う際に発生する、典型的な問題とそれに対する解決策・設計方法を、分かりやすく分類して、ノウハウとして利用できるように整理したものである。



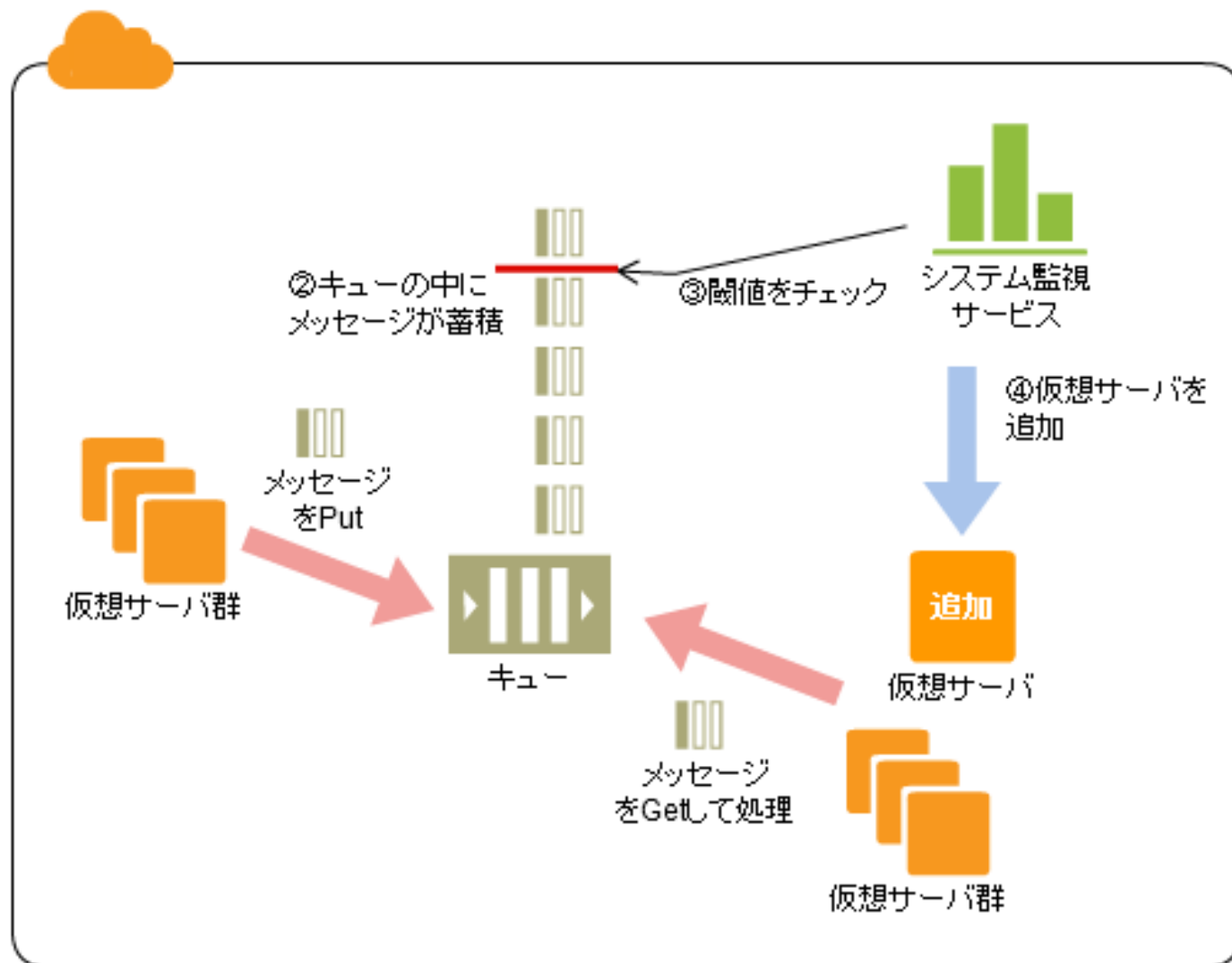
例: Clone Serverパターン



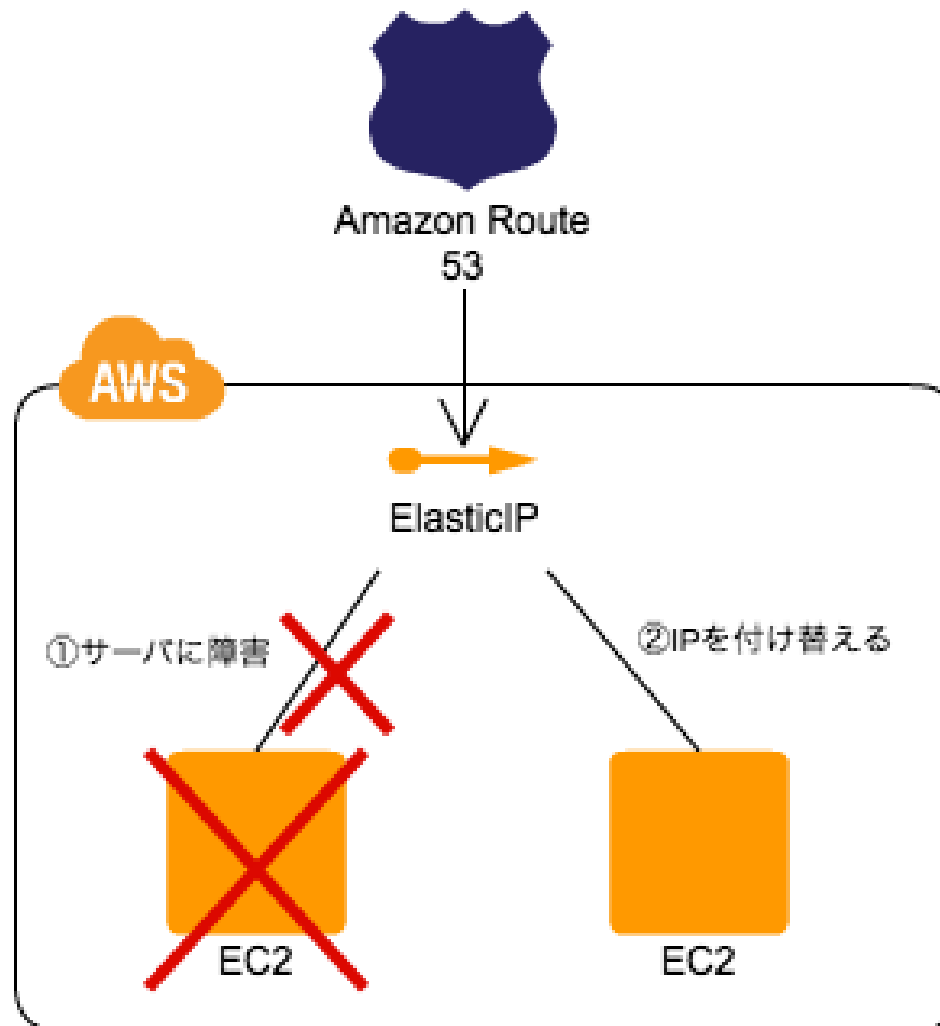
各パターン毎に下記を整理

- ❖ 解決したい課題
- ❖ 解決方法
- ❖ 実装
- ❖ 構造(図)
- ❖ 利点
- ❖ 注意点

例: Job Observerパターン



例：FloatingIPパターン



Floating AC Adaptor パターン



CDPカテゴリ (10月5日時点)



基本パターン

- Snapshot
- Stamp
- Scale Up
- OnDemand Disk

静的コンテンツを処理

- Web Storage
- Direct Hosting
- Private Distribution
- Cache Distribution

バッチ処理

- Queuing Chain
- Priority Queue
- Job Observer
- Scheduled Autoscaling

可用性を高める

- Multi-Server
- Multi-Datcenter
- Floating IP
- Deep Health Check

動的コンテンツを処理

- Scale Out
- Clone Server
- NFS Sharding
- NFS Replica
- State Sharing
- URL Rewriting
- Rewrite Proxy
- Cache Proxy
- Scheduled Scale Out

CDP48

保守

- Bootstrap
- Cloud DI
- Rollback Deployment
- Server Swapping
- Monitoring Integration
- Job Storage Archive
- Global Backup

ネットワーキング

- OnDemand NAT
- Backnet
- Functional Firewall
- Operational Firewall
- Multi Load Balancer
- WAF Proxy
- CloudHub

Wikiでノウハウを共有





案内

- [メインページ](#)
- [コミュニティ・ポータル](#)
- [最近の出来事](#)
- [最近の更新](#)
- [おまかせ表示](#)
- [ヘルプ](#)

ツールボックス

- [リンク元](#)
- [関連ページの更新状況](#)
- [特別ページ](#)
- [印刷用バージョン](#)
- [この版への固定リンク](#)

ログインまたはアカウント作成

ページ 議論

閲覧 編集 履歴表示

表示

検索

メインページ

AWSクラウドデザインパターンとは？ [編集]

AWSクラウドデザインパターン (AWS Cloud Design Pattern, 略してCDPと呼ぶ)とは、AWSクラウドを使ったシステムアーキテクチャ設計を行う際に発生する、典型的な問題とそれに対する解決策・設計方法を、分かりやすく分類して、ノウハウとして利用できるように整理したものである。

これまで多くのクラウドアーキテクト達が発見してきた、もしくは編み出した設計・運用のノウハウのうち、クラウド上で利用が可能なものをデザインパターンという形式で一覧化し、暗黙知から形式知に変換したものであるといえる。

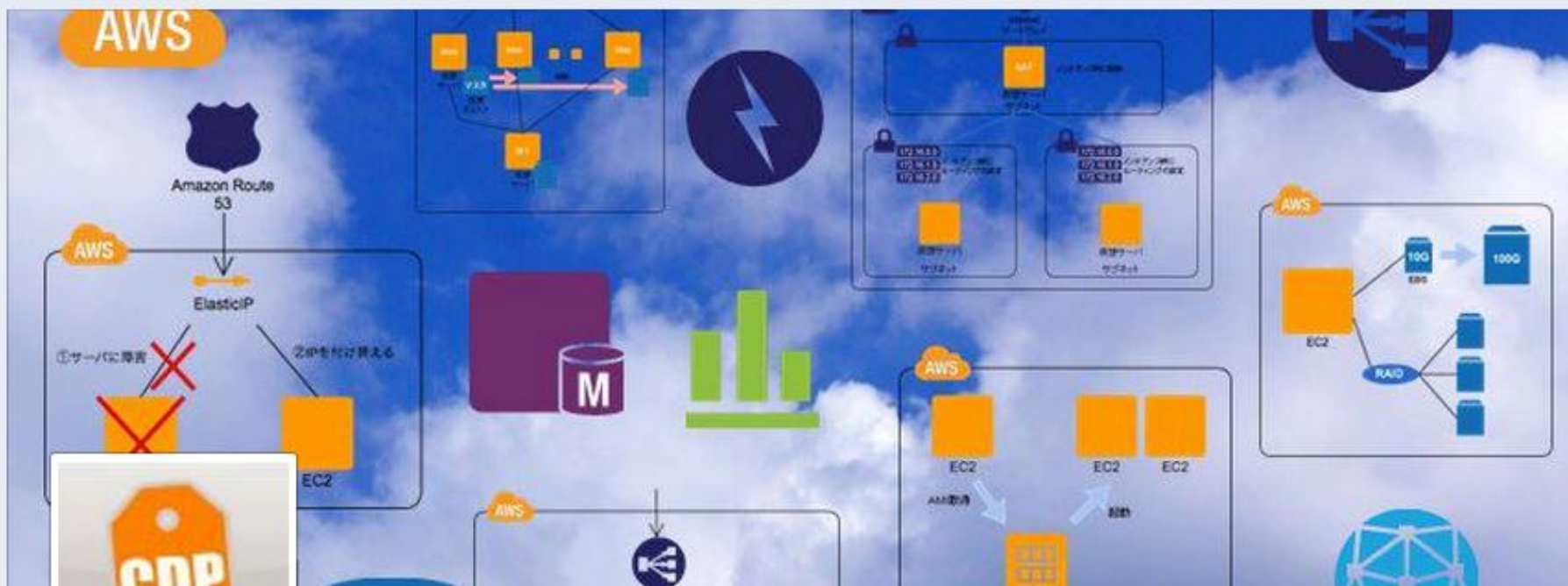
パターンの中には、クラウドでなくても実現できるもの、今まででも実現されていたものも含まれているが、クラウド上でも今まで通りのアーキテクチャが実現でき、かつクラウドを利用する事で、より安価にそしてより容易に実現できるものは、CDPとして収録することとした。

AWSクラウドデザインパターンの説明について [編集]

AWSクラウドデザインパターンは、以下の項目に沿って説明を行う。

- パターン名/サマリー
 - パターンの名前と、どのようなパターンなのかを簡単に説明したサマリー
- 解決したい課題
 - このパターンが解決する課題や、このパターンが適用できる課題、またパターンとして定義するに至った動機についての説明
- クラウドでの解決/パターンの説明
 - クラウド上でどのように課題を解決するかという点の説明や、どのようなパターンなのか、どのような構成のパターンとなっているかについての説明
- 実装

facebook.com/awscdp



クラウドデザインパターン (cdp)

いいね！ 997人・話題にしている人60人

✓ 「いいね！」しました

メッセージ

コンピュータ/インターネット

AWSクラウドデザインパターンのファンページです。AWSクラウドの設計・構築・運用のベ
 ースを形式知化するデザインパターンを考えていきます。

基本データ

1,009

Amazon Web Services クラウドデザインパターン 設計ガイド



クラウドデザインパターン設計ガイド



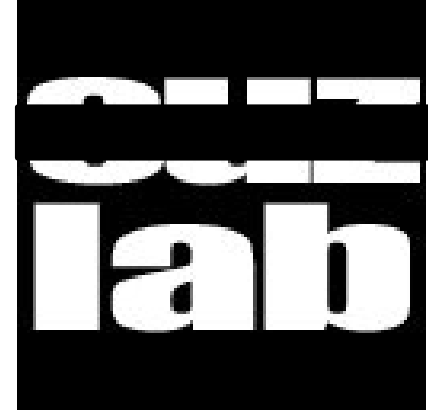
<http://www.amazon.co.jp/dp/4822211967/>

CDP、誰が作ったの？



Ninja of Three

Ninja of Three



表千家



裏千家

CDPは日本発！



**日本から
世界に発信
していこう！**

Amazon Web Services' first global customer and partner conference.

Deep technical content for your entire team: developers, architects, admins and executives.

Architecture

AWS Infrastructure Automation: Learn how to scale operations with AWS CloudFormation, Puppet and Chef

On Designing and Deploying Internet-Scale Services: Learn how to build services designed to run at Internet scale

HA Architecture at Netflix: How Netflix built a massively scalable fault tolerant service with AWS

Office Hours: Bring your architecture diagrams and discuss best practices with our knowledgeable solutions architects and evangelists

AWS Database Tier Architecture Best Practices: Understanding how to architect your database tier for the cloud

Extend Your Existing Datacenter to the Cloud with Amazon VPC: Learn the four most common AWS networking architectures

Intro to Chaos Monkey & the Simian Army: How Netflix is using automation to ensure HA architecture best practices

Architecting for High Availability & Multi-AZ on AWS: Learn how to architect your apps for high-availability, multiple availability zones and fault tolerance

The Convergence of IaaS and PaaS: Come hear from RightScale about how to architect your solutions to leverage a wide variety of cloud services and what it means for your customers and your business

Cloud Design Patterns: Learn repeatable solutions to commonly occurring challenges in cloud architectures



Jeff Bezos
Founder & CEO
Amazon



Andy Jassy
Sr. Vice President
Amazon Web Services



Werner Vogels
CTO
Amazon



Reed Hastings
CEO
Netflix



Thorsten von Eicken
GTO & Founder
RightScale



Matt Tavis
Solutions Architect
Amazon Web Services



James Hamilton
VP & Distinguished Engineer
Amazon Web Services



Jinesh Varia
Technical Evangelist
Amazon Web Services



Carlos Conde
Principal Solutions Architect
Amazon Web Services



Ariel Tseitlin
Director of Cloud Solutions
Netflix



Adrian Cockcroft
Sr. Manager, Cloud Architect
Netflix



Attila Narin
Sr. Manager, Solutions Architect
Amazon Web Services



Matt Wood
Product Manager, Data-Intensive Computing
Amazon Web Services



Steve Morad
Solutions Architect
Amazon Web Services



Ken Tamagawa
Manager, Solutions Architect
Amazon Web Services

AWSの初のグローバルカンファレンス
11月27-29日@ラスベガス！

パターンを活用した実装シナリオ



❖画像動画配信サイト

- ❖大量のユーザに配信したい

❖Eコマースサイト

- ❖可用性、耐障害性高くしたい

❖キャンペーンサイト

- ❖突発的アクセス増加に耐えたい

❖他にも多数。。



JAWS-UG
AWS User Group - Japan

AWSクラウドデザインパターン -コンテンツ配信編-



このシナリオの背景

- ❖ 雲の写真を載せるブログサイト開始
- ❖ はじめは個人的に開始



この実装シナリオの狙い



- ❖ コンテンツ配信サイトをとりあげ、
 - ❖ 運用効率
 - ❖ コスト効果
- ❖ を高めるパターンを中心にAWSを使用した実装方法を解説

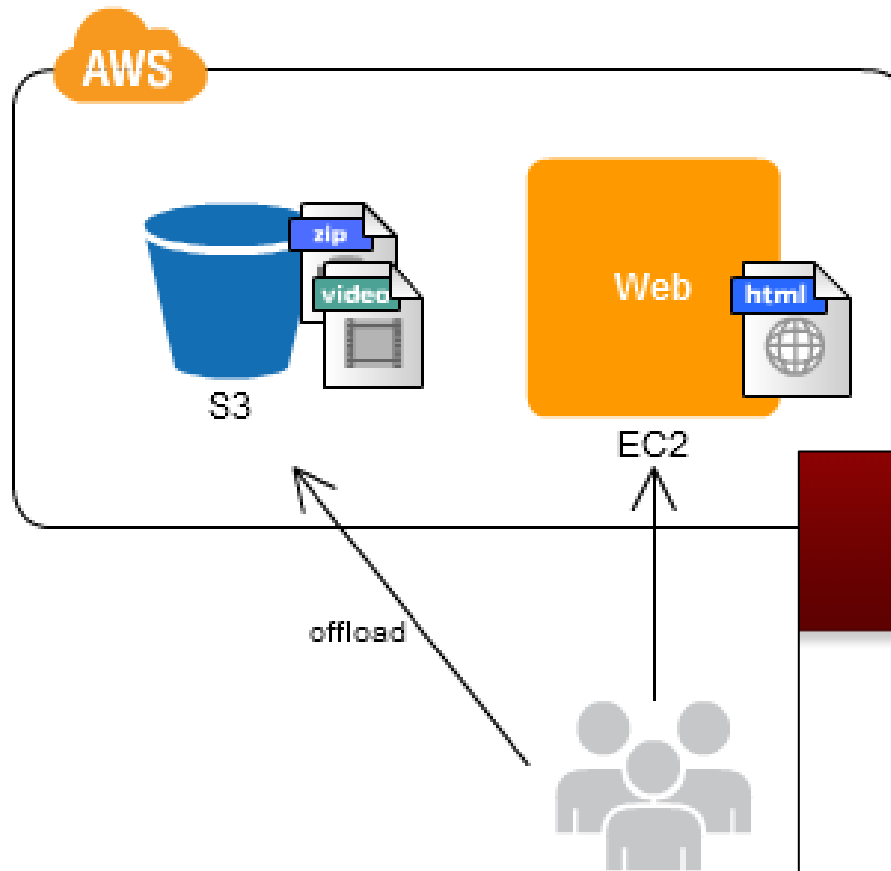
- ❖ Amazon Linux (64bit)
- ❖ Movable Type 5.12 (MTOS)
- ❖ Apache HTTP Server 2.2.21
- ❖ MySQL 5.1.52
- ❖ Perl 5.10.1

❖課題：

ユーザー要望により動画を公開

❖サイズ大きい、容量読めない

Web Storageパターン



Cloud Design Pattern I

雲動画ははじめました

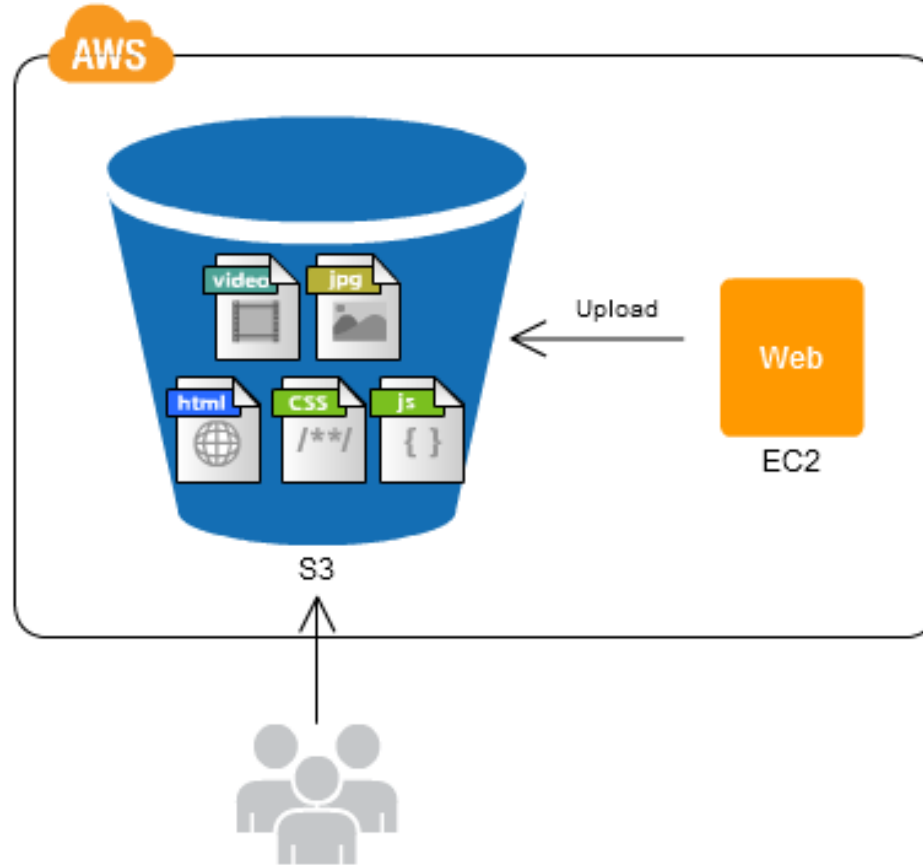
mtuser (2011年12月30日 22:15) | [コメント\(0\)](#) | [トラックバック\(0\)](#)

ダウンロードは[こちら](#)から。

❖ 雲マニア御用達のサイトへ

❖ アクセス増により、
アクセス出来ない状況に

Direct Hostingパターン



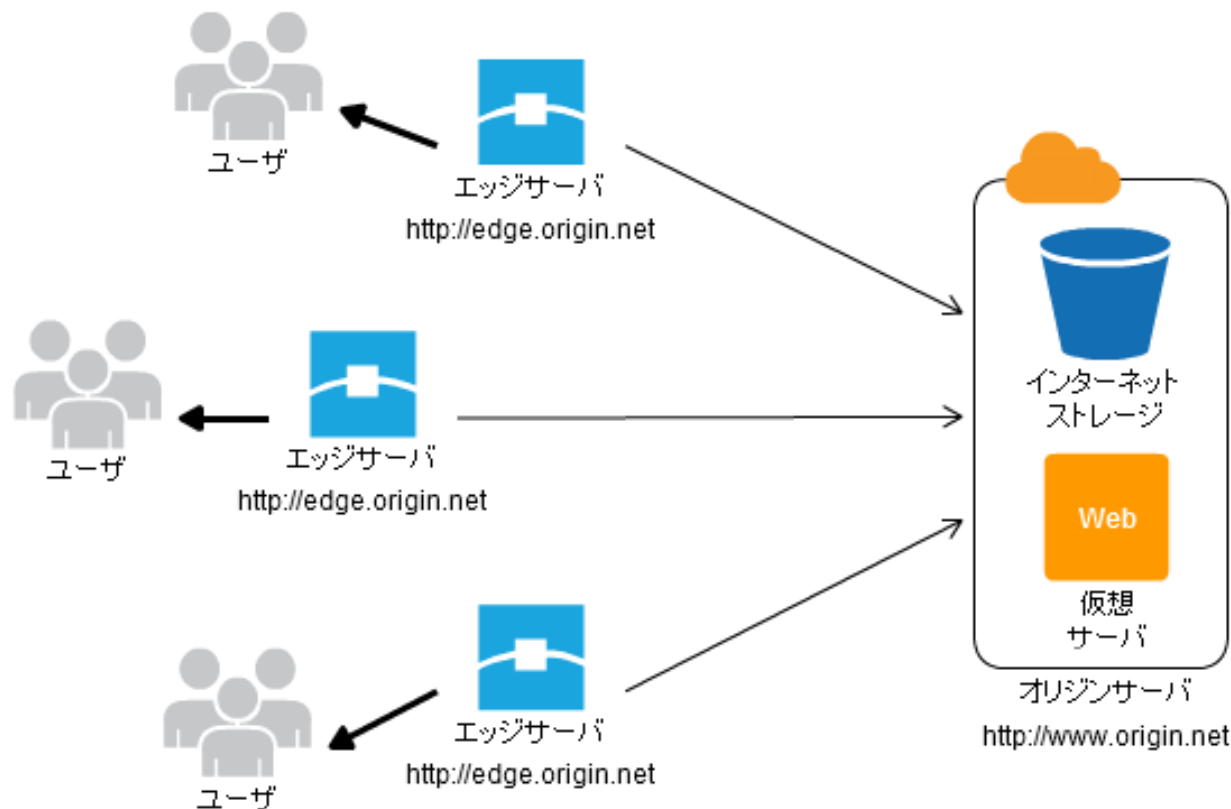
まさかの海外展開



- ❖ 海外有名ニュースサイトへの掲載が決定
- ❖ 掲載までに、
海外からのアクセスへの対応が必要

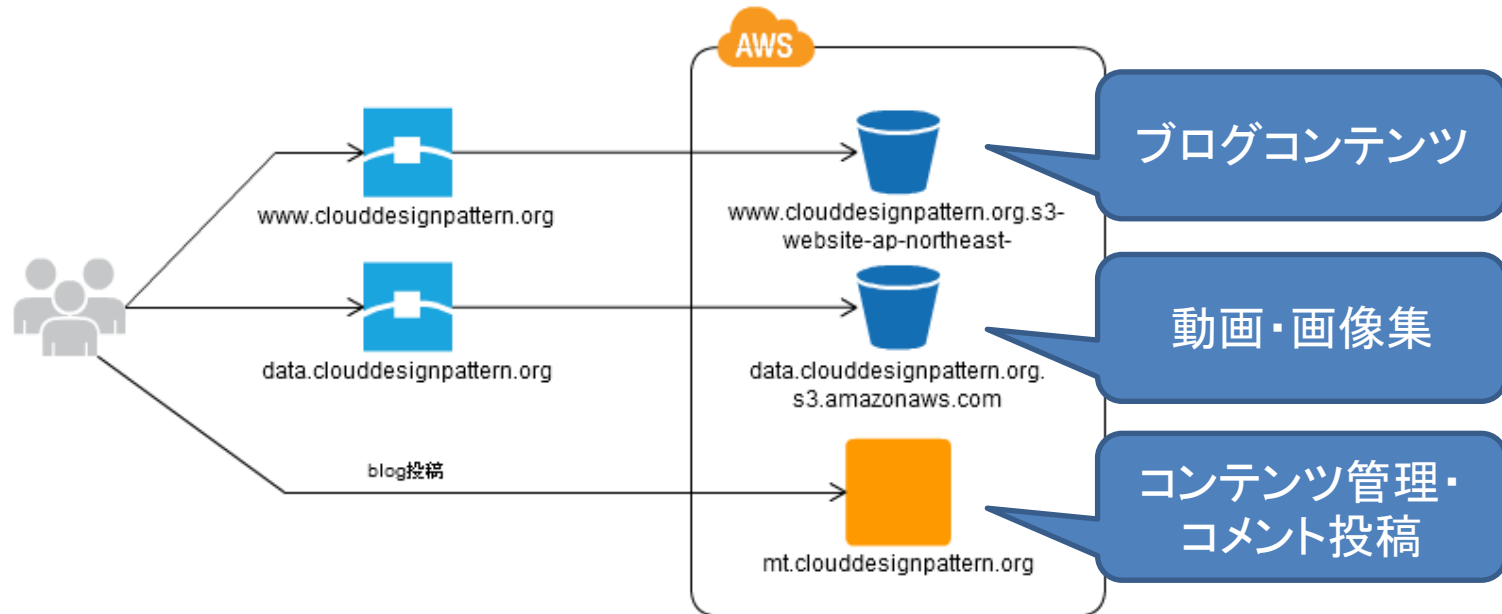
A screenshot of the New York Times website from March 2, 2012. The browser address bar shows 'www.nytimes.com'. The page features the 'The New York Times' masthead and a navigation bar with links like 'HOME PAGE', 'TODAY'S PAPER', 'VIDEO', and 'MOST POPULAR'. The main content area displays a large article titled 'Syria Bars Red Cross Convoy From Fallen Rebel Bastion' by Neil MacFarquhar and Alan Cowell. To the right of this article, there is a red banner for 'Cloud Design Pattern' with a link to 'Cloud Design Pattern Blog'. Below this, another link for 'Cloud5 - Cloud Design Pattern Blog' is visible. The right sidebar contains a 'Sunday Review' section, a 'MARKETS' section with stock prices for S&P 500, Dow, and Nasdaq, and a 'GET QUOTES' section. The bottom of the page shows a small 'NoT' logo.

Cache Distributionパターン

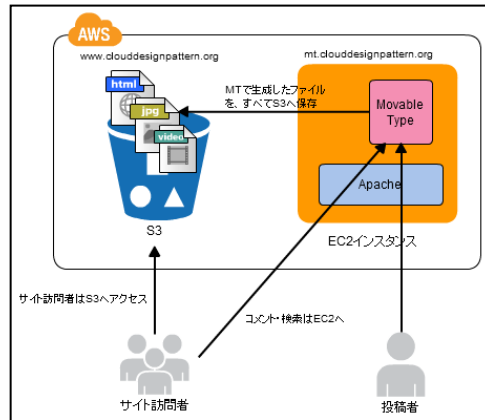
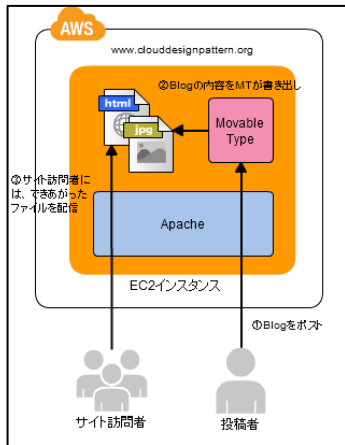


Amazon CloudFront

最終的に



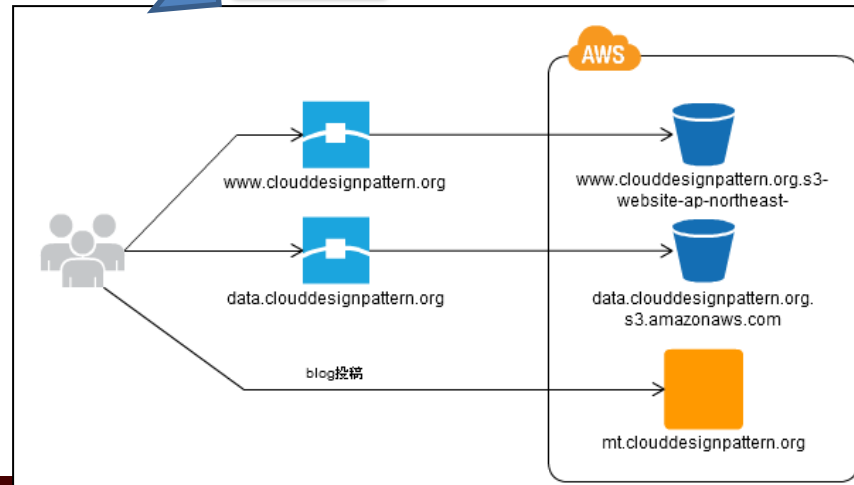
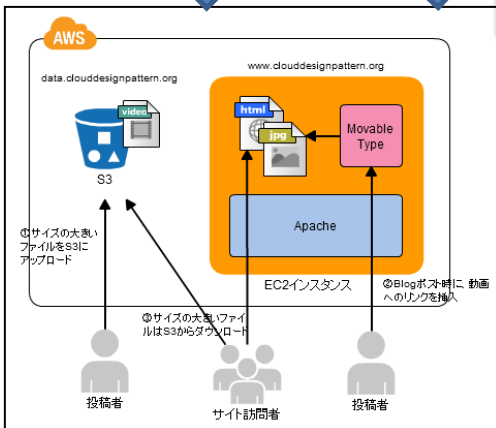
デザイン推移



動画

人気

海外





JAWS-UG
AWS User Group - Japan

AWSクラウドデザインパターン -Eコマース編-



このシナリオの背景



❖まさかの 雲グッズ販売サイト開始



この実装シナリオの狙い



- ❖Eコマースサイトを取りあげ、
 - ❖可用性
 - ❖耐障害性
- ❖を高めるパターンを中心にAWSを使用した実装方法を解説

- ❖ EC-CUBEバージョン 2.11.4
- ❖ Amazon Linux (64bit)
- ❖ PHPバージョン PHP 5.3.8
- ❖ DBバージョン MySQL 5.1.52

EC-CUBEは日本発のオープンソースECサイト構築システムです。

EC CUBE ONLINE SHOPPING SITE

☐ 記憶[ログイン](#)[パスワードを忘れた方](#)[MYページ](#)[会員登録](#)[カゴの中を見る](#)

■ 現在のカゴの中

合計数量: 0
商品金額: 0円

[カゴの中を見る](#)

■ 商品カテゴリー

[▶ 雑貨 \(2\)](#)[▶ 食品 \(3\)](#)[▶ レシピ \(1\)](#)[当サイトについて](#)[お問い合わせ](#)[特定商取引に関する表記](#)[プライバシーポリシー](#)

Cloud コマース

プロカメラマンの雲写真
撮影現場から直送！

[→ 詳細はこちら](#)

¥78～

■ おすすめ商品情報



デザイナー写真: 空と雲
販売価格(税込): 104 円
プロのカメラマンが撮った写真です



デザイナー写真: 青空
販売価格(税込): 84 円
デザイナーが加工した写真です

■ ログイン

>> メールアドレス

☐ コンピューターに記憶する

>> パスワード

[パスワードを忘れた方はこちら](#)

[ログイン](#)

■ 検索条件

>> 商品カテゴリから選ぶ

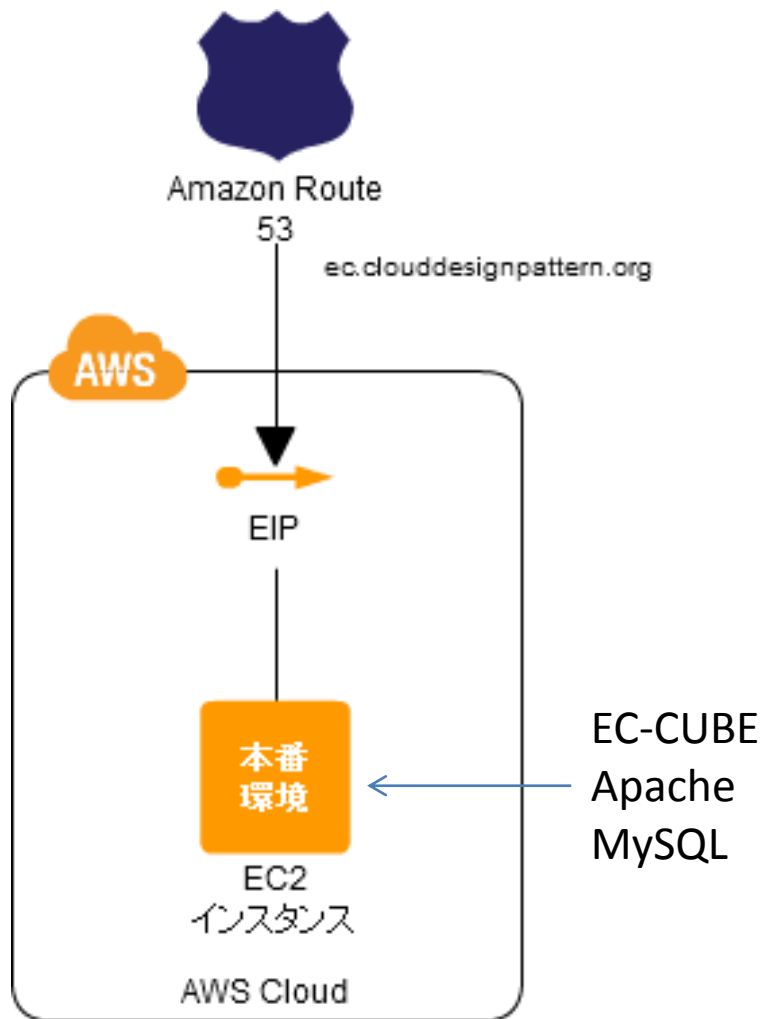
全ての商品

>> 商品名を入力

[検索](#)

■ 営業カレンダー

初期のデザイン



❖課題：
利用ソフトのバージョンアップが必要

Floating IPパターンの適用後



Amazon
Route 53

ec.clouddesignpattern.org

AWS



EIP

EIP 「46.51.xxx.xxx」

④EIPを付け替える

本番
環境

EC2

テスト
環境

EC2

③ソフトウェアをアップデー
トしテストを行う

①AMIを
取得する

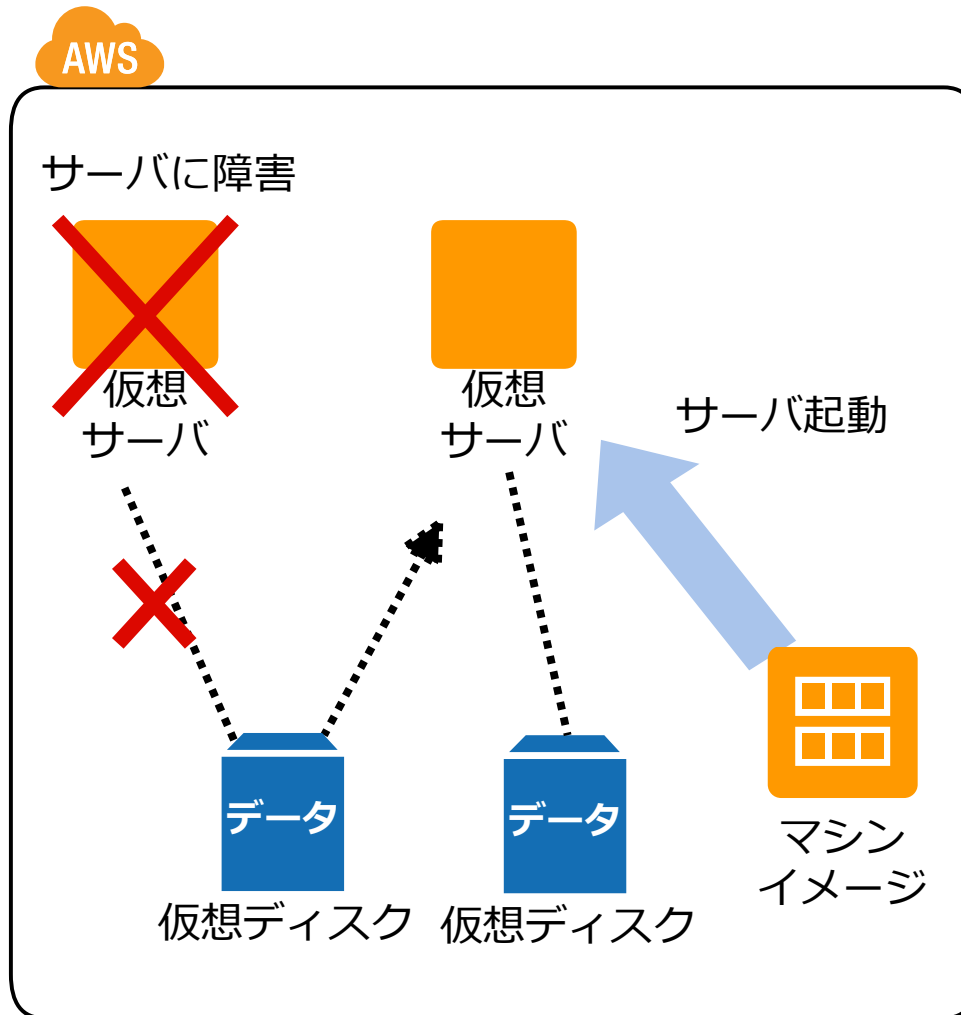
②テスト環境用の
EC2インスタンスを
立ち上げる



EC2 AMI

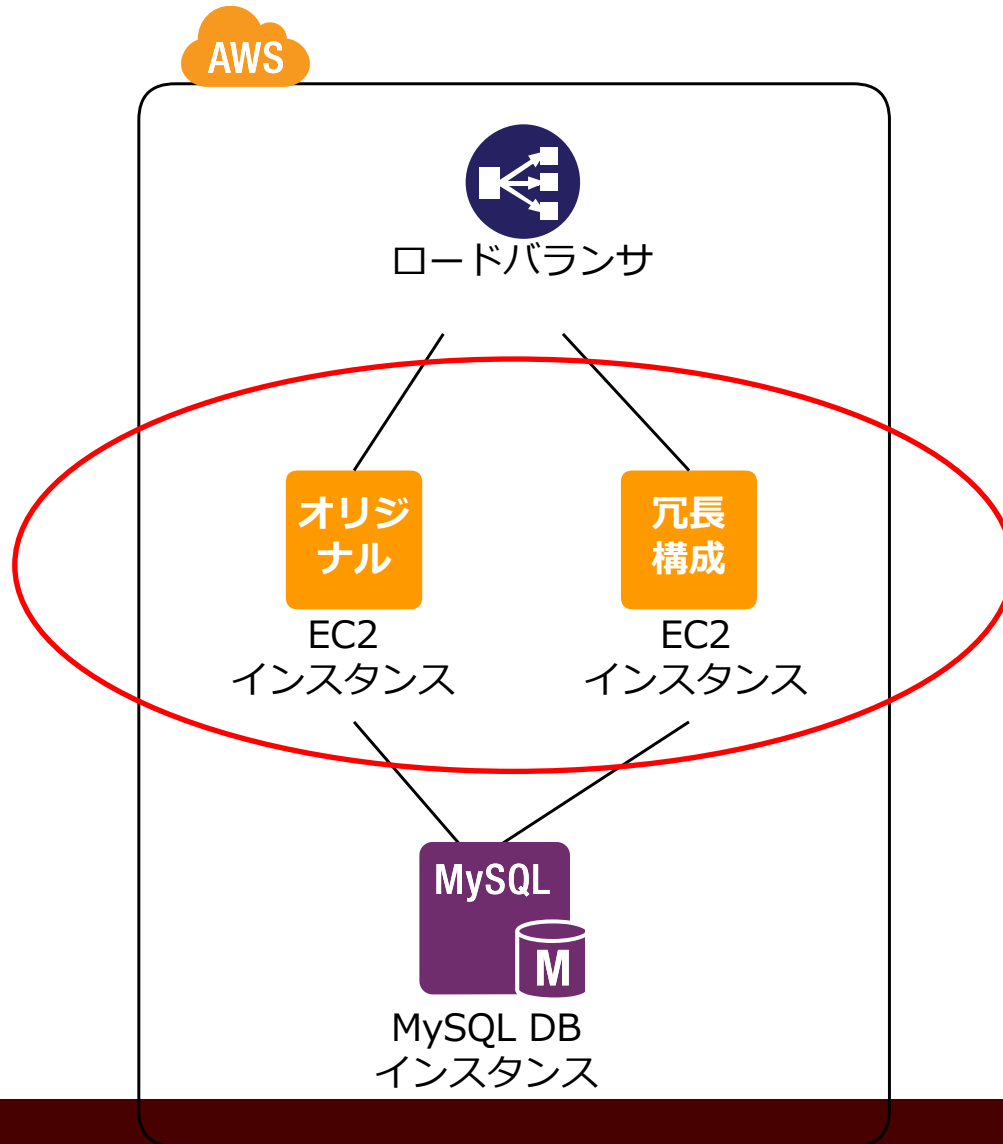
❖課題： サーバに障害が発生して、
速やかに復旧したい

Server Swappingパターンの適用後



❖課題： Webサーバが落ちても、システム全体で稼働し続けるようにしたい

Multi-Serverパターン



RDSの作成

Launch DB Instance Wizard

Cancel

ENGINE SELECTION

DB INSTANCE DETAILS

ADDITIONAL CONFIGURATION

MANAGEMENT OPTIONS

REVIEW

To get started, choose a DB Instance engine and class below

Engine: mysql

License Model: general-public-license

DB Engine Version: 5.5.12

DB Instance Class: db.m1.small

Multi-AZ Deployment: No

Auto Minor Version Upgrade: ☒ Yes ☐ No

Provide the details for your RDS Database Instance.

Allocated Storage:* 10 GB (Minimum: 5 GB, Maximum 1024 GB)

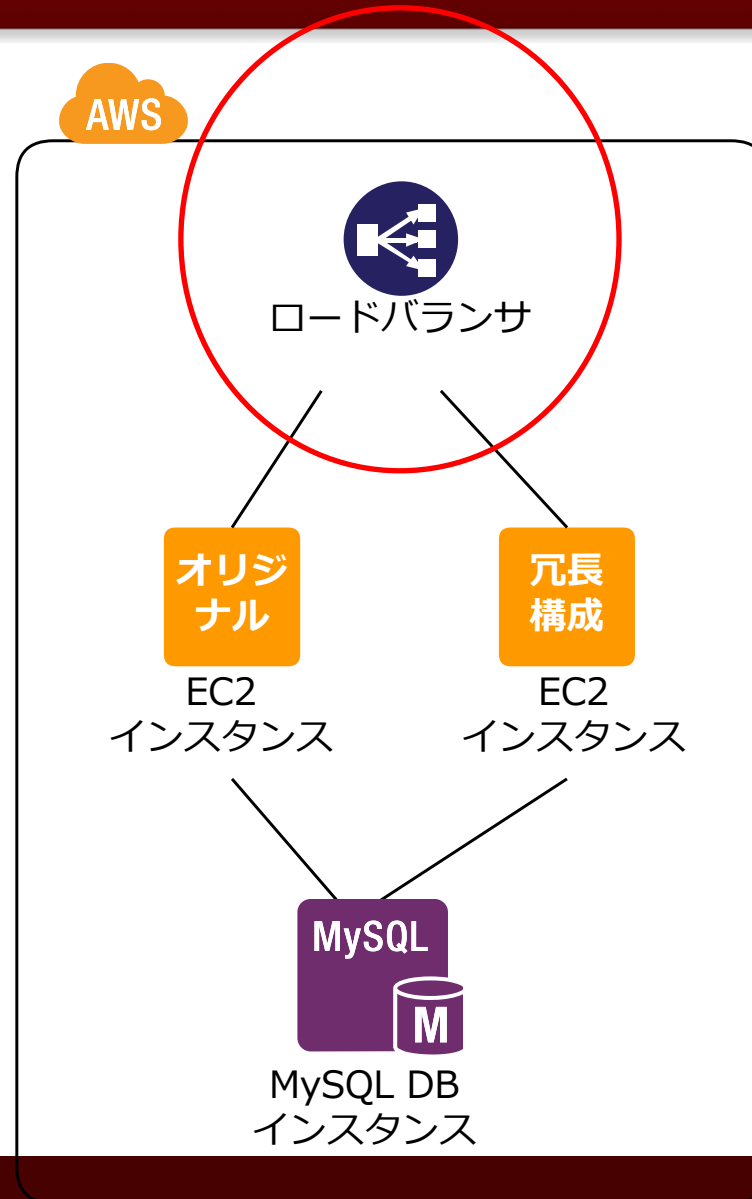
DB Instance Identifier:* eccubedbinstance (e.g. mydbinstance)

Master User Name:* eccube_db_user (e.g. awsuser)

Master User Password:* (e.g. mypassword)

< BackContinue

Multi-Serverパターンの適用後



ロードバランサの起動



AWS Management Console > Amazon EC2

Ken Tamagawa ▼

AVVS
Elastic Beanstalk

Amazon
S3

Amazon
EC2

Amazon
VPC

Amazon
CloudWatch

Amazon
Elastic MapReduce

Amazon
CloudFront

AVVS
CloudFormation

Amazon
RDS

Amazon
ElastiCache

Amazon
SQS

AVVS
IAM

Amazon
SNS

Amazon
SES

Amazon
Route

Navigation

Region:
Asia Pacific (Tokyo) ▼

EC2 Dashboard
Scheduled Events

INSTANCES
Instances
Spot Requests
Reserved Instances

IMAGES
AMIs
Bundle Tasks

ELASTIC BLOCK STORE
Volumes
Snapshots

NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces

Load Balancers

Create Load Balancer Delete

Show/Hide Refresh

	Load Balancer Name	DNS Name	Port Configuration
☐	MyLoadBalancer	MyLoadBalancer-633431090.a	80 (HTTP) forwarding to 80 (HTTP)
☒	mylb2	mylb2-1360775126.ap-northeast-1.elb.amazonaws.com	80 (HTTP) forwarding to 80 (HTTP), 443 (TCP)

1 Load Balancer selected

Load Balancer: mylb2

Description Instances Health Check Security

DNS Name:
mylb2-1360775126.ap-northeast-1.elb.amazonaws.com (A Record)
ipv6.mylb2-1360775126.ap-northeast-1.elb.amazonaws.com (AAAA Record)
dualstack.mylb2-1360775126.ap-northeast-1.elb.amazonaws.com (A or AAAA Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your LoadBalancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see the [Using Domain Names With Elastic Load Balancing](#)

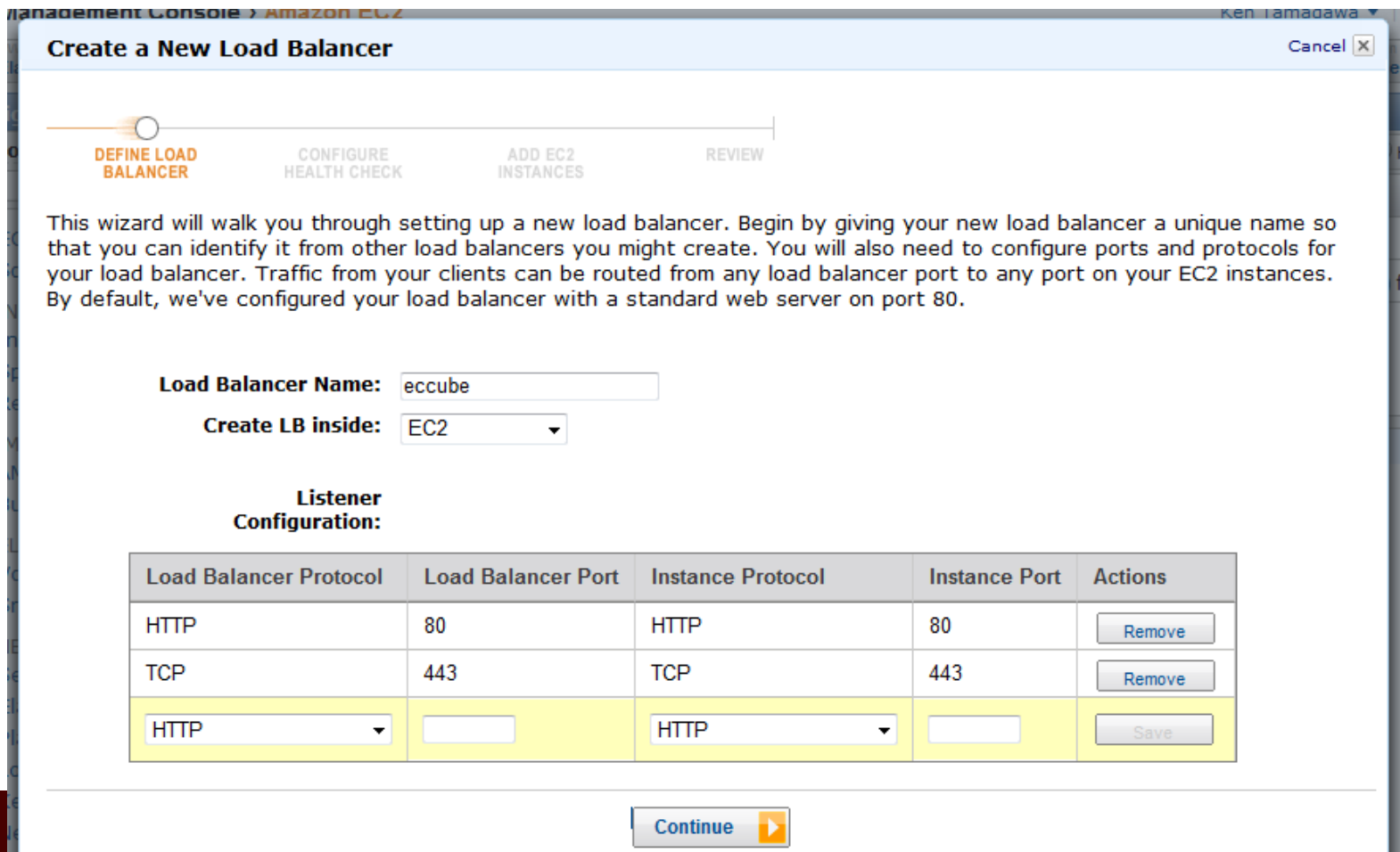
Status: 1 of 1 instances in service



ロードバランサの起動

EC-CUBEでは、SSLをサポート。

ELBでも対処可能だが、今回はELBではSSLの処理はしないことに。



Create a New Load Balancer Cancel X

DEFINE LOAD BALANCER CONFIGURE HEALTH CHECK ADD EC2 INSTANCES REVIEW

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer Name:

Create LB inside:

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Actions
HTTP	80	HTTP	80	Remove
TCP	443	TCP	443	Remove
HTTP		HTTP		Save

[Continue](#)

ELB配下にEC2を追加

AMIからEC2起動

	Name	Instance	Type	State
☐	EC-CUBE	 i-89ea2589	m1.large	 running
☒	ECCUBE-MultiServer	 i-8f4e818f	m1.large	 running

ELB配下に追加

Add and Remove Instances

The table below lists all your running EC2 Instances that are not already behind another load balancer or part of an auto-scaling capacity group. Check the boxes in the Select column to add those instances to this load balancer. If you select instances from an availability zone that isn't already part of the load balancer, we will add it for you.

Manually Add Instances to Load Balancer:

Select	Instance	Name	State	Security Groups
☐	i-89ea2589	EC-CUBE	 running	Lab Web Tier
☒	i-89ea2589	EC-CUBE	 running	Lab Web Tier
☒	i-8f4e818f	ECCUBE-MultiServer	 running	Lab Web Tier

[select all](#) | [select none](#)

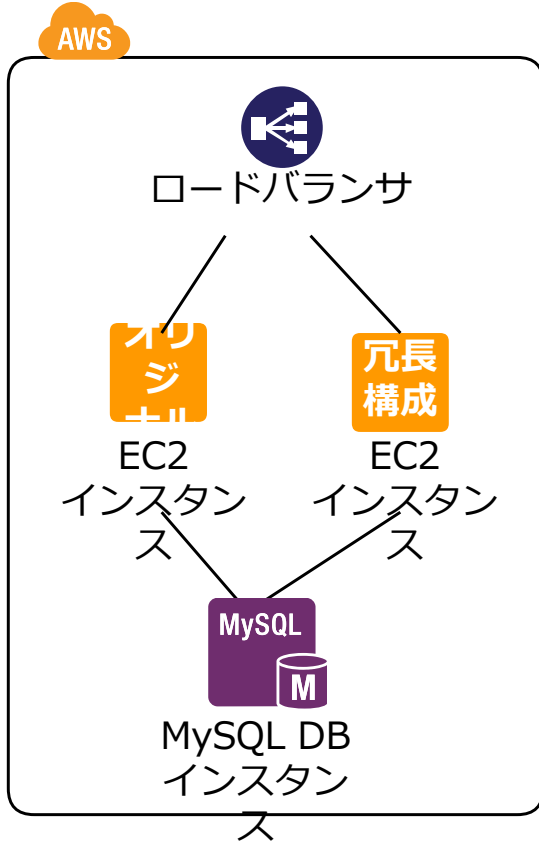
Availability Zone Distribution:

0 instances in
ap-northeast-1b

2 instances in
ap-northeast-1a

[Save](#)

Multi-Serverパターンの適用後



サーバーわざと
停止させても動く??

 **Load Balancer:** eccube

Description Instances Health Check Security

Instances			
Instance	Availability Zone	Status	Actions
i-89ea2589 (EC-CUBE)	ap-northeast-1a	In Service	Remove from Load Balancer
i-8f4e818f (ECCUBE-MultiServer)	ap-northeast-1a	Out of Service (why?)	Remove from Load Balancer

- ❖ 課題： DB部分のSPOFを解消したい
- ❖ DB Replicationパターンを適用し、DBをマルチ構成に切り替える

マルチAZに変更

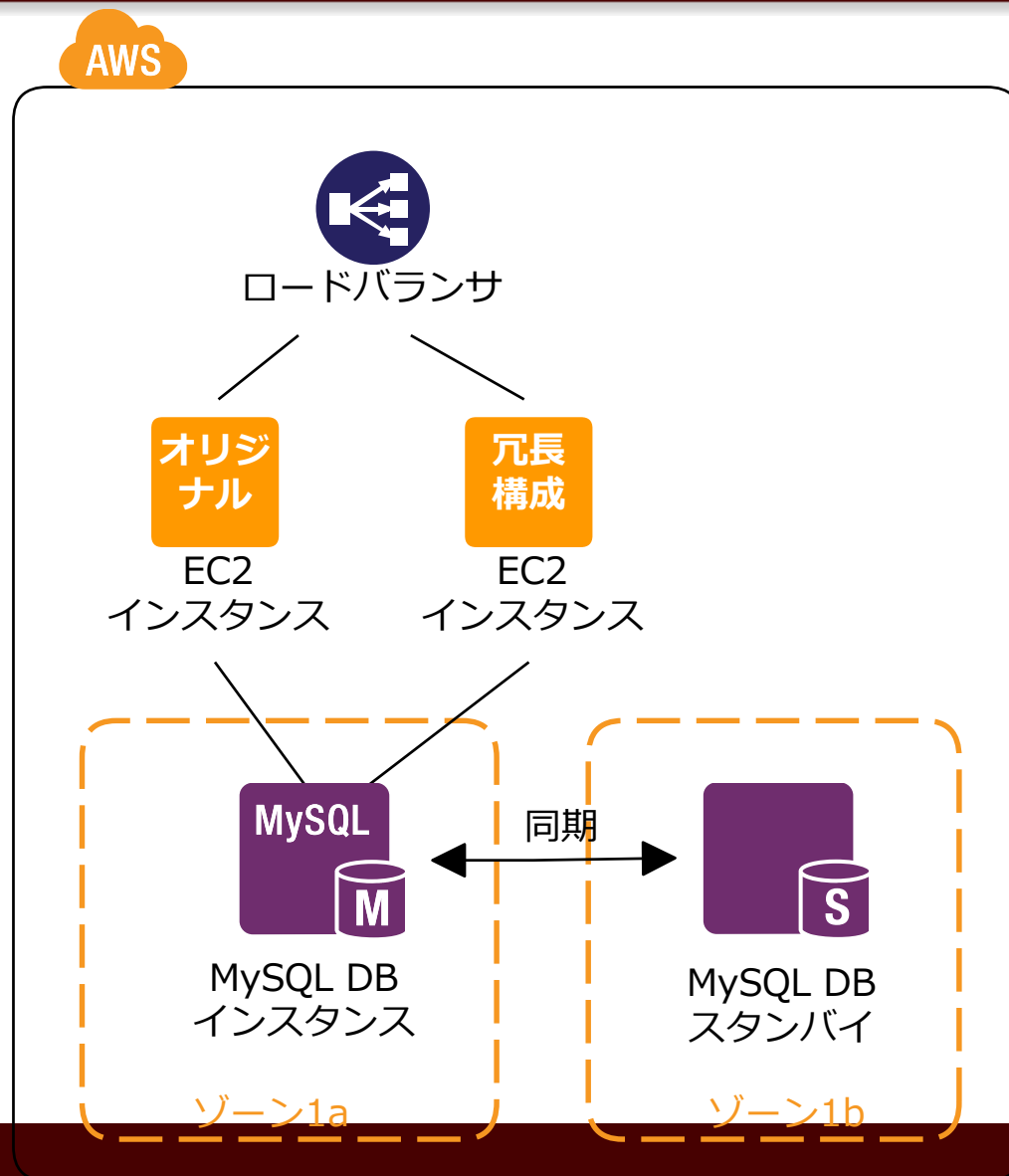
Modify DB Instance: eccubedbinstance Cancel

DB Instance Class:	db.m1.large	DB Parameter Group:	default.mysql5.5
DB Engine Version:	5.5.12	DB Security Groups:	default mydbsecuritygroup webtier
Multi-AZ Deployment:	Yes	Master User Password:	
Auto Minor Version Upgrade:	☒ Yes ☐ No		
Allocated Storage:	20 GB		
Backup Retention Period:	8 days		
Backup Window:	17:30 - 18:00 UTC		
Maintenance Window:	Sunday 18:00 - Sunday 18:30 UTC		
Apply Immediately:	☒		

 Note: choosing to change your DB Instance Class and/or Backup Retention Period along with selecting "Apply Immediately" causes your DB Instance to reboot immediately.

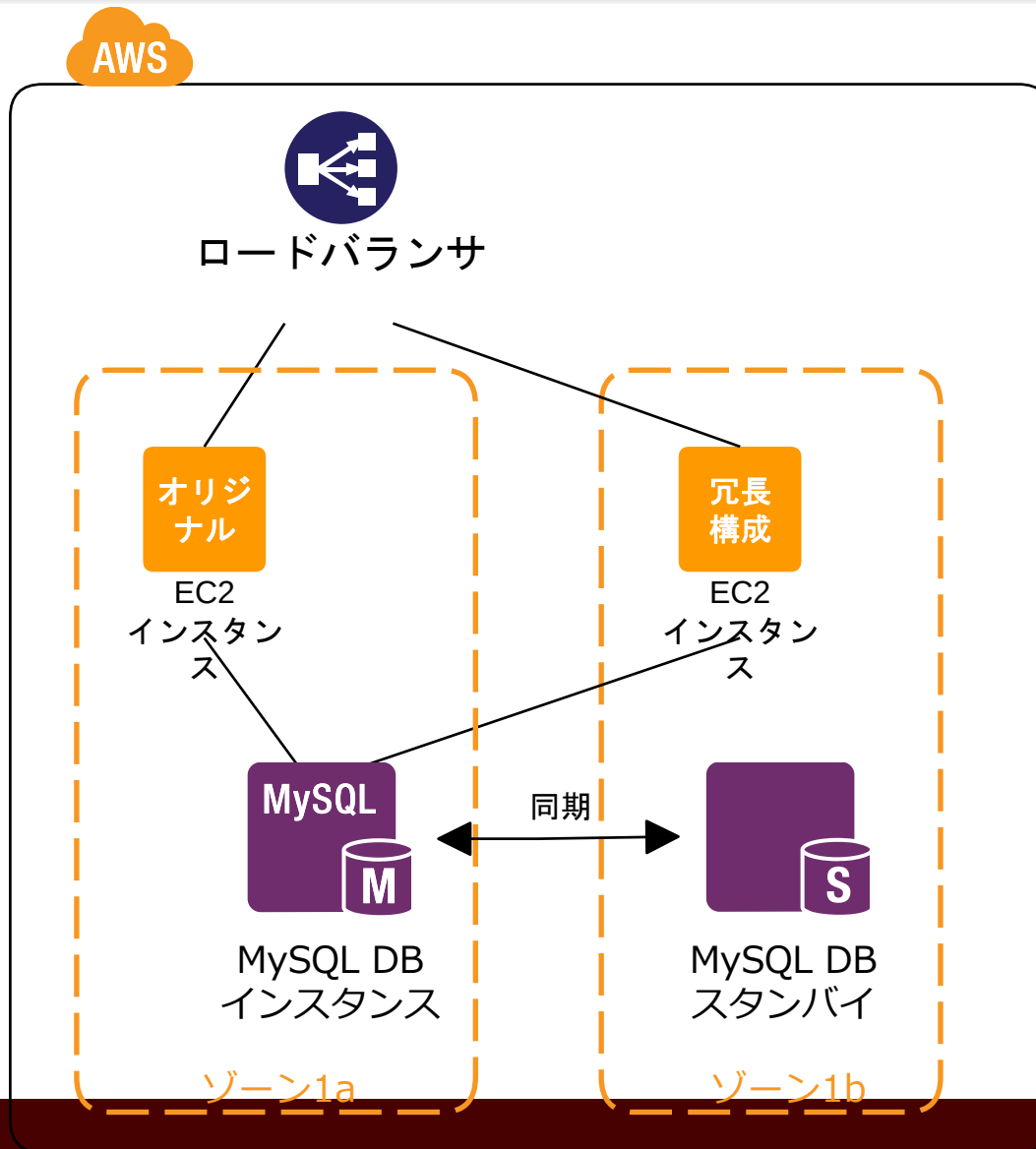
Cancel Yes, Modify

DB Replicationパターンの適用後

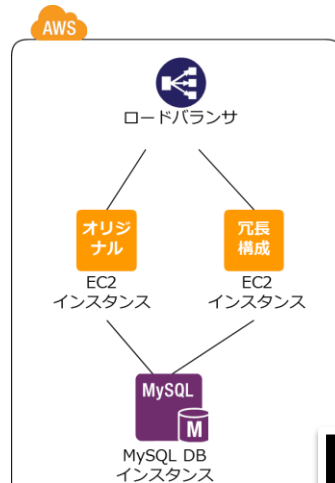
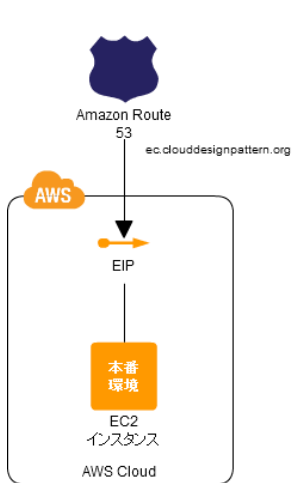


- ❖ 課題： サーバレベルだけでなく、データセンタレベルの障害にも対応できるようにする
- ❖ Multi-Datacenterパターンを適用し、すべてのレイヤで冗長化を行う

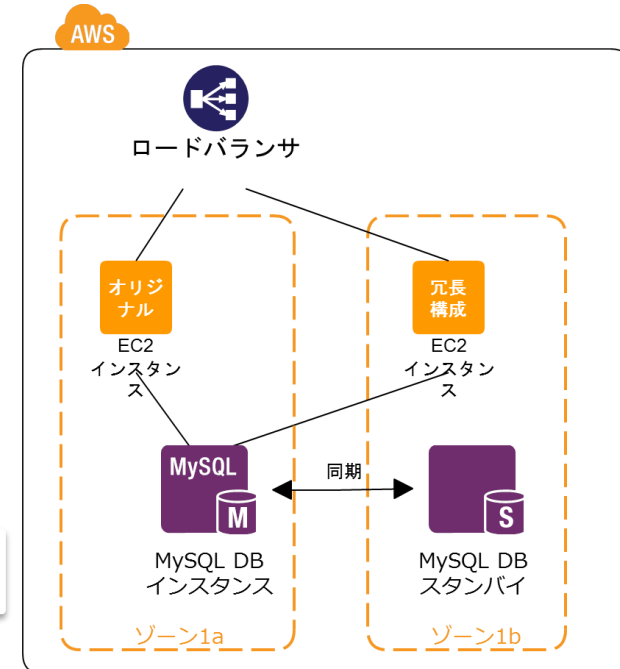
Multi-Datacenterパターンの適用後



デザイン推移

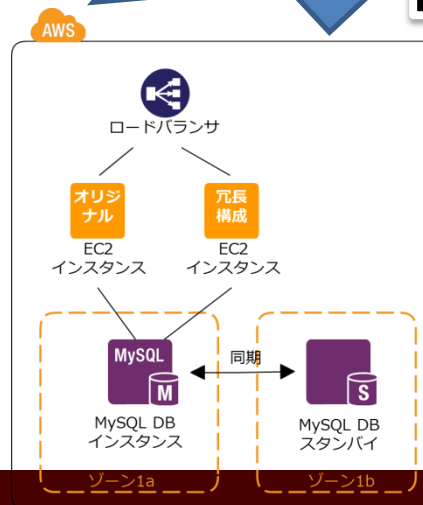
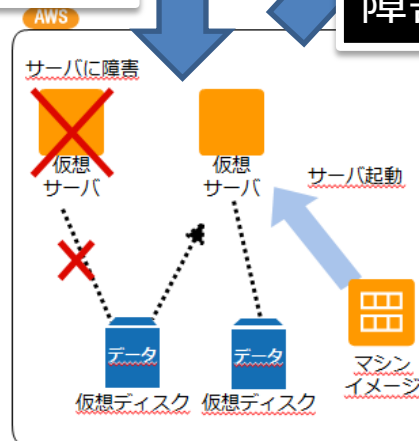


SPOF回避



最終

障害対策





JAWS-UG
AWS User Group - Japan

AWSクラウドデザインパターン -キャンペーンサイト編-



❖ECサイトの商品が大人気

❖さらなる販促のためキャンペーンサイト



- ❖ Amazon **L**inux (32bit)
- ❖ **A**pache HTTP Server 2.2.21
- ❖ **M**ySQL 5.1.52
- ❖ **P**HP 5.3.8
- ❖ WordPress 3.3.1

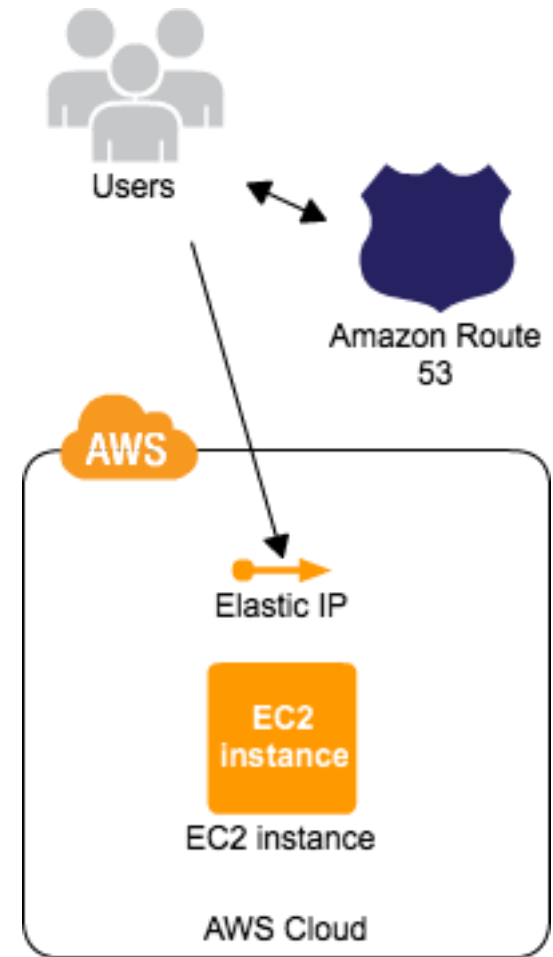


キャンペーンサイト開設



❖ スモールスタート

❖ EC2は1インスタンスで運用



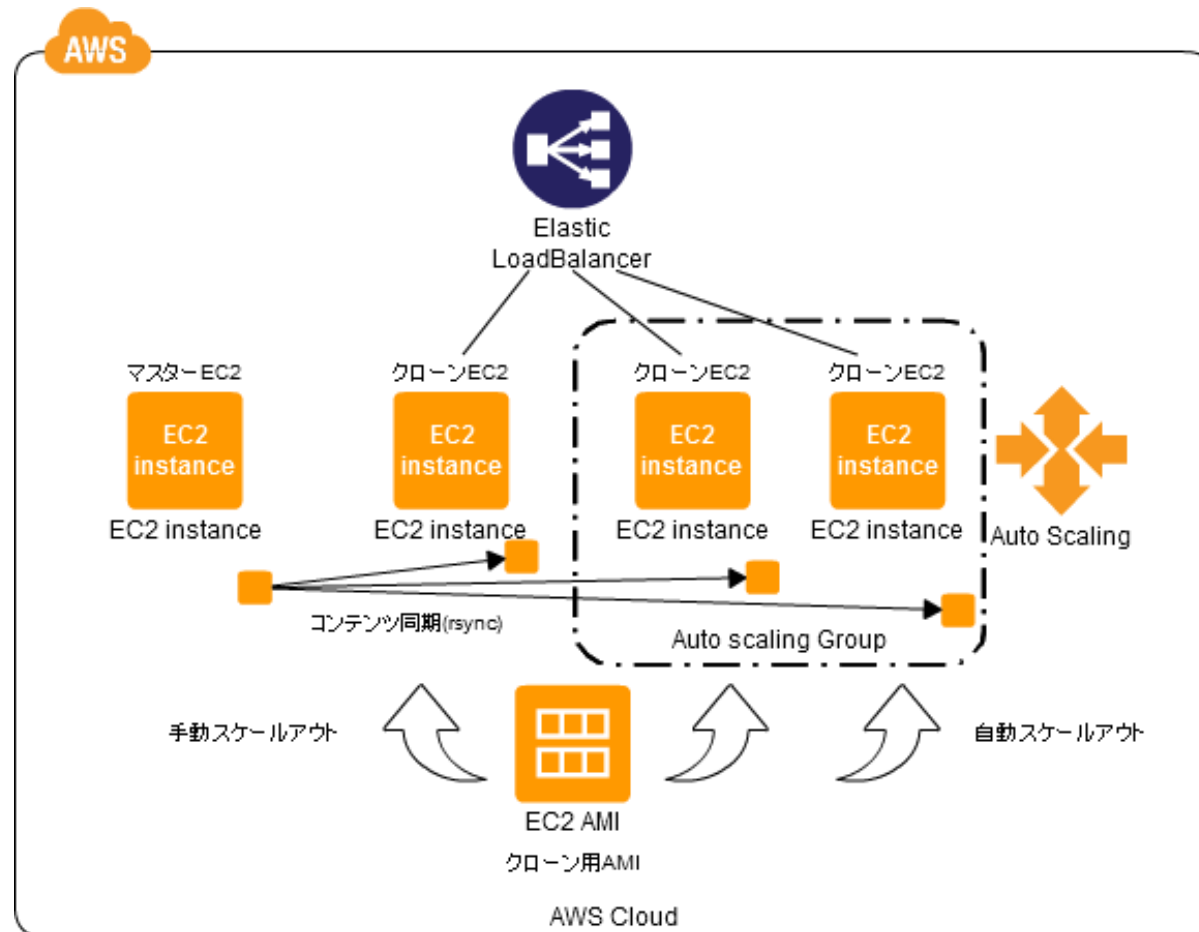
❖課題：

シングル構成のサーバーをスケールアウトさせたい

Clone Server パターン

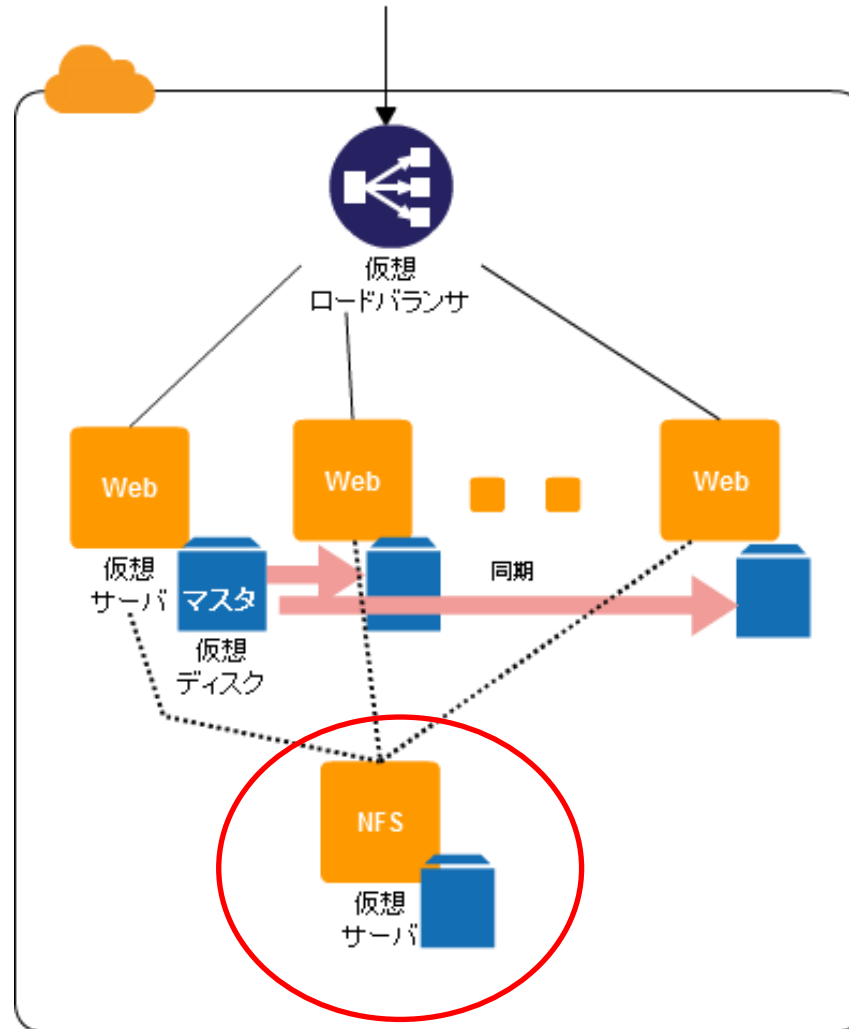


❖ クローン用AMIからスケールアウト



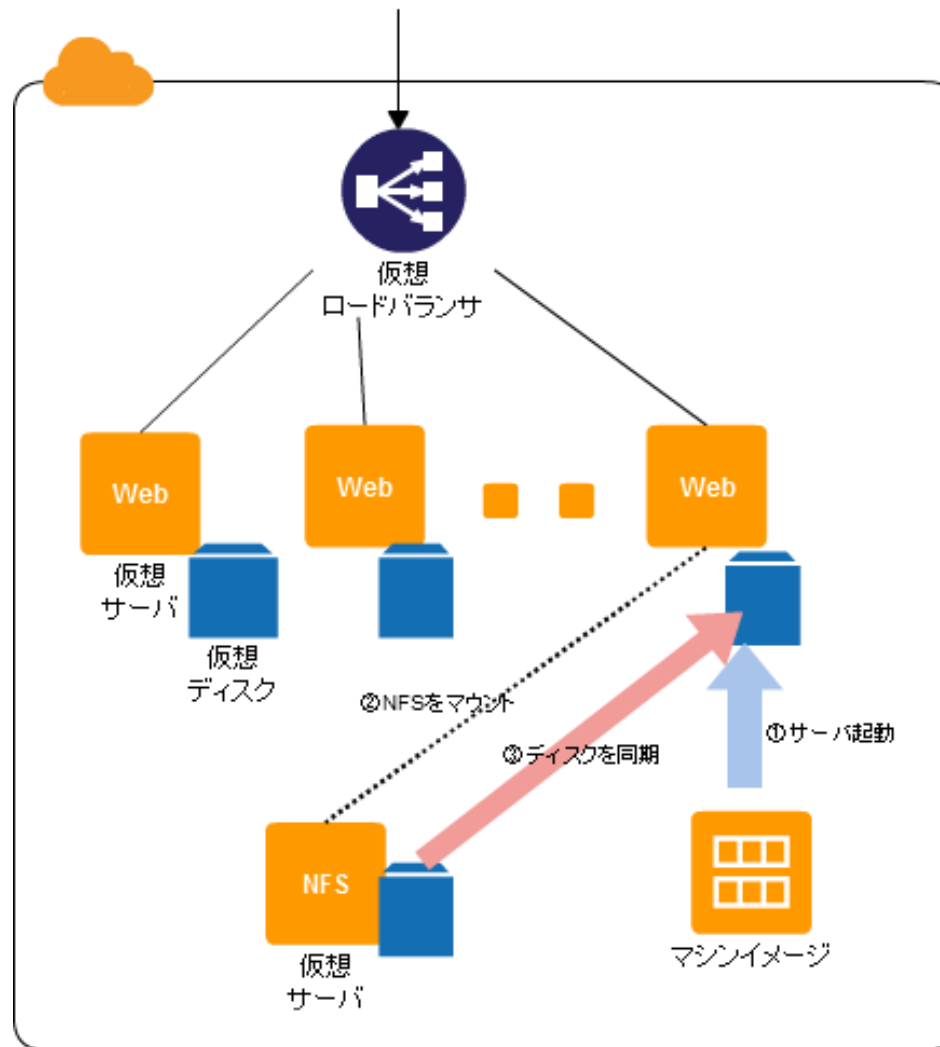
❖課題：
リアルタイムで共有コンテンツを反映させたい

NFS Sharing パターン



❖課題：
NFS部分にパフォーマンスの問題が。。

NFS Replica パターン



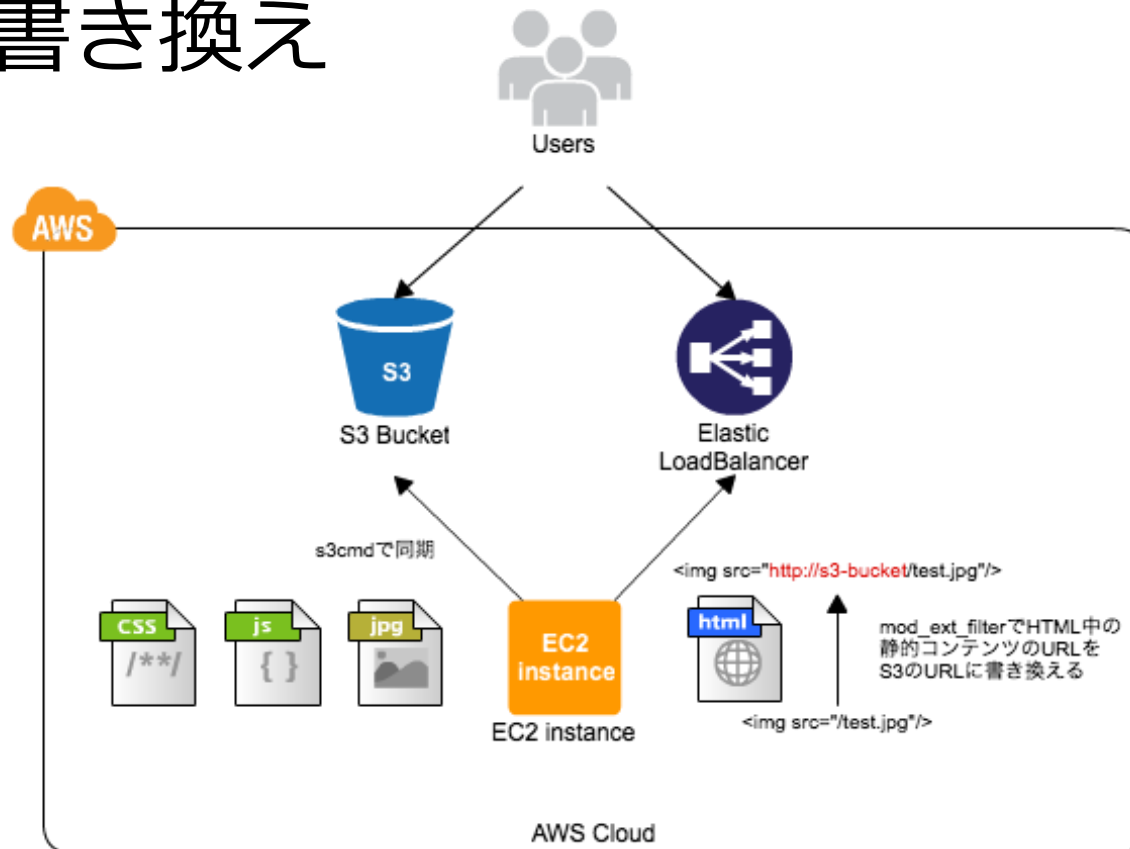


❖課題：
コスト効果高く、リクエストをさばく

URL Rewriting パターン



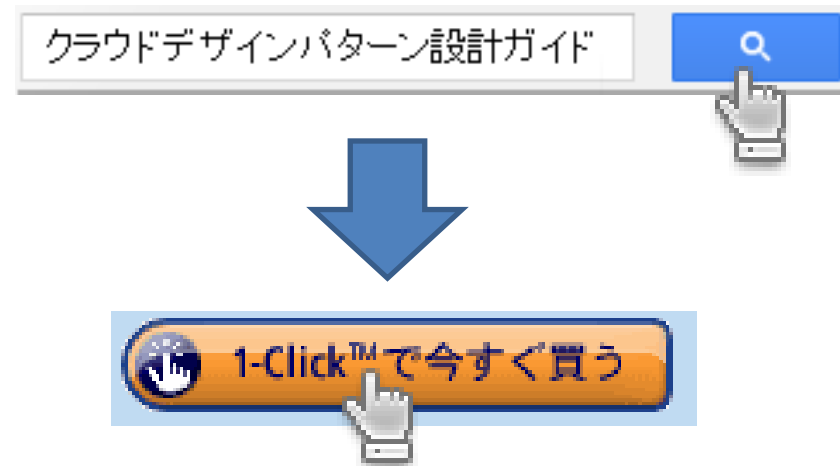
- ❖ S3に静的コンテンツを分散
- ❖ Mod_ext_filterでコンテンツ中のURLを動的に書き換え



その他の実装シナリオ

- ❖ 画像動画配信サイト
 - ❖ 大量のユーザに配信したい
- ❖ Eコマースサイト
 - ❖ 可用性、耐障害性高くしたい
- ❖ キャンペーンサイト
 - ❖ 突発的アクセス増加に耐えたい
- ❖ 他にも多数
 - ❖ ログ解析のシナリオ
 - ❖ 監視のシナリオ
 - ...

Amazon Web Services クラウドデザインパターン 設計ガイド



<http://www.amazon.co.jp/dp/4822211967/>



JAWS-UG
AWS User Group - Japan

CDPアンチパターン！





JAWS-UG
AWS User Group - Japan

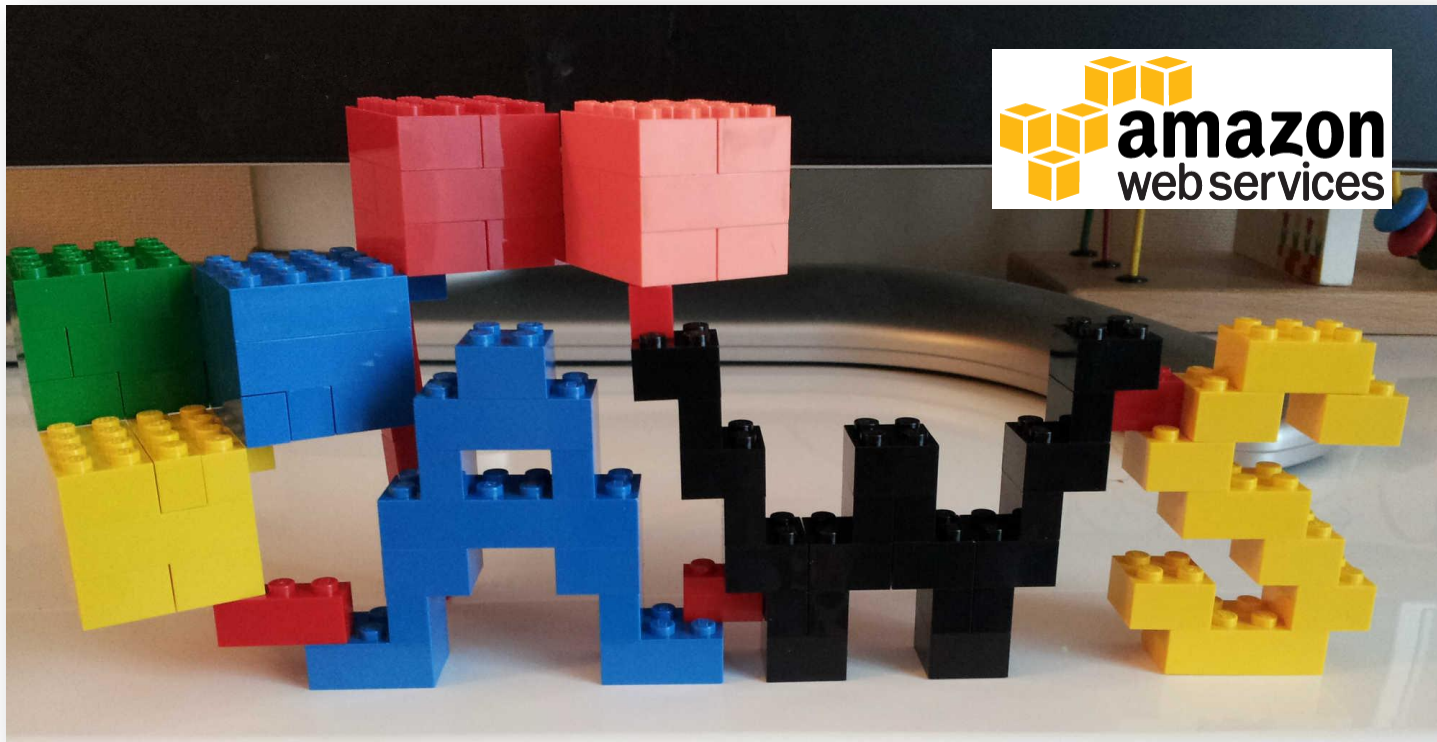
さいごに



- できるだけサービスを利用
- 机上実験よりも実証実験
- スモールスタートからスケールアウト
- 変化に対し全レイヤで対処
- 故障のための設計(Design For Failure)
- 最初だけでなく周期的なカイゼン

One More Thing

❖ CDPでAWSをもっと楽しく





ご清聴ありがとうございました
@KenTamagawa