AWSクラウド演習

AWSクラウド演習授業資料



AWS WELL-ARCHITECTED FRAMEWORK

- AWS Well-Architected Frameworkとは

クラウドのベストプラクティスに適切に従っているか確認するためのもの(ベストプラクティスをまとめたホワイトペーパー)。質問・回答のチェックリストを提供しており、AWSのユーザーが設計・構築・運用の各フェーズでクラウドのノウハウを活用できる。

構成

「運用上の優秀性」、「セキュリティ」、「信頼性」、「パフォーマンス効率」、「コスト最適化」、 「接続可能性」の項目で構成されている。

AWS WELL-ARCHITECTED FRAMEWORKの5つの柱

- AWS Well-Architected Framework の6つの柱
 「運用上の優秀性」、「セキュリティ」、「信頼性」、「パフォーマンス効率」、「コスト最適化」
 「接続可能性」の6つの項目があげられる。
- 運用上の優秀性
 - ビジネスにおいての価値を提供するために運用面でどのようなこと点に考慮すべきか。考慮すべき項目 としては、リスク評価、軽減戦略、変更の管理と自動化、イベントへの対応、変換への対応が該当する。
 - *該当するサービス Cloud Formation、Codeシリーズ、Runbook Playbook、System Manager、Service Catalog、CloudTrail、 AWS Artifact、AWS GuardDuty、Cloud Watch、AWS Config、API Gatewayなど。

セキュリティ

システムとデータを**どのように保護するか**(システムの**保護とモニタリング**)について。データ保護、権限管理、インフラ保護、検出制御の項目が該当する。またインシデントへの対応。

設計時にはすべてのレイヤーにセキュリティを適用し、追跡可能性を実現する。また<u>最小権限の原則</u>を 適用。またシステムの保護に焦点を当て、<u>セキュリティのベストプラクティスを自動化</u>する。責任共有 モデルに基づく対象範囲の保護に集中する。

*該当するサービス

データ保護(ELB、EBS、S3、RDS、KMS)、権限管理(IAM、MFA)、インフラ保護(VPC)、

検出制御(Cloud Trail、Cloud Watch、AWS Guard Duty、Amazon Inspector)

■ 信頼性

障害からの復旧と中断による影響の軽減について。障害による中断や停止と障害復旧による影響を軽減するインフラストラクチャを構成する。サービスの**障害復旧の自動化**、スケーラビリティによる**高可用性**の確保、モニタリングと自動化などを考慮。項目として基盤、変更管理、障害管理がある。

*該当するサービス

基盤(IAM、VPC、Auto Scaling、ELB、Cloud Formation)、変更管理(Cloud Trail、AWS Config)、 障害管理(Cloud Watch) パフォーマンスの効率化

リソース使用量の軽減。レビュー、モニタリング、トレードオフコンピューティングリソースの効率的な使用について。コンピューティングリソースやAWSインフラの効率化など。

*該当するサービス

コンピューティング(Autoscaling、Lambda)、ストレージ(EBS、S3、Glacier、EFS)、データベース(RDS、

DynamoDB、Elastic Search、Aurora、Redshift)、容量と時間のトレードオフ(CloudFront、ElastiCache)

■ コストの最適化

不要なコスト削減、変動コストへと転換。また、スケールの利用、データセンターへの投資不要化など。

*該当するサービス

需要と供給の一致(Autoscaling)、コスト効率の高いリソース(EC2の購入方式、Trusted Advisor)、

支出の認識(Cloud Watch、見積もりツール)、継続した最適化(AWS最新情報、Trusted Advisor)

- 接続可能性インフラに関連するエネルギー使用率を80%削減する。
- AWSの対応 サーバー使用率、電力と冷却の効率、カスタムデータセンターの設計、 2025年までに100%の再生エネルギーでAWSを運用する。
- ユーザーの対応
 効率的なストレージ技術の使用、コードの効率化、効率的なインフラのデプロイとスケーリング、 効率的なアプリケーションデザイン、データの効率的なデザイン活用
 *具体的なベストプラクティス

所在地に合わせた**地理的配置、コード領域の最適化、ユーザーデバイスの最適化、不要なデータの削除、** データ移動の最小化、GPU使用の最適化

クラウドの設計

クラウド設計原則

クラウドにおいて設計原則は次のとおり。

必要なキャパシティを勘に頼らない。

本番規模でのシステムテストを行う。

アーキテクチャ試行の回数を増やすために自動化を取り入れる。

<u>発展的なアーキテクチャ</u>を取り入れる。

データ計測に基づいてアーキテクチャを決定する。

本番で想定されるトラブルをあらかじめテストし、対策する。

■ 信頼性

システムが一定期間機能する確率のこと。システムにはハードウェア、ソフトウェア、ファームウェアが含まれる。指標としてMTBF(平均故障間隔)があげられる。

*故障率 = 障害数/サービスの合計時間

予測が不要である。

- *回復性の高いアーキテクチャとは 復旧の自動化(Cloud Formation、AMI、Opsworksなど)できる、スケーラブルであること、またキャパシティ
- *該当するサービス
 Cloud Watch、Auto Scaling、マルチAZ(RDS)、Elastic Cache、DynamoDB

■ 可用性 正常に動作する時間の割合のこと。稼働率が該当する。

- 高可用性(High Availability)
 高可用性とは、ダウンタイムが最小限であり、ユーザーの介入が最小限で復旧する。
 *例 99.999% → 5分/1年
- 耐障害性・・・冗長性と運用を継続する力のこと。
- 復元可能性・・・災害発生後のサービスの復元
- スケーラビリティ・・・拡張性に対応している。