**OpenHack –** **サーバーレス**

**概要**

**この OpenHack で、参加者は** Azure Functions、Logic Apps、Event Grid、Service Bus、Event Hubs、Cosmos DB などの最先端のコンピューティング サービスからなる Azure サーバーレス ソリューションのビルドとデプロイをすばやく行うことができます。

**この OpenHack では実際のシナリオをシミュレートします。**アイスクリーム会社がサービスとしてのプラットフォーム (PaaS) を使用して API のビルドとリリースを行い、ディストリビューターのアプリケーションに統合するというものです。

**"hacking" の期間中、参加者は次のことに注目します。**

1.サーバーレス関数、Web API、およびこれらをサポートするための CI/CD パイプラインを構築する

2.基幹業務アプリのワークフローを統合し、ユーザーとデータのテレメトリを処理し、KPI に基づいたレポートを作成するために、サーバーレス テクノロジを実装する

**参加者は OpenHack の終了までに、**システム間のワークフローを作成し、イベント、ファイル、データ インジェストを処理できる完全なサーバーレス ソリューションを構築します。

**テクノロジ**

[Azure Functions](https://azure.microsoft.com/ja-jp/services/functions/)、[Logic Apps](https://azure.microsoft.com/ja-jp/services/logic-apps/)、[Event Grid](https://azure.microsoft.com/ja-jp/services/event-grid/)、[Cosmos DB](https://azure.microsoft.com/ja-jp/services/cosmos-db/)、[API Management](https://azure.microsoft.com/ja-jp/services/api-management/)、[Azure Event Hubs](https://azure.microsoft.com/ja-jp/services/event-hubs/)、[Azure DevOps](https://docs.microsoft.com/ja-jp/azure/devops/) または [GitHub](https://github.com/) (チームで選択), [Azure Monitor](https://azure.microsoft.com/ja-jp/services/monitor/)、[Dynamics 365/Office 365](https://www.microsoft.com/dynamics365-predictiveinsights/?&OCID=AID2100366_SEM_XwNeogAAAIOzPhXi:20200729193938:s&msclkid=40ce4b585b7f18ebee5514798b0d4f1b&ef_id=XwNeogAAAIOzPhXi:20200729193938:s)、[Cognitive API](https://azure.microsoft.com/ja-jp/services/cognitive-services/)、[Service Bus](https://azure.microsoft.com/ja-jp/services/service-bus/)

**前提条件**

**知識に関する前提条件**

この OpenHack を最大限に活用して成功するには、参加者は、API 統合の経験があり、作業するために選択した言語に関する深い知識を持っていることが推奨されます。リストされているテクノロジに慣れている参加者はより迅速に進むことができます。DevOps の基礎に関する実用的な知識は役立ちます。

[Azure の基礎の知識が必要です。](https://docs.microsoft.com/ja-jp/learn/paths/azure-fundamentals/)

**言語固有の前提条件**

* 次のプログラミング言語のうち少なくとも 1 つでハンズオン コーディングが必要です。C#、JavaScript、Node、Python.

**ツールの前提条件**

ツールのダウンロードやインストールの遅延を避けるため、OpenHack に先だって以下を準備してください。

* Windows 10 (1703 以降)、Mac OSX (10.12 以降)、または[これらの Ubuntu バージョン](https://github.com/Azure/azure-functions-core-tools#linux)のいずれかを実行する最新のノート PC
* 最新バージョンの [Azure CLI](https://docs.microsoft.com/ja-jp/cli/azure/install-azure-cli?view=azure-cli-latest)のダウンロード
* 任意の IDE以下を使用する場合:
  + **Windows 用 Visual Studio**: 最新の[Visual Studio](https://visualstudio.microsoft.com/) (‘Azure development’ ワークロードを選択)、[Azure Functions および Web Jobs Tools 拡張機能](https://docs.microsoft.com/ja-jp/azure/azure-functions/functions-develop-vs#check-your-tools-version)のインストール
  + **Windows、OSX、または Linux での Visual Studio Code**:お使いの OS 向けの最新の [Visual Studio Code](https://code.visualstudio.com/download) 、[Azure Functions 拡張機能、](https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-azurefunctions)および [Azure Functions Core Tools](https://docs.microsoft.com/ja-jp/azure/azure-functions/functions-run-local?tabs=windows%2Ccsharp%2Cbash#v2) のインストール

Microsoft では、登録時に指定した優先言語を使用するチームにお客様が配置されるよう最善を尽くします。OpenHack の準備として、下で優先言語を特定し、下にリストされているツールをダウンロードしてください。

* C# .NET Core
  + [サポートされているプラットフォーム](https://code.visualstudio.com/docs/supporting/requirements#_platforms)での [Visual Studio 2019](https://visualstudio.microsoft.com/) または [Visual Studio Code](https://code.visualstudio.com/download)
    - Visual Studio Code [C# 拡張機能](https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.csharp) が推奨されます
  + .NET Core 2.2
* Java または Maven
  + [Java Developer Kit](https://aka.ms/azure-jdks) バージョン 8
  + [Apache Maven](https://maven.apache.org/) バージョン 3.0 以降
* JavaScript
  + [サポートされているプラットフォーム](https://code.visualstudio.com/docs/supporting/requirements#_platforms)での [Visual Studio Code](https://code.visualstudio.com/download)
  + [Node.js](https://nodejs.org/jp/) アクティブ LTS およびメンテナンス LTS バージョン (8.11.1 および 10.14.1 を推奨)
* Python
  + [Python 3.6.x](https://www.python.org/downloads/)(公式 CPython ディストリビューション)
    - この[既知の問題](https://github.com/protocolbuffers/protobuf/issues/5046)のため、3.6.8 64 ビットが推奨されます
    - Python インタープリターの選択は、Windows の VS Code ユーザーには厄介になる場合があります
* Mac OSX および Linux ユーザーについては [Azurite v2](https://www.nuget.org/packages/Azurite/) のインストールが推奨されます

**開発環境の構成**

* ありません。

**リンクとリソース**

* [Application Insights により Azure Web アプリでのページの読み込み時間をキャプチャして表示する](https://docs.microsoft.com/ja-jp/learn/modules/capture-page-load-times-application-insights/)
* [Azure での API 統合の設計](https://docs.microsoft.com/ja-jp/learn/paths/architect-api-integration/)
* [Azure Stream Analytics を使用してデータを変換する](https://docs.microsoft.com/ja-jp/learn/modules/transform-data-with-azure-stream-analytics/)
* [サーバーレス アプリケーションの作成](https://docs.microsoft.com/ja-jp/learn/paths/create-serverless-applications/)

**学習後の推奨事項**

* [Azure Pipelines を使用して Azure Functions のデプロイを自動化する](https://docs.microsoft.com/ja-jp/learn/modules/deploy-azure-functions/)
* [DevOps OpenHack](https://openhack.microsoft.com/)
* [Azure でメッセージ ブローカーとサーバーレス アプリケーションを設計する](https://docs.microsoft.com/ja-jp/learn/paths/architect-messaging-serverless/)
* [Durable Functions を使って実行時間の長いサーバーレス ワークフローを作成する (モジュール)](https://docs.microsoft.com/ja-jp/learn/modules/create-long-running-serverless-workflow-with-durable-functions/)

**課題**

**課題 1:環境構成**

この課題では、開発環境でサーバーレス アプリケーションを開発するための準備を整えます。

学習目標:

* Azure Functions をローカルにビルドしてテストするために必要なソフトウェアと前提条件をインストールする

**課題 2:初めてのサーバーレス関数とワークフローを作成する**

この課題では、初めての Azure 関数とロジック アプリを作成してデプロイします。

学習目標:

* 初めての Web API の関数を作成して Azure クラウドにデプロイする
* Azure Portal に慣れて、API をホストするために必要なリソースを作成する
* 初めての Logic Apps ワークフローをポータルで直接ビルドする

**課題 3:ビジネス ニーズをサポートするために API のフル セットを展開および構築する**

この課題では、データ ストアにデータを保持、取得し、CI/CD パイプラインを構成するための API を構築します。

学習目標:

* JSON でさまざまな種類のデータを受け入れて出力する複数の Web API を構築する経験を積む
* ローカル マシンから呼び出しを行って、API のローカルおよびクラウドのテストを体験する
* ソース管理から継続的インテグレーションと継続的デプロイ (CI/CD) パイプラインを構築する

**課題 4:API の管理レイヤーをデプロイして API をモニターおよび追跡する**

この課題では、API のモニタリングを有効にして、API 管理戦略を考案します。

学習目標:

* クラウドでホストされた API に関するテレメトリをキャプチャしてレポートする方法を学ぶ
* API を管理し、共通のベース エンドポイントを介して公開するための API 管理レイヤーの構築を体験する

**課題 5:ワークフロー プロセスを構築する**

この課題では、CRM システムと統合するビジネス プロセス ワークフローを構築します。

学習目標:

* グラフィカル ユーザー インターフェイス エディターを使用して、Logic Apps を使用してワークフローを構築する経験を積む
* ドラッグ アンド ドロップ コネクタを使用して、基幹業務アプリでフローを作成し、メールで連絡先に通知する体験をする

**課題 6:大量の非構造化データを処理する**

この課題では、ファイルをバッチ プロセスで処理し、データ ストアに保持します。

学習目標:

* ストレージ アカウントに入ってくるバッチ ファイルを処理するソリューションを構築する
* フラット ファイルを解析してデータ ベース (構造化または非構造化) に保存する方法を確認する

**課題 7:大量の受信イベントを処理する**

この課題では、Event Hubs からのメッセージのバッチを処理し、データ ストアに保持します。また、以前作成した監視アプローチを拡張して、実行中の Azure Functions インスタンスの数を追跡します。

学習目標:

* JSON の販売イベントを解析してバックエンドのデータ ストアに保存するように API を拡張する方法を確認する
* App Insights を使用して、Azure Functions がスケールアップしたインスタンスの数を追跡した方法を実証する

**課題 8:メッセージングのパターンと Virtual Network 統合**

この課題では、メッセージをフィルター処理するメッセージング ソリューション機能を構成します。また、Virtual Network 統合を使用してデータをプライベート データ ストアに保存します。

学習目標:

* フィルタリング ルールでパブリッシャーとサブスクライバーのメッセージング パターンを使用して、サブスクライバーが特定のメッセージを処理できるようにする方法を確認する
* 仮想ネットワークのアクセス制限を利用する Azure リソースを操作できるサーバーレス ソリューションを構築する

**課題 9:ユーザー センチメントに基づくアラート**

この課題では、機械学習機能を現在のソリューションに統合し、ユーザー センチメントを検出してアラートします。

学習目標:

* 感情分析を活用して、フィードバックに基づいてユーザー センチメントを分析する
* センチメントを一括でレビューし、センチメント スコア製品が受信しているものを担当の連絡先にアラートする

**課題 10:まとめ**

この課題では、完全なソリューションから導出された主要なビジネス メトリックを要約したレポートを作成します。

学習目標:

* すべてをまとめて、物事がエンド ツー エンドでどのように流れるかを実証する
* 製品の動作に関する分析情報をレポートする

**価値の提案**

* Azure サーバーレス テクノロジを使用して、Azure のクラウド ネイティブのイベント ドリブン アプリケーション (中間層およびバックエンド) を構築する
* 非同期通信用のメッセージング サービスを使用してスケーラブルなイベント ドリブン アプリケーションを構築することについての認識を深める
* 迅速なアプリケーション開発に集中できるよう、インフラストラクチャ関連のタスクと考慮事項を開発者から取り除く
* ビジネスの成果達成のために Azure サービスを使用するタイミングと選択する方法について決定を下すことを支援する
* サーバーレス用の CI/CD 構築と、イベントの大規模な処理のためのベストプラクティスを学ぶ

**技術シナリオ**

サーバーレス イベントの処理、ワークフロー、さまざまなシステムとの統合、ストリーミングと分析、VNET 統合、高い列データのバッチ処理によって API を構築します。

* エンタープライズ統合 – さまざまな基幹業務システムとアプリケーションをまとめ、大規模なインフラストラクチャ フットプリントをプロビジョニングする必要なくプロセスを調整します
* DevOps プラクティス – コード管理のソース管理のベスト プラクティスを支援するために開発されたサービス用の CI および CD を構築します
* スケーリングおよびログ – イベントが急増した場合のサービスのスケーリングの側面を理解し、ビジネス データを収集するためのアプリケーション分析情報を視覚化します

**対象者**

* 対象者:
  + Microsoft – CSE、CSA、GBB、ATT、SE、TPM
  + 顧客 – アプリ開発者
* 対象業種:さまざまな業界
* 顧客プロファイル:例:
  + 最新のアプリケーション開発 – クラウド ネイティブ アプリケーションを構築し、Azure で サービスとしてのプラットフォーム (PaaS) サービスを使用して、インフラストラクチャ管理のオーバーヘッドを取り除き、アプリケーションの市場投入までの時間を短縮したいと考えているお客様。開発された迅速なアプリケーションに重点を置いています。
  + イベント ドリブンの分散アーキテクチャ – システムとサービス間の通信をメッセージング サービスに依存し、効果的なスケーリングと管理のために非同期処理に焦点を当てた、設計されたソリューションを求めているお客様。

**登録時の質問**

|  |  |  |
| --- | --- | --- |
| **必須** | **フィールド** | **回答オプション** |
| はい | 現時点で、Azure の使用についてどの程度理解していますか? | まったく理解していない/少し理解している Azure でパイロット作業を行ったことが少しある 現在、クラウドには Azure を利用している |
| はい | Azure Functions、Event Grid、Logic Apps に費やした時間はどのくらいですか? | [チェックボックスまたはラジオ ボタン]  3 か月以上  3 - 6 か月  6 か月 - 1 年  1 年以上 |
| はい | お好みのアプリケーション プログラミング言語は何ですか? | [ドロップダウン]  C#  Java  JavaScript  Python  その他 |

**コーチ調達のための質問**

次の質問は、適切な知識を持つコーチを調達するために使用できます。詳細情報については、[OpenHack プレイブック](https://aka.ms/openhackplaybook)のコーチの選定に関するページをご覧ください。

|  |  |  |
| --- | --- | --- |
| 質問 # | 質問 | 回答の要注目点 |
| 1 | OpenHack での過去の経験にはどのようなものがありますか? | * オリジナル (v1) のサーバーレス OpenHack.ですべての課題を正常に完了している必要があります。 * サーバーレス OpenHack 2.0 の更新された課題を完了している必要があります。 すべての課題を完了していることが強く推奨されます。 |
| 2 | Durable Functions での経験を含め、Azure Functions での経験を説明してください。 | * C#、Java、Node.js、および/または Python を使用して Azure Functions を開発およびデプロイした経験が必要です。 * Azure DevOps を使用して CI/CD パイプラインを構成した経験が必要です。 * C#、Java、Node.js、および Python を使用して Azure Functions をデプロイした経験があるコーチが必要です。 * Durable Functions を使用した開発経験がある回答者を優先します。 |
| 3 | Logic Apps を使用して Azure Event Grid からイベントを受信し、イベント データに基づいてバッチ ベースの処理を行った経験について説明してください。 | * Azure Event Grid でイベントを生成し、それらのイベントをロジック アプリ トリガーを使用して受信するように Azure Storage を構成する方法を理解していることが強く推奨されます。 * Logic Apps を使用してバッチ ベースの処理を行ったことのある回答者を優先します。 |
| 4 | Azure API Management で Azure Functions と Logic Apps の使用を構成した経験 (特に Open API 仕様の作成について) を説明してください。 | * Open API (Swagger) 仕様の使用を含め、Functions および/または Logic Apps と連携するように API Management を構成した経験があるコーチを強くお勧めします。 |
| 5 | Event Hubs、Kafka、または Event Grid を使用したエンゲージメントについて説明してください。 | * Event Hubs および Event Grid (特にAzure Storage イベント) を使用した経験があることが推奨されます。 * Event Hubs および Kafka (特に Event Hubs と Kafka 対応エンドポイント) を使用したことのある回答者を優先します。 |
| 6 | Log Analytics を使用して Azure Functions をモニターした経験について説明してください。 | * Log Analytics を使用して、クエリ (チャートを含む) を作成し、Azure Functions に関連するメトリックに基づいてアラートを構成した経験が必要です。 |

**FAQ**

Q:FaaS を使用している顧客はこの OpenHack に参加すべきですか?

A: はい。コンテンツでは、CI/CD、スケーリング、DR、VNet 要件に焦点を当てて、エンド ツー エンドのアプリケーション サイクルを強調しているためです。

Q:参加者が事前に参加しておくべき OpenHacks の推奨フローはありますか?

A: いいえ。前述の「*前提条件*」を参照してください。