ECS on Fargateでリリース3分クッキング

株式会社セゾンテクノロジー
北野　佑典

自己紹介

北野　佑典

- 弊社で3社目（入社してまだ1年未満）
- 普段の業務：実はAWSを殆ど触っていない保守開発
- 今年度の目標：AWS-DVA取得
- 最近のマイブーム：24時就寝8時起床
- ひとこと：ゆるゆるで行くので質問はお手柔らかに

このLTのターゲット層

- Webアプリ開発に注力するからなるべくインフラ面を考えたくない
- リリース作業にかける時間を極限まで減らしたい
- 運用/管理の手間はなるべく減らしたい

このLTのターゲット層

- Webアプリ開発に注力するからなるべくインフラ面を考えたくない
- リリース作業にかける時間を極限まで減らしたい
- 運用/管理の手間はなるべく減らしたい
- 金銭面でコストを度外視できる（重要）

※EC2を使う方が安くなるケースでも
ランニングコストが気にならない企業/個人の方

サーバーを用意してあれこれ…

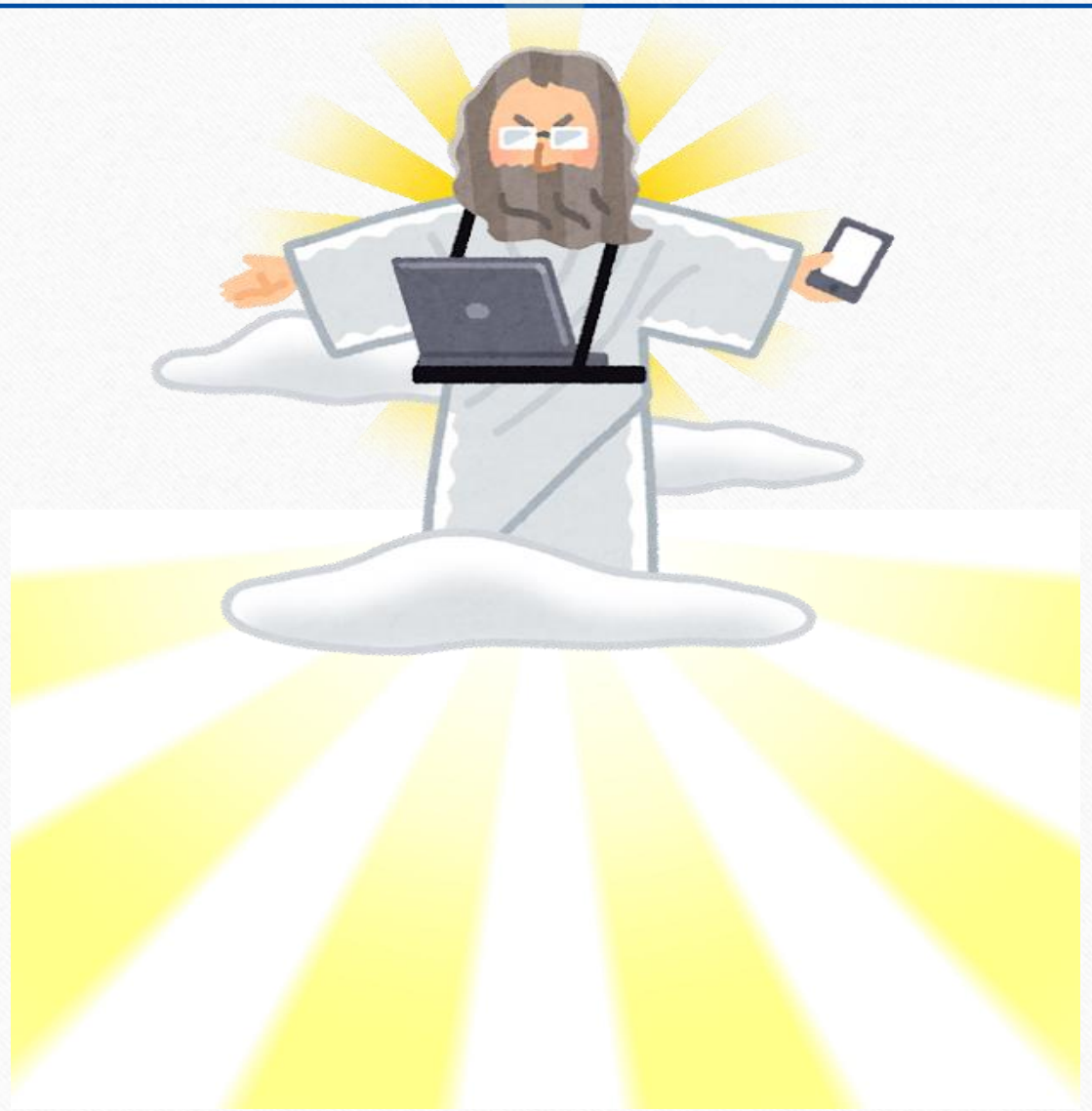
- リリースするための色々な設定 (OS, ミドルウェア等)
- クラウド / オンプレともに発生する管理コスト
- 運用保守周りも考える必要が…
- etc…

新規サーバへのリリース

サーバーを用意してあ

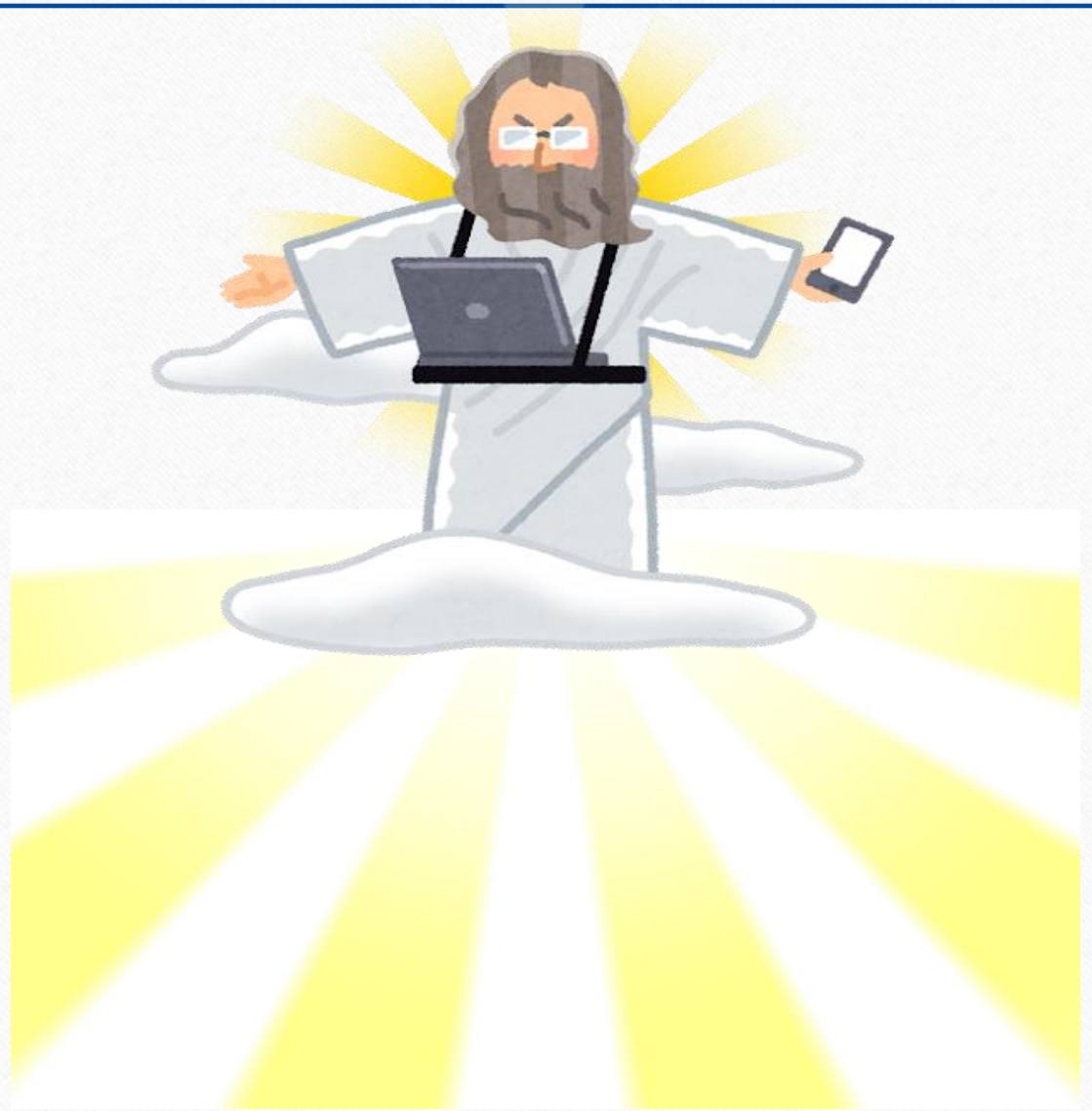
- リリースするための
S, ミドル
- クラウド / オンプレ
理コス
- 運用保守周りも
- etc…

救世主 ECS on Fargate



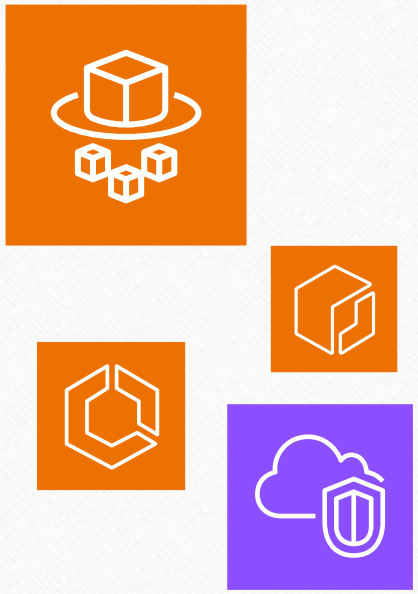
救世主 ECS on Fargate

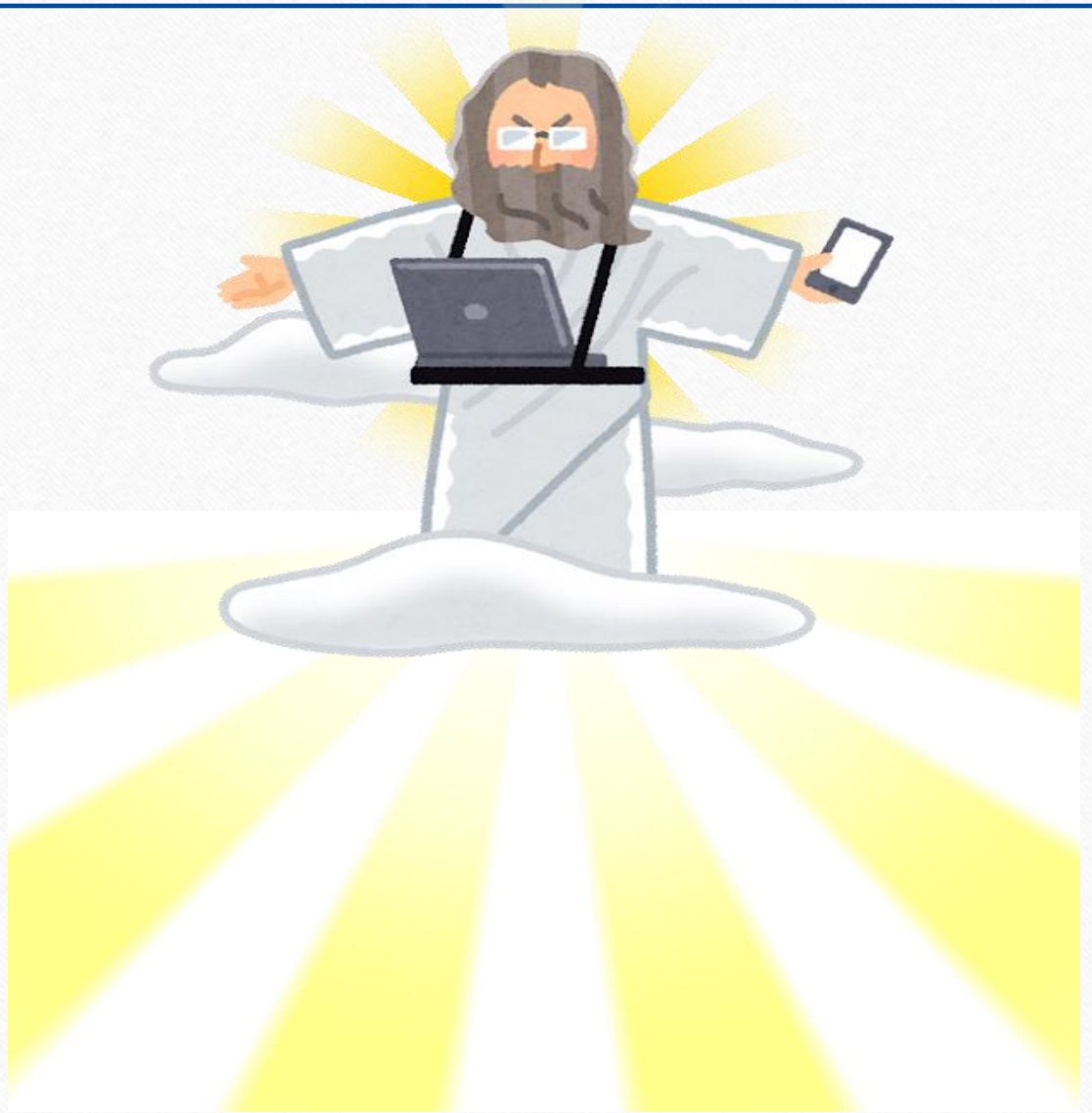
主は「Fargateあれ」と言われた



救世主 ECS on Fargate

主は「Fargateあれ」と言われた
するとFargateがあった





Fargateとは何か？

コンテナをサーバーレスで実行できるサービス

・ECS/EKSを実行環境として利用

・支払い金額は従量課金

・AWSマネージドなので、OSメンテナンス等がいらない

・プロビジョニング、スケール等もAWSが自動で実行

→ つまりインフラ面に取られる時間がほぼなくなる！

準備するものが少ない

用意するのはなんと

だけ！

簡単にWEBアプリがリリースできちゃう！

準備するものが少ない

用意するのはなんと

コンテナイメージ(とネットワーク設定) ※1

だけ！

簡単にWEBアプリがリリースできちゃう！

※1 このLTではネットワーク設定については省略します。

作業手順はたったの5Step

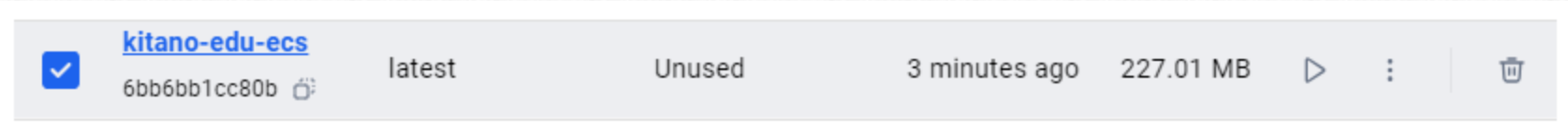
1. Dockerイメージの作成
2. ECRへの作成イメージpush
3. ECSクラスター作成
4. ECSタスク定義作成
5. ECSサービス作成

作業手順はたったの5Step

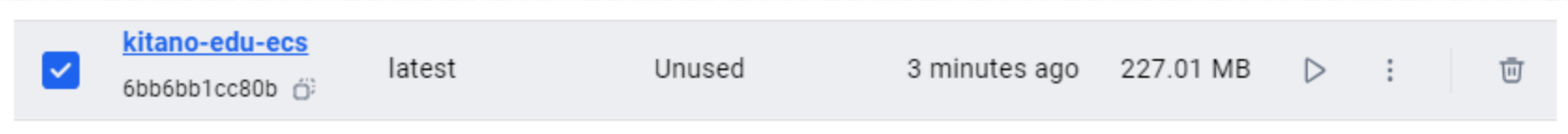
1. Dockerイメージの作成
2. ECRへの作成イメージpush
3. ECSクラスター作成
4. ECSタスク定義作成
5. ECSサービス作成

更にこれらは自動化できる！
※今回は省略

実際にやってみた step1, 2

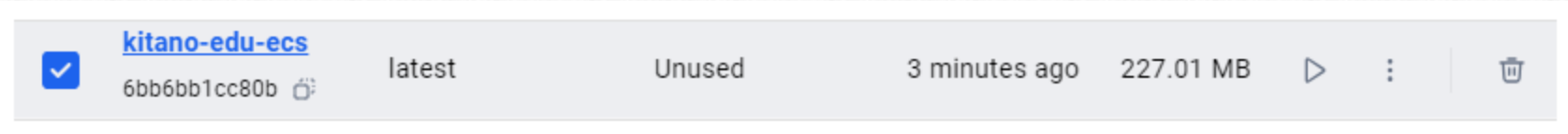


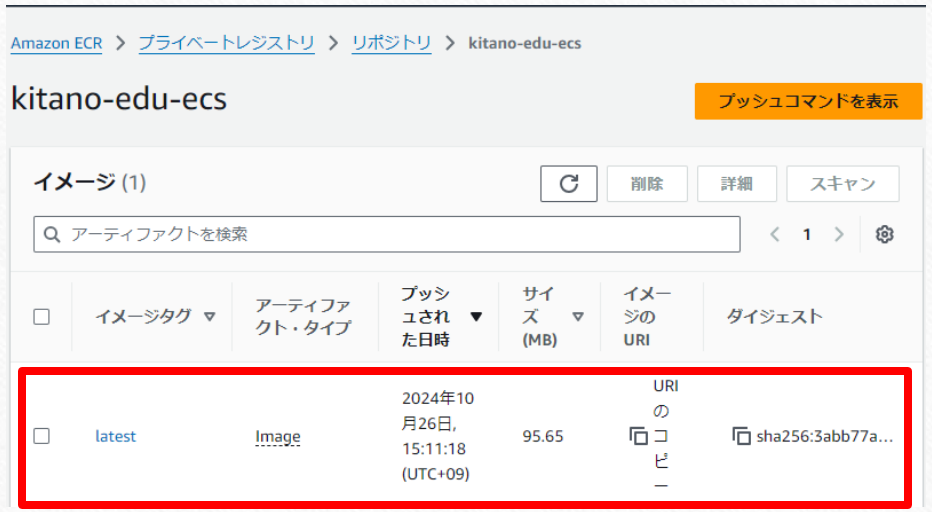
実際にやってみた step1, 2




```
PS C:¥work¥dokcer-ecr> docker push [redacted].dkr.ecr.ap-northeast-1.amazonaws.com/kitano-edu
-ecs:latest
The push refers to repository [redacted].dkr.ecr.ap-northeast-1.amazonaws.com/kitano-edu-ecs
]
64d40d22ea50: Pushed
7e0a21823581: Pushed
44f5fc899462: Pushed
a46a5fb872b5: Pushed
latest: digest: sha256:3abb77a6e693307842ade82f8b9566d7acc0124cd8edc863a827b25d16394361 size: 1
155
PS C:¥work¥dokcer-ecr>
```

実際にやってみた step1, 2





実際にやってみた step3, 4

Step3注意点

Step4注意点

実際にやってみた step3, 4

実際にやってみた step5

実際にやってみた step5

環境 AWS Fargate

既存のクラスター

kitano-edu-ecs-fagate

▼ コンピューティング設定 (アドバンスト)

コンピューティングオプション 情報

コンピューティングタイプ間でタスクを分散させるには、適切なコンピューティングオプションを使用します。

○ キャパシティープロバイダー戦略
1 つ以上のキャパシティープロバイダーにタスクを分散する起動戦略を指定します。

◉ 起動タイプ
キャパシティープロバイダー戦略を使用せずに、タスクを直接起動します。

起動タイプ 情報

マネージドキャパシティー (Fargate) またはカスタムキャパシティー (EC2 またはユーザーマネージド、External インスタンス) のいずれかを選択します。External インスタンスは ECS Anywhere 機能を使用してクラスターに登録されます。

FARGATE

プラットフォームバージョン 情報

サービスを実行するプラットフォームバージョンを指定します。

LATEST

アプリケーションタイプ 情報

実行するアプリケーションのタイプを指定します。

◉ サービス
停止して再起動できる長時間のコンピューティング作業を処理するタスクグループを起動します。例としては、ウェブアプリケーションが挙げられます。

○ タスク
実行および終了するスタンドアロンタスクを起動します。例としては、バッチジョブが挙げられます。

タスク定義

既存のタスク定義を選択します。新しいタスク定義を作成するには、タスク定義 にアクセスしてください。

☐ リビジョンの手動指定
選択したタスク定義ファミリーに最新の 100 個のリビジョンを選択する代わりに、リビジョンを手動で入力します。

ファミリー

kitano-edu-ecs-task

リビジョン

1 (最新)

サービス名

このクラスターに一意のサービス名を割り当てます。

kitano-edu-ecs-service

最大 255 文字のアルファベット (大文字と小文字)、数字、アンダースコア、ハイフンを使用できます。サービス名はクラスター内で一意である必要があります。

マイアプリのダッシュボー... Microsoft Edge

Hello World!

最後に

所感

- EC2よりもリリースに関する設定で考えることが少なくて良い
- 想像してるよりも数倍簡単に公開までもっていける
- Code Pipeline / Github Actions等を駆使するとリリース作業が
  全て自動化できるので、できれば其処らをこのLTに組み込みたかった