CloudShift

Stylez

株式会社スタイルズ

2024年11月26日

運用保守費の低減と迅速デプロイを両立！
実例で語るAWSコンテナ移行の最適解

CloudShift

自己紹介

- **馬場 潤一（ばば じゅんいち）**

- **株式会社スタイルズ**
  業務推進グループ
  ソリューション第2チーム　リーダー

- **AWSを主軸として動くチームを管理**
  - AWSインフラの設計構築、アプリケーションの開発、監視運用セキュリティに関する導入　をお客様に提供
  - コンテナ、CI/CDが得意領域

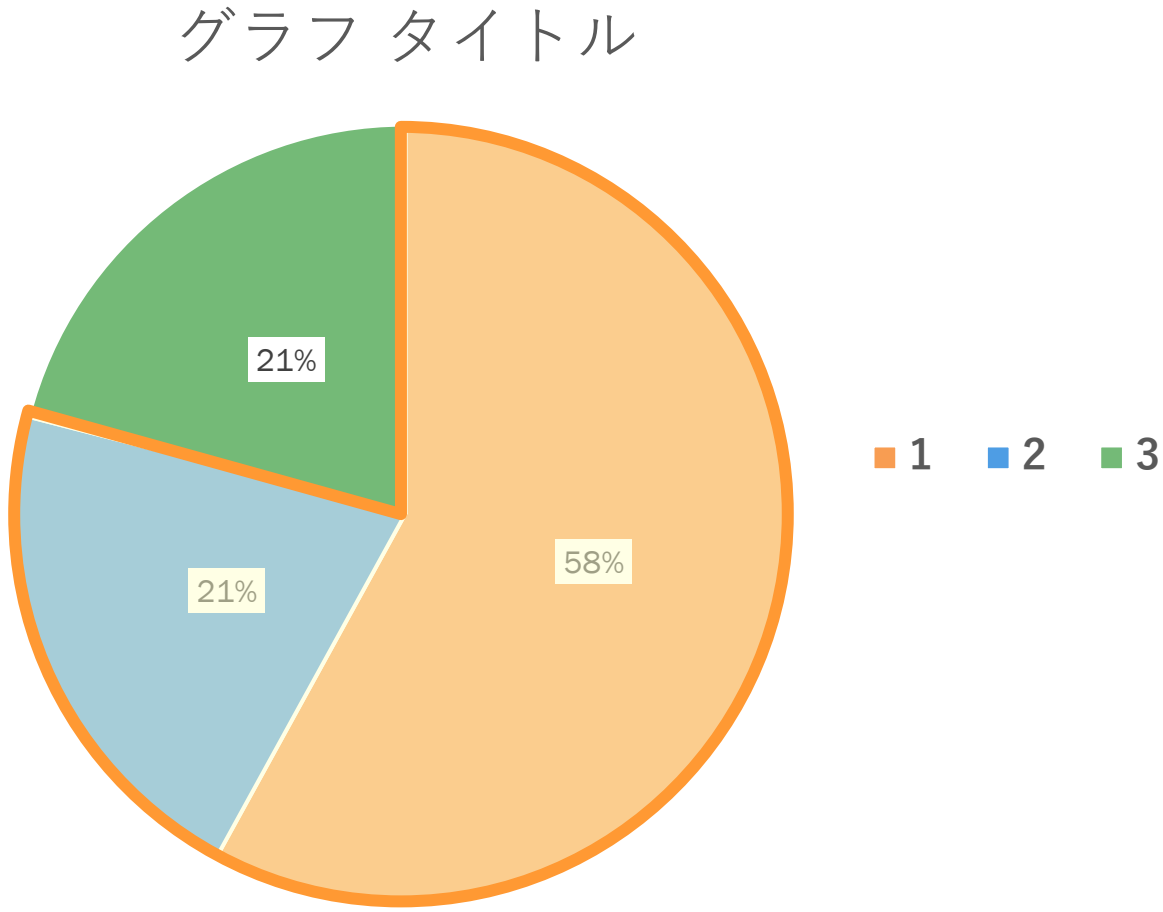
Stylez

CloudShift

なぜコンテナなのか

Stylez

新規アプリの導入環境の割合

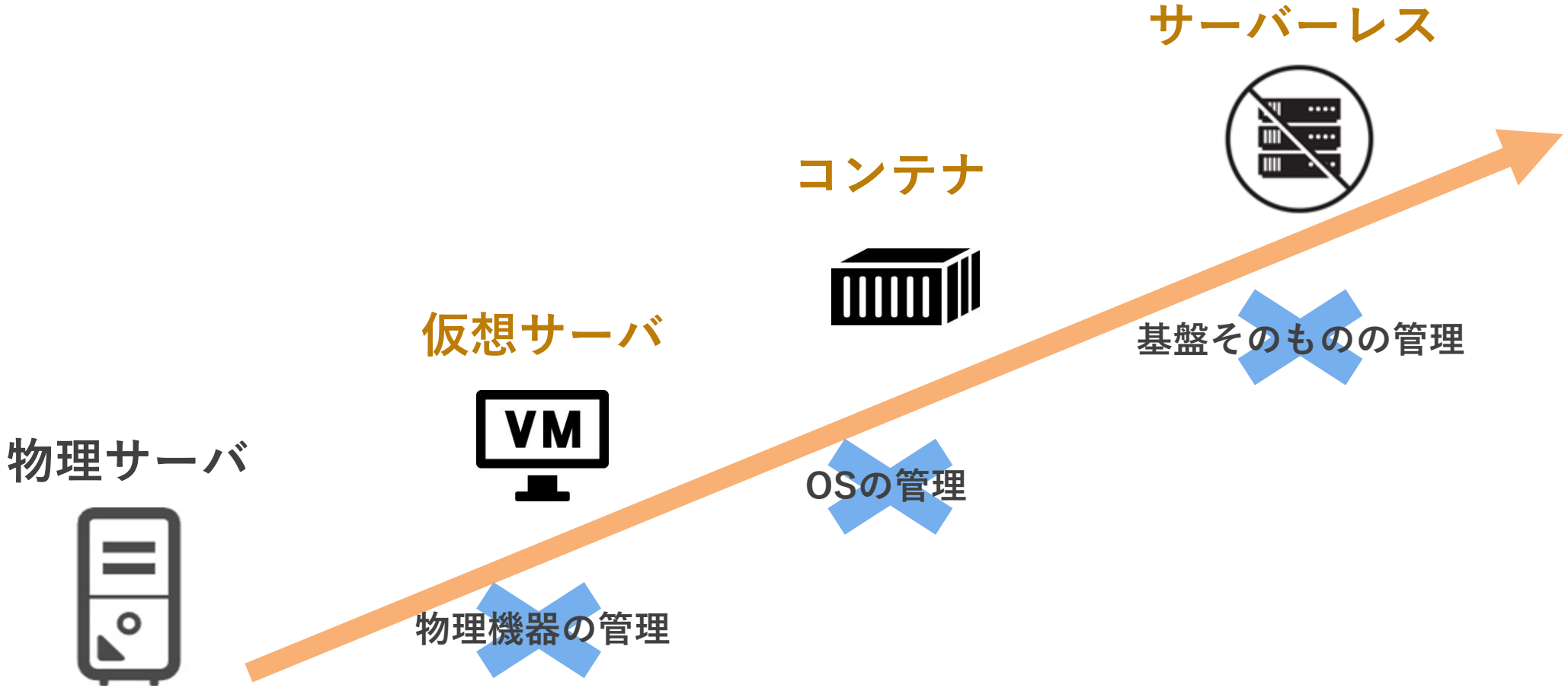


- 約80%の新規アプリはコンテナまたはサーバーレス

※某セキュリティベンダ調べ

コンテナやサーバーレスが選ばれる背景

- プラットフォームの**抽象化**が進んでいる



インフラ抽象化によるメリット

管理の容易さ

物理機器やOSの管理が不要

コスト削減

将来を見据えた過剰なスペックを容易する必要がない
運用に係る作業負荷が減る

セキュリティ向上

OSセキュリティパッチ適用不備に伴う
セキュリティホールは無くなる
セキュリティ対策機能が備わっていれば、
簡単にセキュリティ対策ができる

抽象化が進む　＝　サーバ管理のような非生産的な作業：減少
生産性のあるシステムの開発・改善に集中

CloudShift

サーバーレスでも良いのでは？

サーバーレス開発　＝　サーバーレスサービスの仕様に合わせる

既存アプリの改修箇所が大きくなる

特定サービスに依存し、他サービスに移行できない

サーバ環境からの移行はコンテナがベスト！

CloudShift

コンテナ移行はどうやるの？

Stylez

CloudShift

コンテナ移行はどうやるの？

- どんな構成になる？
- 何を意識しないといけない？
- どう進める？
- コンテナ以外に考えないといけないことは？

Stylez

CloudShift

サーバ運用のお悩みを解決するコンテナ移行のアプローチ

AWSマネージドサービスを活用

1. インフラを抽象化した構成
2. 冗長化を意識
3. CI/CDも検討
4. とりあえず作ってみる
5. AWSセキュリティサービスも一緒に導入

アプローチ1. インフラを抽象化した構成

ECS ＋ Fargate ＝ 初学者でも使いやすいコンテナ環境

ECS：Dockerベースのコンテナ管理サービス

- シンプルな構成
- AWS管理画面上で操作
- AWSサービスとの連携（ストレージ,セキュリティ）

Fargate：コンテナ実行環境の抽象化

- コンテナ実行環境（ホスト）をAWSが管理

管理の容易さ

CloudShift

アプローチ2. 冗長化を意識

複数コンテナで処理 → データ（DB,セッション,ファイル,ログ）をコンテナ外部で管理

↓ 自動スケーリング

耐障害性向上
コストメリット

↓ 抽象化出来る

管理の容易さ

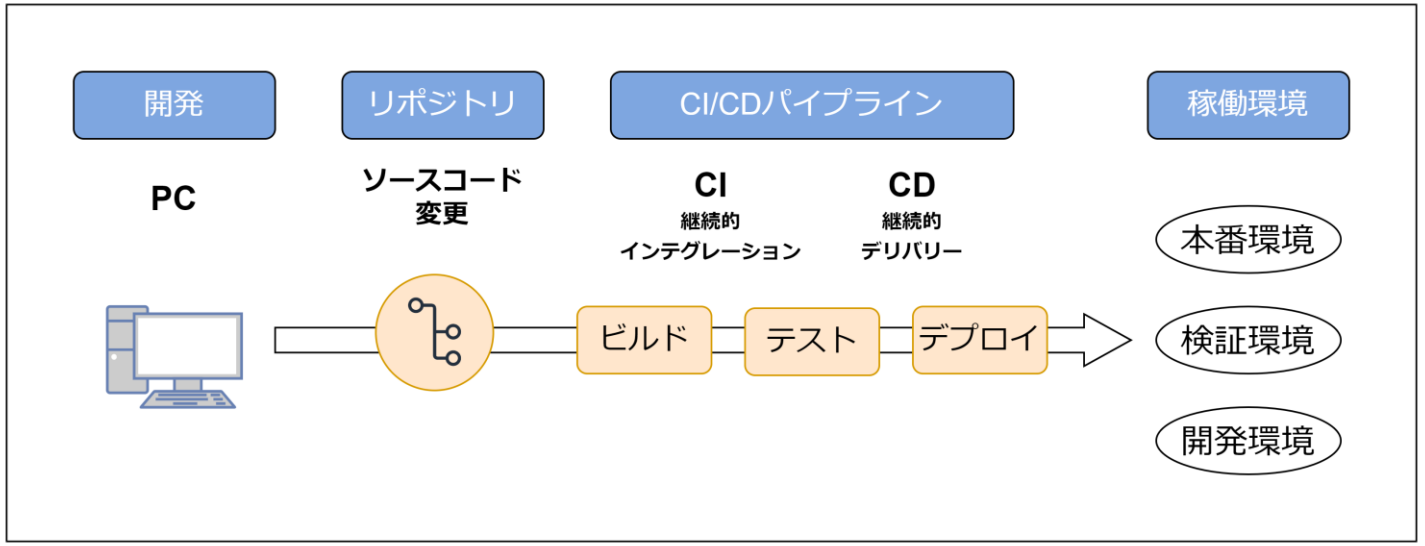
↓ アプリ改修箇所は最低限

市場投入までの
時間短縮

Stylez

アプローチ3. CI/CDも検討

CI/CD：　開発→ビルド→デプロイの自動化



開発サイクルの向上

デプロイ方法の選択肢
Blue/Green、カナリアリリース

市場投入までの時間短縮　　品質向上

切替/戻しが楽に

CloudShift

アプローチ4. とりあえず作ってみる

懸念材料の早期払拭　＆　動くものが出来上がる

懸念材料の検証を実施

- 方向性修正が早期に可能。
- 一部でも動くアプリを見ることで、具体的な改善点が見える。
- 仕様変更にも柔軟に対応。

市場投入までの時間短縮

CloudShift

アプローチ5. AWSセキュリティサービスも一緒に導入

セキュリティ対処検討は構成検討時から始める

- セキュリティ要件に対する対策内容で構成が変わる
- セキュリティ指摘事項の可視化は簡単＆安価

セキュリティ向上　　市場投入までの時間短縮

CloudShift

簡単にできることではない・・・

Stylez

CloudShift

コンテナ移行における
お客様のお悩み

Stylez

CloudShift

お悩み (1)

コンテナは触ったことがない。
学ぼうにも学習コストが高そう。
運用保守出来るイメージがない・・・。

シンプルな構成で運用を実践すれば、
仕組みを理解しやすい。

伴走しながら構築＋勉強会を実施してくれる
ベンダと組めると安心。

Stylez

お悩み (2)

アプリとインフラで担当者(ベンダ)が別。
統率を取って進めていけるのか・・・。

**・コンテナやCI/CDの難しい部分は
アプリとインフラ双方の技術要素が絡み合う所。**

・双方の領域で対応できるベンダが望ましい。

**・双方の領域上グレーな部分で課題が出るので、
責任の押し付け合いが無い体制になっている事が
望ましい。**

お悩み (3)

アプリ開発やインフラの担当はいるが、
AWS(またはコンテナ)の経験が無いので、
設計フェーズだけ支援してほしい。

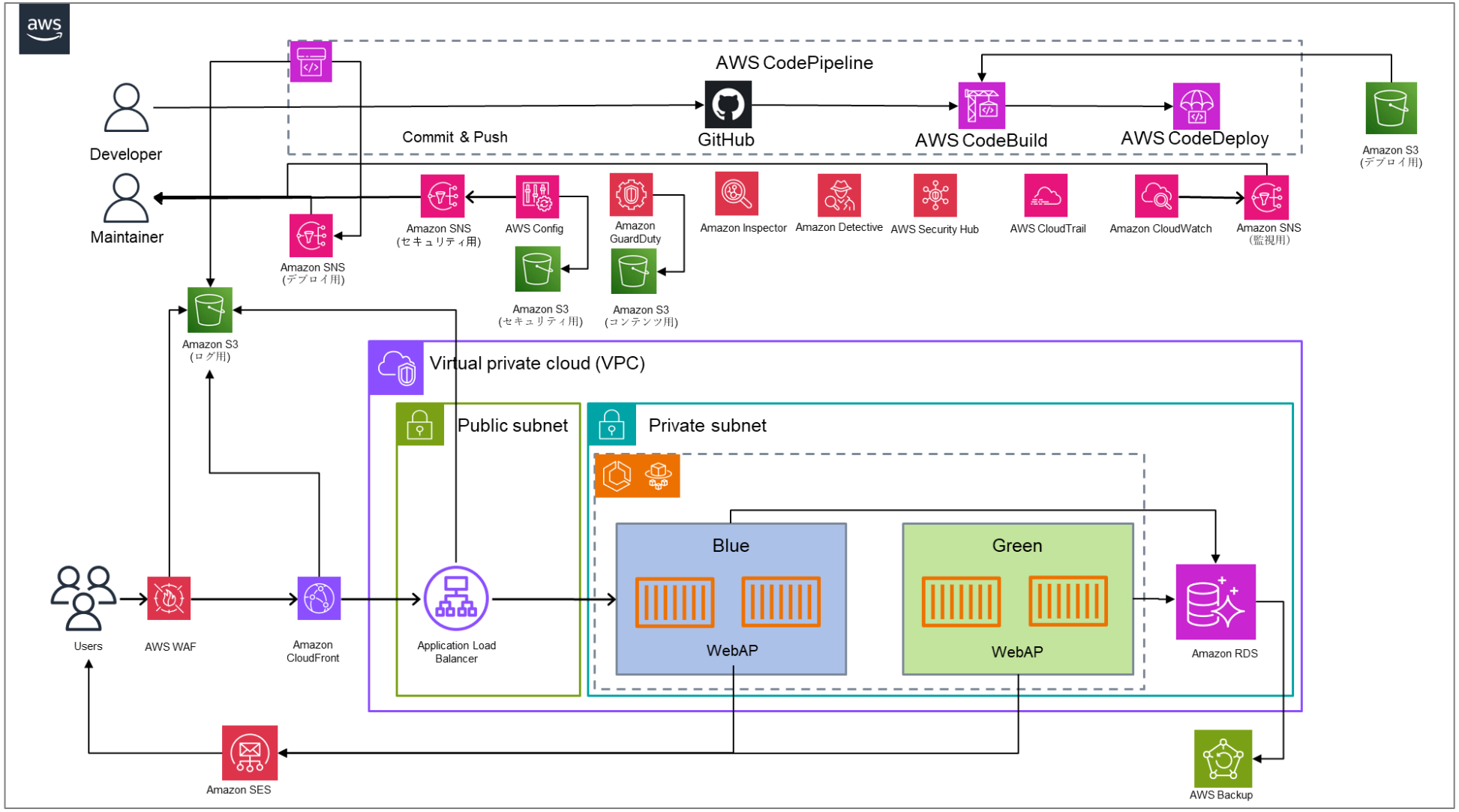
特定フェーズ/作業のみ支援する契約が組めると良い。

例えば・・・

- 設計のみ
- CI/CDの設計・構築のみ
- Q&Aや調査検討

スタイルズのお客様事例

お客様事例



システム概要

レンタル事業社

コーポレートサイト
兼 ECサイト

移行元環境は
オンプレWebDB構成
※WordPressベース

課題と解決方針

【課題】

・オンプレミス環境で
WordPressサーバを
冗長化できない事による
耐障害性への懸念

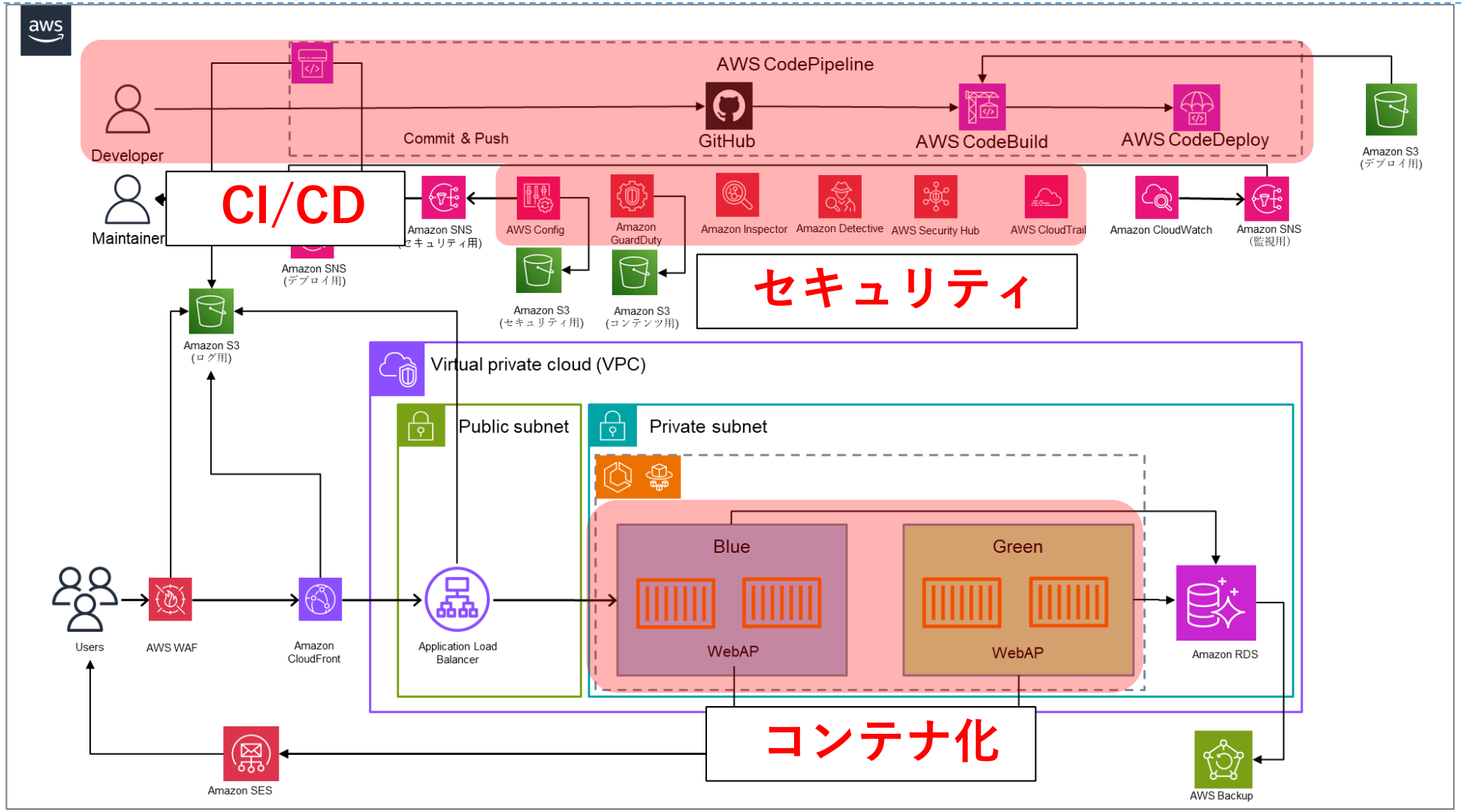
・WordPressプラグインや
PHPの脆弱性対応に追われ
運用保守性の悪さ、
セキュリティリスク

【解決方針】

- ✓ 冗長構成
- ✓ シンプルなコンテナ構成
- ✓ WordPress撤廃（ヘッドレスCMS）
- ✓ CI/CDによる自動化
- ✓ AWSセキュリティサービス導入

障害時のサービス影響：減　　運用負荷：減　　セキュリティリスクに対する安心感

お客様事例(再掲)


aws
AWS CodePipeline
Developer
Commit & Push
GitHub
AWS CodeBuild
AWS CodeDeploy
Amazon S3
(デプロイ用)
Maintainer
CI/CD
Amazon SNS
(セキュリティ用)
Amazon SNS
(デプロイ用)
AWS Config
Amazon GuardDuty
Amazon Inspector
Amazon Detective
AWS Security Hub
AWS CloudTrail
Amazon CloudWatch
Amazon SNS
(監視用)
Amazon S3
(セキュリティ用)
Amazon S3
(コンテンツ用)
セキュリティ
Amazon S3
(ログ用)
Virtual private cloud (VPC)
Public subnet
Private subnet
Blue
Green
WebAP
WebAP
Users
AWS WAF
Amazon CloudFront
Application Load Balancer
Amazon RDS
コンテナ化
Amazon SES
AWS Backup

CloudShift

まとめ

インフラの抽象化により生産性の高い作業に注力できる

サーバ構成からのステップアップとしてコンテナは最適解

お客様が必要とする領域だけ支援できるベンダがいると安心

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

オープンソースソフトウェア推進