

Developing Serverless Solutions on AWS

AWS クラスルームトレーニング

コースの説明

このコースでは、AWS Lambda とその他の AWS サーバーレスプラットフォームのサービスを使用してサーバーレスアプリケーションを構築するためのベストプラクティスを紹介し、開発者はそれを実践します。簡単なトピックから複雑なトピックへと進化するハンズオンラボで、AWS フレームワークを使用してサーバーレスアプリケーションをデプロイします。コース全体で AWS ドキュメントを使用し、クラスルームの枠を超えて学習および問題解決をするための信頼性の高い方法を開発します。

レベル	実施形式	所要時間
中級	クラスルームトレーニング、ハンズオンラボ、動画、知識確認テスト、グループ演習	3 日間

アクティビティ

このコースには、プレゼンテーション、ハンズオンラボ、デモンストレーション、動画、知識確認テスト、グループ演習が含まれます。

コースの目標

このコースでは、以下について学習します。

- 適切な AWS サービスを使用して、イベント駆動型のベストプラクティスをサーバーレスアプリケーション設計に適用する
- サーバーレス開発への移行に伴う課題とトレードオフを特定し、開発組織と環境に適した推奨事項を作成する
- AWS マネージドサービスを相互に接続するパターンを使用してサーバーレスアプリケーションを構築し、サービスクォータ、利用可能な統合、呼び出しモデル、エラー処理などのサービス特性を考慮する
- AWS CloudFormation、AWS Amplify、AWS サーバーレスアプリケーションモデル (AWS SAM)、AWS クラウド開発キット (AWS CDK) など、Infrastructure as Code を実現するための利用可能な選択肢を比較する
- エラー処理、ロギング、環境の再利用、レイヤーの使用、ステートレス、冪等性、同時実行数とメモリの構成を含む Lambda 関数の記述にベストプラクティスを適用する

Developing Serverless Solutions on AWS

AWS クラスルームトレーニング

- オブザーバビリティとモニタリングを構築するためのベストプラクティスをサーバーレスアプリケーションに適用する
- セキュリティのベストプラクティスをサーバーレスアプリケーションに適用する
- サーバーレスアプリケーションにおけるスケーリングの重要な考慮事項を特定し、各考慮事項を方法、ツール、またはベストプラクティスに一致させて管理する
- AWS SAM、AWS CDK、AWS 開発者ツールを使用して CI/CD ワークフローを設定し、サーバーレスアプリケーションのデプロイを自動化する
- サーバーレスリソースのリストを作成してそれを積極的に維持し、継続的なサーバーレス開発とサーバーレスコミュニティとのエンゲージメントを支援する

対象者

このコースは以下のような方を対象としています。

- サーバーレスの基礎知識があり、AWS クラウドでの開発経験を持つ開発者

前提条件

このコースを受講するにあたって、以下の前提条件を満たしておくことをお勧めします。

- AWS クラウドアーキテクチャの基本を理解している
- AWS でのアプリケーション開発を理解している (Developing on AWS クラスルームトレーニングの修了時と同等の知識を習得している)
- サーバーレスデジタルトレーニングの修了時と同等の知識を習得している (AWS Lambda Foundations および Amazon API Gateway for Serverless Applications)

登録

<https://www.aws.training/training/schedule?courseId=53785&countryName=JP&trainingProviderId=1>

Developing Serverless Solutions on AWS

AWS クラスルームトレーニング

コースの概要

1 日目

モジュール 0: はじめに

- 構築するアプリケーションの概要
- コースリソースへのアクセス (受講者ガイド、ラボガイド、オンラインコース補足)

モジュール 1: サーバーレス思考

- サーバーレスアプリケーションを構築するためのベストプラクティス
- イベント駆動型設計
- イベント駆動型のサーバーレスアプリケーションを支える AWS サービス

モジュール 2: API 駆動の開発と同期イベントソース

- 同期イベントソース
- Lambda へのサーバーレス API プロキシ用 Amazon API Gateway
- GraphQL API と AWS AppSync

モジュール 3: 認証、認可、アクセスコントロールの概要

- 認証と認可
- API Gateway オーソライザー
- Amazon Cognito を使用したサーバーレス認証と認可

モジュール 4: サーバーレスデプロイのフレームワーク

- Infrastructure as Code
- CloudFormation
- AWS Cloud Development Kit (AWS CDK)
- AWS Serverless Application Model (AWS SAM)
- Amplify フレームワーク

モジュール 5: Amazon EventBridge と Amazon SNS を使用したコンポーネントの疎結合化

- Lambda の非同期イベントソース

Developing Serverless Solutions on AWS

AWS クラスルームトレーニング

- 非同期イベントソースとしての EventBridge
- Try-it-out 演習: EventBridge ルールの設定
- 非同期イベントソースとしての Amazon SNS
- Try-it-out 演習: Amazon SNS トピックとサブスクリプションの設定
- Amazon SNS と Amazon EventBridge

モジュール 6: キューとストリームを使用したイベント駆動型開発

- Lambda がイベントソースマッピングを使用してイベントを呼び出す仕組み (ポーリング)
- Try-it-out 演習: Amazon SQS コンソールの使用
- Lambda が Amazon SQS キューをイベントソースとして管理する仕組み
- Lambda がイベントソースとして Kinesis データストリーム と Amazon DynamoDB ストリームを管理する仕組み

ハンズオンラボ

- ハンズオンラボ 1: シンプルなサーバーレスアプリケーションのデプロイ
- ハンズオンラボ 2: Amazon EventBridge を使用したメッセージのファンアウト

2 日目

モジュール 7: 優れた Lambda 関数を書く

- Lambda のライフサイクルに関するベストプラクティス
- 関数設定
- 関数コード
- バージョンとエイリアス
- 関数の実行とテスト
- Lambda エラー処理

モジュール 8: Step Functions を使用したオーケストレーション

- Step Functions を使用したステートマシンの作成
- Try-it-out 演習: Step Functions コンソール

Developing Serverless Solutions on AWS

AWS クラスルームトレーニング

- ステートを使用したオーケストレーションロジック
- 入力と出力の強化
- ワークフロータイプの選択
- Try-it-out 演習: Step Functions の Express ワークフローをトラブルシューティングする

モジュール 9: オブザーバビリティとモニタリング

- オブザーバビリティの 3 つの柱
- Amazon CloudWatch Logs
- Try-it-out 演習: API Gateway のログ記録と X-Ray を有効化する
- Try-it-out 演習: CloudWatch Logs
- AWS X-Ray
- Try-it-out 演習: X-Ray
- Amazon CloudWatch メトリクス
- Try-it-out 演習: CloudWatch メトリクス
- CloudWatch メトリクスと埋め込みメトリックフォーマット
- Try-it-out 演習: CloudWatch ServiceLens

ハンズオンラボ

- ハンズオンラボ 3: AWS Step Functions を使用したワークフローオーケストレーション
- ハンズオンラボ 4: オブザーバビリティとモニタリング

3 日目

モジュール 10: サーバーレスアプリケーションのセキュリティ

- セキュリティの脅威
- セキュリティのベストプラクティスをサーバーレスアプリケーションに適用することに関する考慮事項
- すべてのレイヤーにセキュリティを適用する
- アイデンティティとアクセスコントロール

Developing Serverless Solutions on AWS

AWS クラスルームトレーニング

- データ保護
- 攻撃からの保護
- 監査、トレーサビリティ、オートメーション

モジュール 11: サーバーレスアプリケーションにおけるスケール処理

- サーバーレスアプリケーションのスケールに関する考慮事項
- API Gateway を使用してスケールを管理する
- Lambda 同時実行スケール
- Lambda を使用して異なるイベントソースをスケールする方法
- 選択肢を継続的に評価する

モジュール 12: デプロイパイプラインの自動化

- サーバーレスアプリケーションにおける CI/CD の重要性
- サーバーレスパイプラインのツール
- サーバーレスデプロイ用の AWS SAM 機能
- 自動化のベストプラクティス

ハンズオンラボ

- ハンズオンラボ 5: サーバーレスアプリケーションのセキュリティ保護
- ハンズオンラボ 6: AWS でのサーバーレス CI/CD