**OpenHack –** **Serverless**

**Vue d’ensemble**

**Cet OpenHack permet aux participants** de créer et de déployer rapidement des solutions serverless Azure composées de services de calcul de pointe tels qu’Azure Functions, Logic Apps, Event Grid, Service Bus, Event Hubs et Cosmos DB.

**Ce OpenHack simule un scénario réel** dans lequel une entreprise de crèmes glacées souhaite utiliser une solution PaaS pour créer et publier une API qui s’intègre à l’application de son distributeur.

**Pendant le hack, les participants se concentreront sur :**

1. Création de fonctions serverless, d’API web et d’un pipeline CI/CD pour les prendre en charge

2. Implémentation de technologies serverless pour intégrer les flux d’applications métier, traiter la télémétrie des utilisateurs/données et créer des rapports alignés sur les indicateurs clés de performance de l’entreprise.

**À la fin de l’OpenHack, les participants ont développé une solution technique** complète serverless qui peut créer des workflows entre les systèmes et gérer les événements, les fichiers et l’ingestion des données.

**Technologies**

[Azure Functions](https://azure.microsoft.com/fr-fr/services/functions/), [Logic Apps](https://azure.microsoft.com/fr-fr/services/logic-apps/), [Event Grid](https://azure.microsoft.com/fr-fr/services/event-grid/), [Cosmos DB](https://azure.microsoft.com/fr-fr/services/cosmos-db/), [API Management](https://azure.microsoft.com/fr-fr/services/api-management/), [Azure Event Hubs](https://azure.microsoft.com/fr-fr/services/event-hubs/), [Azure DevOps](https://docs.microsoft.com/fr-fr/azure/devops/) ou [GitHub](https://github.com/) (choix de l’équipe), [Azure Monitor](https://azure.microsoft.com/fr-fr/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 APIs](https://azure.microsoft.com/fr-fr/services/cognitive-services/), [Service Bus](https://azure.microsoft.com/fr-fr/services/service-bus/)

**Prérequis**

**Connaissances prérequises**

Pour réussir et tirer le meilleur parti de cet OpenHack, il est fortement recommandé aux participants d’avoir une expérience préalable de l’intégration d’API et une connaissance approfondie du langage dans lequel ils ont choisi de travailler. Les participants qui sont familiarisés avec les technologies énumérées pourront progresser plus rapidement. Une connaissance pratique des principes fondamentaux de DevOps est utile.

Connaissance requise des [Principes de base d’Azure](https://docs.microsoft.com/fr-fr/learn/paths/azure-fundamentals/).

**Conditions préalables spécifiques au langage**

* Une expérience de codage pratique est requise dans au moins un des langages de programmation suivants : C#, JavaScript, Node ou Python.

**Outils prérequis**

Pour éviter tout retard dans le téléchargement ou l’installation des outils, préparez les éléments suivants avant le OpenHack :

* Un ordinateur portable moderne sous Windows 10 (1703 ou version ultérieure), macOS X (10.12 ou version ultérieure) ou l’une de [ces variantes Ubuntu](https://github.com/Azure/azure-functions-core-tools#linux)
* Téléchargez la dernière version d’[Azure CLI](https://docs.microsoft.com/fr-fr/cli/azure/install-azure-cli?view=azure-cli-latest).
* L’IDE de votre choix. Si vous utilisez :
  + **Visual Studio pour Windows**: Installez [la dernière version de Visual Studio](https://visualstudio.microsoft.com/) avec la charge de travail ’Azure development’ sélectionnée et l’extension [Azure Functions et Web Jobs Tools](https://docs.microsoft.com/fr-fr/azure/azure-functions/functions-develop-vs#check-your-tools-version)
  + **Visual Studio Code sur Windows, OS X ou Linux**: Installez la dernière version de [Visual Studio Code](https://code.visualstudio.com/download) pour votre système, [l’extension Azure Functions](https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-azurefunctions) et [Azure Functions Core Tools](https://docs.microsoft.com/fr-fr/azure/azure-functions/functions-run-local?tabs=windows%2Ccsharp%2Cbash#v2)

Nous ferons de notre mieux pour vous placer dans une équipe qui utilisera le langage préféré que vous avez indiqué lors de votre inscription. Veuillez identifier votre langage préféré ci-dessous et télécharger les outils répertoriés en dessous en Préparation pour le OpenHack.

* C# .NET Core
  + [Visual Studio 2019](https://visualstudio.microsoft.com/) ou [Visual Studio Code](https://code.visualstudio.com/download)sur une [plateforme prise en charge](https://code.visualstudio.com/docs/supporting/requirements#_platforms)
    - Il est recommandé d’utiliser l’[extension C#](https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.csharp) de Visual Studio Code
  + .NET Core 2.2
* Java/Maven
  + [Java Developer Kit](https://aka.ms/azure-jdks) version 8
  + [Apache Maven](https://maven.apache.org/), version 3.0 ou ultérieure
* JavaScript
  + [Visual Studio Code](https://code.visualstudio.com/download) sur une [plateforme prise en charge](https://code.visualstudio.com/docs/supporting/requirements#_platforms)
  + [Node.js](https://nodejs.org/fr/) versions Active LTS et Maintenance LTS (8.11.1 et 10.14.1 recommandées)
* Python
  + [Python 3.6.x](https://www.python.org/downloads/)(distribution officielle CPython)
    - La version 3.6.8 64 bits est recommandée en raison de ce [problème connu](https://github.com/protocolbuffers/protobuf/issues/5046)
    - La sélection de l’interpréteur Python peut poser des problèmes aux utilisateurs de VS Code sous Windows
* Il est recommandé aux utilisateurs de macOS X et Linux d’installer [Azurite v2](https://www.nuget.org/packages/Azurite/)

**Configuration de l’environnement de développement**

* Aucun.

**Liens et ressources**

* [Capturer et afficher les temps de chargement des pages dans votre web app Azure avec Application Insights](https://docs.microsoft.com/fr-fr/learn/modules/capture-page-load-times-application-insights/)
* [Concevoir l’intégration d’API dans Azure](https://docs.microsoft.com/fr-fr/learn/paths/architect-api-integration/)
* [Transformer des données à l’aide d’Azure Stream Analytics](https://docs.microsoft.com/fr-fr/learn/modules/transform-data-with-azure-stream-analytics/)
* [Créer des applications serverless](https://docs.microsoft.com/fr-fr/learn/paths/create-serverless-applications/)

**Recommandations post-apprentissage**

* [Automatiser les déploiements Azure Functions avec Azure Pipelines](https://docs.microsoft.com/fr-fr/learn/modules/deploy-azure-functions/)
* [OpenHack DevOps](https://openhack.microsoft.com/)
* [Concevoir une répartition des messages et des applications serverless dans Azure](https://docs.microsoft.com/fr-fr/learn/paths/architect-messaging-serverless/)
* [Créer un workflow serverless de longue durée avec Durable Functions (module)](https://docs.microsoft.com/fr-fr/learn/modules/create-long-running-serverless-workflow-with-durable-functions/)

**Problématiques**

**Défi 1 : Configuration de l’environnement**

Dans ce défi, vous allez préparer votre environnement de développement pour développer des applications serverless.

Objectifs d’apprentissage :

* Installez les logiciels requis et les éléments requis pour créer et tester les fonctions Azure localement

**Défi 2 : Créer votre première fonction et votre premier workflow serverless**

Dans ce défi, vous allez créer et déployer votre première fonction Azure et Logic App.

Objectifs d’apprentissage :

* Créez votre première API web Functions localement et déployez-la dans le cloud Azure
* Familiarisez-vous avec le portail Azure pour créer les ressources nécessaires pour héberger vos API
* Générez votre premier workflow Logic Apps directement sur le portail

**Défi 3 : Développer et créer un ensemble complet d’API pour répondre aux besoins de l’entreprise**

Dans ce défi, vous créez une API pour conserver et récupérer des données d’un magasin de données et configurez un pipeline CI/CD.

Objectifs d’apprentissage :

* Obtenez de l’expérience de création de plusieurs API web qui acceptent et génèrent un type de données différent en JSON
* Expérimentez les tests locaux et cloud des API en effectuant des appels à partir de votre ordinateur local
* Créez un pipeline de déploiement continu et d’intégration continue (CI/CD) à partir du contrôle de code source

**Défi 4 : Déployer une couche de gestion pour les API pour surveiller et suivre vos API**

Dans de défi, vous permettrez d’analyser les API, et établirez une stratégie de gestion des API.

Objectifs d’apprentissage :

* Découvrez comment capturer et signaler la télémétrie concernant les API hébergées sur le cloud
* Découvrez la création d’une couche de gestion d’API pour gérer les API et les exposer via un point de terminaison de base commun

**Défi 5 : Créer un processus de workflow**

Dans ce défi, vous créez un workflow de processus d’entreprise qui s’intègre à un système CRM.

Objectifs d’apprentissage :

* Acquérez de l’expérience en utilisant un éditeur d’interface utilisateur graphique pour créer des workflows à l’aide de Logic Apps
* Découvrez les connecteurs par glisser-déplacer pour mettre en place des applications métier afin de créer un flux et notifier les contacts par e-mail

**Défi 6 : Traiter une grande quantité de données non structurées**

Dans ce défi, vous traitez les fichiers dans un processus de traitement par lots et les conservez dans un magasin de données

Objectifs d’apprentissage :

* Créez des solutions pour traiter des fichiers de commandes entrant dans un compte de stockage
* Découvrez comment analyser les fichiers plats et les stocker dans une base de données (structurée ou non structurée)

**Défi 7 : Traiter une grande quantité d’événements entrants**

Dans de défi, vous traitez des lots de messages provenant d’Event Hub et les conserverez dans un magasin de données. Vous étendez aussi l’approche de surveillance créée précédemment pour suivre le nombre d’instances Azure Function en cours d’exécution.

Objectifs d’apprentissage :

* Découvrez comment étendre les API pour analyser les événements de ventes JSON et les stocker dans un magasin de données principal
* Démonstration du suivi du nombre d’instances auxquelles vos fonctions Azure ont été adaptées avec App Insights

**Défi 8 : Modèles de messagerie et intégration de réseau virtuel**

Dans ce défi, vous configurez une solution de messagerie capable de filtrer les messages, ainsi que d’utiliser l’intégration de réseaux virtuels pour enregistrer les données dans un magasin de données privé.

Objectifs d’apprentissage :

* Découvrez comment utiliser des modèles de messagerie éditeur/abonné avec des règles de filtrage pour permettre aux abonnés de traiter des messages spécifiques
* Créer une solution serverless pouvant fonctionner avec des ressources Azure qui utilisent des restrictions d’accès au réseau virtuel

**Défi 9 : Alertes basées sur les sentiments de l’utilisateur**

Dans ce défi, vous intégrez des capacités de Machine Learning dans la solution actuelle pour détecter et alerter sur le sentiment des utilisateurs.

Objectifs d’apprentissage :

* Tirer parti de l’analyse des sentiments pour évaluer les sentiments des utilisateurs en fonction des commentaires
* Passez en revue les sentiments dans le traitement par lots et avertissez les contacts responsables des opinions produit reçues

**Défi 10 : Exemple complet**

Dans ce défi, vous créez un rapport qui résume les principales mesures commerciales dérivées de la solution complète.

Objectifs d’apprentissage :

* Rassemblez tous les éléments et montrez comment les éléments sont transmis de bout en bout
* Rapportez les insights sur la façon dont les produits se comportent

**Proposition de valeur**

* Créer des applications d’événement natives au cloud Azure (niveau intermédiaire et back-end) en utilisant les technologies Azure serverless
* Sensibilisation à la création d’applications évolutives, axées sur les événements, avec des services de messagerie pour la communication asynchrone
* Débarrasser les développeurs des tâches et considérations liées à l’infrastructure pour qu’ils puissent se concentrer sur le développement rapide des applications
* Contribuer à prendre des décisions sur l’utilisation et le choix des services Azure pour atteindre les objectifs de l’entreprise
* Apprendre les meilleures pratiques pour mettre en place un système CI/CD pour le serverless et traiter les événements à l’échelle

**Scénarios techniques**

API de génération avec le traitement d’événements serverless, les workflows et l’intégration avec différents systèmes,   
la diffusion en continu et l’analyse, l’intégration de réseaux virtuels et le traitement par lots de colonnes à données volumineuses.

* Intégration dans l’entreprise - rassembler les divers systèmes et applications métier pour orchestrer le processus sans avoir à approvisionner une grande empreinte d’infrastructure
* Pratique de DevOps - mise en place de CI et CD pour les services développés afin de contribuer aux meilleures pratiques de contrôle de la source pour la gestion du code
* Mise à l’échelle et journalisation - comprendre l’aspect de la mise à l’échelle des services lorsque vous avez un pic d’événements, et visualisation des insights d’application pour recueillir des données métier

**Audience**

* Public cible :
  + Microsoft – CSE, CSA, GBB, ATT, SE, TPM
  + Client - Développeurs PHP
* Secteurs verticaux cibles : Intersectoriel
* Profil client : Exemples :
  + Développement d’application moderne - clients souhaitant créer des applications natives cloud et utiliser les services PaaS (Platform as a Service) sur Azure pour éliminer les frais généraux de gestion de l’infrastructure et réduire le délai de commercialisation des applications. L’accent est mis sur le développement rapide des applications.
  + Architecture distribuée pilotée par les événements - les clients recherchent une solution architecturée qui s’appuie sur des services de messagerie pour la communication entre les systèmes et les services et se concentre sur le traitement asynchrone pour une mise à l’échelle et une gestion efficaces.

**Questions d’inscription**

|  |  |  |
| --- | --- | --- |
| **Obligatoire** | **Champ** | **Options de réponse** |
| Oui | Quel est votre niveau de compréhension de l’utilisation d’Azure aujourd’hui ? | Aucune Une certaine compréhension  J’ai une expérience de travail de base sur Azure  Je compte sur Azure aujourd’hui pour le cloud |
| Oui | Combien de temps avez-vous passé avec Azure Functions, Event Grid et Logic Apps ? | [Liste déroulante ou boutons radio]  < 3 mois  3-6 mois  6 mois - 1 an  > 1 an |
| Oui | Quel est votre langage de programmation préféré ? | [Liste déroulante]  C#  Java  JavaScript  Python  Autre |

**Questions de recherche de coach**

Les questions suivantes peuvent être utilisées pour trouver des coachs ayant les connaissances appropriées. Consultez Sélection du coach dans le [playbook OpenHack](https://aka.ms/openhackplaybook) pour plus d’informations.

|  |  |  |
| --- | --- | --- |
| Question # | Question | Ce qu’il faut rechercher dans la réponse |
| 1 | Quelle est votre expérience antérieure avec OpenHack | * Vous devez avoir réussi tous les défis du OpenHack Serverless original (v1). * Vous devez avoir terminé les défis modifiés dans OpenHack Serverless 2.0. Il est fortement recommandé de terminer tous les défis. |

|  |  |  |
| --- | --- | --- |
| 2 | Veuillez décrire votre expérience avec Azure Functions, y compris votre expérience avec Durable Functions. | * Vous devez avoir une expérience préalable du développement et du déploiement de fonctions Azure à l’aide de C#, Java, Node.js et/ou Python. * Vous devez avoir une expérience de la configuration d’un pipeline CI/CD à l’aide d’Azure DevOps. * Nous avons besoin de coachs qui ont de l’expérience dans le développement d’Azure Functions en utilisant C#, Java, Node.js et Python. * Donnez la priorité aux répondants qui ont une expérience du développement avec Durable Functions. |
| 3 | Veuillez décrire votre expérience de l’utilisation de Logic Apps pour recevoir des événements Azure Event Grid et effectuer un traitement par lots basé sur les données d’événement. | * Il est fortement recommandé de savoir comment configurer Stockage Azure pour émettre des événements via Azure Event Grid et recevoir ces événements à l’aide d’un déclencheur Logic App. * Priorisez les répondants qui ont utilisé Logic Apps pour effectuer des traitements par lots. |
| 4 | Veuillez décrire votre expérience avec la gestion des API Azure en configurant l’utilisation d’Azure Functions et de Logic Apps, notamment en ce qui concerne la création de spécifications d’API ouvertes. | * Il est fortement recommandé que le coach ait de l’expérience dans la configuration de la gestion des API pour travailler avec Functions et/ou Logic Apps, y compris l’utilisation de la spécification d’API ouvertes (Swagger). |
| 5 | Veuillez décrire une mission dans laquelle vous avez utilisé Event Hubs, Kafka ou Event Grid. | * Il est recommandé d’avoir de l’expérience dans l’utilisation d’Event Hubs et d’Event Grid (en particulier les événements de stockage Azure). * Donnez la priorité aux répondants qui ont utilisé Event Hubs et Kafka, en particulier Event Hubs avec des points d’extrémité avec Kafka. |
| 6 | Veuillez décrire votre expérience de l’utilisation de Log Analytics pour surveiller Azure Functions. | * Vous devez avoir une expérience de l’utilisation de Log Analytics pour créer des requêtes (y compris des graphiques) et configurer des alertes basées sur des métriques liées à Azure Functions. |

**FAQ**

Q : Les clients qui utilisent FaaS doivent-ils quand même participer à cet OpenHack ?

R : Oui, car le contenu met en évidence le cycle d’application de bout en bout en mettant l’accent sur les exigences en matière de CI/CD, de mise à l’échelle, de récupération d’urgence et de réseau virtuel.

Q : Y a-t-il une liste de OpenHacks auxquels un participant devrait assister d’abord, avant de se rendre au vôtre ?

R : Non. Voir *Prérequis*, ci-dessus.