環境の自動化

イノベーションノウハウ

デザイン思考

１共感　　 顧客の現場を見る：観察

２問題定義 インサイトを分解・統合し、焦点を絞る

３創造 不可能なものを多くの解決策を創出する

４プロトタイプ 案をプロトタイプとして簡易に具体化

５テスト　　 ユーザに試しフェードバックを得て修正する

リーンスタートアップ

構築 → 　計測 　 →　 学習 →　　構築

　製品 データ アイデア

アジャイル開発がこのイノベーションノウハウにあっている

疎結合やマイクロサービス化などに繋がっていく

DevOpsとはシステム開発とシステム運用とを相互連携させる体制・仕組み

→　変更が迅速に

DevOps

自動化ツールをAWSで実現する

CloudFrontやECSなど

AWSでの自動化ツール

主要サービス

AutoScaling

CloudFormation

Codeシリーズ（Git上の処理をAWS上で実現自動化）

ECS（インフラ設定のコード化）

ElasticBeanstalk(Webアプリケーション特化)

OpsWorks（インフラ設定自動化）

EC2 Run Command

Codeシリーズ

コーディング・ビルド・テスト・展開（デプロイ）を自動化するサービス

CodeCommit（Gitベースのリポジトリをホストする）

CodeBuild（コンパイルしてテスト実行　デプロイ可能なPackage作成可能）

CodeDeploy（デプロイの自動化）

↓

CodePipelineで一連を管理

ElasticBeanstalk

速く簡単にWebアプリケーションの定番構成の構築・デプロイの自動化

Java,PHP,Ruby,Python,Goなどの言語に対応

ウェブサーバ環境やワーカー環境を提供

構成要素

アプリケーション・バージョン・環境・環境設定

ユースケース

ＷＥＢアプリケーションのデプロイを容易にする

タスク時間の長いワークロードの展開に利用する

OpsWorks

アプリケーションの設定状況をコードにして運用管理するサービス

Chef専用環境はフルマネージド型で提供される(OpsWorks for Chef Automation)

Chef

Recipeという単位でインフラ設定・運用の仕組みをコードで保持

インフラの展開が容易に

OpsWorks for Chef Automation

AWSが提供しているChef専用環境をフルマネージド型で提供する

OpsWorks for Puppet Enterprise

パペット構造・・・インフラ構造

構造は共有可能

オンプレミスのデバイスのデプロイまでサポート

OpsWorksスタック

スタックとアプリケーションの作成及び管理のためのシンプルで柔軟な

方法を提供するAWSオリジナルサービス

Amazonのコンテナサービス

レジストリ ：コンテナのイメージが保管される場所

AmazonECR

コントロールプレーン：コンテナを管理するサービス

AmazonECS

AmazonEKS

データプレーン ：コンテナが実行される環境

AWS Fargate

CloudFormationの概要

AWSクラウド環境内の全インフラリソースを記述してテンプレート化して展開

する環境自動設定サービス

ユースケース

AWSリソースの構築の効率化

開発・テスト・本番環境のインフラを標準化

毎回同じリソースやプロビジョニング設定を正確に利用したい

ソフトウェアと同じ環境構成を管理したい

テンプレート（JSON/YAML）

CloudFormation

スタック

CloudFormationの機能

変更セット

ドリフト

スタックセット

スタック間のリソース参照機能

CloudFormationデザイナー

利用することで視覚的にテンプレートを作成できる

テンプレート

テンプレートバージョン

テンプレートの説明

テンプレートに関する追加情報（MetaData）

実行に必要なパラメータとしてKeypearやユーザ名などを記述(Parameters)

条件パラメータ値を指定するためのキーと値のマッピング記述(Mappings)

リソース作成時の条件名と条件内容を記述（conditions）

サーバレスアプリのＳＡＭバージョンを記述(Transform)

実際にスタックに生成するリソースとプロパティを記述(Resouces)

リソースの名称やタイプ・プロパティなどの設定内容を記述(その下から)

リソース間の依存関係を記述(DependOn)

組み込み関数(!Ref )

出力先(Outputs)

C