



aws SUMMIT

TOKYO | APRIL 20-21, 2023

ミッションクリティカルシステムを AWS に載せるには？

小田 圭二

アマゾン ウェブ サービス ジャパン合同会社

プロフェッショナルサービス本部 クラウドインフラストラクチャアーキテクト

自己紹介

名前：小田 圭二

所属：

プロフェッショナルサービス本部

クラウドインフラストラクチャアーキテクト

プロフィール：

外資系ソフトウェアベンダーでミッションクリティカルなシステムのコンサルティングに長年従事。AWS においてもミッションクリティカルシステムのコンサルティングを担当

好きな AWS サービス：
(AWS サービスではなく) 静的安定性



アジェンダ

- アーキテクチャの検討と可用性
- 静的安定性とは
- リソースレベルの可用性
- AWS サービスレベルの可用性
- リージョンを越える範囲の可用性
- まとめ

持って帰っていただきたいこと

- 静的安定性という AWS 内で培われた高可用性の特性(Characteristic)があり、AWS サービスの可用性を支えているだけでなく、ユーザー側でも活かせること
- AWS 内の静的安定性、マルチAZなどの冗長化、ユーザー側の静的安定性の組み合わせでかなりの障害ケースに対応できること
- 静的安定性はどのようなシステムにとっても重要な特性である

本日説明しないこと

- AWS サービスの一般的な説明

ミッションクリティカルシステムと AWS

「ミッションクリティカルシステム」と聞いたときにオンプレミスを無意識にイメージした方、手を挙げていただけないでしょうか？



アーキテクチャの検討と可用性

アーキテクチャの検討と可用性

静的安定性とは

リソースレベルの可用性

AWS サービスレベルの可用性

リージョンを越える範囲の可用性

まとめ

システム企画や方式設計とアーキテクチャ



機要件

機要件(プロセス)
機要件(データ)
機要件(インター
フェイス)

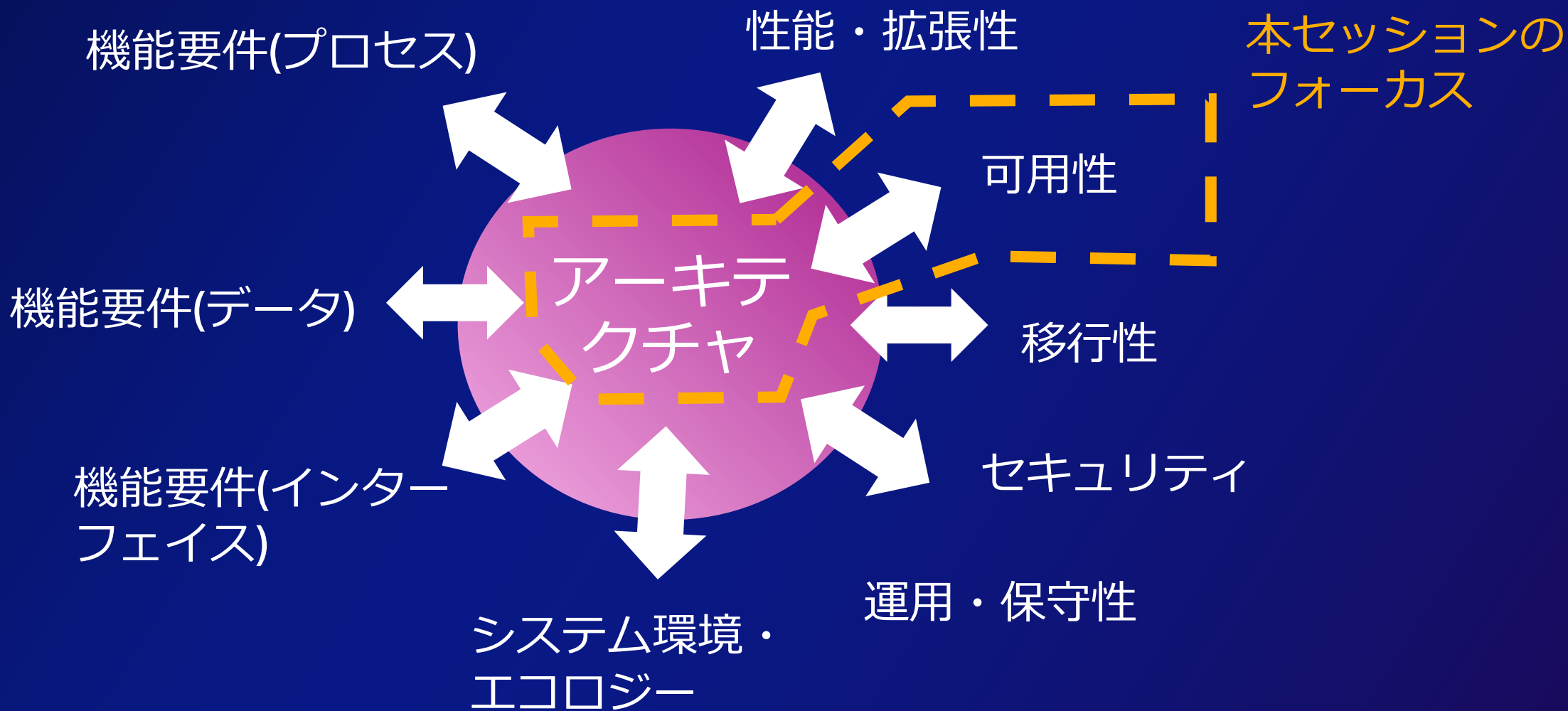
非機要件

可用性
性能・拡張性
運用・保守性

移行性
セキュリティ
システム環境・
エコロジー

横串で関係するのが
アーキテクチャ

本日のフォーカスポイント



ミッションクリティカルとは

ミッションクリティカル システム



重要なサービスを提供し続ける



一般に **可用性** が重視される

可用性観点でアーキテクチャを評価する

ミッションクリティカルシステムを中心に、可用性観点でのアーキテクチャ評価は様々なものが行われてきました。

CFIA : Component Failure Impact Analysis

ATAM : Architecture Tradeoff Analysis Method

システムのRCM : Risk Control Matrix

etc.

システムを部分(コンポーネント)に分解し、必要な対策を検討します。コンポーネントのレベルは様々です。たとえば、AWS ではアベイラビリティゾーン(AZ)といった単位で検討することやインスタンスといった単位で検討することもあります。

検討するシステムコンポーネント一覧

コンポーネント

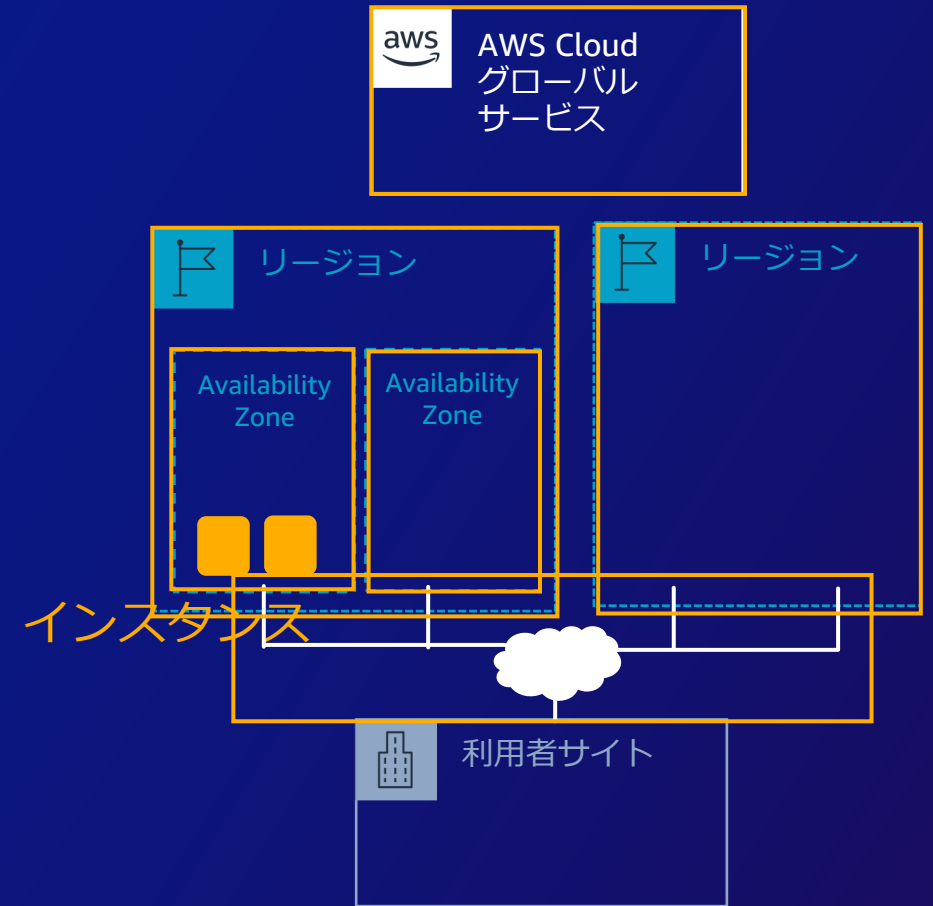
インスタンス(リソース)

AWS サービス(ゾーナルサービス)

AWS サービス(リージョナルサービス)

AWS サービス(グローバルサービス)

ネットワーク



これらに対して、どのようなアプローチが有効でしょうか？

静的安定性とは

アーキテクチャの検討と可用性

静的安定性とは

リソースレベルの可用性

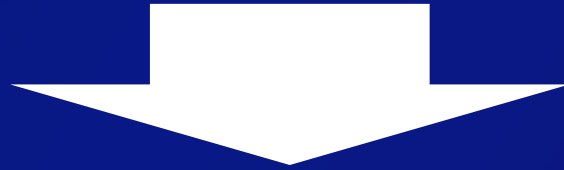
AWS サービスレベルの可用性

リージョンを越える範囲の可用性

まとめ

静的安定性(Static stability)

静的安定性とは、「依存関係が損なわれたとしても、設定変更が必要になることもなく、通常と同じくシステムが稼働を続ける」特性です。



他のサービスが障害になっても、システムが影響を受けずに稼働する。
障害時に設定変更する必要がなく、何もしなくても稼働を続ける。

静的安定性

可用性は重要です。静的安定性は、そのために AWS 内部で長年かけて培ったものです。

AWS 利用者の
メリット

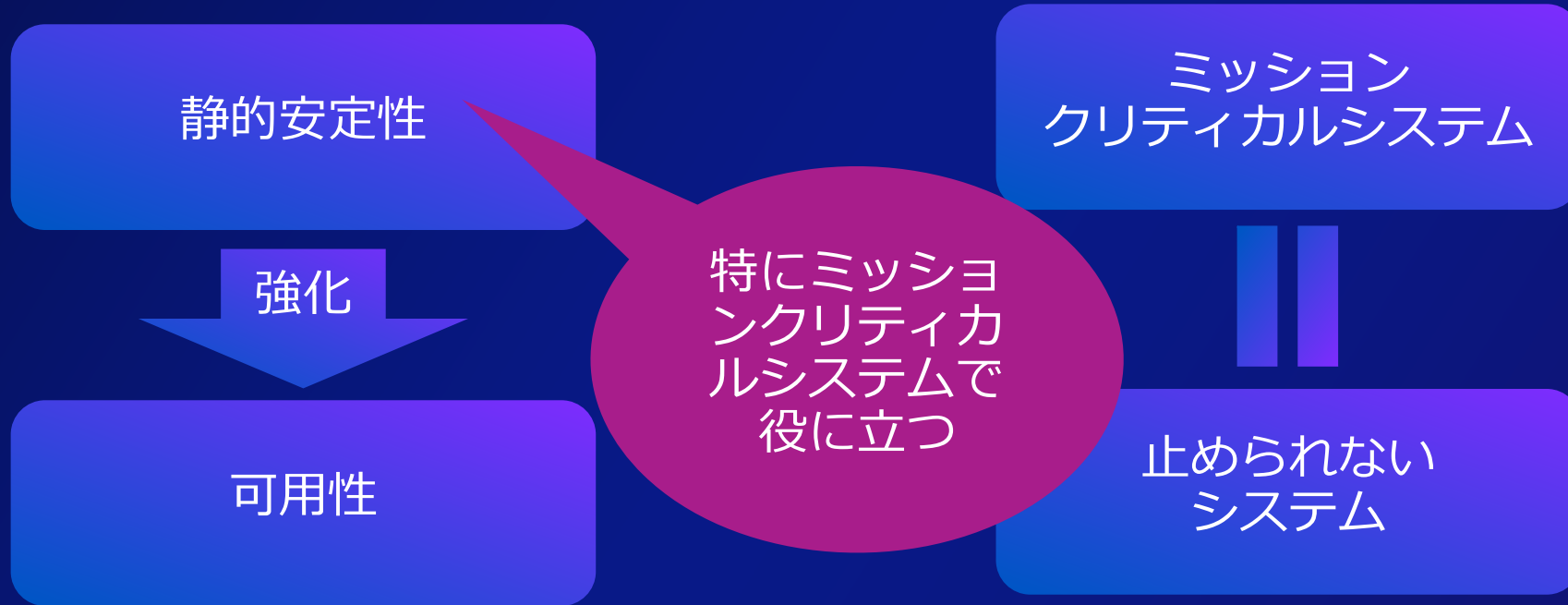
AWS サービスの
可用性

システムの
可用性 Up

手作業不要、
運用自動化

など

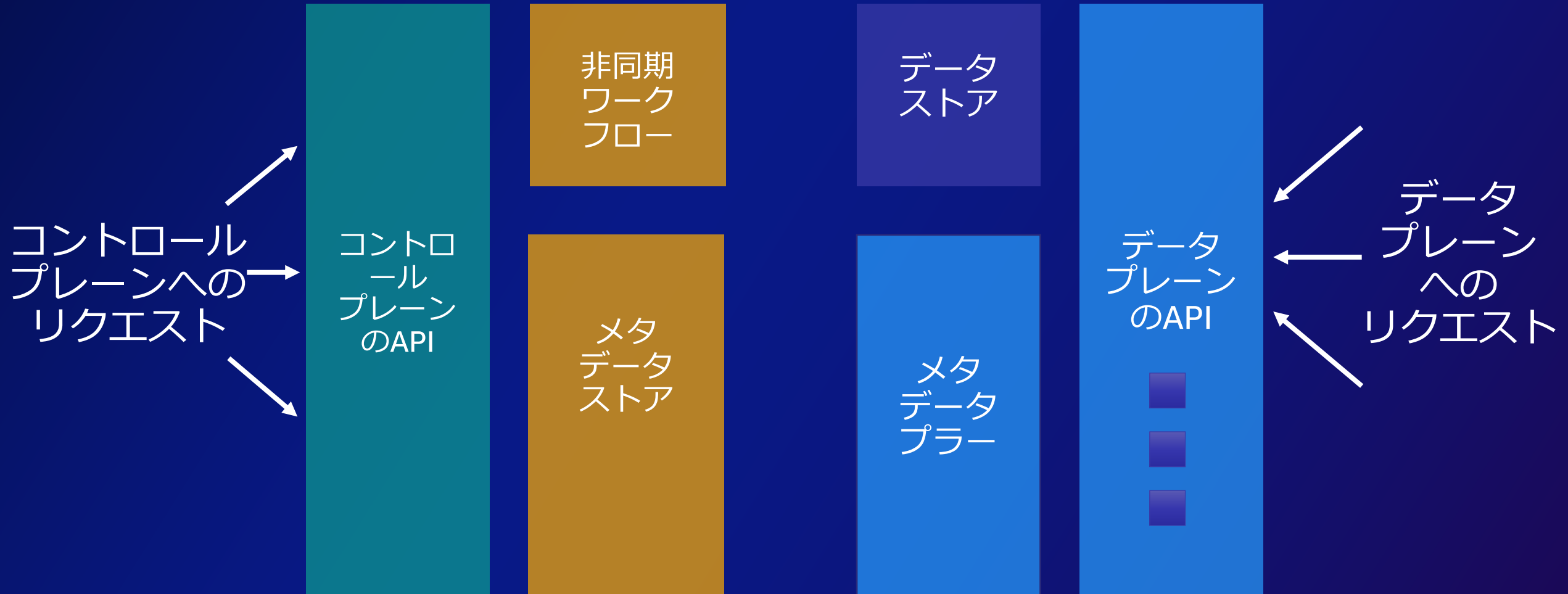
静的安定性とミッションクリティカルシステム



本セッションでは、AWS サービス内部の静的安定性を「AWS 内の静的安定性」と呼び、ユーザーが使う静的安定性を「ユーザーの静的安定性」と呼ぶことにします。

静的安定性の前提となる AWS アーキテクチャの紹介

AWS サービスのほとんどは、コントロールプレーンとデータプレーンで構成されています。



コントロールプレーンとデータプレーンの連携

Amazon Elastic Compute Cloud (Amazon EC2) の例で連携を説明します。

①例として
Amazon
EC2を
CreateするAPIを
実行

②コント
ロールプ
レーンで
処理

非同期
ワーク
フロー

メタ
データ
ストア

③データ
プレーン
に連絡が
行く

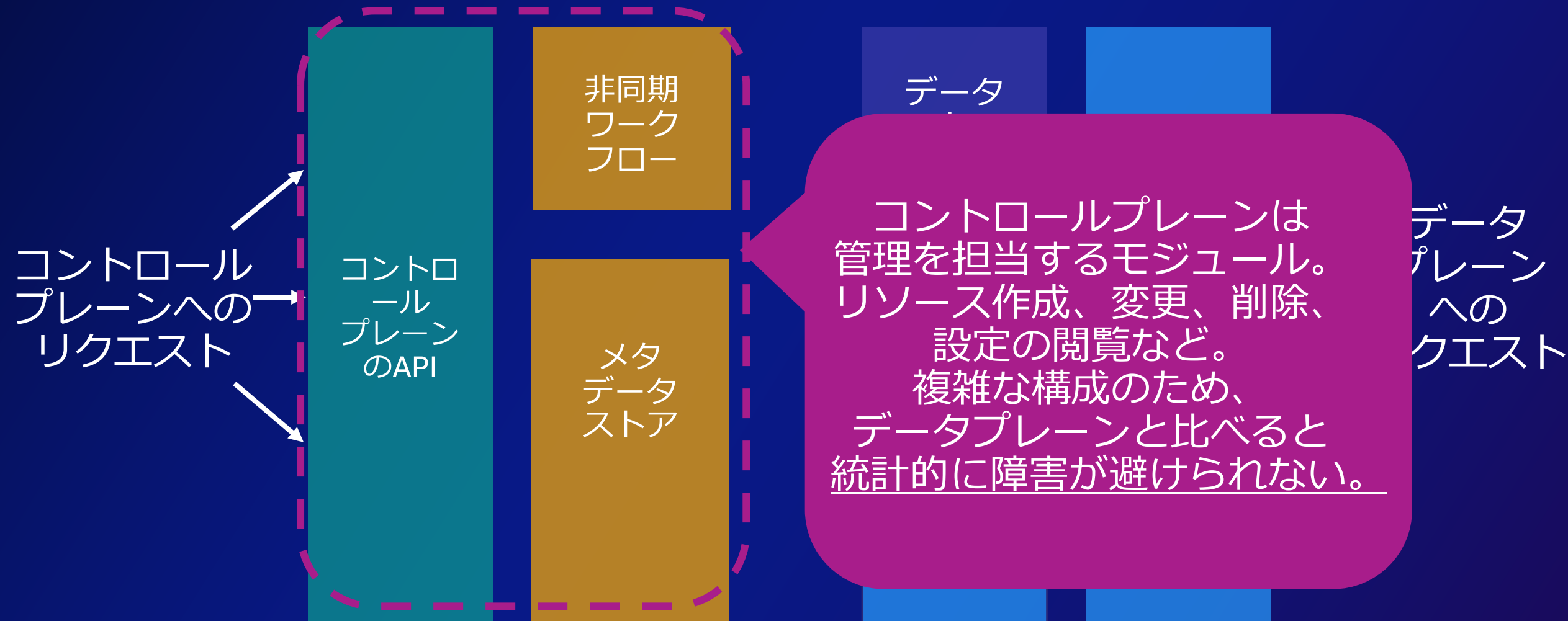
データ
ストア

メタ
データ
プレー

④データ
プレーン
内に
Amazon
EC2が立
ち上がる

⑤
Amazon
EC2が処
理を続け
る(稼働
する)

コントロールプレーンとは



データプレーンとは

非同期
ワーク

データプレーンはリソース稼働のためのモジュール。
作成済みのリソースが普段の処理を行う。
意図的にシンプルな構成になっていて、統計的に障害が発生しにくい。

データ
ストア

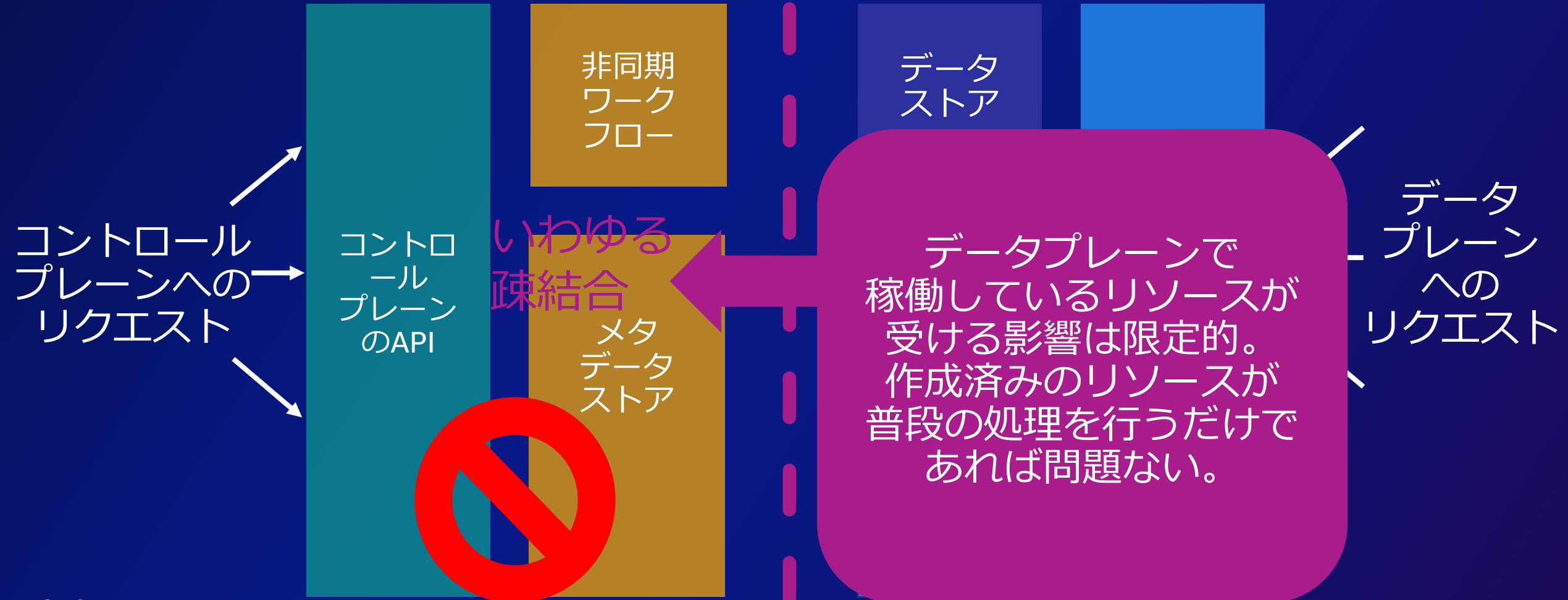
メタ
データ
プレー

データ
プレーン
のAPI

データ
プレーン
への
リクエスト

コントロールプレーンとデータプレーンは疎結合

コントロールプレーンが障害になっても、作成済みのリソースが普段の処理を行うことができます。



AWS サービスとユーザー側の静的安定性

障害時にコントロールプレーンに新規作成や設定変更などを依頼しない。
障害復旧時に手作業などの特別なアクションを必要無いようにする。



リソースレベルの可用性

アーキテクチャの検討と可用性

静的安定性とは

リソースレベルの可用性

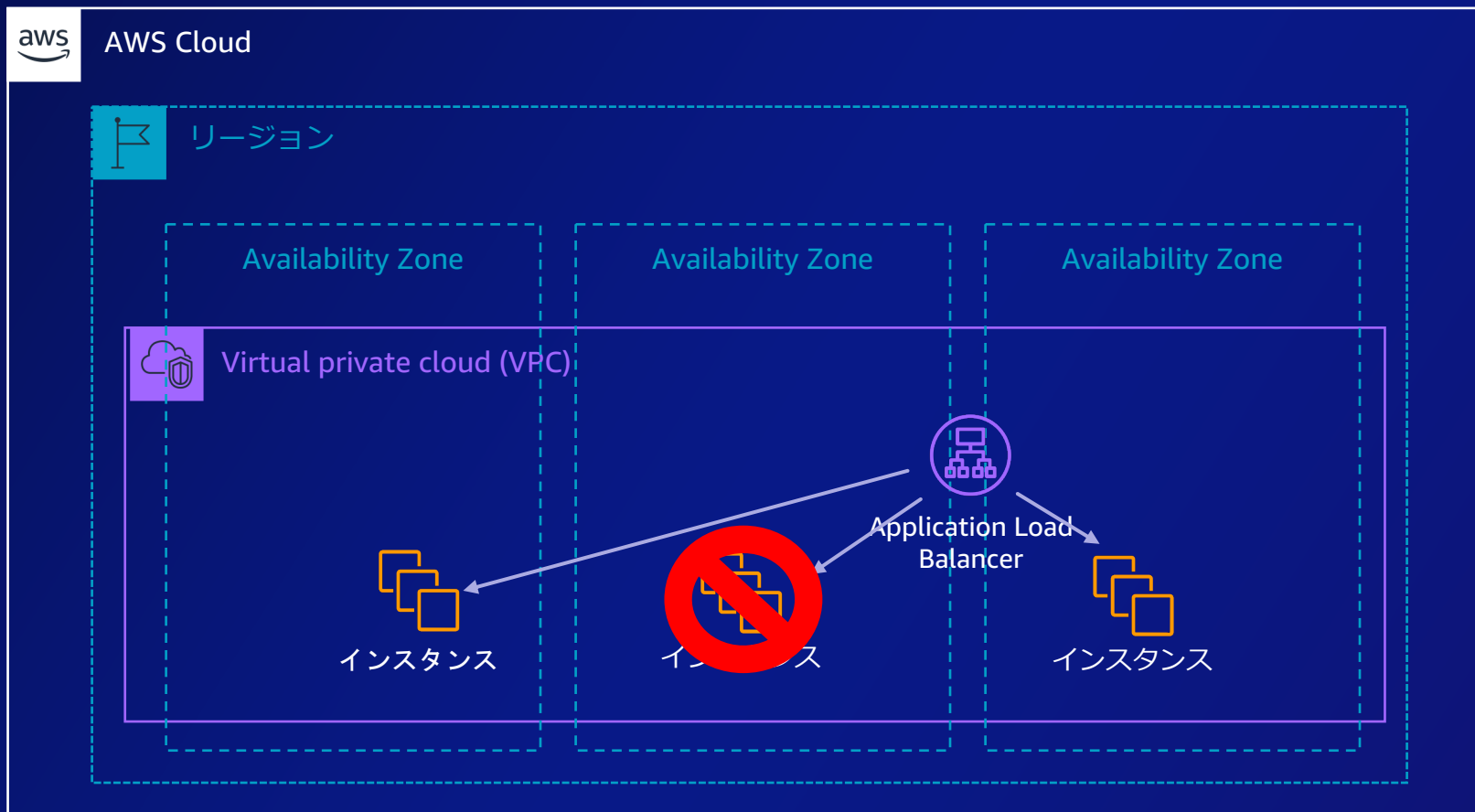
AWS サービスレベルの可用性

リージョンを越える範囲の可用性

まとめ

リソースレベル – Amazon EC2

コンピュータのH/W障害などに備えた冗長化。静的安定性の例。

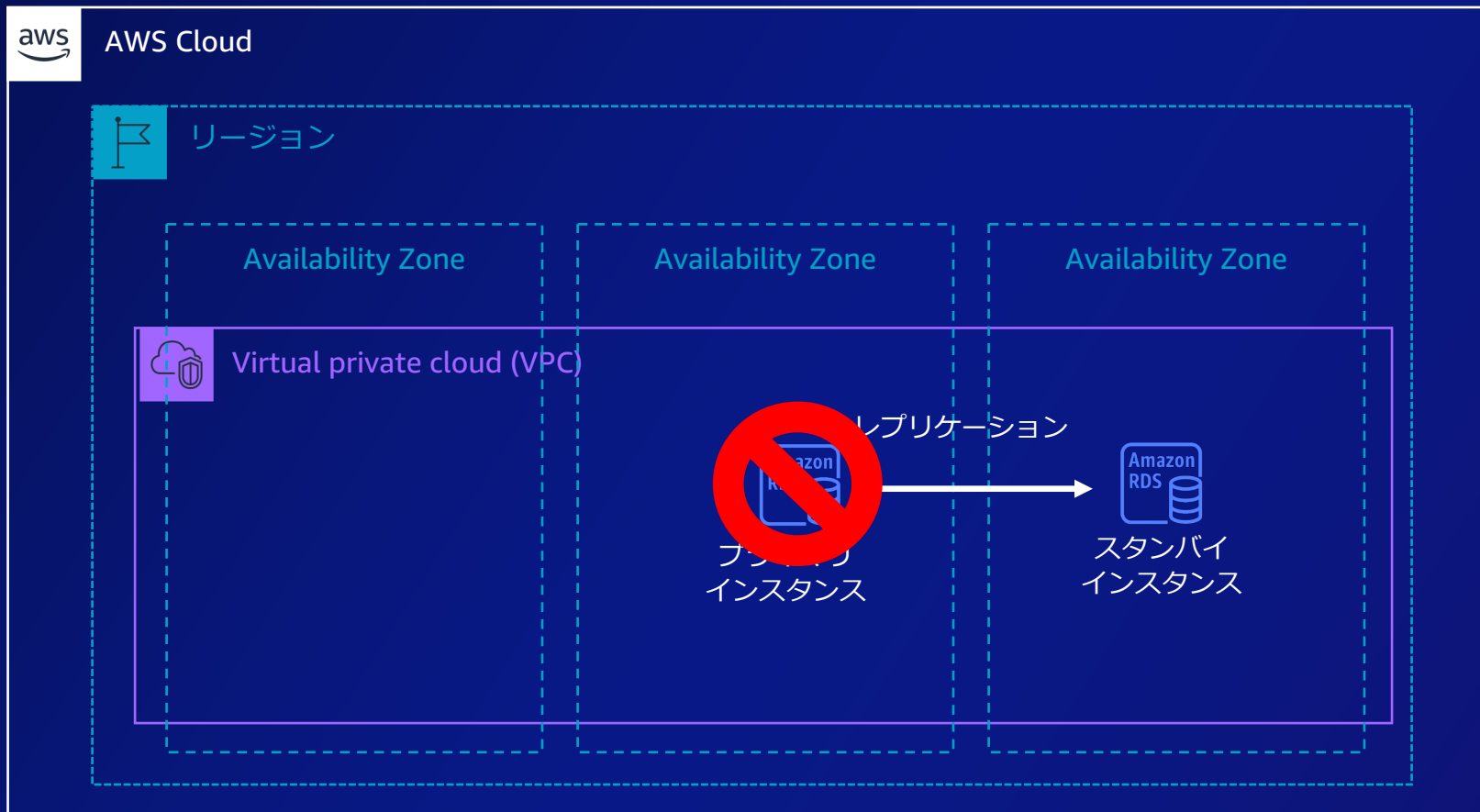


H/W障害などで
インスタンスが
落ちてもインスタンスを
追加することなく、
既存リソースで処理を
続けるようにする。
既存リソースは
データプレーンのみで
稼働することができる。

参考となる資料 : <https://aws.amazon.com/jp/builders-library/static-stability-using-availability-zones/>

リソースレベル – Amazon RDS

コンピュータのH/W障害などに備えた冗長化。静的安定性の例。



H/W障害などで
プライマリインスタンスが
落ちててもインスタンスを
追加することなく、
既存リソース(スタンバイ)
で処理を続けられる。
特にリソース追加の
アクションも不要。

参考となる資料 : <https://aws.amazon.com/jp/builders-library/static-stability-using-availability-zones/>

AWS サービスレベルの可用性

アーキテクチャの検討と可用性

静的安定性とは

リソースレベルの可用性

AWS サービスレベルの可用性

リージョンを越える範囲の可用性

まとめ

AWS サービス単体の可用性

可用性の高い設計、AWS サービス内の静的安定性



AWS Well-Architected Framework信頼性の柱 付録 A 一部のAWSのサービスの可用性設計

サービス	コンポーネント	可用性の設計目標
Amazon API Gateway	コントロールプレーン	99.950%
	データプレーン	99.990%
Amazon Aurora	コントロールプレーン	99.950%
	シングルAZ データプレーン	99.950%
	マルチAZ データプレーン	99.990%
Amazon CloudFront	コントロールプレーン	99.900%
	データプレーン (コンテンツ配信)	99.990%
	⋮	
AWS X-Ray	コントロールプレーン (コンソール)	99.990%
	データプレーン	99.950%
Elastic Load Balancing	コントロールプレーン	99.950%
	データプレーン	99.990%

目標となる可用性と冗長化

AWS Well-Architected Framework 信頼性の柱

▼ 可用性目標の実装例

▼ 単一リージョンのシナリオ

99% のシナリオ

99.9% のシナリオ

99.99% のシナリオ

▼ 複数リージョンのシナリオ

可用性がスリーアンドハー
フナイン (99.95%) で、復
旧時間が 5~30 分

ファイブナイン (99.999%)
以上のシナリオで、復旧時
間が 1 分未満

実装例の項目

- リソースをモニタリングする
- 需要の変化に対する適応方法
- 変更の実装
- データのバックアップ方法
- 弾力性(resilience)のためのアーキテクト
- 回復力をテストする方法
- 災害対策(DR)を計画する
- 可用性の設計目標

可用性のために重要なトピックが記載されています。ぜひ見てください。

マルチAZ

▼ 可用性目標の実装例

▼ 単一リージョンのシナリオ

99% のシナリオ

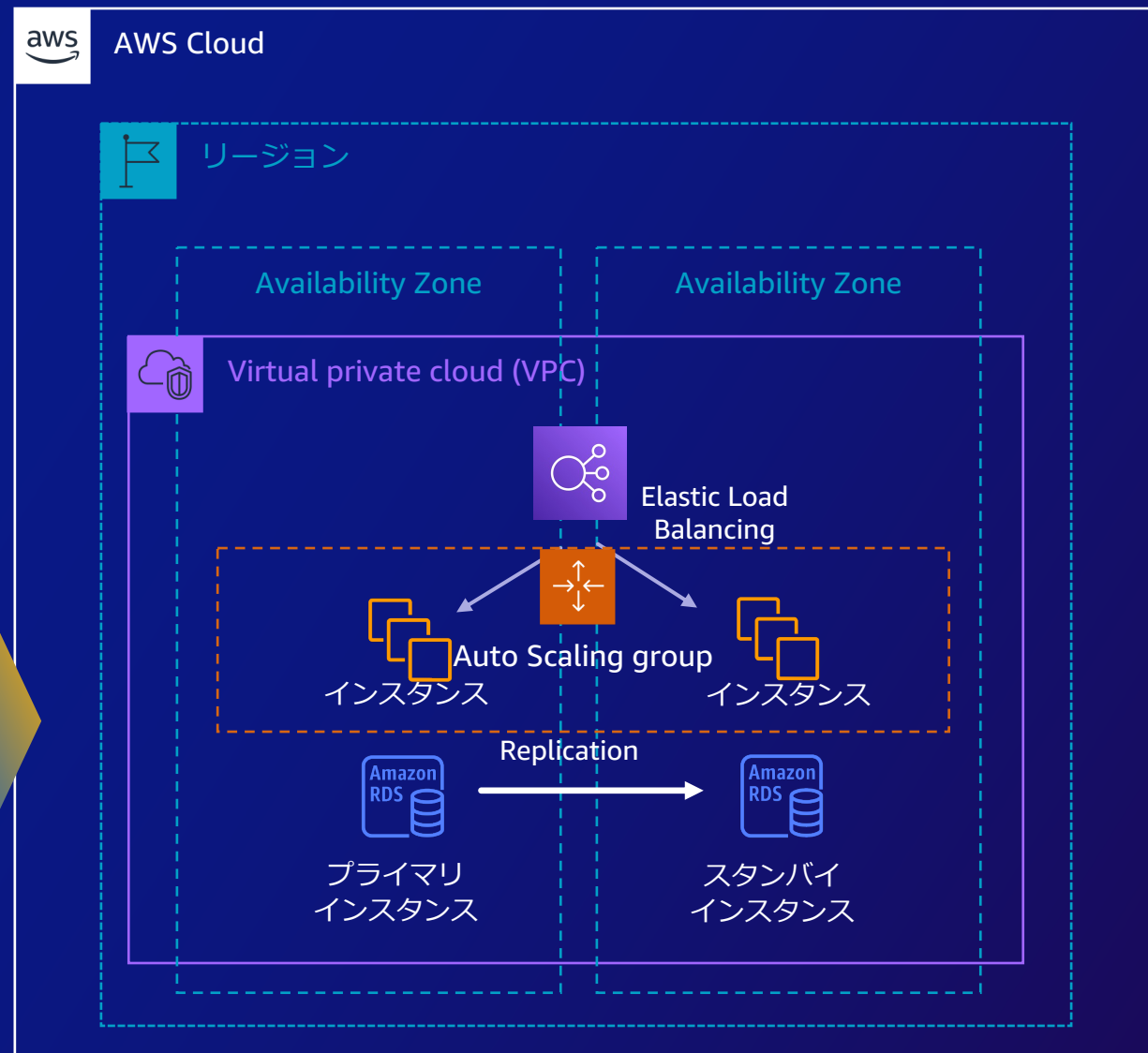
99.9% のシナリオ

99.99% のシナリオ

▶ 複数リージョンのシナリオ

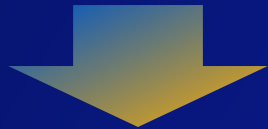
ポイント

- AZを2つ
- Elastic Load Balancing
- Amazon RDS マルチAZ

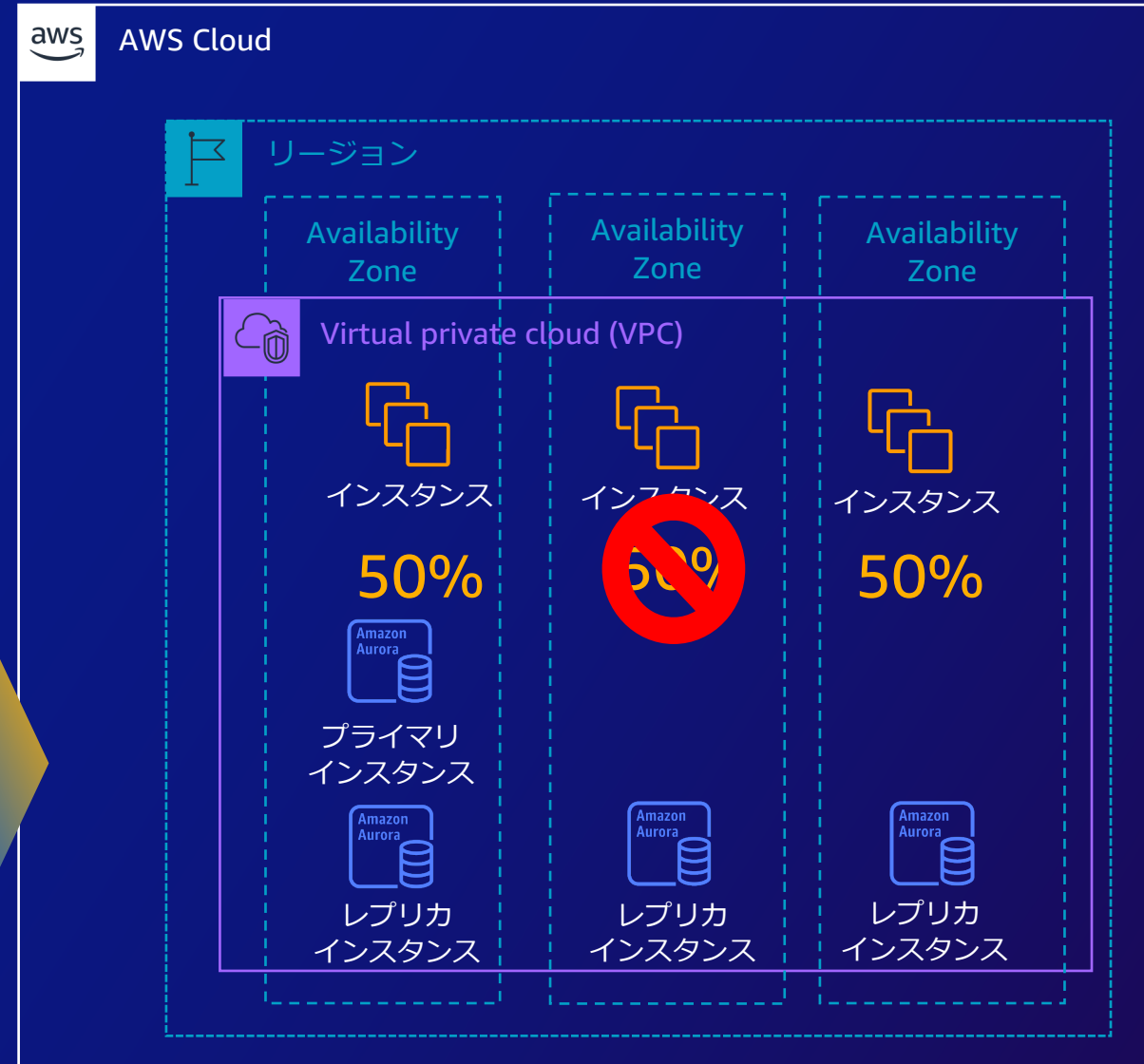


99.99%の可用性目標 – ミッションクリティカルレベル

▼ 可用性目標の実装例
▼ 単一リージョンのシナリオ
99% のシナリオ
99.9% のシナリオ
99.99% のシナリオ



3AZの構成で各AZはピーク性能の50%のキャパシティを持つようにする。障害に耐えるためのコントロールプレーンへの変更を必要としない。



複数リージョン

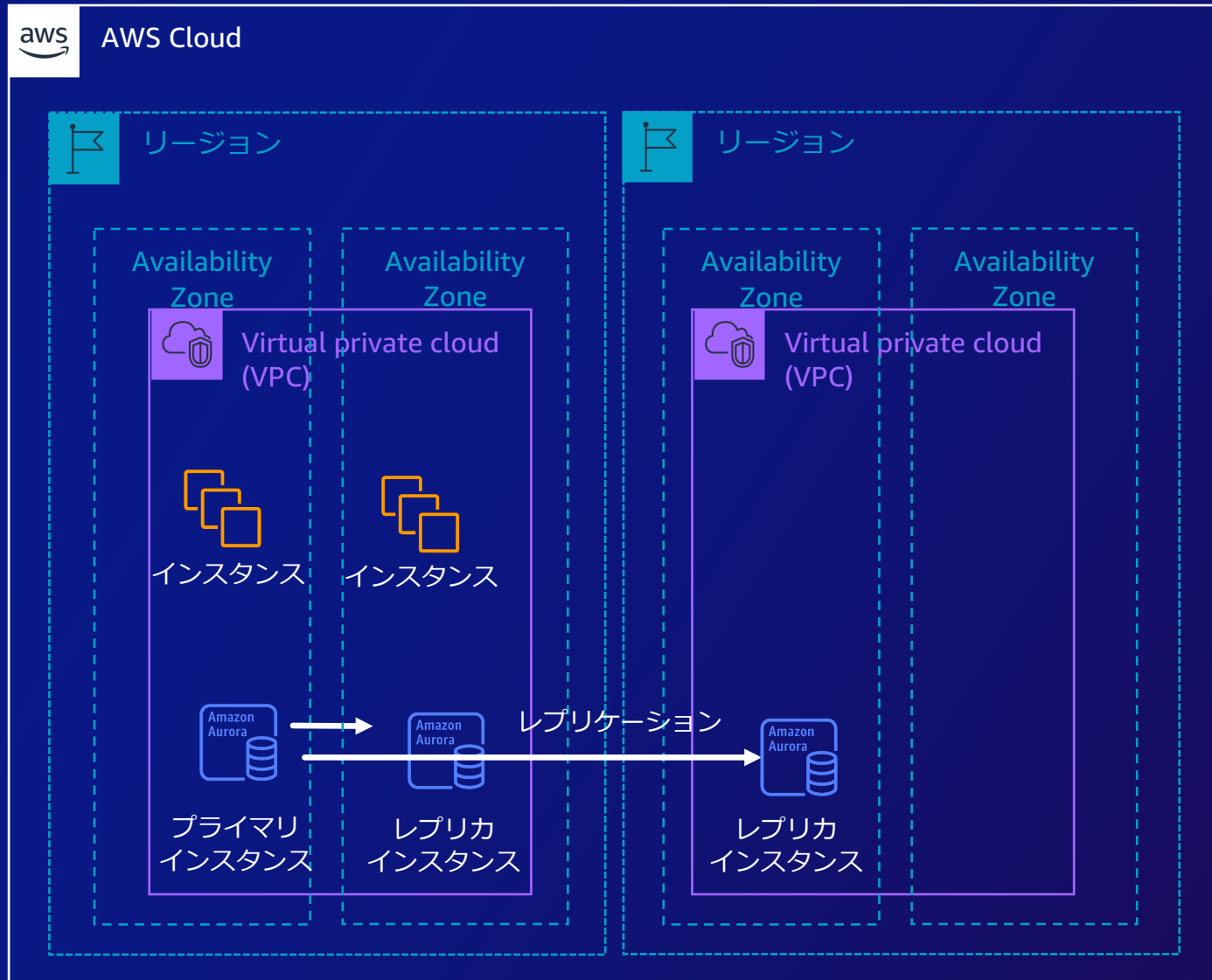
▼ 可用性目標の実装例

▶ 単一リージョンのシナリオ

▼ 複数リージョンのシナリオ

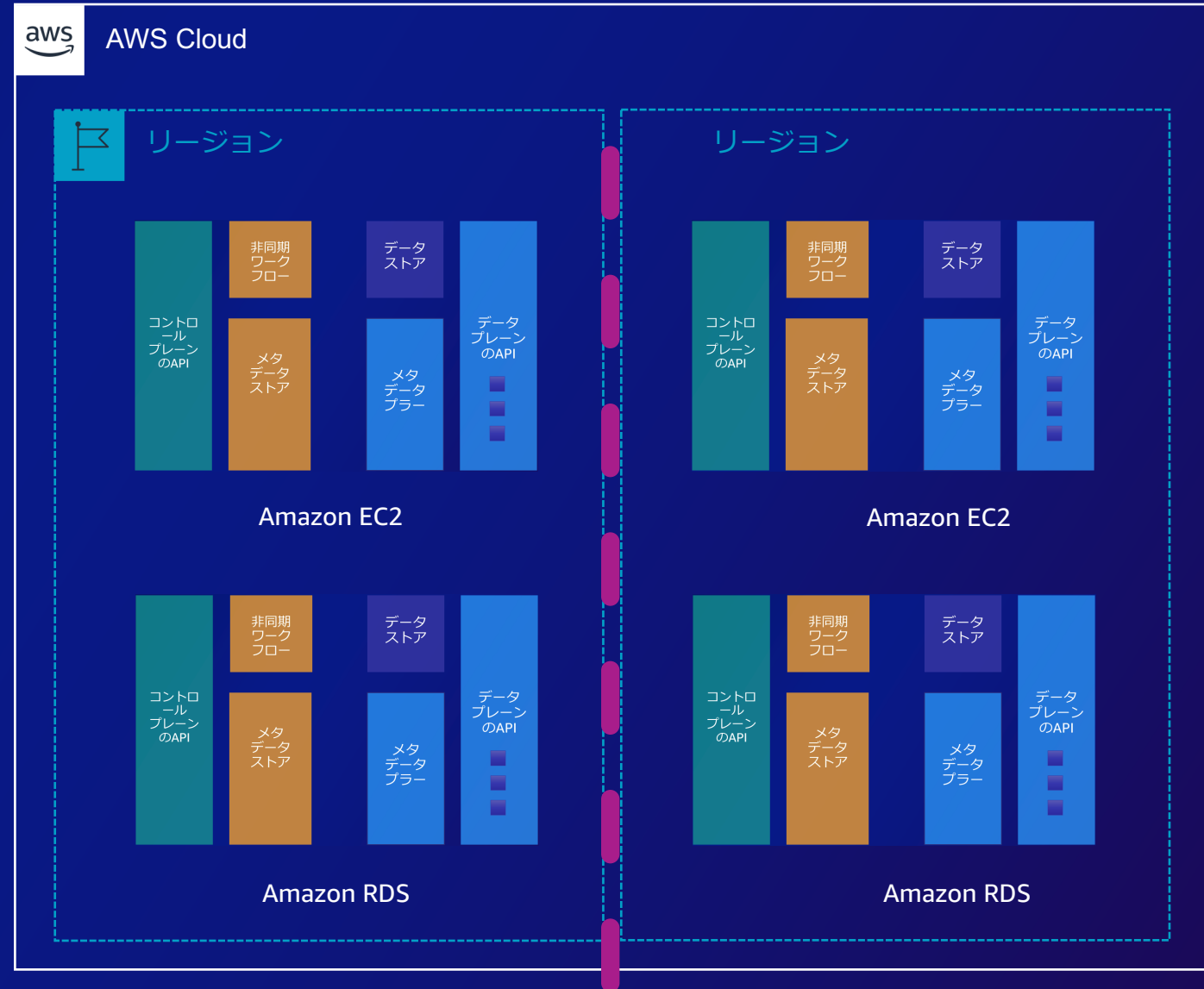
可用性がスリーアンドハーフナイン (99.95%) で、復旧時間が 5~30 分

ファイブナイン (99.999%) 以上のシナリオで、復旧時間が 1 分未満



複数リージョン

リージョン同士は独立しているため、複数リージョン構成でデータプレーンの障害に耐えるといった高い可用性を得られるのがわかんと思います。




ここまでのまとめ

1 AWS 内の静的安定性で AWS サービスは可用性が高い。

2 冗長化で可用性を高くできる。

関係がある。
意識すべき観点。



3 ユーザー側の静的安定性で可用性を高くできる。

活かしている。

リージョンを越える範囲の可用性

アーキテクチャの検討と可用性

静的安定性とは

リソースレベルの可用性

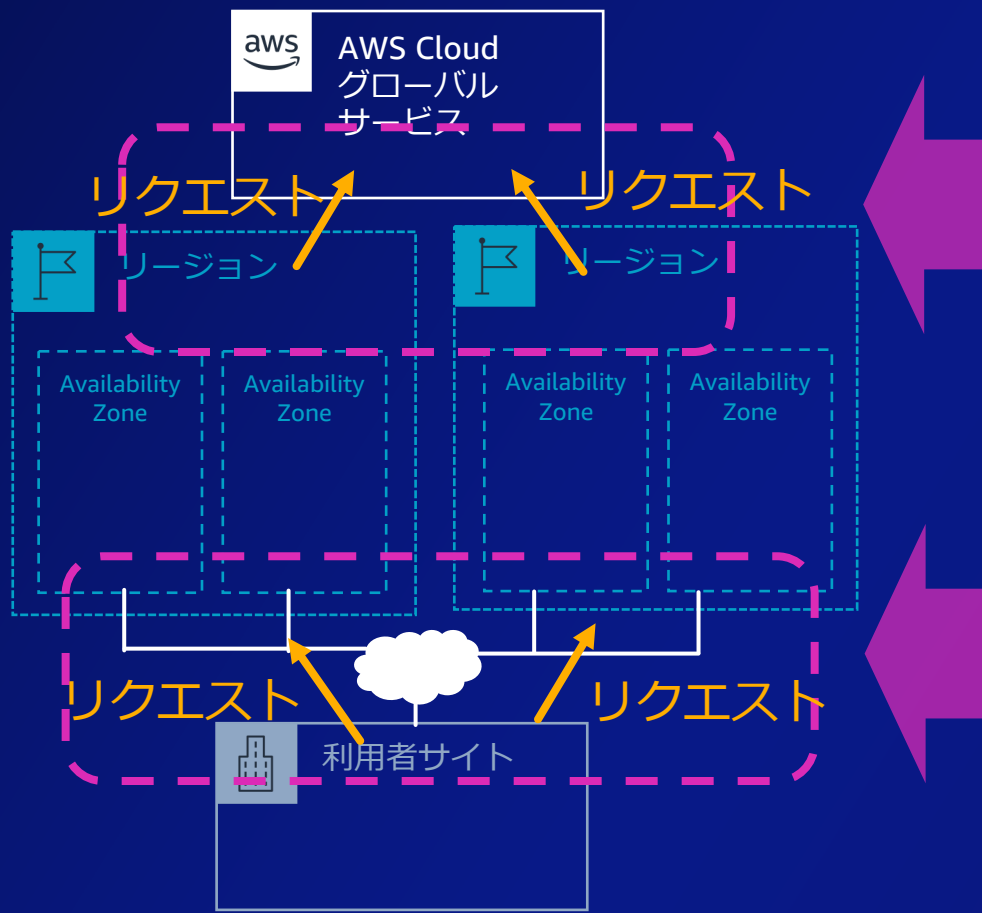
AWS サービスレベルの可用性

リージョンを越える範囲の可用性

まとめ

リージョンを越えて考える

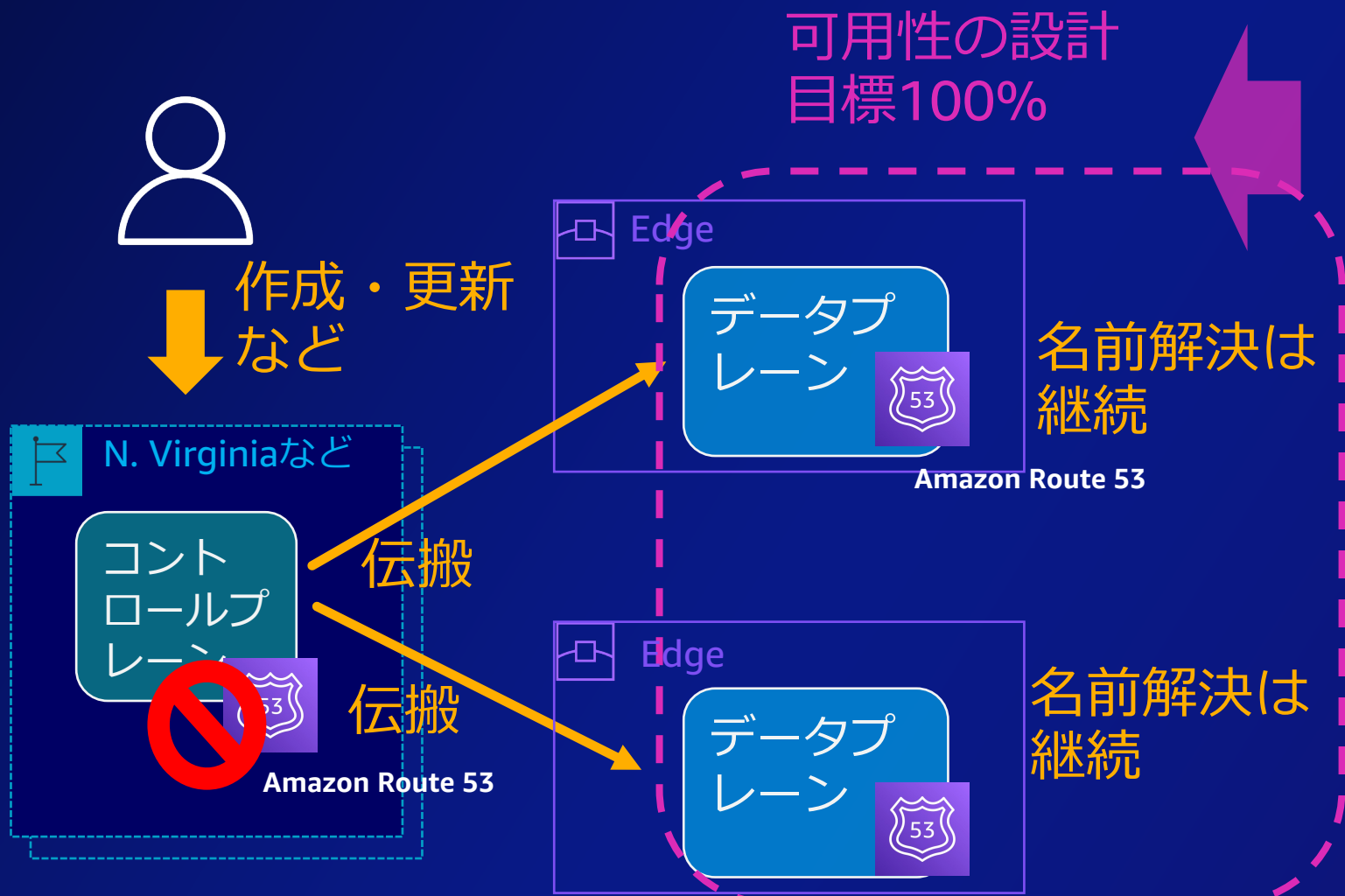
グローバルサービス、利用者とのネットワーク(インターネット、閉域)



グローバルサービスの
リクエスト(利用)がある。
マルチリージョン構成では
なくても必要なサービス
(例 : Amazon Route 53)

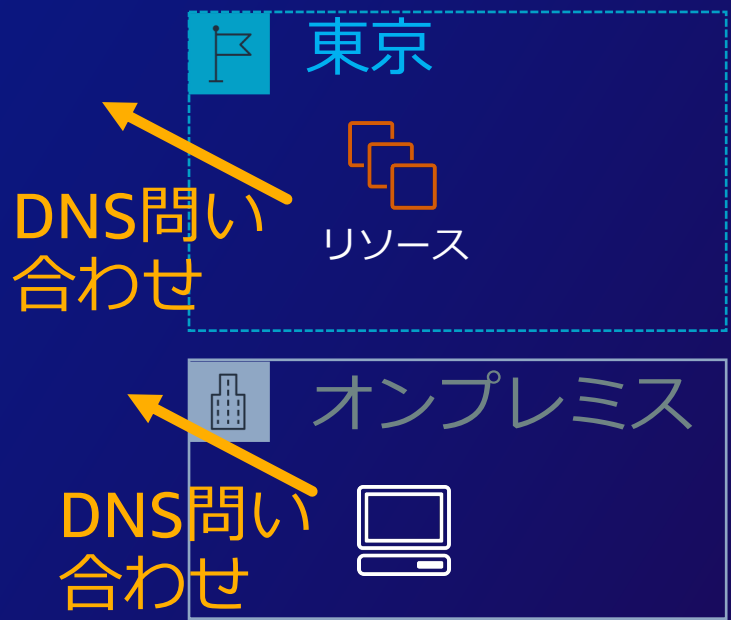
クラウドに関わらず、
昨今のシステムでは
ネットワークは必要な
インフラ

グローバルサービス



Amazon Route 53の設計目標100%を活かすためには、ユーザー側の静的安定性が必要です。シングルリージョンでも必要なケースは多いです。

お勧めのブログ：[Amazon Route 53 を用いたディザスタリカバリ \(DR\) のメカニズム](#)



Public DNSを説明しています。VPC DNSサービスはAWS Fault Isolation Boundariesを参照のこと



静的安定性を支援する AWS サービス

Amazon Route53 Application Recovery Controllerを使う方法もあります。

名前解決



概要：

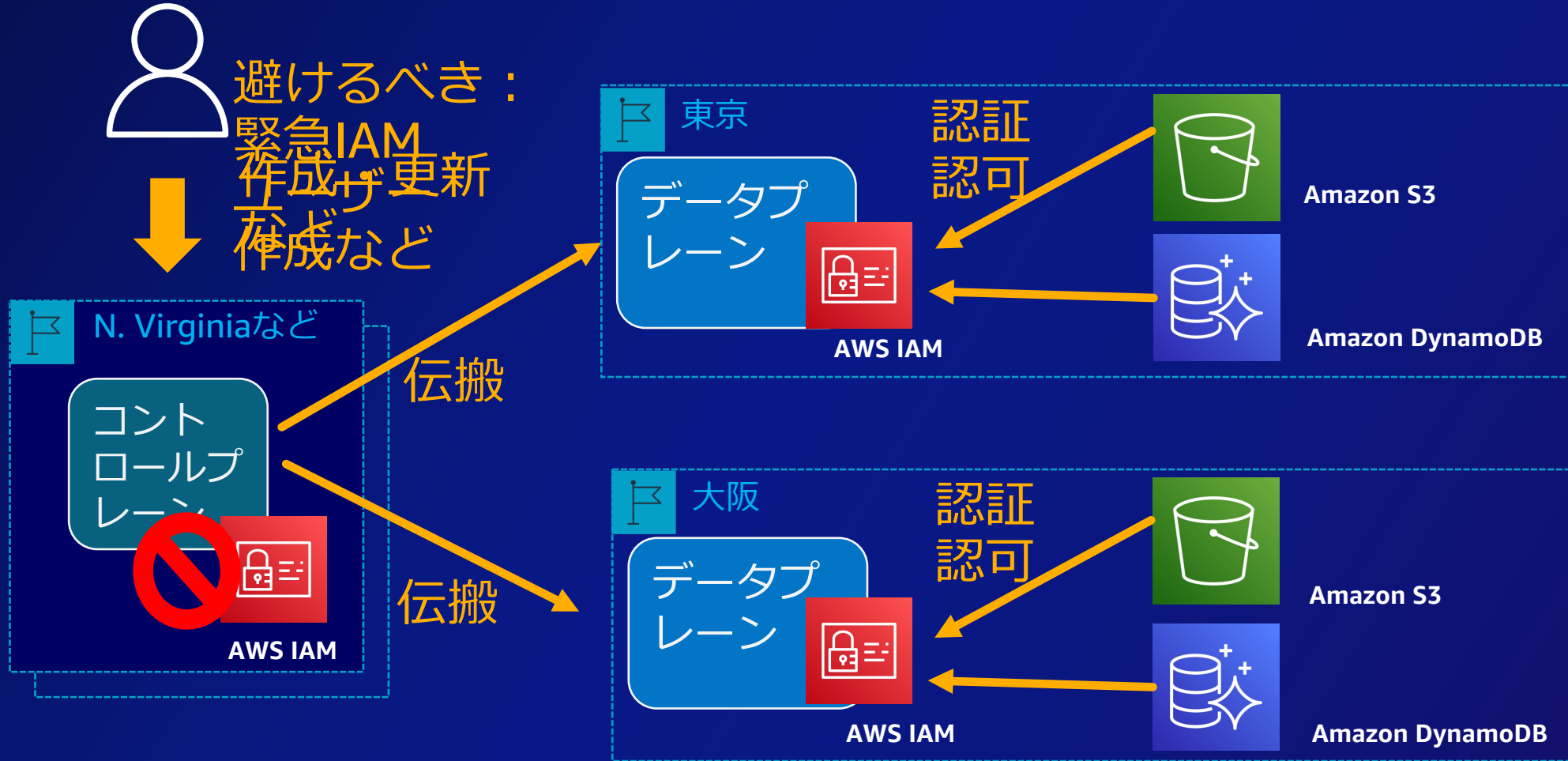
- 複数リージョンや複数AZで構成されるアプリケーションに対して、横断でのリカバリーを支援するサービス
- 大部分の障害に対して数分でのリカバリーを可能にするよう設計・実装されたアプリケーションの構築と管理を支援

特徴：

- コントロールプレーンに依存せず動作する

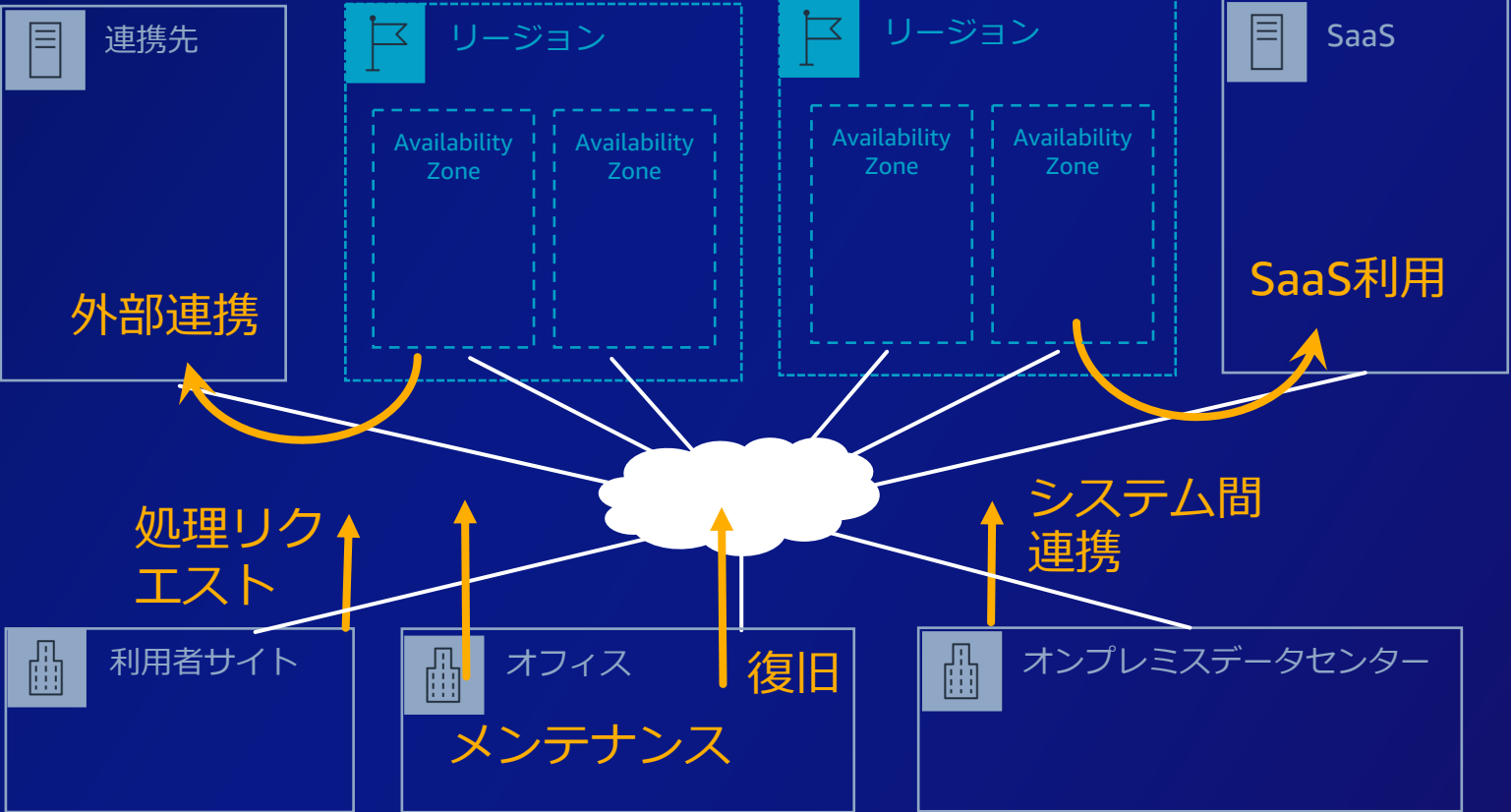
グローバルサービス

AWS IAMに関するユーザー側の静的安定性も大事。作成や更新を避ける。



STSの注意点は[AWS Fault Isolation Boundaries](#)を参照のこと。他のグローバルサービスについての説明も [AWS Fault Isolation Boundaries](#) を参照のこと。

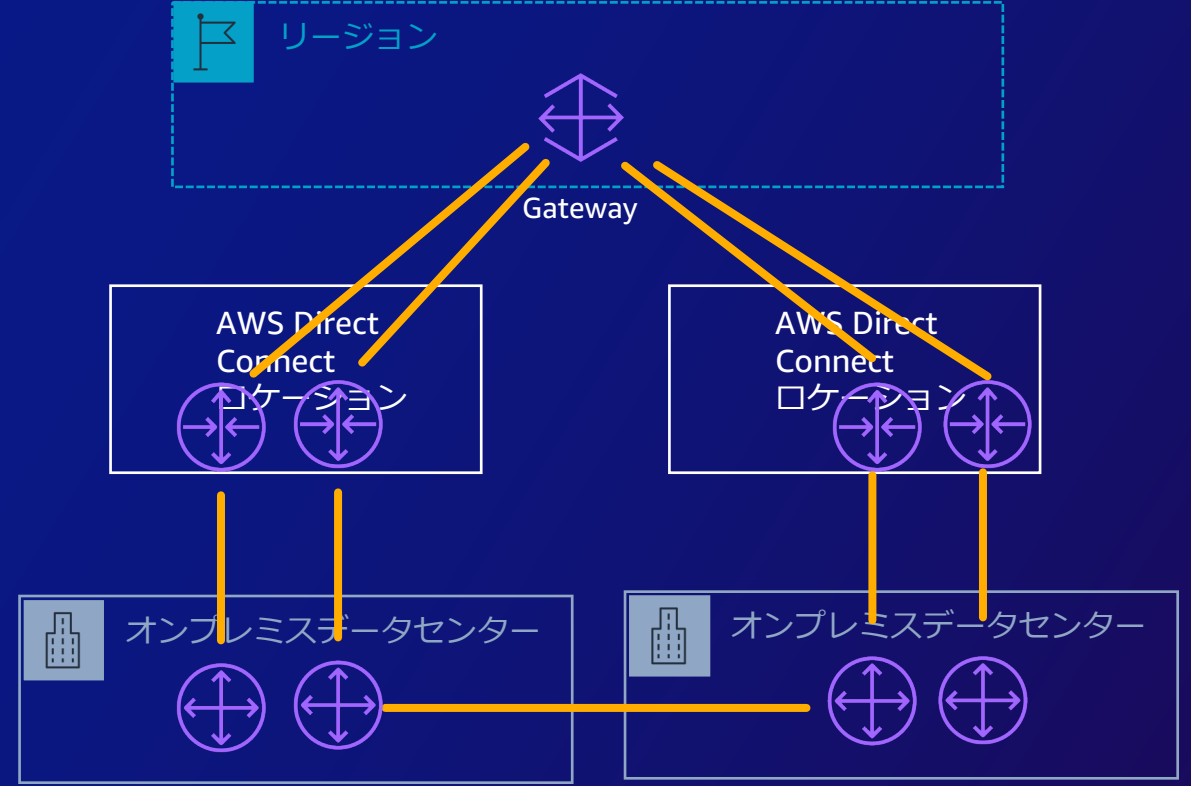
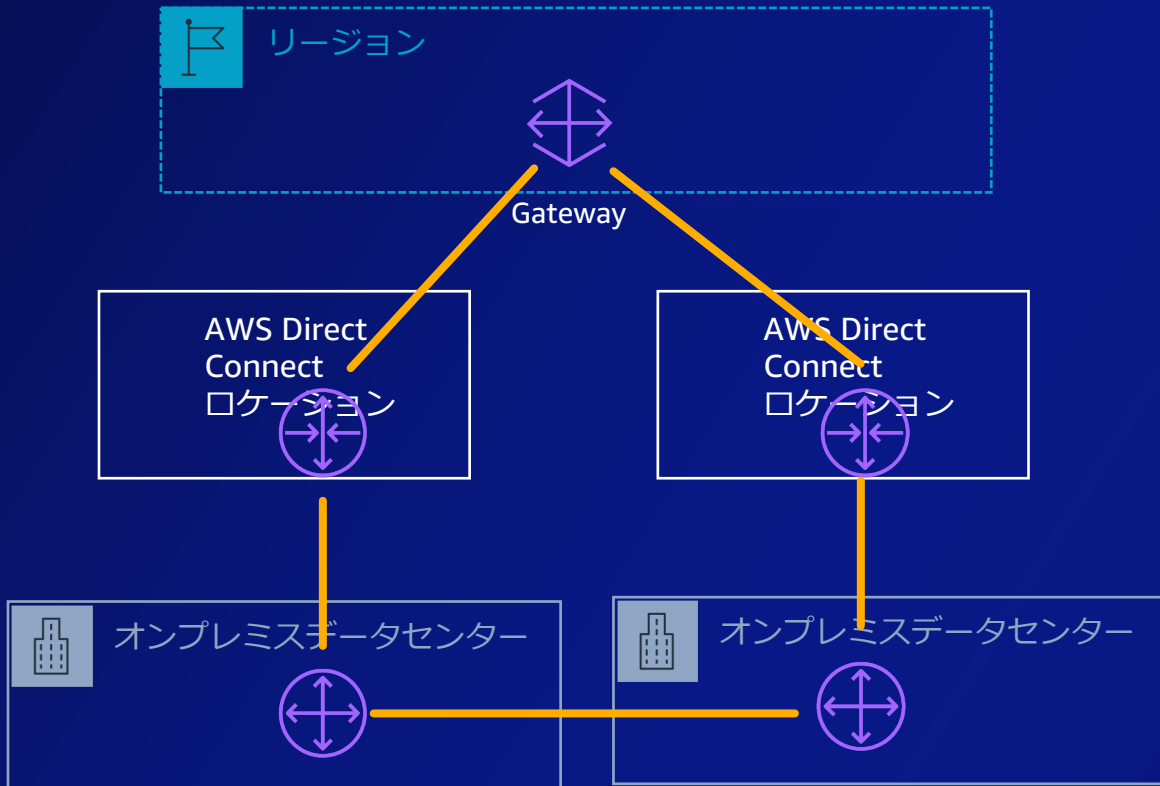
ネットワークの可用性と冗長性は大事



利用者とのネットワーク

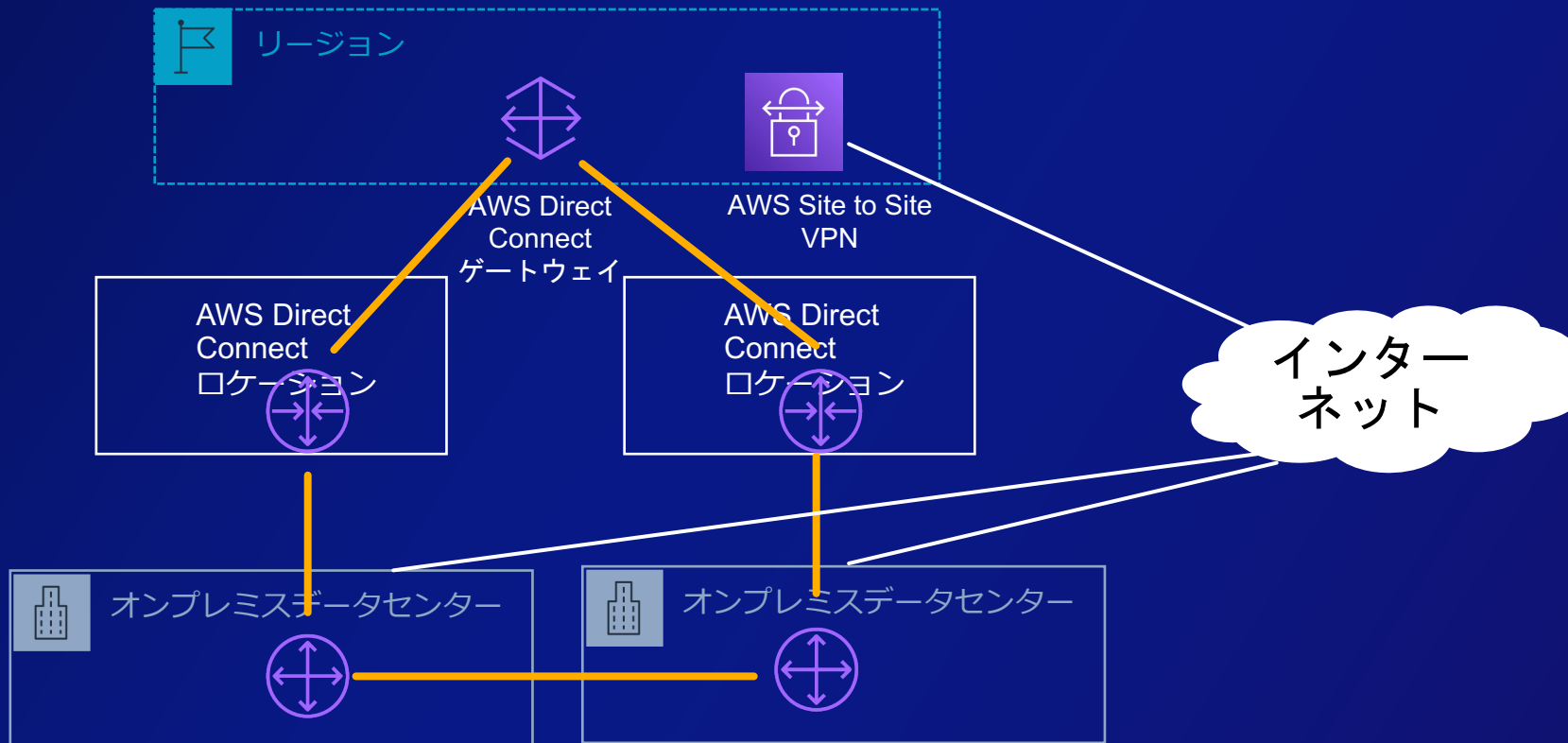
クリティカルな
ワークロードの高い回復性

クリティカルな
ワークロードの最大回復性



利用者とのネットワーク

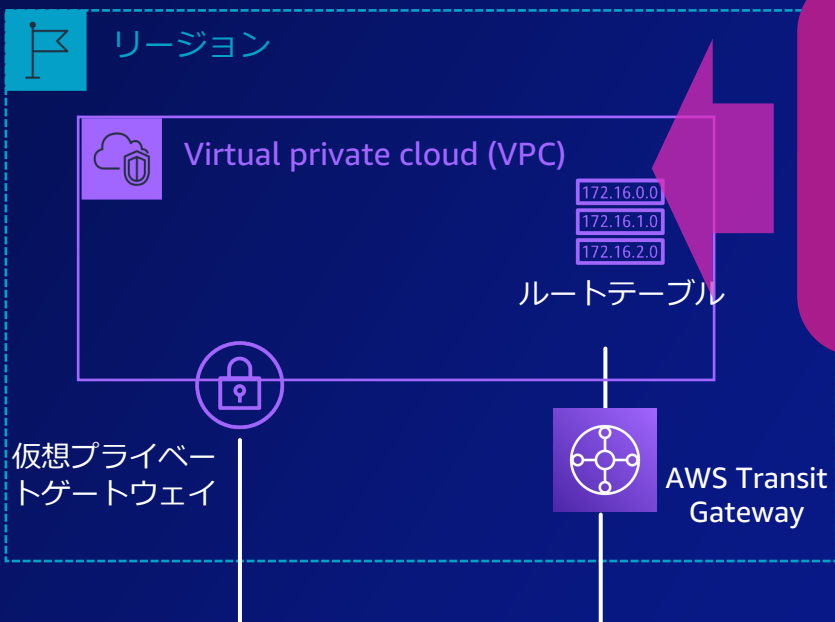
AWS Site to Site VPNによるAWS Direct Connectの迂回経路の用意も有用です。



参考資料 : <https://aws.amazon.com/jp/directconnect/resiliency-recommendation/>

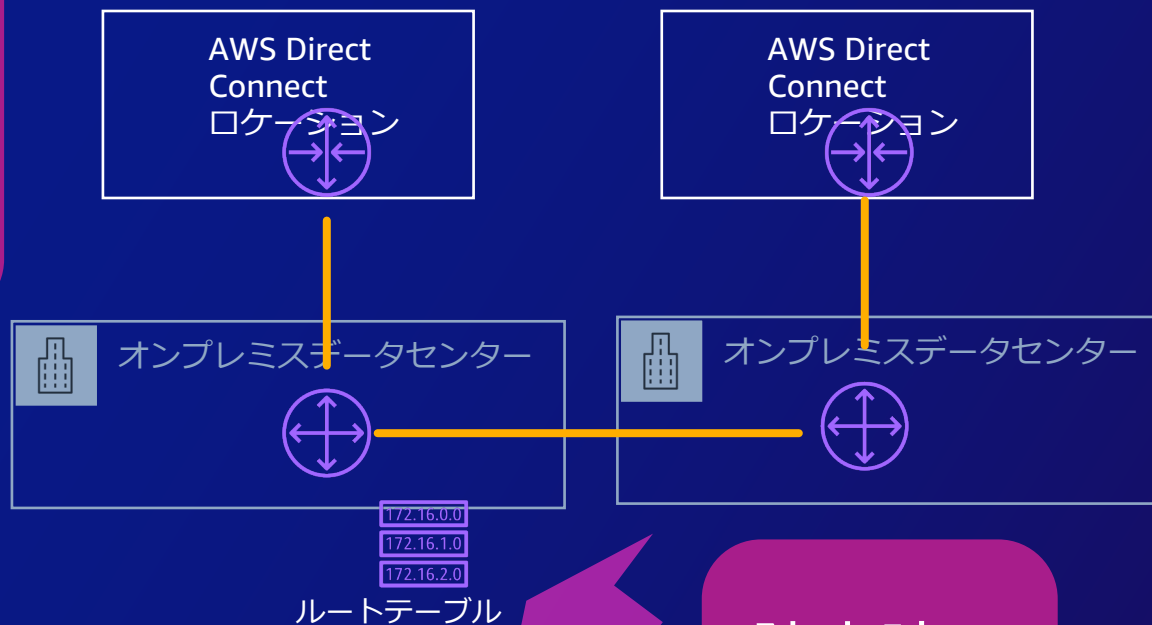
ネットワークのユーザー側の静的安定性

静的安定性が低い例



障害時に
手動で
書き換え

静的安定性が低い例



障害時に
手動で
書き換え

まとめ

アーキテクチャの検討と可用性

静的安定性とは

リソースレベルの可用性

AWS サービスレベルの可用性

リージョンを越える範囲の可用性

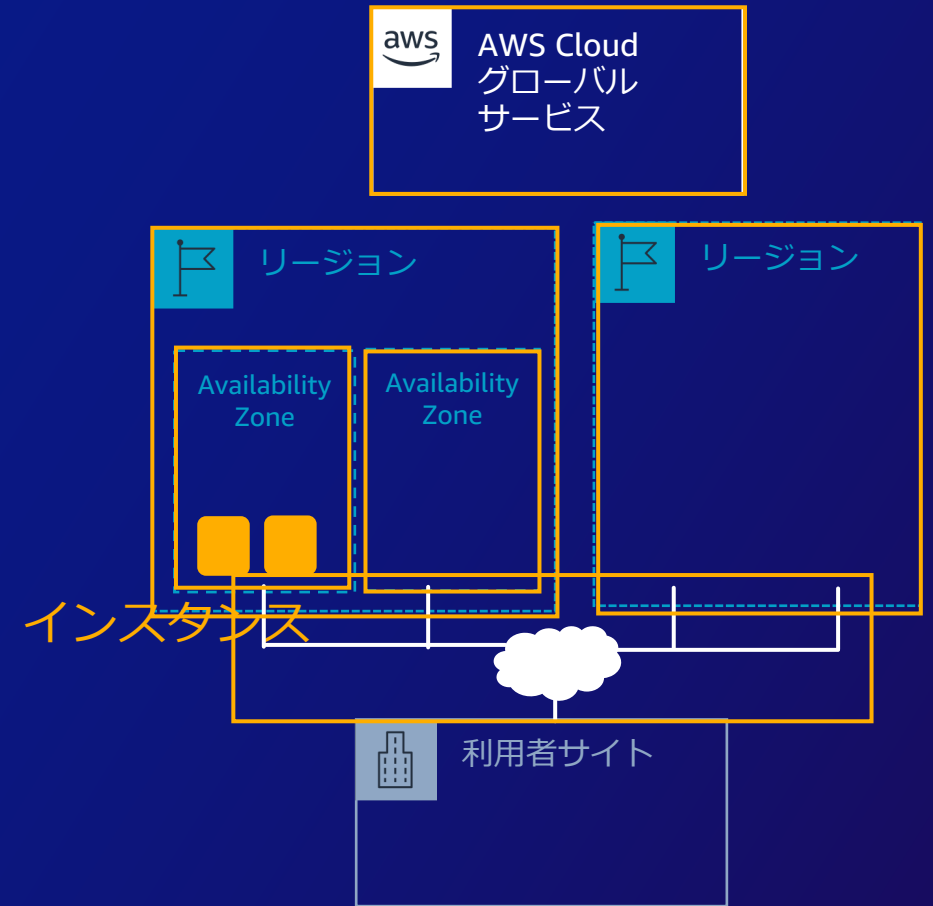
まとめ

ポイント3つ

静的安定性という AWS 内で培われた高可用性の特性があり、AWS サービスの可用性を支えているだけでなく、ユーザー側でも活かせる。

AWS 内の静的安定性、マルチAZなどの冗長化、ユーザー側の静的安定性の組み合わせでかなりの障害ケースに対応できる。

静的安定性はどのようなシステムにとっても重要な特性である。



静的安定性は至るところに存在する。
至るところで活用できる。

さらなる理解のために

参考になる資料

AWS Fault Isolation Boundaries(英語)

https://docs.aws.amazon.com/ja_jp/whitepapers/latest/aws-fault-isolation-boundaries/abstract-and-introduction.html

アベイラビリティゾーンを使用した静的安定性(日本語)

<https://aws.amazon.com/jp/builders-library/static-stability-using-availability-zones/>

信頼性の柱 - AWS Well-Architected フレームワーク

https://docs.aws.amazon.com/ja_jp/wellarchitected/latest/reliability-pillar/welcome.html

マルチリージョンでディザスタリカバリ(DR) 戦略を検討するためのポイント

https://pages.awscloud.com/rs/112-TZM-766/images/AWS-50_Key_points_to_consider_for_a_multi-region_Disaster_Recovery_DR_strategy_KMD52.pdf

さらなる理解のために

参考になる資料

AWS Direct Connect の回復性に関する推奨事項

<https://aws.amazon.com/jp/directconnect/resiliency-recommendation/>

AWS Black Belt Online Seminar AWS Direct Connect

<https://pages.awscloud.com/rs/112-TZM-766/images/20210209-AWS-Blackbelt-DirectConnect.pdf>

AWS Lambda: Resilience under-the-hood

<https://aws.amazon.com/jp/blogs/compute/aws-lambda-resilience-under-the-hood/>

Amazon Route 53 を用いたディザスタリカバリ (DR) のメカニズム

<https://aws.amazon.com/jp/blogs/news/creating-disaster-recovery-mechanisms-using-amazon-route-53/>

Thank you!

小田 圭二

アマゾン ウェブ サービス ジャパン合同会社

プロフェッショナルサービス本部 クラウドインフラストラクチャアーキテクト

