

Guide d'Azure pour les développeurs

Deuxième édition



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

Copyright © 2018 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 mspinput@microsoft.com. Veuillez nous faire part de votre avis sur ce livre à l'adresse <http://aka.ms/tellpress>.

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 « Marques commerciales » accessible à l'adresse <http://www.microsoft.com> sont des marques du groupe Microsoft. Toutes les autres marques sont détenues par leurs propriétaires respectifs.

Sommaire

Chapitre 1 : Guide d'Azure pour les développeurs	1
Introduction	2
L'informatique : un service, pas un fardeau.....	2
L'informatique sous forme de service.....	2
Types de déploiement de Cloud computing	4
Cloud public	4
Cloud privé	4
Cloud hybride	5
Démystifier les types de Cloud computing	5
Infrastructure en tant que service.....	6
Plate-forme en tant que service.....	6
Fonction en tant que service	6
Logiciel en tant que service	7
Pourquoi Azure ?	7
Azure a une portée mondiale	9
Azure est une solution extrêmement résiliente	9
Azure est conforme à pratiquement tous les secteurs d'activités	10
Azure se concentre sur les fonctionnalités pour les développeurs.....	10
Azure est une solution ouverte qui prend en charge les infrastructures de votre choix	10
Vous pouvez surveiller vos services Azure pendant vos déplacements.....	11
Nous sommes là pour vous aider	11
Chapitre 2 : Mise en route avec Azure.....	12
Vos IDE et éditeurs sont les bienvenus.....	12
Utilisez l'interface de ligne de commande d'Azure pour la création de scripts	12
Exécutez ce que vous voulez dans Azure.....	13
Qu'en est-il des coûts ?	13
Sélection des services Azure appropriés	13

Où héberger votre application	13
Pourquoi et quand l'utiliser ?	19
Accélérer votre application	20
Azure Content Delivery Network	20
Azure Redis Cache	20
Azure Traffic Manager	21
Où stocker vos données	22
Chapitre 3 : Ajout de fonctions intelligentes à votre application	27
Azure Search	27
Azure Cognitive Services	28
Azure Bot Service	28
Utilisation des événements et des messages dans votre application	29
Azure Service Bus	29
Azure Event Hubs	30
Azure IoT Hub	30
Azure Event Grid	31
Pourquoi et quand l'utiliser ?	32
Chapitre 4 : Sécurisation de votre application	33
Azure Active Directory	33
Gestion des API Azure	34
Azure Key Vault	34
Protection Azure DDoS	34
Pare-feu d'applications Web Azure	35
Chiffrement par défaut des données	35
Chapitre 5 : Où et comment déployer vos services Azure	36
La mentalité DevOps	36
Modèles Azure Resource Manager	37
Azure Service Fabric	37
Conteneurs dans Azure	38
Azure Stack	39
Où effectuer le déploiement et quand ?	39
Chapitre 6 : Une visite guidée d'Azure	40
Visite guidée n°1 : le portail Azure	40
Vignettes du portail Azure	40
Création d'une machine virtuelle	42
Visite guidée n°2 : développement d'une application Web Node.js avec MongoDB sur Azure	46
Création de l'application Web et de la base de données par l'intermédiaire du portail Azure	46
Procédure pas à pas n° 3 : création d'un back-end pour votre application mobile avec Azure	55

Créer une application mobile via le portail Azure.....	55
Fonctionnalités supplémentaires et mise en production.....	59
Chapitre 7 : Utilisation d'Azure Marketplace.....	60
En quoi Azure Marketplace constitue-t-il un avantage pour un développeur ?.....	60
Solutions Azure.....	61
Résumé et prochaines étapes	61
Continuer à apprendre avec un compte Azure gratuit.....	61
À propos des auteurs.....	62

Guide d'Azure pour les développeurs

La demande en logiciels et services connexes n'a jamais été aussi forte. Et cette tendance ne fera que s'accroître, de façon exponentielle, au fur et à mesure que des applications de toutes sortes s'immisceront dans tous les aspects de notre quotidien. En tant que développeur, vous êtes le « magicien » qui apporte cette révolution numérique à vos utilisateurs, et ce n'est pas facile. En plus de créer les fonctions tangibles uniques qu'offrira votre logiciel, vous devez vous charger de toute une série de qualités « invisibles » : vos applications doivent présenter une disponibilité élevée, rester performantes dans n'importe quel coin du monde, opérer de façon transparente sur un large choix d'appareils, être sécurisées et fonctionner à grande échelle pour un prix raisonnable. Ces défis sont communs à la plupart des logiciels et des secteurs d'activités. Les solutions, en revanche, sont éphémères et nous les réinventons en permanence. Leur conception et leur maintenance peuvent s'avérer complexes et coûteuses.

Nous avons élaboré ce *Guide d'Azure pour les développeurs* afin de faciliter votre migration vers le Cloud, que vous l'envisagiez à peine ou que vous ayez pris votre décision et amorcé la transition. Cet e-book a été rédigé par des développeurs pour les développeurs. Il a été spécifiquement conçu pour vous transmettre, en tant que développeur, les connaissances fondamentales relatives à Azure, ses avantages pour vous et pour votre entreprise et la façon d'en tirer pleinement parti.

La seconde moitié du guide vous propose différents scénarios, comme une visite du portail Azure ou la création d'une machine virtuelle. Nous abordons également le développement et le déploiement d'une application Web qui utilise Node.js et MongoDB. Nous passons en revue les tâches types comme CI/CD (intégration continue et déploiement continu), les environnements de tests ou de préproduction, la mise à l'échelle, la journalisation et le pilotage. Pour finir, nous créons un « backend » pour votre application mobile, incluant l'authentification et la synchronisation hors ligne.

Introduction

Les vertus du Cloud computing font l'objet d'un nombre considérable de livres, d'articles, de publications, de blogs, etc. Vous pouvez également trouver des zéta-octets d'informations techniques sur le Cloud computing. Mais avant de vous plonger dans toutes ces données, posez-vous quelques questions simples : pourquoi devrais-je migrer vers le Cloud ? Quels seront les avantages d'une telle migration pour moi, pour mon équipe et pour mon entreprise ?

L'informatique : un service, pas un fardeau

Le Cloud computing, c'est de l'informatique sous forme de service. Qu'entendons-nous par-là ? Pour vous aider à visualiser ce concept, imaginez un autre service omniprésent dans votre vie : l'électricité.

Qui penserait, aujourd'hui, que l'installation d'une mini-centrale électrique devant chaque domicile ou entreprise serait un modèle efficient pour la distribution d'électricité ? Nous sommes habitués à appuyer simplement sur un interrupteur pour allumer une lampe, puis à nous adonner à nos activités. Jamais, dans vos rêves, vous n'imagineriez devoir acheter, installer et entretenir tous les équipements requis pour éclairer votre domicile ou votre bureau. Au lieu de cela, vous payez votre compagnie d'électricité locale pour la puissance que vous consommez tous les mois. Vous n'avez jamais à vous préoccuper des principes physiques liés à la génération de l'électricité, de la gestion du capital, de la maintenance des équipements, de leur réparation après une tempête, rien de tout cela. Contentez-vous d'appuyer sur l'interrupteur et d'être productif dans vos propres activités.

De même que vous n'avez ni l'envie ni la nécessité de savoir précisément comment cette électricité est acheminée jusque chez vous, vous n'avez pas besoin de savoir exactement comment l'informatique et les services que vous consommez dans le Cloud sont mis à votre disposition. Quelqu'un d'autre s'occupe de cela pour vous. Ainsi, avant tout, la migration de votre informatique, de votre stockage et de votre réseau vers le Cloud vous apporte une facilité accrue, comme lorsque vous allumez une lampe. De plus, vous payez uniquement ce que vous utilisez, de même que vous payez uniquement pour l'électricité que vous utilisez chez vous.

Bien sûr, au-delà de notre analogie, le Cloud offre de nombreux avantages plus spécifiques pour vos opérations informatiques, comme l'évolutivité et la possibilité d'utiliser des services intelligents comme la reconnaissance faciale et la recherche dans vos applications, pour n'en citer que deux.

L'informatique sous forme de service

Donc, en fin de compte, vous pouvez comparer les services Cloud tels qu'Azure à *l'informatique sous forme de service*. Bien sûr, Azure offre beaucoup plus en termes de produits et services que votre fournisseur local en électricité. C'est pourquoi le Cloud computing est parfois difficile à définir de façon claire. En fait, certains affirment que le Cloud se contente d'utiliser *les ordinateurs d'un autre*, ce qui est techniquement vrai, mais le Cloud est tellement plus :



Le Cloud est un système qui fournit des services **intelligents, robustes et flexibles**, ainsi que **des ressources informatiques illimitées, élastiques et globales**.

Examinons de plus près tous ces qualificatifs :

Le Cloud et Azure fournissent des services qui vous aident à accomplir tant de choses : du plus banal, comme l'ajout d'une fonction de recherche dans votre application, au plus exotique, comme la mise en œuvre de processus d'intégration continue (CI) et de déploiement continu (CD). Vous pouvez automatiquement paramétrer votre base de données ou configurer des notifications push vers vos appareils mobiles, de façon simple et rapide. Ce ne sont là que quelques exemples d'opérations courantes que de nombreux développeurs ont créées pour eux-mêmes mais qui sont désormais disponibles sous forme de services. Vous pouvez ainsi utiliser ces services sans effort, un peu comme lorsque vous appuyez sur un interrupteur pour allumer la lumière ! Vous pouvez alors vous concentrer sur les parties de votre application qui la rendent unique, à savoir les fonctions qui apportent une réelle valeur ajoutée à vos utilisateurs.

Outre les services, le Cloud offre des ressources informatiques sous forme de machines virtuelles, conteneurs, bases de données, etc. Vous pouvez utiliser ces différents supports pour héberger des applications ou pour fournir une infrastructure complète à vos utilisateurs.

Comme mentionné plus haut, le Cloud puise toute sa puissance dans la robustesse et la résilience de ses services et de ses ressources. Il est très improbable que leur exécution n'échoue. Ceci est dû au fait que le Cloud est intelligent. En cas de problème, leur rétablissement est automatique. Et dans le cas d'Azure par exemple, les datacenters sont répartis dans le monde entier, avec plusieurs dizaines de milliers de serveurs. Si un serveur tombe en panne, un autre prend le relais. Si un datacenter entier devait tomber en panne (un scénario hautement improbable en soi), un autre datacenter prendrait le relais. Tout cela est possible en raison de l'échelle massive du Cloud.

L'un des arguments les plus convaincants en faveur du Cloud est que vous pouvez étendre vos services et vos ressources pratiquement à l'infini et certainement à un degré que ne permettent pas les ressources sur site, sauf si vous êtes prêt à dépenser d'énormes sommes d'argent en équipements et en personnel pour administrer tout cela. De plus, vous pouvez les étendre à l'échelle mondiale. Vous pouvez proposer vos services n'importe où dans le monde afin de fournir une expérience performante à vos utilisateurs, où qu'ils soient. Cela signifie également que vous pouvez conserver vos données là où vous en avez besoin.

Mais surtout, et c'est sans doute le principal avantage lorsque vous utilisez des ressources dans le Cloud, vous pouvez réduire vos services et ressources en cas de baisse de la demande. Pour revenir à notre analogie avec la distribution d'électricité : si vous invitez des amis et votre famille pour une grande fête, vous allez allumer un grand nombre de lampes à l'intérieur et à l'extérieur de votre maison, utiliser votre micro-ondes de façon répétée, mettre de la musique et donc, de façon générale, consommer davantage d'électricité. Mais une fois la fête terminée et vos invités partis, votre consommation d'électricité reviendra à son niveau normal, de même que votre facturation. Il en va de même lorsque vous tirez parti du Cloud et d'Azure : vous payez uniquement pour ce que vous utilisez, pas pour ce que vous pourriez utiliser en cas de forte demande.



Plus d'informations Pour en savoir plus sur le portail Azure et créer votre première machine virtuelle, visitez le site <https://docs.microsoft.com/azure/virtual-machines/windows/quick-create-portal>.

En plus de son évolutivité massive, de ses services intelligents et de l'efficacité du paiement à l'utilisation, le Cloud offre une sécurité accrue.

Le Cloud est utilisé par des millions de personnes, 24 heures sur 24 et 7 jours sur 7, dans le monde entier. Évidemment, il fait aussi l'objet de nombreuses attaques. Les fournisseurs de Cloud réputés et expérimentés comme Microsoft connaissent les comportements d'utilisation normaux des utilisateurs. Ils savent les différencier des comportements malveillants. Ils savent donc également comment protéger le Cloud contre les attaques, les plus courantes comme les plus originales. Grâce à des outils de surveillance intelligents, à des algorithmes d'apprentissage automatique et à l'intelligence artificielle, les fournisseurs de Cloud sont en mesure de détecter et de contrer les attaques en temps réel.

[Des décennies d'expérience en matière de sécurité](#) et de trafic à très grande échelle, mais également la meilleure expertise du secteur en matière de sécurité font du Cloud un environnement beaucoup plus sécurisé que n'importe quel datacenter sur site.



Plus d'informations Pour découvrir comment Azure protège vos applications et vos données, consultez les pages [Comment Azure Security Center détecte les attaques de déni de service distribué \(DDoS\) grâce aux informations sur les cybermenaces](#), [Présentation de la sécurité avec Azure](#) et [Comment Microsoft gère la cybercriminalité](#).

Nous avons exploré brièvement les raisons pour lesquelles vous devriez amorcer votre migration vers le Cloud et Azure. Examinons maintenant les principaux composants du Cloud, leur fonction et leur place dans le tableau global.

Types de déploiement de Cloud computing

Le Cloud offre différents niveaux de produits et de services, comme l'infrastructure en tant que service (IaaS) ou la plate-forme en tant que service (PaaS). Mais avant d'aborder ces deux concepts, revenons un peu en arrière et voyons de plus près les trois types de plates-formes Cloud principaux : *le Cloud public*, *le Cloud privé* et *le Cloud hybride*. Ces termes ont rapidement fait le « buzz » dans le monde des technologies modernes. Nous allons les démystifier.

Cloud public

Dans le modèle de Cloud public, vous n'êtes pas propriétaire des équipements utilisés. Votre fournisseur de Cloud est responsable de l'achat et de la maintenance de tout le matériel. Le Cloud public inclut l'ensemble des services et des ressources informatiques que vous ne possédez pas vous-même mais que vous utilisez par l'intermédiaire de votre fournisseur de Cloud. Le fournisseur est chargé d'en assurer le bon fonctionnement et de respecter les accords de niveaux de services (SLA). *Vous payez pour ce que vous utilisez, pas pour ce que vous possédez.*

Cloud privé

Avec un Cloud privé, vous possédez tout le matériel ou, du moins, vous le contrôlez intégralement. Le matériel qui exécute vos services et héberge vos données est situé quelque part dans votre datacenter local. Évidemment, ce contrôle total a un prix : vous devez acheter tous les équipements et en assurer la maintenance. *Vous payez pour ce que vous possédez, pas pour ce que vous utilisez.*

Mais comment installer une plate-forme Cloud privée dans un datacenter traditionnel sur site ? Heureusement, Azure propose une solution unique, spécialement conçue pour fonctionner dans cet environnement : Azure Stack. En un mot, Azure Stack est votre propre instance privée d'Azure, prête à l'emploi, que vous pouvez exécuter dans votre datacenter local.

L'exécution de services dans un Cloud privé diffère de l'exécution de services dans votre datacenter sur site. [Azure Stack](#) offre tous les avantages d'Azure, y compris des services intelligents que vous pouvez utiliser dans vos applications, le tout à l'intérieur du périmètre de votre datacenter local. Si pour des raisons spécifiques de sécurité ou de gouvernance, vous devez conserver certaines opérations « à proximité » dans votre datacenter sur site, vous continuez malgré tout à profiter de tous les avantages d'Azure et du Cloud.

Cloud hybride

Le modèle de Cloud hybride est, comme son nom l'indique, un mélange entre le Cloud public et le Cloud privé. Vous pouvez recourir à des services de Cloud public utilisant des ressources dans votre Cloud privé, et vice-versa. Avec Azure, vous pouvez exécuter, dans une application Web Azure, une application qui se connecte à une base de données sur site via les connexions hybrides d'Azure. Ceci ouvre tout un champ de possibilités. Vous décidez où héberger vos applications et vos données, tout en continuant à tirer parti des services Cloud intelligents.

Azure fournit des solutions pour tous les types de déploiements de Cloud computing. Les entreprises disposent ainsi d'une plus grande liberté pour utiliser Azure où et comme elles le souhaitent. De plus, elles peuvent déterminer le niveau de contrôle qu'elles souhaitent garder sur leurs applications et leurs données.

Démystifier les types de Cloud computing

Dans le Cloud computing, nous pouvons classer les services et les ressources en différents types. Il existe de nombreuses définitions différentes pour les types de Cloud computing. Ces types se différencient par le niveau de contrôle et de responsabilités que vous conservez par rapport au temps et aux efforts consentis pour apporter une valeur ajoutée à votre entreprise ou à vos clients. Les figures 1-1 et 1-2 illustrent la manière dont nous définissons les types de Cloud computing pour Azure.

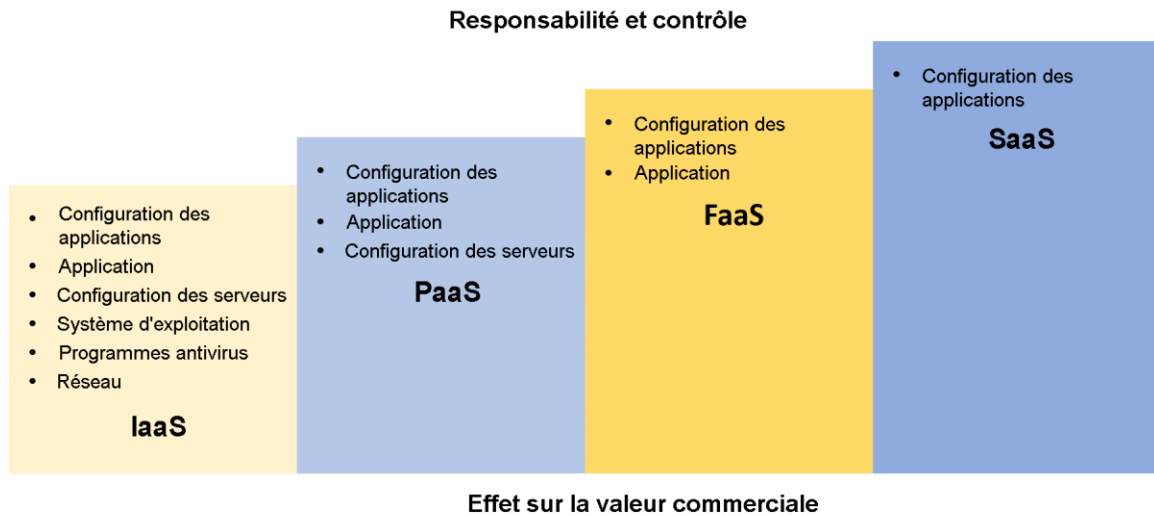


Figure 1-1 : explication des types de Cloud computing

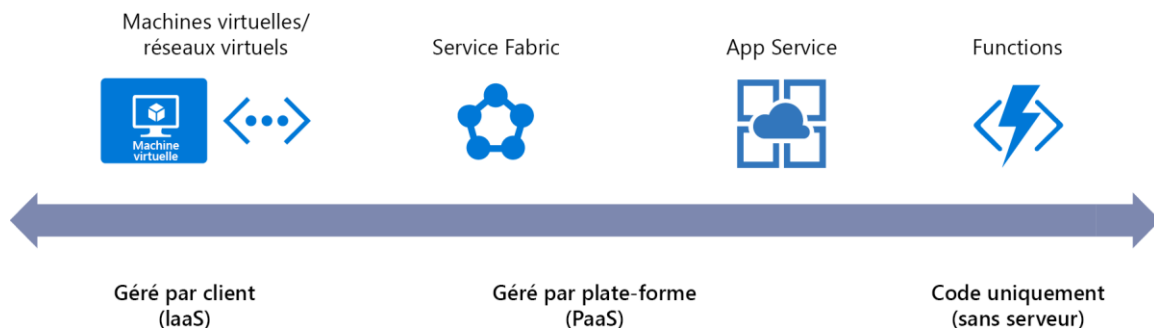


Figure 1-2 : exemples d'offres de Cloud computing

Infrastructure en tant que service

Avec l'infrastructure en tant que service (IaaS), vous êtes responsable de tout, à l'exception du matériel. Azure prend soin de tout le matériel, notamment des disques SSD, du câblage réseau, des routeurs, des alimentations, des générateurs de secours, des systèmes de refroidissement, etc. Les services qui entrent dans cette catégorie incluent, entre autres, [les machines virtuelles Azure](#), [les réseaux virtuels Azure](#) et [les conteneurs Azure](#).

Vous prenez soin de tout le reste. Votre application, le déploiement de correctifs pour le système d'exploitation (OS), la configuration du réseau logique et même l'actualisation des antivirus sur les ordinateurs. En d'autres mots, vous conservez un contrôle total sur ces éléments. Mais cela signifie aussi que vous consacrez beaucoup de temps et d'efforts à vous assurer que vos services et ressources restent opérationnels. Vous passez donc moins de temps à apporter une valeur ajoutée à vos produits métier.

Plate-forme en tant que service

Lorsque vous utilisez le modèle de plate-forme en tant que service (PaaS), vous êtes responsable de votre application et de sa configuration. Azure fournit le système d'exploitation et le matériel. Comme le montre la Figure 1-1, la PaaS présente un niveau d'abstraction supérieur à celui de l'IaaS. Vous avez beaucoup moins de responsabilités mais aussi moins de contrôle direct sur vos ressources matérielles qu'avec l'IaaS. Les services qui entrent dans la catégorie de la PaaS sont [Azure App Service](#), [Azure SQL Database](#), [Azure Redis Cache](#) et [Azure Batch](#).

Avec la PaaS, vous pouvez héberger votre application, mais ce niveau offre également des fonctionnalités supplémentaires prêtes à l'emploi, telles que les notifications push pour les appareils mobiles via Azure Mobile App ou le paramétrage automatique de votre base de données dans Azure SQL Database.

Vous déployez votre application et la configurez. Vous procédez également à la configuration de base des serveurs, comme l'adaptation du niveau de tarification ou du nombre d'instances où sera exécutée votre application. Votre application s'exécute toujours sur un serveur quelque part, mais vous n'êtes pas responsable de ce serveur (ni n'avez le contrôle sur celui-ci). Vous pouvez donc consacrer davantage votre temps à apporter une valeur ajoutée à vos applications et moins à assurer le fonctionnement d'un serveur.



Essayer Créez une application Node.js avec une MongoDB sur Azure dans cette présentation à l'adresse <https://docs.microsoft.com/azure/app-service/app-service-web-tutorial-nodejs-mongodb-app>.

Fonction en tant que service

Dans le modèle de fonction en tant que service (FaaS), vous êtes uniquement responsable de votre application et de votre logique métier. Vous ne vous préoccupez pas du dimensionnement, du système d'exploitation ni du matériel quel qu'il soit. La FaaS est parfois appelée *l'informatique sans serveur*. La catégorie FaaS inclut les services [Azure Functions](#), [Azure Logic Apps](#) et [Azure Event Grid](#).

Dans la FaaS, tout ce que vous devez faire est de créer votre application ou de coordonner votre logique métier, puis de l'exécuter. La mise à l'échelle est automatique. L'un des avantages les plus notables de la FaaS réside dans le fait que vous payez uniquement pour les ressources que vous utilisez lorsque la logique est exécutée, plutôt que de payer pour un service activé en permanence et qui attend que quelqu'un l'utilise.

Mais la FaaS ne se contente pas d'héberger votre application. Elle relie votre logique à des déclencheurs et à des sources. Des sources extérieures, telles que les files d'attente d'Azure Storage ou WebHooks, peuvent initier Azure Functions ou Logic Apps et leur fournir des données à traiter. Vous n'avez pas besoin de créer leur logique d'activation ni de configurer la plomberie pour les relier à des services ou banques de données externes. Vous pouvez donc vous concentrer sur votre logique et vos fonctions orientées clients.

Logiciel en tant que service

Avec le logiciel en tant que service (SaaS), votre seule préoccupation est la configuration du logiciel, rien d'autre. En effet, le SaaS offre le niveau d'abstraction le plus élevé parmi les différents types de Cloud computing. La catégorie SaaS inclut les services tels que [Azure Cognitive Services](#) et [Azure IoT Suite](#). [Office 365](#) est un autre exemple de suite d'applications SaaS Azure, offrant des applications professionnelles fiables comme Microsoft Word, Excel, PowerPoint et plus encore. Et tout cela est à votre disposition sans aucune installation ni maintenance.

Avec le SaaS, le logiciel est prêt à l'emploi. Tout ce que vous devez faire est de le configurer selon vos préférences et vous voilà opérationnel. Vous ne créez pas le logiciel et vous n'avez pas besoin de penser au déploiement, au dimensionnement, aux systèmes d'exploitation ni au matériel.

Pourquoi Azure ?

Vous pensez que le Cloud est peut-être la solution dont votre entreprise a besoin pour créer des logiciels d'exception et pour offrir à vos clients la valeur ajoutée et l'expérience utilisateur qu'ils attendent. Certes, la plate-forme Cloud Azure peut vous aider en ce sens. Avec ses vastes ressources et sa puissance informatique considérable, ainsi qu'un large éventail de services intelligents, vous pouvez créer des applications hautement évolutives et dynamiques, et vous paierez uniquement pour ce que vous utilisez. Toutefois, les fournisseurs de Cloud sont nombreux, chacun ses propres offres en matière de capacités et de produits. Alors, encore une fois, pourquoi Azure ?

Avec Azure, vous bénéficiez des connaissances approfondies et de l'expertise de Microsoft, une entreprise qui, depuis plus de 40 ans, est à l'avant-garde des solutions informatiques pour les particuliers comme pour les entreprises. Depuis des décennies, Microsoft conçoit, développe et exécute des réseaux et opérations dans des datacenters à très grande échelle pour ses propres besoins. Cette expérience et ce savoir-faire forment l'essence même d'Azure. C'est la base même de cette solution.

Qui utilise Azure ?

[Adobe](#), [Jet.com](#), [Geico](#), [GeekWire](#) et [DocuSign](#) ne sont que quelques exemples de clients qui, aux quatre coins du monde, se tournent vers la plate-forme Azure pour réaliser leurs objectifs professionnels et offrir davantage de valeur ajoutée à leurs clients. Certains de ces objectifs incluent, entre autres, la demande d'App Services, l'intelligence artificielle sans serveur, Azure IoT et même la diffusion multimédia en streaming. Examinons comment différents clients ont, de façon intéressante, utilisé la plate-forme Azure comme réponse à leurs besoins professionnels.

Le géant industriel sud-coréen [Samsung](#) fabrique toutes sortes de produits, des téléphones aux téléviseurs, en passant par les machines à laver, les principaux appareils domestiques et les climatiseurs. L'entreprise a décidé d'utiliser Azure pour augmenter l'efficacité et la robustesse de ses climatiseurs en effectuant leur surveillance et leur maintenance à distance. Elle a ainsi recours à des Azure IoT Hubs pour collecter des données de façon massive. Celles-ci sont transmises par des capteurs intégrés à ses climatiseurs et surveillent l'état des appareils, ainsi que l'environnement dans lequel ils sont utilisés. Capturées dans Azure SQL, les données servent à réaliser des analyses en temps réel. Les ingénieurs peuvent ensuite utiliser les informations extraites pour ajuster le comportement des climatiseurs, ce qui permet aux clients de réaliser des économies d'énergie de 15 à 18 % en moyenne.

[Coca-Cola](#) Company, l'une des marques les plus connues au monde, a lancé son premier soda éponyme en 1886. Depuis, l'entreprise est devenue un leader à l'échelle mondiale. En 2012, l'entreprise a lancé Coca-Cola Journey, un projet très ambitieux et son site Web associé. Le but est d'aider Coca-Cola à mieux communiquer avec une nouvelle génération de clients qui, de plus en plus, prennent leurs repères en matière d'achats auprès de canaux marketing non traditionnels. Avec l'aide des réseaux sociaux et d'un service de Microsoft appelé How-Old.net, l'entreprise souhaitait créer une expérience interactive, ludique et attrayante pour ses utilisateurs, tout en recueillant une meilleure visibilité sur le profil de ses clients. Basé sur la technologie des services cognitifs, le site How-Old.net analyse les données extraites de l'imagerie faciale afin de déterminer l'âge et le sexe des individus. Toutefois, Coca-Cola a demandé à l'équipe de Microsoft de dépasser la simple reconnaissance faciale pour ce projet, en utilisant des modèles d'apprentissage automatique afin de détecter différents objets présents dans les photos. Pendant un mois, le site a invité les utilisateurs à envoyer une photo d'eux avec une bouteille de Coca-Cola et un logo visible pour voir ce qui se passait. Si l'application Web détectait une bouteille, l'âge de la bouteille apparaissait et la page devenait rouge. Ensuite, les fans de Coca-Cola étaient invités à partager leur expérience sur les réseaux sociaux, puis d'explorer le site Coca-Cola Journey pour en savoir plus sur cette bouteille à la silhouette si reconnaissable.

[CarMax](#), le plus grand revendeur de véhicules d'occasion aux États-Unis, a remodelé son site Web avec l'aide des services de la plate-forme Azure. Son but était d'offrir une réponse plus rapide aux 16 millions de personnes qui visitent son site chaque mois. Plutôt que de « consolider et déplacer » son site vers un environnement IaaS Azure, CarMax a choisi de le concevoir à nouveau à partir de l'offre de PaaS d'Azure. Ceci a permis à CarMax de moderniser son site avec de nouvelles fonctionnalités de Cloud, de le rendre 100 % compatible avec les appareils mobiles et, surtout, d'évoluer vers un paradigme de développement DevOps dynamique. CarMax avance rapidement vers une architecture de micro-services, ce qui lui permettra d'adapter et d'actualiser certaines parties des applications indépendamment des autres parties. Même si cela prendra plusieurs années, le choix d'Azure comme base de développement va accélérer la migration du concessionnaire vers les micro-services.

[NBC Sports](#) collabore avec Microsoft la diffusion de méga-concerts. En 2016, les deux entreprises ont travaillé ensemble pour diffuser les Jeux Olympiques d'été de Rio de Janeiro. NBC a utilisé Azure Media Services pour diffuser simultanément les contenus de plus de 50 événements olympiques, fournissant un profil de 1080p et couvrant le chiffre énorme de 100 millions d'utilisateurs uniques, soit une consommation de 2,71 milliards de minutes de contenus sur un large éventail d'appareils et de formats. Media Services a fourni l'évolutivité et la robustesse permettant de distribuer ces contenus de façon impeccable aux téléspectateurs.

Azure permet aux entreprises, grandes et petites, ainsi qu'aux organismes gouvernementaux, aux établissements d'éducation et à n'importe quel utilisateur de services informatiques, de gérer leurs univers numériques de façon plus simple et plus économique, pour eux comme pour leurs clients. En un mot, voici comment :

- **Fournir des expériences logicielles multi-plates-formes (mobile, ordinateur, Web et hybride) :** Azure fournit des services avec lesquels vous pouvez mettre en œuvre ces scénarios « prêts à l'emploi », comme Azure App Service, avec Web App et Mobile App, ainsi que Mobile Center et Visual Studio Team Services, pour le déploiement et la gestion de tous types d'applications.
- **Offrir des services partout où ils sont requis :** Azure est unique par l'ampleur de ses options de déploiement : Cloud public, solutions hybrides et Azure sur site, dans votre propre datacenter (via Azure Stack). Toutes ces options peuvent être déployées à très grande échelle pour un prix abordable, tout en conservant vos données là où elles doivent être.

- **Proposer une plate-forme de données intelligente à très grande échelle** : Azure dispose de nombreux services que vous pouvez utiliser pour capturer, stocker, analyser et présenter vos données. Il s'agit de services tels que Azure SQL Databases, Data Lake Store et Data Lake Analytics, HDInsight, Event Hubs, Cognitive Services, et bien d'autres encore. Tous ces services permettent de créer une plate-forme de données abordable, intelligente et hautement évolutive. Celle-ci fournit aux entreprises les outils nécessaires pour extraire des informations analytiques précieuses parmi les volumes considérables de données générées aujourd'hui.
- **Délivrer rapidement des logiciels de haute qualité** : vous devez tester en permanence et rapidement les failles éventuelles afin de délivrer des logiciels de qualité. Avec Azure, vous pouvez le faire par l'intermédiaire de mécanismes de CI et CD intégrés directement dans les services comme Web App ou par l'intermédiaire de Visual Studio Team Services. Les fonctionnalités telles que les emplacements de déploiement pour App Services vous permettent de délivrer vos produits rapidement et sans interruption. Ensuite, pour vous assurer que votre appli fonctionne comme prévu en production, Azure fournit des services de pilotage, tels qu'Application Insights, qui vous permettent de déterminer le fonctionnement exact de votre appli ainsi que les améliorations possibles.

Azure a une portée mondiale

Avec des datacenters dans plus de 42 pays et régions dans le monde entier, la couverture exceptionnelle d'Azure offre de nombreux avantages. Tout d'abord, elle permet de limiter les temps d'arrêt que vos applications subiraient au cas où un datacenter tout entier venait à tomber en panne (une probabilité extrêmement faible en soi). En effet, vos applications peuvent basculer vers un autre datacenter et continuer de fonctionner, comme si rien ne s'était passé ou presque. Ceci permet également de réduire vos coûts de récupération d'urgence.

Si vous utilisez des réseaux de distribution de contenus (CDN) afin de réduire la latence, vous disposerez certainement, près du site de vos clients, d'un datacenter Azure où vous pourrez délivrer leurs contenus. Autre avantage qu'Azure peut offrir avant autant d'infrastructures réparties un peu partout : vous bénéficiez d'un choix plus étendu en matière de stockage des données. Ceci permet à votre entreprise de maintenir sa conformité avec les réglementations et législations locales ou régionales concernant la souveraineté des données.

Microsoft continue d'investir massivement dans l'infrastructure du datacenter et dans l'innovation [par l'intermédiaire du développement de matériel Open Source](#) et de proposer de nouvelles solutions pour datacenter, telles que les *datacenters sous-marins* du [projet Natick](#).

Azure est une solution extrêmement résiliente

Azure est une plate-forme intelligente, avec surveillance automatique et auto-régénération, qui assure le bon fonctionnement et la disponibilité de vos applications. Pour étayer cela, tous les services Azure sont régis par des [contrats SLA complets](#) qui définissent leurs performances dans des termes précis. Ces SLA garantissent une disponibilité allant de 99,9 % (trois « neuf ») à 99,99 % (quatre « neuf »). Cela signifie que les services Azure sont extrêmement résilients et fiables. Si vos services n'offrent plus les performances garanties par les spécifications de votre SLA, vous pouvez bénéficier d'une remise (jusqu'à 100 %) sur les coûts de service sur votre facture Azure.

Pour vous garantir en permanence des performances optimales, Microsoft surveille l'intégrité de chaque service Azure, dans chaque datacenter, dans le monde entier. Vous pouvez également contrôler ces indicateurs de mesure [sur la page publique relative à l'état des services Azure](#).

Comme nous l'avons mentionné précédemment, Microsoft gère des datacenters et fournit des services informatiques à grande échelle depuis 1989. Depuis, l'entreprise a ainsi accumulé énormément de connaissances. Aujourd'hui, Microsoft emploie certains des cerveaux les plus brillants du secteur informatique. Ceux-ci travaillent sans relâche pour assurer le bon fonctionnement et la fiabilité d'Azure afin que vous puissiez héberger vos services et stocker vos données. C'est pour appuyer cet engagement que tous les services Azure sont encadrés par des contrats SLA.

Azure est conforme à pratiquement tous les secteurs d'activités

[Azure offre l'éventail le plus complet d'offres de conformité de tous les fournisseurs de Cloud](#). La liste des conformités est impressionnante et ne cesse de s'enrichir. Les normes auxquelles Azure est conforme incluent, entre autres, [ISO 27001](#), [HIPAA](#) et [SOC 3](#). Parmi les conformités régionales, citons notamment le [Bouclier de protection des données UE-États-Unis](#) et le [DJCP \(Chine\)](#).

Par exemple, en se conformant à la certification ISO 27001, Microsoft garantit la mise en œuvre, la surveillance, la maintenance et l'amélioration permanente de ses normes de sécurité pour son réseau mondial de datacenters, ainsi que pour les services individuels dans Azure.



Plus d'informations Pour obtenir une liste complète des déclarations de conformité, visitez la page <https://www.microsoft.com/trustcenter/compliance/complianceofferings>.

Azure propose même une offre spéciale pour les administrations américaines. Baptisée [Azure Government](#), cette « édition spéciale » du Cloud Azure répond aux besoins spécifiques des entités gouvernementales. L'offre Azure Government est gérée séparément du Cloud public Azure, dans sa propre *instance*. Elle est donc exécutée sur des serveurs distincts et dédiés, dans une infrastructure isolée et complètement détachée.

Azure se concentre sur les fonctionnalités pour les développeurs

Azure vous aide, en tant que développeur, à devenir plus productif en fournissant la majeure partie de la « plomberie », vous évitant ainsi de devoir l'installer vous-même. Grâce à des fonctionnalités telles que les outils de mise à l'échelle automatique et d'authentification / d'autorisation, vous pouvez enrichir vos applications de fonctions et fonctionnalités avancées pratiquement sans effort. Vous pouvez intégrer des services comme Azure Search ou Cognitive Services, lesquels ne manqueront pas d'optimiser vos applications et de ravir vos utilisateurs.

Azure est une solution ouverte qui prend en charge les infrastructures de votre choix

Créez vos applications en JavaScript et déployez-les dans Web App. Ou créez-les en Ruby, Django, Java, PHP ou .NET, choisissez votre environnement de prédilection. Azure prend en charge un grand nombre de langages. Vous pouvez apporter votre propre application déjà existante ou en créer une toute nouvelle, toutes fonctionnent dans Azure.

En outre, vous pouvez choisir l'écosystème que vous préférez. Azure prend en charge la plupart des systèmes d'exploitation, comme Linux et Windows, et vous pouvez rédiger tous les scripts souhaités, en utilisant Bash le cas échéant. Vous pouvez également exécuter votre application comme vous le souhaitez, avec des conteneurs, Azure Service Fabric, App Services, Azure Stack, etc.

Une fois que vous avez appris comment utiliser Azure avec un ensemble d'outils, vous pouvez l'utiliser avec un autre ensemble d'outils. Les services et le portail Azure fonctionnent de la même façon pour tout.

Vous pouvez surveiller vos services Azure pendant vos déplacements

Vous êtes le genre de personne qui aime, en permanence, garder un œil sur les opérations ? Avec Azure, vous pouvez surveiller les performances de vos services et corriger les problèmes éventuels depuis l'[application mobile Azure](#). Cette application utile a été conçue pour vous donner une visibilité sur vos ressources, même lorsque vous n'êtes pas derrière votre bureau et devant votre ordinateur principal. Vous pouvez gérer des tâches comme le redémarrage d'une application Web ou l'arrêt d'une machine virtuelle. Elle inclut également une expérience complète d'interface de ligne de commande (CLI) via Cloud Shell. Vous pouvez ainsi globalement réaliser les mêmes tâches qu'avez le portail Azure.

Par ailleurs, certains services d'Azure effectuent des détections intelligentes et des recommandations pour vos services. Ils peuvent également vous envoyer des alertes concernant ceux-ci. Par exemple, ces services peuvent détecter la lenteur anormale d'une application Web ou vous informer de la mise à l'échelle automatique d'un App Service. Azur Security Center et Azure Monitor sont deux exemples de services capables de vous alerter de ce type de situations. Ainsi, vous ne ratez rien d'important.

Nous sommes là pour vous aider

Vous disposez de différents outils pour obtenir de l'aide avec Azure si vous en avez besoin.

Vous pouvez acheter [un plan de support](#), lequel vous donne accès aux équipes de support technique d'Azure et inclut d'autres services, tels que des conseils basés sur les meilleures pratiques pour la conception dans le Cloud ou une assistance pour la planification de votre migration. Selon vos besoins, vous pouvez acheter un plan de support qui garantit une réponse dans les 15 minutes de la part des équipes de support technique.

En outre, vous pouvez obtenir de l'aide gratuitement, 24 heures sur 24 et 7 jours sur 7, à travers de nombreux canaux différents. Vous pouvez envoyer un tweet à [@AzureSupport](#) ou contacter les équipes de support via les [forums MSDN](#), [StackOverflow](#), [Reddit](#) ou la [Communauté technique Microsoft](#).

Mise en route avec Azure

Après réflexion approfondie, vous avez rejoint la communauté Azure. Maintenant, vous souhaitez commencer à créer des applications. De quoi avez-vous besoin ? Pas grand-chose en fait. Le plus important est de disposer d'une connexion à Azure pour déployer votre application. Vous pouvez utiliser les outils, les applications et les infrastructures de votre choix.

Vos IDE et éditeurs sont les bienvenus

Quel que soit l'outil que vous utilisez pour créer ou modifier vos applications dans votre environnement sur site, vous pourrez vous en servir avec Azure. Le choix est vôtre : du Bloc-notes à [Visual Studio Code](#), en passant par [Sublime Text](#), [Visual Studio](#) ou les éditeurs de code sur le portail Azure. C'est vous qui décidez !

Des plug-ins sont disponibles pour tous les principaux IDE, ce qui vous permet de réaliser des tâches comme publier directement dans Azure. Et même si vous préférez utiliser un outil rudimentaire tel que le Bloc-notes, vous pourrez quand même automatiser vos déploiements vers Azure à l'aide de pratiques d'intégration continue (CI)/de distribution continue (CD).

Utilisez l'interface de ligne de commande d'Azure pour la création de scripts

Azure fournit une puissante [interface de ligne de commande \(CLI\)](#) avec laquelle vous pouvez pratiquement tout faire dans Azure. Vous pouvez utiliser l'interface de ligne de commande d'Azure pour démarrer et arrêter les machines virtuelles, déployer votre application vers une appli dans Azure Web App, pour créer de nouvelles ressources, et tout le reste. Elle est également très utile pour l'automatisation des tâches et leur exécution dans vos flux de CI/CD.

Vous pouvez utiliser l'interface de ligne de commande d'Azure disponible sur le portail Azure dans [Azure Cloud Shell](#) ou localement, sur votre machine, par l'intermédiaire des outils Azure CLI. Un autre avantage est que vous pouvez appliquer, dans l'interface de ligne de commande Azure, votre expertise actuelle en Bash ou en Windows PowerShell.

Exécutez ce que vous voulez dans Azure

Azure est idéal pour les applications Web et les API. La solution est également excellente pour héberger les applications mobiles et de bureau. Vous pourriez, par exemple, l'utiliser pour authentifier les utilisateurs dans votre application de bureau ou pour envoyer des notifications push vers votre application mobile. Quel que soit le scénario auquel vous pensez, Azure peut y apporter une valeur ajoutée.

Qu'en est-il des coûts ?

De toute évidence, il est important de connaître les coûts liés à Azure. Pour vous aider en ce sens, Microsoft propose la [Calculatrice de prix Azure](#). Avec cet outil très pratique, vous sélectionnez les services que vous allez utiliser et précisez dans quelle mesure vous prévoyez de les utiliser. La calculatrice vous indique alors le montant de votre facture mensuelle et de votre facture annuelle. Ceci, en plus du volume de données que vous allez utiliser, détermine vos coûts.



Remarque La Calculatrice de prix Azure estime les coûts des abonnements avec paiement à l'utilisation. Il existe d'autres options de paiement pour les entreprises, les partenaires et les abonnés MSDN, qui peuvent affecter le coût des services.

Vous pouvez également maîtriser vos coûts à l'aide de l'[API de facturation Azure](#) et des [alertes de prix](#). Vous pouvez utiliser ces solutions pour mieux comprendre les tendances en termes de prix et de dépenses afin de maîtriser vos coûts.

Sélection des services Azure appropriés

Azure propose un grand nombre de services pour vous aider à créer et exécuter vos applications. Il existe de nombreux services et il s'avère parfois compliqué de choisir les services adaptés à vos besoins. Quels sont les services que vous devez choisir en tant que développeur ? Découvrons les options disponibles.

Où héberger votre application

Dès le début, vous devrez faire un choix concernant l'hébergement de votre application. Azure propose plusieurs options d'hébergement.

Machines virtuelles

L'une des solutions proposées pour héberger votre application est une machine virtuelle dans [Azure Virtual Machine](#). Elle vous permet de bénéficier d'un large contrôle sur la méthode d'hébergement de votre application, mais vous êtes responsable de la maintenance de l'environnement, des correctifs du système d'exploitation et de la mise à jour des programmes antivirus.

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

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 ordinateurs virtuels, [Azure Batch](#) est un bon choix. 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'utilisez 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 des échelles telles que des modèles de risque financier, de transcoding 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 plate-forme de calcul.



Essayer Devenez opérationnel sur Azure Batch en cinq minutes grâce à [ces tutoriels pas à pas](#).

Conteneurs

Les conteneurs sont beaucoup plus légers que les machines virtuelles et vous pouvez les démarrer et les arrêter en quelques secondes. Les conteneurs offrent également une large portabilité ; ils sont donc parfaitement adaptés au développement local d'une application sur votre machine, puis à l'hébergement dans le Cloud à des fins de test et, enfin, à la 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.

À l'instar des machines virtuelles, les conteneurs vous fournissent un large contrôle de votre environnement. Vous pouvez installer tout ce dont vous avez besoin pour exécuter vos applications. Vous êtes encore une fois responsable des correctifs et de la maintenance du système d'exploitation qui s'exécute dans le conteneur, ainsi que des produits connexes comme les programmes antivirus.

Héberger des conteneurs avec les instances de conteneur Azure

Vous pouvez héberger votre conteneur à l'aide des [instances de conteneur Azure](#). Grâce à ce service, vous pouvez lancer rapidement un conteneur sans utiliser d'orchestrateur, comme Kubernetes, et sans avoir à gérer les ressources qui hébergent le conteneur.

Le service d'instances de conteneur est facturé par seconde, processeur virtuel, gigaoctet ou mémoire.



Plus d'informations Pour en savoir plus sur les instances de conteneur, consultez la section [Découvrir les instances de conteneur Azure](#).

Héberger des conteneurs avec Azure Container Service

Il est également possible d'héberger des conteneurs à l'aide d'[Azure Container Service](#). Grâce à ce service, vous pouvez adapter et gérer vos conteneurs à l'aide d'orchestrateurs tels que Mesosphere, DC/OS, Docker Swarm ou Kubernetes. Ce service est un excellent moyen pour commencer à transférer vos conteneurs vers le Cloud.



Plus d'informations Pour en savoir plus sur Container Service, consultez la section [Découvrir Azure Container Service](#).

Azure Functions

Avec Azure Functions, vous pouvez écrire uniquement 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 un calendrier. Des liaisons d'entrée et de sortie relient le code de votre fonction à d'autres services, comme Azure Blob Storage, Azure Cosmos DB et Azure Service Bus, avec un code minimal. Avec Functions, vous pouvez créer rapidement de petites parties de fonctionnalité et les héberger dans un environnement flexible qui gère automatiquement la mise à l'échelle.

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

Vous pouvez écrire des fonctions Azure Functions en C#, F#, Node.js, Java, PHP et un nombre croissant de langages.

L'application permettant d'activer une fonction chaque fois qu'un nouveau fichier image est chargé vers Azure Blob Storage est un exemple d'application utilisant Functions. La fonction redimensionne ensuite l'image et l'écrit dans un autre compte Blob Storage. La signature de fonction pour cet exemple se présente comme suit (dans un script C#) :

```
public static void Run(Stream myBlob, string name, TraceWriter log, Blob outputBlob) { }
```

Les données de Blob déclenchant la fonction sont passées dans la fonction en tant que paramètre myBlob, qui comprend l'URL Blob. Vous pouvez utiliser 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 à Blob Storage, il vous suffit de la configurer.



Essayer [Créez votre première fonction Azure Function 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 dans Azure Storage Queue. Vous pouvez lier des appels d'API aux connecteurs pour créer un workflow (éventuellement complexe) pouvant impliquer des ressources dans le Cloud et sur site. Logic Apps compte de [nombreux connecteurs aux API disponibles](#), par exemple pour la connexion à Azure SQL Databases, 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 Logic App.

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

Voici un exemple de workflow dans Logic Apps :

1. La Logic App est activée par un e-mail contenant un ordre d'expédition envoyé dans Office 365.
2. En utilisant les données de cet e-mail, la Logic App vérifie la disponibilité de l'article commandé dans SQL Server.
3. La Logic App envoie un message texte vers le téléphone du client en utilisant Twilio (le numéro de téléphone se trouve également dans l'e-mail), indiquant que la commande a été reçue et que l'article a été expédié.



Essayer [Découvrir Azure Logic Apps.](#)

Azure App Service

Vous pouvez également héberger vos applications dans l'un des principaux services disponibles dans Azure : [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 App Services peuvent sécuriser une application à l'aide d'[Azure Active Directory](#) et utiliser des domaines personnalisés.

Azure App Service comprend les éléments suivants :

- **Web App** : [Web App](#) est l'un des services Azure les plus utilisés. Vous pouvez utiliser ce service pour héberger vos applications Web ou vos API. Le service Web App est en fait l'abstraction d'un serveur Web, comme Internet Information Services (IIS) ou Tomcat, que vous utilisez pour héberger des applications axées HTTP.

Web App peut héberger des applications écrites en .NET, Node.js, PHP, Java ou Python et il existe des extensions que vous pouvez utiliser pour exécuter encore plus de langages.



Essayer Nous vous détaillons un exemple d'application Node.js et MongoDB dans notre section d'exemples.

- **Web App pour les conteneurs** : [Web App pour les conteneurs](#) vous permet de déployer et d'exécuter facilement des applications Web en conteneur et à l'échelle. 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 plate-forme traite automatiquement les corrections du système d'exploitation, le dimensionnement de la capacité et l'équilibrage de charge.



Essayer Nous détaillons le déploiement d'une [application Web](#) avec un conteneur formaté Docker dans la section des exemples.

- **Mobile Apps** : [Mobile Apps](#) fournit un serveur principal pour vos applications mobiles. Vous hébergez une API dans Mobile Apps, à laquelle vos applications mobiles se connectent via le client multi-plateforme SDK. Cette fonction est disponible pour iOS, Android, Windows et Xamarin pour iOS et Android et Xamarin Forms. Mobile Apps fournit des fonctionnalités uniques, comme la synchronisation hors connexion et les notifications Push, qui vous aident à créer une expérience mobile moderne, performante et sûre.

Vous pouvez écrire votre serveur principal Mobile Apps en .NET ou Node.js.



Essayer Nous détaillons la création d'un serveur principal mobile dans notre section d'exemples.

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 apporte des fonctionnalités uniques, mais ils partagent tous certaines caractéristiques communes :

Mise à l'échelle

Azure App Service fonctionne sur les [plans App Service](#), qui sont des abstractions de machines virtuelles. Une ou plusieurs machines virtuelles exécutent vos services Azure App Service, mais vous n'avez pas besoin de savoir lesquelles car elles sont gérées par Azure. 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. Azure App Service peut également adapter automatiquement le nombre d'instances pour vous, en fonction d'un calendrier ou de mesures comme l'UC, la mémoire ou la capacité de la file d'attente HTTP.

Emplacements de déploiement

Il s'agit d'une fonctionnalité très utile d'Azure App Service. Vous pouvez déployer une nouvelle version de votre application à un emplacement de déploiement, à partir duquel vous pouvez ensuite vérifier si la version fonctionne correctement, avant de la déplacer dans votre emplacement de production. Vous pouvez également utiliser la fonction de *Test en production* [d'Azure](#) pour acheminer un pourcentage du trafic depuis votre application de production vers un emplacement de déploiement. Par exemple, vous pouvez déplacer 10 % de vos utilisateurs vers la nouvelle version de votre application dans l'emplacement de déploiement pour voir si les nouvelles fonctionnalités s'exécutent comme prévu et si les utilisateurs s'en servent.

Lorsque vous êtes satisfait de la façon dont la nouvelle version de votre application s'exécute dans l'emplacement de déploiement, vous pouvez effectuer un *basculement*, pour échanger l'application de l'emplacement de déploiement avec celle de votre emplacement de production. Vous pouvez également basculer l'application d'un emplacement de développement vers un emplacement intermédiaire, puis vers l'emplacement de production, comme l'illustre la Figure 2-1. Avant cela, l'opération de basculement vérifie que la nouvelle version de votre site Web est opérationnelle et prête à être lancée. Lorsque cette vérification est effectuée, l'opération de basculement commute les emplacements, et vos utilisateurs peuvent alors voir la nouvelle version de l'application, sans qu'il y ait de temps d'interruption. Si vous le souhaitez, vous pouvez également annuler le basculement et revenir au déploiement de la nouvelle version.

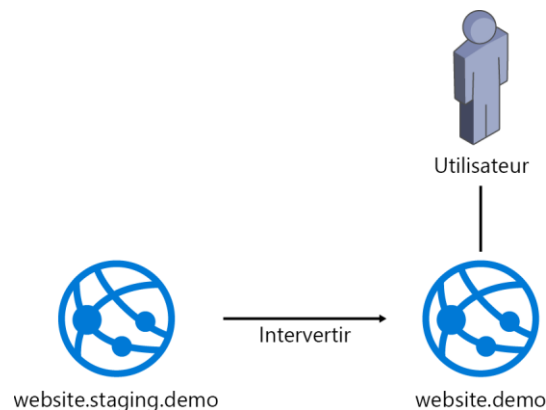


Figure 2-1 : basculement d'un emplacement de déploiement

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 d'App Service, et vous voulez les séparer 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 l'API de gestion d'Azure. Cela permet aux outils tels que Visual Studio Team Services d'effectuer des opérations de basculement lors du lancement d'une version.

Un emplacement de déploiement est un autre élément d'Azure App Service (comme Web App) qui s'exécute dans le même plan Azure App Service, parallèlement à votre Azure App Service d'origine. Les emplacements de déploiement s'exécutent dans le même plan Azure App Service que votre Azure App Service d'origine ; ils n'entraînent donc pas de frais d'utilisation supplémentaires.

Livraison en continu

Pour publier votre application sur App Services, vous pouvez utiliser des services externes tels que Visual Studio Team Services, Jenkins ou Octopus Deploy. Vous pouvez également utiliser la [fonction Livraison en continu](#) d'App Services. Cette fonction vous permet de créer un canal de version 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 tests 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 doivent pas être situées dans Azure ; elles peuvent être n'importe où, sur site par exemple, dans votre propre Datacenter. Vous pouvez vous connecter aux services sur site au moyen de nombreux mécanismes, selon vos besoins. Vous pouvez utiliser les [connexions hybrides Azure](#), les [réseaux virtuels Azure](#) et [Azure ExpressRoute](#) pour vous connecter aux ressources sur site.

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 nom de domaine personnalisé, vous pouvez le faire en mappant ce nom de domaine à 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). Vous pouvez fournir votre propre certificat ou en acheter un [directement depuis le portail Azure](#). Lorsque vous achetez un certificat SSL à partir du portail Azure, vous achetez un certificat Azure App Service, que vous pouvez configurer pour l'utiliser avec les liaisons de votre domaine personnalisé.



Essayer [Découvrir comment acheter et configurer un certificat dans cette procédure.](#)

Azure App Service Environment

Dans une application Web multi-niveaux, une base de données ou des services sont souvent utilisés par votre application dans Web App. Idéalement, vous voulez que ces services soient exposés uniquement sur l'application et non sur Internet. L'application elle-même est souvent publiée sur Internet, car représentant le point d'entrée pour vos utilisateurs.

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 App, de façon que ces services de support soient exposés uniquement sur l'application, et non sur 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 App et que celle-ci fasse partie de votre serveur principal. Dans ce cas, vous pouvez utiliser un [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. Notez cependant qu'Azure App Service Environment est disponible uniquement pour les niveaux de tarification premium d'App Services.



Remarque Azure App Service Environment ne fonctionne pas actuellement pour Web App pour les conteneurs.

Pourquoi et quand l'utiliser ?

Même si App Service, Functions et Logic Apps fonctionnent souvent conjointement, chaque option est conçue pour des besoins d'applications spécifiques.

Le tableau 2-1 présente brièvement leur objectif selon le type de besoins que vous pourriez avoir.

Tableau 2-1 : Quel App Service devez-vous utiliser en cas de

Catégorie	Objectif	Type d'Azure App Service
Hébergement d'applications	Héberger des applications Web et des API	Web App et/ou Web App pour conteneurs d'App Service
	Hébergement de serveur principal pour applications mobiles	Mobile App d'App Service
Orchestration de processus	Orchestrer une étape dans un processus	Fonctions
	Orchestrer l'ensemble du processus	Logic Apps

Accélérer 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 Content Delivery Network

[Azure Content Delivery Network](#) est l'un des services d'Azure pouvant vous aider à rendre votre application plus rapide. 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>. Cette opération est très facile à faire et améliore les performances de votre application de l'une des manières suivantes :

- Elle déporte le contenu utile de votre application. Ce contenu est maintenant transmis par Content Delivery Network, libérant ainsi les cycles de traitement de 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 les applications mobiles et de bureau.

Content Delivery Network peut, par exemple, servir à fournir des vidéos pour une application mobile. Les vidéos peuvent être volumineuses et vous ne voulez pas les stocker sur un appareil mobile (vos utilisateurs non plus !). À l'aide de Content Delivery Network, ces vidéos sont fournies à partir des points de présence, ce qui améliore également les performances car les points de présence sont proches de l'utilisateur.



Essayer [Découvrir Azure Content Delivery Network.](#)

Azure Redis Cache

Toutes les applications modernes fonctionnent avec 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 le cache [Azure Redis Cache](#). L'avantage du cache est qu'il stocke les données dans un format simple, sous forme de 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 ne sont pas disponibles, 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 en particulier, elle les retrouvera dans le cache, économisant un déplacement 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 et les ingénieurs Microsoft. Il est très performant et comprend des options avancées, comme le clustering et la géo-réplication.



Essayer [Découvrir Azure Redis Cache.](#)

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 la latence. La latence est le temps qu'il faut pour qu'un signal ou une demande parvienne 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 que vous placez en face de vos applications Web App, par exemple, partout dans le monde. Web App sert de points de terminaison, qu'Azure Traffic Manager surveille pour assurer l'intégrité et la performance de votre application. Comme le montre la figure 2-2, lorsqu'un utilisateur accède à votre application, Traffic Manager l'achemine vers l'application Web App à proximité la plus performante.

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

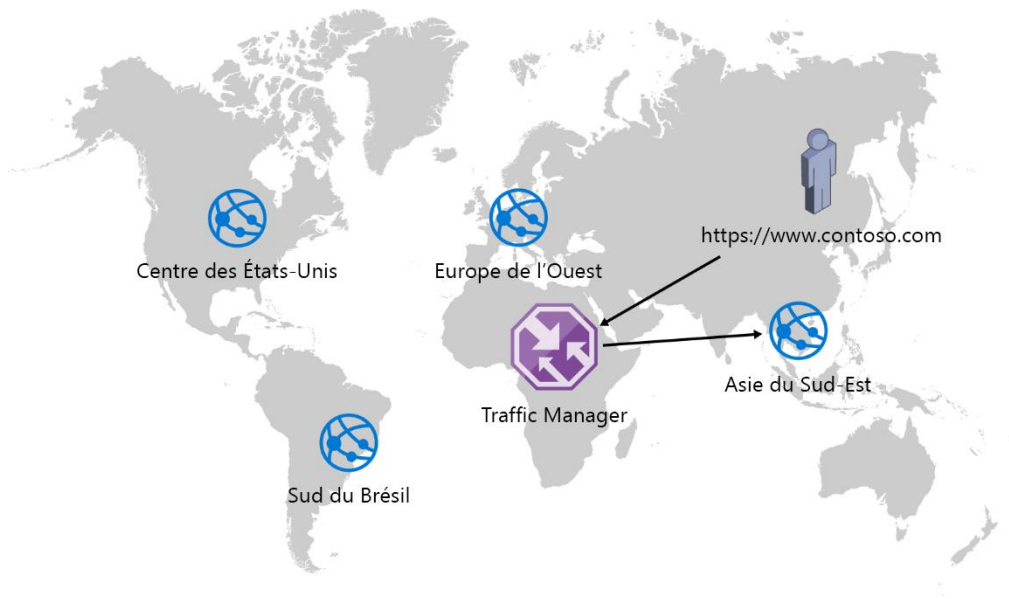


Figure 2-2 : Azure Traffic Manager redirige le trafic vers le point de terminaison géographique le plus performant

Où stocker vos données

Les données, de toutes formes et tailles, représentent un aspect très important de toutes les applications modernes. 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 2-2 présente les options de stockage disponibles dans Azure.

Tableau 2-2 : options de stockage dans Azure

	Bases de données SQL	MySQL	PostgreSQL	Cosmos DB	Blob	Table	File d'attente	Fichier	Disque	Data Lake Store	SQL Data Warehouse
Données relationnelles	X	X									
Données relationnelles-objet			X								
Données non structurées				X	X						
Données semi-structurées						X					
Messages de file d'attente							X				
Fichiers sur disque								X			
Fichiers haute performance sur disque									X		
Stockage de données volumineuses					X			X	X	X	X
Stockage de données de petite taille	X	X	X	X		X	X	X	X		
Réplication de données géographiques	X			X							



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.

Azure SQL Databases

Si vous souhaitez utiliser des tables avec des colonnes et des lignes pour stocker des données, la base de données [Azure SQL Database](#) est l'option idéale. Il s'agit d'un système de base de données relationnelles semblable au système Microsoft SQL Server sur site. La base de données SQL Database s'exécute dans le Cloud, elle est donc entièrement gérée, performante, évolutive, sauvegardée automatiquement et elle propose de nombreuses fonctions avancées.

Vous pouvez utiliser la base de données SQL Database avec vos outils préférés, comme SQL Server Management Studio et Entity Framework.

Voici certaines des fonctionnalités plus avancées :

- La géo-réplication, qui réplique les données dans d'autres régions géographiques en temps réel ([Découvrir la géo-réplication](#))
- Le masquage des données dynamiques, qui masque dynamiquement les données sensibles pour certains utilisateurs lors de l'exécution ([Découvrir le masquage des données dynamiques](#))
- L'audit, qui fournit une piste de vérification complète de toutes les actions liées aux données qui se produisent ([Découvrir l'audit d'Azure SQL Database](#))

Extrêmement fiables et robustes, les bases de données SQL Database sont couvertes par un contrat de niveau de service garantissant une disponibilité de 99,99 %.

Base de données Azure pour MySQL

MySQL est un système de base de données relationnelle open source utilisé par des millions d'applications, telles que WordPress, Joomla et Drupal. MySQL est désormais disponible sous forme de service de base de données managé via le service [Base de données Azure pour MySQL](#). Vous pouvez ainsi être rapidement opérationnel, sans avoir à installer ni à gérer de serveurs. Azure se charge de tout pour vous. Vous bénéficiez d'une sécurité et d'une évolutivité immédiates. Le service Base de données pour MySQL offre la même fiabilité que SQL Database, avec une disponibilité de 99,99 % grâce à un SLA comptant parmi les meilleurs du secteur.

Comme cela est le cas pour SQL Database et le service Base de données Azure pour PostgreSQL, des sauvegardes incrémentielles sont effectuées toutes les cinq minutes, et des sauvegardes complètes toutes les heures. Elles vous permettent de restaurer vos données à un état antérieur (jusqu'à 35 jours).



Essayer [Créez et connectez-vous à une base de données MySQL dans Azure.](#)

Base de données Azure pour PostgreSQL

Reposant sur PostgreSQL, la technologie de base de données la plus populaire du marché, [Base de données Azure pour PostgreSQL](#) offre un service de base de données objet-relationnel. Les bases de données objet-relationnel diffèrent légèrement des bases de données relationnelles dans le sens où elles peuvent stocker des types de données plus complexes, comme des types de données imbriqués, et sont donc plus étroitement alignées sur les langages de programmation orientés objet que nous utilisons dans nos applications.

Base de données pour PostgreSQL est un service managé qui possède les mêmes caractéristiques que le service Base de données pour MySQL. Il peut évoluer à la hausse ou à la baisse, offre un haut niveau de disponibilité (SLA de 99,99 %) et fait l'objet d'une sauvegarde automatique.



Essayer [Créez une base de données Azure pour PostgreSQL à l'aide de la CLI Azure.](#)

Azure Cosmos DB

[Azure Cosmos DB](#) est, entre autre, la nouvelle version et la nouvelle appellation de DocumentDB. Il s'agit d'un nouveau type de base de données réellement pensé pour le Cloud. Voici quelques-unes de ses principales caractéristiques :

- SLA garantissant 99,99 % de disponibilité, avec des temps de latence faibles (inférieurs à 10 ms pour les lectures et à 15 ms pour les écritures).
- Géomé-réplication : réplication en temps réel des données dans d'autres régions géographiques ([Comment distribuer des données mondialement avec Azure Cosmos DB](#)).

- Gestion du trafic : redirige les utilisateurs vers le réplica de données le plus proche.
- Évolutivité illimitée à l'échelle mondiale ; vous payez uniquement pour le débit et le stockage dont vous avez besoin.
- Indexation automatique des données. Plus besoin de gérer ni d'ajuster la base de données.

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



Essayer [Démarrez avec l'API DocumentDB d'Azure Cosmos DB.](#)

Azure Storage

Une autre option de stockage des données est disponible : [Azure Storage](#). C'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 :

- Géo-redondance : réplication des données dans différents Datacenters en vue de leur récupération en cas de panne du Datacenter suite à un sinistre
- Chiffrement des données lors de l'exécution
- Domaines personnalisés

Azure Storage offre notamment les types de stockage suivants : *Objet blob*, *File d'attente*, *Fichier* et *Disque* (Figure 2-3).

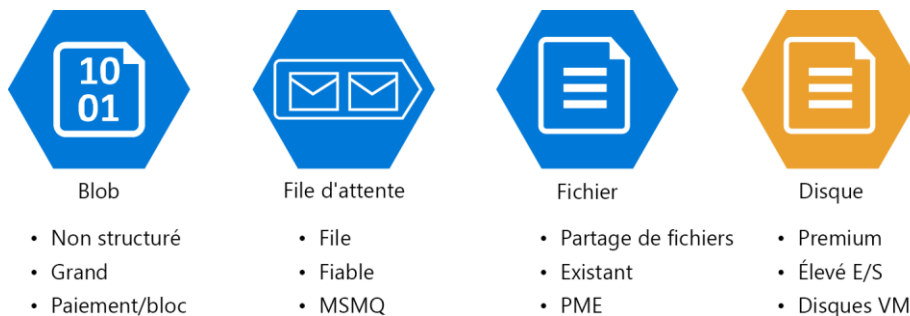


Figure 2-3 : Présentation des services Azure Storage

Stockage d'objets blob

Vous pouvez utiliser le [stockage d'objets blob Azure](#) pour stocker de grands volumes de données non structurées. Il peut s'agir de fichiers vidéo, image, audio ou texte, ou même de fichiers de disque dur virtuel pour VM.

Il existe deux types d'objets blob : [Page et Bloc](#). Les objets blob Page sont optimisés pour les opérations de lecture et d'écriture aléatoires. C'est la solution idéale pour stocker 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.



Essayer [Démarrez avec le stockage d'objets blob Azure.](#)

Stockage de files d'attente

Le [stockage de files d'attente Azure](#) est un type de stockage inhabituel. Il est utilisé pour stocker de petits messages de données, mais 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 de files d'attente Azure repose sur la fonction Microsoft Message Queueing présente dans des versions précédentes de Windows.



Essayer [Démarrez avec 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 types de stockage Azure, le stockage de fichiers est évolutif et économique.



Essayer [Démarrez avec 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 VM exigeant un haut niveau de performances. Par exemple, une VM qui exécute SQL Server. Notez toutefois que 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 VM. [Le stockage Azure Data Lake Store](#) est 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. Les principales caractéristiques d'Azure Data Lake sont les suivantes :

- 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.
- Vous pouvez stocker des données 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.

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



Essayer [Démarrez avec Azure Data Lake Store à l'aide du portail Azure.](#)

Azure SQL Data Warehouse

Utilisez [Azure SQL Data Warehouse](#) si vous recherchez une solution de stockage (data warehouse) traditionnelle entièrement managée, évolutive en taille, performante et sécurisée. Vous pouvez stocker les données dans des schémas prédéfinis et les interroger en utilisant le dialecte SQL de SQL Server auquel vous êtes habitué.

SQL Data Warehouse s'exécutant dans Azure, vous bénéficiez de fonctions avancées, comme la détection automatique des menaces qui s'appuie sur l'apprentissage automatique pour analyser les tendances de votre scénario d'usage et, tel un système d'alarme, vous informer des violations potentielles.

Par exemple, utilisez SQL Data Warehouse 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.



Essayer [Créez un entrepôt de données Azure SQL Data Warehouse.](#)

Ajout de fonctions intelligentes à votre application

Héberger votre application et vos données sur une plate-forme évolutive, sécurisée et hautement performante constitue un réel avantage. Mais que diriez-vous d'y ajouter des fonctions intelligentes ?

Azure Search

La recherche est une caractéristique commune à la plupart des applications. Pourtant, il s'agit généralement d'une fonction difficile à mettre en œuvre. Le service [Azure Search](#) fournit une grande partie des éléments nécessaires 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. C'est tout. 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 permet aux utilisateurs d'explorer les données selon la proximité d'un résultat de recherche avec un emplacement physique.
- Des analyseurs linguistiques Lucene, ainsi que des processeurs de langage naturel Microsoft, sont disponibles dans 56 langues pour gérer intelligemment les caractéristiques linguistiques propres à la langue, notamment les temps des verbes, le masculin et le féminin, les pluriels irréguliers (par exemple, « cheval » et « chevaux »), la décomposition des mots, leur césure (pour les langues sans espaces), etc.
- Grâce à la surveillance et à la création de rapports, vous pouvez connaître l'objet des recherches, et savoir si 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.

Vous pouvez déployer Azure Search pour aider les utilisateurs à effectuer des recherches dans votre catalogue de produits au sein d'une application de commerce électronique.



Essayer [Créez votre premier index Azure Search dans le portail.](#)

Azure Cognitive Services

Avec [Azure Cognitive Services](#), vous pouvez ajouter des fonctions intelligentes de haut niveau à vos applications. Cognitive Services se compose d'un ensemble d'API utilisables pour accomplir des opérations qui relèvent presque de la magie. Il existe plus de 25 API, réparties dans cinq catégories : Vision, Parole, Langage, Connaissance et Recherche. Voici un aperçu des API disponibles :

- L'[API Face](#) sert à la reconnaissance et à l'identification faciale. Elle va jusqu'à déterminer la couleur des yeux d'une personne, même si elle porte des lunettes.
- L'[API Emotion](#) analyse les émotions exprimées par un visage sur une photo.
- L'[API Language Understanding Intelligent Services](#) (ou LUIS) est capable de comprendre du texte, selon le modèle utilisé pour son apprentissage.
- L'[API Translator Speech](#) transforme la parole en texte et le traduit en temps réel.

Cognitive Services tire son intelligence de l'apprentissage automatique et de l'intelligence artificielle, que vous pouvez former à l'aide de vos propres jeux de données. Ces services sont tout simplement incroyables et ouvrent des possibilités encore inimaginables il y a quelques années.

Azure Bot Service

[Azure Bot Service](#) facilite la création d'un *bot*, c'est-à-dire un logiciel capable d'interagir avec les utilisateurs de manière automatisée et autonome.

La création d'un bot n'est en aucun cas une tâche simple. Vous devez effectuer le suivi du contexte de votre interaction avec l'utilisateur, et vous devez être prêt à réagir face à une multitude de paramètres d'interaction possibles. Dans ce cadre, Bot Service vous aide des façons suivantes :

- Ce service vous aide à effectuer le suivi du contexte d'interaction et vous fournit des modèles pour démarrer à partir de Bot Framework.
- Il dispose d'intégrations étroites avec Cognitive Services, ce qui vous permet de rendre votre bot « intelligent » plus facilement.
- Il vous aide à intégrer votre système à des services tels que Facebook, Slack, Microsoft Teams, Telegram, et ainsi de suite.
- Il offre tous les avantages d'un service géré dans Azure : une évolutivité impressionnante, la livraison continue intégrée, et vous payez uniquement pour ce que vous utilisez.

Un exemple de bot que vous pouvez créer grâce à Bot Service est un bot qui fournit à vos utilisateurs les réponses à leurs questions les plus fréquentes. L'interface du bot peut être une fenêtre de chat située sur votre site Internet.



Faites un essai [Prise en main de Chatbot en utilisant Azure Bot Service.](#)

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 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 qu'en matière de modèles de messagerie.

Azure Service Bus

Dans Azure, le cœur de la messagerie est le service [Azure Service Bus](#). Service Bus englobe un éventail de services que vous pouvez utiliser pour les modèles de messagerie. Les plus importants de ces services sont les files d'attente et les rubriques Azure Services Bus.



Faites un essai Mise en route des services Azure Service Bus [Files d'attente](#) et [Rubriques](#).

Files d'attente Azure Service Bus

Vous utilisez les [files d'attente Azure Service Bus](#) pour découpler les systèmes entre eux. Voici un exemple : une application Web reçoit des commandes d'utilisateurs et doit invoquer un service Web pour traiter les commandes en question. Le service Web va prendre longtemps, peut-être jusqu'à cinq minutes, pour traiter l'intégralité d'une commande. Il est évidemment inacceptable que l'application Web attende cinq minutes entre le placement de la commande par l'utilisateur et l'affichage des commentaires. Une façon efficace 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 dans une file d'attente Azure Service Bus. Une fois cette tâche effectuée, l'application peut informer 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 terminé le traitement d'une commande, il envoie une notification par e-mail à l'utilisateur ayant commandé l'article.

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 des applications.

La 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 dispose de certaines fonctionnalités astucieuses permettant de travailler avec des messages dans la file d'attente, comme la détection des doublons et une sous-file d'attente lettre morte 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 les différences entre la file d'attente et les rubriques :

- 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 rubriques Service Bus fonctionnent exactement comme une file d'attente, et plusieurs applications peuvent traiter le même message. Les applications peuvent créer un abonnement à la rubrique, ce qui indique quel type de messages les intéressent. Tout comme les files d'attente, les rubriques sont dotées de fonctionnalités 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 Rubriques Service Bus d'Azure Storage Queues

Il y a les files d'attente Service Bus, mais il y a également les files d'attente Azure Storage. Ces services effectuent essentiellement le même travail, mais de subtiles différences les distinguent. Le tableau 3-1 examine chacun d'eux.

Tableau 3-1 : comparaison des fonctionnalités entre les files d'attente Service Bus et les files d'attente Azure Storage

Files d'attente Azure Service Bus	Files d'attente Azure Storage
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

Dans notre monde de logiciels omniprésents, les données sont générées à partir de nombreux appareils, applications et emplacements, en permanence. Aujourd'hui, il devient de plus en plus courant pour les entreprises de capturer des quantités massives de données afin de les analyser ou de les transformer et de les déplacer pour une utilisation ultérieure. [Azure Event Hubs](#) peut vous aider.

Event Hubs est conçu pour l'assimilation massive de données. Vous pouvez inonder le service de millions de messages par seconde, et il se chargera de ces données pour vous sans effort. Ce service est en mesure de conserver les messages pendant sept jours ou de les conserver indéfiniment en les écrivant dans un magasin de données grâce à la fonctionnalité 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 si vous le souhaitez.



Faites un essai [Commencez à envoyer des messages à Azure Event Hubs dans .NET Standard.](#)

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 ; voir l'encadré dans le chapitre 1, « Qui utilise Azure ? » pour consulter un exemple), comme les thermostats ou les capteurs intelligents dans les voitures. Il dispose d'un grand nombre des mêmes propriétés qu'Event Hubs, comme la capacité de conserver les messages pendant sept jours et de les relire.

Ce qui rend IoT Hub unique, c'est qu'en plus de recevoir des messages provenant d'appareils, il peut également leur en envoyer. Il a la capacité de gérer l'intégralité de votre infrastructure IoT. IoT Hub vous permet d'inscrire des appareils et de générer des rapports sur leur état, de les gérer en les sécurisant et de les redémarrer, ainsi que d'envoyer des données à des appareils.



Faites un essai [Connectez votre appareil à votre hub IoT par l'intermédiaire de .NET.](#)

Azure Event Grid

[Azure Event Grid](#) propose un autre type de messagerie. Event Grid propose un service de publication/abonnement entièrement géré, compatible avec virtuellement tous les services dans Azure et capable de s'adapter aux 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 les files d'attente et les rubriques 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 dans et à l'extérieur d'Azure publient des événements (par exemple, lorsqu'un nouveau Blob est ajouté ou qu'un nouvel utilisateur est ajouté à un abonnement Azure). Azure Event Grid est capable de détecter ces événements et de les mettre à la disposition des gestionnaires d'événements et des services qui se sont abonnés à ces événements (Figure 3-1). 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.

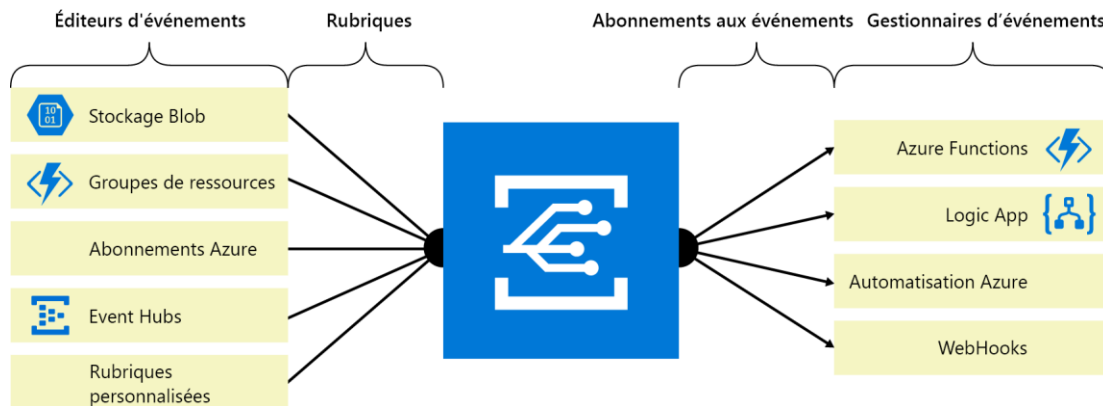


Figure 3-1 : Présentation d'Azure Event Grid

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, et non pas simplement pour sa présence.

Voici un exemple d'utilisation d'Azure Event Grid :

Vous souhaitez être notifié par e-mail chaque fois qu'un utilisateur est ajouté ou supprimé de votre liste de diffusion dans MailChimp.

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. L'étape suivante consiste à envoyer un e-mail contenant le nom de l'utilisateur qui a été ajouté ou supprimé et l'action effectuée (ajout ou suppression). Lorsqu'un nouvel utilisateur est ajouté à la liste de diffusion, Event Grid avertit Logic Apps, qui envoie un e-mail.



Faites un essai [Surveillez les modifications apportées aux machines virtuelles avec Azure Event Grid et Logic Apps.](#)

Pourquoi et quand l'utiliser ?

Azure fournit de nombreuses options permettant de créer des applications de messagerie et de découplage. Laquelle devez-vous utiliser et quand ? Le tableau 3-2 résume les différences pour vous aider à faire le tri.

Tableau 3-2 : quand utiliser quel service Azure pour les messages ou les événements

	Event Grid	Event Hubs	IoT Hub	Rubriques	Files d'attente Service Bus	Files d'attente de stockage
Assimilation d'événements	X	X	X			
Gestion des appareils			X			
Messagerie	X	X	X	X	X	X
Plusieurs consommateurs	X	X	X	X		
Plusieurs expéditeurs	X	X	X	X	X	X
Utilisation pour le découplage		X	X	X	X	X
Utilisation pour publication/abonnement	X					
Taille de message maximale	64 Ko	256 Ko	256 Ko	1 Mo	1 Mo	64 Ko

Sécurisation de votre application

La sécurité est aujourd'hui l'un des aspects les plus importants de n'importe quelle application, et il n'est pas évident de trouver la bonne solution. Fort heureusement, Azure fournit de nombreux services pour vous aider à sécuriser votre application. Nous examinons certains d'entre eux 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. L'authentification n'est pas nécessairement 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](#) fournit tous ces éléments et plus encore en configuration d'usine. Vous stockez vos identités d'utilisateurs dans Azure Active Directory 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 Active Directory s'occupe de la gestion des mots de passe, y compris pour les scénarios courants comme « J'ai oublié mon mot de passe ».

Azure Active Directory est utilisé quotidiennement par des millions d'applications, y compris le [portail Azure](#), [Outlook.com](#) et [Office 365](#). Pour cette raison, ce service est capable de détecter plus facilement les comportements malveillants et agir en conséquence. 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 Active Directory signale cela comme un comportement malveillant et demande à l'utilisateur des informations d'identification supplémentaires par l'intermédiaire de l'authentification multifacteur.

Gestion des API Azure

Les API doivent être sécurisées. Cela est vrai pour les API que vous créez vous-même ainsi que pour celles que vous utilisez et 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 ou celles provenant de fournisseurs tiers, qui ajoute des fonctionnalités telles que la mise en cache, la limitation et l'authentification/autorisation.

Vous pouvez sécuriser votre API avec la Gestion des API en demandant aux consommateurs de créer un abonnement à votre API. Ainsi, les applications appelantes 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.



Faites un essai [Mise en route avec la Gestion des API Azure.](#)

Azure Key Vault

Dans le cadre de votre architecture de sécurité, vous avez besoin d'un lieu sûr pour stocker les certificats, les clés et autres secrets. [Azure Key Vault](#) vous le fournit. Avec Key Vault, vous disposez d'un emplacement central où stocker les secrets que vos applications utilisent.

Il peut s'agir d'informations de connexion dans un `connectionstring` utilisé par votre application. Votre application obtiendrait le `connectionstring` à partir de 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.

Vous utilisez également Key Vault pour stocker des certificats comme les certificats SSL (Secure Sockets Layer) qui vous permettent de sécuriser le trafic vers et à partir de vos applications sur HTTPS.



Faites un essai [Mise en route avec Azure Key Vault.](#)

Protection Azure DDoS

Vous l'avez entendu trop souvent aux informations, et vous ne voulez certainement pas que cela arrive à votre entreprise : votre application est victime d'une attaque de déni de service distribué (DDoS). Ces types d'attaques deviennent de plus en plus fréquentes et peuvent submerger votre application au point que personne ne puisse plus l'utiliser.

Azure vous protège des attaques DDoS grâce à sa couche de protection DDoS Azure. Il s'agit d'une couche du réseau physique Azure qui n'est pas accessible aux clients. Cela fait partie de la plateforme Azure, et vous l'obtenez gratuitement avec tous vos services. Vous n'avez pas besoin de faire ni de configurer quoi que ce soit pour l'utiliser. Elle est là en permanence pour vous protéger.



Plus d'informations [En savoir plus sur la protection Azure DDoS.](#)

Pare-feu d'applications Web Azure

Vous devez sécuriser votre application contre les nombreuses menaces possibles, y compris celles définies dans la spécification Open Web Application Security Project (OWASP), telles que l'injection SQL et le script de site à site (XSS). [Le pare-feu d'applications Web Azure](#) peut vous y aider. Le pare-feu d'applications Web, une fonctionnalité du service [Azure Application Gateway](#), fournit une protection en temps réel de votre application.

Il peut détecter une attaque malveillante, telle que définie dans l'[ensemble de règles de base OWASP](#), et empêcher cette attaque d'atteindre votre application. Il génère également des rapports sur les tentatives d'attaques ou les attaques en cours, afin de vous permettre d'identifier les menaces qui tentent d'atteindre votre application. Cela fournit une couche supplémentaire de sécurité qui protège vos applications.

Chiffrement par défaut des données

Vous protégez vos magasins de données du mieux que vous pouvez en mettant en place des `connectionstrings` et des règles de pare-feu. Mais si un utilisateur malveillant parvient d'une façon ou d'une autre à accéder à vos données, vous souhaitez rendre ces dernières inutilisables par l'intermédiaire du chiffrement.

Par défaut, vos données sont chiffrées 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](#). Et tout cela fonctionne automatiquement. Vous n'avez strictement rien à faire pour le configurer ni pour chiffrer ou déchiffrer les données lorsque vous l'utilisez.

Où et comment déployer vos services Azure

Les services Azure avec lesquels vous choisissez de travailler déterminent vos options de déploiement et vice versa. Par conséquent, il est important de comprendre les options de déploiement dont vous disposez dans Azure et leurs ramifications.

La mentalité DevOps

La livraison rapide et la réitération rapide sont déterminantes pour la création d'un excellent logiciel. Par conséquent, votre nouveau code doit être fusionné avec le code de votre équipe, et le résultat doit être déployé et testé aussi rapidement et aussi souvent que possible pour vérifier que tout fonctionne et que ce que vous avez créé correspond aux besoins de l'utilisateur.

Un grand nombre des services Azure que nous avons examinés jusqu'à présent dans ce livre sont susceptibles de bénéficier d'un code qui leur est livré automatiquement, souvent par l'intermédiaire d'un pipeline de [livraison en continu](#) que vous avez configuré au sein du service.

Outre les fonctionnalités natives des services Azure, vous pouvez utiliser Visual Studio Team Services pour créer, tester et déployer votre application. Vous [pouvez facilement créer de nouveaux pipelines de création et de déploiement dans Visual Studio Team Services](#), mais également procéder à des opérations telles que l'automatisation des tests de chargement et la permutation des emplacements de déploiement en production.



Faites un essai [Mise en route avec Visual Studio Team Services.](#)

Modèles Azure Resource Manager

Tous les services Azure présentés dans ce guide sont basés sur Azure Resource Manager. Nous pouvons utiliser ce service pour exécuter un script qui transforme notre environnement en Infrastructure-as-Code (IaC), par l'intermédiaire de [modèles Resource Manager](#). Ces modèles sont des fichiers JSON qui décrivent ce que vous voulez déployer et les paramètres correspondants.

Vous pouvez créer des modèles Resource Manager dans Visual Studio et Visual Studio Code par l'intermédiaire du modèle de projet Azure Resource Group. Vous pouvez également générer des modèles 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 génère le modèle Resource Manager pour la ressource donnée et génère même le code permettant de créer la ressource par l'intermédiaire de l'interface de ligne de commande Azure (CLI), Windows PowerShell, .NET et Ruby.

Une fois que vous disposez d'un modèle Resource Manager, vous pouvez le déployer vers Azure par l'intermédiaire de Windows PowerShell, de la CLI Azure ou de Visual Studio. Sinon, vous pouvez automatiser son déploiement dans un pipeline en livraison continue par l'intermédiaire de Visual Studio Team Services.

Les modèles Resource Manager vous permettent de déployer des applications à exécuter sur la plateforme Azure, soit dans le Cloud public soit en local, sur Azure Stack.

Un excellent exemple de déploiement de ressources dans le Cloud par l'intermédiaire de Resource Manager est le bouton Déployer sur Azure que vous trouvez dans de nombreux référentiels GitHub, comme l'illustre la figure 5-1.

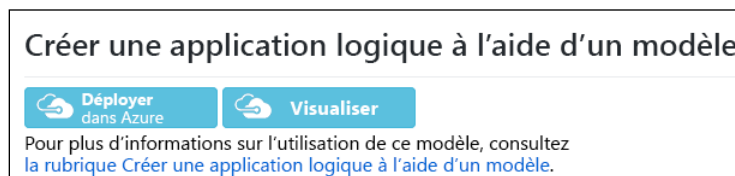


Figure 5-1 : déploiement via un bouton d'une application Azure Logic par l'intermédiaire d'un modèle Resource Manager dans GitHub

Azure Service Fabric

Une autre façon d'exécuter vos applications consiste à les exécuter sur [Azure Service Fabric](#). Service Fabric constitue le fondement que Microsoft utilise pour exécuter de nombreuses ressources Azure et les rendre hautement disponibles, performantes et capables d'auto-régénération. Vous pouvez désormais utiliser Service Fabric vous-même, pour héberger vos propres services.

Le kit de développement logiciel Azure Service Fabric vous permet de créer des applications pour Service Fabric. Vous pouvez également exécuter tout fichier exécutable dans Service Fabric, et vous pouvez même utiliser ce service pour héberger des conteneurs.

Service Fabric représente une excellente solution pour rendre vos applications tout aussi performantes, fiables et sûtes que de nombreux services Azure. Vous pouvez l'utiliser n'importe où : vous pouvez déployer Service Fabric dans Azure, localement, sur votre propre ordinateur et même sur des machines virtuelles dans d'autres Clouds.

Après avoir déployé votre application dans Azure Service Fabric, ce dernier vous fournit des avantages tels que ceux-ci :

- Équilibrage de charge
- Évolutivité automatique
- Haute disponibilité
- Auto-régénération
- Réplication et basculement
- Mises à niveau propagées
- Et restauration automatique

Conteneurs dans Azure

Conteneurs est l'un de ces mots 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 en fait 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 proposé par les conteneurs est qu'un conteneur individuel demeure toujours le même. Vous exécutez un conteneur localement lorsque vous développez votre application, et vous utilisez la même configuration de conteneur dans le Cloud ou 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 *ainsi* qu'en production. Avec les conteneurs, la vieille déclaration de secours 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, et [Docker](#) est l'une d'entre elles. Azure peut exécuter et gérer les conteneurs pour vous avec [Azure Container Instances](#) et [Azure Container Service](#), et même dans Service Fabric et [Azure App Service en exécution sur Linux](#).

Le tableau 5-1 illustre quel service vous pouvez utiliser pour un scénario donné lorsque vous utilisez des conteneurs.

Tableau 5-1 : choisir le service Azure à utiliser pour les conteneurs

	Azure Container Services	Azure Container Instances	Azure Service Fabric
Pour les déploiements en production de systèmes complexes (avec un orchestrateur de conteneurs)	X		
Pour l'exécution de configurations simples (éventuellement sans orchestrateur)		X	
Pour des scénarios d'usage de longue durée sur des conteneurs	X		
Pour des scénarios d'usage de courte durée sur des conteneurs		X	
Pour l'orchestration d'un système basé sur des conteneurs	X		X

	Azure Container Services	Azure Container Instances	Azure Service Fabric
L'orchestration avec des orchestrateurs open source (DC/OS , Docker Swarm , Kubernetes)	X		
L'orchestration avec un orchestrateur intégré			X



Remarque N'oubliez pas que lorsque vous utilisez des conteneurs, vous utilisez un produit IaaS (infrastructure as a service) et que vous êtes responsable du système d'exploitation, des correctifs, de l'équilibrage des charges et ainsi de suite.

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 avec lesquelles vous pouvez utiliser la CLI Azure, Windows PowerShell ou votre IDE préféré.

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

Où effectuer le déploiement et quand ?

Le tableau 5-2 résume les options de déploiement pour Azure. Notez qu'utiliser tous les services PaaS (platform as a service) que nous avons évoqués dans le présent guide est uniquement possible lorsque vous utilisez le Cloud public avec les modèles Resource Manager ou localement avec Azure Stack.

Tableau 5-2 : comparaison des options de déploiement Azure

	IaaS	PaaS
Sur site	Azure Stack Conteneurs Service Fabric	Azure Stack Service Fabric
Cloud public	Conteneurs Service Fabric	Service Fabric Modèles Resource Manager

Une visite guidée d'Azure

Maintenant que vous savez tout d'Azure et que vous connaissez les services qu'il propose, commençons à l'utiliser.

Visite guidée n°1 : le portail Azure

L'un des plus importants outils Azure est le portail Azure, que vous trouverez sur <https://portal.azure.com/>. Il s'agit de votre hub Azure central, où vous faites tout ce que vous voulez avec Azure. La plupart des opérations que vous pouvez effectuer dans le portail Azure, vous pouvez également les effectuer via l'API Azure, l'interface de ligne de commande Azure (CLI) et Azure PowerShell.

Vous pouvez bénéficier d'un aperçu du portail Azure en visitant <https://preview.portal.azure.com>. L'aperçu contient déjà des fonctionnalités qui ne sont généralement pas encore disponibles, mais que vous pouvez essayer et sur lesquelles vous pouvez commenter. Le reste de ce guide portera sur la version généralement disponible du portail, et non sur l'aperçu.

La première chose que vous verrez dans le portail Azure est un tableau de bord comportant des vignettes. Vous pouvez créer et personnaliser les tableaux de bord et les partager avec les membres de votre équipe ou les garder pour vous.

Vignettes du portail Azure

Les tableaux de bord contiennent des vignettes qui affichent des informations concernant un service, ou servent de raccourci pour accéder à un service. Vous trouverez ces vignettes, illustrées à la figure 6-1, partout dans le portail, sur les pages de tous les services. Elles peuvent être très utiles pour obtenir un aperçu rapide de la façon dont un service fonctionne.

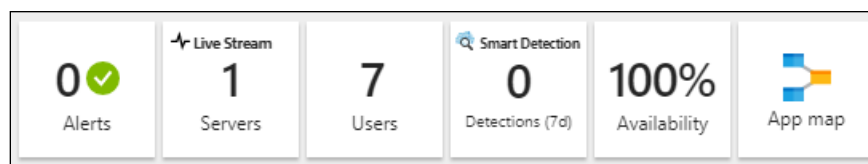


Figure 6-1 : vignettes informationnelles dans le portail Azure

Par exemple, le service Application Insights, par défaut, affiche des vignettes qui vous informent de l'existence d'alertes actives, de l'entrée de données en temps réel, du nombre d'utilisateurs ayant été actifs au cours des dernières 24 heures, de la disponibilité, et ainsi de suite. Il s'agit d'informations très utiles dans un format facilement assimilable. Vous pouvez personnaliser la taille des vignettes et les informations qu'elles contiennent. Vous pouvez également personnaliser l'apparence des graphiques en ajustant leurs chronologies, et leur faire afficher les données dans différents formats, comme des lignes ou des barres.

Vous pouvez également épingler des vignettes (figure 6-2) directement sur vos tableaux de bord, afin qu'elles soient la première chose que vous voyez quand vous accédez au portail. Vous pouvez, par exemple, épingler des vignettes de toutes les métriques des services qui vous intéressent, afin de créer un tableau de bord de surveillance que vous [partagez avec votre équipe](#), ou affichez sur un écran dans la salle de l'équipe.

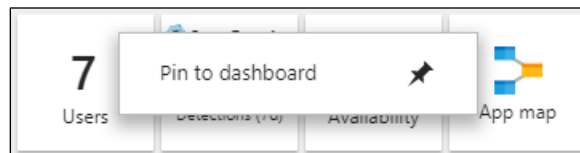


Figure 6-2 : vous pouvez épingler des vignettes sur un tableau de bord

Trouver des services

Les services Azure constituent le sujet central du portail Azure. Vous pouvez les ajouter et les trouver de plusieurs façons.

Pour créer de nouveaux services, dans le coin supérieur gauche de la fenêtre du portail, cliquez sur le signe plus (« + »). Cette opération ouvre la boîte de recherche pour le marché, où vous trouverez tout, d'une application Web aux serveurs Linux, comme l'illustre la figure 6-3.

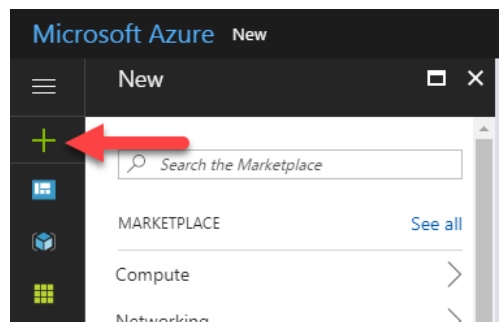


Figure 6-3 : le bouton Nouvelle ressource du portail Azure

Lorsque vous avez trouvé le service qui vous intéresse (figure 6-4), un assistant vous fournit les instructions nécessaires à la configuration et au déploiement.

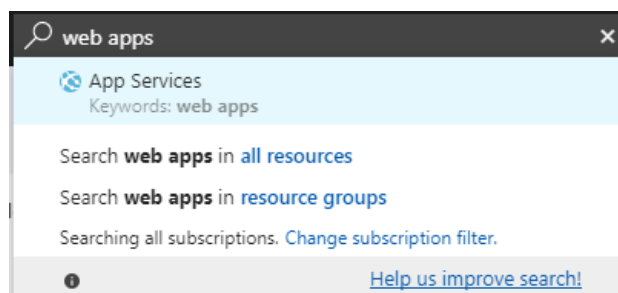


Figure 6-4 : résultats de recherche de ressource

Lorsque vous disposez de ressources, vous pouvez les trouver par l'intermédiaire de la fonction de recherche. La zone de recherche située en haut du portail (figure 6-5) vous permet de rechercher toutes vos ressources et d'y accéder directement.

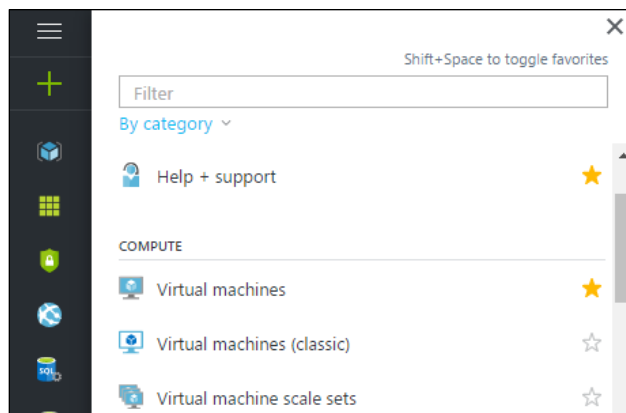


Figure 6-5 : le menu Favoris du portail Azure

Le menu Favoris se trouve dans le volet situé sur le côté gauche du portail. Ce menu affiche les catégories de ressources (représentées par leurs icônes), comme Azure App Service. 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 Favoris et en cliquant sur les étoiles correspondant à ces catégories.

Mieux comprendre les panneaux

Dans Azure, les pages sont également appelées *panneaux*. Les panneaux sont partout, et vous pouvez même les épingler à vos tableaux de bord. Lorsque vous ouvrez une application Web, vous voyez d'abord le panneau d'aperçu. Ce panneau spécifique vous fournit des outils permettant d'arrêter, de démarrer et de redémarrer l'application Web et d'afficher des vignettes montrant ses métriques, 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. Donc, si vous ouvrez le panneau Emplacements de déploiement et que vous cliquez sur le bouton 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.

Création d'une machine virtuelle

Nous allons utiliser le portail Azure pour créer une machine virtuelle qui exécute Red Hat Linux. Après l'avoir créée, nous vous montrerons comment la fermer et la supprimer afin de ne plus avoir à payer pour son utilisation.

Un mot au sujet des groupes de ressources

La machine virtuelle sera déployée dans un *groupe de ressources*. Toutes les ressources Azure résident dans des groupes de ressources. Un groupe de ressources est un conteneur logique qui contient vos ressources. Vous pouvez gérer la sécurité d'un groupe de ressources, et vous pouvez 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. Cela vous permet également de connaître facilement leur coût.

1. Dans le portail Azure, dans le coin supérieur gauche, cliquez sur le bouton Créer un service.

2. Dans la zone de recherche, tapez **machine virtuelle Linux**.

Vous obtiendrez de nombreux résultats de recherche, y compris Red Hat Enterprise Linux. Nous allons essayer sa dernière version.

3. Cliquez sur la dernière version de Red Hat Enterprise Linux.

4. Cliquez sur Créer.

L'assistant de création de machine virtuelle s'ouvre.

- a. Choisissez un nom pour la machine virtuelle.
- b. Choisissez le type de disque. Un disque SSD fournit une machine virtuelle plus rapide, mais est également plus onéreux. Pour cette visite guidée, choisissez SSD.
- c. Tapez un nom d'utilisateur.
- d. Pour le type d'authentification, sélectionnez Mot de passe.
- e. Tapez un mot de passe et confirmez-le.
- f. Dans la zone Groupe de ressources, tapez un nouveau nom.
- g. Choisissez l'emplacement de la machine virtuelle, puis cliquez sur OK.

5. Choisissez la taille de la machine virtuelle.

Les performances de la machine virtuelle déterminent son coût. Il existe de nombreuses options de taille pour les machines virtuelles, certaines sont de faible taille, d'autres sont de taille incroyablement importantes. L'assistant vous permet de sélectionner le nombre de cœurs et la quantité de mémoire souhaités, et de choisir des options en fonction de ces bases. En outre, d'autres fonctionnalités sont associées aux options de taille :

- a. Type de disque dur (SSD ou disque dur normal).
- b. Le montant maximal d'IOPS (opérations d'entrée/sortie par seconde). Cela détermine les performances de la machine virtuelle d'une manière significative, surtout si les applications que vous exécutez effectuent de nombreuses opérations de lecture et d'écriture à partir du disque dur et vers celui-ci.
- c. La quantité de disques de données qui peuvent être installés sur la machine virtuelle.
- d. La capacité à effectuer l'équilibrage de charge.
- e. La carte graphique qui est installée sur la machine virtuelle. Cela est utile si vous devez effectuer beaucoup de rendus de graphiques ou gérer des scénarios d'usage informatiques lourds.

6. Une fois que vous avez sélectionné la taille, vous pouvez configurer des paramètres supplémentaires tels que le réseau virtuel et l'adresse IP. Vous pouvez également installer des extensions sur la machine. Pour l'instant, laissez tout en l'état, puis cliquez sur OK.

7. Consultez le résumé, acceptez les termes, puis cliquez sur Acheter.

La machine virtuelle va maintenant être déployée. Cela ne prend généralement que quelques minutes.

Connectez-vous à la machine virtuelle par l'intermédiaire de Secure Shell

Maintenant que vous avez créé une machine virtuelle fonctionnant avec Red Hat Linux, vous pouvez vous y connecter et commencer à l'utiliser. Vous pouvez utiliser divers outils comme SSH (Secure Shell) ou même utiliser un [bureau à distance dans la machine virtuelle](#). Nous allons utiliser SSH pour nous connecter à la machine virtuelle par l'intermédiaire du Cloud Shell dans le portail Azure.

Cloud Shell est l'interface de ligne de commande (CLI) dans le navigateur. Vous n'avez plus besoin d'installer quoi que ce soit sur votre ordinateur. Cloud Shell utilise la CLI Azure version 2.0. Vous pouvez utiliser Bash ou PowerShell dans Cloud Shell pour interagir avec vos ressources Azure.

Pour démarrer Cloud Shell, dans le coin supérieur droit du portail Azure, cliquez sur le bouton illustré ici :



Pour vous connecter à la machine virtuelle, vous avez besoin de son adresse IP. Trouver la machine virtuelle dans le portail Azure (soit en la recherchant, soit en naviguant jusqu'à elle). Dans le coin supérieur gauche du panneau de présentation de la machine virtuelle, cliquez sur le bouton de connexion (voir la figure 6-6). Lorsque vous cliquez dessus, le panneau affiche l'adresse IP à utiliser pour se connecter à la machine virtuelle, comme l'illustre la figure 6-6. Copiez l'adresse IP.

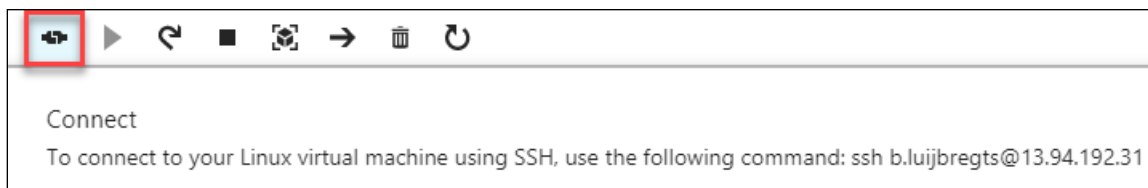


Figure 6-6 : se connecter à une machine virtuelle dans le portail Azure

Vous pouvez désormais utiliser cette commande directement dans Cloud Shell. Il vous suffit de la coller et d'appuyer sur Entrée. Un avertissement s'affiche pour vous informer qu'il n'y a pas de clé d'authentification. C'est parce que nous avons choisi de travailler plutôt avec un mot de passe. Sélectionnez Oui, puis tapez votre mot de passe. Vous y êtes ! Vous êtes désormais connecté à Red Hat Linux Enterprise, et vous pouvez commencer à l'utiliser. La figure 6-7 illustre que nous sommes connectés à la machine virtuelle via Cloud Shell et parcourons le système de fichiers :

```
Bash v | [power] ? [gear]
Requesting a Cloud Shell. Succeeded.
Connecting terminal...

barry@Azure:~$ ssh b.luijbregts@13.95.29.180
Password:
Last login: Thu Sep 28 07:00:23 2017 from 52.170.90.72
[b.luijbregts@myredhatserver ~]$ ls -al /etc
total 1132
drwxr-xr-x. 84 root root    8192 Sep 28 06:54 .
dr-xr-xr-x. 17 root root    224 Aug  9 23:05 ..
drwxr-xr-x.  3 root root    101 Aug  9 23:01 abrt
-rw-r--r--.  1 root root     16 Aug  9 23:03 adjtime
-rw-r--r--.  1 root root   1518 Jun  7  2013 aliases
drwxr-xr-x.  2 root root     69 Aug  9 23:05 alternatives
-rw-----.  1 root root    541 Mar 29 16:34 anacrontab
-rw-r--r--.  1 root root     55 Mar  1  2017 asound.conf
-rw-r--r--.  1 root root      1 May 23  2016 at.deny
```

Figure 6-7 : parcourir le système de fichiers Linux par l'intermédiaire de Cloud Shell dans le portail Azure

Ajout de stockage supplémentaire à la machine virtuelle

Si vous utilisez votre machine virtuelle comme serveur de base de données, vous souhaitez souvent disposer d'espace de stockage supplémentaire pour gérer les données de la base de données. Vous pouvez créer cet espace de stockage supplémentaire en ajoutant un disque supplémentaire via le portail Azure.

1. Accédez à la machine virtuelle, sélectionnez l'onglet Disques, puis cliquez sur Ajouter un disque de données, comme l'illustré la figure 6-8.

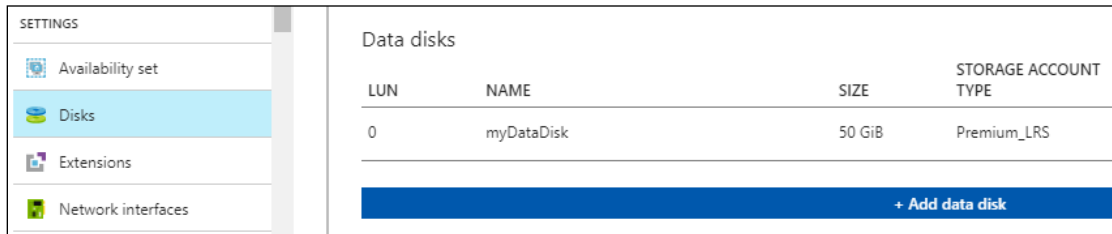


Figure 6-8 : l'onglet Disque d'une machine virtuelle dans le portail Azure

Étant donné que nous ne disposons pas de disque existant à associer, sélectionnez Créer un disque.

2. L'assistant de création d'un disque géré s'affiche. Cet assistant vous permet de créer un disque géré (plutôt qu'un disque non géré).

Les disques gérés le sont entièrement par Azure. Vous n'avez pas besoin de créer et d'entretenir un compte de stockage sous-jacent pour ceux-ci. Vous pouvez également choisir la taille du disque et son contenu, mais nous allons tout laisser en l'état. Tapez un nom, puis cliquez sur Créer. Vous disposez maintenant d'un disque supplémentaire associé à votre machine virtuelle.

Vous pouvez également créer et associer un nouveau disque via Cloud Shell ou la CLI Azure en utilisant cette commande :

```
az vm disk attach -g myResourceGroup --vm-name myVM --disk myDataDisk --new --size-gb 50
```

Si vous avez besoin d'aide en ce qui concerne la CLI Azure, vous pouvez taper **az -h**. Toutes les commandes de la CLI Azure suivent la même structure : nom(s) az (par ex., vm) verbe (par ex., attach) options.

Une fois que vous avez créé un disque pour votre machine virtuelle, vous devez l'initialiser dans Linux. [Ce document](#) vous guide tout au long du processus.

La création de disques supplémentaires pour disposer d'espace de stockage supplémentaire sur les machines virtuelles est une opération simple. Les étapes suivantes peuvent s'avérer plus complexes, en fonction du système d'exploitation que vous utilisez.

Nettoyage des ressources de la visite guidée

Si vous en avez terminé avec la machine virtuelle, vous pouvez la fermer et la supprimer en supprimant le groupe de ressources que nous avons créé au début du processus de création de la machine virtuelle. Celui-ci contient la machine virtuelle et toutes les autres ressources qui sont créées automatiquement. Une fois le groupe de ressources supprimé, vous ne payez plus pour aucune des ressources que vous avez utilisées lors de cette visite guidée.

Visite guidée n°2 : développement d'une application Web Node.js avec MongoDB sur Azure

Au cours de cette visite guidée, nous déployons une application Node.js simple, qui se connecte à MongoDB, et nous allons l'héberger dans Azure. Nous allons héberger l'application dans Azure Web App et utiliser Azure Cosmos DB comme notre base de données, car elle est dotée d'une interface MongoDB.

Pour suivre cette visite, vous devez avoir installé sur votre ordinateur la [CLI Azure \(v 2.0 ou supérieure\)](#), [Git \(v 2 ou supérieure\)](#), [Node.js et NPM \(v 6 ou supérieure\)](#), ainsi que [MongoDB \(3.4 ou supérieure\)](#).

Nous allons utiliser un exemple d'application qui utilise l'infrastructure Web [MEAN.js](#).

Création de l'application Web et de la base de données par l'intermédiaire du portail Azure

Pour héberger l'application Node.js, nous allons créer une nouvelle application Web dans Azure Web App.

1. Dans le portail Azure, cliquez sur le bouton Création d'un nouveau service.
2. Recherchez l'application Web. Le panneau de l'application Web s'affiche. Cliquez sur Créer pour commencer.
3. Le panneau de création d'application Web s'ouvre.
 - a. Attribuez un nom à l'application Web.
 - b. Créez un groupe de ressources en lui attribuant un nom.
 - c. Sélectionnez le système d'exploitation souhaité pour l'exécution (Windows ou Linux).
 - d. Sélectionnez ou créez un plan de service d'application, puis cliquez sur créer.

L'application Web s'exécute sur les plans de service d'application

Les App Services, comme les applications Web, s'exécutent sur les plans de service d'application. Un plan de service d'application est une représentation abstraite des ressources et des fonctions, comme le processeur et la mémoire sont représentés dans les niveaux de tarification. Les plans de service d'application sont également liés à une zone géographique spécifique que vous choisissez. Vous pouvez, par exemple, exécuter votre application Web dans un plan de service d'application de tarification niveau S1 (voir la Figure 6-9), qui dispose d'un cœur et de 1,75 Go de RAM.

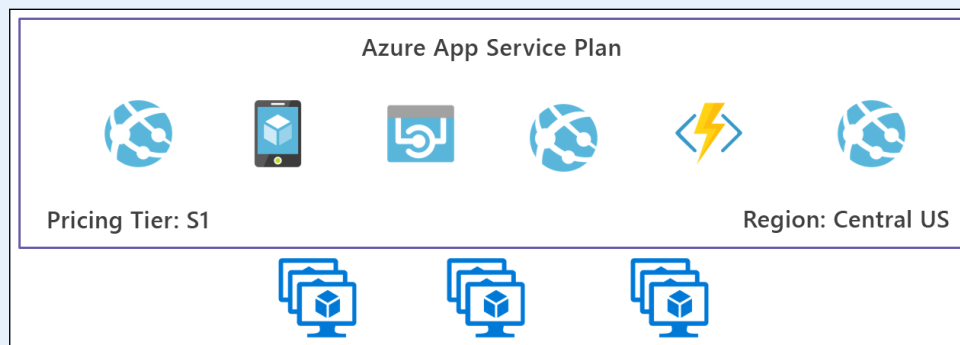


Figure 6-9 : Illustration du plan de service d'application Azure

Vous pouvez exécuter autant d'App Services que vous souhaitez sur un plan de service d'application, mais vous devez être conscient que vous devez partager les ressources entre tous les App Services.

Pour héberger la base de données, créez une base de données de Cosmos DB qui utilise l'API MongoDB. Le fonctionnement est identique à celui de MongoDB, avec tous les avantages offerts par Cosmos DB.

1. Dans le portail Azure, cliquez sur le bouton Création d'un nouveau service.
2. Recherchez Cosmos DB. Le panneau Cosmos DB s'ouvre. Cliquez sur Créer pour commencer.
 - a. Attribuez un nom (ID) au compte de base de données.
 - b. Sélectionnez l'API MongoDB.
 - c. Sélectionnez le groupe de ressources que vous avez créé pour l'application Web.
 - d. Sélectionnez l'emplacement. Il doit de préférence correspondre à celui de l'application Web ou en être proche.
 - e. Cliquez sur Créer.

Exécuter l'application Node.js et la base de données MongoDB localement

Vous allez maintenant exécuter l'application localement avant de l'exécuter dans Azure.

Si vous ne l'avez pas déjà fait, [installez MongoDB Community Edition](#) :

1. Ouvrez une invite de commandes sur votre ordinateur et accédez au répertoire bin de votre installation MongoDB. Ce répertoire doit ressembler à C:\Program Files\MongoDB\Server\3.4\bin.
2. Tapez **mongo** pour vous connecter au serveur local MongoDB.
3. Si la connexion échoue, vérifiez que [votre base de données MongoDB a été démarrée](#). Dans la plupart des cas, vous devez exécuter mongod pour la démarrer.

Maintenant, clonez l'application de démarrage Node.js et exécutez-la :

4. Dans la fenêtre de commande, accédez au répertoire vers lequel vous souhaitez télécharger la source de l'application.
5. Exécutez `git clone https://github.com/Azure-Samples/meanjs.git` pour cloner l'application.
6. Démarrez l'application en exécutant les commandes suivantes :

```
cd meanjs
npm install
npm start
```

7. Une fois l'application complètement chargée, un message contenant son URL s'affiche. Ouvrez un navigateur et accédez à l'adresse du serveur (qui doit ressembler à `http://localhost:3000`). Une fenêtre similaire à celle de la Figure 6-10 devrait s'afficher.

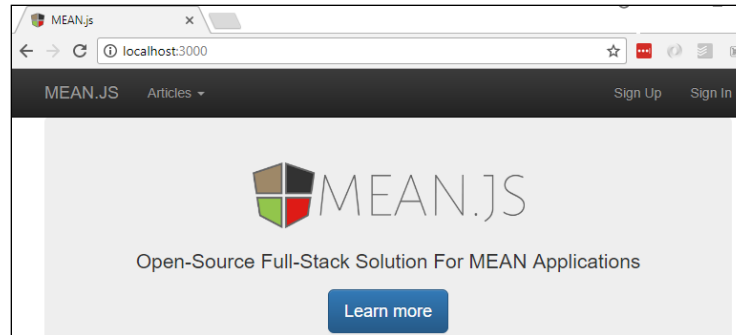


Figure 6-10 : Application Mean.JS standard

8. Dans la barre de menu, cliquez sur Sign Up (S'inscrire), puis essayez de créer un nouvel utilisateur. Si la procédure fonctionne, l'application peut accéder à la base de données MongoDB locale, dans laquelle elle inscrit les utilisateurs.

Déployer l'application Node.js dans Azure

Vous disposez maintenant d'une application exécutée localement. Vous pouvez à présent la déployer dans Azure. Pour commencer, vous devez déployer l'application Node.js dans l'application Web que vous avez créée précédemment.

Pour cela, vous allez transférer l'application de Git vers Azure à l'aide de l'interface de ligne de commande d'Azure. Assurez-vous que vous disposez de la version 2.0 ou une version ultérieure de l'interface de ligne de commande d'Azure sur votre ordinateur. Vous devez ensuite exécuter la commande `az login` suivante pour vous connecter à votre abonnement Azure. Cette commande ouvre un site Web sur lequel vous pouvez vous authentifier et autoriser l'interface de ligne de commande à utiliser de votre abonnement Azure.

Pour le FTP et le Git local, vous devez avoir un utilisateur de déploiement configuré sur le serveur pour authentifier votre déploiement. Cet utilisateur de 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. Dans l'interface de ligne de commande d'Azure, exécutez la commande suivante, où *nom d'utilisateur* et *mot de passe* sont des valeurs que vous devez remplacer (n'incluez pas les caractères < et >) :

```
az webapp deployment user set --user-name <nom d'utilisateur> --password <mot de passe>
```

Ensuite, associez le dépôt Git local à l'application Web à l'aide de la commande suivante :

```
az webapp deployment source config-local-git --name <nom de l'application> --resource-group <nom du groupe de ressources>
```

Cette commande renvoie une URL sous la forme suivante :

```
https://<nom d'utilisateur>@<nom de l'application>.scm.azurewebsites.net:443/<nom de l'application>.git
```

Utilisez maintenant cette URL pour transférer l'application dans Azure. Tout d'abord, ajoutez un accès distant Azure au dépôt Git local. Accédez au répertoire meanjs contenant le code source de l'application à l'aide de la commande `cd`. Maintenant, exécutez la commande suivante :

```
git remote add azure <url pour le déploiement>
```

Pour déployer l'application, exécutez `git push azure master`. Vous êtes invité à entrer les identifiants que vous avez créés précédemment. Le déploiement peut prendre un certain temps. Une fois cette opération terminée, vous recevez une notification.

Vous pouvez alors accéder à l'application à l'aide de l'URL `http://<nom de l'application>.azurewebsites.net`.

Maintenant, l'application est déployée au niveau de l'application Web. Vous disposez également d'une base de données Cosmos DB exécutable via l'API MongoDB, mais elle n'est pas encore associée à l'application. Voici comment procéder.

Connecter l'application et la base de données

Pour que l'application puisse se connecter à la base de données, elle doit en connaître l'emplacement. Vous devez donc renseigner la variable d'environnement `MONGODB_URI` dans l'application web. Pour ce faire, saisissez la commande suivante dans l'interface de ligne de commande d'Azure :

```
az webapp config appsettings update \  
--name <nom de l'application> \  
--resource-group myResourceGroup \  
--settings MONGODB_URI="mongodb://<nom de la base de données Cosmos DB>:<clé principale  
primaire>@<nom de la base de données Cosmos DB>.documents.azure.com:10250/mean?ssl=true"
```

La chaîne à utiliser comme valeur `MONGODB_URI` pour la connexion à Cosmos DB se trouve sur le portail Azure, dans le menu Chaîne de connexion du compte Cosmos DB.

Une fois que vous avez exécuté la commande via l'interface de ligne de commande d'Azure, les paramètres de l'application Web sont similaires à ceux illustrés dans la Figure 6-11.

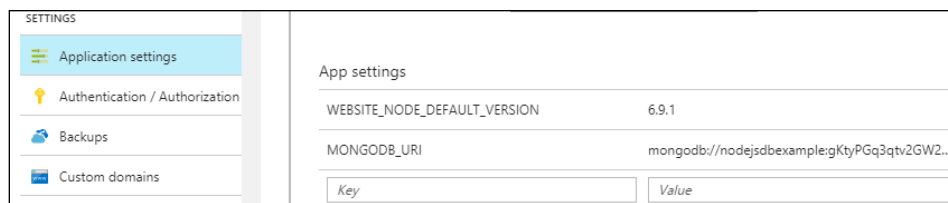


Figure 6-11 : Paramètres de l'application Web Azure

Notez qu'au lieu d'exécuter la commande dans l'interface en ligne de commande d'Azure, vous auriez également pu ajouter ces paramètres via le portail Azure.

Maintenant, accédez à l'application Web en saisissant l'URL `http://<nom de l'application>.azurewebsites.net`. Essayez d'inscrire un nouvel utilisateur. Si cette opération réussit, cela signifie que l'application communique avec la base de données. Maintenant, vous disposez d'une application Node.js avec une base de données Azure Cosmos DB qui utilise l'API MongoDB.

Configuration d'une livraison en continu avec GitHub

Jusqu'à présent, vous deviez transmettre votre code du 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 Visual Studio Team Services ou GitHub.

Vous allez donc, à l'aide de GitHub, transmettre votre code à votre application Web et les associer de manière que les modifications soient automatiquement déployées via un flux de livraison continue (CD, Continuous Delivery).

1. Accédez à <https://github.com/> et créez un nouveau dépôt.
Pour ce faire, vous devez vous connecter ou créer un compte.
2. Créez un nouveau dépôt à l'adresse <https://github.com/new>.
 - a. Attribuez un nom au dépôt.

- b. Laissez les autres paramètres tels quels (dépôt public, ne pas créer de README).
- c. Créez le dépôt. Un écran tel que celui de la Figure 6-12 s'affiche.

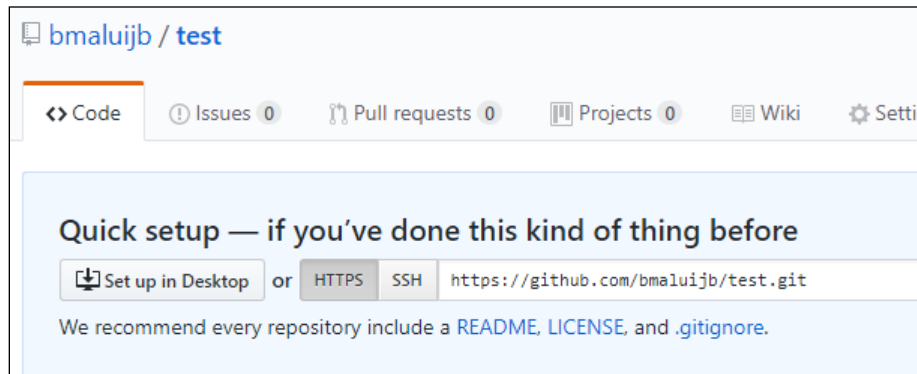


Figure 6-12 : Lien vers le dépôt dans GitHub

3. Utilisez cette URL pour associer une destination distante au dépôt local Git. Pour ce faire, vous pouvez utiliser la fenêtre de commande.
 - a. Accédez au répertoire meanjs du code source de l'application à l'aide de la commande `cd`.
 - b. Exécutez la commande suivante :

```
git remote add github https://github.com/bmaluijb/test.git
```
 - c. Exécutez la commande suivante pour transmettre le code GitHub :

```
git push github
```

Maintenant, votre code est accessible dans GitHub, et vous pouvez le partager avec votre équipe. Ensuite, vous devez configurer la livraison CD. Vous pouvez utiliser la fonction Options de déploiement de l'application Web via le portail Azure. Il est également possible d'utiliser la fonction Livraison continue dans l'application Web, mais pour cela vous devez disposer d'un compte Visual Studio Team Services. La première option est donc la plus simple.

1. Dans le portail Azure, accédez à l'application Web qui héberge l'application Node.js.
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, vous devez cliquer sur Déconnecter.
4. Dans Choisir une source, sélectionnez GitHub.
5. Dans la section Autorisation, autorisez Azure à utiliser GitHub. Pour ce faire, cliquez sur Autoriser et accordez la permission correspondante.
6. Dans la section Choisir un projet, sélectionnez le dépôt GitHub que vous venez de créer.
7. Laissez l'option Branche définie sur la branche maîtresse.
8. Cliquez sur OK et patientez quelques secondes.

Revenez au menu Options de déploiement. Comme vous pouvez le voir, la connexion a bien été effectuée. Dorénavant, chaque fois que vous transmettez une nouvelle version du code source sur GitHub, elle sera automatiquement générée et déployée sur l'application Web, comme illustré dans la Figure 6-13. Vous pouvez également forcer ce processus en cliquant sur le bouton de synchronisation.

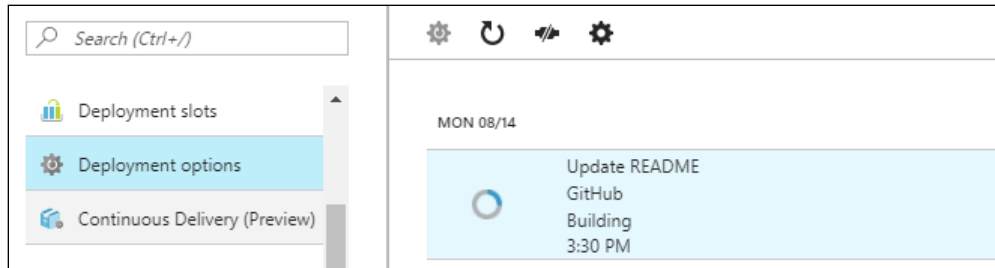


Figure 6-13 : Panneau Option de déploiement pour l'application Web

Configuration des environnements de tests ou de préproduction

À l'aide de l'application Web, vous pouvez configurer un emplacement de test pour les nouvelles versions de votre application. Vous pouvez utiliser des [emplacements de déploiement](#). Les emplacements de déploiement sont des App Services à part entière que vous pouvez utiliser pour tester votre code avant de le basculer vers l'emplacement suivant.

Les emplacements de déploiement sont utiles aussi bien que pour les tests que pour la préproduction ou encore la production (il s'agit toujours de l'App Service d'origine, donc en l'occurrence de l'application Web Node.js). 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 un même plan de service d'application. Ce sont ces éléments qui vous sont facturés. En effet, gardez à l'esprit que plus vous ajoutez d'emplacements de déploiement dans un plan de service d'application, plus vous consommerez de ressources comme le processeur et la mémoire.

Pour créer un emplacement de déploiement, cliquez sur Emplacements de déploiement dans le menu de l'application Web. Assurez-vous que votre application Web est exécutée avec un niveau de tarification Standard ou Premium, car l'option gratuite ne propose pas d'emplacements 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 et, une fois satisfait, *l'échanger* avec l'emplacement de production. Vous pouvez ainsi éprouver l'application et la déployer sans période d'interruption.

Vous allez à présent créer un emplacement de déploiement et y intégrer l'application.

1. Dans le portail Azure, accédez à l'application Web qui héberge l'application Node.js.
2. Dans la barre de menu, cliquez sur Emplacements de déploiement. Le panneau des emplacements de déploiement s'ouvre.
3. Cliquez sur le signe plus (+) pour créer un nouvel emplacement de déploiement.
 - a. Attribuez un nom à l'emplacement (par exemple, test).
 - b. Pour la Source de configuration, sélectionnez l'application Web Node.js (les paramètres de l'application seront copiés dans le nouvel emplacement).
 - c. Cliquez sur OK pour créer l'emplacement.

4. Une fois l'emplacement créé, il est similaire à l'application Web d'origine.
 - a. Configurez la livraison CD en suivant la même procédure que pour l'application Web.
 - b. Désactivez la connexion avec la livraison CD dans l'application Web Node.js d'origine. De cette façon, lorsque vous transmettez un nouveau code, il est livré uniquement dans l'emplacement de test.
5. Apportez une modification à l'application Node.js.
 - a. Modifiez une partie du texte dans le fichier `home.client.view.html` (il se trouve dans le dossier `meanjs\modules\core\client\views`).
 - b. Validez votre modification dans Git transmettez la nouvelle version dans GitHub, de la même manière que lorsque vous avez déployé l'application Node.js.

La nouvelle version est maintenant dans l'emplacement de test 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 Node.js et à l'URL de l'emplacement de test (que vous pouvez trouver dans le panneau d'aperçu de l'emplacement, comme pour l'application Web d'origine).

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

1. Dans le portail Azure, accédez à l'application Web Node.js.
2. Dans la barre de menu, cliquez sur Emplacements de déploiement pour ouvrir le panneau des emplacements de déploiement.
3. Cliquez sur le bouton Échanger pour ouvrir le panneau correspondant.
 - a. Ne modifiez pas les paramètres par défaut.
 - b. Cliquez sur OK pour commencer l'échange.

Une fois l'opération terminée, la nouvelle version de l'application Web Node.js est en production (pour le vérifier, accédez à l'URL de l'application Web Node.js). Il s'agit là d'une utilisation très 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 période d'interruption.

Diagnostiques supplémentaires avec les journaux de diagnostic

Lorsque l'application est en cours d'exécution, il est essentiel de connaître ses performances. Les journaux de diagnostic sont un moyen très efficace de contrôler l'application Web. Vous pouvez même les afficher dans la fenêtre de console. Pour ce faire, exécutez la commande suivante dans l'interface de ligne de commande d'Azure :

```
az webapp log tail --name <nom de l'application> --resource-group myResourceGroup
```

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

Configurer une surveillance et des alertes sophistiquées

Un autre moyen excellent de contrôler le bon fonctionnement de votre application consiste à utiliser [Application Insights](#). Il s'agit d'un outil de surveillance dans Azure qui vous permet de vous tenir au courant du fonctionnement de votre application, par exemple, combien de visiteurs ont utilisé votre application ou encore combien d'exceptions se sont produites ainsi que les parties du code concernées. Contrairement aux journaux de diagnostic, Application Insights est disponible à un prix modique.

Vous pouvez configurer Application Insights dans le portail Azure à partir de l'application Web.

1. Accédez au portail Azure, puis à l'application Web qui héberge l'application Node.js.
2. Dans la barre de menu, cliquez sur Application Insights.
3. Sélectionnez Créer une ressource.
 - a. Saisissez un nom et sélectionnez un emplacement pour l'instance Application Insights.
 - b. Cliquez sur OK. Application Insights est à présent déployée et commence à collecter les données de l'application.

Vous devez configurer votre application pour commencer à envoyer des données à Application Insights. Pour l'application Node.js d'exemple que vous avez créée, vous devez effectuer les opérations suivantes :

1. Dans l'invite de commandes, accédez au dossier meanjs de l'application Node.js à l'aide de la commande `cd`.
2. Exécutez `npm install applicationinsights --save`.
3. Accédez au portail Azure et récupérez la clé d'instrumentation Application Insights.
Elle est disponible dans l'instance Application Insights, sous Propriétés.
4. Ouvrez le fichier `server.js` de l'application Web Node.js.
5. Ajoutez les trois lignes de code JavaScript suivantes *avant* la ligne :

```
var app = require('./config/lib/app');  
const appInsights = require("applicationinsights");  
appInsights.setup("<clé d'instrumentation>");  
appInsights.start();
```

6. Transmettez les modifications dans GitHub et commencez le déploiement automatique. Dans l'invite de commandes, exécutez la commande suivante :

```
git commit -a -m "added Application Insights support"
```
7. Exécutez `git push github`, et les modifications seront transmises dans GitHub et déployées automatiquement dans 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 fonctionnalité puissante détecte les anomalies et vous en avertit. Elle détecte, par exemple, les augmentations subites de demandes non abouties et les ralentissements inhabituels de l'application. Vous pouvez également créer vos propres événements personnalisés pour toutes sortes de statistiques et de conditions dans le menu Alertes d'Application Insights, comme illustré dans la Figure 6-14.

NAME	CONDITION	LAST ACTIVE
MYGITHUBTEST (COMPONENTS)		
Failed requestspers second	Failed Requests per Second > 1	Never

Figure 6-14 : Alertes Application Insights

Maintenant, accédez au portail Azure, repérez Application Insights et cliquez dessus. Un récapitulatif s'affiche. Il vous présente des statistiques de base comme le temps de réponse du serveur, le temps de chargement des pages vues, le nombre de demandes envoyées au serveur et le taux de demandes non abouties. Si vous voyez des données sur cette page, cela signifie qu'Application Insights fonctionne correctement.



Remarque : Vous pouvez collecter des informations supplémentaires via Application Insights. Pour savoir comment procéder, suivez [ce guide](#).

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 le volume de trafic nécessaire. Lorsque le nombre d'utilisateurs est réduit, vous devez effectuer une nouvelle mise à l'échelle pour économiser des frais. Pour ce faire, vous pouvez utiliser la fonctionnalité de mise à l'échelle automatique de l'App Service. Sachez que cette fonctionnalité est disponible uniquement si votre application Web utilise un niveau de tarification Standard ou Premium.

Au niveau de l'application Web, une option Monter en charge est disponible dans le menu, comme illustré dans la Figure 6-15. Elle vous permet de mettre votre application à l'échelle manuellement ou automatiquement.

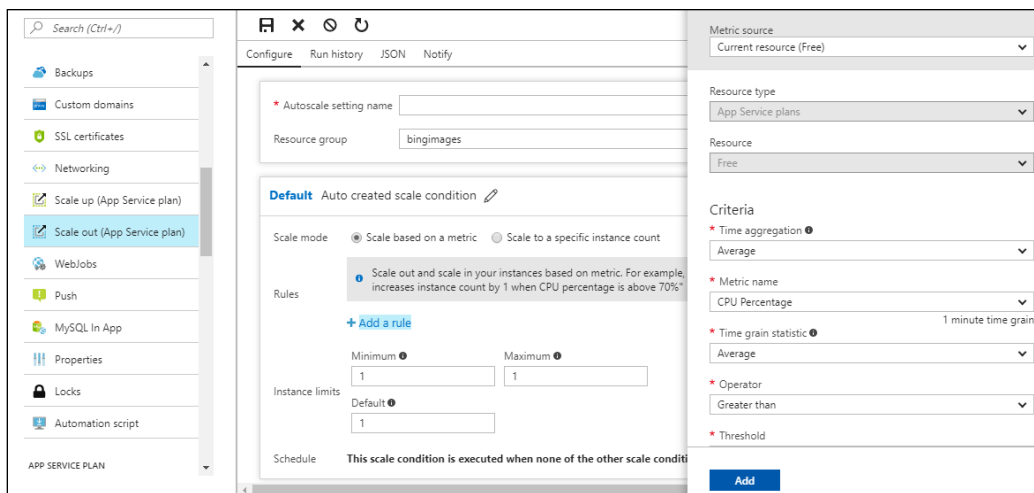


Figure 6-15 : Mise à l'échelle d'une application Web via le portail Azure

La montée en charge consiste à ajouter des instances de votre application pour gérer le trafic. Si vous effectuez une mise à l'échelle automatique, vous pouvez prendre en compte des statistiques comme le pourcentage d'utilisation du processeur ou de la mémoire, définir un calendrier (tous les jours à 17 h, par exemple) ou les deux. La mise à l'échelle automatique est très facile à configurer et à contrôler.

Passons à la production ! Ajout d'un protocole Secure Sockets Layer

À présent que l'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 choses les plus importantes que vous pouvez effectuer. En effet, sans HTTPS, des intrus peuvent observer le trafic entre vos ressources et l'utiliser à des fins négatives (comme se connecter à votre application). De plus, HTTPS constitue une exigence pour certaines fonctions de pointe comme les [opérateurs de services](#).

L'acheminement de trafic vers votre application Web au moyen du protocole SSL (Secure Sockets Layer) est possible en important un certificat SSL dans l'application Web et en le liant à l'un de vos noms de domaine (personnalisé). Vous pouvez importer votre propre certificat SSL acheté par vous ou votre entreprise, ou en acheter un nouveau sur le site Web de [certificats Azure App Service](#), qui facilite l'achat et la validation du certificat. 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.

Procédure pas à pas n° 3 : création d'un back-end pour votre application mobile avec Azure

De nos jours, presque tout le monde possède un appareil mobile. Pour la plupart des applications, le trafic provient majoritairement d'appareils mobiles, et cette tendance ne cesse de se développer. Il ne suffit pas de créer une application mobile pour votre entreprise, encore faut-il proposer une expérience solide et fiable. C'est à ce stade qu'Azure peut être utile, en fournissant un back-end avancé via [la fonctionnalité Mobile Apps d'Azure App Service](#).

Les applications Azure Mobile App se composent de deux parties :

- Un back-end, qui fournit des API permettant de communiquer avec une base de données et d'effectuer certaines actions comme l'authentification, la synchronisation hors connexion et les notifications push
- Des SDK pour toutes sortes de plateformes, qui permettent de se connecter facilement au back-end

Vous pouvez générer le back-end de l'application créée via Mobile Apps en Node.js et C#. Le SDK client est disponible pour pratiquement toutes les plateformes mobiles couramment utilisées, ainsi que pour les applications mobiles inter-plateformes, natives et hybrides. Il est disponible pour Android, Cordova, iOS, Windows, Xamarin.Android, Xamarin.Forms et Xamarin.iOS.

Dans cette procédure pas à pas, vous allez créer un nouveau back-end d'application mobile avec C#, à utiliser avec une application Android créée avec Xamarin.Android.

Ensuite, vous devrez installer la charge de travail [Développement mobile en .NET](#) à l'aide du programme d'installation de [Visual Studio 2017](#). Cette procédure permet d'installer les outils Xamarin pour Android et les SDK nécessaires.

Cette procédure [est également disponible](#) si vous travaillez sur un Mac dans Visual Studio pour Mac.

Créer une application mobile via le portail Azure

La mise en route est facile. Vous allez maintenant créer le back-end mobile et une application de démarrage pour le client via le portail Azure.

1. Sur le portail Azure, cliquez sur Création d'un nouveau service puis, dans la zone de recherche, tapez **Guide de démarrage rapide application mobile**.
2. Sélectionnez le résultat correspondant, puis cliquez sur Créer.

L'assistant de création d'applications mobiles s'ouvre.

- a. Attribuez un nom à votre application mobile.
- b. Créez un nouveau groupe de ressources.

- c. Créez un nouveau plan App Service pour l'exécution de l'application mobile. Vous pouvez utiliser le niveau tarifaire Gratuit pour cette procédure pas à pas.
- d. Laissez le paramètre Application Insights désactivé.
- e. Cliquez sur Créer. Très bien ! L'application mobile va maintenant être déployée.

Une fois l'application mobile déployée, vous pouvez commencer à l'utiliser. Sur le portail Azure, dans l'application mobile, cliquez sur l'élément de menu Démarrage rapide, puis sélectionnez Xamarin.Android. L'assistant Démarrage rapide s'ouvre.

Étape 1 : connexion à une base de données

Pour stocker des données, vous devez vous connecter à une base de données. Cliquez sur l'icône d'information pour démarrer. Dans l'onglet Connexions de données qui vient de s'ouvrir, cliquez sur Création d'un nouveau service. Dans l'assistant Ajouter la connexion de données qui s'ouvre, procédez comme suit :

1. Pour Type, sélectionnez Base de données SQL.
2. Créez une nouvelle base de données avec un nouveau serveur cible.
3. Ne modifiez pas la chaîne de connexion.
4. Cliquez sur OK.

La connexion de données est créée.

Étape 2 : création d'une API de tableau

Pour communiquer avec la base de données, vous avez besoin d'une API. Pour créer un back-end d'API hébergé dans votre application mobile, procédez comme suit :

1. Dans la zone Étape 2, sélectionnez C# et cliquez sur Télécharger. Cela permet de télécharger le projet de back-end que vous allez déployer dans l'application mobile.
2. Décompressez le fichier que vous venez de télécharger, puis ouvrez-le avec Visual Studio 2017 ou Visual Studio pour Mac.
3. Dans Visual Studio, effectuez un clic droit sur le fichier projet, puis sélectionnez Publier.
4. Sélectionnez App Service, puis Existant, puis cliquez sur Publier.
5. Identifiez votre application mobile, puis cliquez sur OK pour commencer la publication. Très bien ! Le back-end mobile est opérationnel.

Quittez l'Étape 3 pour **créer une nouvelle application**, puis cliquez sur **Télécharger**.

Décompressez le fichier que vous venez de télécharger, puis ouvrez-le dans Visual Studio 2017 ou Visual Studio pour Mac.

Exécutez l'application en appuyant sur F5. L'émulateur Android est alors démarré et déploie l'application cliente. L'application cliente est une application todo. Vous pouvez ajouter un élément dans la zone de texte, qui sera ensuite écrit dans le back-end de votre application mobile.

Ajout d'une authentification à l'application

Maintenant que vous disposez d'une application mobile avec un back-end, assurez-vous que celui-ci est sécurisé. Vous allez maintenant ajouter une authentification via Azure Active Directory. Pour exécuter cette procédure, vous devez d'abord enregistrer l'application mobile à des fins d'authentification auprès d'Azur Active Directory. ([Ce tutoriel](#) vous explique comment procéder.)

Sécurisation de l'application

Dans les API de back-end, vous devez définir les applications à sécuriser.

1. Accédez à votre projet de back-end d'application mobile dans Visual Studio 2017 ou Visual Studio pour Mac.
2. Accédez à Contrôleurs > TodoItemController.cs.
3. Ajoutez l'attribut [Authorize] à la classe TodoItemController.
Cela restreint toute utilisation non authentifiée de ce contrôleur.
4. Cliquez avec le bouton droit de la souris sur le fichier de projet, puis publiez-le sur votre application mobile.

Ajout d'une authentification à l'application mobile

Le back-end est sécurisé. Vous devez maintenant en informer l'application mobile pour que celle-ci puisse se connecter. Vous devez demander aux utilisateurs de s'authentifier avant de pouvoir utiliser l'application.

1. Dans Visual Studio, accédez au projet d'application mobile Xamarin.
2. Ajoutez le code suivant à la classe TodoActivity :

```
// Define a authenticated user.
private MobileServiceUser user;
private async Task<bool> Authenticate()
{
    var success = false;
    try
    {
        // Sign in with AAD login using a server-managed flow.
        user = await client.LoginAsync(this,
            MobileServiceAuthenticationProvider.WindowsAzureActiveDirectory);
        CreateAndShowDialog(string.Format("you are now logged in - {0}",
            user.UserId), "Logged in!");

        success = true;
    }
    catch (Exception ex)
    {
        CreateAndShowDialog(ex, "Authentication failed");
    }
    return success;
}

[Java.Interop.Export()]
public async void LoginUser(View view)
{
    // Load data only after authentication succeeds.
    if (await Authenticate())
    {
        //Hide the button after authentication succeeds.
        FindViewById<Button>(Resource.Id.buttonLoginUser).Visibility = ViewStates.Gone;

        // Load the data.
        OnRefreshItemsSelected();
    }
}
```

Une nouvelle méthode permettant d'authentifier un utilisateur et un gestionnaire de méthodes pour un nouveau bouton Se connecter est ainsi créée.

3. Dans la méthode OnCreate, supprimez ou commentez la ligne de code suivante :
`OnRefreshItemsSelected();`
4. Dans le fichier Activity_To_Do.xml, ajoutez la définition de bouton LoginUser suivante avant le bouton AddItem existant :

```
<Button
android:id="@+id/buttonLoginUser"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="LoginUser"
android:text="@string/login_button_text" />
```

5. Ajoutez l'élément suivant au fichier de ressources Strings.xml :

```
<string name="login_button_text">Se connecter</string>
```

6. Exécutez l'application.

Un bouton Se connecter apparaît à présent en regard du bouton Ajouter. Vous ne pourrez pas ajouter d'éléments sans vous connecter, car toutes les actions dans `ToDoItemController` sont maintenant sécurisées. Lorsque vous cliquez sur Se connecter, vous êtes redirigé vers l'écran d'authentification Azure Active Directory. Ici, servez-vous du compte que vous utilisez pour vous connecter au portail Azure. Celui-ci est ajouté à Azure Active Directory par défaut. Connectez-vous. Un message vous indiquant que vous êtes bien connecté s'affiche, comme illustré dans la figure 6-16, et vous pouvez commencer à ajouter des éléments todo.

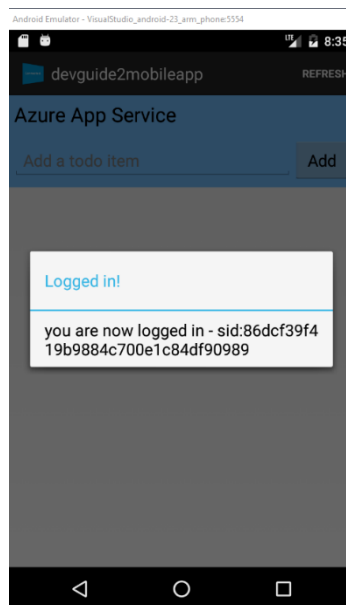


Figure 6-16 : Utilisateur venant d'être authentifié dans l'émulateur de l'application mobile

Ajout de la synchronisation hors connexion à l'application

Dans le cas d'une utilisation mobile, il est possible que les utilisateurs n'aient plus accès à Internet. Dans ces situations, ils doivent cependant toujours être en mesure de travailler. Pour y répondre, ils peuvent utiliser la fonctionnalité de synchronisation hors connexion de l'application mobile Azure. Nous allons l'activer pour notre application :

1. Dans Visual Studio 2017 ou Visual Studio pour Mac, accédez à l'application cliente Xamarin Android.
2. Dans la classe `ToDoActivity`, supprimez la marque de commentaire pour la ligne suivante :

```
#define OFFLINE_SYNC_ENABLED
```
3. C'est ça ! Exécutez l'application pour la déployer dans l'émulateur Android.

L'application fonctionne désormais exactement comme avant, à quelques exceptions près. Elle utilise un objet `IMobileServiceSyncTable` à la place d'un objet `IMobileServiceTable`. Elle utilise également le package `SQLite NuGet`, qui lance sur l'appareil Android une base de données `SQLite`, que l'application utilise pour enregistrer les données en cas d'absence de connexion. Une fois l'accès Internet rétabli, les modifications de la base de données `SQLite` locale sont transmises à l'application mobile, et les éventuelles modifications survenues dans la base de données principale sont synchronisées avec l'application sur l'appareil Android. Nous allons essayer :

1. Exécutez l'application dans l'émulateur Android.
2. Connectez-vous comme d'habitude.
3. Ajoutez un élément.
4. Désactivez la connexion Internet. Dans l'émulateur Android, accédez aux contrôles étendus et désactivez la force du signal du réseau.
5. Ajoutez un autre élément et complétez un élément en le vérifiant. L'application continue à utiliser la base de données `SQLite` locale.

Vous pouvez vérifier que les éléments ne sont pas écrits dans la base de données principale en vous connectant à la base de données SQL Azure créée précédemment, via un outil tel que [SQL Server Management Studio](#). Vous pouvez trouver l'élément `connectionstring` de la base de données SQL Azure lorsque vous y accédez dans le portail Azure.

6. Reconnectez l'appareil Android à Internet. Dans l'émulateur Android, accédez aux contrôles étendus et définissez la force du signal du réseau au maximum.
7. Dans votre application mobile, cliquez sur le bouton d'actualisation. Les modifications locales seront désormais synchronisées avec l'application mobile Azure.

Consultez la base de données SQL Azure pour voir les modifications qui sont maintenant synchronisés avec la base de données principale.

Fonctionnalités supplémentaires et mise en production

Il est facile de configurer une application avec l'application mobile et d'ajouter des fonctionnalités avancées telles que l'authentification et la synchronisation hors ligne.

L'application mobile étant un service Azure App Service, elle offre les mêmes fonctionnalités que les autres App Services. Cela signifie que l'application mobile peut également s'adapter automatiquement, que vous pouvez disposer des mêmes fonctionnalités de pilotage et d'alerte, que vous pouvez effectuer des diagnostics supplémentaires avec des journaux de streaming, et même incorporer le débogage d'application.

Lorsque vous procédez à la mise en production de l'application, vous pouvez renforcer la sécurité à l'aide des certificats App Service. Vous pouvez également configurer un déploiement continu, tout comme nous l'avons fait pour l'application Node.js. Vous pouvez par ailleurs utiliser des emplacements de déploiement pour déployer l'application sans interruption de service.

Cette configuration n'était pas une partie de plaisir, mais les problèmes sont désormais résolus. Laissez Azure s'occuper des tâches difficiles et concentrez-vous sur ce qui compte vraiment !

Utilisation d'Azure Marketplace

Avec l'écosystème Azure, vous pouvez créer plus rapidement et efficacement des applications dynamiques. Azure Marketplace vous permet de rechercher de nombreuses solutions pour soutenir vos efforts de développement. Cela vous permet également de proposer votre produit à des utilisateurs du monde entier.

En quoi Azure Marketplace constitue-t-il un avantage pour un développeur ?

[Azure Marketplace](#) propose des configurations prêtes à l'emploi de divers produits, qu'il s'agisse de Windows Server, Octopus Deploy, de bases de données Oracle, d'instances WordPress ou encore de [serveurs Minecraft](#). Il vous suffit de cliquer sur une solution, de la configurer et de la déployer pour être prêt à l'utiliser. Azure Marketplace regorge de produits et de solutions de Microsoft et des partenaires Microsoft. Cet espace propose également bon nombre de solutions open source.

La convivialité de Marketplace constitue l'un de ses avantages. Une fois déployées, les solutions sont prêtes à l'emploi avec l'ensemble des éléments requis. Il s'agit parfois d'une machine virtuelle sur laquelle sont installés des logiciels, d'un réseau virtuel configuré avec une licence intégrée installée ou une licence personnelle. Il peut également s'agir d'un service que vous acquérez, comme un abonnement à l'API Maps de Bing pour les entreprises. Vous pouvez utiliser Marketplace pour facilement tester de nouveaux logiciels et pour exécuter des solutions de niveau entreprise. Quels que soient vos besoins, la solution qui permettra d'y répondre se trouve certainement sur Marketplace.

Solutions Azure

Apprenez des réussites d'autres entreprises qui utilisent Azure, par l'intermédiaire des [solutions Azure](#). Celles-ci vous donnent accès à la documentation produit, à des études de cas et aux meilleures pratiques en matière d'architecture pour des cas d'utilisation Azure fréquents, parmi lesquels le marketing digital, SharePoint sur Azure, des applications métiers et des applications mobiles.

Pour les développeurs, les [architectures de solution](#) constituent un avantage non négligeable des solutions Azure. Vous disposez ainsi d'architectures de référence pour les problèmes courants et d'informations expliquant pourquoi il convient d'utiliser certains produits dans des cas précis.

Résumé et prochaines étapes

Le présent guide a mis en avant les avantages offerts par Azure à vos applications. Azure vous permet de faire des choses incroyables avec vos applications, qu'il s'agisse de mettre en œuvre un système de reconnaissance faciale et vocale, de gérer vos appareils sur l'Internet des Objets dans le Cloud, d'adapter votre environnement à vos besoins et de payer à l'utilisation.

Quel que soit le langage de programmation utilisé ou la plateforme créée pour vos applications, Azure vous accompagne dans presque toutes les situations grâce aux nombreux services proposés.

Nous espérons que vous continuerez à consulter cet e-book pour vous familiariser davantage avec les nombreux services Azure et pour déterminer les services les mieux adaptés à vos besoins.

Vous n'êtes désormais plus tenu d'écrire le code compliqué des solutions : profitez d'un ensemble de solutions prêtes à l'emploi. Employez utilement votre temps et laissez Azure résoudre les problèmes.

Continuer à apprendre avec un compte Azure gratuit

Créez un [compte Azure gratuit](#) et bénéficiez des avantages suivants :

- Crédit de 170 € à utiliser sur n'importe quel produit Azure pendant 30 jours
- Accès gratuit à nos produits les plus populaires du marché pendant 12 mois, y compris le calcul, le stockage, le réseau et la base de données
- Plus de 25 produits disponibles en permanence gratuitement

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 cet e-book.



Michael Crump travaille chez Microsoft sur la plateforme Azure, et est un 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 le contacter sur Twitter [@mbcrump](https://twitter.com/mbc Crump) ou depuis son blog accessible à l'adresse <https://www.michaelcrump.net>.



Architecte et développeur logiciel indépendant, Barry Luijbregts est un passionné du Cloud. Il rédige également des contenus pour Pluralsight. Fort de ses 10 années d'expérience au sein de diverses entreprises, il est désireux de partager ses connaissances avec la communauté. Il dispose d'une connaissance étendue et approfondie de la pile Microsoft avec un attrait tout particulier pour les technologies Web et le Cloud. Barry est coopérateur d'un groupe d'utilisateurs appelé .NET Zuid (South), qui s'intéresse à la technologie et aux compétences relationnelles, et donne actuellement des cours sur les avantages offerts par le Cloud. Il vit aux Pays-Bas avec sa femme et sa petite fille, et il adore jouer avec ses deux huskies de Sibérie.

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