



2023年5月23日

AWS有効活用術！
目指せEC2からの脱却
～弊社事例を交えてご紹介～



自己紹介

□ 馬場 潤一 (Junichi Baba)

□ 株式会社スタイルズ

- ▶ AWS上でのアプリ開発・インフラ構築
チームリーダー

□ 好きなAWSサービス

- ▶ CloudFormation、Systems Manager、ECS(Fargate)



本日、お話しする内容

□対象

- ▶ EC2を多く利用している方
 - ※オンプレミス環境と大きな違いのない構成で運用をしている方
 - ※他環境から移行してEC2で構築したがそのまま運用している方

□ゴール

- ▶ AWSのサービスを活用した構成にするメリットは何かを把握し、EC2からの脱却という観点でその進め方を理解する

EC2とは

EC2とは

▶ 一般的な仮想サーバサービス

- ・ オンプレミス環境における仮想サーバと同じような形で利用できる
- ・ AWSへの移行時は、まずはEC2を選択する事も多い

▶ EC2のメリットとデメリット

- OS内は自由に構成可能
- ・ スペック変更が容易
- ・ サーバ停止中は費用がほぼかからない

- ✕ OS内はブラックボックスになりがち
- ・ OSやソフトウェアの保守(パッチ適用)が大変
- ・ 障害時の復旧が大変
- ・ 常時起動すると費用は高くなりがち

デメリットに記載した状況に該当しているケースはAWSを活用しているとは言えない。
クラウドネイティブな構成にすることで、AWSのメリットをより活かせるようになる。

クラウドネイティブ

クラウドネイティブとは

▶ クラウドを活用する上での考え方やアーキテクチャ

- ・ クラウドサービスを最大限活用し、効率的にアプリケーションを構築・運用すること
例) DBを冗長化する場合、EC2上でクラスタウェアを導入するのではなく、RDSの冗長化機能 (Multi-AZ) を利用する。

▶ クラウドネイティブのメリットとデメリット

- インフラ運用 (保守対応等) の極小化
- スケーリングしやすい
- アプリ更新の効率化がしやすい (CI/CD)
- コストメリットが出しやすい
※構成上冗長化することで増える可能性も

- ✗ クラウドの独自機能を利用する
- ✗ クラウドサービスの仕様に沿った開発が必要

⇒ 脱EC2を目指し、クラウドネイティブな構成に近づけることで、上記のメリットを得られやすくなる

クラウドネイティブ化の課題

➤ アーキテクチャ

- ・ EC2上に構成済の一部の機能をクラウドサービスに置き換える必要があるが、
 どのような構成がベストなのかわからない
- ・ アプリ改修が必要になる

➤ 人材やスキル

- ・ 学習する時間がない
- ・ 社内にAWSに詳しい人材がおらず、付き合いのあるベンダも知見が無い

➤ 効果

- ・ 本当にメリットを得ることができるのかわからない

クラウドネイティブ化のポイント

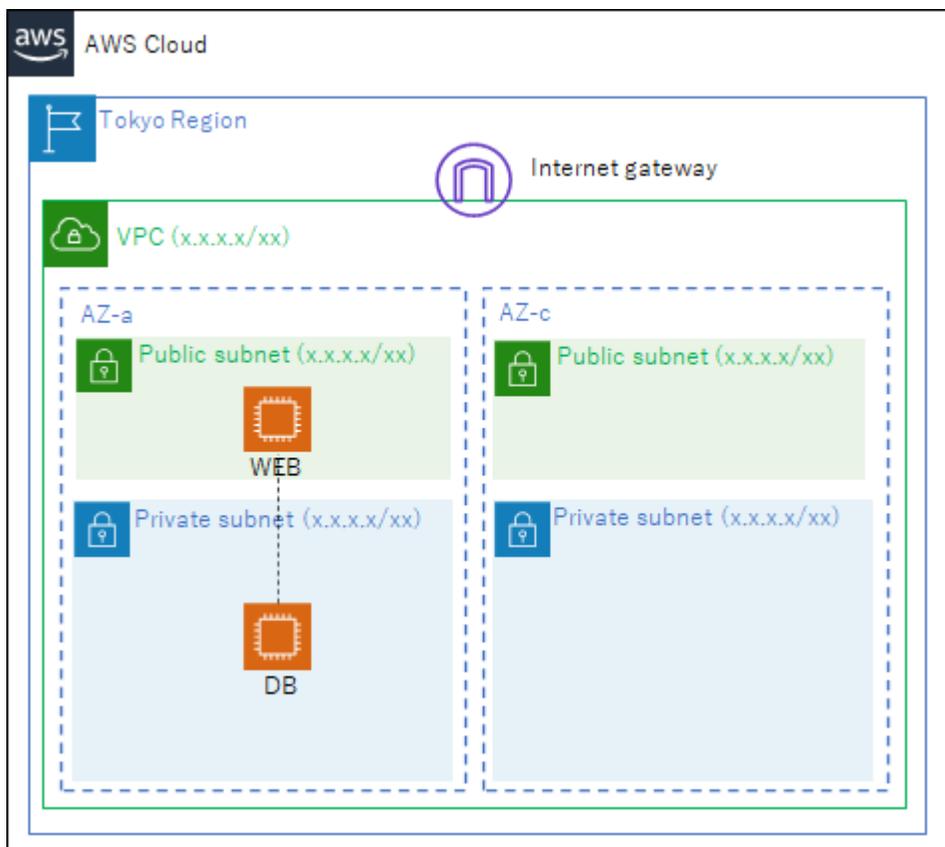
▶ 簡単にできる部分から少しずつ変えていく

- 一気に変える = 難易度もリスクも高い
- ミニマムスタートできず、費用も大きくなってしまうため、余計着手判断し難くなる
- 簡単にできる部分、変更リスクが小さい部分から進めることで、安全に移行できる

⇒ 段階的にクラウドネイティブな構成にしていくケースをご説明します

構成で見るクラウドネイティブ化 ～WebDBシステム～

EC2のみ

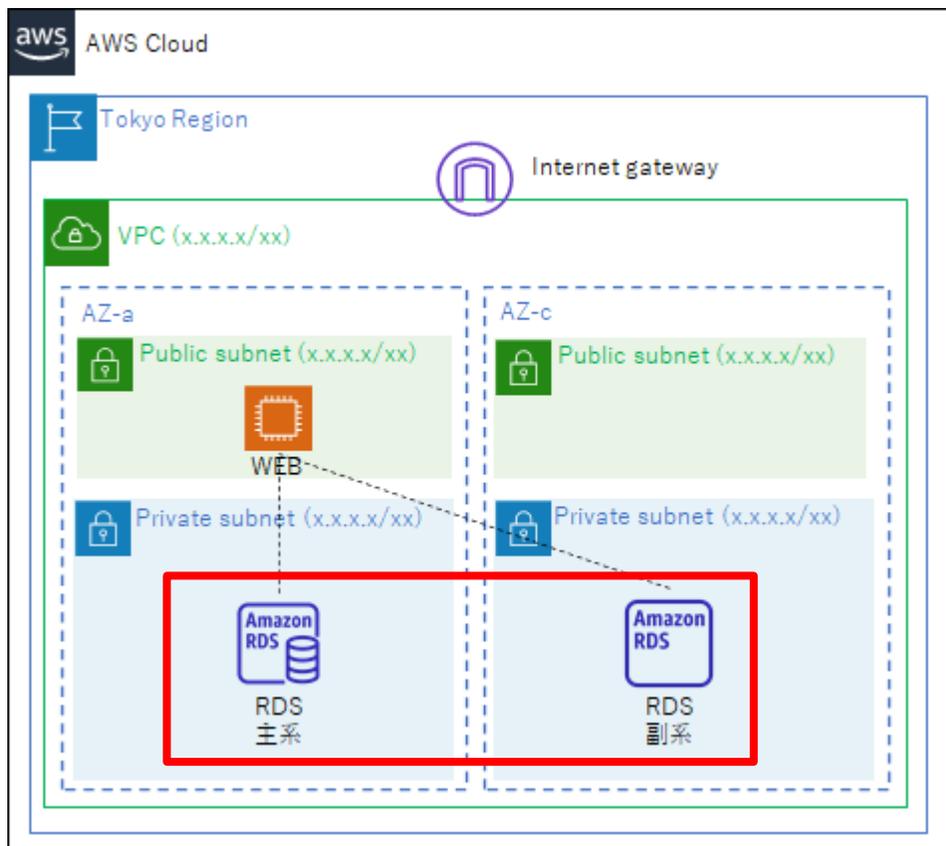


- オンプレミス環境と変わらない構成
- とりあえずEC2で
- AWS移行してそのまま



- パッチ適用やサーバ障害対応 (WEB、DB)
- 非冗長化 (WEB、DB)
- システムがあまり稼働しなくても費用はそこそこ掛かる

DBのRDS化



・ DBをRDSに置き換え

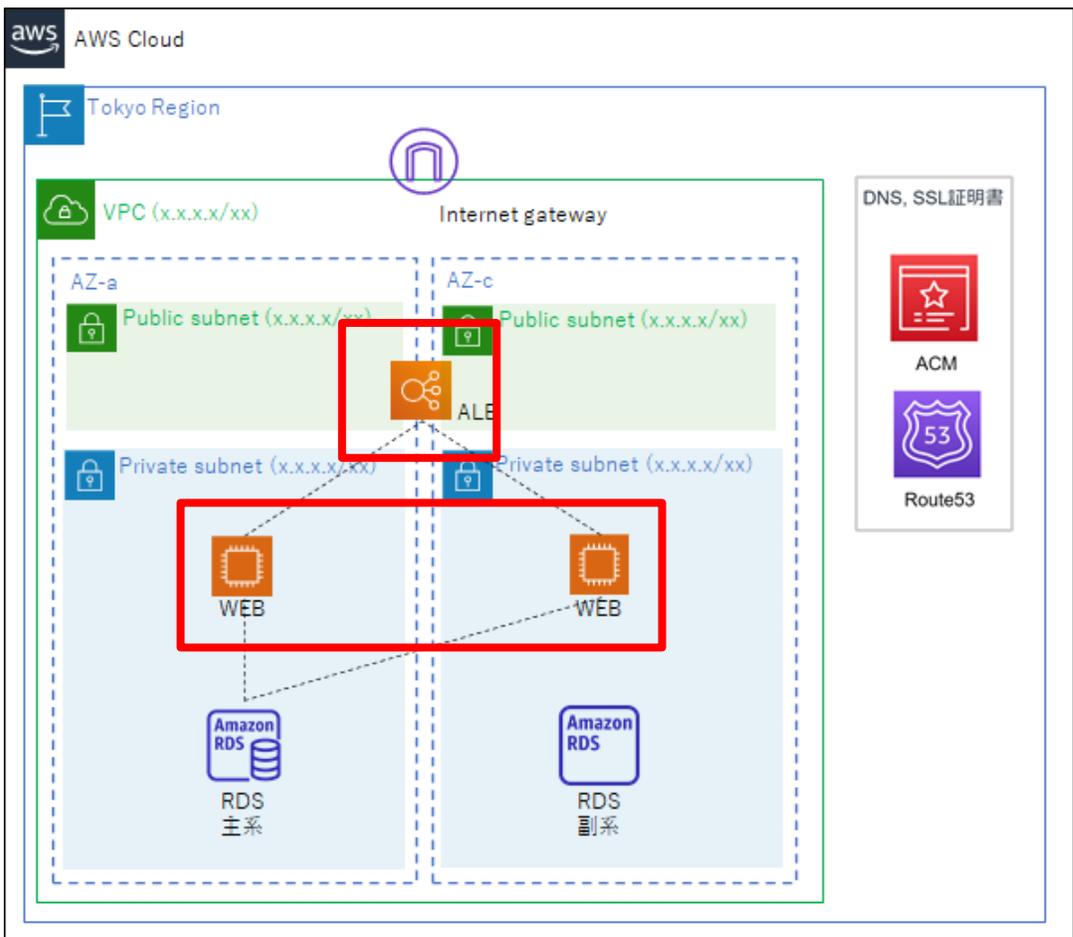


- ・ パッチ適用やサーバ障害対応負荷低減 (DB)
- ・ 冗長化 (DB)



- ・ パッチ適用やサーバ障害対応 (WEB↔DB)
- ・ 非冗長化 (WEB↔DB)
- ・ システムへのアクセス数にかかわらず費用はそこそこ掛かる(WEB・DB)
- ・ DBサーバ上のバッチ処理は手法変更が必要
- ・ DBエンジンとバージョンに制限あり

WEBを冗長化



- ロードバランサー配置 (WEB冗長化)
- WEBはAutoScaling
- DNS、SSL証明書はAWS管理

- パッチ適用やサーバ障害対応負荷低減 (WEB、DB、LB、DNS)
- 冗長化 (WEB、DB、LB)
- アクセス負荷に応じたリソース (コスト効率化) (WEB)

- ✗ システムへのアクセス数にかかわらず費用はそこそこ掛かる (WEB、DB)
- DB上のバッチ処理は手法変更が必要 (移行時)
- DBエンジンとバージョンに制限あり
- 冗長化できるようアプリ改修が必要 (WEB)

サーバレス化

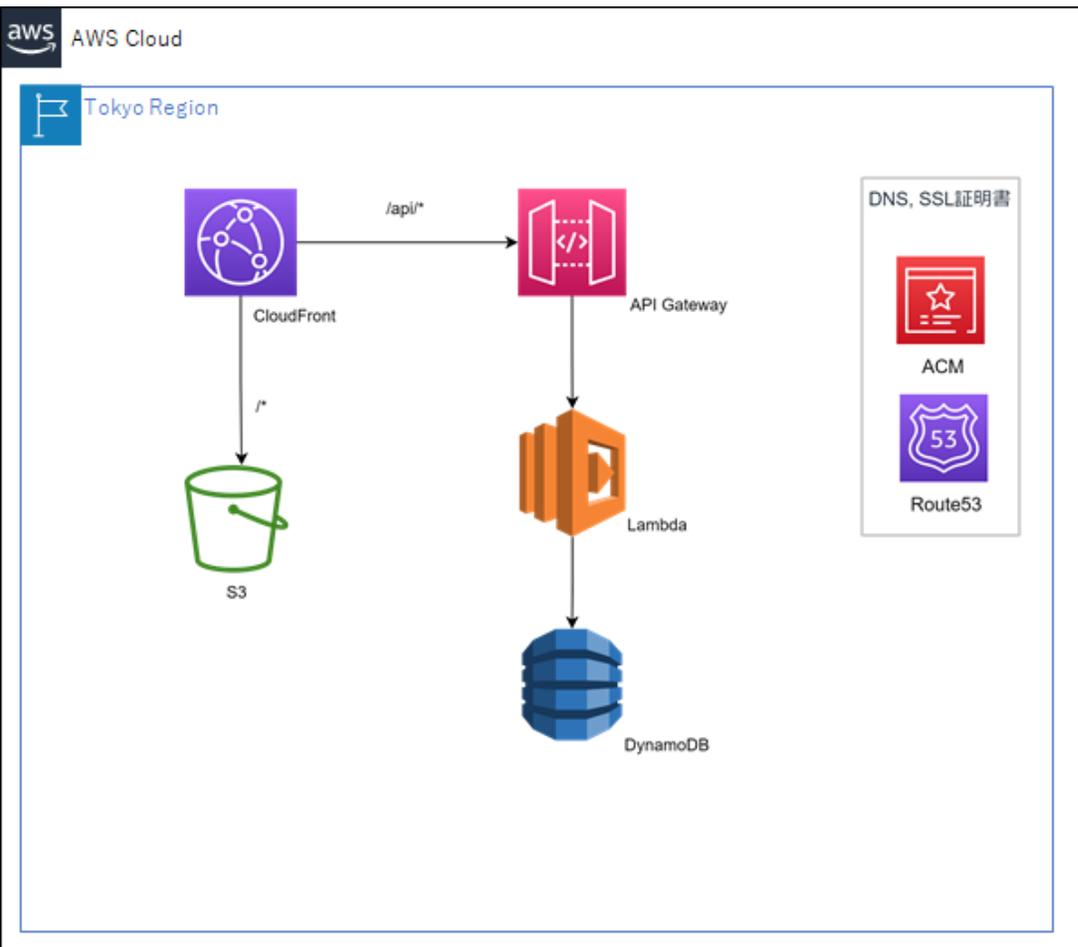
【サーバレス】

アプリケーション稼働に必要な専用サーバを用意せず、クラウド事業者が用意するサービス基盤上で稼働させる構成

- ・ 最終的なゴールの一例
- ・ CDN利用によるレスポンス向上
- ・ S3 Webホスティング機能による静的コンテンツ配信
- ・ API GW、Lambda、DynamoDBによるバックエンド

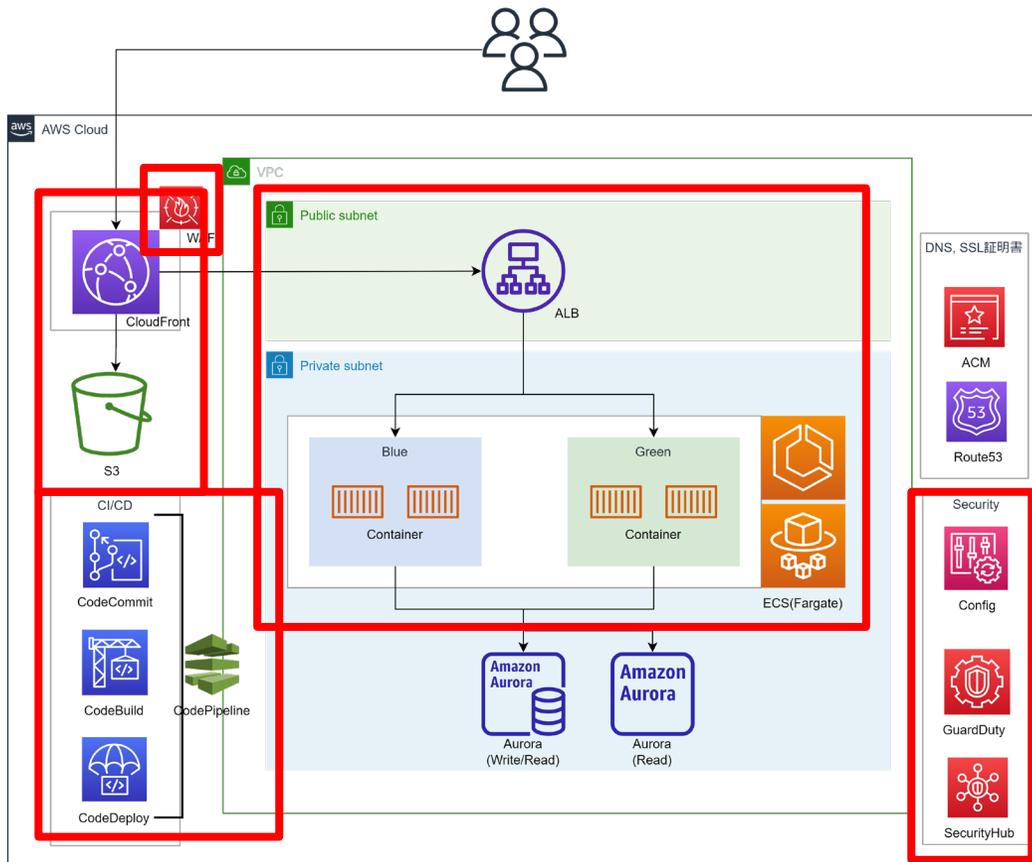
- パッチ適用やサーバ障害対応負荷低減
 - ・ 冗長化、スケーリング
 - ・ アクセス負荷に応じたリソース (コスト効率化)
- ※従来構成の1/10のコストになったケースも

- ✕ RDBではなくなる
- ・ サービス仕様に合わせたアプリ改修



弊社事例

某社：マスタ情報管理Webシステム



- フロントエンドをS3ウェブホスティングに、バックエンドをコンテナで構築
- CI/CDの導入
アプリのソースコード更新に伴い、自動的にフロントエンド/バックエンドのリソースを更新
- WAFを始めとした各種セキュリティサービスを導入
- 全てCloudFormationで構築することで、本番/検証/開発といった類似環境構築を効率的に実施（インフラコード化）
ブラックボックス化も無くせる

コンテナ化やCI/CD導入はスタイルズの得意分野です！

CI/CDに関する過去のウェビナーはスタイルズウェブサイトから御覧ください。

クラウドネイティブ化したくなったら

スタイルズにご相談ください

▶ AWSに特化した技術支援サービスを提供

- **AWSに関するよろず相談窓口**
 - 構成の相談、設計構築
 - AWSサービスに対するQA、ノウハウ提供
 - 不明点・懸念点に対する調査・検証（PoC）
- **AWS以外の範囲も含め、可能な範囲で対応可能**
 - アプリ改修、新規開発
 - CI/CD
 - セキュリティ
 - 監視運用保守
- **サービス料金は調整可能**
 - 想定稼働時間に基づいた料金設定（1人日/月～数人月/月　で実績あり）
- **アプリ開発者/インフラ構築者の混成チームで対応**
 - アプリ観点、インフラ観点 両面での的確な意見が出せる

**クラウドネイティブ化に関するご相談や作業支援は
ぜひスタイルズまでお声がけください！**



実績豊富なエンジニア集団の技術と開発ツールで
「開発期間/コスト削減」「品質向上」を実現

株式会社スタイルズ

<https://www.stylez.co.jp>

東京都千代田区神田小川町1-2 風雲堂ビル6階

Tel:03-5244-4111

オープンソースソフトウェア推進