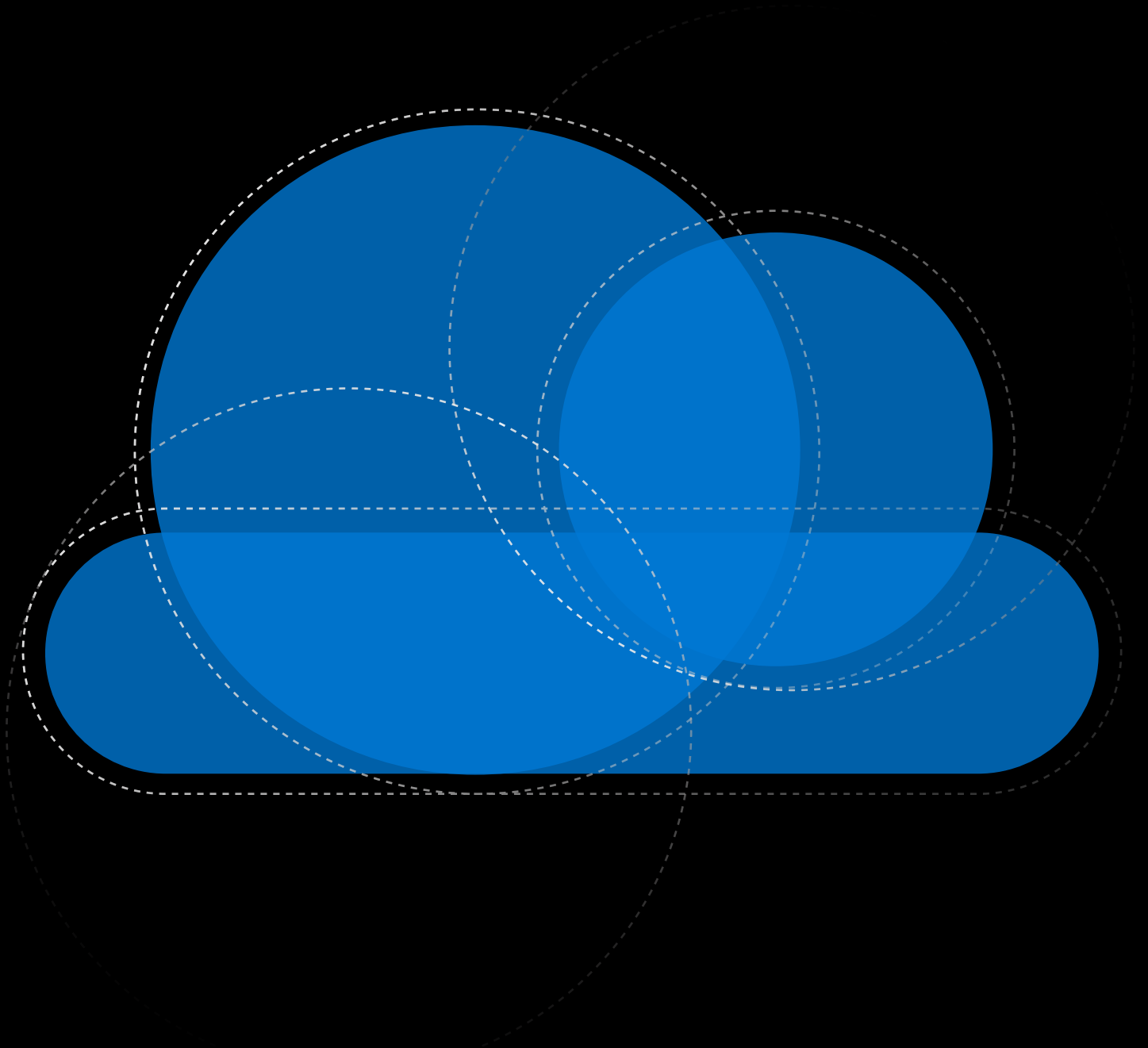


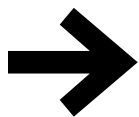
Série d'Ebooks



Guide d'Azure pour les développeurs

Publication de mai 2019





03 /

Introduction

Nous sommes là pour vous aider

05 /

Chapitre 1 : Prise en main d'Azure

Que peut vous apporter Azure ?
Où héberger votre application
Fonctions d'Azure App Service
Azure Functions
Azure Logic Apps
Azure Batch
Conteneurs
Qu'utiliser, et quand ?
Augmenter les performances de votre application
Accès principal Azure
Azure Content Delivery Network
Cache Redis Azure

22 /

Chapitre 2 : Connecter votre application et vos données

Que peut faire Azure pour vos données ?
Où stocker vos données
Azure Cosmos DB
Azure SQL Database
Bases de données Azure pour MySQL, PostgreSQL et MariaDB
Azure Storage
Solutions d'analyse de données Azure
Procédure guidée : publier un site Web local sur Azure avec une base de données SQL

40 /

Chapitre 3 : Sécurisation de votre application

Comment Azure peut-il contribuer à sécuriser votre application ?
Chiffrement
Azure Security Center
Journalisation et surveillance

51 /

Chapitre 4 : Ajout de fonctions intelligentes à votre application

Comment Azure peut-il intégrer l'IA à votre application ?
Azure Search
Cognitive Services
Azure Bot Service
Azure Machine Learning Studio
Outils de développement pour l'IA
L'IA et la réalité mixte
Utilisation des événements et des messages dans votre application

72 /

Chapitre 5 : Votre entreprise connectée avec IoT

Comment Azure peut-il connecter, sécuriser, gérer, surveiller et contrôler vos appareils dans le cloud ?
Azure IoT Hub
Azure IoT Central
Accélérateurs de solution Azure IoT
Azure IoT Edge
Azure Digital Twins
Azure Sphere
En savoir plus sur Azure IoT
Qu'utiliser, et quand ?

82 /

Chapitre 6 : Où et comment déployer vos services Azure

Comment Azure peut-il déployer vos services ?
Infrastructure en tant que code
Plans Azure
Conteneurs dans Azure
Azure Stack
Où et quand effectuer le déploiement ?

89 /

Chapitre 7 : Partager le code, suivre les travaux et livrer des logiciels

Comment Azure peut-il vous aider à améliorer votre planification et votre collaboration et accélérer la livraison de vos applications ?
Tableaux de bord Azure
Azure Repos
Pipelines Azure
Plans de test Azure
Azure Artifacts

98 /

Chapitre 8 : Azure en action

Procédure guidée : le portail Azure
Procédure guidée : développer une application Web et une base de données
Procédure guidée : étendre les applications
Procédure guidée : mise en production

119 /

Chapitre 9 : Résumé et ressources

Poursuivez votre apprentissage avec Azure
À propos des auteurs

Guide d'Azure pour les dévelop- peurs

Ce guide est destiné aux développeurs et aux architectes qui se lancent dans l'utilisation de Microsoft Azure. Nous vous exposerons dans ce guide les tenants et les aboutissants de Microsoft Azure. Vous y apprendrez comment démarrer et à quels services faire appel pour vos différents scénarios d'utilisation.

Qu'il s'agisse de la création de sites Web, de bases de données et d'applications mobiles et de bureau, ou de l'intégration des plus récentes technologies dans vos applications, Azure fait pour vous le plus gros du travail. Les services Azure sont étudiés pour fonctionner ensemble, ce qui vous permet d'élaborer des solutions complètes qui prolongent la durée de vie de votre application.

Nous sommes là pour vous aider

Nous pouvons vous aider de diverses manières selon vos besoins.

Avec nos [programmes de support](#), vous avez accès aux équipes de support technique d'Azure, à des instructions pour la conception cloud et à une aide à la planification de la migration. Vous pouvez même acquérir un programme de support qui garantit une réponse dans les 15 minutes de la part des équipes de support technique.

Vous pouvez aussi obtenir de l'aide par d'autres voies, telles que :

[La documentation et les guides](#) qui vous donnent une vue d'ensemble de tous les éléments d'Azure et expliquent chaque fonctionnalité de manière détaillée.

[Les contrats de niveau de service \(SLA\)](#), qui vous informent sur les garanties de disponibilité et les politiques d'avoir-service en cas de temps d'arrêt.

[@AzureSupport sur Twitter](#), qui est tenu par de compétents ingénieurs Azure qui répondent rapidement aux questions que vous leur posez par tweet.

[Stack Overflow](#), qui apporte des réponses aux questions concernant Azure et comprend de nombreux billets actifs rédigés par les membres des équipes techniques d'Azure.

[La communauté de support Azure](#), qui est un lieu de discussion pour la communauté Azure et qui regroupe les réponses aux questions de celle-ci.

[Azure Advisor](#), qui émet automatiquement des recommandations personnalisées pour vos ressources Azure, notamment pour améliorer votre sécurité, votre disponibilité et vos performances, et réduire vos coûts.

[Azure Service Health](#), qui vous offre une vue personnalisée de l'intégrité de vos services Azure.

01 /

Prise en main d'Azure

Vous avez décidé d'élaborer des applications sur Azure et maintenant, vous souhaitez vous lancer. Il ne vous faut pas grand-chose : il vous suffit d'ouvrir un [compte gratuit Azure](#). Ce compte est assorti de crédits vous permettant d'explorer les services payants d'Azure et plus de 25 services utilisables gratuitement à vie.

Choisissez simplement les outils, les applications et les infrastructures que vous souhaitez utiliser, puis commencez à exécuter vos applications sur Azure.

Que peut vous apporter Azure ?

Que vous soyez développeur professionnel ou amateur, le développement avec Azure vous fait bénéficier des toutes dernières technologies cloud et des outils de développement les meilleurs de leur catégorie. Ils vous rendent plus facile de créer des applications pour le cloud dans votre langage favori.

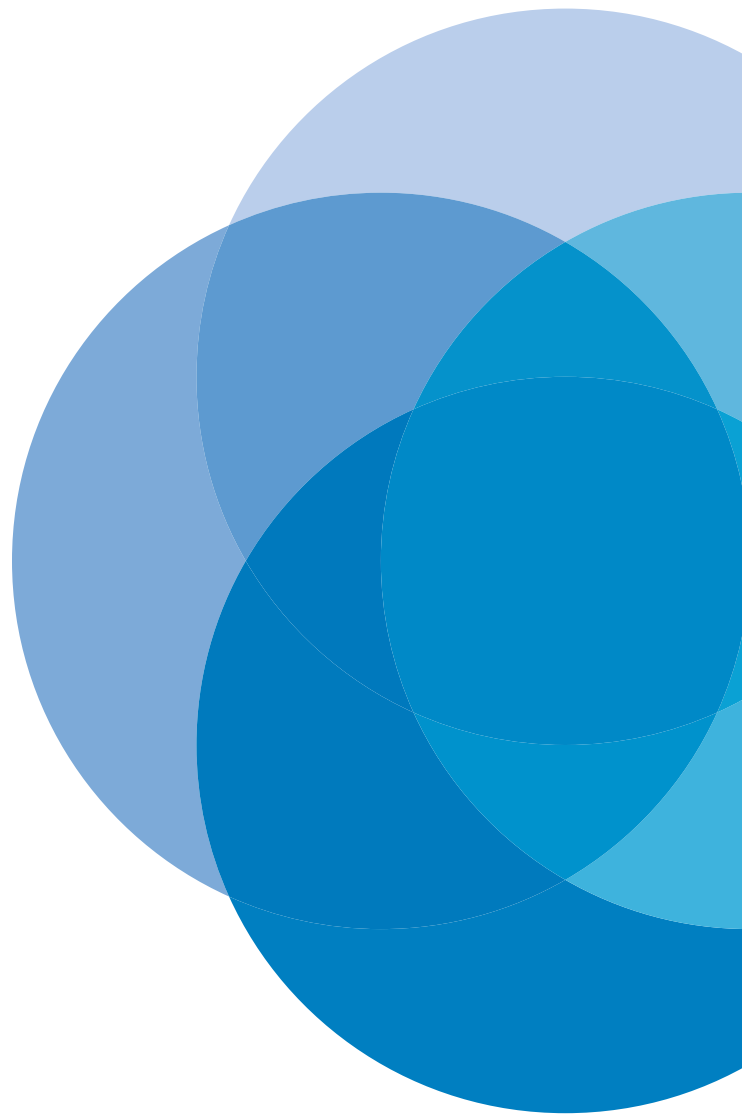
Avec Azure, vous obtenez des résultats plus rapides, faites progresser vos compétences et imaginez et créez les applications de demain.

Démultipliez votre impact grâce à :

- une plateforme cloud ;
- des outils de développement ;
- des services de gestion.

Tous ces éléments, étroitement intégrés ensemble, forment un véritable écosystème qui vous permet de créer de magnifiques applications et des expériences numériques fluides exécutables sur n'importe quel appareil.

Profitez des incroyables fonctionnalités d'Azure qui ne cessent de se développer. Ne perdons pas de temps et découvrons ce que vous pouvez faire.



Où héberger votre application

Azure offre des services conçus pour fournir tout le nécessaire à la livraison et à la mise à l'échelle de toutes les applications. En utilisant les services Azure pour exécuter votre application, vous bénéficiez d'évolutivité, de haute disponibilité, d'une plateforme entièrement gérée et de services de base de données. Azure offre également les options suivantes pour exécuter votre application.

Azure App Service

Vous pouvez héberger vos applications dans une plateforme d'application entièrement gérée qu'apprécient beaucoup les développeurs professionnels : [Azure App Service](#).

Azure App Service est une collection de services d'hébergement et d'orchestration, partageant des caractéristiques et fonctionnalités communes. Par exemple, tous les services Azure App Service peuvent sécuriser une application à l'aide d'[Azure Active Directory](#) et peuvent utiliser des domaines personnalisés.

Azure App Service est composé des éléments suivants :

Web Apps : l'un des services Azure les plus utilisés, [Web Apps](#) peut héberger vos applications Web ou vos API. Une application Web est pour l'essentiel une abstraction d'un serveur Web, comme Internet Information Services (IIS) ou Tomcat, utilisés pour héberger des applications HTTP. Web Apps peut héberger des applications écrites en .NET, Node.js, Python, Java ou GO, et vous pouvez utiliser les extensions disponibles pour exécuter encore plus de langages.

```
// À essayer : créer une  
application Web ASP.NET Core  
dans Azure
```

Mobile Apps : assurez le back-end de vos applications mobiles avec [Mobile Apps](#).

Lorsque vous hébergez une API dans Mobile Apps, vos applications mobiles s'y connectent via le SDK client multi-plateforme. Cette option est disponible pour Windows, iOS et Android. Mobile Apps fournit des fonctionnalités telles que la synchronisation hors connexion et les notifications Push, qui vous aident à créer une expérience mobile moderne, performante et sûre.

Fonctions d'Azure App Service

Azure App Service est l'un des principaux services d'Azure que vous pouvez utiliser pour héberger vos applications. Chacun de ces services offre des fonctionnalités uniques, mais ils partagent tous certaines caractéristiques communes :

Mise à l'échelle

Azure App Service fonctionne dans le cadre des [plans App Service](#), qui sont des abstractions de machines virtuelles (MV). Une ou plusieurs machines virtuelles exécutent vos services Azure App Service, mais comme c'est Azure qui les gère, vous n'avez pas besoin de savoir lesquelles. Vous pouvez toutefois adapter les ressources exécutant Azure App Service.

Vous pouvez choisir un niveau de facturation supérieur (de gratuit à premium) ou augmenter le nombre d'instances d'applications en cours d'exécution. Il est même possible de faire en sorte qu'Azure App Service ajuste automatiquement le nombre d'instances pour votre compte, sur la base d'une planification ou d'éléments de mesure tels que l'utilisation des processeurs et de la mémoire ou la longueur de la file HTTP.

Emplacements de déploiement

Après avoir déployé une nouvelle version de votre application sur un emplacement de déploiement, vous pouvez la tester pour voir si elle fonctionne comme prévu, puis la faire passer à un emplacement de production.

Vous pouvez également utiliser la fonction de [Test en production](#) d'Azure pour acheminer un certain pourcentage du trafic depuis votre application de production vers un emplacement de déploiement. Par exemple, si vous faites basculer 10 pour cent de vos utilisateurs vers la nouvelle version de votre application dans l'emplacement de déploiement, vous pouvez vérifier si les nouvelles fonctions marchent comme prévu et si les utilisateurs s'en servent.

Une fois satisfait des performances de la nouvelle version de votre application dans l'emplacement de déploiement, vous pouvez opérer un « échange » entre l'application qui est dans l'emplacement de déploiement et celle qui se trouve dans l'emplacement de production. Vous pouvez également basculer l'application d'un emplacement de développement vers un emplacement intermédiaire, puis vers l'emplacement de production. Avant cela, l'opération de basculement vérifie que la nouvelle version de votre site Web est opérationnelle et prête à être lancée. Une fois cette vérification effectuée, l'opération de basculement commute les emplacements, et vos utilisateurs peuvent alors voir la nouvelle version de l'application, sans aucun temps d'interruption. Vous pouvez également annuler le basculement et revenir au déploiement de la nouvelle version.

Vous pouvez utiliser les emplacements de déploiement dans un environnement de développement, de test ou de production. Vous ne pouvez pas utiliser les emplacements de déploiement comme des environnements, car ils résident tous dans le même plan App Service.

Ils doivent être maintenus séparés pour des raisons de sécurité, d'échelle, de facturation et de performances. Vous pouvez permuter les emplacements de déploiement manuellement, au moyen de l'interface de ligne de commande (CLI) d'Azure et de l'API de gestion d'Azure. Cela permet aux outils tels qu'Azure DevOps d'effectuer des opérations de basculement lors du lancement d'une version.

Déploiement continu

Pour publier votre application sur App Service, vous pouvez utiliser des services externes tels que Jenkins, Octopus Deploy ou d'autres. Vous pouvez également utiliser la [fonction Déploiement continu \(CD\) d'Azure DevOps](#) dans App Service. Cette fonction vous permet de créer un canal de développement-test-lancement au sein d'App Service.

Le processus effectue les opérations suivantes :

1. Il récupère le dernier code source dans le référentiel indiqué
2. Il génère le code selon un modèle que vous choisissez (ASP.NET, Node.js ou autres)
3. Il déploie l'application dans un environnement de test ou de préproduction et effectue un test de chargement
4. Il déploie l'application en production après approbation (vous pouvez indiquer si vous souhaitez utiliser un emplacement de déploiement)

Connexion aux ressources locales

Vous pouvez vous connecter à des ressources externes telles que les magasins de données dans App Services. Ces ressources ne sont pas nécessairement situées dans Azure ; elles peuvent être n'importe où, sur site ou dans votre propre datacenter. Vous pouvez vous connecter aux services sur site au moyen de nombreux mécanismes, selon vos besoins : [connexions hybrides Azure](#), [réseaux virtuels Azure](#) et [Azure ExpressRoute](#).

Domaines personnalisés et certificats Azure App Service

Lorsque vous lancez une application dans Azure App Service, une URL est exposée, par exemple : `https://myazurewebsite.azurewebsites.net`. Si vous souhaitez utiliser votre propre domaine personnalisé, vous pouvez le faire en mappant le nom de ce domaine sur App Services. [Voici les étapes à suivre](#).

Vous pouvez également vous assurer que votre application est exécutée sur HTTPS en utilisant un certificat Secure Sockets Layer (SSL). Fournissez votre propre certificat ou en achetez-en un [directement depuis le portail Azure](#). Lorsque vous achetez un certificat SSL à partir du portail Azure, vous achetez un certificat Azure App Service. Vous pouvez le configurer pour l'utiliser avec les liaisons de votre domaine personnalisé.

```
// À essayer : acheter et  
configurer un certificat SSL  
dans cette procédure guidée
```

App Service Environment

Dans une application Web multi-niveaux, votre application dans Web Apps utilise souvent une base de données ou des services. Idéalement, vous voulez que ces services soient exposés uniquement à l'application et non à Internet. Toutefois, comme l'application représente le point d'entrée pour vos utilisateurs, elle est souvent ouverte sur Internet.

Pour isoler ces services de support d'Internet, vous pouvez utiliser un réseau virtuel Azure.

Ce service inclut vos services de support et les relie à votre application dans Web Apps, de manière à ce que ces services de support soient exposés uniquement à l'application, et non à Internet.

[Cet article](#) décrit ce service plus en détails et vous montre comment l'utiliser.

Vous souhaitez peut-être encore plus de contrôle. Vous voulez sans doute que votre application soit incluse dans un réseau virtuel, afin de pouvoir contrôler son accès.

Ou vous souhaitez que votre application soit appelée par une autre application dans Web Apps et que celle-ci fasse partie de votre back-end.

Dans ce cas, vous pouvez utiliser un environnement [Azure App Service Environment](#). Cet environnement vous offre une très grande échelle et vous donne le contrôle sur l'isolation et l'accès au réseau.

Débogueur d'instantané pour .NET

Le débogage des applications peut être une opération difficile, en particulier si l'application est exécutée en production. Avec le débogueur d'instantané, vous pouvez prendre une « photo » de votre application en production au moment où le code qui vous intéresse est exécuté.

Le débogueur vous permet de voir exactement le problème sans impacter aucunement le trafic de votre application en production. Le débogueur d'instantané vous permet de réduire significativement les délais nécessaires à la correction des problèmes qui apparaissent dans les environnements de production. En outre, vous pouvez utiliser Visual Studio pour définir des points d'instantané afin de réaliser le débogage pas à pas.

Application automatique de correctifs au SE et à .NET Framework

Comme vous utilisez une plateforme entièrement gérée, vous n'avez aucun besoin de gérer votre propre infrastructure et bénéficiez de l'application automatique de correctifs au système d'exploitation (SE) et à l'infrastructure.

Machines virtuelles

Le fait d'héberger votre application dans une machine virtuelle des [Machines virtuelles Azure](#) vous permet de bénéficier d'un large contrôle sur la méthode d'hébergement de votre application. Vous êtes toutefois responsable de la maintenance de l'environnement, notamment de l'application des correctifs du système d'exploitation et de la tenue à jour des programmes antivirus.

Vous pouvez utiliser une machine virtuelle pour tester la dernière version préliminaire de Visual Studio, sans « polluer » votre machine de développement.

Azure Functions

Avec Azure Functions, vous pouvez écrire le code dont vous avez besoin pour une solution, sans vous soucier de la création d'une application complète ou de l'infrastructure permettant de l'exécuter. Une fonction est une unité de logique de code qui est déclenchée par une requête HTTP, par un événement dans un autre service Azure ou selon une planification.

Des liaisons d'entrée et de sortie relient le code de votre fonction à d'autres services, comme Azure Storage, Azure Cosmos DB et Azure Service Bus, et même à des services tiers tels que Twilio et SendGrid. Avec Functions, vous pouvez créer rapidement de petites fonctionnalités et les héberger dans un environnement flexible qui gère automatiquement la mise à l'échelle.

Azure Functions vous permet de choisir de payer uniquement les fonctions qui s'exécutent, au lieu de devoir faire fonctionner les instances de calcul durant tout le mois. Cette solution est également appelée sans serveur, car il vous suffit de créer votre application sans vous préoccuper des serveurs ni même de la mise à l'échelle de ceux-ci.

Vous pouvez écrire des fonctions Azure Functions en .NET, JavaScript, Java et dans un nombre croissant de langages.

Une application utilisant Functions active une fonction chaque fois qu'un nouveau fichier image est chargé vers le stockage blob Azure. La fonction redimensionne ensuite l'image et l'écrit dans un autre compte de stockage blob.

Les données du blob qui ont déclenché la fonction sont transmises à la fonction en tant que paramètre myBlob, qui comprend l'URL Blob. Utilisez le paramètre de liaison de sortie outputBlob pour spécifier l'objet blob dans lequel écrire le résultat. Vous n'avez pas besoin d'écrire la structure de connexion au stockage blob, il vous suffit de la configurer.

```
// À essayer : créer votre  
première fonction Azure en  
utilisant le portail Azure
```

Azure Logic Apps

Vous pouvez orchestrer une logique métier avec [Logic Apps](#) en automatisant un processus métier ou en l'intégrant à des applications de logiciel en tant que service (SaaS). Comme Azure Functions, le service Logic Apps peut être activé par une source extérieure, par exemple un nouveau message. En reliant les appels d'API aux connecteurs, vous pouvez créer un workflow (éventuellement complexe) pouvant impliquer des ressources provenant aussi bien du cloud que du site.

Logic Apps [compte de nombreux connecteurs aux API disponibles](#). Ils permettent d'assurer la connexion à Azure SQL Database, Salesforce, SAP, entre autres.

Vous pouvez également exposer vos propres API ou Azure Functions en tant que connecteurs à utiliser dans Logic Apps, ce qui vous permet d'exécuter facilement des actions sur des systèmes externes dans votre workflow ou de faire en sorte que l'un de ces systèmes active votre application logique.

Tout comme Azure Functions, Logic Apps fonctionne sans serveur, effectue une mise à l'échelle automatique et demande uniquement un paiement sur la base de l'exécution.

Voici un exemple de workflow dans Logic Apps :

1. L'application logique est activée lorsqu'un e-mail contenant un ordre d'expédition arrive dans Office 365.
2. En utilisant les données de cet e-mail, l'application logique vérifie la disponibilité de l'article commandé dans SQL Server.
3. En utilisant Twilio, l'application logique envoie un message texte vers le téléphone du client, indiquant que la commande a été reçue et que l'article a été expédié.

```
// À essayer : découvrir  
Azure Logic Apps
```

Azure Batch

Si vous devez exécuter des applications de calcul haute performance (HPC, High Performance Computing) ou de traitement par lots à grande échelle sur des machines virtuelles, utilisez [Azure Batch](#). Azure Batch crée et gère une collection pouvant rassembler des milliers de machines virtuelles, installe les applications que vous souhaitez exécuter et planifie les travaux sur les machines virtuelles. Vous n'avez pas à déployer ou à gérer individuellement les machines virtuelles ou les clusters de serveurs. Batch planifie, gère et dimensionne automatiquement vos tâches, de sorte que vous n'utilisiez que les machines virtuelles dont vous avez besoin.

Batch est un service gratuit. Vous ne payez que les ressources sous-jacentes consommées, telles que les machines virtuelles, le stockage et la mise en réseau.

Batch est adapté à l'exécution de charges de travail parallèles à l'échelle, telles que des modèles de risque financier, de transcodage multimédia, de rendu d'images VFX et 3D, de simulations d'ingénierie et de nombreuses autres applications à forte intensité de calcul. Utilisez Batch pour dimensionner une application ou un script que vous avez déjà exécuté sur des stations de travail ou un cluster sur site, ou développer des solutions SaaS qui utilisent Batch comme plateforme de calcul.

```
// À essayer : découvrir  
Azure Batch grâce à ces  
tutoriels pas à pas
```

Conteneurs

Les conteneurs sont similaires à des machines virtuelles, en beaucoup plus légers. Vous pouvez les démarrer et les arrêter en quelques secondes. Les conteneurs offrent également de formidables qualités de portabilité, ce qui en fait un atout idéal pour développer localement une application, sur votre machine, puis l'héberger dans le cloud, en test, puis ultérieurement en production.

Vous pouvez même exécuter des conteneurs sur place ou dans d'autres clouds. L'environnement que vous utilisez sur votre machine de développement suit votre conteneur, afin que votre application s'exécute toujours dans le même écosystème.

Mise à l'échelle et orchestration de conteneurs avec Azure Kubernetes Service

[Azure Kubernetes Service](#) (AKS) facilite la création, la configuration et la gestion d'un cluster de machines virtuelles préconfigurées pour exécuter des conteneurs. Cela signifie que vous pouvez utiliser vos compétences existantes pour gérer et déployer des applications qui s'exécutent dans des conteneurs sur Azure.

AKS réduit la complexité et la charge opérationnelle de la gestion d'un cluster Kubernetes en déchargeant une grande partie de cette responsabilité sur Azure. En tant que service Kubernetes hébergé, Azure traite pour votre compte les tâches critiques telles que la surveillance de l'intégrité et la maintenance. De plus, vous ne payez que pour les nœuds agents au sein de vos clusters, pas pour les nœuds maîtres. En tant que service Kubernetes géré, AKS fournit automatiquement les mises à niveau de version

et les correctifs de Kubernetes, la mise à l'échelle simple des clusters, un plan de contrôle hébergé à réparation spontanée (nœuds maîtres) et est également source d'économies parce que vous ne payez que l'exécution des nœuds de pool d'agents.

Azure étant chargé de gérer les nœuds de votre cluster AKS, beaucoup de tâches ne doivent plus être effectuées manuellement, telles que la mise à niveau des clusters. Comme Azure gère ces tâches de maintenance critiques pour votre compte, AKS ne fournit pas d'accès direct au cluster (comme avec SSH).

// Plus d'informations : Découvrir comment utiliser Azure Kubernetes Service

Héberger des conteneurs avec Azure Instances de conteneur

Vous pouvez héberger votre conteneur à l'aide des [instances de conteneur Azure](#) (ACI, Azure Container Instances). ACI assure un calcul rapide et isolé pour répondre au trafic qui arrive par pics, sans nécessiter de gérer des serveurs. Par exemple, Azure Container Service (ACS) peut utiliser le Kubelet virtuel pour provisionner des pods (conteneurs) à l'intérieur d'ACI, qui démarrent en quelques secondes. Cela permet à ACS de fonctionner avec juste assez de capacité pour une charge de travail moyenne. À mesure que vous épuisez votre capacité dans votre cluster ACS, vous pouvez ajouter d'autres pods ACI sans avoir de serveurs supplémentaires à gérer. Le service ACI est facturé par seconde, processeur virtuel, gigaoctet ou mémoire.

// Plus d'informations : En savoir plus sur les instances de conteneur Azure

Héberger des conteneurs dans Azure App Service Web App pour conteneurs

[Web App pour conteneurs](#) vous permet de déployer et d'exécuter facilement des applications Web en conteneur à l'échelle souhaitée. Il suffit d'obtenir des images de Docker Hub ou d'un registre de conteneur privé Azure et Web App pour conteneurs déploie l'application conteneurisée avec vos dépendances préférées en production en quelques secondes. La plateforme traite automatiquement les correctifs du système d'exploitation, le dimensionnement de la capacité et l'équilibrage de charge. Vous pouvez exécuter des conteneurs Docker (Linux) et Windows dans Web App pour conteneurs.

Azure Container Registry

Une fois que vous avez créé une image de conteneur pour y exécuter votre application, vous pouvez stocker ce conteneur dans [Azure Container Registry](#) (ACR). C'est un service de stockage hautement disponible et sécurisé, tout particulièrement conçu pour stocker des images de conteneurs. Il est parfaitement adapté au stockage de vos images Docker privées.

Vous pouvez aussi utiliser ACR pour vos pipelines de développement et de déploiement de conteneurs existants. Utilisez ACR Build pour élaborer des images de conteneurs dans Azure. Au choix, vous pouvez les créer à la demande ou les automatiser entièrement à l'aide de déclencheurs de mise à jour de l'image de base et de validation du code source.

Azure Service Fabric

Une autre façon d'exécuter vos applications consiste à les exécuter sur [Azure Service Fabric](#). C'est en fait le service qui exécute de nombreux services Azure internes à Microsoft, comme Azure SQL Database et Azure App Service. Exécutez vos applications dans Azure Service Fabric pour bénéficier d'une haute disponibilité, lancer des exécutions à grande échelle et effectuer des propagations de mise à niveau.

Vous pouvez utiliser Azure Service Fabric pour exécuter des applications .NET basées sur des microservices. Ces derniers sont des solutions constituées de nombreux petits services qui communiquent entre eux et sont employés par les interfaces utilisateur et d'autres composants. Service Fabric est parfaitement adapté à de telles solutions parce qu'il orchestre le concert des composants d'application et les exécute à un haut niveau de disponibilité et de performance.

Le point fort d'Azure Service Fabric est que vous pouvez l'exécuter n'importe où. Installez Service Fabric sur votre ordinateur de développement local, sur site, ou dans n'importe quel cloud (y compris Azure). Vous pouvez également utiliser Azure Service Fabric Mesh pour exécuter des conteneurs sur un cluster Service Fabric que Microsoft gère pour vous en tant que service. Cela ouvre tout un champ de possibilités.

Il est aussi très simple de déployer des applications sur Azure Service Fabric et les gérer avec vos outils préférés, tels que Visual Studio et les services Azure DevOps. En outre, Service Fabric est récemment devenu open source.

Qu'utiliser, et quand ?

Certains des services qui exécutent votre application dans Azure peuvent fonctionner ensemble en une solution, tandis que d'autres sont plus adaptés à d'autres objectifs.

Comme il peut être difficile de choisir les services les plus adaptés, le Tableau 1-1 vous aide à identifier les services Azure qui conviennent à votre scénario.

Tableau 1-1

	Web Apps*	Web App pour conteneurs*	Mobile Apps*	Functions*	Logic Apps*	Machines virtuelles*	Kubernetes Service*	Service Fabric*	Instances de conteneurs*	IoT*
Applications monolithiques et N-niveaux		●				●**			●	●
Back-end d'application mobile			●			●**				
Applications à architecture de microservices				●			●	●		
Workflows et orchestration de processus d'entreprise				●	●					
Tâches de calcul intensif										●
Exécution de l'application n'importe où (y compris localement)		●					●	●	●	

* Les services marqués d'un astérisque comportent un niveau gratuit utilisable pour démarrer sans frais.

** Pour la réplication des applications existantes vers Azure.

Augmenter les performances de votre application

Lorsque votre application est opérationnelle et exécutée dans Azure, vous souhaitez qu'elle soit la plus performante possible. Azure propose une gamme de services qui peuvent vous aider.

Azure Traffic Manager

La plupart des applications modernes comptent des utilisateurs partout dans le monde. Offrir à chacun une expérience performante est particulièrement difficile. Le problème principal auquel vous devez faire face est le temps de latence, c'est-à-dire le délai nécessaire pour qu'un signal ou une requête soit acheminé à un utilisateur. Plus les utilisateurs sont éloignés de votre application, plus le temps de latence est long.

[Azure Traffic Manager](#) met les capacités à l'échelle dans toutes les régions et contribue à réduire la latence et à fournir aux utilisateurs une expérience performante, quelle que soit leur localisation. Traffic Manager est un mécanisme de routage intelligent à placer au-devant de vos applications Web Apps. Web Apps sert de points de terminaison, qu'Azure Traffic Manager surveille pour assurer l'intégrité et la performance de votre application.

Lorsque des utilisateurs accèdent à votre application, Traffic Manager les dirige vers l'application Web Apps à proximité la plus performante.

Inclure Traffic Manager dans votre architecture est un excellent moyen d'améliorer les performances de votre application.

Accès principal Azure

Vos utilisateurs sont peut-être disséminés dans le monde entier, et voyagent occasionnellement. Cela peut rendre plus difficile de leur garantir une expérience performante, ainsi que la disponibilité et la sécurité de votre application, où qu'ils se trouvent.

[Azure Front Door](#) peut vous y aider. Ce service achemine le trafic des utilisateurs vers le point de terminaison de l'application le plus performant pour eux afin d'obtenir les meilleurs résultats. Azure Front Door exploite les points de terminaison disponibles tout en évitant ceux qui sont en panne.

Azure Traffic Manager fait la même chose, mais d'une manière différente de celle d'Azure Front Door. Azure Front Door fonctionne sur la [couche OSI 7](#) ou sur la couche HTTP/HTTPS, alors qu'Azure Traffic Manager fonctionne avec des DNS. En d'autres termes, Azure Front Door fonctionne dans la couche application alors qu'Azure Traffic Manager fonctionne au niveau du réseau. C'est une différence fondamentale qui détermine les capacités des services.

À cause de cette différence, Azure Front Door peut faire bien plus que diriger les utilisateurs vers les points de terminaison disponibles et performants.

Azure Front Door vous permet de créer des règles personnalisées de pare-feu d'application Web (WAF) afin de réguler les accès et protéger vos charges de travail HTTP/HTTPS de l'exploitation abusive reposant sur les adresses IP client, les codes pays et les paramètres HTTP.

En outre, Azure Front Door vous permet d'établir des règles de restriction pour contrecarrer le trafic des robots malveillants. Ces fonctionnalités ne sont qu'un échantillon des possibilités exclusives d'Azure Front Door.

Les autres fonctionnalités d'Azure Front Door sont notamment :

- **Le routage par URL**
Il vous permet d'acheminer les requêtes de différentes URL vers différents pools back-end (des applications qui reçoivent le trafic, comme Web Apps). Par exemple, `http://www.contoso.com/users/*` va dans un pool et `http://www.contoso.com/products/*` dans un autre.
- **La réécriture d'URL**
Elle vous permet de personnaliser l'URL transmise au pool back-end.
- **La terminaison SSL**
Elle vous permet de sécuriser le trafic d'un bout à l'autre, du navigateur jusqu'à l'application dans le pool back-end.
- **L'affinité de session**
Lorsque vous souhaitez que vos utilisateurs soient acheminés vers le même point de terminaison à chaque fois, l'affinité de session s'avère très utile. Cela a de l'importance lorsque l'état d'une session utilisateur est enregistré localement en back-end.

Si vous êtes confronté à un choix entre Azure Front Door et Traffic Manager, reportez-vous aux instructions suivantes :

	Azure Traffic Manager	Accès principal Azure
Vous n'avez besoin que de routage (fondé sur les performances ou la géographie) et de haute disponibilité	●	
Vous avez besoin d'une terminaison SLL (également appelée déchargement SSL)		●
Vous avez besoin de fonctions de couche d'application, telles que la réécriture d'URL et les règles WAF		●

Azure Content Delivery Network

L'un des services d'Azure pouvant vous aider à rendre votre application plus rapide est [Azure Content Delivery Network](#). Chargez vos fichiers statiques (vidéos, images, JavaScript, CSS et même des fichiers HTML statiques) vers un magasin de données tel qu'Azure Blob Storage, puis associez Azure Content Delivery Network à ce magasin.

Content Delivery Network reproduit alors ces fichiers statiques dans des centaines de points de présence dans le monde entier. Il vous suffit alors de modifier la référence aux fichiers statiques dans votre application avec une URL différente.

Par exemple, si la référence était auparavant `~/images/image.png`, elle doit être désormais `https://example.azureedge.com/image.png`.

Non seulement cette opération est très facile, mais elle améliore les performances de votre application des manières suivantes :

- Elle déporte le contenu utile de votre application. Comme ce contenu est maintenant transmis par Content Delivery Network, des cycles de traitement sont désormais libérés pour votre application
- Elle fournit du contenu statique plus près physiquement de vos utilisateurs en le distribuant à des points de présence partout dans le monde entier.

Vous pouvez utiliser Content Delivery Network pour les applications Web, mais aussi pour les applications mobiles et de bureau. Content Delivery Network peut, par exemple, servir à fournir des vidéos pour une application mobile. Les vidéos pouvant être volumineuses, vous ne voulez pas les stocker sur un appareil mobile (et vos utilisateurs non plus). Grâce à Content Delivery Network ces vidéos seront servies à partir de points de présence. Comme ceux-ci sont plus proches des utilisateurs, les performances s'en trouvent aussi améliorées.

// À essayer : Découvrir Azure
Content Delivery Network

Cache Redis Azure

Toutes les applications modernes utilisent des données. Lorsque vous récupérez des données d'un magasin de données, comme une base de données, cela implique généralement une analyse des différentes tables ou documents dans un serveur distant, puis l'association des résultats et l'envoi de ces résultats à l'appareil demandeur. Évidemment, ces opérations prennent du temps et peuvent frustrer et ennuyer vos utilisateurs.

Pour supprimer certains de ces « allers-retours », vous pouvez mettre en cache les données qui ne changent pas souvent. Ainsi, au lieu d'interroger la base de données à chaque fois, vous pouvez récupérer certaines données dans un cache, comme [Azure Redis Cache](#). L'avantage du cache est qu'il stocke les données dans un format simple, sous forme de paire clé-valeur. Vous n'avez pas besoin d'exécuter une requête complexe pour obtenir ces données, il vous suffit de connaître la clé pour récupérer la valeur.

Cette solution peut améliorer de façon spectaculaire les performances de votre application.

Voici comment fonctionne ce workflow :

1. L'application a besoin de certaines données et tente de les récupérer dans le cache.
2. Si les données n'y sont pas, l'application les récupère dans la base de données et les stocke dans le cache.
3. La prochaine fois que l'application cherchera ces données, elle les retrouvera dans le cache, ce qui économise un aller-retour vers la base de données.

Azure propose une solution de cache en tant que service avec Redis Cache. Celui-ci est basé sur le projet open source Redis et est maintenant pris en charge par les contrats de niveau de service des principaux acteurs du secteur. Il est très performant et comprend des options avancées, comme le clustering et la géo-réplication.

```
// À essayer : découvrir  
Azure Redis Cache.
```

Autres lectures

Si vous souhaitez davantage d'informations sur Azure Kubernetes Service, les instances de conteneur Azure Container, ou d'autres services Azure pour créer des applications distribuées, téléchargez et lisez ces Ebooks gratuits :

```
// Mettre vos applications en  
conteneurs avec Docker et  
Kubernetes
```

```
// Conception de systèmes  
distribués
```

02 /

Connecter votre application et vos données

Que peut faire Azure pour vos données ?

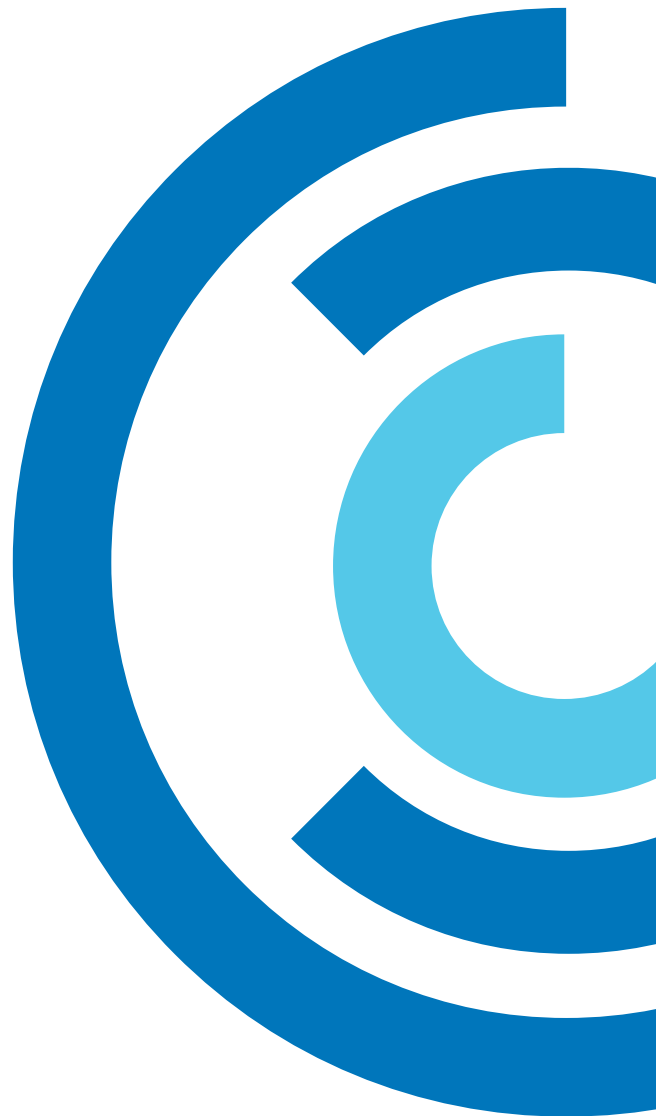
Azure vous permet d'exploiter le plein potentiel de vos données, où qu'elles se trouvent et quelle que soit leur forme. Soutenez un rythme de croissance élevé et gagnez du temps pour mieux innover au moyen d'un portefeuille de services de base de données de classe entreprise et hautement sécurisés capables de prendre en charge les moteurs de base de données open source.

Les services de base de données Azure sont entièrement gérés, ce qui libère pour vous un temps précieux à consacrer à la recherche de nouvelles façons de satisfaire votre clientèle et de nouvelles opportunités, au lieu de gérer votre base de données. Leurs performances de classe entreprise et leur haute disponibilité intégrée vous permettent d'évoluer rapidement et de vous doter d'une portée mondiale sans vous soucier des temps d'arrêt si onéreux.

Les développeurs peuvent tirer parti de leurs innovations à la pointe du secteur, telles que la sécurité intégrée avec surveillance et détection des menaces automatiques, ajustement automatique des paramètres pour optimiser les performances et diffusion à échelle mondiale clé en main. Par-dessus tout, vos investissements sont protégés grâce à la garantie financière qui accompagne les SLA.

Quelles que soient vos créations, nous pouvons vous aider à les lancer rapidement sur le marché, à leur offrir une vaste diffusion et à les gérer de manière simple et confidentielle.

Penchons-nous sur tout cela.



Où stocker vos données

Azure propose différents types de magasins de données qui peuvent vous aider à maintenir et à récupérer vos données dans n'importe quelle situation. Le tableau 2-1 présente les options de stockage disponibles dans Azure.

Tous les services incluent un niveau gratuit que vous pouvez utiliser pour commencer.

// Remarque : vous pouvez utiliser presque toutes les options de stockage mentionnées dans cette section en tant qu'activateurs et liaisons pour Azure Functions.

Examinons plus en détails chaque option de stockage.

Tableau 2-1

	SQL Database*	MySQL*	PostgreSQL*	MariaDB*	Azure Cosmos DB*	Blob*	Table*	File d'attente*	Fichier*	Disque*	Data Lake Store*	SQL Data Warehouse*
Données relationnelles	●	●	●	●							●	●
Données non structurées					●	●					●	
Données semi-structurées					●		●					
Messages de file d'attente								●				
Fichiers sur disque									●			
Fichiers haute performance sur disque										●		
Stockage de données volumineuses					●	●			●	●	●	●
Stockage de données de petite taille	●	●	●	●	●	●	●	●	●	●		
Géo-réplication de données	●	●	●	●	●	●	●	●	●	●		
Cohérence des données réglable					●							

* Les services marqués d'un astérisque comportent un niveau gratuit utilisable pour démarrer sans frais.

Azure Cosmos DB

Azure Cosmos DB est un nouveau type de base de données réellement pensé pour le Cloud. Ses principales fonctionnalités sont notamment :

- Un SLA garantissant 99,99 % de disponibilité (99,999 % en lecture), avec des temps de latence faibles (inférieurs à 10 ms pour les lectures et à 15 ms pour les écritures).
- La géo-réplication, qui [réplique les données dans d'autres régions géographiques](#) en temps réel.
- [Des niveaux de cohérence des données réglables](#) qui vous permettent de mettre en place un système de données véritablement distribué à l'échelle mondiale. Vous avez le choix parmi un large éventail de modèles de cohérence des données, notamment la cohérence forte, la cohérence de session et la cohérence finale.
- Traffic Manager, qui dirige les utilisateurs vers le point de terminaison du service le plus proche.
- Une évolutivité illimitée à l'échelle mondiale ; vous payez uniquement pour le débit et le stockage dont vous avez besoin.
- [L'indexation automatique des données](#), qui élimine le besoin de maintenance ou de réglage de la base de données.

Outre ces fonctions, Azure Cosmos DB offre différentes API permettant de stocker et de récupérer les données (SQL, JavaScript, Gremlin, MongoDB, stockage Table Azure et Apache Cassandra). Chaque API gère les données de manière différente. Vous pouvez vous servir de documents, de tables non structurées, de graphiques ou d'objets blob en guise de données. Choisissez l'API qui répond à vos besoins, et Azure Cosmos DB se charge du reste.

Vous bénéficiez des performances, de l'évolutivité et de la fiabilité du cloud, tout en continuant à utiliser le modèle de programmation auquel vous êtes habitué.

```
// À essayer : découvrir Azure  
Cosmos DB
```

Azure SQL Database

Si vous souhaitez utiliser des tables avec des colonnes et des lignes pour stocker des données, [Azure SQL Database](#) est l'option idéale. Système de base de données relationnelle semblable au système Microsoft SQL Server sur site, SQL Database s'exécute dans le cloud ; il est donc entièrement géré, performant, évolutif, sauvegardé automatiquement et il comprend de nombreuses fonctions avancées.

Avec SQL Database, vous pouvez faire [presque tout](#) e que vous pouvez faire avec SQL Server sur site. En effet, les nouvelles fonctions de SQL Server sont d'abord incorporées dans Azure SQL Database, puis dans SQL Server sur site.

Vous pouvez utiliser SQL Database avec vos outils préférés, comme SQL Server Management Studio et Entity Framework. Extrêmement fiables et robustes, les bases de données SQL Database sont couvertes par un SLA qui garantit une disponibilité de 99,99 %.

Voici certaines des fonctionnalités les plus avancées de SQL Database :

- [La géo-réplication](#), qui réplique les données dans d'autres régions géographiques en temps réel
- [Le masquage des données dynamiques](#), qui masque dynamiquement les données sensibles pour certains utilisateurs lors de l'exécution

- [L'audit](#), qui fournit une piste de vérification complète de toutes les actions liées aux données
- [Le réglage automatique des bases de données](#), qui surveille les performances de votre base de données et l'optimise automatiquement

SQL Database offre plusieurs niveaux de service orientés vers différents scénarios.

- **Usage général/Standard** : ce niveau offre des options de calcul et de stockage axées sur les économies, équilibrées et évolutives. Entièrement géré, avec des performances comparables aux machines virtuelles Azure SQL, ce niveau convient parfaitement à la plupart des charges de travail d'entreprise.
- **Critique pour l'entreprise/Premium** : ce niveau offre une résilience maximale aux pannes grâce à plusieurs réplicas isolés. Avec un taux d'E-S élevé en permanence, il comprend un groupe de disponibilité intégré aux fins de haute disponibilité. C'est la meilleure option pour les applications métier OLTP (Online Transactional Processing) vitales (opérations CRUD normales) ayant des exigences de débit d'E-S élevés.
- **Hyperscale** : ce niveau offre la prise en charge de bases de données très volumineuses (VLDB) sans aucun problème. Avec une architecture de stockage hautement évolutive et conçue spécifiquement pour le cloud, un cache multicouche optimisé pour les charges de travail intensives et volumineuses, il assure un haut débit et une latence faible, indépendamment du dimensionnement des opérations sur les données. Il s'agit du niveau le plus adapté aux charges de travail très intensives présentant de fortes exigences en matière de lecture et d'évolutivité du stockage.

Bases de données Azure pour MySQL, PostgreSQL et MariaDB

Azure fournit des bases de données [MySQL](#), [PostgreSQL](#) et [MariaDB](#) en tant que bases de données gérées, ce qui signifie qu'il vous suffit de les créer sans vous soucier de l'infrastructure sous-jacente. Tout comme les bases de données Azure SQL Database et Azure Cosmos DB, ces bases de données sont universellement disponibles, évolutives, hautement sécurisées et entièrement gérées.

Chacune de ces bases de données est adaptée à des scénarios d'utilisation légèrement différents mais, de manière générale, leur fonctionnalité est très semblable. On recommande d'utiliser des bases de données Azure pour MySQL, PostgreSQL et MariaDB lorsque l'on utilise déjà l'une de leurs versions locales et que l'on souhaite bénéficier de leur exécution entièrement gérée dans le cloud.

Azure Storage

[Azure Storage](#) est l'un des services Azure les plus anciens, fiables et performants. Azure Storage propose cinq types de stockage qui partagent tous les caractéristiques suivantes :

- La géo-redondance, qui réplique les données dans différents datacenters en vue de leur récupération en cas de panne d'un datacenter suite à un sinistre
- Chiffrement des données lors de l'exécution
- Domaines personnalisés

Azure Storage offre notamment les cinq types de stockage suivants : Objet blob, Table, File d'attente, Fichier et Disque (Figure 2-1).



Figure 2-1

Hébergement de sites Web statiques sur Azure Storage

Une autre fonctionnalité très utile d'Azure Storage est [l'hébergement de site Web statique](#). Cette fonction dédiée aux sites Web statiques n'utilise que du stockage Blob comme magasin de données, et vous pouvez l'utiliser pour héberger un site Web statique dans Azure Storage. Il vous suffit de charger les fichiers de votre site Web statique vers le stockage blob et d'indiquer quel fichier est le document par défaut (par exemple, index.html) et lequel est le document d'erreurs (par exemple, 404.html). L'exécution de votre site Web sera rapide à moindre frais. En réalité, vous ne payez que le stockage que vous consommez ; la fonction de site Web statique ne coûte rien de plus. En outre, lorsque vous utilisez la géo-redondance, qui est activée par défaut, votre site Web reste à l'état opérationnel même en cas de panne de votre datacenter principal.

Stockage blob

[Azure Blob Storage](#) permet de stocker de grands volumes de données non structurées. Il peut s'agir de données vidéo, image, audio ou texte, ou même de fichiers de disque dur virtuel pour machine virtuelle.

Il existe trois types d'objets blob : [les blobs Page](#), [Ajout et Bloc](#). Les objets blob Page sont optimisés pour les opérations de lecture et d'écriture aléatoires. Ils conviennent parfaitement au stockage d'un disque dur virtuel. Les objets blob Bloc sont optimisés pour un chargement efficace de grands volumes de données.

Ils sont parfaits pour stocker des fichiers vidéo volumineux qui ne changent pas souvent. Les objets blob Ajout sont optimisés pour les opérations d'ajout, telles que le stockage des journaux d'opérations qui ne peuvent être supprimés, ni mis à jour.

```
// À essayer : découvrir le stockage  
blob Azure
```

Stockage en table

[Le stockage de table Azure \(Table Storage\)](#) est un magasin de paires clé-valeur NoSQL extrêmement rapide et peu coûteux que vous pouvez utiliser pour stocker des données dans des tables flexibles. Une table peut contenir une ligne qui décrit une commande et une autre ligne qui décrit les informations client. Il n'est pas nécessaire de définir de schéma de données, ce qui rend le stockage de table très flexible.

```
// À essayer : découvrir le  
stockage de table Azure
```

Stockage en file d'attente

Le stockage en file [d'attente Azure](#) est un type de stockage peu courant. Bien qu'il soit utilisé pour stocker de petits messages de données, il sert principalement de file d'attente. Vous placez des messages en file d'attente et d'autres processus les récupèrent. [Ce modèle](#) dissocie l'expéditeur du message du processus chargé de le traiter, ce qui accroît les performances et la fiabilité. Le stockage en file d'attente Azure existe dans des versions antérieures de Windows.

```
// À essayer : découvrir le stockage  
de files d'attente Azure
```

Stockage de fichiers

Le [stockage de fichiers Azure](#) peut faire office de lecteur pour le partage de fichiers. Il utilise le protocole SMB (Server Message Block), ce qui signifie qu'il est compatible Windows et Linux. Il est en outre accessible depuis le cloud ou des systèmes sur site. À l'instar des autres services Azure Storage, le stockage de fichiers est évolutif et économique.

```
// À essayer : découvrir le stockage  
de fichiers Azure
```

Stockage sur disque

[Le stockage sur disque Azure](#) est semblable au stockage de fichiers, mais est spécialement conçu pour des performances d'E-S élevées. Il peut être utilisé en tant que disque d'une machine virtuelle exigeant un haut niveau de performances, par exemple, pour exécuter SQL Server. Le stockage sur disque est uniquement disponible dans le niveau de tarification Premium d'Azure Storage.

Azure Data Lake Store

Les magasins de données précédents étaient destinés à des applications standard ou à des machines virtuelles. Le stockage [Azure Data Lake Store](#) est, quant à lui, conçu pour les applications Big Data. Il permet de stocker de grandes quantités de données dans leur format natif, qu'elles soient structurées, semi-structurées ou non structurées. Grâce à lui, vous pouvez conserver vos données brutes afin de les analyser ou de les transformer avant de les déplacer.

Voici les principales caractéristiques d'Azure Data Lake Store :

- Capacité de stockage illimitée. Un seul fichier peut contenir plus d'un pétaoctet de données, une taille 200 fois supérieure à celle prise en charge par les autres fournisseurs cloud.
- Performances évolutives pour des analyses massivement parallèles.
- Les données peuvent être stockées dans n'importe quel format, sans schéma.

Cette approche est très différente de l'entrepôt de données traditionnel qui exige la définition préalable de schémas de données.

Par exemple, vous pouvez stocker toutes les données provenant d'appareils IoT (Internet des objets) chargés de relever des températures dans Data Lake Store. Vous pouvez laisser les données dans le magasin, les filtrer, et créer une vue par heure ou par semaine. Le stockage Data Lake est assez économique. Plusieurs années de données peuvent y être conservées à un coût très bas.

```
// À essayer : découvrir Azure Data  
Lake Store à l'aide du portail  
Azure.
```

Azure SQL Data Warehouse

Si vous recherchez une solution d'entrepôt de données classique entièrement gérée, évolutive en taille, performante et sécurisée, [Azure SQL Data Warehouse](#) peut répondre à vos besoins. Stockez vos données selon des schémas prédéfinis et interrogez-les en utilisant le dialecte SQL Server auquel vous êtes habitué.

Comme SQL Data Warehouse fonctionne sous Azure, il vous offre un grand nombre de fonctions avancées. L'une d'elles est la détection automatique des menaces, qui s'appuie sur le Machine Learning pour analyser les tendances de votre scénario d'usage et, tel un système d'alarme, vous informer des violations potentielles.

L'un des cas d'utilisation les plus judicieux de SQL Data Warehouse se présente lorsque vous savez quels rapports présenter aux utilisateurs et connaissez le schéma de données de ces rapports. Vous pouvez ensuite créer des schémas dans SQL Data Warehouse et y ajouter des données qui seront consultables par les utilisateurs.

```
// À essayer : créer un entrepôt  
de données Azure SQL
```

Solutions d'analyse de données Azure

Un aspect presque aussi important que le stockage des données est la capacité de les analyser pour en tirer des informations. Azure fournit de nombreux services pour divers scénarios d'analyse de données. Ils vous permettent de tirer des informations utiles et précieuses de vos données, quelles que soient leur volume et leur complexité.

Azure Data Factory

Déplacer et transformer des données ne sont pas des tâches triviales, mais [Azure Data Factory](#) peut justement vous y aider. Dans Data Factory, vous pouvez créer un pipeline complet qui exécute d'un bout à l'autre le processus d'extraction, transformation et chargement (processus ETL).

Data Factory peut déplacer de manière fiable des données du site vers le cloud, à l'intérieur du cloud ou vers le site. La localisation des sources des données n'a pas d'importance. Data Factory fournit également de nombreux connecteurs qui vous permettent de vous connecter facilement à votre source de données, tels que SQL Server, Azure Cosmos DB, Oracle et [bien d'autres](#).

Lorsque vous déplacez des données, vous pouvez également les filtrer avant de les envoyer vers leur destination finale, les nettoyer ou les transformer par une activité dans le pipeline, telle qu'une activité [Apache Spark](#). En outre, Azure Data Factory vous permet de planifier et piloter les pipelines, ainsi que de [répliquer vos packages SSIS \(SQL Server Integration Services\)](#) dans le cloud.

```
// À essayer : créer une fabrique  
de données à l'aide du portail  
Azure
```

Azure Analysis Services

Avec [Azure Analysis Services](#), vous pouvez créer un modèle sémantique de vos données auxquels vos utilisateurs pourront accéder directement à l'aide d'outils de visualisation tels que Power BI. Construit sur les outils des [SQL Server Analysis Services](#) qui s'exécutent localement avec SQL Server, il fonctionne à présent de manière gérée dans le cloud. Cela signifie que le service est évolutif et que les données sont stockées de manière redondante. En quand vous ne l'utilisez pas, vous pouvez mettre le service en pause pour réduire les coûts.

Avec Azure Analysis Services, vous pouvez fournir des données modélisées directement aux utilisateurs, d'une manière très performante. Les utilisateurs peuvent interroger des millions d'enregistrements en quelques secondes parce que le modèle est entièrement dans la mémoire et qu'il est actualisé régulièrement.

Vous pouvez introduire dans le modèle sémantique des données provenant de n'importe où, notamment de sources de données dans le cloud ou sur site. Vous pouvez utiliser le stockage d'objets blob Azure, Azure SQL Database, Azure SQL Data Warehouse et [beaucoup d'autres services](#) comme sources de données pour le modèle. Vous pouvez également utiliser des sources de données comme un Active Directory local et des bases de données Access et Oracle.

```
// À essayer : créer un serveur  
Azure Analysis Services  
à l'aide du portail Azure
```

Azure Data Lake Analytics

Un autre service Azure dédié à l'exécution de tâches d'analyse de données est [Azure Data Lake Analytics](#). Avec ce service, vous pouvez analyser, traiter et transformer des quantités de données potentiellement énormes à partir d'Azure Storage et d'Azure Data Lake Store.

Azure Data Lake Analytics vous permet de créer et envoyer des tâches qui interrogent les données, les analysent ou les transforment. Vous pouvez écrire ces tâches en U-SQL, qui est un langage similaire à SQL, et prolonger U-SQL avec Microsoft R et Python.

Vous ne payez que les tâches que vous envoyez et exécutez ; le service se met à l'échelle automatiquement en fonction de la puissance nécessaire. Azure Data Lake Analytics sert le plus souvent à exécuter des tâches d'analyse de longue durée sur de grandes quantités de données.

```
// À essayer : créer votre  
premier script U-SQL à l'aide  
du portail Azure
```

Azure Stream Analytics

Vous pouvez utiliser le service [Azure Stream Analytics](#) pour analyser, interroger et filtrer les flux de données en temps réel. Par exemple, lorsque vous recevez un flux de données de température en provenance d'un appareil IoT, il vous indique s'il fait chaud dehors. Il peut indiquer la même température chaque seconde pendant une heure, jusqu'à ce que la température varie. Or, seules ces variations vous intéressent. Azure Stream Analytics peut interroger les données en temps réel et ne stocker que les données différentielles dans une base de données Azure SQL Database.

Azure Stream Analytics peut recevoir des données de nombreux services, y compris le stockage d'objets blob Azure, Azure Event Hubs ou Azure IoT Hub. Vous pouvez analyser les données en utilisant un langage simple de type SQL ou du code personnalisé. Après interrogation et filtrage du flux de données, Azure Stream Analytics peut envoyer ses résultats à de nombreux services Azure, notamment Azure SQL Database, Azure Storage et Azure Event Hubs.

```
// À essayer : créer une tâche  
Stream Analytics à l'aide  
du portail Azure
```

Azure Time Series Insights

Vous pouvez utiliser [Azure Time Series Insights](#) pour extraire rapidement des informations de grandes quantités de données de type IoT. Ce service extrait les données d'Azure Event Hubs, Azure IoT Hub et de vos propres entrées de référence, puis il conserve ces données pendant une durée spécifiée.

Avec Azure Time Series Insights, les utilisateurs peuvent donc interroger et analyser les données à l'aide d'un outil de visualisation au fur et à mesure qu'elles arrivent. Time Series Insights ne sert pas seulement à analyser les données, il peut aussi les assimiler et les conserver pendant un certain temps. Ce comportement est similaire à celui d'Azure Analysis Services, où les données résident dans un modèle in-memory interrogeable par les utilisateurs. Les principales différences sont que Time Series Insights est optimisé pour des données temporelles et IoT et qu'il contient son propre outil de visualisation de données.

```
// À essayer : explorer un
environnement de démonstration
Time Series Insights à partir
de votre navigateur
```

Azure Databricks

[Azure Databricks](#) vous permet d'exécuter un cluster Databricks évolutif et géré dans le cloud. Databricks fournit une plateforme d'analyse uniforme dotée d'un grand nombre d'outils et de fonctionnalités. Dans Databricks, vous pouvez exécuter des versions optimisées d'Apache Spark pour exécuter des analyses avancées de données.

En plus de l'analyse assurée par Spark, Databricks fournit des blocs-notes interactifs ainsi que des flux de production et des espaces de travail intégrés qui vous permettent de collaborer en équipe, notamment avec les data scientists, les data engineers et les analystes d'entreprise. Tous ont accès aux outils adaptés à leurs besoins spécifiques.

Azure Databricks est entièrement intégré à Azure Active Directory, qui vous donne la possibilité de mettre en œuvre une sécurité de niveau granulaire.

Avec Databricks, vous effectuez une analyse fondée sur Spark sur des données provenant de divers endroits, dont Azure Storage et Azure Data Lake Store. Databricks fonctionne également avec les données issues de Azure SQL Data Warehouse, Azure SQL Database et Azure Cosmos DB. De plus, vous pouvez relier Databricks à Power BI pour créer et afficher des tableaux de bord efficaces.

```
// À essayer : exécuter une tâche
Spark sur Azure Databricks
à l'aide du portail Azure
```

Azure HDInsight

[Azure HDInsight](#) est une plateforme interne à Azure permettant d'exécuter des services d'analyse de données open source. Vous pouvez également l'utiliser pour exécuter des clusters spécialisés de vos outils d'analyse de données open source préférés. L'avantage d'exécuter ces outils dans Azure HDInsight est qu'ils sont gérés, ce qui signifie que vous n'avez pas besoin de maintenir de machines virtuelles ni d'appliquer de correctifs aux systèmes d'exploitation. De plus, ils peuvent évoluer et se connecter facilement les uns aux autres, à d'autres services Azure, ainsi qu'aux sources de données et aux services locaux.

La plupart des types de clusters d'analyse de données open source spécialisés d'Azure HDInsight utilisent le stockage d'objets blob Azure ou Azure Data Lake Store pour accéder aux données ou les stocker, car ces services fonctionnent avec le système de fichiers Hadoop.

Vous pouvez exécuter différents types de clusters spécialisés, de taille potentiellement massive, tels que le cluster **Apache Hadoop**. Cela permet de traiter et d'analyser des données avec les outils Hadoop tels que Hive, Pig et Oozie.

Vous pouvez également lancer un **cluster Apache HBase**, qui fournit une base de données NoSQL très rapide. Les données résident en réalité au sein de Azure Storage ou de Azure Data Lake Store, mais HBase y ajoute au-dessus une couche d'abstraction, dotée de ses propres fonctionnalités et de performances uniques.

Vous pouvez créer un **cluster Apache Storm**, lequel est orienté vers l'analyse des flux de données, à l'instar d'Azure Stream Analytics. En outre, vous pouvez avoir un cluster **Apache Spark**, qui fournit une infrastructure de traitement et d'analyse de quantités massives de données. HDInsight peut également exécuter un cluster pour **Microsoft Machine Learning Server** (auparavant Microsoft R server).

Cela vous permet d'exécuter des tâches fondées sur R pour analyser les données. Enfin, vous pouvez créer un cluster qui exécute **Apache Kafka**, qui est un système de messagerie par publication-abonnement utilisé pour créer des applications comportant des mécanismes de mise en file d'attente.

Il existe plusieurs types de cluster, ainsi que d'outils utilisables dans ces clusters. Vous pouvez effectuer presque n'importe quelle tâche d'analyse et de traitement de données grâce à une combinaison de ces clusters. Ceux-ci sont tous gérés dans le cloud. Le Tableau 2-2 peut vous aider à choisir les services Azure adaptés à l'analyse de vos données.

// À essayer : extraire, transformer et charger des données à l'aide de Apache Hive sur Azure HDInsight

Tableau 2-2

	Data Factory*	Services d'analyse*	Data Lake Analytics*	Stream Analytics*	Time Series Insights*	Azure Databricks*	Azure HDInsight*
Déplacer les données d'un magasin à un autre	●						
Transformer les données	●	●	●	●	●	●	●
Interroger et filtrer les flux de données				●		●	●
Fournir aux utilisateurs un modèle sémantique in-memory		●			●		●
Permettre aux utilisateurs d'interroger les données et de créer des tableaux de bord					●		
Analyser les données pour utilisation ultérieure			●		●		●

* Les services marqués d'un astérisque comportent un niveau gratuit utilisable pour démarrer sans frais.

Procédure guidée : publier un site Web local sur Azure avec une base de données SQL

Avant de commencer, il vous faut :

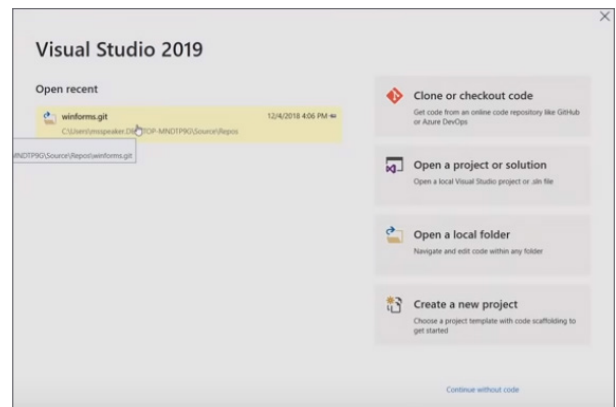
- Microsoft Visual Studio 2017 ou version ultérieure
- Un [compte Azure gratuit](#) pour suivre ce script de démonstration
- [Le code source Tailwind Traders Rewards](#)

Procédure guidée : migrer une application .NET vers Azure sans modification du code

Quand votre application se développe au-delà de la capacité de votre infrastructure locale, vous devez chercher de nouveaux moyens de répondre à la demande sans surcharger pour autant vos équipes.

Le cloud Azure propose un large éventail d'offres de services et de plateformes pour l'hébergement des applications. Pour commencer, utilisez **Azure App Service** pour héberger votre application sans modifier le code existant.

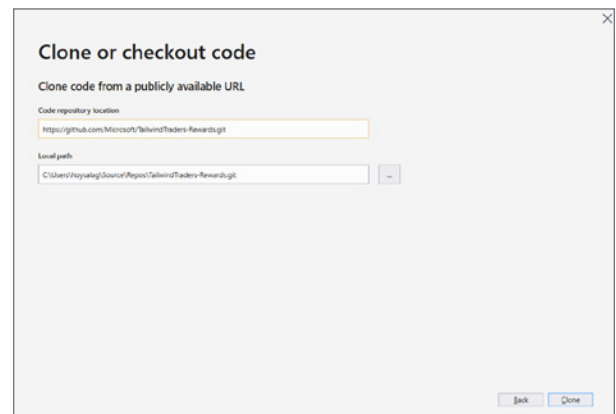
1. Lancez **Visual Studio 2019**. Vous remarquerez immédiatement la simplicité de son ouverture.



Lancer Visual Studio

2. Cliquez sur l'option **Cloner ou télécharger le code** et entrez l'URL du référentiel **Tailwind Traders Rewards** (<https://github.com/Microsoft/TailwindTraders-Rewards.git>) à l'emplacement du référentiel de code.

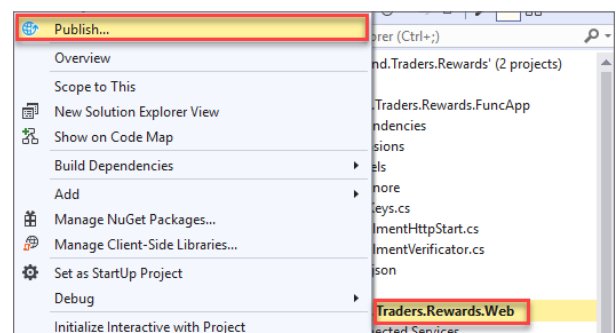
Cliquez sur **Cloner**. Sous **Solutions et dossiers**, cliquez sur **Tailwind.Traders.Rewards.sln** pour ouvrir la solution.



Clone Rewards

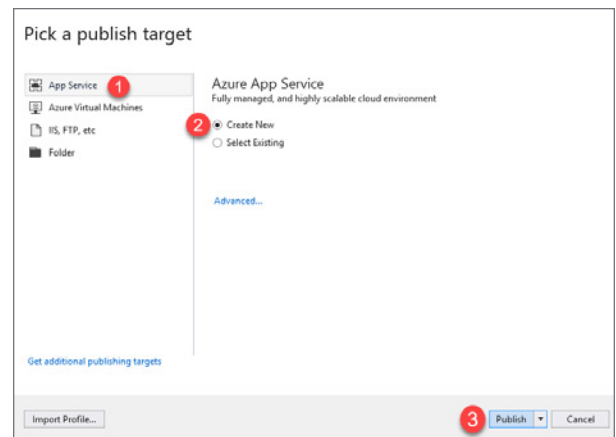
3. Cliquez avec le bouton droit sur le projet **Tailwind.Traders.Rewards.Web** et sélectionnez **Publier**. Il s'agit de la même boîte de dialogue Publier que vous utilisez pour effectuer un déploiement sur **IIS6** dans votre infrastructure locale.

Cette boîte de dialogue Publier va vous permettre de déployer l'application sur la plateforme cloud Azure.



Publier l'application

4. Sélectionnez **App Service** comme cible de la publication. Dans la fenêtre **Azure App Service**, sélectionnez **Créer** et cliquez sur **Publier**.

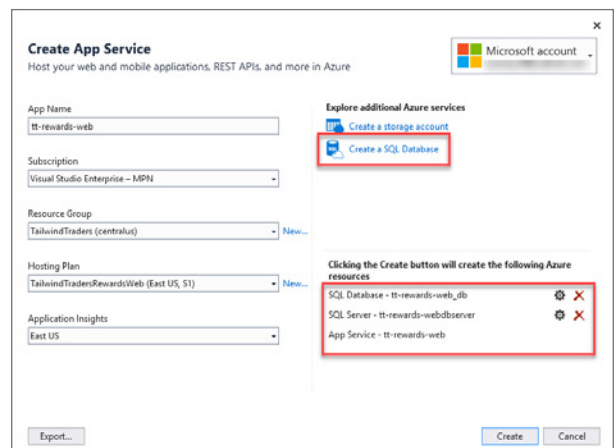


Options de publication

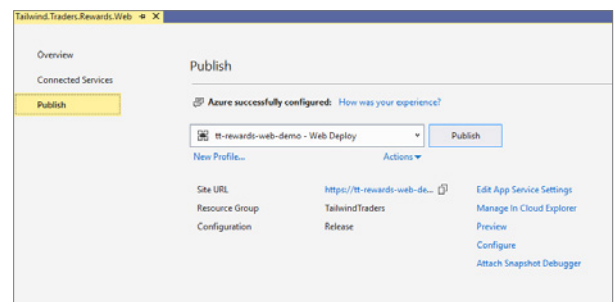
5. Dans la fenêtre suivante, entrez les informations de votre abonnement Azure, sélectionnez le **Service d'application**, puis choisissez les options existantes pour **Groupe de ressources**, **Plan d'hébergement** et **Application Insights**, ou créez-en de nouvelles.

Cliquez sur l'option **Créer une base de données SQL** sur le côté droit et créez un nouveau serveur et une nouvelle base de données dans les fenêtres qui s'ouvrent.

Pour finir, cliquez sur **Créer** pour créer un profil de publication. Vous pouvez également créer la base de données Azure SQL directement sur le portail Azure.



Créer un profil

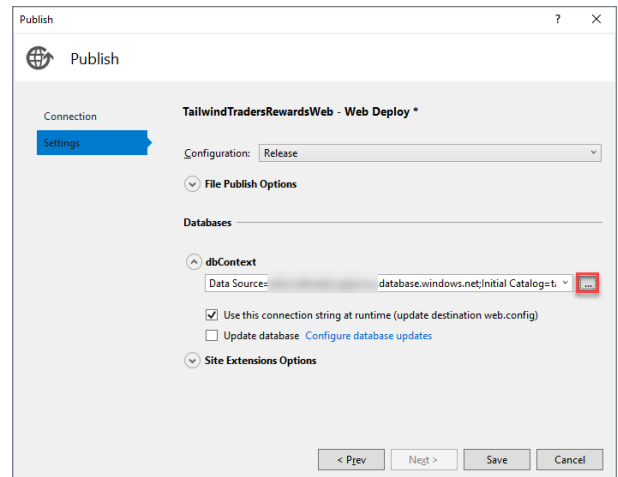


Créer un profil

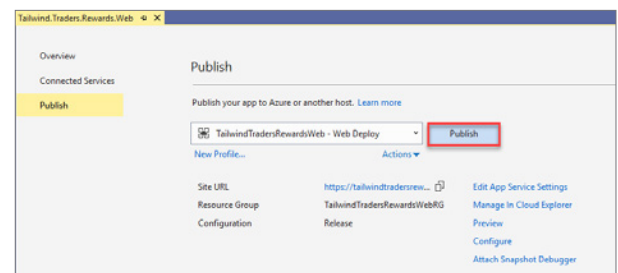
6. Cliquez sur **Configurer** dans la fenêtre Publier pour vérifier les chaînes de connexion à la base de données. Vous pouvez renseigner les chaînes de connexion à la base de données en sélectionnant le bouton avec les points de suspension et en saisissant les informations de la base de données SQL. Quand vous cliquez sur **Publier**, le fichier de configuration Web est mis à jour avec cette chaîne de base de données, qui pointe vers une base de données SQL.

Lors du débogage local de l'application, ce sont les services IIS (Internet Information Services) et le serveur SQL locaux qui entrent en jeu, mais lorsque l'application est publiée, ils sont remplacés par les services Azure créés.

7. Cliquez sur **Publier** pour déployer l'application sur Azure App Service et le back-end sur la base de données SQL.

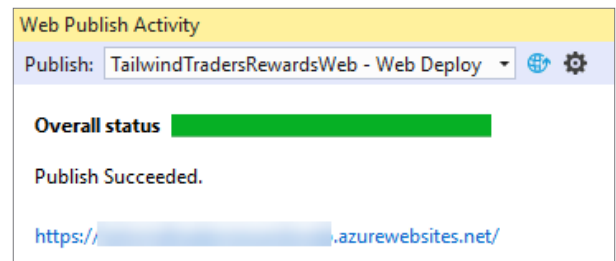


Chaînes de connexion à la base de données

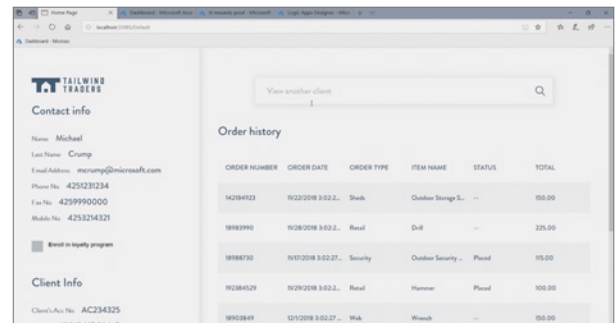


Publier l'application

8. Une fois l'application publiée, son statut affiche **Publication réussie** et l'application Web s'ouvre dans le navigateur. Le site Web affiche désormais les données de la base de données SQL.



Publication réussie



Application Web

Autres lectures

Si vous souhaitez en savoir plus sur les données et leur analyse dans Azure, téléchargez et lisez ces Ebooks gratuits :

// Guide de NoSQL avec Azure
Cosmos DB

// Azure pour les architectes

// Migration d'applications .NET
vers Azure

03 /

Sécurisation de votre application

Comment Azure peut-il contribuer à sécuriser votre application ?

Une de vos applications a-t-elle jamais présenté un incident de sécurité ? Peut-être que oui, bien que vous ne le sachiez pas. Avec Azure, vous pouvez protéger vos données, vos applications et votre infrastructure avec des services de sécurité intégrés assortis du renseignement de sécurité. Ils vous permettent d'identifier sans délai les menaces en évolution constante et d'y répondre rapidement.

Azure vous permet de mettre en place une stratégie de défense en profondeur, avec plusieurs niveaux, aussi bien sur les identités que sur les données, les hôtes et les réseaux. Grâce aux services tels que [Azure Security Center](#), vous bénéficiez d'une vue d'ensemble de votre posture de sécurité et de recommandations pour l'améliorer.

Mieux encore, vous êtes averti des incidents de sécurité dès leur survenue. Vous êtes ainsi toujours au fait des menaces présentes. Et vous pouvez prendre immédiatement les mesures adéquates pour protéger vos actifs. Nous examinons certains d'entre elles dans ce chapitre.

Azure Active Directory

Une partie importante de la sécurité de votre application consiste à authentifier les utilisateurs avant qu'ils puissent l'utiliser. Mais l'authentification n'est pas toujours facile à mettre en œuvre. Vous



devez stocker les identités et les identifications de connexion des utilisateurs quelque part, mettre en œuvre la gestion des mots de passe, créer une poignée de main d'authentification sécurisée et ainsi de suite.

[Azure Active Directory](#) (Azure AD) fournit tous ces éléments et plus encore, prêts à l'emploi. Vous stockez vos identités d'utilisateurs dans Azure AD et faites en sorte que les utilisateurs s'authentifient par rapport à celles-ci, et vous les redirigez vers votre application une fois qu'ils sont effectivement authentifiés. Azure AD s'occupe de gérer les mots de passe, ainsi que de répondre aux scénarios courants comme « J'ai oublié mon mot de passe ».

Comme Azure AD est utilisé quotidiennement par des millions d'applications, y compris le [portail Azure](#), [Outlook.com](#) et [Office 365](#), il a la capacité de détecter sans délai les comportements malveillants et d'y réagir. Par exemple, si un utilisateur se connecte à une application à partir d'un emplacement situé en Europe et puis, une minute plus tard, se connecte à partir de l'Australie, Azure AD signale cela comme un comportement malveillant et demande à l'utilisateur des informations d'identification supplémentaires par l'intermédiaire de l'authentification multifacteur.

Azure Key Vault

Dans le cadre de votre architecture de sécurité, vous avez besoin d'un lieu sûr pour stocker et gérer les certificats, les clés et autres secrets. [Azure Key Vault](#) offre cette possibilité. Avec Azure Key Vault, vous pouvez stocker les secrets que vos applications utilisent dans un emplacement centralisé.

// À essayer : découvrir
Azure Key Vault

Il peut s'agir d'informations de connexion stockées dans une chaîne de connexion. Votre application obtiendra la chaîne de connexion à partir d'Azure Key Vault plutôt qu'à partir du système de configuration. De cette façon, les administrateurs peuvent contrôler les secrets, et les développeurs n'ont jamais besoin de s'en préoccuper. Azure Key Vault stocke également les certificats SSL et autres qui vous servent à sécuriser le trafic à destination et en provenance de vos applications sur HTTPS.

Azure Sentinel

Pour avoir une bonne vue d'ensemble de l'état de sécurité de votre entreprise et de tous ses utilisateurs, applications, services et données, vous pouvez faire appel à une plateforme de gestion des informations et des événements de sécurité (SIEM, Security Information and Event Manager). Azure offre aujourd'hui une plateforme SIEM assistée par l'IA sous le nom d'[Azure Sentinel](#).

Utilisez Azure Sentinel pour recueillir des données à partir de votre entreprise, notamment en ce qui concerne les utilisateurs, les applications, les serveurs et les éléments d'infrastructure tels que les pare-feu et les appareils en exécution dans le cloud et sur site. Grâce aux connecteurs intégrés, la collecte des informations de votre entreprise est facile. À mesure que les données sont recueillies, Azure Sentinel détecte les menaces de sécurité et réduit au minimum les faux positifs grâce à ses algorithmes intelligents de Machine Learning.

Quand la menace est réelle, vous recevez une alerte et pouvez lancer vos investigations avec l'IA en profitant des décennies de savoir-faire de Microsoft en matière de cybersécurité. Vous pouvez répondre aux incidents par l'automatisation des tâches et l'orchestration du workflow intégré d'Azure Sentinel.

// Découvrir l'intégration d'Azure
Sentinel

Gestion des API Azure

Les API doivent être sécurisées. Cela est vrai pour les API que vous créez vous-même ainsi que pour celles qui proviennent de fournisseurs tiers. Pour vous aider à sécuriser vos API, vous pouvez utiliser la [Gestion des API Azure](#). Il s'agit essentiellement d'un proxy que vous placez devant vos API, qui ajoute des fonctionnalités telles que la mise en cache, la limitation, l'authentification ou l'autorisation.

Avec la Gestion des API, vous pouvez sécuriser chaque API en demandant aux utilisateurs de créer un abonnement correspondant. Ainsi, les applications doivent s'authentifier avant de pouvoir utiliser votre API. Vous pouvez utiliser différentes méthodes d'authentification comme les jetons d'accès, l'authentification de base et les certificats. En outre, vous pouvez effectuer le suivi de qui appelle votre API et bloquer les appelants indésirables.

Au-delà de la sécurité

La sécurité est un aspect essentiel. Mais la Gestion des API Azure propose d'autres fonctionnalités susceptibles de rationaliser votre workflow de développement et de test, telles que la [simulation de réponse des données de test](#), la [publication de plusieurs versions de l'API](#), l'[introduction en toute sécurité de modifications non-critiques dans les révisions](#) et l'accès pour les développeurs à la documentation des API générée automatiquement, au catalogue d'API et à des exemples de code.

// À essayer : découvrir la Gestion des API Azure

Proxy d'application Azure AD

Le [proxy d'application Azure AD](#) fournit l'authentification unique (SSO, Single Sign-On) et l'accès à distance sécurisé aux applications Web hébergées sur site. Les applications que vous voudrez probablement publier sont notamment des sites SharePoint, Outlook Web Access ou d'autres applications métier Web. Ces applications Web locales s'intègrent à Azure AD, la même plateforme de contrôle et d'identité qu'utilise Office 365. Les utilisateurs finaux peuvent accéder à vos applications locales de la même manière qu'ils accèdent à Office 365 et à d'autres applications SaaS intégrées avec Azure AD.

Identités gérées pour les ressources Azure

Comment gardez-vous les informations d'identification complètement à l'écart de votre code ? Vous pouvez commencer par utiliser Azure Key Vault, mais où stocker les informations d'identification permettant de se connecter à Key Vault ? [Les Identités gérées pour les ressources Azure](#) sont la solution.

Vous pouvez utiliser les identités gérées pour [un grand nombre de services Azure](#), notamment Azure App Service. Il vous suffit d'activer les identités gérées à l'aide d'un simple bouton pour injecter les informations d'identification dans votre application au moment de l'exécution, puis d'utiliser ces informations d'identification pour accéder à d'autres services, tels qu'Azure Key Vault. Toutes les authentifications entre services se font au niveau de l'infrastructure, si bien que votre application n'a pas à s'en occuper et peut simplement utiliser les autres services.

// À essayer : utiliser les identités gérées pour les ressources Azure dans App Service et Azure Functions

Chiffrement

Chiffrement par défaut des données

Par défaut, vos données sont chiffrées dans Azure lorsque vous les stockez dans Azure SQL Database, Azure SQL Data Warehouse, Azure Database pour MySQL, Azure Database pour PostgreSQL, Azure Storage, Azure Cosmos DB ou Azure Data Lake Store. Ce chiffrement fonctionne automatiquement et vous n'avez rien à configurer lorsque vous l'utilisez.

Pour répondre à vos exigences de sécurité et de conformité, vous pouvez utiliser les fonctions suivantes pour chiffrer les données au repos :

- Le [Chiffrement de disque Azure](#) chiffre le démarrage des machines virtuelles IaaS (Infrastructure as a Service) Windows et Linux, ainsi que les volumes de données à l'aide de clés gérées par les clients
- Le [Chiffrement des services Azure Storage](#) chiffre automatiquement les données avant de les laisser résider dans Azure Storage, puis les déchiffre automatiquement au moment où vous les extrayez.
- Le [Chiffrement Azure côté client](#) prend en charge le chiffrement des données au sein des applications clientes avant de les charger vers Azure Storage ou d'autres points de terminaison, puis les déchiffre au moment de les télécharger vers le client.
- [SQL Transparent Data Encryption](#) (TDE) chiffre les fichiers de données [SQL Server](#), [Azure SQL Database](#) et [Azure SQL Data Warehouse](#). Les données et les fichiers journaux sont chiffrés à l'aide d'algorithmes de chiffrement standards du secteur. Les pages des bases de données sont chiffrées avant d'être écrites sur le disque et sont déchiffrées lorsqu'elles sont lues.
- La fonctionnalité SQL [Always Encrypted](#) (chiffrement permanent) chiffre les données au sein des applications clientes avant de les entreposer dans Azure SQL Database. Elle permet de déléguer l'administration des bases de données locales à des tiers et maintient la séparation entre ceux qui possèdent et peuvent afficher les données et ceux qui les gèrent mais ne doivent pas y accéder.
- [Azure Cosmos DB](#) ne nécessite aucune action de votre part parce que les données utilisateur stockées dans Azure Cosmos DB en mémoire rémanente (disques SSD) sont chiffrées par défaut, et il n'existe aucun contrôle pour activer ou désactiver ce chiffrement.

Azure Security Center

[Azure Security Center](#) assure une gestion de la sécurité unifiée et une protection avancée contre les menaces au niveau des charges de travail de cloud hybride. Il offre des contrôles de stratégie centralisée pour limiter l'exposition aux menaces et détecter et corriger rapidement les vulnérabilités.

En outre, Security Center prend en charge l'intégration avec des solutions tierces et peut être personnalisé avec des fonctionnalités de programmation et d'automatisation. Vous pouvez utiliser Security Center pour analyser l'état de sécurité de vos ressources informatiques, de vos réseaux virtuels, de vos services de stockage et de données et de vos applications.

L'évaluation continue vous permet de détecter les problèmes de sécurité potentiels, tels que les systèmes dont les mises à jour de sécurité sont absentes ou dont les ports réseaux sont exposés. La liste des découvertes et des recommandations, triée par ordre de priorité, peut permettre de déclencher des alertes ou d'autres mesures correctives orientées.

Protection Azure DDoS

Vous en avez entendu parler et vous ne voulez certainement pas que cela arrive à votre entreprise : certaines applications subissent des attaques de déni de service distribué (DDoS). Ces types d'attaques deviennent de plus en plus fréquents et peuvent submerger votre application au point que personne ne puisse plus l'utiliser. Le [service de protection Azure DDoS](#) protège contre les attaques par déni de service selon deux niveaux : un gratuit (le niveau De base) et un payant (le niveau Standard).

Vous n'avez rien à faire pour activer le niveau De base ; il est activé automatiquement pour tous les clients dans le cadre de la plateforme Azure. Ce service protège vos applications contre les attaques DDoS le plus courantes en effectuant une surveillance et en prenant des mesures d'atténuation en temps réel. Il déploie les mêmes défenses que celles qu'utilise Microsoft Online Services (MOS).

Le niveau Standard fournit des capacités d'atténuation supplémentaires qui sont ajustées tout spécialement pour les ressources de réseau virtuel Azure. Il est simple à activer, et vous n'avez pas à modifier vos applications ; tout est fait au niveau du réseau. De plus, avec le niveau Standard, vous pouvez personnaliser la protection du niveau De base à l'aide de vos propres stratégies qui mettent l'accent sur vos applications et scénarios d'utilisation spécifiques.

// Plus d'informations : En savoir plus sur la protection Azure DDoS

Passerelles VPN Azure

La [passerelle VPN Azure](#) est l'une des nombreuses options permettant de connecter Azure à votre réseau local. Vous avez la possibilité de configurer une connexion VPN de site à site (S2S) chiffrée entre un réseau virtuel Azure et votre réseau local.

Le trafic est sécurisé parce qu'il est chiffré (même s'il circule sur l'Internet public). La passerelle VPN peut envoyer le trafic chiffré entre réseaux virtuels Azure sur le réseau Microsoft. Vous pouvez également créer des connexions point à site (P2S) chiffrées entre votre ordinateur et Azure. Vous bénéficiez de cette manière d'une connexion privée et sécurisée à Azure, même en déplacement.

```
// Prise en main et création  
d'une passerelle VPN Azure  
avec PowerShell
```

Azure Application Gateway

[Azure Application Gateway](#) est un dispositif virtuel dédié qui fournit un contrôleur de fourniture d'application (ADC) en tant que service. Il offre diverses fonctionnalités d'équilibrage de charge de couche 7 pour votre application et permet aux clients d'optimiser la productivité de leur batterie de serveurs web et déportant la terminaison SSL (très gourmande en ressources de processeur) sur la passerelle d'application. La passerelle fournit également d'autres fonctionnalités de routage de couche 7, y compris la répartition en tourniquet du trafic entrant, l'affinité de session basée sur les cookies, le routage par chemin d'URL et la capacité d'héberger plusieurs sites web derrière une seule passerelle d'application.

Pare-feu d'applications web Azure

Vous devez protéger votre application contre de nombreuses menaces, notamment l'injection SQL, les scripts intersites (XSS) et toutes les autres menaces définies dans la spécification Open Web Application Security Project (OWASP). [Le pare-feu d'applications Web Azure](#) peut vous y aider. Le pare-feu d'applications Web Azure, une fonctionnalité du service [Azure Application Gateway](#), assure la protection en temps réel de votre application. Il détecte les attaques malveillantes, telles qu'elles sont définies dans [l'ensemble de règles de base OWASP](#), et empêche ces attaques d'atteindre votre application. Il génère également des rapports sur les attaques en cours, afin de vous permettre d'identifier les menaces actives contre votre application. Cela ajoute une couche de sécurité supplémentaire.

Azure Network Watcher

[Azure Network Watcher](#) est un service régional qui vous permet de surveiller et diagnostiquer l'état du réseau à l'intérieur comme en provenance ou à destination d'Azure.

Ses nombreux outils de diagnostic et de visualisation peuvent vous aider à comprendre plus en profondeur votre réseau dans Azure.

Voici quelques exemples :

- [Topologie](#) : fournit une vue du réseau qui met en évidence les différentes interconnexions et associations entre les ressources réseau dans un groupe de ressources.
- [Capture des paquets variables](#) : capture des paquets de données entrant et sortant d'une machine virtuelle (VM). Les options de filtrage avancées et les contrôles fins, tels que la possibilité de définir des limites de taille et de durée, sont gages de souplesse. Les données des paquets peuvent être stockées dans un magasin d'objets blob, ou sur le disque local au format .cap.
- [Vérification du flux IP](#) : vérifie si un paquet est accepté ou refusé d'après les 5 paramètres d'informations de flux et de paquet (IP de destination IP, IP Source, Port de destination, Port source et Protocole). Si le paquet est refusé par un groupe de sécurité, la règle et le groupe qui ont refusé le paquet sont retournés.

Groupes de sécurité réseau

Un [groupe de sécurité réseau](#) (NSG) contient une liste de règles de sécurité qui autorisent ou refusent le trafic réseau vers les ressources connectées aux réseaux virtuels Azure. Les NSG peuvent être associés à des sous-réseaux, à des VM individuelles (classiques) ou à des contrôleurs d'interface réseau individuels (NIC) reliés aux machines virtuelles (de type Resource Manager). Lorsqu'un NSG est associé à un sous-réseau, ses règles s'appliquent à toutes les ressources connectées au sous-réseau. Vous pouvez restreindre encore davantage le trafic en associant aussi un NSG à une machine virtuelle ou à un NIC

Zones privées Azure DNS

Le DNS est responsable de traduire (ou résoudre) un nom de service en son adresse IP. Azure DNS est un service d'hébergement de domaines DNS. Il assure la résolution des noms à l'aide de l'infrastructure Azure. En plus des domaines DNS connectés à Internet, Azure DNS prend désormais en charge les domaines DNS privés (comme fonctionnalité d'évaluation) avec les Zones privées Azure DNS. Les zones privées DNS renforcent la sécurité, notamment grâce à la possibilité de créer une infrastructure DNS fractionnée. Cela vous permet de créer des zones DNS publiques et privées portant le même nom sans pour autant exposer les noms internes. En outre, l'utilisation de la fonction de zones privées DNS supprime la nécessité d'introduire des solutions DNS personnalisées, lesquelles augmentent la surface d'attaque globale en raison d'exigences de gestion et de mise à jour distinctes.

// Plus d'informations : En savoir plus sur les zones privées DNS

VPN sur plusieurs sites

Azure prend en charge deux types de connexions VPN entre sites : VPN P2S et VPN S2S. Une connexion VPN P2S (de point à site) vous permet de créer une connexion sécurisée à votre réseau virtuel à partir d'un ordinateur client individuel. Ce type de connexion est établi à partir de l'ordinateur client, ce qui est utile pour les télétravailleurs qui veulent se connecter à des réseaux virtuels Azure depuis un emplacement distant. Une connexion VPN P2S est également utile lorsque vous n'avez que quelques clients qui ont besoin de se connecter à un réseau virtuel. Par opposition, une connexion VPN S2S (de site à site) est utilisée pour connecter votre réseau local à un réseau virtuel Azure sur un tunnel VPN IPsec/IKE (IKEv1 ou IKEv2). Ce type de connexion nécessite un dispositif VPN situé sur site disposant d'une adresse IP publique exposée à l'extérieur.

// Plus d'informations : En savoir plus sur les VPN [P2S](#) et [S2S](#)

Azure ExpressRoute

[Azure ExpressRoute](#) vous permet d'étendre vos réseaux locaux dans le cloud Microsoft sur une connexion privée sécurisée, assurée par un fournisseur de connectivité, sans passer par Internet. Avec ExpressRoute, vous pouvez établir des connexions aux services de cloud Microsoft, par exemple Azure, Office 365 et Dynamics 365.

Azure Load Balancer

Vous pouvez utiliser des équilibreurs de charge pour augmenter la disponibilité de vos applications. Azure prend en charge des équilibreurs de charge externes et internes, pouvant être utilisés en configuration publique ou interne.

En outre, vous pouvez configurer des équilibreurs de charge pour prendre en charge les ports à haute disponibilité (HA), quand une règle des ports HA est une variante d'une règle d'équilibrage de charge configurée sur équilibreur de charge interne standard. Vous pouvez indiquer une règle unique pour équilibrer la charge de tous les flux TCP et UDP arrivant sur tous les ports d'un équilibreur de charge interne.

```
// Plus d'informations : En savoir  
plus sur les équilibreurs de  
charge et les règles de ports HA
```


Journalisation et surveillance

Azure Log Analytics

[Azure Log Analytics](#) vous aide à collecter et analyser les données générées par des ressources dans votre environnement cloud et sur site. Il fournit des informations en temps réel à l'aide de la recherche intégrée et de tableaux de bord personnalisés pour analyser des millions d'enregistrements des charges de travail et serveurs quel que soit leur emplacement physique.

Azure Monitor

[Azure Monitor](#) assure une surveillance de base des services Azure par la collecte de mesures, de journaux d'activité et de journaux de diagnostic. Les éléments de mesure collectés fournissent des statistiques de performance pour différentes ressources, notamment pour le système d'exploitation associé à une machine virtuelle.

Le journal d'activité signale quand des ressources sont créées ou modifiées. Vous pouvez afficher ces données avec l'un des explorateurs du portail Azure, et les envoyer à Azure Log Analytics pour établir les tendances et des analyses détaillées, ou vous pouvez créer des règles d'alerte qui signalent les problèmes critiques de manière proactive.

Journaux de flux des NSG Azure

Fonction de Network Watcher, [les journaux de flux des NSG Azure](#) vous permettent d'afficher des informations sur le trafic IP entrant et sortant par un groupe de sécurité réseau. Les journaux de flux peuvent être analysés en vue de tirer des informations sur le trafic et la sécurité réseau, ainsi que sur les problèmes de performance liés au trafic réseau.

Les journaux de flux ciblent des NSG ; ils ne se présentent donc pas de la même manière que les autres journaux et ne sont conservés que dans un compte de stockage.

Azure Monitor Application Insights

[Azure Monitor Application Insights](#) est un service extensible de gestion des performances d'application (APM) destiné aux développeurs Web qui travaillent sur plusieurs plateformes. Il comprend de puissants outils d'analyse pour vous aider à diagnostiquer les problèmes et à comprendre ce que font les utilisateurs avec votre application. Il convient aux applications exécutées sur une grande variété de plateformes ou dans le cloud, y compris .NET, Node.js et J2EE.

Applications Insights s'intègre à votre processus DevOps et a des points de connexion avec une grande diversité d'outils de développement. Il peut surveiller et analyser les données de télémétrie des applications mobiles en s'intégrant avec Visual Studio App Center.

Plan de sécurité et conformité Azure

Le plan de sécurité et de conformité Azure pour les données de santé HIPAA/HITRUST et l'intelligence artificielle fournit des outils et des conseils pour le déploiement d'un environnement de plateforme en tant que service conforme aux directives HIPAA (Health Insurance Portability and Accountability Act) et HITRUST (Health Insurance Portability and Accountability Act).

Cette offre PaaS prend en charge l'ingestion, le stockage, l'analyse et l'interaction avec les dossiers médicaux personnels et non personnels dans un environnement cloud à plusieurs niveaux déployé en tant que solution end-to-end. Ce plan d'action met en avant une architecture de référence commune applicable à d'autres scénarios d'usage que ceux du secteur de la santé. Il est en outre conçu pour faciliter l'adoption d'Azure.

// Plus d'informations : En savoir plus sur le plan de sécurité et de conformité Azure

Documentation technique et architecturale de la sécurité Azure

Azure entretient une vaste bibliothèque de documentation technique concernant la sécurité. Elle complète les informations de sécurité incluses dans chaque service individuellement. Livres blancs, documents sur les meilleures pratiques et check-lists sont présents sur la page d'informations sur la sécurité d'Azure.

Elle contient également de la documentation relative à la sécurité du cloud public dans divers domaines, notamment la sécurité réseau, la sécurité du stockage, la sécurité du calcul, la gestion des accès et des identités, la journalisation et la préparation d'audit, la protection des charges de travail cloud, la sécurité PaaS, et bien d'autres sujets.

// Plus d'informations : En savoir plus sur la page d'informations sur la sécurité d'Azure

Autres lectures

Vous en saurez plus sur la sécurité d'Azure dans les Ebooks gratuits suivants :

// Stratégie de cloud d'entreprise

// Azure pour les architectes

04 /

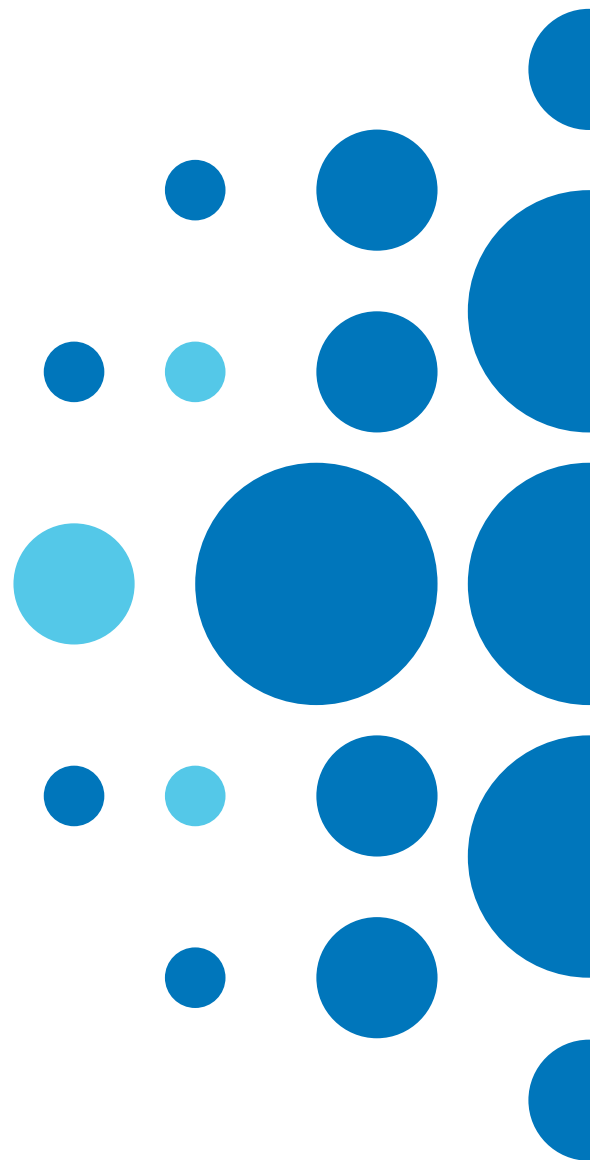
Ajout de fonctions intelligentes à votre application

Comment Azure peut-il intégrer l'IA à votre application ?

L'IA peut donner à votre application une longueur d'avance sur la concurrence. Imaginez le champ des possibles : une application qui traduit les paroles en temps réel au fil du discours, ou une autre qui reconnaît les pièces d'un moteur dans une formation en réalité mixte. Ce champ est illimité. Mais la création d'une application assistée par l'IA peut sembler difficile. Par exemple, comment créer un algorithme capable de comprendre la parole ?

Heureusement, vous n'êtes pas obligé de tout faire vous-même. Azure est là pour vous aider. Vous y trouverez de nombreux services qu'il vous suffira de connecter à votre application, tels que le service [Traduction de parole](#) (un service d'[Azure Cognitive Services](#)) qui traduit le discours en temps réel.

Si vous utilisez un service comme celui-ci, vous êtes simple consommateur d'IA. Mais vous pouvez aussi créer vos propres algorithmes de Machine Learning en vous appuyant sur des services tels que [Azure Machine Learning Studio](#) et le service [Azure Machine Learning](#).



Qu'utiliser, et quand ?

Avant d'explorer plus avant les options IA dans Azure, examinons le Tableau 4-1, qui récapitule les services disponibles et leurs fonctionnalités.

Tableau 4-1

	Azure Search*	Azure Cognitive Services*	Azure Bot Service*	Azure Machine Learning Studio*	Azure Machine Learning*	Azure Spatial Anchors*	Azure Remote Rendering*
Déplacer les données d'un magasin à un autre	●						
Transformer les données	●	●	●	●	●	●	●
Interroger et filtrer les flux de données				●		●	●
Fournir aux utilisateurs un modèle sémantique in-memory		●			●		●
Les utilisateurs peuvent interroger les données et créer des tableaux de bord					●		
Analyser les données pour utilisation ultérieure			●		●		●

* Les services marqués d'un astérisque comportent un niveau gratuit utilisable pour démarrer sans frais.

Azure Search

[Azure Search](#) est une fonctionnalité courante.

Pourtant, la recherche est généralement une fonction difficile à mettre en œuvre. Azure Search effectue le plus gros du travail nécessaire à l'exécution de recherches. Il vous suffit de lancer une instance d'Azure Search, de créer un index pour faciliter la recherche et de l'alimenter en données. Cela signifie par exemple que vous pouvez déployer facilement Azure Search pour que les utilisateurs effectuent des recherches dans votre catalogue de produits au sein d'une application de commerce électronique.

Vous disposez d'un grand nombre d'options pour affiner Azure Search, et de nombreuses fonctions facilitent la recherche pour vos utilisateurs :

- La recherche géospatiale, qui permet aux utilisateurs d'explorer les données selon la proximité d'un résultat de recherche avec un emplacement physique.
- Les analyseurs linguistiques Apache Lucene et le traitement du langage naturel de Microsoft, disponibles dans 56 langues, qui traitent intelligemment les caractéristiques linguistiques, notamment les temps des verbes, le masculin et le féminin, les pluriels irréguliers, la décomposition des mots, leur césure (pour les langues sans espaces), etc.
- La surveillance et la création de rapports, qui renseignent sur l'objet des recherches et sur le fait que ces dernières ont été rapides et ont abouti.

- Des fonctions d'expérience utilisateur, comme le tri et la pagination des résultats de recherche, le filtrage intelligent et les suggestions de recherche, sont également disponibles.
- [La recherche cognitive](#), qui est une approche de compréhension du contenu reposant principalement sur l'IA. La recherche cognitive est assurée par Azure Search avec les Cognitive Services (services cognitifs) intégrés. Elle extrait des données de pratiquement n'importe quelle source et y applique un ensemble de facultés cognitives modulables qui en tirent les informations. Ces informations sont alors organisées et stockées dans un index, et constituent un nouveau potentiel d'exploration des données par Azure Search.

La recherche cognitive est utilisée dans les entreprises pétrolières et gazières, dont les équipes de géologues et d'autres spécialistes sont chargées de comprendre les données sismiques et géologiques. Ces équipes s'appuient souvent sur des décennies de fichiers PDF contenant des photos d'échantillons et des masses de notes de terrain manuscrites. Elles doivent relier les lieux, les experts et les événements et parvenir à s'y retrouver dans cette masse d'informations pour pouvoir prendre d'importantes décisions.

La recherche cognitive utilise Cognitive Services pour analyser toutes ces données, y compris les fichiers PDF et les images, pour extraire des informations et les corréler ; tout cela sans nécessiter de fabriquer de complexes logiciels de reconnaissance d'image ou de reconnaissance optique de caractères.

// À essayer : créez votre premier index Azure Search dans le portail

Cognitive Services

[Cognitive Services](#) offre en tant que service des algorithmes créés par Microsoft et des données de Machine Learning. Pour la plupart des services, Microsoft fournit aussi les données sur lesquelles s'entraînent ces algorithmes. Vous pouvez toutefois, pour certains services, utiliser vos propres données personnalisées pour entraîner les algorithmes.

Cognitive Services offre un moyen exceptionnellement facile d'incorporer le Machine Learning et l'IA dans votre application, par simple appel [d'API](#).

Le Tableau 4-2 présente les API actuellement disponibles. Notez que cette liste continue de s'allonger.

Tous les services incluent un niveau gratuit que vous pouvez utiliser pour commencer.

Chaque catégorie du tableau regroupe plusieurs services que vous pouvez utiliser en appelant une API. Certaines catégories contiennent des services personnalisés, comme le service Custom Vision (Vision personnalisée), le service Language Understanding (Compréhension linguistique) et le service de recherche personnalisée Bing. Ces services personnalisés fournissent des algorithmes de Machine Learning préconfigurés, tout comme les autres services ; ils vous permettent aussi d'utiliser vos propres données pour entraîner le modèle. Outre ces services, vous pouvez utiliser ceux des [laboratoires Cognitive Services](#). Les laboratoires contiennent des services expérimentaux que Microsoft teste actuellement pour voir s'ils sont bien adaptés aux scénarios d'utilisation de ses clients. L'un de ces services expérimentaux est [Project Gesture](#). Il permet de détecter les gestes (un salut de la main, par exemple) et de les intégrer dans l'expérience utilisateur.

Regardons de plus près quelques composants des Cognitive Services.

// À essayer : explorer Cognitive Services

Tableau 4-2

Vision	Parole	Langue	Connaissances	Recherche
Vision par ordinateur Visage Indexeur vidéo Modérateur de contenu Custom Vision	Reconnaissance vocale Conversion de texte en parole Traduction de parole Reconnaissance d'interlocuteur	Analyse de texte Traduction de texte Vérification orthographique Bing Modérateur de contenu Compréhension linguistique	QnA Maker	Recherche Web Bing Recherche visuelle Bing Recherche d'entité Bing Moteur de recherche Bing News Recherche personnalisée Bing Moteur de recherche Bing Image Suggestions automatiques Bing Moteur de recherche Bing Video Recherche Bing Adresses

* Tous les services incluent un niveau gratuit que vous pouvez utiliser pour commencer.

Compréhension linguistique

Utilisez le service [Compréhension linguistique \(LUIS\)](#) pour comprendre ce que vous disent les usagers des réseaux sociaux, les utilisateurs des chatbots et des applications vocales. Vous pouvez par exemple réserver un vol ou programmer une réunion.

Pour utiliser le service Compréhension linguistique, donnez-lui des exemples de ce que vous souhaitez qu'il comprenne, comme « Réserver un vol pour Lyon » ou « Programmer une réunion à 13 h 00 avec Paul », et dites-lui les mots que vous recherchez. Dans les exemples précédents, vous recherchez probablement la destination du vol (Lyon) et la personne et l'heure du rendez-vous (Paul et 13 h 00).

Après que le service Compréhension linguistique a créé un modèle de Machine Learning fondé sur les exemples que vous lui avez donnés, il peut extraire des informations du langage naturel que les utilisateurs lui soumettent.

```
// À essayer : créer une  
application dans le portail LUIS
```

Custom Vision

Avec le [service Custom Vision](#), vous pouvez détecter des informations dans les images sur la base de vos propres données d'apprentissage. Custom Vision fonctionne de manière similaire aux autres Cognitive Services en cela qu'il est accompagné d'un algorithme de Machine Learning prédéfini. Tout ce que vous avez à faire est d'alimenter le service en données.

Imaginons que vous souhaitiez créer un modèle capable de détecter les types de nuages de pluie dans le ciel (cumulus ou stratus) à partir d'une image. Pour créer ce modèle, vous chargez des images de différents types de nuages sur le portail

Custom Vision et leur affectez des étiquettes pour indiquer au service comment entraîner le modèle. Dans cet exemple, vous étiquetez ainsi telle ou telle image avec le mot « cumulus » ou « stratus ».

Une fois que vous avez chargé suffisamment d'images, vous pouvez entraîner votre modèle. Plus vous chargerez d'images étiquetées et pousserez l'apprentissage, plus votre modèle sera précis.

Une fois que votre modèle donne des résultats satisfaisants, vous pouvez l'utiliser en effectuant des appels à l'API Custom Vision et en l'alimentant de nouvelles images. Chaque fois que vous chargez une nouvelle image, le service vous dit s'il la reconnaît d'après les images précédemment chargées. La Figure 4-1 illustre à quoi ressemble le point de terminaison de l'API.

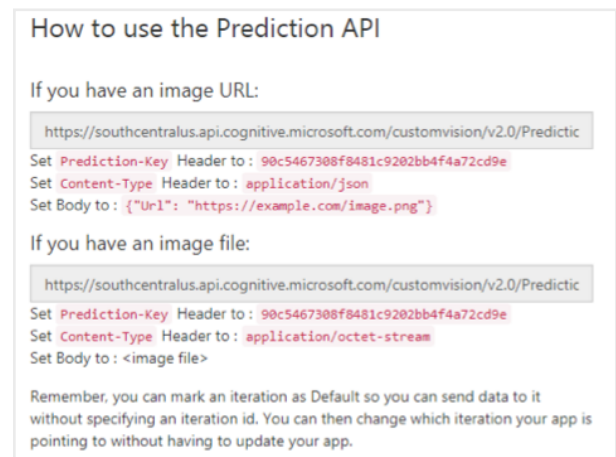


Figure 4-1

La détection d'informations par Custom Vision qui exploite votre propre modèle est déjà impressionnante. Mais Custom Vision va encore plus loin ! Le modèle créé en entraînant le service Custom Vision avec vos données peut être déployé sur la « périphérie intelligente ». Cela signifie que le modèle et l'API peuvent s'exécuter ailleurs que dans le cloud, par exemple sur un serveur local dans un conteneur Docker ou sur un autre appareil, comme votre téléphone.

Cela offre une grande souplesse car il n'est pas nécessaire d'avoir une connexion Internet active pour exploiter les possibilités du service Custom Vision ; vous pouvez également l'exécuter localement, avec d'excellentes performances. Par ailleurs, le modèle exécuté en périphérie n'est pas très volumineux (sa taille s'élève à quelques dizaines de mégaoctets), car vous ne déployez que le modèle et l'API, pas les données d'apprentissage.

```
// À essayer : créer votre propre  
projet Custom Vision
```

Indexeur vidéo

Le service [Indexeur vidéo](#) analyse les fichiers audio et vidéo que vous y chargez. Ce service cognitif fait également partie de la suite [Media Analytics](#) d'[Azure Media Services](#). Il fournit un algorithme de Machine Learning prédéfini. De votre côté, vous fournissez les données.

Parmi ses [nombreuses fonctionnalités](#), l'Indexeur vidéo a notamment les suivantes :

- La transcription du texte d'une vidéo. Vous pouvez affiner la transcription manuellement et apprendre à l'Indexeur vidéo à reconnaître les termes spécifiques d'un secteur, tels que « DevOps ».
- Le suivi des visages et l'identification des personnages d'une vidéo à un moment donné. L'Indexeur vidéo peut faire la même chose pour l'audio : il reconnaît alors quelle personne parle à chaque instant.
- La reconnaissance du texte visuel d'une vidéo, comme le texte d'une diapositive, et son intégration dans la transcription ;

- L'analyse de sentiment, qui identifie si ce qui est dit ou montré est positif, négatif ou neutre ;

Comme le montre l'étendue de ces fonctionnalités, l'Indexeur vidéo combine de nombreux services cognitifs, tels que la [reconnaissance vocale](#) et la [reconnaissance d'interlocuteur](#). Ensemble, ces services offrent de puissantes fonctionnalités qui rendent les contenus plus faciles à trouver, plus accessibles et plus utiles.

Vous pouvez charger des fichiers multimédias dans l'Indexeur vidéo au travers du portail Indexeur vidéo ou de l'API. La Figure 4-2 montre les résultats d'une vidéo [Azure Friday](#) qui a été chargée dans le service Indexeur vidéo.

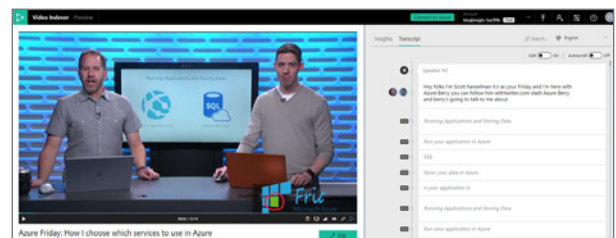


Figure 4-2

Comme le montre l'illustration, L'Indexeur vidéo a créé une transcription du son de la vidéo. Vous pouvez modifier la transcription et même la traduire dans d'autres langues. Vous pouvez aussi voir que l'Indexeur vidéo a reconnu le texte présent sur la diapositive derrière les intervenants et l'a marqué comme « SQL ». Vous pouvez accéder à ce texte en cliquant dessus. L'Indexeur vidéo fournit cette fonctionnalité pour d'autres applications grâce à l'intégration du widget [Cognitive Insights](#).

```
// À essayer : charger votre  
première vidéo dans l'Indexeur  
vidéo
```

QnA Maker

Le [service cognitif QnA Maker](#) offre un moyen simple de créer une couche conversationnelle par-dessus les données existantes, telle que des pages de questions fréquentes, des sites Web de support et des manuels d'utilisation de produits. QnA Maker vous aide à analyser et extraire les informations et à les convertir en paires question-réponse faciles à gérer. En termes simples, QnA Maker vous permet de créer des applications capables de renseigner vos utilisateurs selon un mode conversationnel.

Avec QnA Maker, il est possible de créer et gérer des bases de connaissances à l'aide du [portail QnA Maker](#) convivial ou des API REST. Nous avons simplifié le processus de création de robots en vous permettant de créer facilement un robot à partir de votre base de connaissances (sans aucun code ou modification de paramètres). Vous trouverez plus d'informations ici : [Créer un bot QnA](#). Vous pouvez bien sûr utiliser QnA Maker pour créer un bot à l'aide d'[Azure Bot Service](#) et augmenter votre bot QnA en y ajoutant le service [Compréhension linguistique](#). Pour plus de personnalisation, vous pouvez [ajouter une fonctionnalité de discussion](#) à votre bot, et suivre de petits scénarios de conversation prêts à l'emploi.

Vous [ne payez que l'hébergement de QnA Maker](#), pas à hauteur du nombre de fois que les utilisateurs interrogent le service qui en résulte.

```
// À essayer : créer votre propre
service de base de connaissances
avec QnA Maker
```

Suggestions automatiques Bing

Le service de suggestion [automatique Bing](#) assure la suggestion de texte au fil de la frappe. Cela vous permet d'offrir à vos utilisateurs une expérience de recherche similaire aux recherches Bing ou Google, dans lesquelles les termes de recherche sont automatisés ou complétés.

Saisissez le texte de la recherche caractère par caractère au service de suggestion automatique Bing ; celui-ci renvoie rapidement des suggestions de recherche au format JSON.

Par exemple, lorsque vous entrez le texte de recherche « Que dois-je rechercher », le service retourne le JSON suivant (voir une illustration plus grande en page 59) :

```
{
  "type": "Suggestions",
  "instrumentation": null,
  "queryContext": {
    "originalQuery": "what should I search for"
  },
  "suggestionGroups": [
    {
      "name": "Web",
      "searchSuggestions": [
        {
          "url": "https://www.bing.com/search?q=what+should+I+search+for&FORM=USBAPI",
          "urlPingSfx": null,
          "displayText": "what should I search for",
          "query": "what should I search for",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+I+search+for+on+bing&FORM=USBAPI",
          "urlPingSfx": null,
          "displayText": "what should I search for on bing",
          "query": "what should I search for on bing",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+I+search+for+on+the+internet&FORM=USBAPI",
          "urlPingSfx": null,
          "displayText": "what should I search for on the internet",
          "query": "what should I search for on the internet",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+I+search+for+today&FORM=USBAPI",
          "urlPingSfx": null,
          "displayText": "what should I search for today",
          "query": "what should I search for today",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+I+search+for+in+dna+raw+data&FORM=USBAPI",
          "urlPingSfx": null,
          "displayText": "what should I search for in dna raw data",
          "query": "what should I search for in dna raw data",
          "searchKind": "WebSearch"
        }
      ]
    }
  ]
}
```

Il contient toutes les suggestions. La requête de recherche originale y apparaît en tête des résultats.

```
// À essayer : recevoir une clé
d'API et tester les suggestions
automatiques Bing gratuitement
```

```
{
  "_type": "Suggestions",
  "instrumentation": null,
  "queryContext": {
    "originalQuery": "what should I search for"
  },
  "suggestionGroups": [
    {
      "name": "Web",
      "searchSuggestions": [
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for",
          "query": "what should i search for",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for+on+bing&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for on bing",
          "query": "what should i search for on bing",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for+on+the+internet&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for on the internet",
          "query": "what should i search for on the internet",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for+today&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for today",
          "query": "what should i search for today",
          "searchKind": "WebSearch"
        },
        {
          "url": "https://www.bing.com/search?q=what+should+i+search+for+in+dna+raw+data&FORM=USBAPI",
          "urlPingSuffix": null,
          "displayText": "what should i search for in dna raw data",
          "query": "what should i search for in dna raw data",
          "searchKind": "WebSearch"
        }
      ]
    }
  ]
}
```

Azure Bot Service

La création d'un bot, une application qui interagit automatiquement et de manière autonome avec les utilisateurs, n'est pas une tâche futile. Vous devez effectuer le suivi du contexte de votre interaction avec chaque utilisateur et être prêt à réagir face à une multitude de paramètres d'interaction possibles.

[Azure Bot Service](#) vous permet de développer des expériences et des bots intelligents, de classe entreprise pouvant étendre votre marque et vous permettre de maintenir vos données sous contrôle. Commencez par un bot simple de questions et de réponses ou développez un assistant virtuel sophistiqué. Utilisez des outils et un kit de développement logiciel (SDK) open-source complets afin de vous connecter facilement à votre bot sur des appareils et des canaux populaires. Permettez à votre bot de parler, d'écouter et de comprendre vos utilisateurs au moyen d'une intégration native sur des Cognitive Services.

Azure Bot Service permet de créer facilement un bot et fournit le support suivant :

- Offre un moyen d'héberger et de gérer des bots que vous avez développés à l'aide de [Microsoft Bot Framework](#), avec [des outils et un kit de développement logiciel \(SDK\) open-source](#) complets pour le développement de bot.
- S'intègre en mode natif avec Cognitive Services.
- Vous aide à connecter votre bot là où se trouvent les clients, à l'aide de connecteurs sur des canaux tels que Facebook, Slack, Microsoft Teams, Line, Telegram, entre autres.
- Présente tous les avantages d'un service géré dans Azure, dont une évolutivité impressionnante et la livraison continue intégrée, et vous payez uniquement pour ce que vous utilisez.

Avec Azure Bot Service, vous pouvez par exemple développer un bot qui fournit aux utilisateurs les réponses à leurs questions les plus fréquentes.

Vous pouvez l'utiliser avec le service cognitif [QnA Maker](#). L'interface du bot peut être une fenêtre de chat située sur votre site web. Vous pourriez également développer un assistant virtuel de marque propre à l'aide de l'[accélérateur de solution d'assistant virtuel](#).

```
// À essayer : prise en main  
de Chatbot en utilisant  
Azure Bot Service.
```

Azure Machine Learning Studio

Vous pouvez ajouter des fonctions intelligentes à vos applications grâce aux services Azure, tels que Cognitive Services. Elles reposent sur des algorithmes de Machine Learning créés par Microsoft pour être utilisés en tant que services. Il existe cependant bien d'autres façons d'utiliser le Machine Learning dans vos applications. Commençons par expliquer en quoi consiste le Machine Learning.

Qu'est-ce que le Machine Learning ?

On assimile souvent le Machine Learning à l'intelligence artificielle (IA), mais ce sont deux choses différentes. L'IA implique des machines capables d'effectuer des tâches caractéristiques de l'intelligence humaine. L'IA peut aussi être installée à l'aide du Machine Learning, en complément d'autres techniques.

Le Machine Learning en soi est un domaine de l'informatique qui permet aux ordinateurs d'apprendre sans programmation explicite. Il est possible de déployer le Machine Learning au moyen d'une ou de plusieurs technologies algorithmiques, telles que les réseaux neuronaux, le Deep Learning et les réseaux bayésiens.

Sur quoi repose le Machine Learning ?

La Figure 4-3 montre le workflow de base pour l'utilisation du Machine Learning.

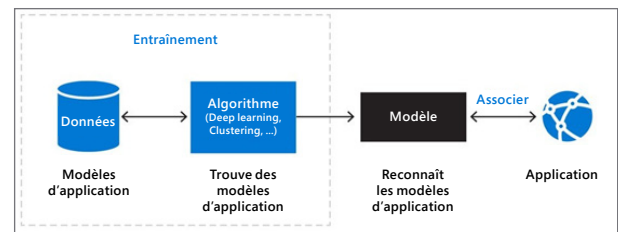


Figure 4-3

Le processus du Machine Learning se déroule comme suit :

- Les données contiennent des modèles d'application. Parfois vous connaissez probablement certains de ces modèles d'application, tels que les habitudes de commande des utilisateurs. Mais vos données recèlent aussi très certainement des modèles d'application que vous ignorez.
- L'algorithme de Machine Learning est le composant logiciel intelligent qui est capable de déceler des modèles d'application dans les données. Vous pouvez créer cet algorithme par vous-même à l'aide de techniques comme le Deep Learning ou l'apprentissage supervisé.
- Le fait de déceler des modèles d'application dans les données à l'aide d'un algorithme de Machine Learning est également désigné sous le terme suivant : « entraînement d'un modèle de Machine Learning ». L'entraînement aboutit à un modèle de Machine Learning. Il contient les enseignements tirés de l'algorithme de Machine Learning.
- Des applications utilisent le modèle en l'alimentant en nouvelles données et en exploitant ses résultats. Les nouvelles données sont analysées en suivant les modèles d'application trouvés dans les données. Par exemple, lorsque vous entraînez un modèle de Machine Learning à reconnaître les chiens dans les images, il doit être capable de reconnaître dans une image un chien qu'il n'a jamais vu auparavant.

Le point crucial de ce processus est son caractère itératif. Le modèle de Machine Learning s'améliore constamment en s'entraînant sur les nouvelles données et en ajustant son algorithme ou en lui apprenant à distinguer les résultats corrects des résultats erronés.

Utilisation d'Azure Machine Learning Studio pour créer des modèles

Vous pouvez utiliser [Azure Machine Learning Studio](#) pour créer vos propres modèles de Machine Learning et les rendre accessibles au travers de services web pour que vos applications puissent les utiliser.

Machine Learning Studio est un service d'Azure avec lequel vous pouvez créer visuellement des projets et des expériences de Machine Learning, coupler des jeux de données, créer des blocs-notes et rendre les modèles accessibles à l'aide de services web. Le studio lui-même est un portail que vous pouvez utiliser depuis votre navigateur web et qui vous permet de créer des algorithmes en utilisant une approche de glisser-déposer.

Dans le studio, vous pouvez démarrer de zéro ou en partant de l'une des nombreuses expériences qui se trouvent dans la [galerie](#), dont l'une sert à prédire la durée des séjours hospitaliers et l'autre à détecter les anomalies dans les flux de données en temps réel. Utilisez ces expériences comme base d'un modèle de Machine Learning ou pour découvrir comment résoudre ces cas.

Une expérience de Machine Learning dans Machine Learning Studio se déroule en plusieurs étapes qui manipulent des données et les soumettent à des algorithmes de Machine Learning. Utilisez des étapes prédéfinies disponibles dans le studio pour composer un algorithme de Machine Learning.



Figure 4-4

La figure 4-4 présente une expérience dans Machine Learning Studio, avec le flux de travail à exécuter pour entraîner un modèle, ainsi que les catégories d'étapes prédéfinies pouvant être utilisées dans le flux de travail.

Lorsque vous avez construit vos expériences et les avez appliquées à vos données pour créer un modèle de Machine Learning, vous pouvez les publier comme services web. Si vos applications utilisent les services web, elles peuvent envoyer des données à votre modèle et en recevoir les prédictions.

// À essayer : [inscription pour utiliser Azure Machine Learning Studio](#)

Azure Machine Learning service

Vous pouvez utiliser [Azure Machine Learning service](#) pour créer des algorithmes d'Analytique données avec des outils open-source tels que Python et Azure CLI. Tout comme avec Machine Learning Studio, vous pouvez créer l'algorithme de votre choix, offrant de la flexibilité pour divers scénarios, tels que l'analyse prédictive, les recommandations de données et la classification des données.

Azure Machine Learning service vous permet de créer de bout en bout des algorithmes de Machine Learning personnalisés. C'est différent de Machine Learning Studio qui vous permet de créer visuellement un algorithme en connectant des éléments prédéfinis d'un algorithme de Machine Learning. Azure Machine Learning service prend entièrement en charge des technologies open-source telles que Google [TensorFlow](#), [PyTorch](#) et [Scikit-learn](#).

Azure Machine Learning service est un service complet qui propose des fonctionnalités de bout en bout. Vous pouvez créer votre algorithme, préparer vos données, entraîner l'algorithme pour ces dernières, tester et déployer l'algorithme, puis le suivre et le gérer lorsqu'il est en cours d'exécution.

Azure Machine Learning service fonctionne avec de nombreux services Azure qui peuvent aider à créer, entraîner et exécuter votre algorithme. Vous pouvez, par exemple, créer votre algorithme dans Jupyter Notebook, l'entraîner au moyen d'[Azure Databricks](#) et le déployer sur un cluster de conteneur Kubernetes dans [Azure Kubernetes Service](#).

```
// Prise en main d'Azure Machine  
Learning service en utilisant  
le portail Azure
```

Outils de développement pour l'IA

Visual Studio Tools for AI

[Visual Studio Tools for AI](#) est une extension gratuite de Visual Studio. Utilisez-la pour accéder à une gamme de services et d'infrastructures d'IA, notamment [Microsoft Cognitive Toolkit \(CNTK\)](#), [TensorFlow](#), [Keras](#) et [Caffe2](#).

Visual Studio Tools for AI vous permet de créer des algorithmes de Machine Learning de manière comparable à Azure Machine Learning Studio. Vous pouvez utiliser des langages comme Python, C, C++ et C#, ou tirer parti de l'un des nombreux exemples de la [galerie](#) d'exemples d'expériences de Machine Learning.

Visual Studio Tools for AI vous permet de créer des éléments de Machine Learning à partir de Visual Studio pour tirer parti de la puissance de Visual Studio afin de déboguer les algorithmes et d'entraîner vos modèles de Machine Learning. Dans Visual Studio, vous pouvez créer des tâches d'entraînement qu'il est possible de déployer sur de nombreuses machines virtuelles dans Azure. Vous pouvez également observer les performances de l'entraînement, puis générer un service web qui utilise le modèle de Machine Learning dans vos applications. Vous pouvez faire tout cela sans jamais quitter Visual Studio.

```
// À essayer : téléchargement de  
l'extension Visual Studio Tools  
for AI
```

Boîte à outils IA pour Azure IoT Edge

L'utilisation de modèles de Machine Learning localement sur des appareils (la périphérie intelligente) offre un puissant atout : elle vous permet d'utiliser la puissance de traitement locale de l'appareil sans vous appuyer sur une connexion Internet ou vous exposer à la latence d'un appel de service web pour obtenir vos résultats.

Décrit précédemment dans la section [Cognitive Services](#), le [Service Custom Visio](#) prend déjà en charge l'exécution en périphérie. Vous pouvez vous attendre à ce que davantage de services offrent cette possibilité à l'avenir.

Pour exécuter des modèles de Machine Learning en périphérie, il vous faut des outils de déploiement des modèles et des services web. La boîte à outils IA pour Azure IoT Edge vous aide en vous permettant d'empaqueter vos modèles de Machine Learning dans des conteneurs Docker compatibles avec Azure IoT Edge et de les mettre à disposition sous forme d'API REST.

La boîte à outils IA pour Azure IoT Edge contient des exemples pour démarrer. Elle est complètement open source et [disponible sur GitHub](#).

IA et réalité mixte

Vos applications ne se limitent plus à un environnement 2D. Le monde est désormais le canevas de votre application, appuyé par une intelligence spatiale d'éléments tels que des capteurs IoT, une réalité mixte et une vision par ordinateur. Avec les services Azure de réalité mixte, vous pouvez donner vie à des données en 3D quand et où vos utilisateurs en ont besoin.

Azure Spatial Anchors

Dans l'univers de la réalité mixte, vous pouvez intégrer des informations numériques dans le contexte de votre environnement physique, tel un hologramme des personnages de votre jeu préféré sur le plan de travail de votre cuisine. [Azure Spatial Anchors](#) vous permet de placer du contenu numérique dans un emplacement physique et de le partager avec des utilisateurs en utilisant votre gamme d'appareils et de plateformes.

Par exemple, en pénétrant dans un hôpital, les gens ont souvent du mal à trouver leur chemin. Avec Azure Spatial Anchors, l'hôpital peut créer une application mobile qui affiche des informations numériques dans l'hôpital physique afin de guider les personnes vers divers emplacements. Dans l'application sur leur appareil iOS, les individus peuvent utiliser les flèches de direction sur les tableaux d'information physiques de l'hôpital pour atteindre leurs destinations.

Azure Spatial Anchors est également utilisé dans une application de formation pour le personnel infirmier. [Pearson Education](#) a permis à des élèves-infirmiers et des professeurs de pratiquer le diagnostic et le traitement de patients en 3D avant d'assumer la pression d'un cas réel. Les étudiants et les professeurs peuvent utiliser des appareils HoloLens ou des téléphones mobiles et des tablettes exécutant iOS ou Android.

Azure Spatial Anchors vous permet de partager des informations numériques et des hologrammes qui sont positionnés dans le monde physique. Il fonctionne avec des applications s'appuyant sur Unity, ARKit, ARCore et Universal Windows Platform (UWP) et peut être utilisé avec un appareil HoloLens, des appareils basés sur iOS prenant en charge ARKit et des appareils basés sur Android prenant en charge ARCore.

Avec Azure Spatial Anchors, vous pouvez facilement sécuriser vos données spatiales et permettre aux utilisateurs d'y accéder via Azure Active Directory. Vous pouvez également intégrer dans votre application spatiale des services IoT, d'analyse de données, d'IA et de stockage.

```
// Commencez par partager  
Azure Spatial Anchors entre  
les sessions et les appareils
```

Azure Remote Rendering

Lorsque vous utilisez des modèles 3D dans des scénarios tels que les révisions de conception et les plans de procédures médicales, vous avez besoin qu'ils soient aussi précis que possible. Chaque détail compte.

De nombreuses entreprises utilisent des modèles 3D complexes contenant des centaines de millions de polygones, et les appareils de périmètre dotés d'une puissance de traitement graphique faible à moyenne ne sont pas capables de les restituer. Traditionnellement, les développeurs ont tenté de gérer ce problème en s'appuyant sur la technique dite de « décimalisation ». Cela simplifie le modèle en supprimant les polygones afin de permettre l'affichage sur ces appareils.

Mais cette perte de détails sacrifie les informations nécessaires pour prendre la bonne décision dans de nombreuses situations. Avec [Azure Remote Rendering](#), les modèles 3D sont restitués dans le cloud et ils sont diffusés sur des appareils en temps réel, sans mettre à mal la qualité visuelle.

Cela vous permet de conserver la qualité d'origine du modèle et d'interagir avec le contenu sur les appareils de périmètre tels que les casques et les téléphones mobiles, en conservant chaque détail.

Utilisation des événements et des messages dans votre application

Les applications modernes, distribuées à l'échelle mondiale, doivent souvent gérer de grandes quantités de messages entrants et doivent donc être conçues en gardant à l'esprit les notions de découplage et d'évolutivité. Azure fournit plusieurs services pour aider en matière d'assimilation et d'analyse d'événements, ainsi que de modèles d'application de messagerie. Ces services sont également essentiels pour créer des applications intelligentes qui tirent parti de l'IA.

Azure Service Bus

Dans Azure, le cœur de la messagerie est [Azure Service Bus](#). Service Bus englobe un ensemble de services que vous utilisez pour les modèles d'application de messagerie. Les principaux services sont les files d'attente et les rubriques Azure Service Bus.

```
// Mise en route des files  
d'attente Azure Service Bus
```

Files d'attente Azure Service Bus

Les files d'attente [Azure Service Bus](#) découplent les systèmes les uns des autres. Par exemple, une application web reçoit des commandes d'utilisateurs et doit invoquer un service web pour traiter les commandes. Il va falloir trop longtemps au service web, peut-être jusqu'à cinq minutes, pour traiter la commande.

Une façon de résoudre ce problème consiste à utiliser une file d'attente pour découpler l'application web du service web. L'application web reçoit la commande et l'écrit dans un message d'une file d'attente Service Bus. Pui l'application web informe l'utilisateur que la commande est en cours de traitement. Le service web prend les messages de la file d'attente, un par un, et les traite. Lorsque le service web a traité une commande, il envoie une notification par e-mail à l'utilisateur indiquant que l'article a été commandé.

En découplant les systèmes, l'application web peut fonctionner à une vitesse différente de celle du service web, et les deux peuvent évoluer individuellement, en fonction des besoins de l'application.

Une file d'attente Service Bus est un mécanisme simple. Plusieurs applications peuvent placer des messages dans la file d'attente, mais un message de la file d'attente peut uniquement être traité par une application à la fois. Il existe des fonctionnalités astucieuses permettant de travailler avec des messages dans la file d'attente, comme la détection des doublons et une sous-file d'attente de lettres mortes vers laquelle les messages sont déplacés lorsqu'ils ne parviennent pas à être traités correctement.

Rubriques Azure Service Bus

Tout comme les files d'attente Service Bus, [les rubriques Azure Service Bus](#) constituent une forme de découplage d'applications.

Voici ce qui les différencie :

- Avec une file d'attente, plusieurs applications écrivent des messages dans la file d'attente, mais une seule *application* à la fois est en mesure de traiter un message.
- Avec une rubrique, plusieurs applications écrivent des messages dans la rubrique, et plusieurs *applications* peuvent traiter un message en même temps.

Les applications peuvent créer un abonnement à la rubrique indiquant quel type de messages les intéressent. Tout comme les files d'attente, les rubriques sont dotées de fonctions comme la détection des doublons et d'une sous-file d'attente lettre morte vers laquelle les messages sont déplacés lorsqu'ils ne parviennent pas à être traités correctement.

Comparaison des files d'attente Service Bus et du Stockage File d'attente Azure

Les files d'attente Service Bus et le stockage File d'attente Azure ont fondamentalement les mêmes fonctions, mais il existe des différences, comme illustré dans le tableau 4-3.

Tableau 4-3

Files d'attente Azure Service Bus	Stockage File d'attente Azure
Durée de vie d'un message > 7 jours	Durée de vie d'un message < 7 jours
Classement (premier entré, premier sorti) garanti	Taille de la file d'attente > 80 Go
Détection des doublons	Journaux des transactions
Taille des messages ≤ 1 Mo	Taille des messages ≤ 64 Ko

Azure Event Hubs

[Azure Event Hubs](#) peut aider les entreprises à capturer des quantités massives de données afin de les analyser ou de les transformer et de les déplacer pour une utilisation ultérieure.

Event Hubs est conçu pour assimiler des données en volume massif. Ce service gère sans effort des millions de messages par seconde. Il conserve les messages jusqu'à sept jours ou indéfiniment en les écrivant dans un magasin de données grâce à la fonction Event Hubs Capture.

Vous pouvez utiliser Event Hubs pour filtrer les données avec des requêtes, au fur et à mesure qu'elles entrent, et les envoyer vers un magasin de données comme Azure Cosmos DB. Vous pouvez même relire les messages.

// À essayer : début d'envoi de messages à Azure Event Hubs

Azure IoT Hub

Tout comme Event Hubs, [Azure IoT Hub](#) est conçu pour l'assimilation massive de données. Il est spécifiquement destiné à gérer l'énorme volume de messages de données provenant d'appareils de l'Internet des objets (IoT), comme les thermostats et les capteurs intelligents des voitures.

Il dispose d'un grand nombre des mêmes propriétés qu'Event Hubs, comme la capacité de conserver les messages jusqu'à sept jours et de les relire.

Ce qui rend IoT Hub unique, c'est qu'il peut envoyer des messages aux appareils. Il a la capacité de gérer l'intégralité de votre infrastructure IoT. Vous pouvez l'utiliser pour inscrire des appareils, générer des rapports sur leur état, les gérer en les sécurisant et les redémarrer, ainsi qu'envoyer des données vers ces derniers.

// À essayer : connexion de votre
appareil à votre hub IoT

Azure Event Grid

[Azure Event Grid](#) propose un type de messagerie différent, un service de publication et d'abonnement entièrement géré, compatible avec presque tous les services dans Azure, ainsi que des services de publication et d'abonnement personnalisés.

Cette méthode est différente du travail avec les files d'attente et les rubriques Service Bus, pour lesquelles vous devez interroger la file d'attente et la rubrique afin de prendre connaissance des nouveaux messages. Event Grid transmet automatiquement les messages aux abonnés, ce qui en fait un service d'événements réactif, en temps réel.

Les services internes et externes à Azure publient des événements lorsqu'un nouveau blob est ajouté, par exemple, ou qu'un nouvel utilisateur est ajouté à un abonnement Azure. Azure Event Grid détecte ces événements et les met à la disposition des gestionnaires d'événements et des services qui s'abonnent aux événements, comme illustré à la Figure 4-5.

Les gestionnaires d'événements peuvent être des éléments comme Azure Functions ou Azure Logic Apps, qui peuvent alors agir sur les données de l'événement.

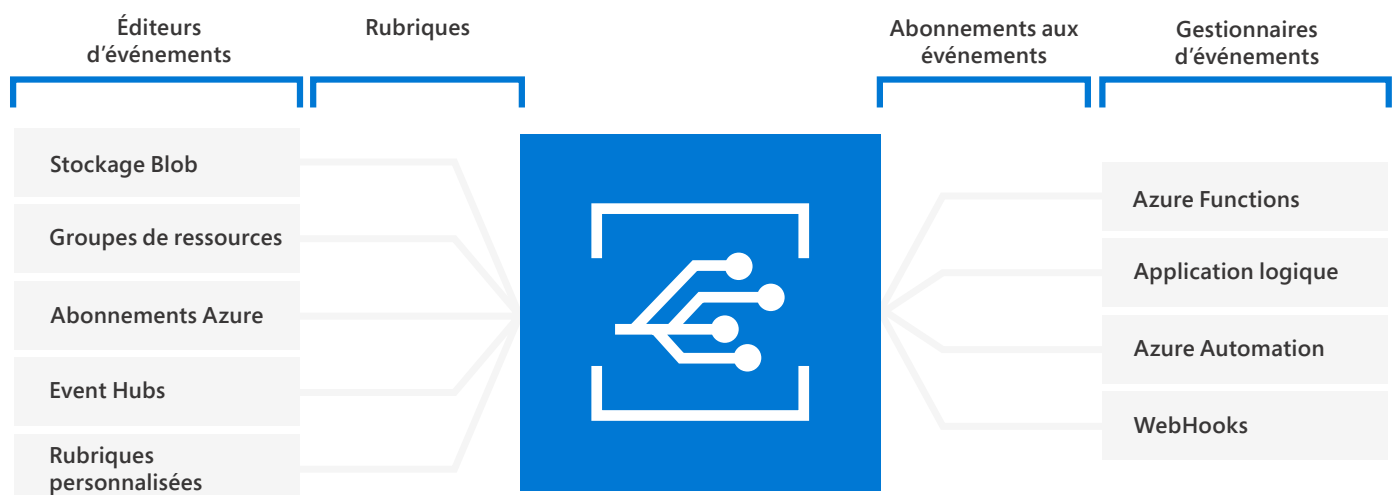


Figure 4-5

Un autre aspect important d'Event Grid est qu'il s'agit d'un service sans serveur. Cela signifie qu'à l'instar d'Azure Logic Apps et d'Azure Functions, Event Grid évolue automatiquement et ne nécessite pas le déploiement d'une instance. Il vous suffit de le configurer et de l'utiliser, et vous payez uniquement lorsque vous l'utilisez.

Vous pouvez utiliser Azure Event Grid si vous souhaitez être notifié par e-mail chaque fois qu'un utilisateur est ajouté à votre liste de diffusion dans MailChimp, ou en est supprimé. Azure Event Grid permet d'activer une application dans Azure Logic Apps et est configuré pour écouter les modifications apportées à la liste de diffusion MailChimp. Azure Event Grid signale ensuite à Logic Apps d'envoyer un e-mail contenant le nom d'un utilisateur qui a été ajouté ou supprimé et l'action effectuée.

```
// À essayer : surveillance  
des modifications apportées  
aux machines virtuelles avec  
Azure Event Grid et Logic Apps
```

Azure SignalR Service

Vous pouvez utiliser Azure SignalR Service pour ajouter des fonctionnalités web en temps réel à vos applications. Ce service est basé sur ASP.NET Core SignalR et il est proposé comme un service autonome et entièrement géré dans Azure.

SignalR peut actualiser en temps réel les applications connectées par protocole HTTP, sans que les applications doivent s'enquérir des mises à jour ou envoyer de nouvelles requêtes HTTP. Cela vous permet de créer des expériences web fluides où les informations sont mises à jour à la volée. Par exemple, une application d'enchères peut utiliser SignalR pour actualiser la dernière enchère dès que celle-ci est déposée, sans complètement rafraîchir la page ou demander sans cesse les nouvelles informations.

Héberger vous-même un serveur SignalR n'est pas une tâche anodine, car il peut être difficile à dimensionner et à sécuriser correctement. En utilisant Azure SignalR Service entièrement géré, l'installation est facile et tous les aspects critiques (sécurité, disponibilité, performance et évolutivité) sont gérés pour vous.

```
// À essayer : création d'une  
salle de conversation avec  
SignalR
```

Qu'utiliser, et quand ?

Azure fournit de nombreuses options permettant de créer des applications de messagerie et de découplage. Laquelle devez-vous utiliser, et quand ? Le Tableau 4-4 résume les différences pour vous aider à choisir.

Autres lectures

Cet Ebook gratuit vous fournit des informations complémentaires sur l'utilisation des services Azure AI dans votre application :

// [Guide de création d'applications d'intelligence artificielle à l'intention des développeurs](#)

Tableau 4-4

	SignalR Service*	Event Grid*	Event Hubs*	IoT Hub*	Rubriques*	Files d'attente Service Bus*	Stockage File d'attente Azure*
Assimilation d'événements		●	●	●			
Gestion des appareils				●			
Messagerie	●	●	●	●	●	●	●
Plusieurs clients	●	●	●	●	●		
Plusieurs expéditeurs	●	●	●	●	●	●	●
Utilisation pour le découplage			●	●	●	●	●
Utilisation pour publication/abonnement	●	●					
Taille de message maximale	64 Ko	64 Ko	256 Ko	256 Ko	1 Mo	1 Mo	64 Ko

* Les services marqués d'un astérisque comportent un niveau gratuit utilisable pour démarrer sans frais.

05 /

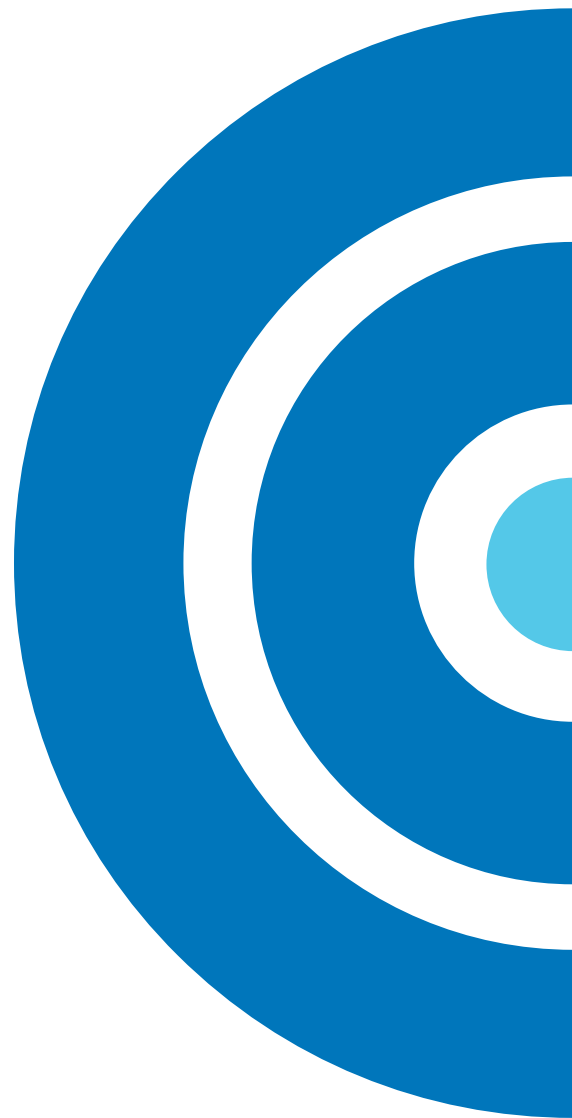
Votre entreprise connectée avec IoT

Comment Azure peut-il connecter, sécuriser, gérer, surveiller et contrôler vos appareils dans le cloud ?

Une des choses les plus passionnantes que vous êtes amené à faire en tant que développeur est d'avoir un impact sur le monde réel. Vous pouvez le faire à l'aide d'appareils tels que des robots, des capteurs et des microcontrôleurs. Lorsque vous créez des applications avec ces appareils, vous pouvez prédire quand une machine aura besoin d'une opération de maintenance avant de véritablement nécessiter des réparations, ou même créer une voiture autonome.

Il peut être difficile de travailler avec ces appareils pour avoir un impact sur le monde réel si vous développez l'intégralité du logiciel par vous-même. Heureusement, Azure offre des solutions qui peuvent considérablement vous simplifier la tâche. Vous pouvez par exemple utiliser [Azure IoT Hub](#) pour assimiler en toute sécurité des messages de capteurs et réaliser la gestion des appareils, comme envoyer des messages aux appareils et les réinitialiser. Lorsque vous développez quelque chose avec un microcontrôleur (ordinateurs monopuce) intégré, envisagez d'utiliser [Azure Sphere](#), qui est une combinaison de matériel, de SE et de services cloud avec la sécurité intégrée dans les circuits.

Consultons les services d'Azure susceptibles de vous aider à développer de magnifiques applications IoT.



Azure IoT Hub

Au cœur de l'IoT Azure se trouve [Azure IoT Hub](#), un PaaS cloud ouvert et flexible qui connecte, surveille et gère appareils d'une manière évolutive et sécurisée. Nous avons déjà évoqué Azure IoT Hub dans un chapitre précédent, mais il y a encore beaucoup à apprendre sur le sujet dans le contexte de l'IoT.

Vous savez déjà que vous pouvez utiliser IoT Hub pour assimiler d'énormes quantités de messages qui émanent généralement d'appareils IoT, tels que les messages qui contiennent des données provenant de capteurs de température. Le caractère unique d'IoT Hub réside dans le fait qu'il ne se contente pas de recevoir les messages, mais qu'il renvoie aussi de commandes aux appareils. Il établit une communication bidirectionnelle avec les appareils et vous permet même d'exécuter du code sur ceux-ci.

IoT Hub est utile parce qu'il vous permet de gérer les appareils de diverses manières, par exemple, en envoyant un message pour leur demander de redémarrer ou en exécutant un script de démarrage. Cela fait d'IoT Hub le service central qui rend possible l'implémentation d'une application IoT robuste dans Azure. Pour apporter des appareils compatibles, Azure fournit le [Service IoT Hub Device Provisioning](#).

Ce service permet la mise en service sans contact et en temps utile de l'IoT Hub adéquat sans aucune intervention humaine, ce qui vous permet d'approvisionner les appareils de manière évolutive et sécurisée. Il peut vous aider dans de nombreux scénarios d'approvisionnement d'appareil, y compris pour la connexion d'appareils à un IoT Hub et l'exécution de leurs scripts de configuration initiale, pour l'équilibrage de charge des appareils sur plusieurs hubs et pour le réapprovisionnement lors du remplacement d'un appareil.

IoT Hub peut également se connecter aux appareils pouvant exécuter des charges de travail dans le Cloud, dont celle qui exécutent le runtime et les modules Azure IoT Edge. Une fois qu'un appareil est connecté à IoT Hub, le hub possède un enregistrement de son identité. Cela permet à IoT Hub d'envoyer des messages et de surveiller l'appareil, ainsi que de sécuriser l'appareil et les communications entre eux. Les appareils doivent s'authentifier auprès d'IoT Hub conformément à plusieurs protocoles de sécurité répondant aux bonnes pratiques sectorielles, tels que les certificats X.509 et l'authentification basée sur les jetons SAS. Vous pouvez gérer la sécurité de chaque appareil connecté et révoquer des privilèges si vous ne voulez plus qu'un appareil en particulier soit connecté.

Lorsque les appareils envoient des messages à Azure IoT Hub, vous pouvez soit stocker les messages, soit les acheminer vers un autre service aux fins d'analyse ou d'action. Il est par exemple possible de router les messages entrants en utilisant le routage de message IoT Hub qui offre simplicité, fiabilité et évolutivité, sans la complexité du développement de solutions de routage personnalisées.

Une autre option pour l'intégration d'événements IoT dans les applications de gestion ou les services Azure consiste à utiliser Azure Event Grid, une solution de routage d'événement complètement managée qui utilise un modèle de publication et d'abonnement. IoT Hub et Event Grid collaborent pour intégrer des événements IoT Hub dans Azure et des services hors Azure en quasi temps réel.

Vous pouvez également créer des tunnels de communication bidirectionnels à l'aide de flux d'appareil. Les flux d'appareil Azure IoT Hub facilitent la création de tunnels TCP bidirectionnels sécurisés pour divers scénarios de communication cloud-à-appareil.

Exemple : Azure IoT Hub

Une entreprise qui fournit des informations sur les déplacements et l'utilisation de camions développe son activité. Auparavant, l'entreprise suivait ses actifs à l'aide d'un code personnalisé dans une application native sur téléphone, laquelle appelait un service web personnalisé, et en interrogeant les dongles GPS qui équipaient les camions. Cette solution était compliquée à gérer en raison de la difficulté liée à l'approvisionnement de nouveaux actifs avec de nouveaux appareils. En outre, l'entreprise avait besoin d'inscrire un nouveau client possédant plus de 2 000 actifs.

Elle utilise à présent Azure IoT Hub pour la communication et la gestion des appareils. Elle peut

utiliser le Service IoT Hub Device Provisioning pour intégrer les 2 000 nouveaux appareils et les relier à un hub IoT spécifique pour ce client. L'application sur téléphone utilise maintenant Node.js et le [SDK pour appareils IoT Azure](#) pour interagir avec IoT Hub. Ce qui est important, c'est que la société contrôle maintenant la sécurité de ses appareils, peut détecter leur statut et, le cas échéant, les réinitialiser. En outre, la société achemine les données émises par ses dongles GPS au travers d'Azure Stream Analytics, de sorte que seules les modifications des données GPS sont conservées. Cela réduit la charge des données, car les dongles envoient leur localisation toutes les secondes.

Utiliser Azure IoT Hub a permis à cette entreprise de développer et de faire mûrir son activité en offrant une gestion des appareils et une sécurité de premier plan. Cela a également ouvert de nouvelles possibilités pour utiliser les appareils à un niveau auparavant inimaginable pour cette entreprise.

Azure IoT Central

[Azure IoT Central](#) est une plateforme de solutions IoT hébergée qui vous permet de créer des applications IoT riches simplement en parcourant des assistants.

Aucune rédaction de code ou configuration approfondie n'est nécessaire ; IoT Central fait tout cela pour vous, approvisionnant et configurant tout ce dont vous avez besoin, y compris Azure IoT Hub.

Vous obtenez les mêmes capacités que si vous aviez créé la solution à partir de zéro, mais sans devoir consacrer des années à sa programmation. Si vous souhaitez avoir davantage de contrôle sur certains aspects, vous pouvez toujours aller plus loin et modifier la solution en fonction de vos besoins.

Accélérateurs de solution Azure IoT

Les [accélérateurs de solution Azure IoT](#) sont un excellent point de départ pour le développement de votre solution IoT.

Ces modèles complets et personnalisables pour les scénarios IoT courants effectuent toutes les tâches : surveillance et sécurisation des appareils jusqu'à la mise à disposition d'une interface utilisateur. Ils aident également à connecter des appareils, nouveaux ou existants. La figure 5-1 présente un exemple de modèle de solution.

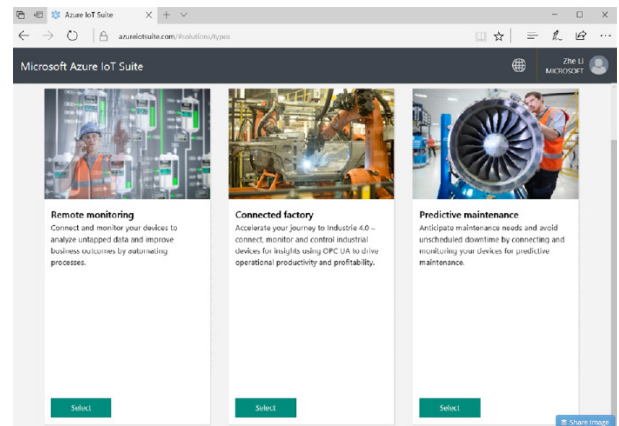


Figure 5-1

Connectez et surveillez vos appareils avec le pilotage à distance. Gagnez en visibilité sur vos appareils, ressources et capteurs, où qu'ils se trouvent. Recueillez et analysez les données de vos appareils en temps réel à l'aide d'un accélérateur de solution de pilotage à distance préconfiguré qui déclenche des alertes et des actions automatiques, depuis des diagnostics à distance jusqu'à des demandes de maintenance.

Il y a de nombreux autres accélérateurs de solution IoT, comme ceux qui améliorent l'efficacité industrielle avec une usine connectée, accroissent la fiabilité des équipements grâce à la maintenance prédictive et développent et testent une solution IoT à l'aide de la simulation d'appareil.

Azure IoT Edge

Dans les applications IoT modernes, le traitement des données peut être effectué dans le Cloud ou du côté de l'appareil. Le traitement du côté de l'appareil est appelé « edge computing » (traitement des données à la périphérie).

On utilise l'Edge computing quand on ne veut (ou ne peut) pas se fier à la connexion au cloud, quand on souhaite améliorer les performances d'une application en éliminant les allers et retours vers le cloud, ou quand on ne peut pas communiquer avec le cloud à partir de l'appareil pour des raisons de sécurité, de confidentialité ou de restrictions réglementaires.

Dans de tels scénarios, utilisez [Azure IoT Edge](#). Azure IoT Edge est géré depuis IoT Hub, ce qui vous permet de déplacer des éléments de votre charge de travail vers la périphérie. Cela réduit le temps que les appareils passent à envoyer des messages au cloud et permet des scénarios hors ligne ainsi que des réactions plus rapides aux changements de statut.

Azure IoT Edge se compose des éléments suivants :

- Les **modules IoT Edge** sont des conteneurs qui exécutent les services Azure, des services tiers ou votre propre code. Ils sont déployés sur les appareils de périmètre IoT et s'exécutent localement sur ceux-ci.
- Le **runtime IoT Edge** s'exécute sur chaque appareil IoT Edge et gère les modules déployés sur chaque appareil.
- **IoT Hub** expose des interfaces spécifiques pour surveiller et gérer à distance les appareils IoT Edge disponibles via le portail Azure, Azure CLI ou les kits de développement logiciel.

Ces trois composants fonctionnent conjointement sur les appareils et dans le Cloud pour exécuter vos charges de travail sur la périphérie intelligente.

Il est possible d'exécuter de nombreux services Azure en périphérie dans le cadre de certains scénarios. La liste des services disponibles ne cesse en outre de s'allonger. Le Tableau 5-1 en énumère seulement quelques-uns.

Tableau 5-1

Si vous voulez	Utilisez ceci sur Azure IoT Edge
Développer et déployer des modèles d'IA	Machine Learning
Personnaliser les modèles de vision informatisée pour votre cas d'utilisation	Service Custom Vision
Traiter des flux de données en temps réel	Stream Analytics
Traiter des événements à l'aide de code sans serveur	Functions
Déployer une base de données SQL Server en périphérie	Bases de données SQL Server
Respecter les normes d'interopérabilité 4.0 du secteur	OPC Unified Architecture
Développer une logique personnalisée	Module personnalisé

Dès que vous aurez commencé à utiliser Azure IoT Edge, vous pourrez créer des applications rapides qui exécutent localement des algorithmes de Machine Learning et renvoient instantanément leurs commentaires.

Azure Digital Twins

Dans l'univers de l'IoT, vous pouvez travailler avec divers capteurs et appareils IoT qui fonctionnent en connexion avec des individus et des objets. Pour comprendre pleinement les données capturées par les appareils IoT, vous devez comprendre l'environnement physique dans lequel ils fonctionnent.

Par exemple, si vous disposez d'un capteur de température, les données de température en elles-mêmes ne vous apportent pas beaucoup d'informations. Néanmoins, lorsque vous disposez des données suivantes : pièce où se trouve le capteur, dimensions de la pièce en question, autres appareils présents et données qu'ils capturent, ainsi que nombre de personnes dans

la pièce et leur niveau d'activité, vous obtenez un aperçu bien meilleur de la signification des données de température. Vous pouvez utiliser [Azure Digital Twins](#) pour modéliser l'environnement physique dans lequel résident vos appareils IoT. Ce service vous permet de créer un graphique de données comprenant des lieux, des personnes et des éléments, par exemple un immeuble de bureaux dans lequel se trouve une pièce avec des individus et des capteurs.

Azure Digital Twins vous permet d'apporter du contexte à des données issues de diverses sources et les associent entre elles (par exemple, la température et les données d'humidité de capteurs se trouvant dans la même pièce). Cela vous permet d'interroger des données dans le contexte d'un espace et non à partir de capteurs individuels.

Azure Digital Twins vous permet également de gérer des autorisations relatives à des données et des appareils dans le contexte du monde physique. Vous pouvez utiliser Azure AD pour préciser que certains utilisateurs sont en mesure d'accéder à des données uniquement à partir d'un emplacement physique spécifique.

```
// Commencez par trouver des  
pièces disponibles à l'aide  
d'Azure Digital Twins.
```

Azure Sphere

De plus en plus d'appareils contiennent des microcontrôleurs (ordinateurs monopuce) pour développer leur intelligence. Cela active des scénarios tels qu'une machine à laver envoyant une notification de maintenance de routine au propriétaire.

Il est néanmoins complexe et compliqué de sécuriser les dispositifs avec des microcontrôleurs intégrés, comme le démontrent les nombreux incidents de sécurité liés à des appareils connectés qui se sont produits ces dernières années.

[Azure Sphere](#) offre une solution pour la sécurisation d'appareils basés sur microcontrôleur. La plateforme Azure Sphere se compose d'une puce de microcontrôleur sécurisée, d'un SE et d'un ensemble de services cloud se connectant au microcontrôleur et dont la mise à jour s'effectue en cours d'exécution. Cette association fournit la base pour un monde connecté, sécurisé.

Vous pouvez utiliser les outils de développement Visual Studio pour développer des applications qui s'exécutent sur Azure Sphere. Cela ouvre l'univers du développement de microcontrôleur sur un groupe bien plus large de développeurs.

// En savoir plus sur Azure Sphere

En savoir plus sur Azure IoT

Les solutions Azure IoT sont faciles à utiliser et associées à nombre de ressources connexes, dont les suivantes :

- [Azure IoT School](#) : cette académie en ligne gratuite offre une formation complète à Azure IoT avec divers cours allant des niveaux débutant à avancé.
- [Développer des solutions IoT avec Azure](#) : cette formation en ligne guidée vous fait découvrir tous les principaux concepts d'Azure IoT à votre propre rythme.
- [Page d'application Azure IoT](#) : cette ressource offre une présentation d'Azure IoT et des exemples de son utilisation possible.
- [Accélérateurs de solutions Azure IoT](#) : utilisez ces modèles pour commencer à utiliser Azure IoT.
- [Azure IoT Hub](#) : cette ressource offre une présentation d'Azure IoT Hub et des exemples de son utilisation possible.
- [Azure IoT Edge](#) : cette ressource offre une présentation d'Azure IoT Edge et un exemple de son utilisation possible.
- [Vidéos techniques Azure IoT](#) : en savoir plus sur IoT sur Channel 9.

Qu'utiliser, et quand ?

Maintenant que vous avez lu des informations concernant les services Azure IoT disponibles, comment savoir quel service utiliser pour votre scénario ? Le Tableau 5-2 présente le contexte d'utilisation de chaque option IoT dans Azure.

Autres lectures

Vous trouverez plus d'informations sur l'utilisation d'Azure pour votre solution IoT dans les Ebooks suivants :

// [Guide d'IoT pour les développeurs](#)

// [Conçu pour perturber](#)

Tableau 5-2

	Azure IoT Hub	Azure IoT Central	Accélérateurs de solution Azure IoT	Azure IoT Edge
Créer une solution IoT accordant beaucoup de contrôle et en rédigeant un code personnalisé	●			
Créer une solution IoT sans se soucier du code et de la gestion des services Azure		●		
Créer une solution IoT pour un scénario courant avec un minimum de configuration et de codage			●	
Exécuter des charges de travail d'IA localement sur des appareils IoT	●*			●

Tous les services incluent un niveau gratuit que vous pouvez utiliser pour commencer.

*Azure IoT Hub est requis pour gérer les déploiements et les appareils d'Azure IoT Edge.

06 /

Où et comment déployer vos services Azure

Comment Azure peut-il déployer vos services ?

Azure dispose d'une option pour chaque type d'organisation, y compris ceux ayant besoin d'Azure pour être dans leur propre centre de données. Vous pouvez déployer vos applications soit dans le Cloud public Azure, soit en local, dans [Azure Stack](#) pour choisir le degré de portabilité de vos applications.

Il est également possible de développer des applications dans des conteneurs pour les déployer en local ou dans un autre cloud, ou en utilisant des [modèles Azure Resource Manager](#) pour définir un script de l'intégralité de votre infrastructure en tant que code.

Explorons ces options plus en détail.



Infrastructure en tant que code

L'Infrastructure en tant que code (IaC) capture les définitions d'environnement sous forme de code déclaratif, comme des documents JSON, pour effectuer automatiquement l'approvisionnement et la configuration. Tous les services Azure présentés dans ce guide reposent sur [Azure Resource Manager](#), qui vous permet de documenter votre environnement en tant qu'IaC, grâce aux [modèles Azure Resource Manager](#). Ces modèles sont des fichiers JSON qui décrivent ce que vous voulez déployer et les paramètres correspondants.

Il est facile de créer des modèles Azure Resource Manager dans Visual Studio et Visual Studio Code à l'aide des modèles de projet Azure Resource Group. Vous pouvez également générer des modèles Azure Resource Manager à partir du portail Azure en cliquant sur le bouton Script d'automatisation, qui se trouve sur la barre de menus de chaque ressource du portail Azure. Cela crée le modèle Azure Resource Manager pour la ressource donnée et génère même le code permettant de développer la ressource par l'intermédiaire d'Azure CLI, de Windows PowerShell, de .NET et d'autres.

Une fois que vous disposez d'un modèle Azure Resource Manager, vous pouvez le déployer vers Azure par l'intermédiaire de PowerShell, d'Azure CLI ou de Visual Studio. Vous pouvez sinon automatiser son déploiement dans un pipeline de déploiement continu (DC) par l'intermédiaire d'Azure DevOps.

Un excellent exemple de déploiement de ressources dans le cloud en utilisant Azure Resource Manager est le [bouton Déployer sur Azure](#) que l'on trouve dans de nombreux dépôts GitHub,

Outre l'utilisation de Resource Manager pour l'IaC, vous pouvez apporter vos propres outils et compétences tels qu'[Ansible](#), [Chef](#) et [Terraform](#) pour configurer et gérer l'infrastructure Azure directement.

Plans Azure

Il est facile d'utiliser les stratégies, droits d'accès, identités de l'utilisateur et groupes de ressources des modèles Azure Resource Manager pour concevoir et créer une infrastructure complète. Mais comment maintenez-vous l'unité de tous ces éléments ? Et comment suivez-vous dans quels environnements chaque élément d'infrastructure a été déployé et quelle version de l'artefact est maintenant déployée ?

Organisez tous vos artefacts d'infrastructure avec les [Plans Azure](#). Les Plans Azure fournissent un mécanisme qui vous permet de créer et de mettre à jour des artefacts, de les affecter à des environnements et de définir des versions. Vous pouvez stocker et gérer ces artefacts ainsi que gérer leurs versions et les associer à des environnements.

Cela vous aidera à organiser votre infrastructure et à créer un contexte pour les stratégies, groupes de ressources, identités de l'utilisateur et modèles Azure Resource Manager.

```
// Commencez par définir et  
attribuer un Azure Blueprint  
au portail Azure.
```

Conteneurs dans Azure

« Mise en conteneur » est l'une de ces expressions à la mode du domaine des nouvelles technologies qu'on entend sans cesse aux informations. Mais les conteneurs représentent plus qu'un simple buzz, ils sont effectivement très utiles pour l'exécution de vos applications. Un conteneur est fondamentalement une machine virtuelle légère qui démarre et s'arrête beaucoup plus rapidement qu'une machine virtuelle classique, et est donc beaucoup plus utile pour le développement, les tests et l'exécution d'applications en production.

Le principal avantage des conteneurs est que chaque conteneur distinct demeure toujours le même. Vous exécutez un conteneur localement lorsque vous développez votre application, avant d'utiliser la même configuration de conteneur dans

le Cloud et partout ailleurs. Toute votre équipe utilise exactement la même configuration de conteneur. Vous savez donc ainsi que l'infrastructure est la même pour tout le monde qu'en production. Avec les conteneurs, la vieille excuse des développeurs, « ça marche sur ma machine », signifie désormais que cela fonctionnera aussi en production.

Il existe de nombreuses technologies pour l'exécution de conteneurs, notamment [Docker](#). Azure peut exécuter et gérer des conteneurs avec des [instances de conteneur Azure](#) et [Azure Kubernetes Service](#). Vous pouvez également exécuter des conteneurs dans [Web App pour conteneurs](#) et dans [Azure Batch](#). Le tableau 6-1 illustre quel service vous pouvez choisir dans le cadre de divers scénarios lorsque vous utilisez des conteneurs.

Tableau 6-1

	Azure Kubernetes Service	Azure Container Instances	Web App pour conteneurs	Conteneurs sur Azure Batch
Pour les déploiements en production de systèmes complexes (avec un orchestrateur de conteneurs)	●			
Pour l'exécution de configurations simples (éventuellement sans orchestrateur)		●	●	
Pour des charges de travail de longue durée sur des conteneurs	●			●
Pour des charges de travail de courte durée sur des conteneurs		●		●
Pour l'orchestration d'un système basé sur des conteneurs	●			
Orchestration avec des orchestrateurs open source Kubernetes	●			
Orchestration avec un orchestrateur intégré				
Utilisation de fonctions App Service telles que des emplacements de déploiement			●	

Azure Stack

Si vous avez besoin que vos applications et vos données demeurent sur site, mais que vous voulez continuer à bénéficier de la puissance d'Azure, [Azure Stack](#) est le produit idéal pour vous. Unique dans le secteur, Azure Stack est une extension d'Azure que vous hébergez dans votre propre environnement. En résumé, vous disposez d'Azure dans une boîte.

Vous utilisez Azure Stack de la même manière que vous utilisez Azure, avec la même expérience du portail Azure et les mêmes API que vous pouvez utiliser avec Azure CLI, Windows PowerShell ou votre IDE préféré.

Vous pouvez exécuter des services comme Azure App Service et Virtual Machines sur Azure Stack. Tout est exactement pareil que dans le Cloud public, sauf que vous l'exécutez localement. Si vous optez pour un transfert vers le Cloud public, il vous suffit de transférer les services d'Azure Stack vers Azure.

Exemple : Azure Stack

Pour faciliter son activité, une entreprise qui organise des croisières de luxe a élaboré divers logiciels, dont une application de gestion des cabines et une autre de gestion des passagers. Le navire de croisière tout entier dépend de ces applications. Par le passé, ces applications étaient exécutées sur des serveurs embarqués sur les navires. La compagnie était dans l'obligation de procéder ainsi car les navires n'avaient pas de connexion Internet tout le long de leur voyage.

La compagnie a considéré que l'exécution locale de ses applications était pesante car il lui fallait maintenir des machines virtuelles et des systèmes d'exploitation et trouver des solutions à de sérieux problèmes de disponibilité.

La compagnie exécute à présent ses applications sur Azure Stack qui est exécuté à bord des navires de croisière. Azure Stack assure les mêmes services qu'Azure, si bien que le déploiement et la gestion des applications sont devenus beaucoup plus faciles. La compagnie utilise également Azure App Service pour exécuter ses applications, ce qui lui permet de se concentrer sur les applications plutôt que sur le maintien des systèmes d'exploitation et des machines virtuelles. Mieux encore, les utilisateurs bénéficient de la disponibilité supérieure qui fait partie d'Azure et, par conséquent, d'Azure Stack.

Où et quand effectuer le déploiement ?

Si vous souhaitez déployer des services IaaS (où vous avez le contrôle du système d'exploitation), pensez aux options suivantes :

- **Localement ou n'importe où ailleurs** (comme sur votre PC local ou un autre cloud), vous pouvez utiliser les éléments suivants :
 - Azure Stack (où vous déployez des services tels que des machines virtuelles)
 - N'importe lequel des services de conteneur Azure (car les conteneurs peuvent s'exécuter n'importe où)
- **Dans le cloud public Azure**, vous pouvez utiliser les éléments suivants :
 - Les conteneurs (car ils peuvent aussi s'exécuter dans n'importe lequel des services de conteneur Azure)

Si vous souhaitez déployer des services reposant sur le PaaS (où vous avez moins de contrôle, mais où la plateforme fait le plus gros du travail), pensez aux options suivantes :

- **Localement ou n'importe où ailleurs** (comme sur votre PC local ou un autre cloud), vous pouvez utiliser les éléments suivants :
 - Azure Stack (car vous pouvez déployer des services PaaS tels qu'App Service dans Azure Stack)
- **Dans le cloud public Azure**, vous pouvez utiliser les éléments suivants :
 - N'importe quel service Azure PaaS que vous utilisez comme script avec un modèle Resource Manager

Autres lectures

Vous trouverez plus d'informations sur le déploiement de vos applications dans Azure et la réduction des coûts dans les Ebooks gratuits suivants :

// [Fondamentaux de la migration vers le Cloud](#)

// [Tirer pleinement parti du Cloud, partout](#)

// [Pratiques DevOps efficaces](#)

// [Azure pour les architectes](#)

07 /

Partager le
code, suivre les
travaux et livrer
des logiciels

Comment Azure peut-il vous aider à améliorer votre planification et votre collaboration et accélérer la livraison de vos applications ?

Vous avez consacré des nuits ou des week-ends complets au déploiement de nouvelles versions de vos applications. Si c'est le cas, vous avez sans doute également passé beaucoup de temps à essayer de résoudre des bogues qui découragent les utilisateurs d'utiliser cette nouvelle version. Il existe une façon plus efficace.

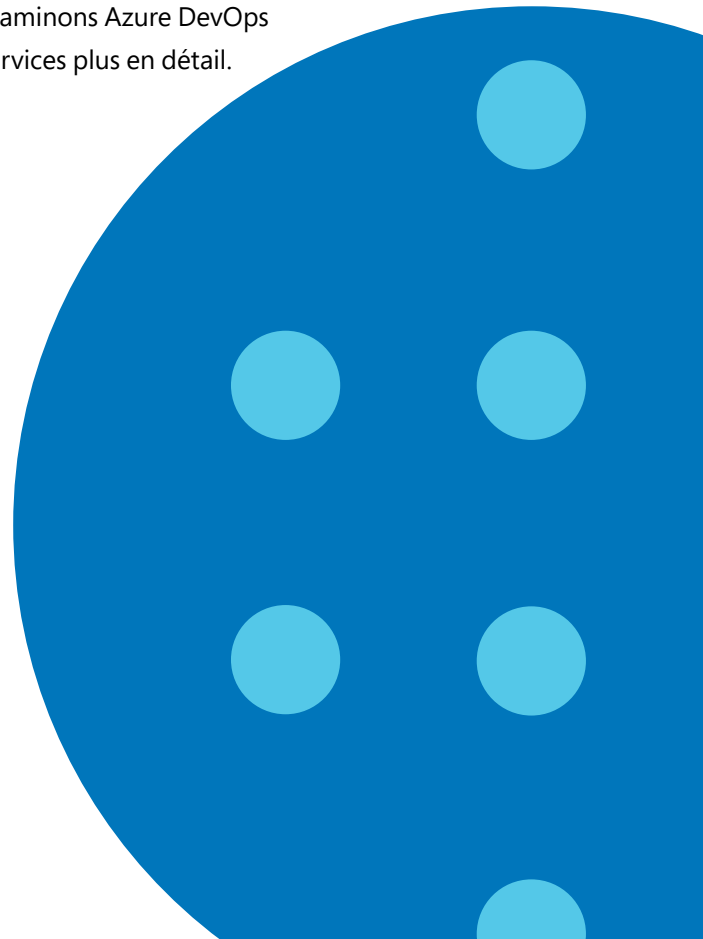
[Azure DevOps](#) est un ensemble de solutions qui peut contribuer à automatiser vos builds et vos déploiements et à tester automatiquement votre code et vos applications avant le lancement.

Pour aider au développement, au déploiement, au test et au suivi de votre code et de vos applications, Azure DevOps inclut les éléments suivants :

- [Tableaux de bord Azure](#)
Utilisez les Tableaux de bord Azure pour planifier, suivre et discuter du travail entre équipes.

- [Azure Repos](#)
Utilisez Azure Repos pour collaborer au développement de code avec des dépôts Git public et privés gratuits, des demandes de tirage (pull requests) et une révision du code.
- [Pipelines Azure](#)
Utilisez les Pipelines Azure pour créer des pipelines de build et de mise en production qui automatisent les builds et les déploiements.
- [Plans de test Azure](#)
Utilisez Azure Test Plans pour améliorer la qualité globale de votre code avec des services de tests manuels et exploratoires pour vos applications.
- [Azure Artifacts](#)
Utilisez Azure Artifacts pour partager des packages de code (tels que npm, NuGet et Maven) au sein de votre organisation.

Examinons Azure DevOps Services plus en détail.



Tableaux de bord Azure

La planification de votre travail et le suivi de votre progression sont des tâches importantes, et les Tableaux de bord Azure peuvent vous aider à les réaliser.

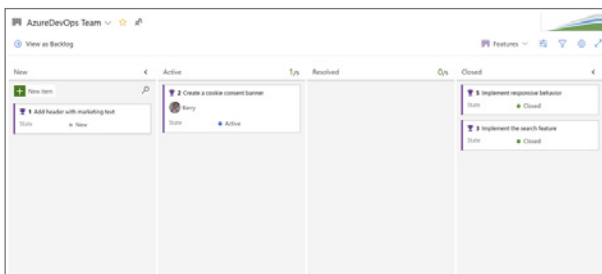


Figure 7-1

Dans les Tableaux de bord Azure, vous pouvez créer un backlog complet d'éléments de travail (tels que des récits utilisateurs) et les planifier en sprints afin que votre équipe puisse travailler de manière itérative pour terminer les tâches.

L'ensemble du système de planification est optimisé pour travailler de manière agile. Il inclut même des tableaux Kanban pour la gestion de votre travail (Figure 7-1).

Tout peut être personnalisé pour mieux fonctionner pour vos équipes, qu'elles utilisent scrum, une autre méthode agile, ou le processus CMMI (Capability Maturity Model Integration). Vous pouvez créer et gérer des tâches, des fonctions, des récits utilisateurs, des bogues, des exigences, des problèmes, des demandes de modifications, etc.

Essayez de personnaliser vos tableaux et de créer des graphiques (tels que des burndown charts ou des listes des tâches) illustrant les informations dont vous avez besoin. Vous pouvez interroger la progression et les éléments de travail, puis les utiliser pour personnaliser vos tableaux, graphiques et listes. À partir de là, partagez-les ou épinglez-les sur votre tableau de bord Azure DevOps afin qu'ils soient visibles de tous.

```
// À essayer : première  
utilisation des Tableaux de  
bord Azure pour suivre des  
problèmes, des tâches et des  
épopées
```

Azure Repos

Le contrôle de version est essentiel pour travailler ensemble et s'assurer que votre atout principal, votre code, est stocké en toute sécurité. [Azure Repos](#) est un ensemble d'outils de contrôle de version pour le stockage de votre code et son partage avec votre équipe. Il est utile à la fois pour les équipes et pour les développeurs individuels. Le contrôle de version conserve un historique de vos développements, ce qui vous permet d'analyser votre code, voire de restaurer vers n'importe quelle version de votre code.

Choisissez entre les deux systèmes suivants de contrôle de version lorsque vous utilisez Azure Repos :

Git

Il s'agit d'un système de contrôle de version largement utilisé par les développeurs et qui est en outre la base pour [GitHub](#). [Git](#) est un système distribué de contrôle de version ce qui signifie que l'intégralité du code source (toutes les versions de tous les fichiers) se trouve sur votre machine, ce qui facilite le travail hors ligne. Avec Git, la source fidèle est fondamentalement sur la machine de chacun et est synchronisée lorsque les développeurs effectuent un push de code sur le serveur Git (dans ce cas, Azure Repos).

Azure Repos utilise le Git standard. Cela signifie que vous pouvez l'utiliser avec n'importe quel outil Git et IDE, y compris [Visual Studio](#) et [Visual Studio Code](#) ainsi que Git pour [Windows](#), [Mac](#), [Eclipse](#) et [IntelliJ](#).

Lorsque vous suivez le [flux de travail Git](#), vous commencez généralement par créer votre propre branche du code, par exemple pour ajouter une fonction. Une fois cela terminé, vous validez votre code pour créer une demande de tirage (pull request) pour cette branche et la soumettre au serveur. Les utilisateurs peuvent consulter, évaluer, tester et discuter cette demande de tirage (pull request). Une fois suffisante pour être tirée dans la branche principale, la demande est acceptée et votre branche de développement peut être supprimée.

Avec Azure Repos, vous disposez d'un ensemble d'outils complet pour prendre en charge le [flux de travail Git](#). Vous pouvez lier des éléments de travail tels que des récits utilisateurs ou des bogues pour réaliser des demandes de tirage (pull requests) afin que vous sachiez à quoi se rapporte chaque modification. Vous pouvez avoir des discussions sur du code validé et même commenter des modifications dans le code. Azure Repos permet également de voter des modifications dans le code, ainsi une modification n'est acceptée qu'une fois que tous les membres de l'équipe l'ont acceptée.

Azure Repos permet de profiter gratuitement de référentiels Git privés et illimités.

// Commencez par apprendre comment coder avec Git

Team Foundation Version Control

Team Foundation Version Control (TFVC) est un système centralisé de contrôle de version qui garantit qu'une source fidèle est toujours maintenue sur le serveur. Les développeurs n'ont généralement qu'une seule version de chaque fichier sur leur machine, ce qui complique le travail hors ligne.

Avec [TFVC](#), vous pouvez choisir de travailler avec les espaces de travail suivants :

Espaces de travail du serveur : les développeurs extraient publiquement des fichiers du serveur afin de pouvoir apporter des modifications uniquement à ces fichiers. Une fois qu'ils ont terminé, ils peuvent réintégrer les modifications et d'autres développeurs peuvent vérifier les fichiers pour apporter des modifications. Cela supprime le besoin de fusionner les modifications ainsi que la possibilité de conflits de code.

Espaces de travail locaux : à l'aide de ces derniers, les développeurs disposent chacun de la version la plus récente des fichiers sur leur machine et peuvent les modifier individuellement. Une fois toutes les modifications apportées, ils intègrent les modifications sur le serveur et résolvent les conflits si nécessaire.

Avec TFVC sur Azure Repos, chacun peut télécharger les versions de branches de code que vous créez sur le serveur. Azure Repos fournit également un ensemble d'outils complet qui vous permet de joindre des éléments de travail à des modifications de code. Il est également possible de demander et de réaliser des révisions du code, afin que votre équipe puisse discuter des modifications et recommander des mises à jour avant qu'elles soient fusionnées dans la branche principale.

```
// À essayer : début de  
développement et de partage de  
votre code dans TFVC à l'aide  
de Visual Studio
```

Pipelines Azure

Une fois que votre code est dans un dépôt comme Azure Repos, vous pouvez démarrer pour automatiser processus de build et de mise en production avec les [Pipelines Azure](#).

Les Pipelines Azure génèrent une valeur importante en un temps réduit. Ils permettent l'intégration continue (IC) pour la compilation et le test du code lorsque des modifications interviennent, ainsi qu'un déploiement continu (DC) appliqué à des applications après que les modifications sont compilées et testées avec succès. Nous encourageons toute organisation à explorer l'IC et le DC. En effet, ces processus améliorent la qualité du code et réduisent les efforts de déploiement.

Les Pipelines Azure peuvent aider en matière d'IC et de DC en proposant des pipelines de build et de déploiement. Chacun contient des étapes permettant de compiler et de tester votre code, ainsi que de le déployer dans un ou plusieurs environnements. La beauté des Pipelines Azure repose sur le fait qu'ils fonctionnent avec tout type de code, où que vous le stockiez, depuis C# sur Azure Repos jusqu'à Java sur BitBucket et tout autre.

Les Pipelines Azure fonctionnent très bien avec les services Azure pour déployer votre application dans une application web Azure, par exemple. Ils fonctionnent également avec tout service s'exécutant dans tout autre environnement, tel que Google Cloud, Amazon, ou même en local dans votre propre datacenter. Si vous utilisez déjà des

outils d'intégration continue tels que [Jenkins](#) ou [Spinnaker](#), vous pouvez facilement faire migrer vos builds et vos pipelines existants vers Azure et profiter des plug-ins d'agents dynamiques pour réduire les contraintes et les coûts d'infrastructure.

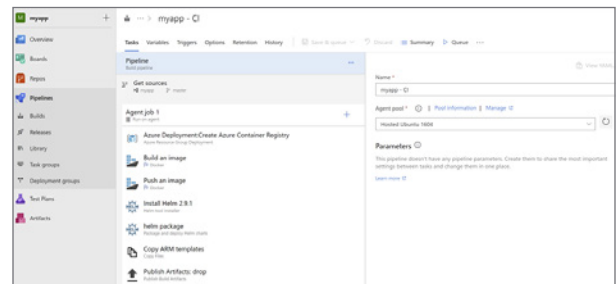


Figure 7-2 : Pipelines Azure : pipeline de build

Il existe deux manières d'utiliser les Pipelines Azure. Vous pouvez [créer des pipelines](#) à l'aide du concepteur visuel dans le portail Azure DevOps, ou bien vous pouvez utiliser l'approche YAML plus avancée. Au cours de cette approche, vous créez un fichier code YAML qui contient toutes les étapes du pipeline et vous le validez au regard du contrôle de source.

La façon la plus simple de débiter consiste à utiliser le concepteur visuel. Observons un exemple de pipeline de build et de mise en production :

Pipeline de build : le pipeline de build (ou pipeline CI) de la Figure 7-2 illustre une liste de tâches qui seront exécutées lorsque ce pipeline s'exécute. Le pipeline est configuré pour s'exécuter dès que de nouvelles modifications sont validées dans Azure Repos. Il va prendre le code dans Azure Repos (qui est une application Node.js), développer une image de conteneur Docker à partir

de là et la placer dans [Azure Container Registry](#). À partir de là, [Helm](#) peut l'utiliser pour compiler l'image dans un package qui peut être déployé sur [Azure Kubernetes Service](#).

Veuillez noter que vous pouvez configurer les hôtes qui exécutent vos pipelines pour vous. Sur la droite de l'image, vous pouvez voir que ce pipeline spécifique s'exécutera sur un pool hébergé de machines Ubuntu. Des hôtes Linux et Windows sont également disponibles et, même un MacOS hébergé que vous pouvez utiliser pour développer vos applications iOS. Ce n'est qu'un exemple d'un pipeline de build. Il est possible d'en créer un pour toute application imaginable. Vous pouvez également intégrer des tests, y compris des tests unitaires et des tests du code statique, dans le pipeline de build.

Pipeline de mise en production : le pipeline de mise en production (ou pipeline CD) s'exécute dès que le pipeline de build s'exécute avec succès, mais vous pouvez également le configurer pour être déclenché manuellement. Le pipeline de mise en production de la Figure 7-3 contient neuf tâches qui commencent par créer un cluster Azure Kubernetes Service, puis déploient le package Helm qui a été produit dans le pipeline de build sur le cluster.

La figure 7-3 illustre les étapes dans l'étape de développement. Les étapes sont comme des environnements. Vous pouvez configurer une étape pour vos environnements de développement, de test, de production, etc. Vous pouvez également configurer des éléments tels que des approbations de prédéploiement, ce qui requiert l'approbation par un tiers du lancement d'une application dans un environnement spécifique (tel que l'environnement

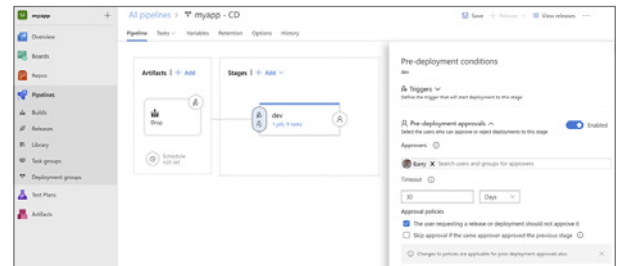


Figure 7-3

de production). Cela signifie que vous pouvez tout automatiser et laisser la décision de lancement en production à un responsable en fonction des résultats des tests pour les étapes précédentes dans le pipeline.

Créez des pipelines aussi simples ou complexes que vous le souhaitez. Idéalement, vous voulez automatiser le plus possible, depuis la création et la destruction de votre infrastructure jusqu'au déploiement et au test de votre application. Les tâches de pipeline sont disponibles pour presque tout, et vous pouvez accéder à davantage de tâches en tant qu'extensions d'Azure DevOps dans [Visual Studio Marketplace](#).

// Découvrez les Pipelines
Azure en créant votre premier
pipeline

Plans de test Azure

Pour améliorer la qualité de vos applications, utilisez [Azure Test Plans](#) pour définir les Test Plans, puis créer et exécuter des tests manuels et exploratoires. Azure Test Plans fournit les outils pour créer des tests, les exécuter, enregistrer les commentaires et suivre les résultats des tests.

Dans Azure Test Plans, commencez par créer un plan de test. Il contient plusieurs suites de tests et cas de tests. Un cas de test peut être un test exploratoire dans lequel l'application est explorée pour voir si elle fonctionne comme prévu ; un test manuel guidé dans lequel les étapes de test et les résultats attendus sont décrits en détails, ou bien un test automatisé. Vous pouvez même enregistrer des étapes de test en enregistrant des clics dans une application et en permettant à Azure Test Plans d'automatiser ces clics dans un test. Vous pouvez également intégrer des tests de stress et de charge dans vos pipelines de build et de mise en production. Les cas de tests sont des éléments de travail, tout comme les tâches et les récits utilisateurs, et ils peuvent donc être planifiés dans une itération.

Une fois que vous avez créé un test (Figure 7-4), un testeur peut l'exécuter. Dans un test manuel, par exemple, le testeur utilise l'outil de test pour l'exécuter sur les étapes de test et les résultats d'enregistrement, y compris l'écran, la voix du testeur, les captures d'écran et les pièces jointes. Le testeur réussit ou échoue à chaque étape du test.

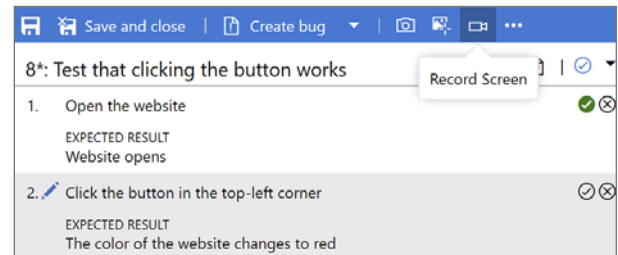


Figure 7-4

Depuis le contexte du test, le testeur peut également créer un bogue qui doit être résolu.

Afin de garantir que les attentes des parties prenantes correspondent à votre plan, Azure Test Plans vous permet également de [demander des commentaires](#) pour des éléments de travail tels que des récits utilisateurs. Cela permet aux parties prenantes de consulter ce que vous proposez et de fournir des commentaires sous la forme de texte, de pièces jointes, de vidéo ou de message vocal.

Élément peut-être le plus important, Azure Test Plans fournit des tableaux de bord et des graphiques sur la progression et le statut des tests dans votre projet. Vous pouvez les utiliser pour vérifier la qualité de votre application et sa progression sur la durée. Cela peut vous aider à identifier les fonctions qui ne sont pas encore prêtes à être déployées.

// Découvrez Azure Test Plans en créant des cas de tests manuels

Azure Artifacts

Du fait que les packages offrent une fonctionnalité que vous n'avez pas à développer vous-même, vous en utilisez sans doute un grand nombre dans vos applications. Et vous y accédez probablement de tout autant de sources : NuGet, npm, Maven, entre autres. Mais que se passe-t-il si votre équipe crée des packages que vous souhaitez utiliser uniquement en interne ? Où les hébergez-vous en toute sécurité et comment les partagez-vous ? [Azure Artifacts](#) offre cette possibilité. Azure Artifacts est un flux de packages qui vous permet d'héberger des packages que vous créez et de les sécuriser pour votre organisation.

Vous pouvez héberger [toutes sortes de packages](#) sur Azure Artifacts, dont NuGet, npm, Maven, Python et Universal Packages. Vous pouvez même utiliser le flux Azure Artifacts pour stocker des packages depuis des sources publiques telles que nuget.org et npmjs.com. Lorsque vous stocker des packages de sources publiques sur votre flux, vous pouvez continuer à les utiliser même s'ils ne sont plus disponibles sur le flux public. Cela est spécialement utile pour les packages stratégiques.

Suivez ces étapes simples pour utiliser Azure Artifacts :

1. [Créez](#) un flux Azure Artifacts.
2. [Publiez](#) votre package dans le flux.
3. [Consommez](#) le flux dans votre IDE préféré, tel que Visual Studio.

```
// Essayez-le maintenant : prise  
en main des packages Python  
dans Azure Artifacts
```

Autres lectures

Si vous souhaitez améliorer la qualité de votre logiciel et obtenir davantage d'informations sur l'automatisation de vos processus de build et de mise en production, téléchargez et lisez ces Ebooks gratuits :

```
// Pratiques DevOps efficaces
```

```
// Livraison en continu dans Java
```

```
// Azure pour les architectes
```

08 /

Azure en action

Procédure pas à pas N° 1 : l'expérience du portail Azure

L'un des outils Azure les plus importants est le point central, le portail Azure. La plupart des opérations que vous pouvez effectuer dans le portail Azure peuvent également l'être via l'API Azure, Azure CLI et Azure PowerShell.

Le portail Azure est un tableau de bord avec des vignettes. Il est facile de créer et de personnaliser des tableaux de bord, puis de les partager avec les membres de votre équipe.

Vignettes du portail Azure

Les vignettes, présentées dans la figure 8-1, affichent des informations concernant un service, ou servent de raccourci pour accéder à un service. Elles s'affichent dans tout le portail, dans les pages de tous les services. Elles représentent un moyen utile d'obtenir une présentation rapide de la façon dont un service fonctionne.

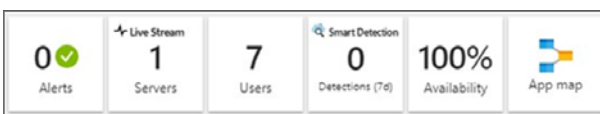


Figure 8-1

Les vignettes de services Application Insights affichent des informations telles que les alertes actives, les données actives entrantes, les utilisateurs actifs au cours des 24 dernières heures et la disponibilité. Vous pouvez personnaliser les informations et la taille d'une vignette ainsi que l'apparence des graphiques en ajustant leurs chronologies et en affichant les données dans différents formats, comme des lignes ou des barres. Vous pouvez également épingler des vignettes directement sur vos tableaux de bord, afin qu'elles soient la première chose que vous voyez quand vous accédez au portail (figure 8-2). Vous pouvez, par exemple, épingler des vignettes de toutes les métriques des services que vous utilisez afin de créer un [tableau de bord de suivi](#) que vous partagez avec votre équipe, ou affichez sur un écran physique.

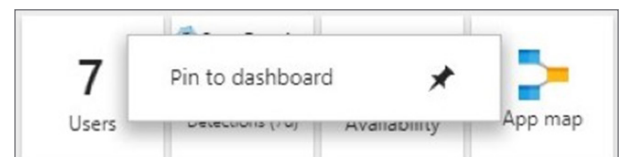


Figure 8-2

Ajouter des services

Vous pouvez trouver et ajouter des services dans le portail Azure de plusieurs façons.

Pour créer de nouveaux services, sélectionnez le signe plus dans le coin supérieur gauche de la fenêtre du portail. Cette opération ouvre la zone de recherche du marché, où vous trouverez tout, depuis des applications web jusqu'aux serveurs Linux, comme l'illustre la figure 8-3.

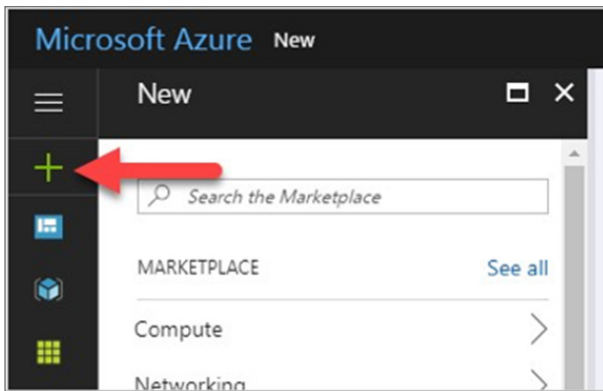


Figure 8-3

Lorsque vous trouvez le service qui vous intéresse dans les résultats de la recherche, comme illustré à la figure 8-4, un assistant vous fournit les instructions nécessaires à la configuration et au déploiement.

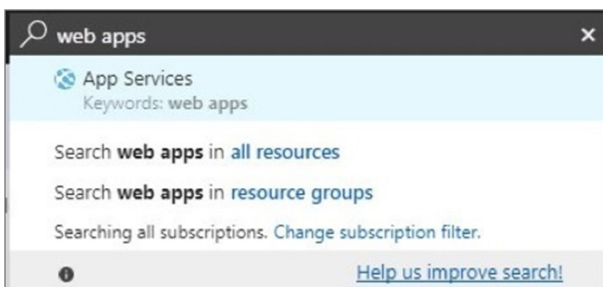


Figure 8-4

La zone de recherche située en haut du portail vous permet de parcourir toutes vos ressources et d'y accéder directement (figure 8-5). Le menu des favoris se trouve dans le volet situé sur le côté gauche du portail.

Ce menu affiche les catégories de ressources telles qu'Azure App Service, représentées par leurs icônes. Vous pouvez réorganiser les icônes en les faisant glisser vers le haut ou vers le bas. Vous pouvez également sélectionner celles que vous souhaitez afficher en agrandissant le menu des favoris et en sélectionnant le symbole d'étoile à côté de ces catégories.

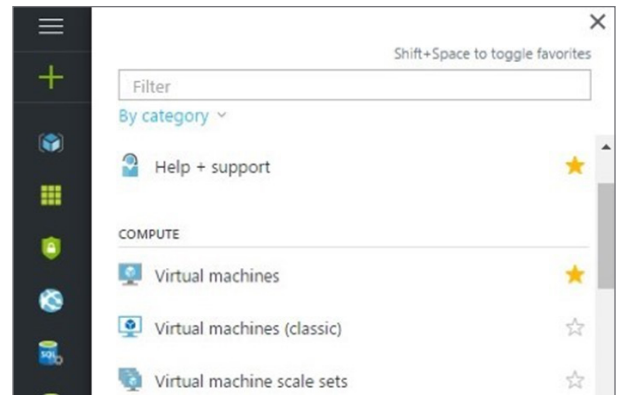


Figure 8-5

Comprendre les panneaux

Les pages dans Azure sont également appelées panneaux, et vous pouvez les épingler sur vos tableaux de bord. Lorsque vous ouvrez une application web, vous voyez d'abord le panneau Présentation, comme illustré sur la Figure 8-6.

Ce panneau fournit des outils permettant d'arrêter, de démarrer et de redémarrer l'application web et d'afficher des vignettes montrant ses mesures, telles que le nombre des requêtes et d'erreurs. Lorsque vous choisissez un autre élément de menu, un nouveau panneau s'ouvre. Les panneaux s'ouvrent toujours en contexte. Par exemple, si vous ouvrez le panneau Emplacements de déploiement et que vous sélectionnez Créer un emplacement de déploiement, un nouveau panneau s'affiche à droite du panneau Emplacements de déploiement, préservant ainsi le contexte dans lequel vous vous trouvez.

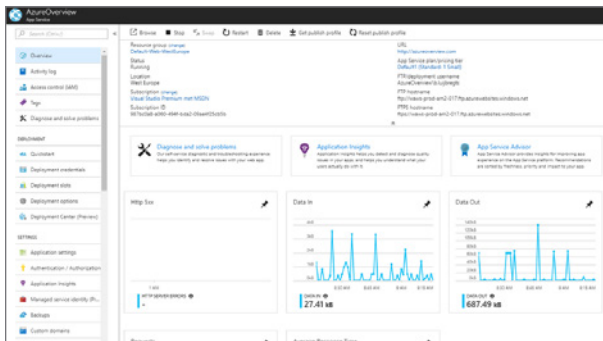


Figure 8-6

Création d'une machine virtuelle

Nous allons utiliser le portail Azure pour créer une machine virtuelle. Une fois cela fait, nous la fermerons et la supprimerons afin que vous ne continuiez pas à payer son utilisation.

Un mot au sujet des groupes de ressources

La machine virtuelle sera déployée dans un groupe de ressources, un conteneur logique qui contient des ressources. Toutes les ressources Azure résident dans des groupes de ressources. Vous pouvez gérer la sécurité d'un groupe de ressources ainsi que voir ce que coûtent les ressources du groupe. Il est courant de regrouper des services connexes au sein d'un groupe de ressources afin de faciliter leur sécurisation.

1. Dans le coin supérieur gauche du portail Azure, sélectionnez **Créer un service**.
2. Dans la zone de recherche, tapez **machine virtuelle Windows Server**.
3. Cliquez sur **Windows Server 2016 Datacenter**.
4. Cliquez sur **Créer**. L'Assistant Créer un ordinateur virtuel s'ouvre.
5. Choisissez un nom pour la machine virtuelle.
6. Choisissez le type de disque. Un disque SSD fournit une machine virtuelle plus rapide, mais est plus onéreux. Pour cette procédure pas à pas, choisissez **SSD**.
7. Tapez un nom d'utilisateur.
8. Pour le type d'authentification, sélectionnez **Mot de passe**.
9. Tapez un mot de passe et confirmez-le.
10. Dans la zone **Groupe de ressources**, tapez un nouveau nom.

11. Choisissez l'emplacement de la machine virtuelle, puis cliquez sur **OK**.
12. Choisissez la taille de la machine virtuelle. Il existe de nombreuses options de dimensionnement de machines virtuelles. La performance de la machine virtuelle en détermine le coût. Utilisez l'assistant pour sélectionner le nombre de cœurs et la quantité de mémoire souhaités, et choisir des options en conséquence. En outre, d'autres fonctions sont associées aux options de taille, telles que les suivantes :

- Type de disque dur (SSD ou disque dur normal).
- Le nombre maximal d'opérations d'entrée/sortie par seconde (IOPS). Cela détermine les performances de la machine virtuelle d'une manière significative, surtout si vos applications effectuent de nombreuses opérations de lecture et d'écriture à partir du disque dur et vers celui-ci.
- La quantité de disques de données qui peuvent être installés sur la machine virtuelle.
- La capacité à effectuer l'équilibrage de charge.
- La carte graphique installée sur la machine virtuelle. Cela est utile si vous devez effectuer un grand nombre de rendus de graphiques ou gérer une charge de calcul lourde.

13. Une fois que vous avez sélectionné la taille, vous pouvez configurer des paramètres supplémentaires tels que le réseau virtuel, l'adresse IP et les extensions installées sur la machine. Pour l'instant, laissez tout en l'état, puis sélectionnez **OK**.

14. Consultez le résumé, acceptez les termes, puis cliquez sur **Créer**.

Il ne faut habituellement que quelques minutes pour déployer une machine virtuelle. Lorsque vous accédez à votre machine virtuelle dans le portail Azure, vous pouvez la configurer davantage et vous connecter à l'aide du protocole RDP (Remote Desktop Protocol).

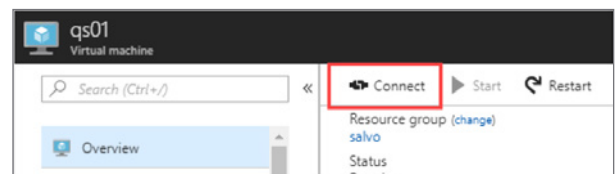


Figure 8-7

Pour vous connecter à la machine virtuelle à l'aide de RDP, cliquez sur **Connecter** dans le panneau **Présentation** de la machine virtuelle du portail Azure (Figure 8-7). Cela déclenche un téléchargement du fichier RDP que vous pouvez utiliser pour vous connecter à la machine virtuelle.

Nettoyage des ressources de la procédure pas à pas

Lorsque vous en avez terminé avec la machine virtuelle, vous pouvez la fermer et la supprimer en effaçant le groupe de ressources que nous avons créé au moment de générer la machine virtuelle. Il contient la machine virtuelle et toutes les autres ressources qui sont créées automatiquement. Une fois le groupe de ressources effacé, vous ne payez plus pour aucune des ressources que vous avez utilisées lors de cette procédure pas à pas.

Procédure pas à pas N° 2 : développement d'une application web et d'une base de données sur Azure

Au cours de cette procédure pas à pas, nous allons déployer une application .NET Core simple qui se connecte à une base de données SQL. Nous allons ensuite l'héberger dans Web Apps.

Pour continuer, [Git v2 ou version ultérieure](#), [.NET Core](#) et [Visual Studio Code](#) devront être installés sur votre appareil. Nous utiliserons également un exemple d'application ASP.NET Core MVC pour gérer une liste de tâches.

Création d'une application web et d'une base de données par l'intermédiaire du portail Azure

Pour héberger l'application .NET Core, nous allons créer une application web dans le portail Azure.

1. Dans le portail Azure, sélectionnez **Créer un service**.
2. Recherchez **Web App**. Le panneau **Web App** s'ouvre. Sélectionnez **Créer**. Le panneau **Créer une application web** s'ouvre.
3. Attribuez un nom à l'application web.
4. Créez un groupe de ressources en lui attribuant un nom.
5. Laissez la sélection du système d'exploitation sur **Windows**.
6. Sélectionnez ou créez un **plan App Service**, puis sélectionnez **Créer**.

Les services tels que Web Apps s'exécutent sur les plans App Service. Un plan App Service est une représentation abstraite des ressources et des fonctions, comme l'UC et la mémoire, et ils sont pris en compte dans les niveaux de tarification.

Les plans App Service sont également liés à une zone géographique spécifique que vous choisissez. Vous pouvez, par exemple, exécuter votre application Web Apps dans un plan App Service de niveau de tarification S1 qui dispose d'un cœur et de 1,75 Go de RAM, comme illustré à la Figure 8-8.

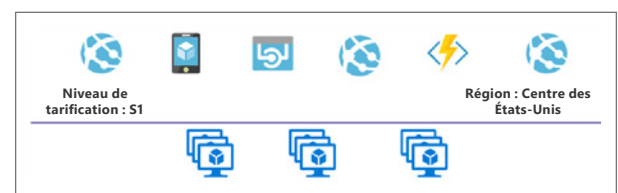


Figure 8-8

Vous pouvez exécuter autant d'App Services que vous le souhaitez sur un plan App Service, mais veuillez noter que vous devez partager les ressources entre tous les App Services.

Pour héberger la base de données, nous allons créer une base de données SQL. Elle fonctionne comme une base de données SQL Server locale et s'exécute désormais de façon entièrement gérée dans Azure.

1. Dans le portail Azure, cliquez sur **Créer un service**.
2. Recherchez **SQL Database**, puis cliquez dessus pour ouvrir le panneau **SQL Database**. Cliquez sur **Créer**. Le panneau **Créer une base de données SQL** s'ouvre.
3. Saisissez un nom de base de données.
4. Sélectionnez le groupe de ressources que vous avez créé pour l'application Web.
5. Laissez la source comme **Base de données vide**.
6. Cliquez sur **Serveur** pour créer un serveur SQL Database.
 - a. Saisissez un nom pour le serveur.
 - b. Saisissez la connexion administrateur au serveur. Il s'agit du nom d'utilisateur pour le serveur.
 - c. Saisissez le mot de passe que vous utiliserez pour vous connecter au serveur.
 - d. Confirmez le mot de passe.
 - e. Choisissez un emplacement. Choisissez le même emplacement que celui que vous avez sélectionné pour le plan App Service.
 - f. Cliquez sur **Sélectionner** pour soumettre la nouvelle configuration de serveur.
7. Sélectionnez un niveau de tarification. À des fins de développement et de test, le niveau **De base** suffit.
8. Cliquez sur **Créer**. La base de données est maintenant créée.
9. Accédez à la base de données SQL et cliquez sur **Afficher la chaîne de connexion de la base de données**.
10. Notez la chaîne de connexion car vous en aurez besoin plus tard dans ce tutoriel.

Exécution locale de l'application .NET Core

Vous allez maintenant exécuter l'application localement, avant de l'exécuter dans Azure. Vous pouvez exécuter l'application localement car, par défaut, elle utilise une base de données SQLite, qui est un moteur de base de données SQL autonome.

1. Ouvrez une invite de commande et accédez à un répertoire que vous souhaitez utiliser comme votre répertoire de code source pour ce projet.
2. Exécutez les commandes suivantes pour obtenir le code source et accéder au dossier du projet :

```
git clone https://github.com/azure-samples/dotnetcore-sqlldb-tutorial  
  
cd dotnetcore-sqlldb-tutorial
```

3. Le projet utilise Entity Framework Core pour alimenter sa base de données. Pour vous assurer que la base de données est à jour et pour exécuter l'application localement, exécutez les commandes suivantes :

```
dotnet restore  
  
dotnet ef database update  
  
dotnet run
```


4. L'application doit maintenant être en exécution, et l'URL de l'application (telle que **http://localhost:5000**) doit figurer dans l'affichage de la fenêtre de commande.
5. Accédez à cette URL depuis un navigateur. Cela va charger l'application, qui aura la même apparence que dans la figure 8-9. Vous pouvez désormais créer des éléments à faire en sélectionnant le lien **Créer nouveau**.
6. Fermez l'application en fermant la fenêtre de commande ou en appuyant sur **Ctrl + C**.

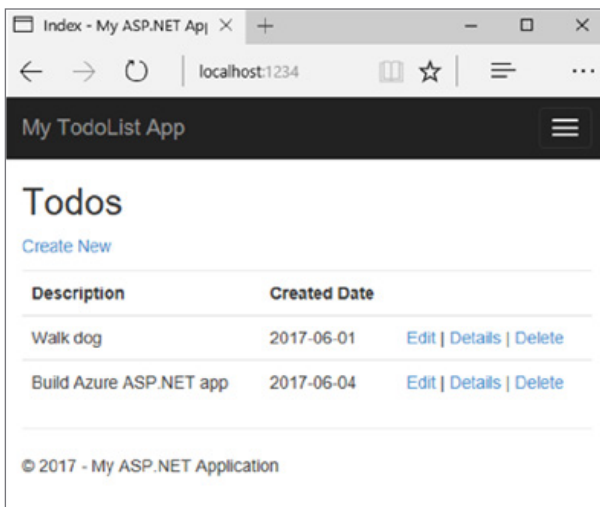


Figure 8-9

Connexion de l'application web locale à la base de données en exécution dans Azure

Vous disposez désormais d'une application en exécution locale. Avant de la déployer dans Azure, vous allez modifier le code source afin qu'il puisse se connecter à la base de données SQL.

1. Dans votre référentiel de code source local, recherchez le fichier **Startup.cs** et localisez le code suivant :

```
services.AddDbContext<MyDatabaseContext>
(options => options.UseSqlite
("Data Source=localdatabase.db"));
```

2. Remplacez le code par le code suivant, qui se connectera à la base de données Azure SQL :

```
// Use SQL Database if in Azure, otherwise,
use SQLite

if(Environment.
GetEnvironmentVariable("ASPNETCORE_
ENVIRONMENT") == "Production")

services.
AddDbContext<MyDatabaseContext>(options =>

options.UseSqlServer(Configuration.
GetConnectionString("MyDbConnection")));
else

services.
AddDbContext<MyDatabaseContext>(options =>

options.UseSqlite("Data
Source=localdatabase.db"));

// Automatically perform database migration
services.BuildServiceProvider().
GetService<MyDatabaseContext>().Database.
Migrate();
```

Ce code examine son environnement d'exécution et modifie sa connexion à la base de données en conséquence. Lorsqu'il est en exécution dans l'environnement de production (Azure, dans le cas présent), le code obtient la chaîne de connexion pour la base de données à partir de la variable `MyDbConnection`, que nous allons configurer dans Azure.

Le code exécute également la méthode `Database.Migrate()`, qui exécute les migrations Entity Framework Core que vous avez précédemment exécutées manuellement.

3. Enregistrez vos modifications, puis exécutez les commandes suivantes pour valider les modifications dans votre référentiel Git local :

```
git add .
git commit -m "connect to SQLDB in Azure"
```

Nous allons maintenant configurer la variable de chaîne de connexion dans Azure.

4. Dans le portail Azure, accédez à l'application web que vous avez créée précédemment.
5. Accédez à **Paramètres de l'application**.

6. Créez une chaîne de connexion nommée **MyDbConnection**. La valeur doit correspondre à la chaîne de connexion vers la base de données SQL (y compris le nom d'utilisateur et le mot de passe) que vous avez enregistrée plus tôt lors de la création de la base de données.
7. Cliquez sur **Enregistrer**. Les paramètres de l'application dans le portail Azure devront être semblables à ceux de la figure 8-10.

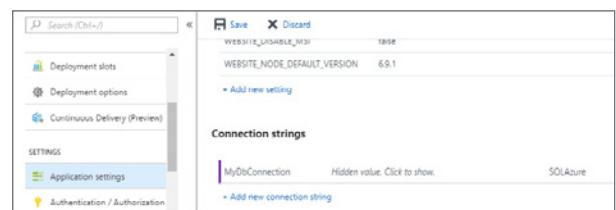
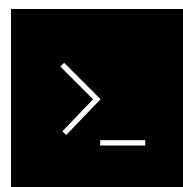


Figure 8-10

Déploiement de l'application web dans Azure

Vous utiliserez Git pour placer l'application dans Azure. Pour connecter le dépôt Git local à Azure, vous devez avoir configuré un utilisateur de déploiement sur le serveur (Azure Web App) afin d'authentifier votre déploiement. L'utilisateur du déploiement se trouve au niveau du compte. Il est différent de votre compte d'abonnement Azure. Vous ne devez configurer cet utilisateur de déploiement qu'une seule fois.

1. Dans le portail Azure, accédez à **Azure Cloud Shell** en sélectionnant le bouton dans la barre supérieure qui se présente comme suit :



2. Azure Cloud Shell vous permet d'utiliser Azure CLI dans le Cloud et gère l'authentification. Lorsque Cloud Shell est entièrement chargé, exécutez la commande suivante pour créer l'utilisateur de déploiement. Remplacez les valeurs `<username>` et `<password>` par celles que vous avez créées. Notez le nom d'utilisateur et le mot de passe parce que vous en aurez besoin ultérieurement.

```
az webapp deployment user set --user-name  
<username> --password <password>
```

3. Les résultats de la commande dans une sortie JSON. Si vous recevez un message d'erreur « **Conflict. Details: 409** », modifiez le nom d'utilisateur. Si vous recevez un message d'erreur « **Bad Request. Details: 400** », créez un mot de passe plus robuste.

Vous devez maintenant déplacer le code source de votre dépôt Git local vers l'application web Azure.

4. Ouvrez l'invite de commande sur votre ordinateur local.
5. Ajoutez une commande distante Azure à votre dépôt Git local en utilisant l'URL distante Git :
 - a. Remplacez la valeur `<username>` par le nom d'utilisateur que vous avez utilisé pour créer l'utilisateur de déploiement.
 - b. Remplacez la valeur `<app_name>` par le nom de l'application web Azure.
 - c. Utilisez l'URL pour exécuter la commande suivante :

```
git remote add azure <deploymentLocalGitUrl>
```

6. Une fois que la cible distante est ajoutée au dépôt Git, vous pouvez y placer votre code en exécutant la commande suivante. Vous devrez fournir vos informations d'identification pour être en mesure de placer le code dans Azure. Utilisez le nom d'utilisateur et le mot de passe que vous avez utilisés pour créer l'utilisateur de déploiement.

```
git push azure master
```

Le placement du code source dans Azure peut prendre quelques minutes la première fois que vous effectuez l'opération. Une fois l'opération terminée, accédez à l'URL de votre application web Azure, dont le format est le suivant :

`http://<app_name>.azurewebsites.net`

7. Ajoutez des éléments à faire dans l'application pour vérifier sa connexion à la base de données.

Vous disposez maintenant d'une application fonctionnelle en exécution sur Azure.

Procédure N° 3 : extension des applications avec Logic Apps et Cognitive Services

Une fonction puissante de notre application est sa capacité à analyser le contenu des éléments à faire, puis à créer automatiquement des rendez-vous dans le calendrier pour des tâches qui incluent une date précise.

Par exemple, si un utilisateur crée un élément à faire comportant le texte « dîner familial vendredi prochain à 19h », l'application crée un élément de calendrier pour ce vendredi spécifique à 19h, avec le sujet « dîner familial ».

Vous allez configurer cela en utilisant la fonction [Logic Apps](#) de Microsoft Azure App Service et [Language Understanding Intelligent Service \(LUIS\)](#), comme suit :

- L'application .NET Core écrit l'élément à faire dans la base de données SQL.
- L'application logique est déclenchée par chaque nouvelle ligne créée dans la base de données.
- L'application logique extrait le texte de l'élément à faire et le transmet au service Language Understanding.
- Le service Language Understanding analyse le texte et crée un élément de calendrier dans votre calendrier Office 365, si le texte contient une date et une heure.

Nous n'avons pas besoin de modifier notre application pour ajouter cette fonction.

Logic Apps et Cognitive Services sont des services supplémentaires qui analysent simplement les données déjà présentes.

C'est parti !

Création du service Language Understanding

Nous allons commencer par créer le service Language Understanding, afin de pouvoir l'utiliser ultérieurement dans notre application logique. Nous conserverons notre modèle pour cet exemple simple ; nous n'allons pas le générer de sorte qu'il soit prêt pour chaque variante dont les utilisateurs pourraient avoir besoin pour une date dans un élément à faire. Vous pouvez apporter des ajouts au modèle vous-même au lieu d'utiliser celui que nous avons créé.

1. Dans le portail Azure, sélectionnez **Créer un service**.
2. Recherchez **Language Understanding** et sélectionnez-le dans les résultats de recherche pour ouvrir le panneau du même nom. Sélectionnez **Créer**. Le panneau **de création de Language Understanding** s'ouvre.
3. Saisissez un nom.
4. Sélectionnez un niveau de tarification (n'importe lequel pour cette procédure).
5. Créez un groupe de ressources appelé **datedetection**.
6. Cliquez sur **Créer**.
7. Accédez au service Language Understanding une fois qu'il est créé.
8. Par défaut, le service s'ouvre dans le panneau **Démarrage rapide**. Là, sélectionnez **Portail Language Understanding**.
9. Si nécessaire, connectez-vous en utilisant **Se connecter** dans le coin supérieur droit.

10. Cliquez sur **Créer une application**.

11. Saisissez un nom.

12. Cliquez sur **Terminé**.

Nous sommes à présent dans le portail Language Understanding et nous pouvons développer un modèle de langage. Nous voulons que le service Language Understanding comprenne l'expression « dîner familial vendredi prochain à 19h ». Pour ce faire, nous allons tout d'abord ajouter certaines entités, c'est-à-dire des éléments dans le texte que le service reconnaîtra.

1. Cliquez sur **Entités**.
2. Cliquez sur **Gérer les entités prédéfinies**.
3. Sélectionnez **DatetimeV2** et **keyPhrase**.
4. Cliquez sur **Terminé**. Nous disposons maintenant de deux entités qui reconnaîtront le texte pour nous.
5. Cliquez sur **Intentions**.
6. Cliquez sur **Créer une intention**.
7. Saisissez un nom, par exemple, « Ajouter un élément de calendrier à faire », ce qui correspond à l'intention que nous voulons détecter dans le texte.
8. Cliquez sur **Terminé**.

Vous pouvez maintenant saisir des **énoncés**. Les énoncés sont des exemples de texte qui représentent l'intention que nous souhaitons détecter.

9. Saisissez « dîner familial vendredi prochain à 19h » pour représenter l'intention d'ajouter un élément à faire au calendrier.
10. Étant donné que nous avons déjà ajouté deux entités, le texte de l'énoncé est analysé et reconnu comme ces entités, comme illustré à la Figure 8-11. Le texte « dîner familial » est reconnu comme **keyPhrase**. Le texte « vendredi prochain à 19h » est reconnu comme valeur **datetimev2**.
11. Utilisons ce modèle pour l'apprentissage de service et publions-le. Sélectionnez **Apprentissage** dans le coin supérieur droit de l'écran.

keyPhrase	datetimev2
-----------	------------

Figure 8-11

- Cela effectue l'apprentissage Machine Learning et crée un modèle Machine Learning basé sur ce que nous venons de saisir.
 - Pour tester si le service fonctionne maintenant comme prévu, saisissez « dîner familial vendredi prochain à 19h » en le testant dans la fenêtre **Test** à proximité du bouton **Apprentissage**.
12. Maintenant que nous disposons d'un service opérationnel, nous devons publier ce modèle en production. Cliquez sur **Publier** dans le menu (à côté du bouton **Apprentissage**) pour ouvrir la page **Publier**.
 13. Laissez l'emplacement défini sur **Production**.
 14. Cliquez sur **Publier**.

15. Le modèle est maintenant publié en production. Faites défiler la page vers le bas pour afficher **Ressources et clés** et notez la chaîne de clé que vous y trouverez car nous en aurons besoin pour notre application logique.

Création de l'application logique

L'application logique que nous créons sera déclenchée par les nouvelles lignes d'éléments à faire écrites dans SQL Database. Elle enverra alors la valeur de chaque élément à faire au service Language Understanding pour analyse. Si le service Language Understanding détecte une date dans l'élément, il crée un événement de calendrier dans votre compte Office 365.

Créons l'application logique :

1. Dans le portail Azure, cliquez sur **Créer un service**.
2. Recherchez **Application logique**, puis cliquez dessus dans les résultats de recherche pour ouvrir le panneau **Application logique**. Cliquez sur **Créer**. Le panneau **Créer une application logique** s'ouvre.
3. Saisissez un nom.
4. Sélectionnez le groupe de ressources que vous avez créé pour le service Language Understanding.
5. Choisissez un emplacement.
6. Cliquez sur **Créer**.
7. Une fois l'application logique créée, une page de démarrage rapide s'affiche et vous invite à préciser si vous souhaitez démarrer l'application logique à partir d'un modèle (figure 8-12). Choisissez **Application logique vide**.

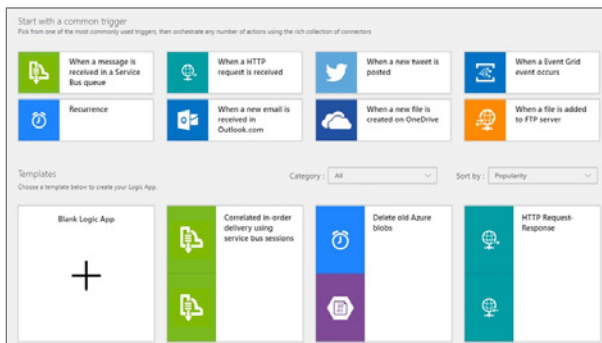


Figure 8-12

Nous devons maintenant créer un déclencheur pour l'application logique.

1. Recherchez SQL.
2. Sélectionnez la tâche **Lorsqu'un élément est créé**. Cette opération demande la connexion à la base de données SQL (figure 8-13).

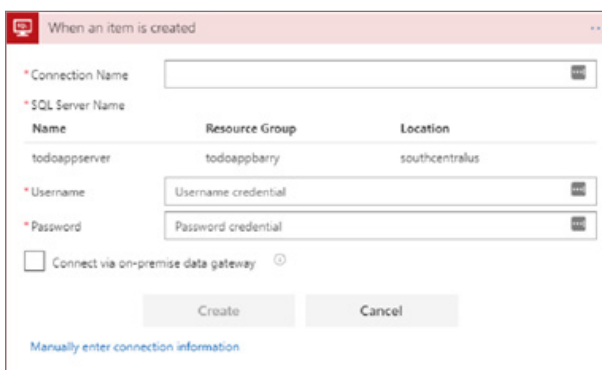


Figure 8-13

3. Dans ce cas, le SQL Server correct est déjà sélectionné, car il n'y en a qu'un seul. Vous devrez peut-être sélectionner le serveur approprié.
4. Entrez un nom pour la connexion, puis saisissez les valeurs nom d'utilisateur et mot de passe correspondant à la base de données SQL que vous avez créée précédemment.

5. Cliquez sur **Créer**. Cela crée la connexion et l'enregistre dans votre abonnement Azure. Vous pouvez réutiliser cette connexion dans d'autres applications logiques.
6. Sélectionnez la table que vous souhaitez contrôler (la table **À faire**).
7. Sélectionnez un intervalle et une fréquence. Certains déclencheurs d'application logique doivent interroger pour être déclenchés, alors que pour d'autres, les informations y sont placées.

L'application logique sera désormais déclenchée chaque fois que vous saisissez un nouvel élément à faire.

Nous ajouterons ensuite une action pour l'application logique.

1. Cliquez sur le **signe plus** sous la tâche SQL, puis sélectionnez **Ajouter une action** pour ajouter l'action suivante (figure 8-14).

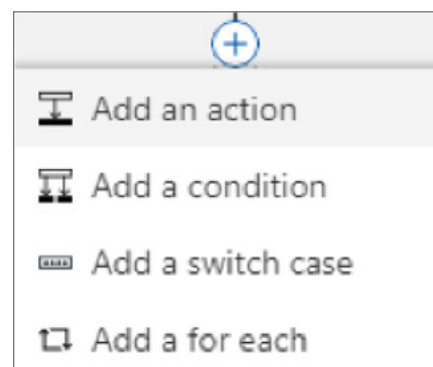


Figure 8-14

2. Recherchez **LUIS** pour afficher le service Language Understanding. Sélectionnez l'**action de recherche de prédiction LUIS**. Vous serez invité à fournir une connexion à un service Language Understanding.

3. Saisissez un nom pour la connexion.
4. Collez la clé de connexion que vous avez enregistrée lors de la publication du modèle Language Understanding.
5. Cliquez sur **Créer**.
6. Sélectionnez l'ID de l'application que vous avez créé sur le portail Language Understanding.
7. Sélectionnez la description de la tâche SQL comme valeur du champ **Énoncé**.
8. Sélectionnez l'élément **Ajouter un élément à faire au calendrier** pour l'intention souhaitée. Cela vous indiquera si la tâche comporte ou non une date.
9. Cliquez sur le **signe plus**, puis sur **Ajouter une condition**. Nous testerons si le texte contient une date en vérifiant si l'intention souhaitée était vraie. Si le texte contient effectivement une date, nous créerons un événement de calendrier. Dans le cas contraire, nous ne ferons rien.
10. Dans la condition, sélectionnez la valeur **Est l'intention souhaitée** à partir de la tâche Language Understanding pour la valeur.
11. Laissez l'instruction **est égal à** en l'état.
12. Ajoutez **vrai** dans la zone de texte Valeur.
13. La condition s'affiche dans les deux cases : **si vrai** et **si faux**. Dans la zone **si vrai**, créez une action.
14. Recherchez le LUIS comme nous l'avons fait précédemment.
15. Sélectionnez l'action **Obtenir l'entité par type**. Il s'agit d'une action Language Understanding qui extrait une entité des résultats Language Understanding en fonction de son type.
16. Sélectionnez l'ID de l'application comme nous l'avons fait précédemment.
17. Sélectionnez **builtin.datetimev2** pour l'entité souhaitée.
18. Sélectionnez l'objet **Prédiction LUIS** pour le champ **luisPredictionObject**.
19. Sous cette action, ajoutez-en une autre pour **Obtenir l'entité par type**.
20. Sélectionnez l'ID de l'application.
21. Sélectionnez **builtin.keyPhrase** pour l'entité souhaitée.
22. Sélectionnez l'objet **Prédiction LUIS** pour le champ **luisPredictionObject**.
23. Créez une action tant que vous êtes dans la zone **si vrai**. Recherchez **Office 365**, puis sélectionnez l'action **Créer un événement V2**. Cette opération peut créer un événement dans votre calendrier Office 365.
24. Veuillez noter que cela nécessite une connexion vers Office 365. Cliquez sur **Ajouter une nouvelle connexion**, puis connectez-vous avec vos informations d'identification Office 365. L'application logique conserve votre connexion dans votre abonnement Azure.
25. Sélectionnez le calendrier pour créer l'événement.
26. Dans les champs **Heure de fin** et **Heure de début**, sélectionnez la **valeur Entité** à partir de l'action où vous filtrez l'entité **datetimev2**.
27. Dans le champ **Sujet**, sélectionnez la **valeur d'entité** à partir de l'action où vous filtrez l'entité **keyPhrase**.

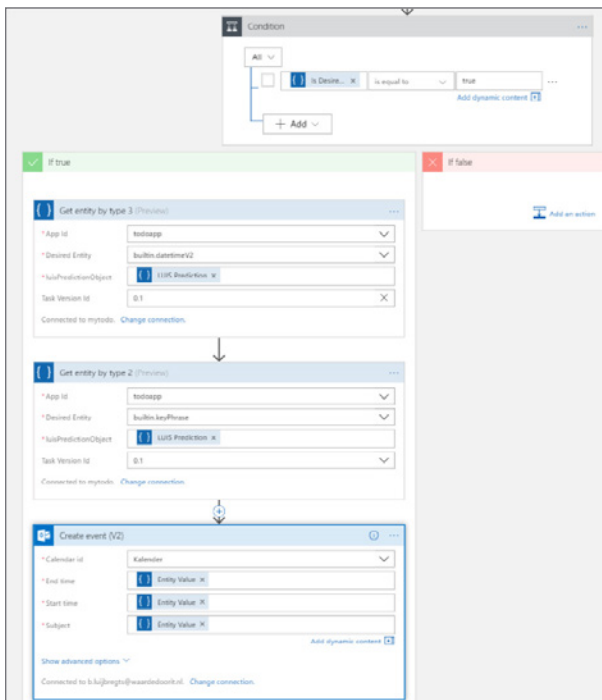


Figure 8-15

28. Enregistrez le flux de l'application logique.
La zone **si oui** doit ressembler à celle présentée à la figure 8-15.
29. Accédez à l'URL de l'application à faire, qui correspond à l'URL de votre application web de la procédure précédente.
30. Créez un élément à faire contenant le texte
« dîner familial vendredi prochain à 19h ». Cela doit créer un événement dans votre calendrier.

Outre la vérification de votre calendrier, vous pouvez voir comment l'application logique s'est exécutée en examinant l'**historique des exécutions**. Vous pouvez accéder à l'historique des exécutions lorsque vous ouvrez l'application logique à partir du portail Azure. A partir de l'historique des exécutions, vous pouvez resoumettre la valeur pour l'exécuter à nouveau par l'intermédiaire de l'application logique.

Cet exemple montre que vous pouvez étendre une application avec les services Azure simplement par l'intermédiaire de sa configuration, sans en modifier le code.

Nous avons fait en sorte que cet exemple reste simple, afin qu'il soit facile à suivre dans cette procédure. Dans un scénario réel, le modèle Language Understanding doit être plus robuste pour être en mesure de comprendre davantage d'énoncés. En outre, vous pouvez faire en sorte que l'application logique se déclenche lors des modifications des éléments à faire, et non uniquement lors de leur création.

Procédure pas à pas N° 4 : mise en production

Maintenant que votre application est en cours d'exécution, vous pouvez utiliser Azure pour la rendre plus robuste et plus facile à mettre à jour.

Configuration d'une livraison en continu avec GitHub

Jusqu'à présent, nous avons transmis du code de notre dépôt Git local vers Azure. Vous pouvez effectuer cette opération facilement si vous travaillez seul, mais si vous faites partie d'une équipe, vous avez besoin d'un autre type de contrôle des sources, comme Azure DevOps Repos ou GitHub.

À l'aide de GitHub, nous allons transmettre notre code à notre application web, puis les associer de sorte que les modifications soient automatiquement déployées via un pipeline de livraison continue.

C'est parti !

1. Connectez-vous depuis <https://github.com/new> pour créer un dépôt sur GitHub.
2. Attribuez un nom au dépôt.
3. Laissez les autres paramètres tels quels (dépôt public, ne créez pas de README).
4. Créez le dépôt, qui doit ressembler à celui présenté à la figure 8-16.

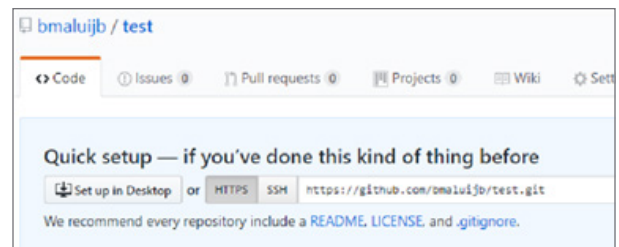


Figure 8-16

5. Utilisez l'URL qui s'affiche à l'écran pour définir une destination distante pour le dépôt Git local. Pour ce faire, vous pouvez utiliser la fenêtre de commande.
6. Utilisez la commande CD pour modifier le répertoire du code source de l'application.
7. Exécutez la commande suivante :

```
git remote add github https://github.com/  
bmaluijb/test.git
```

8. Exécutez la commande suivante pour transmettre le code vers GitHub :

```
git push github
```

Suite à cette opération, le code est accessible dans GitHub, et vous pouvez le partager avec votre équipe.

Configurons à présent la livraison continue en utilisant la fonction Options de déploiement de Web Apps via le portail Azure. Notez que nous pouvons également utiliser la fonction Continue dans Web Apps directement, mais cela nécessite de disposer d'un compte Azure DevOps.

1. Dans le portail Azure, accédez à l'application web qui héberge l'application à faire .NET Core.
2. Dans la barre de menu, cliquez sur **Options de déploiement**.
3. Il est possible que la configuration soit déjà réalisée pour le dépôt Git local. Si c'est le cas, cliquez sur **Déconnecter**.
4. Dans **Choisir une source**, sélectionnez **GitHub**.
5. Dans la section **Autorisation**, autorisez Azure à utiliser GitHub. Pour ce faire, sélectionnez **Autoriser** et accordez l'autorisation.
6. Dans la section **Choisir un projet**, sélectionnez le dépôt GitHub que vous venez de créer.
7. Laissez la branche définie sur **maîtresse**.
8. Cliquez sur **OK**.
9. Revenez au menu **Options de déploiement**. Comme vous pouvez le constater, la connexion à GitHub a bien été réalisée. Dorénavant, chaque fois que vous transmettez une nouvelle version du code source vers GitHub, elle sera automatiquement développée et déployée sur l'application web. Cela est illustré dans la figure 8-17 qui présente le panneau **Options de déploiement**. Vous pouvez également forcer ce processus en cliquant sur **Sync**.

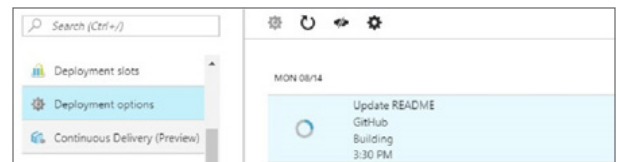


Figure 8-17

Configuration des environnements de tests ou de préproduction

Grâce à Web Apps dans Azure App Service, vous pouvez configurer un emplacement de préproduction pour tester les nouvelles versions de votre application via des emplacements de déploiement. Les emplacements de déploiement sont des App Services que vous pouvez utiliser pour tester votre code avant de le basculer vers l'emplacement suivant.

Il existe des emplacements de déploiement pour la préproduction, les tests de charge et la production (il s'agit toujours de l'App Service d'origine, donc dans le présent exemple de l'application web .NET Core). En fait, vous pouvez définir autant d'emplacements de déploiement que vous le souhaitez sans générer de frais supplémentaires.

Les emplacements de déploiement sont tous exécutés dans le même plan App Service. Ce sont ces éléments qui vous sont facturés. Gardez à l'esprit que plus vous ajoutez d'emplacements de déploiement dans un plan App Service, plus vous consommerez de ressources comme l'UC et la mémoire.

Pour créer des emplacements de déploiement, cliquez sur **Emplacements de déploiement** dans le menu de l'application web. Vous devez exécuter l'application web avec un niveau de tarification standard ou premium, car l'option gratuite du plan ne propose aucun emplacement de déploiement.

Dans chaque emplacement de déploiement que vous créez, vous pouvez configurer les options de déploiement en suivant la procédure décrite précédemment pour déployer le code automatiquement. Vous pouvez même travailler sur différentes branches de code source pour différents environnements et déployer automatiquement des branches spécifiques aux emplacements de déploiement de votre choix.

En outre, vous pouvez tester votre version finale dans un emplacement de déploiement, puis l'échanger avec la version dans l'emplacement de production. Vous pouvez ainsi éprouver l'application avant l'échange et la déployer sans temps d'arrêt.

Nous allons voir comment créer un emplacement de déploiement et effectuer un échange vers celui-ci.

1. Dans le portail Azure, accédez à l'application web qui héberge l'application .NET Core.
2. Dans la barre de menu, cliquez sur **Emplacements de déploiement**. Le panneau **Emplacements de déploiement** s'affiche.
3. Cliquez sur le **signe plus** pour créer un emplacement de déploiement.
4. Saisissez un nom pour l'emplacement (par exemple, « préproduction »).
5. Choisissez l'**application web .NET Core** comme source de configuration. Cette opération copie les paramètres de l'application vers le nouvel emplacement.
6. Cliquez sur **OK** pour créer l'emplacement, qui est similaire à l'application web d'origine.
7. Configurez la livraison continue en suivant la même procédure que pour l'application web.
8. Désactivez la connexion avec la livraison continue dans l'application web .Net Core d'origine. De cette façon, lorsque vous transmettez un nouveau code, il est diffusé uniquement dans l'emplacement de préproduction.

9. Dans l'application .NET Core, modifiez une partie du texte dans le fichier **Index.cshtml** du dossier **Views/Home**.
10. Validez votre modification dans Git et transmettez la nouvelle version vers GitHub, de la même manière que lorsque vous avez déployé l'application .NET Core.

La nouvelle version est maintenant dans l'emplacement de préproduction et non dans l'application web d'origine (que nous appelons l'emplacement de production). Pour le vérifier, accédez à l'URL de l'application web .NET Core et à l'URL de l'emplacement de préproduction (que vous trouverez dans le panneau **Présentation** de l'emplacement).

Plaçons à présent la nouvelle version dans l'emplacement de production.

1. Dans le portail Azure, accédez à l'application web .NET Core.
2. Dans la barre de menu, sélectionnez **Emplacements de déploiement** pour ouvrir le panneau **Emplacements de déploiement**.
3. Cliquez sur **Échanger** pour ouvrir le panneau correspondant. Ne modifiez aucun des paramètres.
4. Cliquez sur **OK** pour démarrer l'échange.

Une fois l'échange terminé, la nouvelle version de l'application web .NET Core est en production. Vous pouvez la tester en accédant à l'URL de l'application web Node.js. Il s'agit là d'une utilisation avantageuse des emplacements de déploiement. En effet, vous pouvez ainsi y tester les nouvelles versions avant de les déployer en production. En outre, le déploiement est alors réalisé sans temps d'arrêt.

Utilisation des journaux de diagnostic

Un moyen efficace de surveiller une application consiste à utiliser les journaux de diagnostic pour voir la journalisation des diagnostics en direct depuis l'application web. Vous pouvez même afficher ces journaux dans la fenêtre de console. Pour ce faire, exécutez la commande suivante dans Azure Cloud Shell :

```
az webapp log tail --name <app_name>
--resource-group <myResourceGroup>
```

Lorsque vous utilisez l'application au niveau de l'application web pour générer du trafic, des entrées sont ajoutées aux journaux.

Configuration de la surveillance et des alertes

[Azure Monitor Application Insights](#) fournit un autre moyen extrêmement efficace d'effectuer le suivi d'applications. Il s'agit d'un outil de surveillance qui vous tient informé de plusieurs aspects de votre application. Par exemple, combien de visiteurs l'ont utilisée ou encore combien d'exceptions se sont produites, ainsi que les parties du code concernées. Contrairement aux journaux de diagnostic, Application Insights est disponible pour un prix modique.

Configurons Application Insights.

1. Accédez au portail Azure, puis à l'application web qui héberge l'application .NET Core.
2. Dans la barre de menu, cliquez sur **Application Insights**.
3. Sélectionnez **Créer une ressource**.
4. Saisissez un nom et sélectionnez un emplacement pour l'instance Application Insights.

5. Cliquez sur **OK**. Application Insights est à présent déployée et commence à collecter les données de l'application.

Vous devez maintenant configurer votre application pour envoyer des données à Application Insights.

6. Dans le menu Visual Studio Code, sélectionnez **Projet > Ajouter Application Insights Telemetry...**
7. Cette opération ouvre l'assistant Application Insights. Connectez-vous via votre compte Azure.
8. Sélectionnez un plan de tarification Application Insights.
9. Cliquez sur **S'inscrire**. Cette opération ajoute automatiquement tout ce dont vous avez besoin pour le projet .NET Core et crée la ressource Application Insights dans Azure.
10. Créez le projet et transmettez les modifications vers GitHub de sorte qu'elles soient déployées vers l'application web. Une fois le déploiement terminé, l'application envoie des données à Application Insights.

Par défaut, Application Insights effectue une détection intelligente. Cette fonction détecte les anomalies, par exemple, les augmentations subites de demandes non abouties ou les ralentissements inhabituels de l'application, et vous prévient. Vous pouvez également créer vos propres événements personnalisés pour toutes sortes de mesures et de conditions dans le menu **Alertes** d'Application Insights, comme illustré dans la figure 8-18.



NAME	CONDITION	LAST ACTIVE
MYOTHUBTEST (COMPONENTS)		
Failed requests per second	Failed Requests per Second > 1	Review

Figure 8-18

11. Pour vérifier si Application Insights fonctionne correctement, accédez au portail Azure, trouvez la ressource Application Insights, et sélectionnez-la. Un récapitulatif s'affiche. Il vous présente des mesures de base comme le temps de réponse du serveur, le temps de chargement des pages consultées, le nombre de demandes envoyées au serveur et le taux de demandes non abouties. Vous devez voir des données indiquant qu'Application Insights fonctionne correctement.

Mise à l'échelle de l'application web

Si vous avez beaucoup d'utilisateurs, vous devez mettre l'application web à l'échelle afin qu'elle puisse gérer l'augmentation du volume de trafic. Lorsque le nombre d'utilisateurs est réduit, vous devez effectuer une nouvelle mise à l'échelle pour réduire les coûts. Pour ce faire, vous pouvez utiliser la fonction **Mise à l'échelle automatique** d'App Service. Pour utiliser cette fonctionnalité, vous devez exécuter Web Apps avec un niveau de tarification standard ou premium.

Web Apps comporte un élément de menu appelé **Monter en charge**, comme illustré dans la figure 8-19. Elle vous permet de mettre votre application à l'échelle manuellement ou automatiquement. La montée en charge consiste à ajouter des instances de votre application pour gérer la charge.

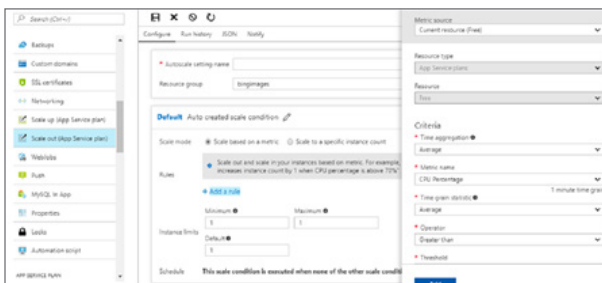


Figure 8-19

Ajout d'un protocole SSL

Lorsqu'une application est prête à entrer en phase de production, vous devez confirmer qu'elle est sécurisée. Outre l'authentification et l'autorisation, la mise à disposition de l'application web via HTTPS est l'une des mesures de sécurité les plus importantes que vous pouvez appliquer. En effet, sans HTTPS, des intrus peuvent observer le trafic entre vos ressources et utiliser ces informations à des fins malveillantes (comme se connecter à votre application). De plus, HTTPS constitue une exigence pour certaines fonctions de pointe comme les [workers du service](#).

L'acheminement de trafic vers votre application web au moyen du protocole SSL est possible en important un certificat SSL dans Web Apps et en le liant à l'un de vos noms de domaine personnalisés. Vous pouvez importer votre propre certificat SSL ou en acheter un via [Azure App Service Certificates](#). Ce service facilite l'achat et la validation des certificats. Après avoir importé le certificat, vous l'associez à l'une des liaisons de nom de domaine de votre application web. Vous pouvez effectuer ces opérations depuis le menu **Certificats SSL** de l'application web.

Notifier les utilisateurs à propos des nouvelles versions

Si vous informez les utilisateurs des lancements de nouvelles versions, cela sera bénéfique pour votre entreprise. L'extension du processus CI/CD (intégration continue/livraison continue) dans les builds Azure vous permet d'utiliser un flux de travail Logic Apps pour gérer la communication sur les réseaux sociaux, par exemple l'envoi de tweets ou la publication de billets contenant des notes de publication.

09 /

Résumé et ressources

Le présent guide a mis en avant les avantages offerts par Azure à vos applications. Azure vous permet de réaliser des opérations incroyables avec vos applications, qu'il s'agisse de mettre en œuvre un système de reconnaissance faciale et vocale, de gérer vos appareils IoT dans le Cloud, d'effectuer autant de mises à l'échelle que vous le souhaitez et de payer uniquement à l'utilisation.

Vous avez appris qu'Azure a des services pour presque tous les scénarios, et qu'il peut donc vous aider quel que soit le langage de programmation utilisé ou la plateforme créée pour vos applications. Nous espérons que vous continuerez à consulter cet Ebook pour vous familiariser davantage avec les nombreux services Azure et pour déterminer ceux qui correspondent le mieux à vos besoins.

Grâce à la richesse des solutions prédéfinies dans Azure, vous n'êtes désormais plus tenu d'écrire le code compliqué des solutions. Libérez-vous du temps pour travailler à ce qui est vraiment important pour vous en tirant profit de toutes les options d'Azure.

Poursuivez votre apprentissage avec Azure

Avec votre [compte Azure gratuit](#), vous bénéficiez de tout cela et vous ne serez facturé(e) que si vous optez pour une mise à niveau :

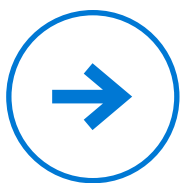
- 12 mois de services populaires gratuits.
- Crédit de 200 \$ pour découvrir le service Azure de votre choix pendant 30 jours.
- + de 25 services toujours gratuits.
- [Prise en main d'Azure](#) : regardez ces courts tutoriels sur le mode d'utilisation d'Azure et commencez à développer des projets immédiatement. Vous pouvez également vous associer à notre [webcast hebdomadaire](#) qui propose des démonstrations des fonctionnalités de base Azure et donne accès à des experts en permanence.
- [Microsoft Learn](#) : acquérez de nouvelles compétences et découvrez la puissance des produits Microsoft grâce à des instructions détaillées. Lancez-vous dès aujourd'hui en explorant nos modules et parcours d'apprentissage.
- [Azure Friday](#) : consultez Azure Services et les fonctions avec l'équipe d'ingénierie de Microsoft.
- [Azure.Source](#) : tenez-vous au courant des nouveautés Azure, notamment les actualités et les mises à jour, les produits en préversion, les produits disponibles au grand public.
- [Conseils et astuces Azure](#) : parcourez un ensemble d'idées utiles vous permettant d'optimiser votre productivité avec Azure.



Festival de ressources gratuites

En plus de ce guide, de nombreuses autres ressources gratuites concernant Azure sont disponibles, dont les suivantes :

- [Learn Azure in a Month of Lunches](#) : une manière pratique pour découvrir Azure en un mois de déjeuners.
- [Azure Serverless Computing Cookbook](#) : Ebook sur tout ce qui est sans serveur dans Azure.
- [Designing Distributed Systems](#) : Ebook sur le développement d'applications conteneurisées, avec des ateliers pratiques en ligne sur Azure Kubernetes Service.
- [Containerize Your Apps with Docker and Kubernetes](#) : un guide pratique pour déplacer des applications dans le Cloud avec Docker et Kubernetes.
- [Guide to NoSQL with Azure Cosmos DB](#) : Ebook sur le développement d'applications dynamiques, stratégiques avec Azure Cosmos DB.
- [Effective DevOps](#) : un guide pratique guide pour l'amélioration de la collaboration entre les équipes, la promotion de l'utilisation efficace des outils et l'utilisation des concepts DevOps.
- [Developers Guide to IoT](#) : Ebook qui fournit une vue d'ensemble des services Azure IoT et facilite votre prise en main.
- [Azure for Architects](#) : un guide complet pour les architectes Azure.
- [Developer's Guide to Building AI Applications](#) : un guide pratique pour la création de votre premier bot intelligent avec l'IA.
- [Designed to Disrupt](#) : inspiration et aide sur la possibilité de créer et d'appliquer des modifications transformationnelles.
- [Practical Microsoft Azure IaaS](#) : conseils et meilleures pratiques sur la manière de migrer des systèmes locaux dans le Cloud avec Azure.
- [Enterprise Cloud Strategy](#) : méthodes reconnues pour faire évoluer votre entreprise vers une stratégie de cloud computing.
- [Cloud Migration Essentials](#) : Ebook sur la manière de simplifier votre transition vers le cloud, tout en minimisant les risques et l'impact pour votre entreprise.
- [Making the Most of the Cloud Everywhere](#) : Ebook qui se concentre sur des pratiques unifiées de développement et de modernisation dans des environnements hybrides.



À propos des auteurs

Michael et Barry sont des passionnés d'Azure et vous encouragent à communiquer avec eux sur Twitter pour toute question concernant ce livre.



Michael Crump travaille chez Microsoft sur la plateforme Azure. Il est codeur, blogueur et conférencier international sur divers sujets liés au développement dans le Cloud. Il aime faire découvrir de façon pragmatique les avantages du Cloud aux développeurs.

Vous pouvez contacter Michael sur Twitter [@mbcrump](https://twitter.com/mbc Crump), suivre son blog à l'adresse suivante <https://www.michaelcrump.net/>, ou vous tenir au courant via un billet récent dans la série [Conseils et astuces Azure](#).



Architecte et développeur logiciel indépendant, **Barry Luijbregts** est un passionné du Cloud et est l'auteur de cours pour Pluralsight.

Vous pouvez contacter Barry sur Twitter [@AzureBarry](https://twitter.com/AzureBarry) et par l'intermédiaire de son site web accessible à l'adresse <https://www.azurebarry.com/>.

PUBLIÉ PAR Microsoft Press, une division de Microsoft Corporation
One Microsoft Way, Redmond, Washington 98052-6399

Copyright © 2019 Microsoft Corporation. Tous droits réservés. Aucune partie du contenu de ce livre ne peut être reproduite ni transmise sous quelque forme ou par quelque moyen que ce soit, sans l'autorisation écrite de l'éditeur.

Les livres Microsoft Press sont disponibles auprès des libraires et distributeurs du monde entier. Si vous avez besoin d'aide au sujet de ce livre, contactez le support Microsoft Press à l'adresse msspinput@microsoft.com. Nous vous remercions de nous donner votre avis sur ce livre en participant à cette [enquête](#).

Le livre en question est fourni « en l'état » et reflète les opinions des auteurs. Les informations et les points de vue exprimés dans cet ouvrage, y compris les adresses URL et les références à d'autres sites Internet, peuvent faire l'objet de modifications sans préavis. Certains des exemples présentés sont fictifs et sont seulement fournis à des fins d'illustration. Toute ressemblance ou tout lien avec des situations existantes ou ayant existé est fortuit.

Microsoft et les marques commerciales répertoriées sur la page web « Marques commerciales », accessible à l'adresse www.microsoft.com, sont des marques du groupe Microsoft. Toutes les autres marques sont détenues par leurs propriétaires respectifs.

