DevAx connect

Amazon ECS で考える安全なデプロイメント

DevAx::connect シーズン 3 rethink CI/CD

Kyosuke Ochimizu
Specialist Solutions Architect, Containers

@otty246

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

自己紹介

落水 恭介 (Ochimizu Kyosuke)

ソリューションアーキテクト

➢ SIer
➢ 教育業界ベンチャー
➢ Cloud Integrator
➢ サポートチーム / Amazon Web Services Japan
➢ 現在のロール

好きな AWS サービス: Amazon Elastic Container Service (Amazon ECS)

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

セッションのゴール

アプリケーションの “安全な” デプロイを実現する

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

“安全な” デプロイとは

"安全な" デプロイとは？

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デプロイにおける "安全" を考える

本セッションでの定義

意図した変更が本番環境にデプロイされること

意図しない状況が発生した場合の影響を最小限にできること

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デプロイにおける “安全” を考える

意図した変更が本番環境にデプロイされること

開発者 → コードリポジトリ → ビルド/デプロイツール → 本番環境

```
$ git commit -m “add - points service”

$ git push origin main
```

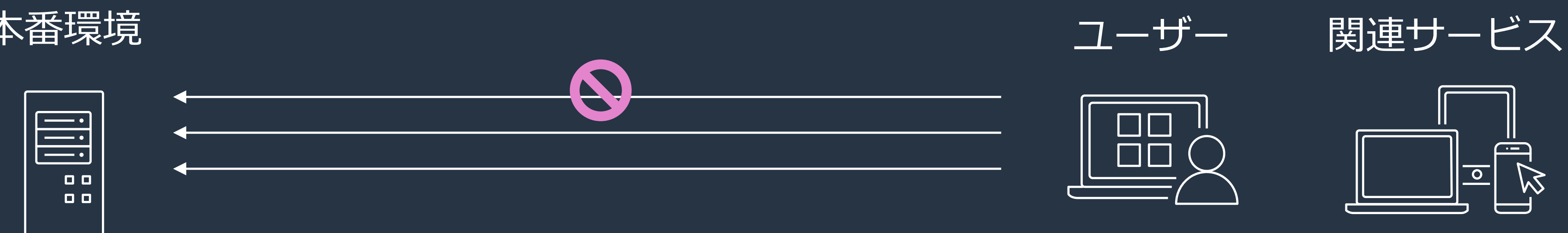
x,xxx 円

yyy ポイント付与

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デプロイにおける "安全" を考える

意図しない状況が発生した場合の影響を最小限にできること



サーバー側

- リソース使用率の増加
- パフォーマンス悪化

クライアント側

- レスポンスタイムの悪化
- 不具合の発生
  - エラー
  - バグ

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

"安全な" デプロイとは？

本セッションでの定義

- 意図した変更が本番環境にデプロイされること
- 意図しない状況が発生した場合の影響を最小限にできること

上記を達成するための機能やプロセスが組み込まれた

CI/CD パイプラインによるデプロイ

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

開発者の意図した変更をデプロイするには

デプロイ前の検証や動作確認を充実させる

静的解析

コードレビュー

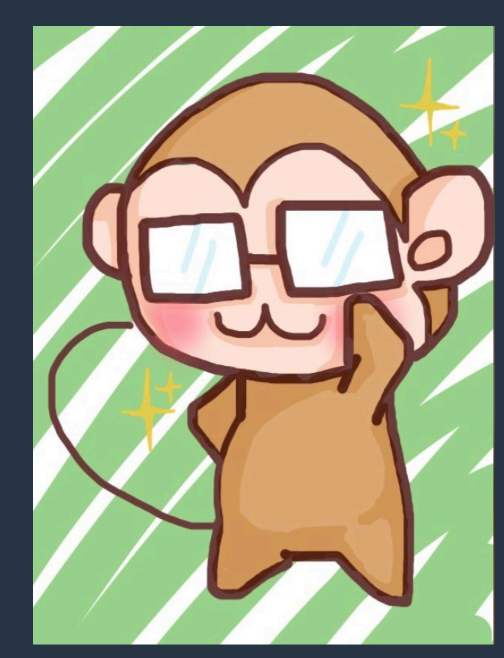
自動テスト

手動テスト

開発者 → コードリポジトリ → ビルド/デプロイツール → 本番環境

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

関連セッション



第5回 CI/CD なのだからちゃんとテストを書いてみよう ~ 分散環境のためのコンシューマ駆動契約を添えて ~

金森 政雄
アマゾンウェブサービスジャパン合同会社
ソリューションアーキテクト

個々のサービスが独立してデプロイ/リリースを行い、アジリティの高い開発を実現できることがマイクロサービスアーキテクチャが採用される理由の1つです。一方で、多数のサービスが協調して動作するマイクロサービスでは、各サービスを互換性を持って変更/運用していくことが難しくなります。 コンシューマ駆動契約は、サービス の利用者(コンシューマ)と提供者の間で仕様を"契約"として定義し、契約に基づいてテストすることで、サービス間連携の整合性を担保するプラクティスです。このセッションでは、コンシューマ駆動契約の基本的な考え方と、それをCI/CD に組み込んだ際の具体的な動きをデモも交えて解説します。

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

意図しない状況を最小限にするには

ロールバックの導入

本番環境

ユーザー

関連サービス

デプロイ

バージョン 1

バージョン 2

ロールバック

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ロールバックの導入

ロールバックの導入において考慮したいポイント

どうやってロールバックを実施する？

ロールバックを実施することの影響は？

いつロールバックをトリガーする？

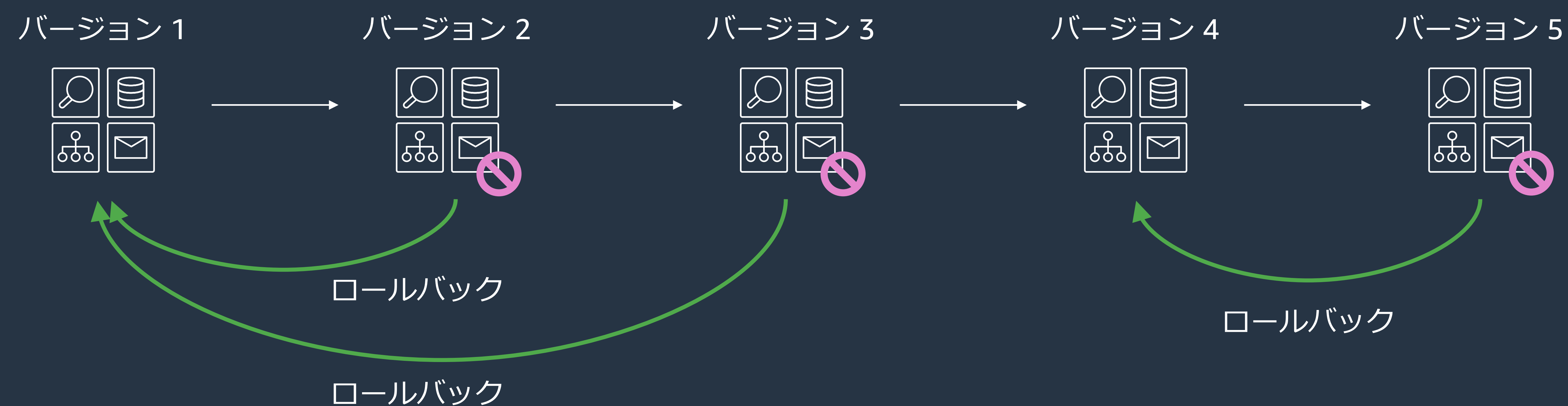
© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

どうやってロールバックを実施する？

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

そもそもアプリケーションのロールバックとは

安定稼働していた最後のバージョンをデプロイすること



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

アプリケーションのデプロイパターン

インプレースデプロイ

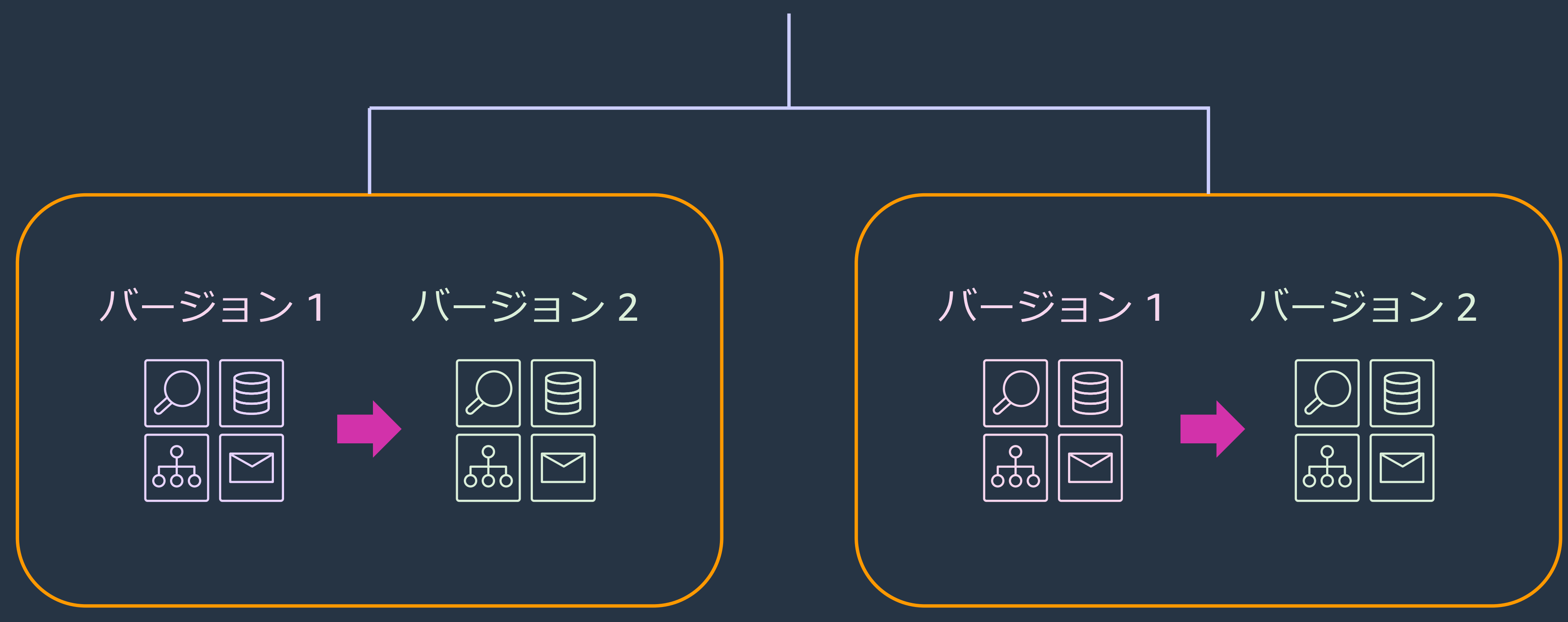
ローリングデプロイ

Blue/Green デプロイ

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

インプレースデプロイ

- 既存のインフラストラクチャのコンポーネントをそのまま利用
- コンポーネント内のアプリケーションを新しいバージョンに置き換える
- ロールバックのためには、稼働中のバージョン情報を保持しておく必要がある

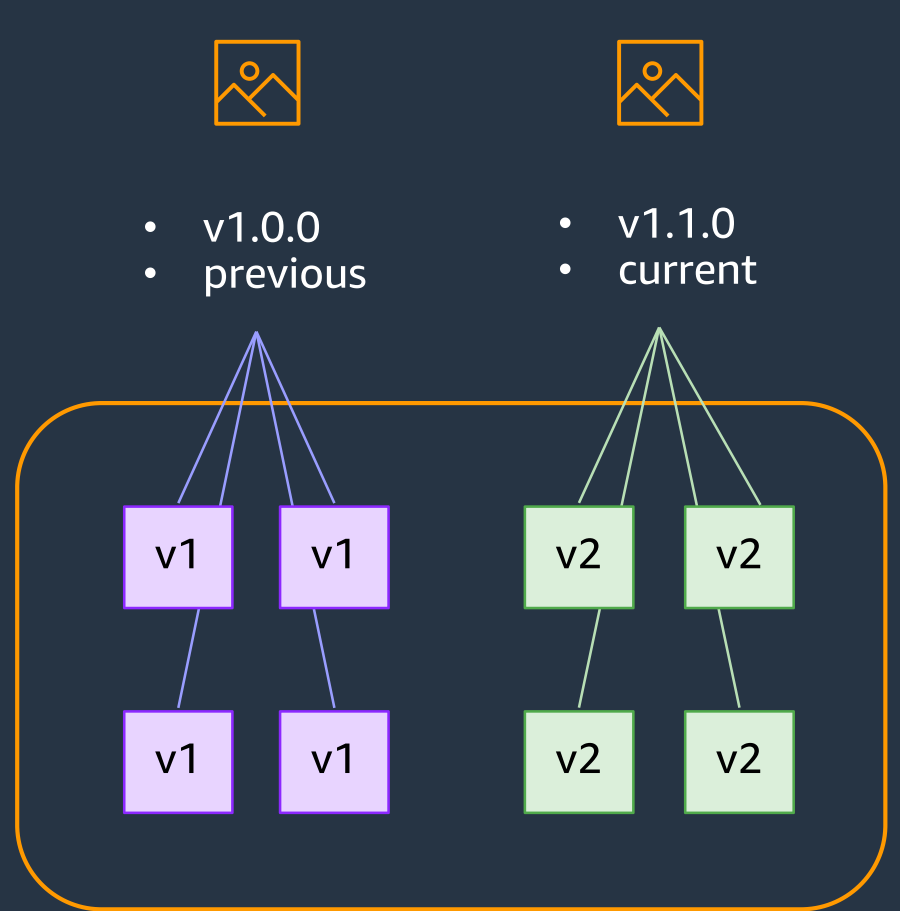


© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

インプレースデプロイのロールバック

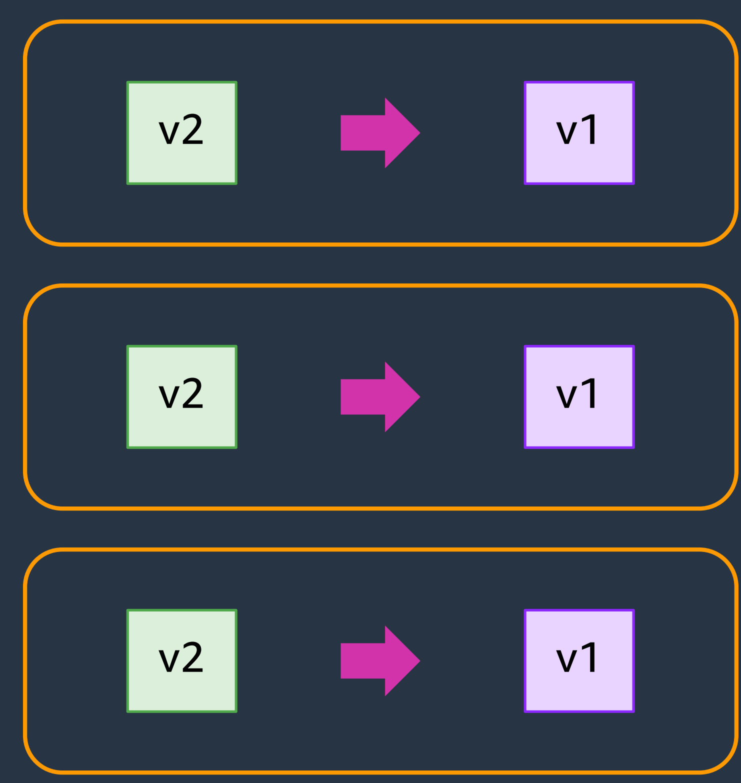
[バージョン情報]

アーティファクトのラベルで管理



[ロールバックの実施]

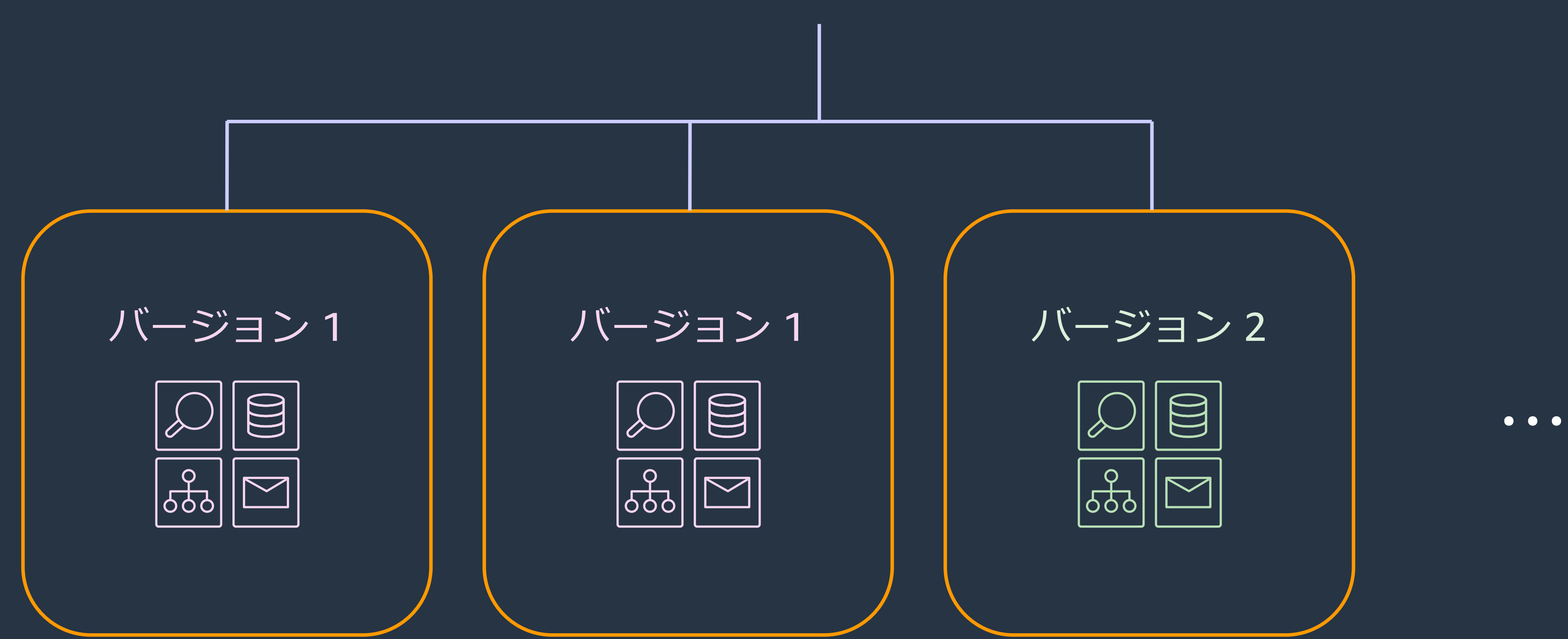
稼働中のバージョンを
既存のコンポーネントに再度デプロイ



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ローリングデプロイ

- 新規のインフラストラクチャのコンポーネントを作成
- 新しいコンポーネントへ段階的に置き換える
- ロールバックのためには、稼働中のバージョン情報を保持しておく必要がある

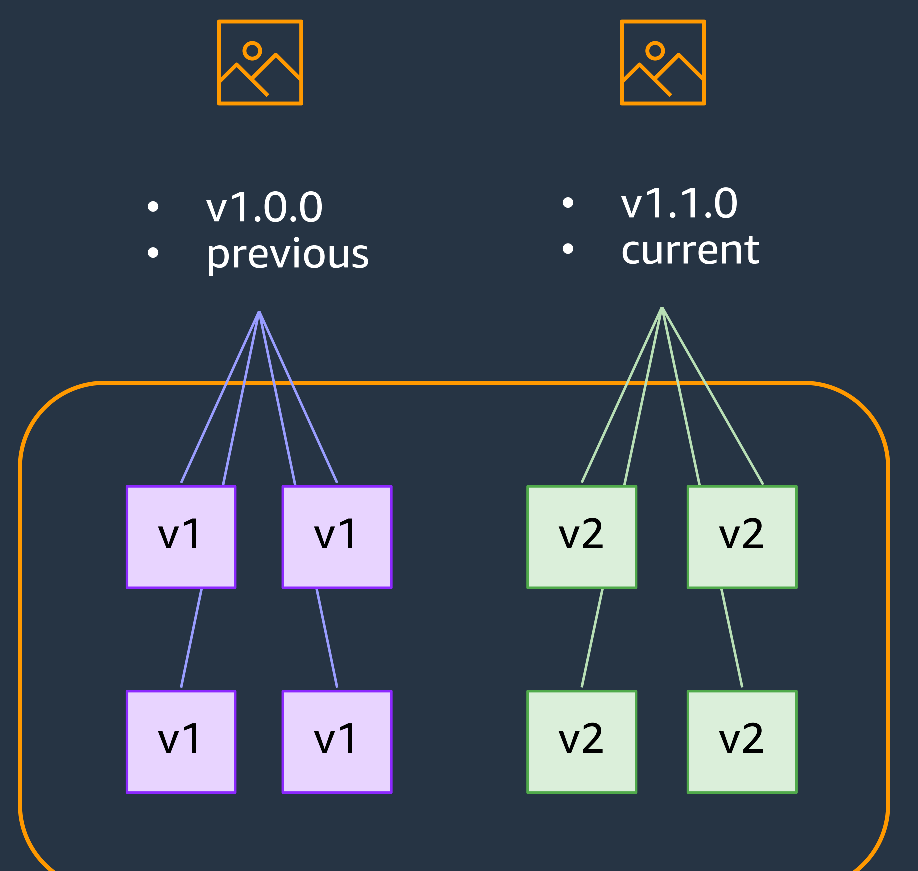


© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ローリングデプロイのロールバック

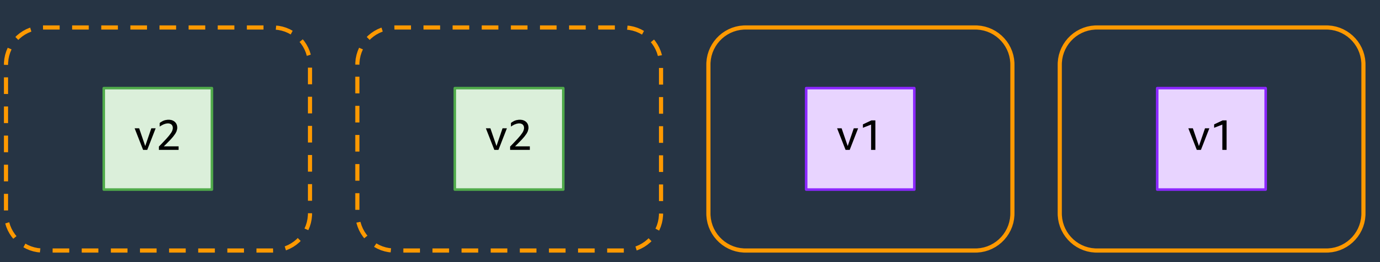
[バージョン情報]

アーティファクトのラベルで管理



[ロールバックの実施]

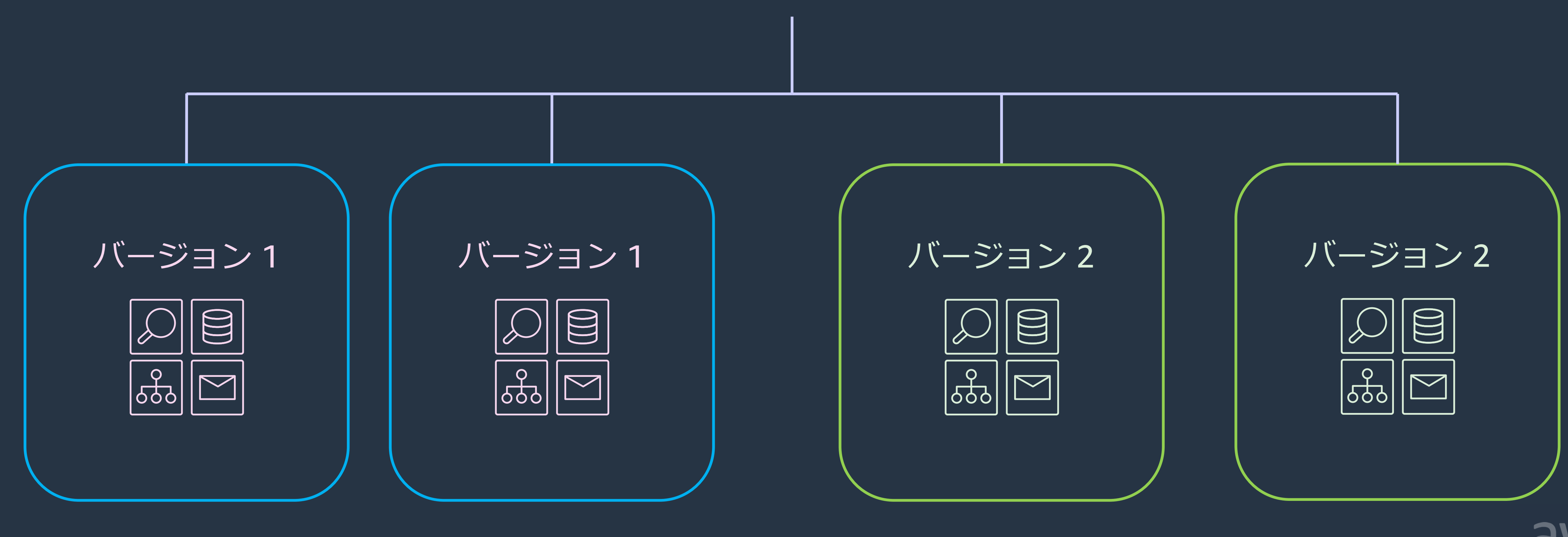
稼働中のバージョンを新しい
コンポーネントにデプロイして置き換え



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

Blue/Green デプロイ

- ➢ 新規のインフラストラクチャのコンポーネントを作成
- ➢ 新しいコンポーネントへ段階的 or 一度に置き換える
- ➢ ロールバックは Blue 環境に切り戻すだけ



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

aws DevAx connect

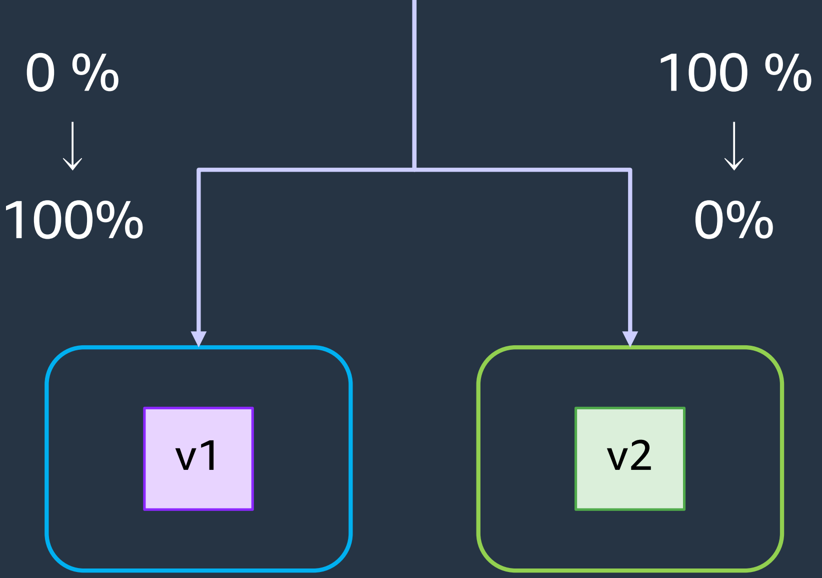
Blue/Green デプロイのロールバック

[バージョン情報]

Blue 環境として保持

[ロールバックの実施]

Green 環境のトラフィックを
Blue 環境に切り替える



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デプロイパターンごとのロールバック

パターン	バージョン情報	ロールバックの実施
インプレースデプロイ	アーティファクトのラベルで管理	稼働中のバージョンを既存の コンポーネントに再度デプロイ
ローリングデプロイ	アーティファクトのラベルで管理	稼働中のバージョンを新しい コンポーネントにデプロイして置き換え
Blue/Green デプロイ	Blue 環境として保持	Green 環境のトラフィックを Blue 環境に切り替える

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ロールバックの導入において考慮したいポイント

どうやってロールバックを実施する？

ロールバックを実施することの影響は？

いつロールバックをトリガーする？

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

いつロールバックをトリガーする？

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ロールバックのトリガーは “意図しない状況の発生”

「何が起こっているのか」を把握する能力がアプリに必要

ワーカーのジョブ処理状況は？

アプリの応答時間は？

API のエラー率は？

アプリケーション

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

オブザーバビリティ

アプリケーション、ユーザー、およびインフラストラクチャのインサイトを得て、パフォーマンスを向上させる

オブザーバビリティとは何?

「私のシステムは上か下か?」「私のエンドユーザーが体験するのは速いのか遅いのか?」「どのような KPI や SLA を設定すべきか、またそれらが満たされているかどうかをどのように確認するのか?」 クラウドのスピードとスケールで運用する場合、何も考えずに行動することはできません。これらのような運用上およびビジネス上の幅広い質問に答えられる必要があります。問題発生時には (理想的にはカスタマーエクスペリエンスを阻害する前に) その問題を発見し、迅速に対応し、可能な限り早く解決する必要があります。このようなインサイトを得るためには、観測可能なシステムが必要です。

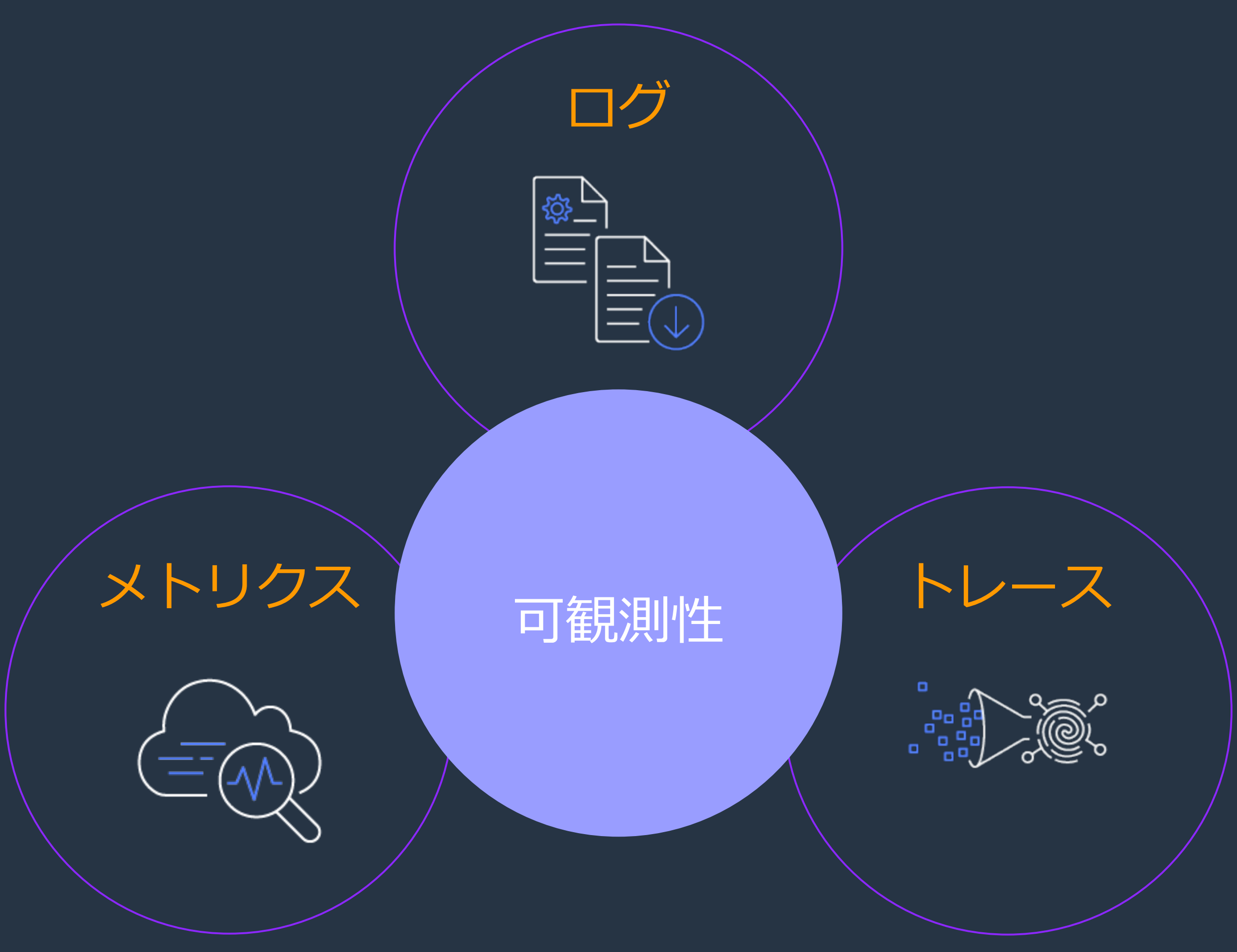
「私のシステムは上か下か?」「私のエンドユーザーが体験するのは速いのか遅いのか?」「どのような KPI や SLA を設定すべきか、またそれらが満たされているかどうかをどのように確認するのか?」 クラウドのスピードとスケールで運用する場合、何も考えずに行動することはできません。これらのような運用上およびビジネス上の幅広い質問に答えられる必要があります。問題発生時には (理想的にはカスタマーエクスペリエンスを阻害する前に) その問題を発見し、迅速に対応し、可能な限り早く解決する必要があります。このようなインサイトを得るためには、観測可能なシステムが必要です。

モニタリングとオブザーバビリティ | AWS マネジメントとガバナンス
https://aws.amazon.com/jp/products/management-and-governance/use-cases/monitoring-and-observability/

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

オブザーバビリティ (可観測性) とは

内部で「何が起こっているのか」を説明できるアプリケーションの能力



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

システムを観測可能にする 3 つの柱

メトリクス

- ある時点のなんらかのシステム状態を表現する 数値情報
- 一定間隔ごとの時系列データとして記録される
- 1 つ以上のディメンション/ラベルをメタデータとして持つ
- 例) CPU 使用率、リクエストレート、ストレージ残容量、など

ログ

- システム内で発生した イベント情報
- 各イベントが独立したレコードとして記録される
- メトリクスよりも多くの情報を含むケースが多い
- 例) アクセスログ、 エラー情報、など

トレース

- 1 つのトランザクションを複数システムで構成する フロー情報
- トランザクションごとにユニークな識別子をもって記録される
- システム間のやりとりに関するメタ情報付与も
- 例) とある HTTP リクエストの受け取りからレスポンスまで

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ロールバックとメトリクス・ログ・トレース

ロールバックにおいて重要なのは “メトリクス”

- ターゲットとなるメトリクスに基づき、意図しない状況をアラームで報告
- アラームのアクションとして自動ロールバックや運用担当者の呼び出しを実施

“ログ” と “トレース” も活用してロールバックをトリガーした原因を分析

- ロールバックが発生した原因は調査・対応が必要
- “メトリクス” に不足している情報を “ログ” や “トレース” で補完

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ターゲットとなるメトリクス

オンラインシステム (例: Web API) : RED メソッド

- Rate: 秒間リクエスト数
- Errors: 秒間の失敗したリクエスト数
- Duration: リクエストの処理時間の分布

オフラインシステム (例: ワーカージョブ) : USE メソッド

- Utilization: ジョブの進行ペース
- Saturation: キューに入っているジョブの数
- Errors: エラーイベントの数

a

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ロールバックの導入において考慮したいポイント

どうやってロールバックを実施する？

ロールバックを実施することの影響は？

いつロールバックをトリガーする？

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ロールバックを実施することの影響は？

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

一方向だけのドア (one way door)

破壊的な (後方互換性のない) 変更に対するロールバックは難しい

Amazon Simple Storage Service (Amazon S3)

.xml .xml .xml

.xml .xml .json

.xml .xml .json

バージョン 1	デプロイ →	バージョン 2	ロールバック →	バージョン 1
読み取り: XML 書き込み: XML		読み取り: XML, JSON 書き込み: JSON		読み取り: XML 書き込み: XML

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

2 フェーズデプロイ

ロールバック前後での破壊的変更を回避

Amazon Simple Storage Service (Amazon S3)

ロールバック後も JSON ファイルの読み込みが可能

.xml .xml .xml

.xml .xml .xml

.json .xml .xml

読み取り: XML
書き込み: XML

バージョン 1

準備フェーズ

読み取り: XML, JSON
書き込み: XML

バージョン 2

アクティブ化

読み取り: XML, JSON
書き込み: JSON

バージョン 3

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

2 フェーズデプロイの注意事項

準備フェーズのデプロイを確実に完了する

- すべてのアプリケーションに準備フェーズを適用後、アクティブ化を実施
- マイクロサービスなど分散アプリケーション間で足並みを揃える場合は特に注意

アクティブ化のデプロイ後、準備フェーズ以前にロールバックはできない

- v3 -> v1 など、準備フェーズとアクティブ化をまたぐ場合は破壊的変更となる
- 準備フェーズのデプロイ後は、影響がないかを見極める待機時間の設定が効果的

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ロールバックの安全性をテスト

変更前後のバージョンを共存させて動作確認

変更のデプロイ後、実際にロールバックを行い動作確認

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ロールバックの導入において考慮したいポイント

ロールバックを実施することの影響は？

いつロールバックをトリガーする？

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

まとめ: ロールバックの導入

どうやってロールバックを実施する？

- 安定稼働していた最後のバージョンをデプロイ
- デプロイパターンごとの、バージョン情報管理とロールバック実施方法を確認

いつロールバックをトリガーする？

- 意図しない状況の発生を把握し、アクションをトリガー
- オブザーバビリティを備えた上で、ターゲットとなるメトリクスを設定

ロールバックを実施することの影響は？

- 破壊的変更に対するロールバックは難しい
- 2 フェーズデプロイにより、ロールバック前後での破壊的変更を回避

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

CI/CD パイプラインと
自動ロールバック

CI/CD パイプラインに自動ロールバックを導入

デプロイの影響確認やロールバックの実施を自動化

Source → Build → Test → Production

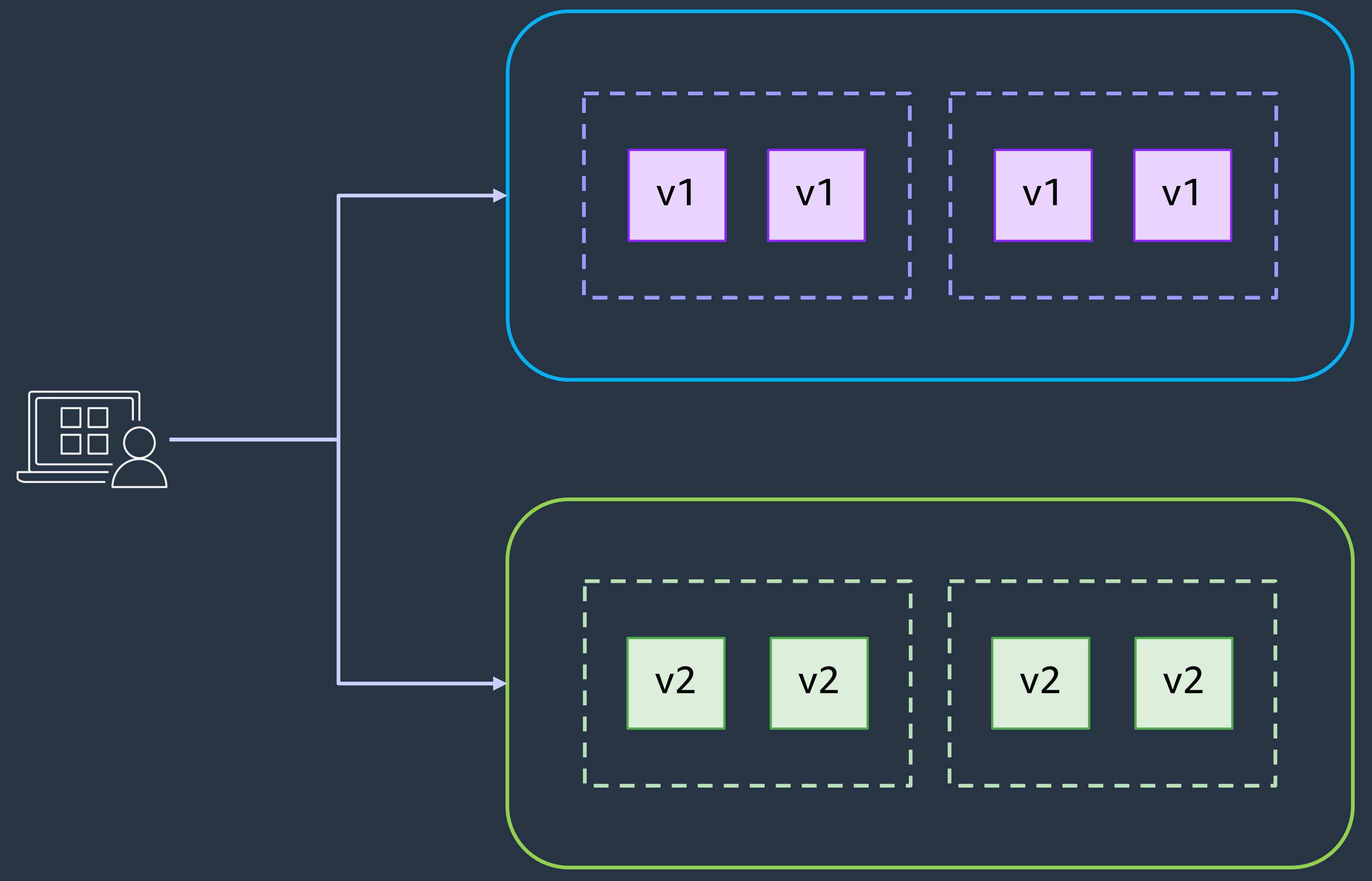
デプロイ　　メトリクスの監視

待機時間　　自動ロールバック

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

自動ロールバックの流れ (Blue/Green デプロイ)

1. 新しいバージョンのアプリケーション (Green 環境) をデプロイ



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

自動ロールバックの流れ

2. 一定時間、デプロイの影響を確認するために待機 & メトリクスを監視

v1 v1 v1 v1

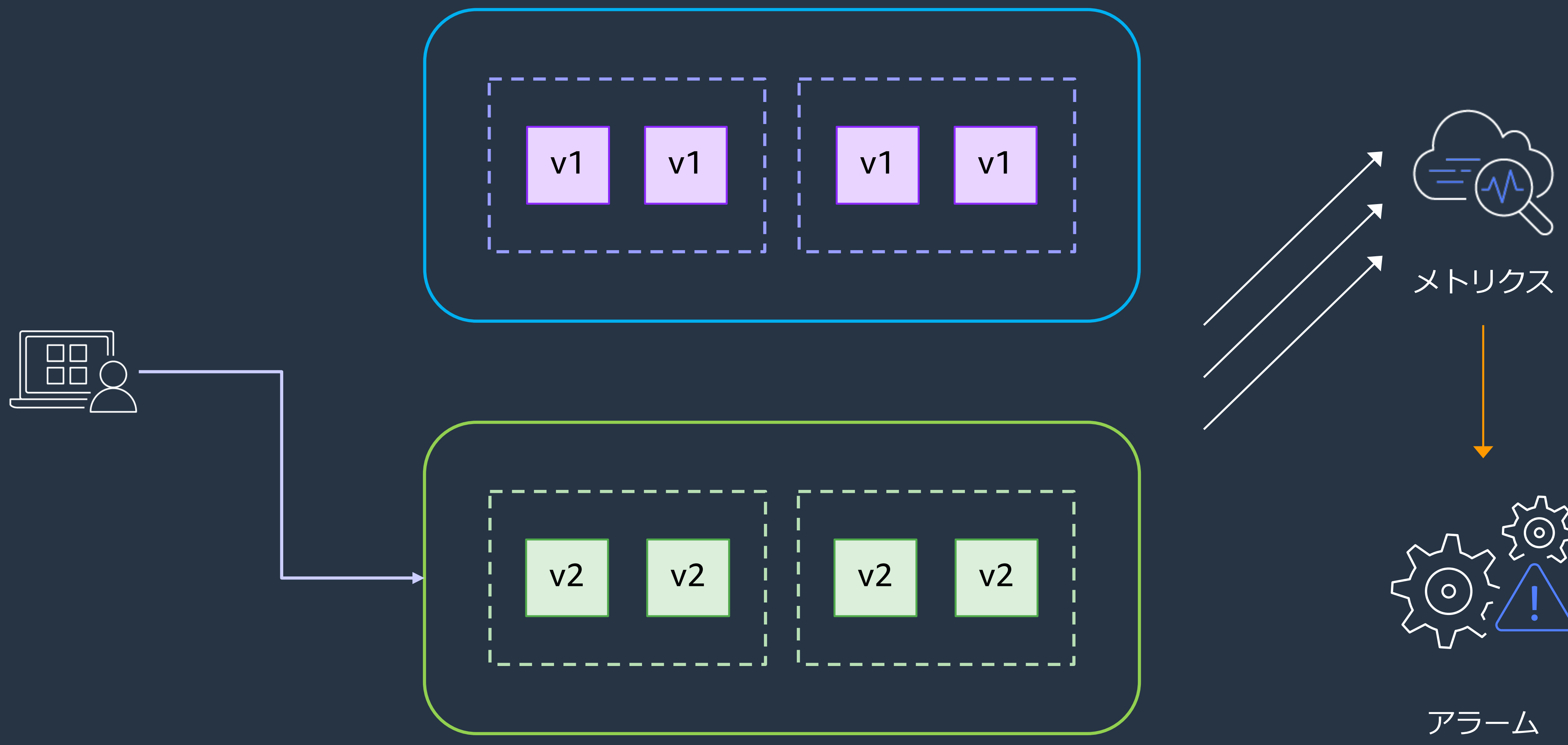
v2 v2 v2 v2

メトリクス

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

自動ロールバックの流れ

3. 問題が発生 & メトリクスに紐づくアラートを発行



メトリクス

アラーム

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

aws DevAx connect

自動ロールバックの流れ

4. Blue 環境にトラフィックを切り戻し

v1
v1
v1
v1

v2
v2
v2
v2

メトリクス

アラーム

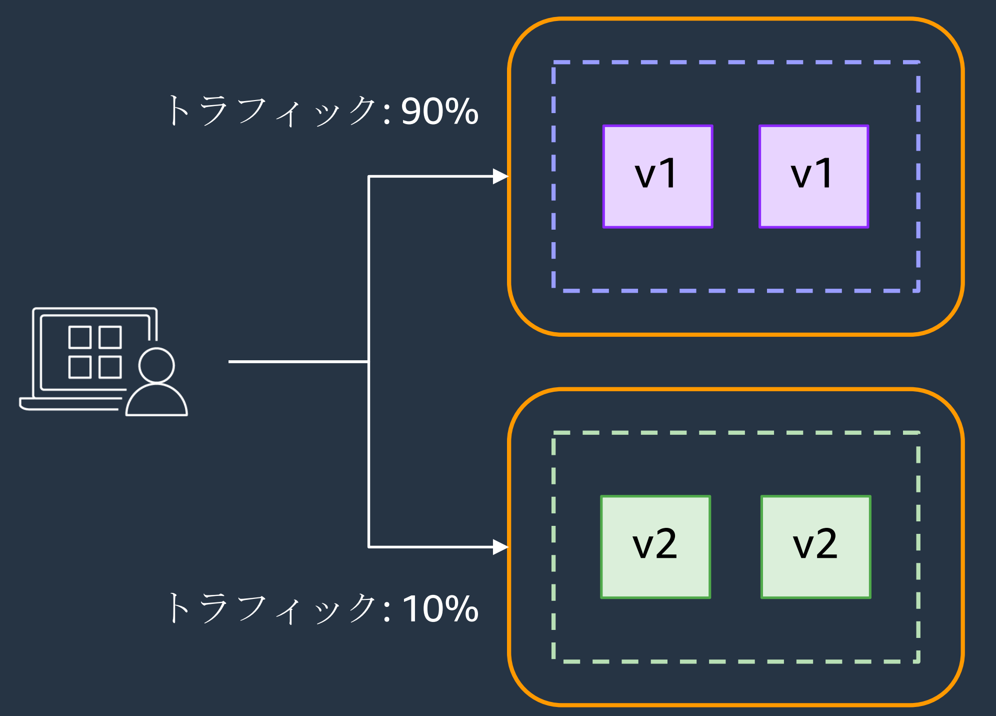
© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

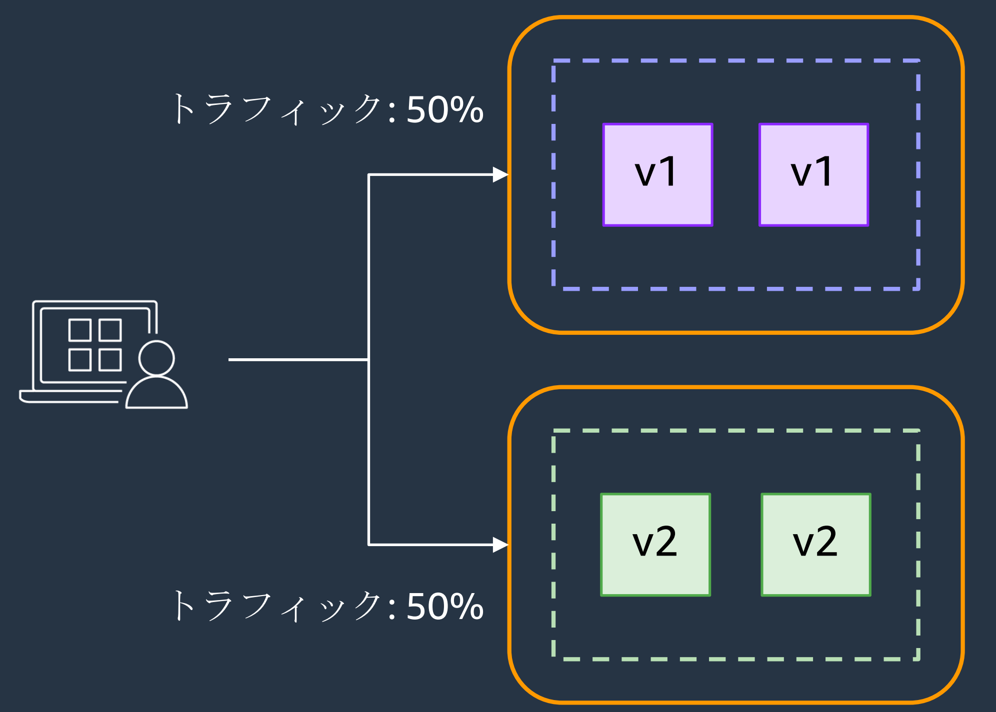
意図しない状況の影響範囲をより小さくするには？

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

新しいバージョンを “少しずつ” 公開する

影響を受けるユーザーやクライアントを最小限に





© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

プログレッシブデリバリー

新バージョンのアプリケーションを段階的に公開

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

Amazon / AWS のプログレッシブデリバリー

Prod

Source → Build → Test → Region A One Box → Region A Prod → Region B One Box → Region B Prod → Regions C-E One Box → Regions C-E Prod →

v2 | v1 v1 v1 v1

One-box | Fleet

本番環境へのデリバリー

複数環境に分割、Progressive にデリバリー

メトリクス監視、アラームで自動ロールバック

Amazon Builders' Library
https://aws.amazon.com/jp/builders-library/automating-safe-hands-off-deployments/

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

まとめ: CI/CD パイプラインと自動ロールバック

CI/CD パイプラインへの自動ロールバックの導入

- デプロイフェーズにロールバックのための処理を追加
- 待機時間、メトリクス監視、自動ロールバック処理

意図しない状況の影響範囲をより小さくするために

- アプリケーションを少しずつ公開する
- プログレッシブデリバリーの検討

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

ECS における "安全な" CI/CD パイプラインの実現例

本日のデモ

"安全な" デプロイのために

- 意図した変更が本番環境にデプロイされること
  - ➢ コードレビューやテストなど、デプロイ前の検証や動作確認を充実させる
- 意図しない状況が発生した場合の影響を最小限にできること
  - ➢ CI/CD パイプラインに自動ロールバックを導入
  - ➢ プログレッシブデリバリーで影響範囲をより小さく

ECS に対して上記を実現する CI/CD パイプラインの例をお見せします

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

Black Belt

Amazon Elastic Container Service (Amazon ECS)

パワフル & シンプル

- クラウドでコンテナを本番環境利用するためのオーケストレーター
- 他の AWS サービスと高度に連携しコンテナを大規模に実行
- フルマネージドなコントロールプレーン
- 多様なワークロードをサポートする「タスク」「サービス」というシンプルなリソース表現

https://aws.amazon.com/jp/blogs/news/aws-black-belt-online-seminar-con201-ecs-overview/

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

Amazon ECS の動作イメージ (on Fargate)

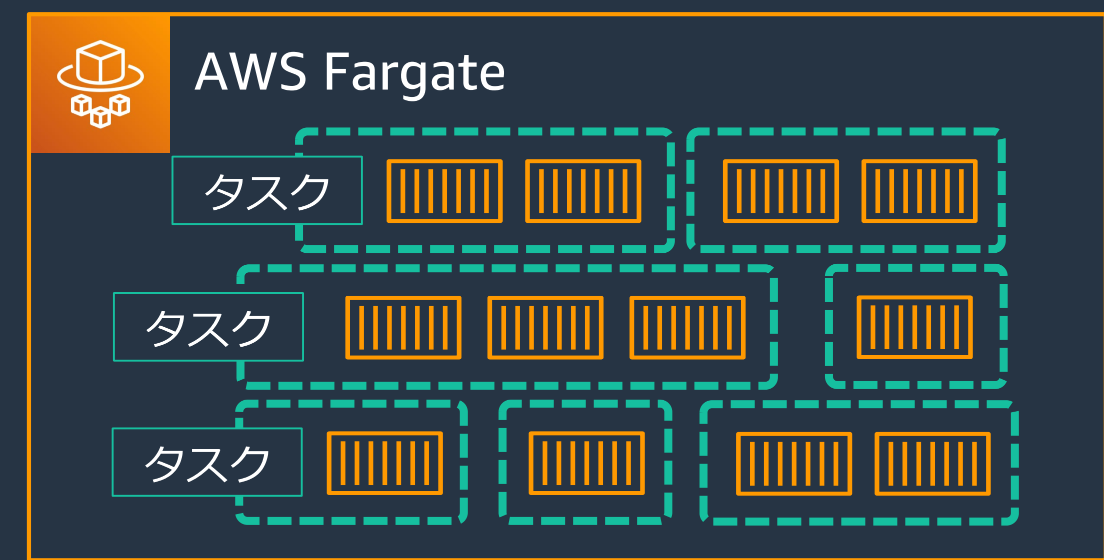
Black Belt

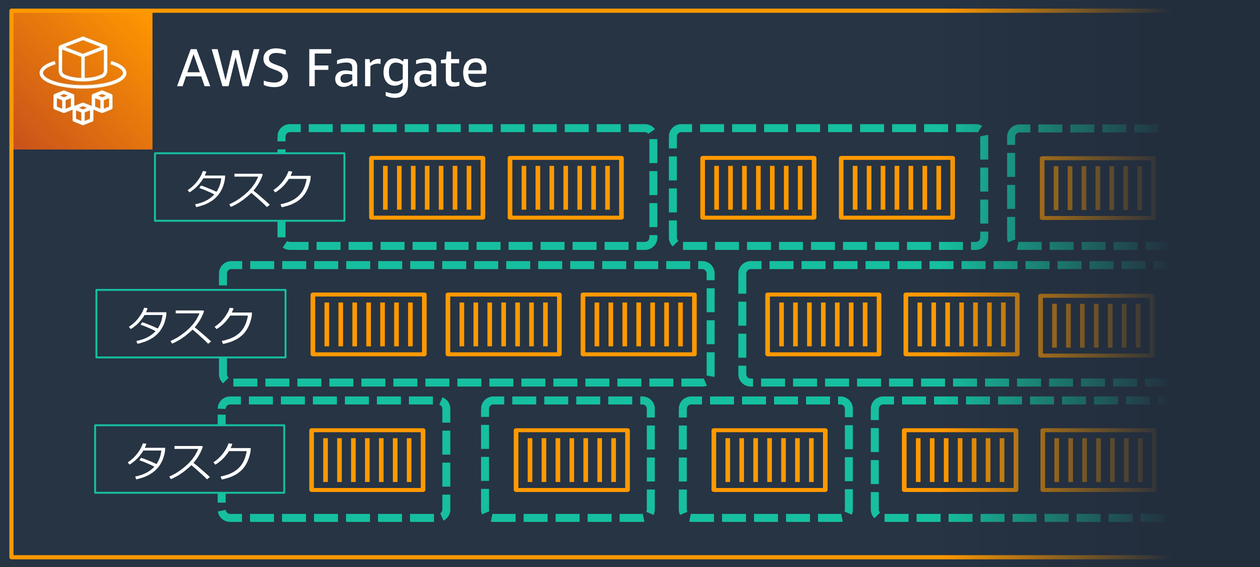
Amazon ECS

スケジューリングとオーケストレーション

クラスター管理 | 配置エンジン

クラスター





https://aws.amazon.com/jp/blogs/news/aws-black-belt-online-seminar-con202-ecs-fargate-overview/

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

aws DevAx connect

デモ環境の概要

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デモ環境の概要

AWS Cloud

線形デプロイ、アラームを検知してロールバック

AWS CodePipeline

AWS CodeCommit

AWS CodeBuild

AWS CodeDeploy

Amazon CloudWatch

Amazon Elastic Container Registry (Amazon ECR)

VPC

レスポンスタイムの増加でアラーム発生

AWS Cloud9

Amazon Elastic Container Service (Amazon ECS)

Elastic Load Balancing

サンプルアプリ

CloudWatch エージェント

メトリクスの収集 & 保存

負荷ツール

N 回 / 秒でリクエストを送信

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デモのシナリオ

1. サンプルアプリ (API) にレートリミットを追加

AWS Cloud

AWS CodePipeline

AWS CodeCommit

AWS CodeBuild

AWS CodeDeploy

Amazon CloudWatch

Amazon Elastic Container Registry (Amazon ECR)

VPC

AWS Cloud9

Amazon Elastic Container Service (Amazon ECS)

Elastic Load Balancing

サンプルアプリ

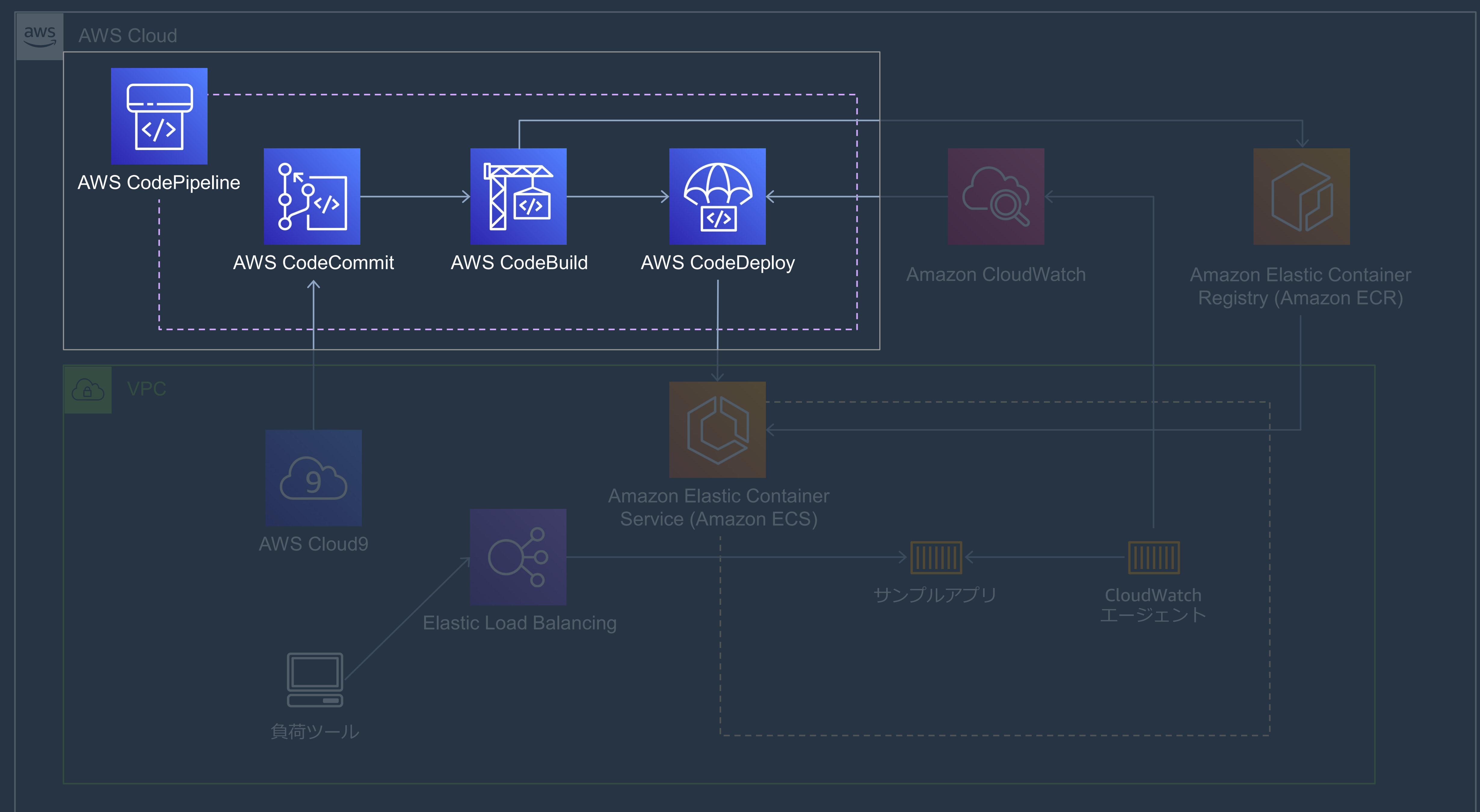
CloudWatch エージェント

負荷ツール

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デモのシナリオ

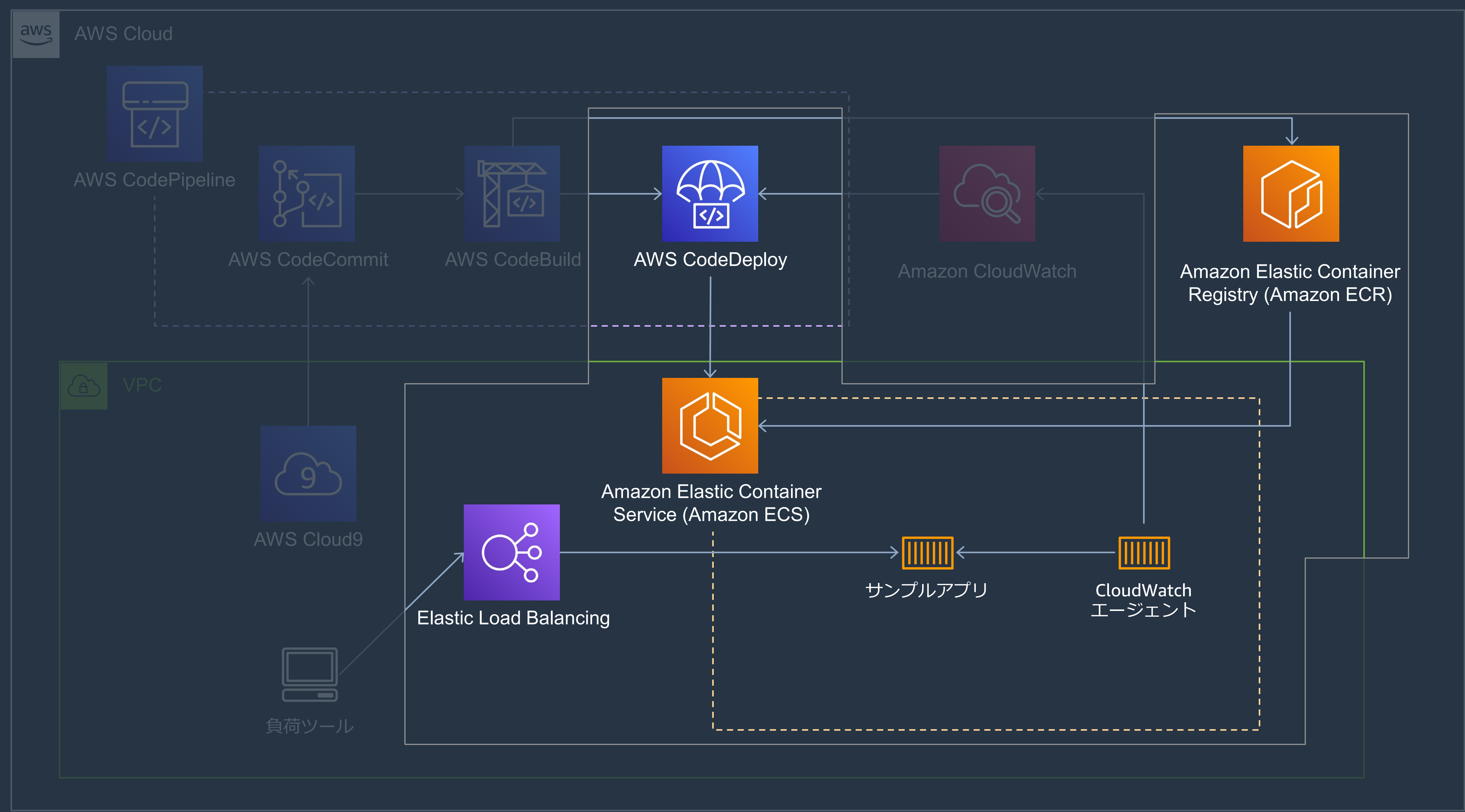
2. CodeCommit に Push してパイプラインを実行



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

aws DevAx connect

デモのシナリオ 3. CodeDeploy によりトラフィックを徐々に移行



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

aws DevAx connect

デモのシナリオ

4. Green 環境のトラフィック増加によりアラームが発生

AWS Cloud

AWS CodePipeline

AWS CodeCommit

AWS CodeBuild

AWS CodeDeploy

Amazon CloudWatch

Amazon Elastic Container Registry (Amazon ECR)

VPC

AWS Cloud9

Amazon Elastic Container Service (Amazon ECS)

Elastic Load Balancing

サンプルアプリ

CloudWatch エージェント

負荷ツール

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デモのシナリオ

5. アラームにより CodeDeploy がロールバックを実行

AWS Cloud

AWS CodePipeline

AWS CodeCommit

AWS CodeBuild

AWS CodeDeploy

Amazon CloudWatch

Amazon Elastic Container Registry (Amazon ECR)

VPC

AWS Cloud9

Amazon Elastic Container Service (Amazon ECS)

Elastic Load Balancing

サンプルアプリ

CloudWatch エージェント

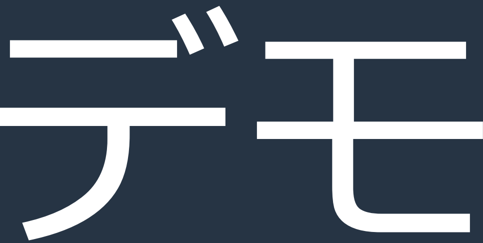
負荷ツール

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

デモのシナリオ

1. サンプルアプリ (API) にレートリミットを追加
   - ➢ 溢れたリクエストのレスポンスタイムが増加していく仕組み
2. CodeCommit に Push してパイプラインを実行
3. CodeDeploy によりトラフィックを徐々に移行
   - ➢ 線形デプロイの利用
4. Green 環境のトラフィック増加によりアラームが発生
   - ➢ レスポンスタイムの平均がターゲットメトリクス
5. アラームにより CodeDeploy がロールバックを実行

© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

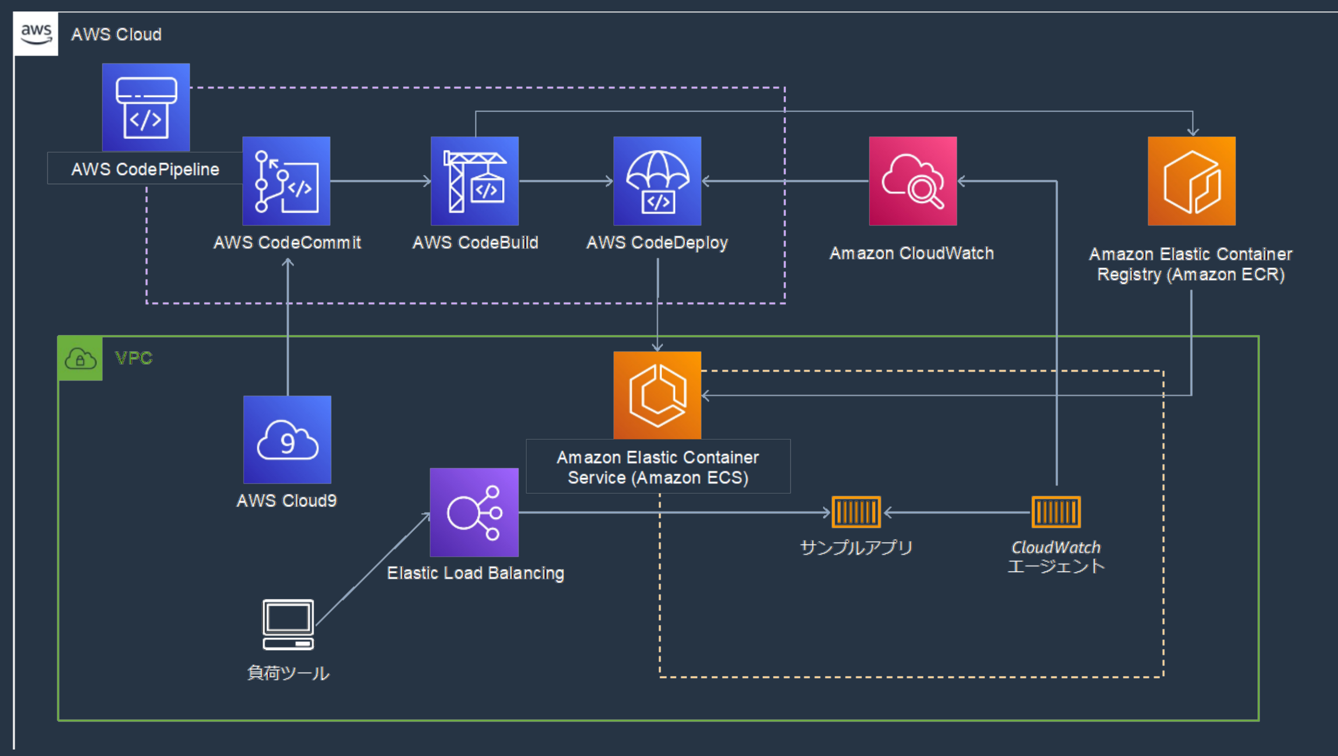


© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

まとめ: ECS における "安全な" CI/CD パイプライン

デモ: CodeDeploy を利用した Blue/Green デプロイ

- 線形デプロイによる段階的なトラフィックの移行
- CloudWatch アラームと連携した自動ロールバック



© 2022, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.

Thank you!

Kyosuke Ochimizu
Specialist Solutions Architect, Containers
AWS Japan
@otty246

© 2021, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Amazon Confidential and Trademark.