aws training and certification

グループディスカッション 1:

既存アプリケーションの "フォークリフト"

投影のみ

目的

- **これまで扱った AWS コアサービスの知識を深めること**
  - 「見る＆聞く」よりも「行う＆話す」のほうが知識の定着がよい

- **Well-Architected Framework の設計方針を体験する**
  - クラウドの特性を生かしたアーキテクチャについて意見し合う
  - 5 つの柱：セキュリティ / 信頼性 / コスト / 性能 / 運用

Mod1 : リージョン / AZ
Mod2 : S3
Mod3 : EC2 / EBS / EFS
Mod4 : RDS / DynamoDB / DMS
Mod5 : VPC
Mod6 : DX / VPC Peering / VPC Endpoint
Mod7 : IAM / Cognito / Organizations
Mod8 : Auto Scaling / CloudWatch

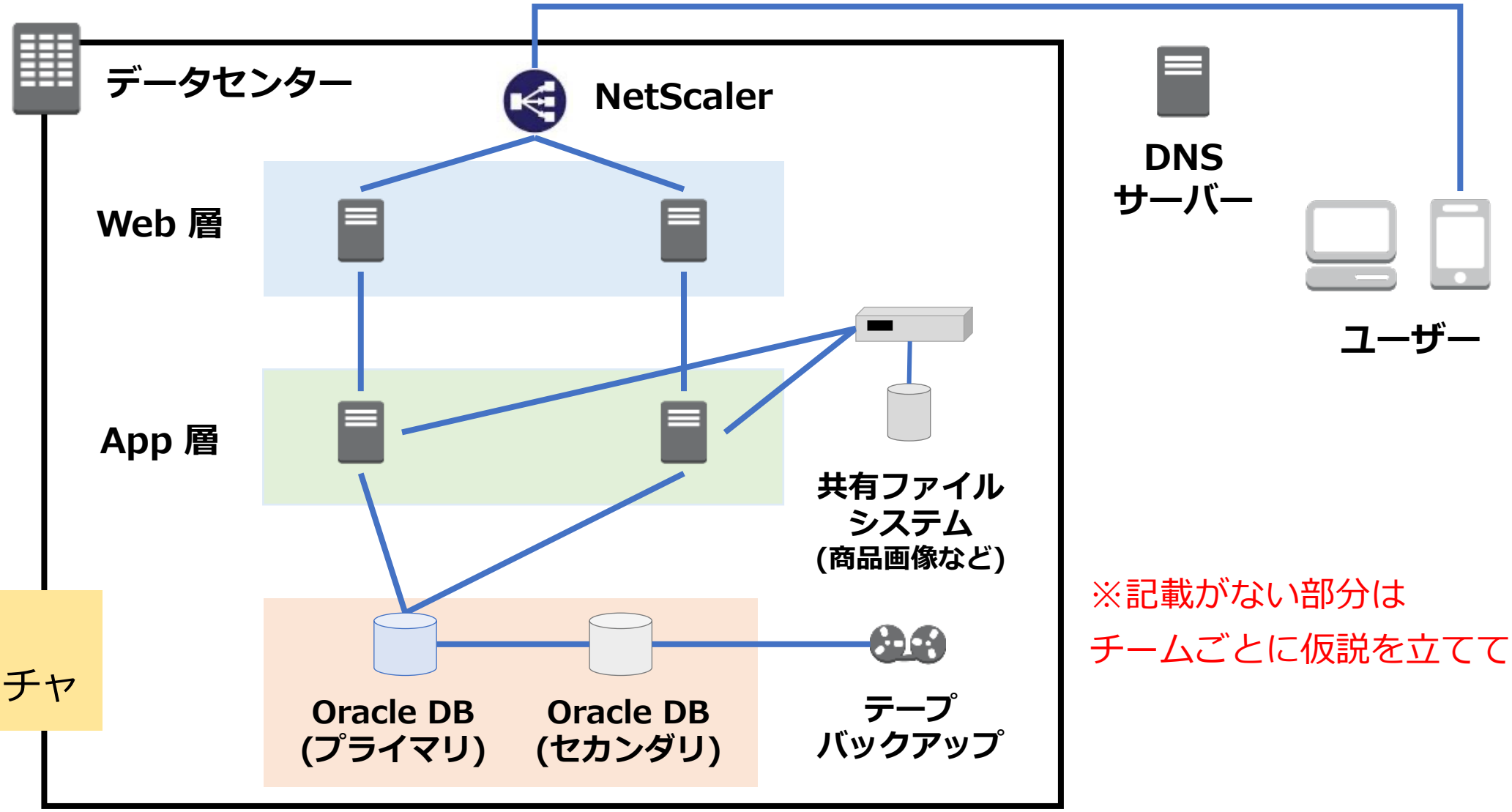
GoGreen

- GoGreen 社は**「EC サイト」**をエンドユーザーに提供している
- ウェブアプリケーション
- **エンドユーザーが使える主要な機能**
  - 商品情報の閲覧
    - 商品名、商品説明、在庫数、商品画像など
  - 商品の購入
    - クレジットカード情報の取扱など、決済手段の詳細は割愛する
  - など

GoGreen の現在の状況

- 現在契約しているデータセンターが**「来年に期限切れ」**になる
- GoGreen は**「コンポーネントの一部または全て」**を AWS に移行することを決定している
- 移行期間は**「6-12ヶ月」**と目安にする

オンプレミスのデータセンター



3層
アーキテクチャ

※記載がない部分は
チームごとに仮説を立てて OK

移行要件

- **リージョン要件**
  - **「任意のリージョン」**を1個選択する
  - エンドユーザーは日本に集中している（主に商品の購入）
- **ウェブアプリケーション要件**
  - 可用性を向上する
  - データの耐久性を強化する
  - 障害時の復旧時間を短縮するアーキテクチャを目指す

検討ポイント（サンプル）

- **運用の優秀性**
  - ワークロードの正常性をどのように把握するか？
- **セキュリティ**
  - どのようにデータをセキュアに保管するか？
- **信頼性**
  - 需要の変化にどのように対応するか？
  - ネットワークをどのように設計するか？
- **パフォーマンス効率**
  - EC2 インスタンスタイプをどのように選ぶか？
- **コスト最適化**
  - どのようにコストを最適化しながらスケールできるか？

AWS Well-Architected

を参考に！

投影のみ

グループディスカッション

- タイムスケジュール
  - グループディスカッション：30分
  - 発表 + フィードバック
    - 構成図 + 検討内容をチャットに添付
    - 他グループの成果にコメント
- Google Slide使って、アーキテクチャなどを描く
  - 全員で考えながら描いていきましょう！
  - 検討内容「どの観点、仮説でこの構成にしたか」をメモ

＼ Enjoy ! ／

投影のみ

参考：Design Principles (設計原則) 11個

aws training and certification

- **Scalability** (スケーラビリティを実現する)
- **Disposable Resources Instead of Fixed Servers** (使い捨て可能なリソースを使用する)
- **Automation** (環境を自動化する)
- **Loose Coupling** (コンポーネントを疎結合化する)
- **Services, Not Servers** (サーバーではなくサービスを設計する)
- **Databases** (適切なデータベースソリューションを選択する)
- **Managing Increasing Volumes of Data** (増え続けるデータを管理する)
- **Removing Single Points of Failure** (単一障害点を避ける)
- **Optimize for Cost** (コストの最適化)
- **Caching** (キャッシュを使用する)
- **Security** (すべてのレイヤーでセキュリティを確保する)

aws training and certification

グループディスカッション 2:

"スケーラブルな" ウェブアプリケーション

投影のみ

GoGreen

- GoGreen 社は、驚異的に売上を伸ばしている
- 具体的な状況
  - アプリケーションに同時にアクセスする**「ユーザーの大幅な増加」**
  - **「様々な国から」**アクセスがある
  - データベース、ストレージなど**「ストレージ容量が増加している」**
  - **「システムコスト」**も増えている

フェーズ 2: 再設計要件 1

- **背景**
  - CTO から**「プラットフォームを再設計して欲しい」**と依頼された
- **リージョン要件**
  - **「任意のリージョン」**を2個選択する
  - Active-Active 環境として設計するのか、
    災害対策環境として設計するのか、チームで検討する
- **ユーザー要件**
  - エンドユーザーは日本に集中している（主に商品の購入）
  - しかし、海外から EC サイトにアクセスし、
    閲覧するエンドユーザーは多い

フェーズ 2: 再設計要件 2

- **信頼性**
  - 障害時の復旧時間を短縮するアーキテクチャを目指す
- **パフォーマンス効率**
  - **「ウェブ層 / アプリケーション層 / データベース層 (RDBMS)」**から構成される**「3層」**アーキテクチャは**変えないこと**
  - ウェブ層とアプリケーション層が以下の要件を満たすこと
    - 疎結合 / スケーラブル / ステートレス
  - セッション情報を**「耐障害性を持つ方法」**で永続的に保存する
  - エンドユーザーに最高のパフォーマンスを提供すること

フェーズ 2: 再設計要件 3

- **運用上の優秀性**
  - できる限り**「運用をコード化」**することで、
    運用を効率化し、ヒューマンエラーを減らす
- **コスト最適化**
  - 「コスト効率の良い」アーキテクチャを検討する
    - コストには AWS コストだけではなく、運用に必要なコストも考慮する
- **もし時間があれば**
  - 「プログラムのデプロイ」など、DevOps な観点も検討する
  - 「障害発生時の対応」など、運用観点も検討する

aws training and certification

Well-Architected 質問サンプル

ディスカッション終了前に
質問サンプルを活用して設計を見直してみましょう！

https://aws.amazon.com/jp/architecture/well-architected/

https://wa.aws.amazon.com/index.ja.html

Well-Architected 質問サンプル 1

- **運用上の優秀性**
  - OPS 6: ワークロードの正常性をどのように把握しますか?
  - OPS 8: ワークロードと運用イベントはどのように管理しますか?
- **セキュリティ**
  - SEC 4: セキュリティイベントをどのように検出し、調査していますか？
  - SEC 6: ネットワークをどのように保護していますか?
  - SEC 7: コンピューティングリソースをどのように保護していますか?
  - SEC 9: 保管中のデータをどのように保護していますか?
- **信頼性**
  - REL 3: システムが需要の変化にどのように対応していますか？
  - REL 4: リソースをどのようにモニタリングしていますか？
  - REL 6: データをどうバックアップするか?
  - REL 9: 災害対策をどのように計画していますか?

投影のみ

Well-Architected 質問サンプル 2

aws training and certification

- **パフォーマンス効率**
  - PERF 2: コンピューティングソリューションをどのように選択していますか?
  - PERF 4: データベースソリューションをどのように選択していますか?
  - PERF 6: ワークロードを進化させるためにどのように新機能を取り込んでいますか？
- **コスト最適化**
  - COST 2: 使用状況とコストをどのようにモニタリングしますか?
  - COST 5: リソースタイプとサイズを選択する際、どうすればコスト目標を達成できるでしょうか?
  - COST 6: コストを削減するには、料金モデルをどのように使用したらよいでしょうか?

投影のみ

参考：Design Principles (設計原則) 11個

aws training and certification

- **Scalability** (スケーラビリティを実現する)
- **Disposable Resources Instead of Fixed Servers** (使い捨て可能なリソースを使用する)
- **Automation** (環境を自動化する)
- **Loose Coupling** (コンポーネントを疎結合化する)
- **Services, Not Servers** (サーバーではなくサービスを設計する)
- **Databases** (適切なデータベースソリューションを選択する)
- **Managing Increasing Volumes of Data** (増え続けるデータを管理する)
- **Removing Single Points of Failure** (単一障害点を避ける)
- **Optimize for Cost** (コストの最適化)
- **Caching** (キャッシュを使用する)
- **Security** (すべてのレイヤーでセキュリティを確保する)

最終ページ

aws training and certification