# Hébergement d'applications Web dans le cloud AWS

*Septembre 2017*

# Mentions légales

Ce document est fourni à titre informatif uniquement. Il présente l'offre de produits et les pratiques actuelles d'AWS à la date de publication de ce document, des informations qui sont susceptibles d'être modifiées sans préavis. Il incombe aux clients de procéder à leur propre évaluation indépendante des informations contenues dans ce document et chaque client est responsable de son utilisation des produits ou services AWS, chacun étant fourni « en l'état », sans garantie d'aucune sorte, qu'elle soit explicite ou implicite. Ce document n'offre pas de garantie, représentation, engagement contractuel, condition ou assurance de la part d'AWS, de ses sociétés apparentées, fournisseurs ou concédants de licence. Les responsabilités et obligations d'AWS vis-à-vis de ses clients sont régies par les contrats AWS. Le présent document ne fait partie d'aucun contrat et ne modifie aucun contrat entre AWS et ses clients.

# Table des matières

# Résumé

L'hébergement Web évolutif et à haute disponibilité peut être complexe et onéreux. Les architectures Web évolutives traditionnelles ont non seulement nécessité l'implémentation de solutions complexes pour assurer un grand niveau de fiabilité, mais requièrent également une prévision efficace du trafic afin de fournir un service client de grande qualité. Les périodes de grands pics et la grande variabilité du trafic conduisent à une faible utilisation de matériel coûteux. Cela génère une augmentation considérable des frais d'entretien du matériel inactif et une utilisation non rentable du capital consacré à du matériel sous-utilisé.

Amazon Web Services (AWS) fournit une infrastructure fiable, évolutive, sûre et ultra performante pour les applications Web les plus exigeantes. Cette infrastructure le coût informatique correspond aux modèles de trafic du client en temps réel.

Ce livre blanc est destiné aux gestionnaires et aux architectes de systèmes informatiques qui comptent sur le cloud pour les aider à obtenir le niveau d'évolutivité répondant à leurs besoins informatiques à la demande.

# Présentation de l'hébergement Web traditionnel

L'évolutivité de l'hébergement web est un problème d'espace bien connu. La figure 1 présente une architecture traditionnelle d'hébergement Web qui met en œuvre un modèle commun d'application Web à trois niveaux. Dans ce modèle, l'architecture est divisée en couche de présentation, couche d'application et couche de persistance. L'évolutivité s'opère par l'ajout d'hôtes à ces couches. L'architecture possède aussi des fonctions de disponibilité, de basculement et de performance intégrées. L'architecture d'hébergement Web traditionnelle est facilement portée vers le Cloud AWS avec seulement quelques modifications.

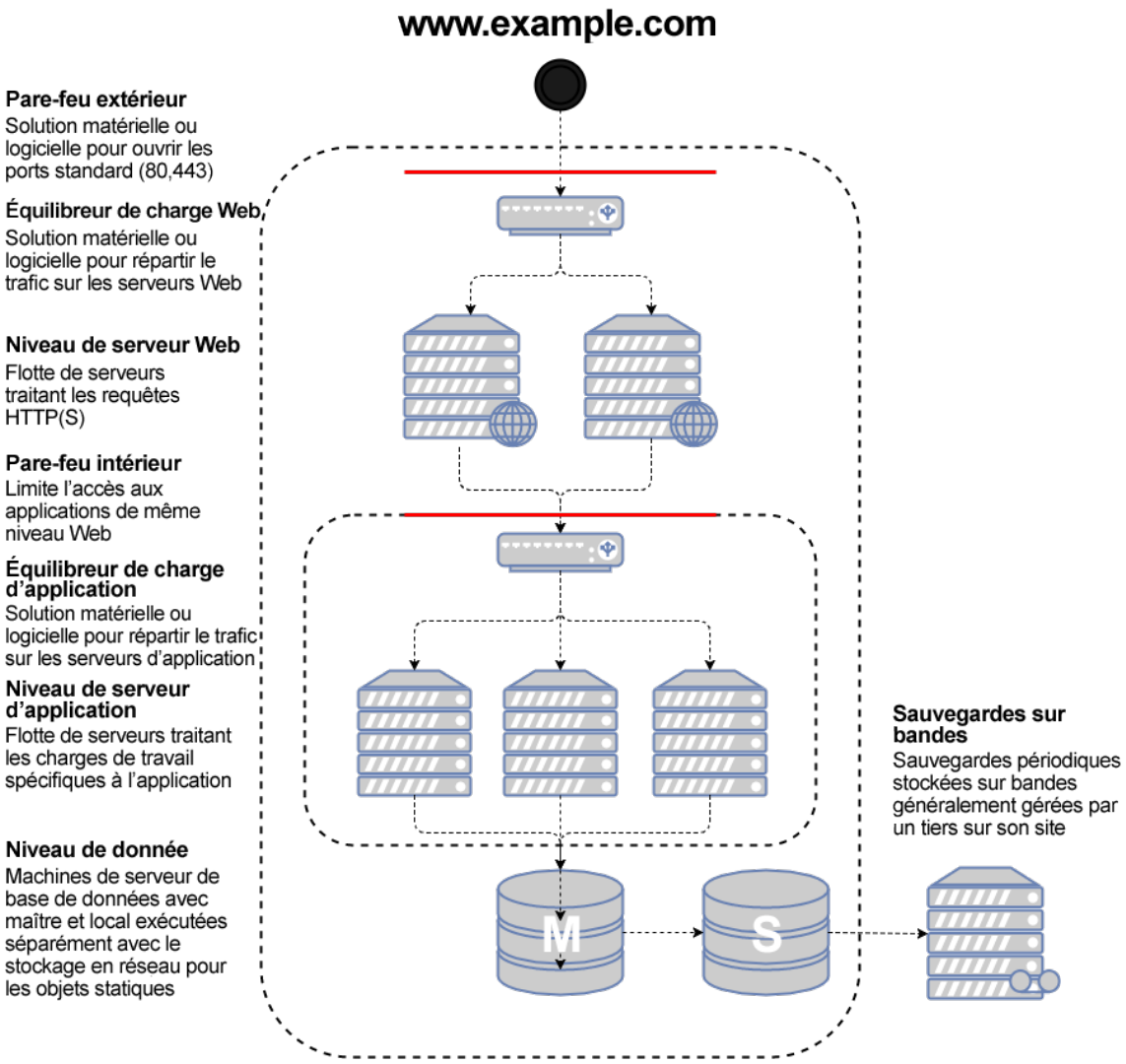


**Figure 1. Une architecture traditionnelle d'hébergement Web**

Dans les sections suivantes, nous examinons les raisons pour lesquelles une telle architecture devrait et pourrait être déployée dans le cloud AWS.

# Hébergement d'applications Web dans le cloud à l'aide d'AWS

La première question à se poser concerne la valeur du déplacement d'une solution classique d'hébergement d'applications Web vers le cloud AWS. Si vous décidez que le cloud vous convient, vous aurez besoin d'une architecture adaptée. Cette section vous aide à évaluer une solution de cloud AWS. Elle compare le déploiement de votre application Web dans le cloud à un déploiement sur site, présente une architecture de cloud AWS pour héberger votre application et décrit les principaux composants de cette solution.

## Comment AWS peut résoudre les problèmes habituels d'hébergement d'application Web

Si vous êtes responsable de l'exécution d'une application Web, vous êtes confronté à différents problèmes d'architecture et d'infrastructure pour lesquels AWS peut fournir des solutions transparentes et économiques. Voici une liste de quelques avantages parmi ceux qu'offre AWS par rapport à un modèle d'hébergement traditionnel.

### Solution économique en remplacement des flottes surdimensionnées devant gérer des pics

Dans le modèle d'hébergement traditionnel, vous devez configurer les serveurs pour gérer les pics de trafic. Les cycles non utilisés sont perdus en dehors de ces périodes de pics. Les applications Web hébergées par AWS peuvent tirer parti de la mise en service à la demande de serveurs supplémentaires, ce qui permet d'ajuster constamment la capacité et les coûts aux modèles de trafic réels.

Le graphique ci-après, par exemple, illustre une application Web avec un pic d'utilisation entre 9 heures et 15 heures et une utilisation minimale durant le reste de la journée. Une approche de dimensionnement automatique basée sur des tendances de trafic réelles qui ne met des ressources en service que si elles sont nécessaires, réduit le gaspillage de capacité et permet une réduction du coût de plus de 50 %.

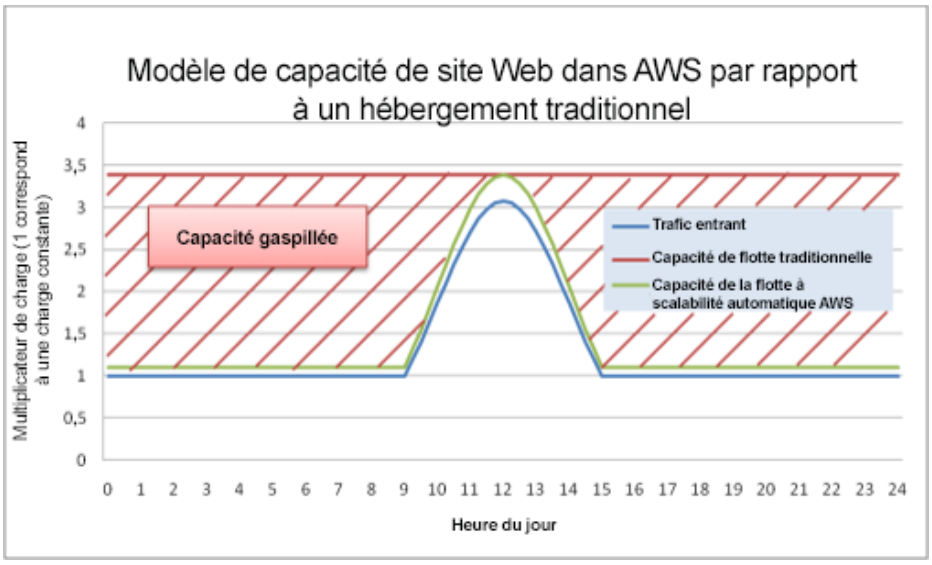


**Figure 2. Exemple de gaspillage de capacité dans un modèle d'hébergement classique**

## Une solution scalable pour gérer les pics de trafic inattendus

Une des conséquences encore plus terrible que la lenteur de mise en service d'un modèle d'hébergement traditionnel est l'incapacité à répondre à temps aux pics de trafic inattendus. Il existe de nombreux cas de blocages d'applications Web pour cause de pic de trafic imprévu une fois que l'existence du site a été annoncée dans les médias populaires. Cette même capacité à la demande qui adapte l'évolutivité des applications Web aux pics de trafic réguliers peut aussi gérer une charge imprévue. Des nouveaux hôtes peuvent être lancés et être opérationnels en quelques minutes, puis être mis hors service dès que le trafic redevient normal.

## Une solution à la demande pour des environnements de test, de chargement, bêta et de préproduction

Les coûts de matériel liés à la création d'un environnement d'hébergement traditionnel pour une application Web de production ne se limitent pas à celui de la flotte de production. Bien souvent, vous devez créer des flottes de préproduction, bêta et de test pour garantir la qualité de l'application Web à chaque étape du cycle de développement. Même si vous avez a possibilité d'apporter diverses optimisations pour garantir la plus haute utilisation possible de ce matériel de test, ces flottes parallèles ne sont pas toujours utilisées de manière optimale : beaucoup de matériels onéreux restent inactifs pendant de longues périodes.

Dans le cloud AWS, vous pouvez mettre en service des flottes de test dès que vous en avez besoin. Vous pouvez même simuler le trafic utilisateur sur le cloud AWS pendant le test de charge. Vous pouvez également utiliser ces flottes en tant qu'environnement intermédiaire pour une nouvelle version de produit. Cela permet un basculement rapide de la version de production actuelle vers une nouvelle version de l'application avec peu ou pas d'indisponibilité de service.

## L'architecture du Cloud AWS pour l'hébergement Web

La figure suivante fournit une autre vue d'une architecture classique d'application Web et illustre la façon dont elle peut tirer profit de l'infrastructure de cloud computing AWS.

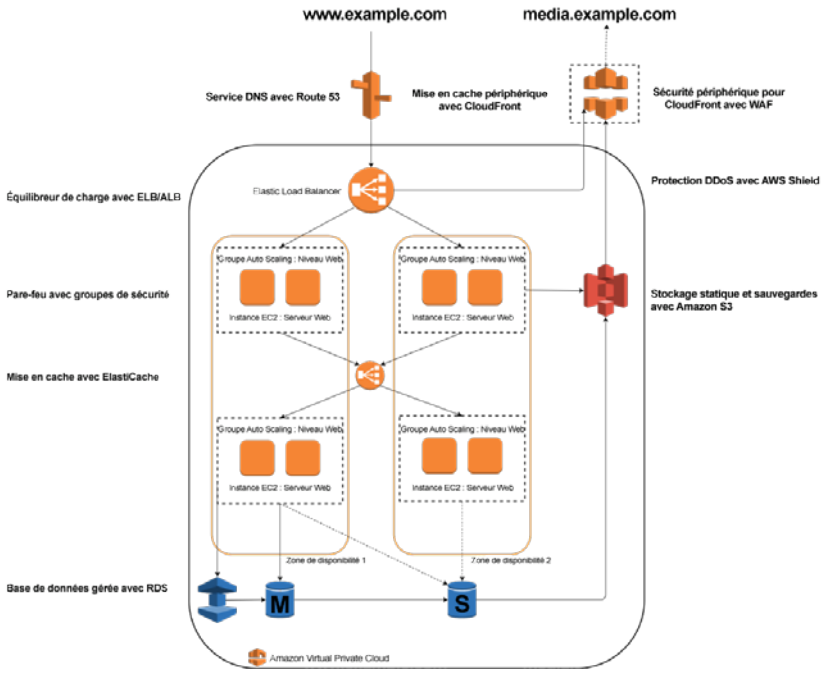


**Figure 3. Voici un exemple d'architecture d'hébergement Web sur AWS**

1. **Équilibrage de charge avec Elastic Load Balancing (ELB)/Application Load Balancer (ALB)** - Permet de répartir la charge sur plusieurs zones de disponibilité et groupes Auto Scaling Amazon EC2 pour la redondance et le découplage des services.

2. **Pare-feux avec groupes de sécurité** - Déplace la sécurité vers l'instance afin de fournir un pare-feu dynamique, au niveau de l'hôte pour les serveurs web et d'application.
3. **Mise en cache avec Amazon ElastiCache** - Fournit des services de mise en cache avec Memcached ou Redis pour retirer la charge de l'application et de la base de données et pour réduire la latence des demandes fréquentes.
4. **Base de données gérée avec Amazon RDS** - Crée une haute disponibilité, l'architecture de base de données multi-AZ avec six moteurs de base de données possibles.
5. **Services DNS d'Amazon Route 53** - Fournit des services DNS pour simplifier la gestion de domaine.
6. **Mise en cache périphérique avec Amazon CloudFront** - Edge met en cache le contenu à haut volume pour réduire la latence pour les clients.
7. **Edge Security pour Amazon CloudFront avec AWS WAF** - Filtre le trafic malveillant, y compris le trafic XSS et l'injection de code SQL par le biais des règles définies par le client.
8. **Protection DDoS avec AWS Shield** - Sauvegarde automatiquement votre infrastructure contre des attaques DDoS de la couche réseau et de transport les plus courantes.
9. **Stockage et sauvegardes statiques avec Amazon S3** - Active le simple stockage d'objets HTTP pour les sauvegardes et les ressources statiques telles que les images et les vidéos.

## Les composants clés d'une architecture d'hébergement Web AWS

Les sections suivantes décrivent certains des principaux composants d'une architecture d'hébergement Web déployée dans le cloud AWS et décrivent de quelle manière ils se distinguent d'une architecture d'hébergement Web traditionnelle.

## Gestion de réseau

Dans un environnement de cloud tel qu'AWS, la possibilité de segmenter votre réseau de celui des autres clients permet une architecture plus sécurisée et évolutive. Si les groupes de sécurité fournissent la sécurité au niveau de l'hôte (consultez la section Sécurité de l'hôte), Amazon Virtual Private Cloud (Amazon VPC) vous permet de lancer des ressources dans un réseau virtuel isolé sur le plan logique que vous avez défini.[1]

Amazon VPC est un service gratuit qui vous permet de contrôler intégralement les détails de votre configuration de mise en réseau dans AWS. Parmi les exemples de ce contrôle figure la création de sous-réseaux destinés au public pour des serveurs web et des sous-réseaux privés sans accès Internet pour vos bases de données. En outre, Amazon VPC vous permet de créer des architectures hybrides à l'aide de réseaux privés virtuels (VPN) matériels, et d'utiliser le cloud AWS comme une extension de votre propre centre de données.

Amazon VPC comprend également la prise en charge d'IPv6 en plus de la prise en charge traditionnelle d'IPv4 pour votre réseau.

## Diffusion de contenu

La mise en cache en périphérie s'avère également pertinente dans l'infrastructure de cloud computing AWS. Toutes les solutions utilisées dans votre infrastructure d'application Web fonctionnent en principe aussi parfaitement dans le cloud AWS. Toutefois, une autre option consiste à utiliser Amazon CloudFront pour la mise en cache en périphérie de votre site Web.[2]

Vous pouvez utiliser CloudFront pour diffuser votre site Web, y compris les contenus dynamiques, statiques et diffusés en continu à partir d'un réseau mondial d'emplacements périphériques. CloudFront achemine directement les requêtes ciblant le contenu vers l'emplacement périphérique le plus proche, de sorte que le contenu puisse être diffusé de manière optimale. CloudFront est optimisé pour être compatible avec les autres services AWS, tels qu'Amazon Simple Storage Service[3] (Amazon S3) et Amazon Elastic Compute Cloud[4] (Amazon EC2). CloudFront fonctionne également sans problème avec n'importe quel serveur d'origine qui n'est pas un serveur d'origine AWS sur lequel vous stockez les versions définitives et originales de vos fichiers.

A l'instar d'autres produits AWS, aucun contrat à long terme ni engagement d'utilisation mensuelle n'est requis pour utiliser Amazon CloudFront – vous ne payez que la quantité de contenu que vous diffusez via le service de diffusion de contenu.

## Gestion de DNS public

Pour déplacer une application Web vers le cloud AWS, des modifications doivent être apportées au DNS afin de tirer parti des différentes zones de disponibilité proposées par AWS. Pour vous aider à gérer le routage DNS, AWS fournit Amazon Route 53,[5] un service Web DNS à haut niveau de disponibilité et scalable. Amazon Route 53 achemine automatiquement les requêtes pour votre domaine vers le serveur DNS le plus proche. En conséquence, les requêtes donnent lieu à des réponses avec les meilleures performances possibles. Amazon Route 53 résout les requêtes pour votre nom de domaine (par exemple, www.exemple.com) dans votre équilibreur de charge classique, ainsi que votre enregistrement de zone apex (exemple.com).

## Sécurité de l'hôte

Contrairement à un modèle d'hébergement Web traditionnel, le filtrage du trafic entrant n'est pas limité à la périphérie du réseau, mais il est appliqué au niveau de l'hôte. Amazon EC2 fournit une solution nommée groupes de sécurité. Un groupe de sécurité est analogue à un pare-feu réseau entrant, pour lequel vous spécifiez les protocoles, les ports et les plages d'adresses IP source qui permettent d'atteindre vos instances EC2. Vous pouvez assigner un ou plusieurs groupes de sécurité pour chaque instance EC2. Chaque groupe de sécurité achemine le trafic approprié vers chaque instance. Les groupes de sécurité peuvent être configurés de façon que seuls des sous-réseaux ou des adresses IP spécifiques puissent accéder à une instance EC2. Ils peuvent également faire référence à d'autres groupes de sécurité afin de limiter l'accès aux instances EC2 appartenant à des groupes spécifiques.

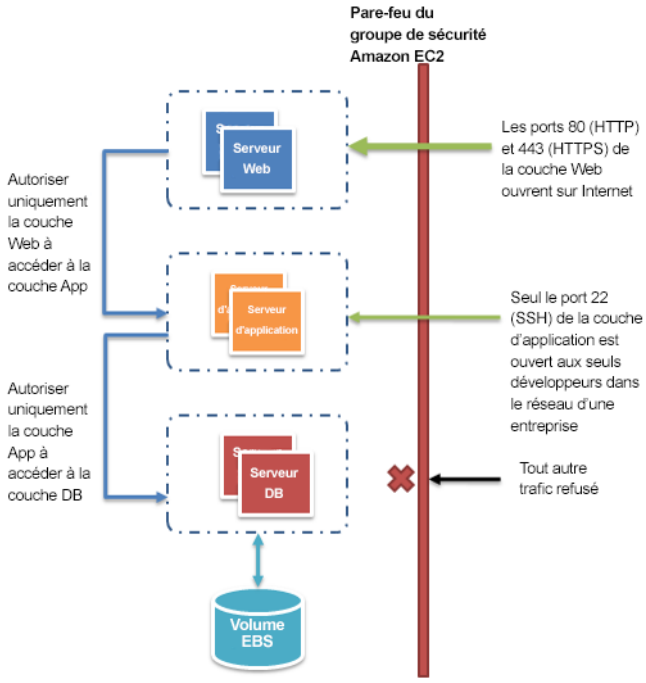


**Figure 4. Groupes de sécurité dans une application Web**

Dans l'exemple d'architecture d'hébergement web AWS de la figure 4, le groupe de sécurité pour le cluster de serveurs Web peut restreindre l'accès à tous les hôtes via TCP sur les ports 80 et 443 (HTTP et HTTPS) et depuis les instances du groupe de sécurité du serveur d'application sur le port 22 (SSH) pour une gestion directe de l'hôte. Le groupe de sécurité du serveur d'applications, d'autre part, peut autoriser l'accès depuis le groupe de sécurité du serveur Web afin de gérer les requêtes Web ainsi que depuis le sous-réseau de votre organisation via TCP sur le port 22 (SSH) pour une gestion directe de l'hôte. Dans ce modèle, vos ingénieurs de support pourraient se connecter directement aux serveurs d'applications à partir du réseau d'entreprise, puis accéder à d'autres clusters depuis les zones de serveur d'applications. Pour en savoir plus sur la sécurité, consultez le Centre de sécurité AWS.[6] Le centre comprend des bulletins de sécurité, des informations de certification et des livres blancs sur la sécurité qui décrivent les capacités d'AWS en matière de sécurité.

## Équilibrage de charge entre les clusters

Les équilibreurs de charge matériels sont des composants courants dans les réseaux utilisés dans les architectures d'hébergement Web traditionnelles. AWS offre cette fonctionnalité par le biais de Elastic Load Balancing (ELB)[7]. ELB est une solution de répartition des charges configurable qui prend en charge la

vérification de l'état des hôtes, la distribution du trafic vers les instances EC2 à travers plusieurs zones de disponibilité ainsi que l'ajout et le retrait dynamiques d'hôtes Amazon EC2 dans la rotation de répartition de charge. ELB peut également augmenter et réduire dynamiquement la capacité d'équilibrage de charge pour l'adapter aux demandes de trafic tout en fournissant un point d'entrée prévisible à l'aide d'un CNAME permanent. ELB prend également en charge les sessions permanentes pour répondre aux besoins de routage plus avancés. Si votre application nécessite des fonctionnalités d'équilibrage de charge plus avancées, vous pouvez exécuter un package d'équilibrage de charge logicielle (tel que Zeus, HAProxy ou NGINX Plus) sur les instances EC2. Vous pouvez ensuite attribuer des adresses Elastic IP à ces instances EC2 spécifiques pour minimiser les changements de DNS.[8]

## Recherche d'autres hôtes et services

Dans l'architecture d'hébergement Web traditionnelle, la plupart de vos hôtes possèdent des adresses IP statiques. Dans le cloud, leurs adresses IP sont généralement dynamiques. Même si chaque instance EC2 peut avoir des entrées DNS publiques et privées et qu'elle peut être adressée sur Internet, les entrées DNS et les adresses IP sont attribuées dynamiquement au lancement de l'instance. Elle ne peut pas être attribuée manuellement. Les adresses IP statiques (adresses IP Elastic selon la terminologie AWS) peuvent être attribuées à des instances en cours d'exécution après leur lancement. Vous devez utiliser des adresses IP Elastic pour les instances et les services qui nécessitent des points de terminaison fiables, comme les bases de données principales, les serveurs de fichiers centraux et les équilibreurs de charge hébergés sur EC2.

Les rôles de serveur susceptibles d'évoluer facilement, tels que les serveurs Web, doivent pouvoir être détectés au niveau de leurs points de terminaison dynamiques en enregistrant leur adresse IP dans un référentiel central. Comme la plupart des architectures d'applications Web possèdent un serveur de base de données activé en permanence, ce dernier constitue un référentiel commun pour la découverte d'informations. Dans les cas où un adressage cohérent est nécessaire, les instances peuvent correspondre à des adresses IP Elastic allouées depuis un pool d'adresses par un script d'action d'amorçage à des instances au moment de leur lancement.

En utilisant ce modèle, les hôtes récemment ajoutés peuvent demander la liste des points de terminaison requis pour assurer la communication à partir de la base de

données pendant la phase d'action d'amorçage. L'emplacement de la base de données peut être fourni en tant que données[9] utilisateur transmises à chaque instance au moment de leur lancement. Vous pouvez utiliser Amazon SimpleDB pour stocker et conserver les informations de configuration.[10] SimpleDB est un service à haute disponibilité accessible sur un point de terminaison bien connu.

## Mise en cache dans l'application Web

Les caches d'application en mémoire peuvent réduire la charge des services et améliorer les performances ainsi que la scalabilité dans la couche base de données en mettant en cache des informations fréquemment utilisées. Amazon ElastiCache[11] est un service Web qui facilite le déploiement, l'utilisation et la mise à l'échelle d'un cache en mémoire dans le cloud. Vous pouvez configurer le cache en mémoire que vous créez de telle sorte qu'il évolue automatiquement en fonction de la charge et qu'il remplace automatiquement les nœuds défaillants. ElastiCache est compatible avec les protocoles Memcached et Redis, ce qui simplifie la migration à partir de votre solution sur site actuelle.

## Configuration de base de données, sauvegarde et basculement

La plupart des applications Web contiennent des formes de persistance, généralement sous la forme d'une base de données relationnelle ou d'une base de données NoSQL. AWS offre une infrastructure de base de données à la fois relationnelle et NoSQL. Vous pouvez également déployer votre propre logiciel de base de données sur une instance EC2. Le tableau suivant résume ces options, et nous les évoquerons de façon plus détaillée dans cette section.

| | Solutions de base de données relationnelle | Solutions NoSQL |
|---|---|---|
| Service de base de données gérée | Amazon RDS, MySQL, Oracle, SQL Server, MariaDB, PostgreSQL, Amazon Aurora | Amazon DynamoDB |
| Auto-gestion | Hébergement d'un système de gestion de base de données relationnelle sur une instance EC2 | Hébergement d'une solution NoSQL sur une instance EC2 |

### *Amazon RDS*

Amazon Relational Database Service (Amazon RDS) vous donne accès aux capacités d'un moteur de base de données classique MySQL, PostgreSQL, Oracle et Microsoft SQL Server.[12] Le code, les applications et les outils que vous utilisez déjà peuvent être utilisés avec Amazon RDS. Amazon RDS applique

automatiquement les correctifs logiciels à la base de données, sauvegarde votre base de données et stocke les sauvegardes pendant une période de rétention définie par l'utilisateur. Il prend également en charge la restauration à un instant dans le passé. Vous bénéficiez d'une grande flexibilité vous permettant de mettre à l'échelle les ressources de calcul ou les capacités de stockage associées à votre instance de base de données relationnelle par un simple appel API.

En outre, les déploiements multi-AZ Amazon RDS améliorent la disponibilité de votre base de données et la protègent des interruptions imprévues. Les réplicas en lecture d'Amazon RDS fournissent des réplicas en lecture seule de votre base de données qui facilitent le dimensionnement au-delà de la capacité inhérente au déploiement d'une seule base de données, dans le cas de charges de travail impliquant une demande de lecture intensive. Comme pour tous les services AWS, aucun investissement initial n'est requis et vous ne payez que les ressources que vous utilisez.

### *Hébergement d'un système de gestion de base de données relationnelle (SGBDR) sur une instance Amazon EC2*

En plus de l'offre Amazon RDS gérée, vous pouvez installer votre choix de SGBDR (tel que MySQL, Oracle, SQL Server ou DB2) sur une instance EC2 et le gérer vous-même. Les clients AWS qui hébergent une base de données sur Amazon EC2 utilisent avec succès une variété de modèles maître/esclave et de réplication, avec mise en miroir de copies en lecture seule et expédition de journaux pour des esclaves passifs toujours prêts.

Lorsque vous gérez vous-même votre logiciel de base de données directement sur Amazon EC2, vous devez également prendre en compte la disponibilité du stockage tolérant aux pannes et du stockage permanent. C'est pourquoi nous recommandons que les bases de données s'exécutant sur Amazon EC2 utilisent des volumes Amazon Elastic Block Store (Amazon EBS),[13] qui sont semblables à un stockage NAS (Network Attached Storage). Pour les instances EC2 exécutant une base de données, vous devez placer toutes les données et les journaux de la base de données sur les volumes EBS. Ils resteront disponibles même en cas de défaillance de l'hôte de la base de données. Cette configuration permet un scénario de basculement simple, qui lance une nouvelle instance EC2 en cas de défaillance de l'hôte et les volumes EBS existants peuvent être attachés à la nouvelle instance. La base de données peut ensuite être récupérée là où elle a été arrêtée.

Les volumes EBS fournissent automatiquement une redondance dans la zone de disponibilité, ce qui augmente leur disponibilité par rapport aux disques simples. Si les performances d'un seul volume EBS ne suffisent pas aux besoins de votre base de données, des volumes peuvent être agrégés par bandes afin d'accroître les performances d'IOPS pour votre base de données. Dans le cas des charges de travail exigeantes, vous pouvez aussi utiliser des IOPS provisionnées EBS, où vous spécifiez l'IOPS requise. Si vous utilisez Amazon RDS, le service gère son propre stockage et vous pouvez vous concentrer sur la gestion de vos données.

### *Solutions NoSQL*

Outre la prise en charge de bases de données relationnelles, AWS propose aussi Amazon DynamoDB,[14] un service de base de données NoSQL entièrement géré offrant des performances élevées et prévisibles, avec une évolutivité aisée. A l'aide d'AWS Management Console ou de l'API DynamoDB, vous pouvez augmenter ou diminuer les capacités, en évitant les temps d'arrêt ou la dégradation des performances. Comme Amazon DynamoDB gère les charges administratives liées au fonctionnement et à la montée en charge des bases de données distribuées vers AWS, vous n'avez pas à vous soucier de l'approvisionnement en matériel, l'installation et la configuration, la réplication, les correctifs logiciels ou encore le dimensionnement des clusters.

Amazon SimpleDB fournit un service de base de données non relationnelle léger, tolérant aux pannes et à haute disponibilité avec des possibilités d'interrogation et d'indexation de données sans nécessiter de schéma fixe. SimpleDB peut s'avérer un substitut très efficace pour les bases de données dans les scénarios d'accès de données qui nécessitent un grand schéma très indexé et flexible.

De plus, vous pouvez utiliser Amazon EC2 pour héberger de nombreuses autres technologies émergentes du mouvement NoSQL, comme Cassandra, CouchDB et MongoDB.

## Stockage et sauvegarde des données et des ressources

Le cloud AWS fournit de nombreuses possibilités en matière de stockage, d'accès et de sauvegarde de vos données et ressources d'applications et Web. Amazon S3 fournit un magasin d'objets à redondance et hautement disponible. Amazon S3 est une solution de stockage idéale pour des objets relativement statiques ou à évolution lente, tels que les images, les vidéos et autres médias

statiques. Amazon S3 prend aussi en charge la mise en cache en périphérie et le streaming de ces ressources via des interactions avec CloudFront.

Pour le stockage semblable à un système de fichiers attaché, les instances EC2 peut avoir des volumes EBS attachés. Ils agissent comme des disques à monter pour exécuter des instances EC2. Amazon EBS est idéal pour des données qui doivent être accessibles sous forme de stockage par bloc et qui doivent perdurer au-delà de l'exécution en cours de l'instance, comme les partitions de base de données et les journaux d'application.

Outre la durée de vie indépendante de l'instance EC2, vous pouvez créer des instantanés des volumes EBS et les stocker dans Amazon S3. Comme les instantanés EBS ne sauvegardent que les modifications ultérieures à l'instantané précédent, des instantanés plus fréquents peuvent en réduire leurs durées. Vous pouvez également utiliser un instantané EBS comme base de référence pour répliquer des données entre plusieurs volumes EBS et attacher ces volumes à d'autres instances en cours d'exécution.

Les volumes EBS peuvent atteindre 16 To, et plusieurs EBS peuvent être agrégés par bandes pour des volumes encore plus grands ou pour accroître les performances d'E/S. Pour augmenter les performances de vos applications gourmandes en E/S, vous pouvez utiliser des volumes d'IOPS provisionnée. Les volumes IOPS dimensionnés sont conçus pour satisfaire les besoins des charges de travail gourmandes en E/S, notamment les charges de travail de base de données qui sont sensibles aux performances de stockage et à l'homogénéité du débit d'E/S. Vous spécifiez la vitesse des opérations d'E/S par seconde lorsque vous créez le volume et Amazon EBS alloue ce débit pour la durée de vie du volume. Amazon EBS prend actuellement en charge jusqu'à 20 000 opérations d'E/S par seconde et par volume. Vous pouvez agréger plusieurs volumes entre eux pour fournir des milliers d'IOPS par instance à votre application.

## Dimensionnement automatique de la flotte

Une des principales différences entre l'architecture du cloud AWS et le modèle d'hébergement traditionnel réside dans le fait qu'AWS peut dimensionner automatiquement la flotte d'applications Web à la demande, pour gérer les variations de trafic. Dans le modèle d'hébergement traditionnel, on utilise généralement les prévisions de trafic pour mettre à disposition des hôtes à l'avance. Dans AWS, des instances peuvent être mises en service à la volée en fonction d'un ensemble de déclencheurs pour augmenter ou diminuer le

dimensionnement de la flotte. Le service Auto Scaling peut créer des groupes de serveurs dont la capacité augmente ou diminue à la demande.[15] Auto Scaling fonctionne aussi directement avec Amazon CloudWatch pour les indicateurs et avec le service Elastic Load Balancing pour l'ajout ou la suppression d'hôtes afin de distribuer la charge. Par exemple, si les serveurs Web indiquent une utilisation du CPU de plus de 80 % sur une période donnée, un serveur Web supplémentaire peut rapidement être déployé, puis être ajouté automatiquement à l'équilibreur de charge pour une inclusion immédiate dans la rotation de répartition de charge.

Comme illustré dans le modèle d'architecture d'hébergement Web AWS, vous pouvez créer plusieurs groupes Auto Scaling pour différentes couches de l'architecture afin de permettre à chaque couche d'évoluer indépendamment. Par exemple, le groupe Auto Scaling du serveur Web peut déclencher l'évolution, et augmenter la taille des instances en réponse à des modifications au niveau des E/S du réseau, tandis que le groupe Auto Scaling du serveur d'applications peut évoluer en fonction de l'utilisation de l'UC. Vous pouvez définir des minima et des maxima pour assurer une disponibilité permanente (24 heures sur 24, 7 jours sur 7) et pour limiter l'utilisation au sein d'un groupe.

Les déclencheurs Auto Scaling peuvent être définis pour accroître et réduire la flotte totale d'une couche donnée afin d'adapter l'utilisation des ressources à la demande réelle. En plus du service Auto Scaling, vous pouvez dimensionner les flottes Amazon EC2 directement via l'API EC2, ce qui permet d'exécuter, d'arrêter et d'inspecter des instances.

## Fonctionnalités de sécurité supplémentaires

Le nombre et la complexité des attaques par déni de service distribué (DDoS) augmentent. Généralement, ces attaques sont difficiles à repousser. Ils sont souvent coûteux en temps d'atténuation et en alimentation, sans compter le coût que représentent les consultations de votre site Web potentielles qui n'ont pas eu lieu au cours de l'attaque. Il existe un certain nombre de facteurs et de services AWS qui peuvent vous aider à vous défendre contre ce type d'attaques. Le premier est la taille du réseau AWS. L'infrastructure AWS est très volumineuse, et nous vous permettons de tirer parti de notre échelle pour optimiser votre défense. Plusieurs services tels qu'Elastic Load Balancing, Amazon CloudFront et Amazon Route 53 sont efficaces pour mettre à l'échelle votre application Web en réponse à une augmentation importante de trafic.

Deux services en particulier pour vous aider dans votre stratégie de défense. AWS Shield est un service de protection DDoS géré qui vous aide à vous protéger contre les différentes formes d'attaque DDoS.[16] L'offre standard de AWS Shield est gratuite et automatiquement active dans votre compte. Cette offre standard contribue à défendre contre les attaques de couche réseau et de transport les plus courantes. En plus de ce niveau, l'offre avancée accorde des niveaux plus élevés de protection contre votre application Web en vous fournissant une visibilité quasiment en temps réel d'une attaque en cours, ainsi que l'intégration à des niveaux plus élevés avec les services mentionnés précédemment. De plus, vous obtenez l'accès à l'équipe de réponse DDoS (équipe DRT) d'AWS pour aider à atténuer les attaques sophistiquées et à grande échelle sur vos ressources.

AWS WAF (pare-feu d'applications Web) est conçu pour protéger vos applications Web contre les attaques qui peuvent compromettre la disponibilité ou la sécurité ou consommer les ressources en excès.[17] AWS WAF fonctionne en ligne avec CloudFront ou un équilibreur de charge d'application, conjointement avec vos règles personnalisées, pour vous défendre contre les attaques telles que les scripts inter-site, l'injection de code SQL et DDoS. Comme pour la plupart des services AWS, AWS WAF est fourni avec une API riche en fonctionnalités qui peut vous aider à automatiser la création et la modification de règles pour votre WAF à mesure que vos besoins évoluent.

## Basculement avec AWS

Un autre avantage essentiel d'AWS par rapport à un hébergement Web traditionnel réside dans les zones de disponibilité qui facilitent l'accès à des emplacements de déploiement redondants. Les zones de disponibilité correspondent à des emplacements distincts physiquement, conçues pour être isolées des défaillances dans d'autres zones de disponibilité. Elles fournissent une connectivité réseau économique et à faible latence à d'autres zones de disponibilité de la même région AWS. Comme l'illustre le schéma de l'architecture d'hébergement Web d'AWS de la figure 3, nous vous recommandons de déployer des hôtes EC2 sur plusieurs zones de disponibilité afin de rendre votre application Web plus tolérante aux pannes. Il est important de veiller à ce que la migration des points d'accès uniques sur différentes zones de disponibilité soit possible en cas de défaillance. Par exemple, vous devez configurer une base de données esclave dans une seconde zone de disponibilité afin de préserver la persistance et la haute disponibilité des données, même

dans l'éventualité peu probable où une défaillance surviendrait. Vous pouvez le faire sur Amazon EC2 ou Amazon RDS en un clic.

Même si certaines modifications architecturales sont nécessaires pour déplacer une application Web existante vers le cloud AWS, les améliorations significatives en matière d'évolutivité, de fiabilité et de rentabilité que génère l'utilisation d'AWS en valent bien la peine. Dans la section suivante, nous aborderons ces améliorations.

# Éléments importants à prendre en compte lors de l'utilisation d'AWS pour l'hébergement Web

Il existe plusieurs différences essentielles entre le cloud AWS et le modèle d'hébergement d'applications Web traditionnel. La section précédente a mis en évidence la plupart des éléments clés à prendre en compte pour déployer une application Web dans le cloud. La présente section décrit certaines modifications architecturales essentielles à appliquer pour déplacer une application quelconque dans le cloud.

## Plus d'appliances de réseau physique

Vous ne pouvez pas déployer des composants de réseau physiques dans AWS. Par exemple, les pare-feux, routeurs et équilibreurs de charge pour vos applications AWS ne peuvent plus résider sur des dispositifs physiques, mais doivent être remplacés par des solutions logicielles. Il existe une grande variété de solutions logicielles, de qualité professionnelle, qu'il s'agisse de répartir la charge (e.g., Zeus, HAProxy, NGINX Plus et Pound) ou d'établir une connexion VPN (par exemple, OpenVPN, OpenSwan et Vyatta). Ce n'est pas une limitation à ce qui peut être exécuté sur le cloud AWS, mais constitue un changement architectural de votre application si vous utilisez actuellement de tels périphériques.

## Pare-feux Everywhere

Là où vous n'aviez auparavant qu'une simple DMZ avant d'ouvrir des communications entre vos hôtes dans un modèle d'hébergement traditionnel, AWS applique un modèle plus sûr dans lequel chaque hôte est verrouillé. Une des étapes de la planification d'un déploiement AWS est l'analyse du trafic entre les hôtes. Cette analyse oriente les décisions à prendre au sujet de ce que les

ports doivent précisément ouvrir. Vous pouvez créer des groupes de sécurité au sein d'Amazon EC2 pour chaque type d'hôte dans votre architecture. En outre, vous pouvez créer une large variété de modèles de sécurité simple et à plusieurs niveaux pour garantir un accès minimum entre les hôtes dans votre architecture. L'utilisation de listes de contrôle d'accès réseau au sein d'Amazon VPC peut vous aider à verrouiller votre réseau au niveau du sous-réseau.

## Considérez la disponibilité de plusieurs centres de données

Pensez aux zones de disponibilité au sein d'une région AWS comme à plusieurs centres de données. Les instances EC2 des différentes zones de disponibilité sont séparées au niveau tant logique que physique et fournissent un modèle facile à utiliser pour déployer votre application dans des centres de données tout en bénéficiant d'un haut niveau de disponibilité et de fiabilité. Amazon VPC en tant que service régional vous permet d'exploiter les zones de disponibilité, tout en conservant l'ensemble de vos ressources dans la même logique.

## Traitez les hôtes comme des éléments éphémères et dynamiques

Le plus gros changement dans la conception architecturale de votre application AWS réside probablement dans le fait que les hôtes Amazon EC2 doivent être considérés comme éphémères et dynamiques. Les applications destinées au cloud AWS ne doivent pas partir du principe qu'un hôte sera disponible en permanence et doivent être conçues en tenant compte du fait que toutes les données qui ne figurent pas dans un volume EBS seront perdues en cas de défaillance d'une instance EC2. De plus, lorsqu'un nouvel hôte est mis à disposition, vous ne devez pas faire d'hypothèses au sujet de son adresse IP ou de son emplacement au sein d'une zone de disponibilité. Votre modèle de configuration doit être souple et votre approche à propos de l'action d'amorçage d'un hôte doit tenir compte de la nature dynamique du cloud. Ces techniques sont essentielles pour concevoir et exécuter une application à haut niveau d'évolutivité et de tolérance aux pannes.

## Prenons l'exemple d'une architecture sans serveur

Ce livre blanc se concentre principalement sur une architecture Web traditionnelle. Toutefois, les nouveaux services tels qu' AWS Lambda[18] et d' Amazon API Gateway[19] vous permettent de développer une application Web sans serveur qui supprime l'utilisation de machines virtuelles pour effectuer les calculs. Dans ce cas, le code est exécuté au cas par cas pour chaque requête, et

vous ne payez que pour le nombre de demandes et la longueur de celles-ci. Vous pouvez en savoir plus sur les architectures sans serveur ici.

# Conclusions

Beaucoup de considérations architecturales et conceptuelles entrent en ligne de compte lorsque vous envisagez de migrer votre application web vers le cloud AWS. Les avantages générés par une infrastructure rentable, hautement évolutive et tolérante aux défaillances qui croît en même temps que votre activité dépassent de loin les inconvénients d'une migration sur le cloud AWS.

# Participants

Les personnes et organisations suivantes ont participé à l'élaboration de ce document :

- Jack Hemion, architecte de solutions associé, AWS
- Matt Tavis, architecte de solutions principal, AWS
- Philip Fitzsimons, directeur principal de l'architecture, AWS

# Suggestions de lecture

- Guide de démarrage - Hébergement d'applications Web AWS pour Linux
- Guide de démarrage - Hébergement d'applications Web AWS pour Windows
- Séries vidéos de démarrages : Applications Web Linux dans le cloud AWS
- Séries vidéos de démarrages : Applications Web.NET dans le cloud AWS

# Révisions du document

| Date | Description |
|---|---|
| **Juillet 2017** | Plusieurs rubriques ajoutées et mises à jour pour les nouveaux services. Mise à jour de schémas pour plus de clarté et de services. Ajout de VPC selon la méthode standard de mise en réseau sur AWS dans « Network Management. » Ajout de la section sur la protection et l'atténuation des risques DDoS dans « Autres fonctionnalités de sécurité. » Ajout d'une petite section sur les architectures sans serveur pour l'hébergement Web. |
| **Septembre 2012** | Plusieurs rubriques mises à jour pour améliorer la clarté. Mise à jour de schémas pour l'utilisation des icônes AWS. Ajout de la section « Gestion de DNS public » |

| Date | Description |
|---|---|
| | pour les détails sur Amazon Route 53. Section « Trouver d'autres hôtes et services » mise à jour pour plus de clarté. Section « Configuration de la base de données, sauvegarde et basculement » mise à jour pour plus de clarté et DynamoDB. « Stockage et sauvegarde des données et de ressources » allongée pour couvrir les volumes d'IOPS provisionnées d'EBS. |
| **Mai 2010** | Première publication |

# Remarques

[1] https://aws.amazon.com/vpc/

[2] https://aws.amazon.com/cloudfront/

[3] https://aws.amazon.com/s3/

[4] https://aws.amazon.com/ec2/

[5] https://aws.amazon.com/route53/

[6] https://aws.amazon.com/security/

[7] https://aws.amazon.com/elasticloadbalancing/

[8] Les adresses IP élastiques sont des adresses IP statiques conçues pour le cloud computing dynamique, qui peuvent être transférées d'une instance à l'autre.

[9] http://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html

[10] https://aws.amazon.com/simpledb/

[11] https://aws.amazon.com/elasticache/

[12] https://aws.amazon.com/rds/

[13] https://aws.amazon.com/ebs/

[14] https://aws.amazon.com/dynamodb/

[15] https://aws.amazon.com/autoscaling/

[16] https://aws.amazon.com/shield/

[17] https://aws.amazon.com/waf/

[18] https://aws.amazon.com/lambda/

[19] https://aws.amazon.com/api-gateway/