



Livre blanc AWS

# Bonnes pratiques pour WordPress sur AWS



# Bonnes pratiques pour WordPress sur AWS: Livre blanc AWS

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et l'habillage commerciaux d'Amazon ne peuvent pas être utilisés en connexion avec un produit ou un service qui n'est pas celui d'Amazon, d'une manière susceptible de causer de la confusion chez les clients ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon sont la propriété de leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

Résumé .....	1
Résumé .....	1
Introduction .....	2
Déploiement simple .....	3
Considérations .....	3
Approches disponibles .....	3
Amazon Lightsail .....	4
Sélection d'un plan tarifaire Amazon Lightsail .....	5
Installation de WordPress .....	5
Reprise sur incident .....	6
Améliorer les performances et la rentabilité .....	7
Accélérer la diffusion de contenu .....	7
Déchargement de contenu statique .....	8
Contenu dynamique .....	8
Mise en cache des base de données .....	10
Mise en cache du bytecode .....	10
Déploiement Elastic .....	12
Architecture de référence .....	12
Mise à l'échelle du niveau web .....	14
Niveau web sans état .....	16
Stockage partagé (Amazon S3 et Amazon EFS) .....	16
Niveau données (Amazon Aurora et Amazon ElastiCache) .....	17
WordPress High Availability by Bitnami sur AWS Quick Start .....	19
Conclusion .....	21
Participants .....	22
Révisions du document .....	23
Annexe A : Configuration de CloudFront .....	24
Origines et comportements .....	24
Création d'une distribution CloudFront .....	24
Annexe B : Installation et configuration des plugins .....	28
Plugin AWS for WordPress .....	28
Installation et configuration des plugins .....	28
Amazon CloudFront et AWS Certificate Manager .....	30
Traduire et vocaliser votre contenu .....	30

---

Podcasting avec Amazon Pollycast .....	33
Lecture de votre contenu via des appareils Amazon Alexa .....	33
Configuration du contenu statique .....	34
Création d'un utilisateur IAM .....	35
Création d'un compartiment Amazon S3 .....	35
Création d'une origine statique .....	36
Annexe C : Sauvegarde et restauration .....	38
Annexe D : Déploiement de nouveaux plugins et thèmes .....	40
Mentions légales .....	41

# Bonnes pratiques pour WordPress sur AWS

Date de publication : 19 octobre 2021 ([Révisions du document](#))

## Résumé

Ce livre blanc fournit aux administrateurs système des conseils spécifiques sur la façon de démarrer avec WordPress sur Amazon Web Services (AWS) et d'améliorer à la fois la rentabilité du déploiement et l'expérience de l'utilisateur final. Il présente également une architecture de référence qui répond aux exigences courantes de capacité de mise à l'échelle et de haute disponibilité.

# Introduction

WordPress est un outil de blog open source et un système de gestion de contenu (CMS) basé sur PHP et MySQL, utilisé pour alimenter de nombreux contenus, des blogs personnels aux sites web à fort trafic.

Lorsque la première version de WordPress a été publiée en 2003, elle n'a pas été conçue dans l'optique d'utiliser des infrastructures modernes, élastiques et évolutives basées sur le cloud. Grâce au travail de la communauté WordPress et la parution de divers modules WordPress, les capacités de cette solution CMS sont en constante évolution. Aujourd'hui, il est possible de concevoir une architecture WordPress qui tire profit de la plupart des avantages du cloud AWS.

# Déploiement simple

Pour les blogs ou les sites web à faible trafic sans exigences strictes en matière de haute disponibilité, un déploiement simple sur un serveur unique peut convenir. Ce déploiement n'est pas l'architecture la plus résiliente ou la plus évolutive, mais c'est le moyen le plus rapide et le plus économique de mettre votre site web en service.

## Rubriques

- [Considérations](#)
- [Approches disponibles](#)
- [Amazon Lightsail](#)

## Considérations

Cette discussion aborde d'abord le déploiement d'un seul serveur web. Il peut arriver que ce déploiement ne soit plus suffisant, par exemple :

- La machine virtuelle sur laquelle votre site web WordPress est déployé est un point de défaillance unique. Un problème avec cette instance entraîne une perte de service pour votre site web.
- La mise à l'échelle des ressources pour améliorer les performances ne peut être obtenue que par une « mise à l'échelle verticale », c'est-à-dire en augmentant la taille de la machine virtuelle sur laquelle s'exécute votre site web WordPress.

## Approches disponibles

AWS propose différentes options pour l'approvisionnement de machines virtuelles. Il existe trois méthodes principales pour héberger votre propre site web WordPress sur AWS :

- Amazon Lightsail
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Marketplace

[Amazon Lightsail](#) est un service qui permet de lancer rapidement un serveur privé virtuel (une instance Lightsail) pour héberger un site web WordPress. Lightsail est le moyen le plus simple de

démarrer si vous n'avez pas besoin de types d'instances hautement configurables ou d'un accès à des fonctionnalités réseau avancées.

[Amazon EC2](#) est un service web qui fournit une capacité de calcul redimensionnable et vous permet de lancer un serveur virtuel en quelques minutes. Amazon EC2 fournit davantage d'options de configuration et de gestion que Lightsail, ce qui est souhaitable dans les architectures plus avancées. Vous disposez d'un accès administratif à vos instances EC2 et pouvez installer tous les packages logiciels de votre choix, y compris WordPress.

[AWS Marketplace](#) est une boutique en ligne où, en tant qu'abonné, vous pouvez trouver, acheter et déployer rapidement des logiciels qui s'exécutent sur AWS. Vous pouvez utiliser le déploiement 1-Click pour lancer des images WordPress préconfigurées directement sur Amazon EC2 dans votre propre compte AWS en quelques minutes seulement. Il existe un certain nombre de fournisseurs Marketplace proposant des instances WordPress prêtes à l'emploi.

Ce livre blanc présente l'option Lightsail comme l'installation recommandée pour un site WordPress à serveur unique.

## Amazon Lightsail

Lightsail est le moyen le plus facile de faire ses premiers pas avec AWS pour les développeurs, les petites entreprises, les étudiants et les autres utilisateurs ayant besoin d'une solution simple de serveur privé virtuel (VPS).

Le service évite à l'utilisateur un grand nombre des éléments les plus complexes de la gestion de l'infrastructure. Il s'agit donc d'un point de départ idéal si vous avez moins d'expérience en matière d'infrastructure ou lorsque vous devez vous concentrer sur l'exécution de votre site web et qu'un produit simplifié est suffisant pour répondre à vos besoins.

Avec Amazon Lightsail, vous pouvez choisir les systèmes d'exploitation Windows ou Linux/Unix et les applications web populaires, y compris WordPress, et les déployer en un seul clic à partir de modèles préconfigurés.

À mesure que vos besoins augmentent, vous pouvez facilement sortir des limites initiales et vous connecter à des services supplémentaires de base de données, de stockage d'objets, de mise en cache et de distribution de contenu AWS.

## Sélection d'un plan tarifaire Amazon Lightsail

Un [plan Lightsail](#) définit le coût mensuel des ressources Lightsail que vous utilisez pour héberger votre site web WordPress. Un certain nombre de plans sont disponibles pour couvrir divers cas d'utilisation, avec différents niveaux de ressources CPU, de mémoire, de stockage sur disque SSD (Solid State Drive) et de transfert de données. Si votre site web est complexe, vous aurez peut-être besoin d'une instance plus importante avec davantage de ressources. Vous pouvez y parvenir en migrant votre serveur vers un plan plus important [à l'aide de la console web](#) ou comme décrit dans la [documentation Amazon Lightsail CLI](#).

## Installation de WordPress

Lightsail fournit des modèles pour les applications couramment utilisées telles que WordPress. Ce modèle est un excellent point de départ pour gérer votre propre site web WordPress, car il est préinstallé avec la plupart des logiciels dont vous avez besoin. Vous pouvez installer des logiciels supplémentaires ou personnaliser la configuration du logiciel en utilisant le terminal intégré au navigateur ou votre propre client SSH, ou via l'interface web d'administration de WordPress.

Amazon Lightsail a conclu un partenariat avec le produit GoDaddy Pro Sites pour aider les clients de WordPress à gérer facilement et gratuitement leurs instances. Les serveurs virtuels WordPress sur Lightsail sont préconfigurés et optimisés pour offrir performances rapides et sécurité, ce qui facilite la mise en route de votre site WordPress en un rien de temps. Les clients qui exécutent plusieurs instances WordPress trouvent qu'il est difficile et fastidieux de mettre à jour et de gérer l'ensemble de leurs sites. Avec cette intégration, vous pouvez facilement gérer vos multiples instances WordPress en quelques minutes et en seulement quelques clics.

Pour plus d'informations sur la gestion de WordPress sur Lightsail, consultez la section [Mise en route avec WordPress à partir de votre instance Amazon Lightsail](#). Une fois que vous avez terminé de personnaliser votre site web WordPress, nous vous recommandons de prendre un instantané de votre instance.

Un [instantané](#) permet de créer une image de sauvegarde de votre instance Lightsail. Il s'agit d'une copie du disque système qui stocke également la configuration originale de la machine (mémoire, CPU, taille du disque et vitesse de transfert de données). Les instantanés peuvent être utilisés pour rétablir une bonne configuration après une erreur de déploiement ou de mise à niveau.

Cet instantané vous permet de récupérer votre serveur si nécessaire, mais également de lancer de nouvelles instances avec les mêmes personnalisations.

## Reprise sur incident

Un serveur web unique constitue un point de défaillance unique. Vous devez donc vous assurer que les données de votre site web sont sauvegardées. Le mécanisme d'instantané décrit précédemment peut également être utilisé à cette fin. Pour effectuer une reprise sur incident, vous pouvez restaurer une nouvelle instance à partir de votre instantané le plus récent. Pour réduire la quantité de données susceptibles d'être perdues lors d'une restauration, vos instantanés doivent être aussi récents que possible.

Pour réduire les risques de perte de données, veillez à ce que des instantanés soient pris régulièrement. Vous pouvez planifier des instantanés automatiques de vos instances Lightsail Linux/Unix. Pour savoir comment procéder, reportez-vous au document [Activation ou désactivation des instantanés automatiques pour les instances ou les disques dans Amazon Lightsail](#).

AWS vous recommande d'utiliser une adresse IP statique : une adresse IP fixe et publique dédiée à votre compte Lightsail. Si vous décidez de remplacer votre instance par une autre, vous pouvez réaffecter l'adresse IP statique à la nouvelle instance. De cette manière, vous n'avez pas besoin de reconfigurer tous les systèmes externes (tels que les enregistrements DNS) afin qu'ils pointent vers une nouvelle IP chaque fois que vous souhaitez remplacer votre instance.

# Améliorer les performances et la rentabilité

Il se peut que votre déploiement sur un serveur unique finisse par être dépassé. Dans ce cas, vous devrez peut-être envisager des options pour améliorer les performances de votre site web. Avant de migrer vers un déploiement multi-serveurs et évolutif (abordé plus loin dans ce document), vous pouvez réaliser un certain nombre d'économies en termes de performances et de coûts. Il s'agit de bonnes pratiques que vous devriez suivre dans tous les cas, même si vous passez à une architecture multi-serveurs.

Les sections suivantes présentent un certain nombre d'options qui peuvent améliorer certains aspects des performances et de la capacité de mise à l'échelle de votre site web WordPress. Certaines peuvent être appliquées à un déploiement sur un serveur unique, tandis que d'autres exploitent la capacité de mise à l'échelle de plusieurs serveurs. Bon nombre de ces modifications nécessitent l'utilisation d'un ou plusieurs plugins WordPress. Bien que différentes options soient disponibles, [W3 Total Cache](#) est un choix répandu qui combine bon nombre de ces modifications dans un seul plugin.

## Rubriques

- [Accélérer la diffusion de contenu](#)
- [Mise en cache des base de données](#)
- [Mise en cache du bytecode](#)

## Accélérer la diffusion de contenu

Tout site web WordPress doit fournir un mélange de contenu statique et de contenu dynamique. Le contenu statique inclut les images, les fichiers JavaScript ou les feuilles de style. Le contenu dynamique inclut tout ce qui est généré côté serveur à l'aide du code PHP WordPress, par exemple, les éléments de votre site qui sont générés à partir de la base de données ou personnalisés pour chaque visiteur.

Un aspect important de l'expérience de l'utilisateur final est la latence du réseau impliquée lors de la diffusion du contenu précédent aux utilisateurs du monde entier. L'accélération de la diffusion du contenu précédent améliore l'expérience de l'utilisateur final, en particulier celle des utilisateurs dispersés géographiquement dans le monde entier. Cette accélération peut être obtenue grâce à un réseau de diffusion de contenu (CDN) tel qu'Amazon CloudFront.

[Amazon CloudFront](#) est un service web qui permet de distribuer du contenu de façon simple et rentable avec une faible latence et des vitesses de transfert de données élevées par le biais de plusieurs emplacements périphériques dans le monde. Les demandes des utilisateurs sont automatiquement acheminées vers un [emplacement périphérique](#) CloudFront approprié afin de réduire la latence. Si le contenu peut être mis en cache (pendant quelques secondes, quelques minutes, voire plusieurs jours) et qu'il est déjà stocké dans un emplacement périphérique particulier, CloudFront le fournit immédiatement. Si le contenu ne doit pas être mis en cache, a expiré ou ne se trouve pas actuellement dans cet emplacement périphérique, CloudFront récupère le contenu d'une ou plusieurs sources de vérité, appelées origines (dans ce cas, l'instance Lightsail) dans la configuration CloudFront. Cette récupération s'effectue via des connexions réseau optimisées, qui permettent d'accélérer la diffusion de contenu sur votre site web. Outre l'amélioration de l'expérience de l'utilisateur final, le modèle présenté réduit également la charge sur vos serveurs d'origine et peut générer d'importantes économies de coûts.

## Déchargement de contenu statique

Le contenu statique inclut les fichiers CSS, JavaScript et image (soit les fichiers qui font partie de vos thèmes WordPress, soit les fichiers multimédias chargés par les administrateurs de contenu). Tous ces fichiers peuvent être stockés dans Amazon Simple Storage Service (Amazon S3) à l'aide d'un plugin tel que W3 Total Cache et fournis aux utilisateurs de manière évolutive et hautement disponible. [Amazon S3](#) offre une infrastructure de stockage de données hautement évolutive, fiable, à faible latence et à faible coût, accessible via des API REST. Amazon S3 stocke de manière redondante vos objets, non seulement sur plusieurs appareils, mais également sur plusieurs sites d'une région AWS, offrant ainsi des niveaux de durabilité exceptionnellement élevés.

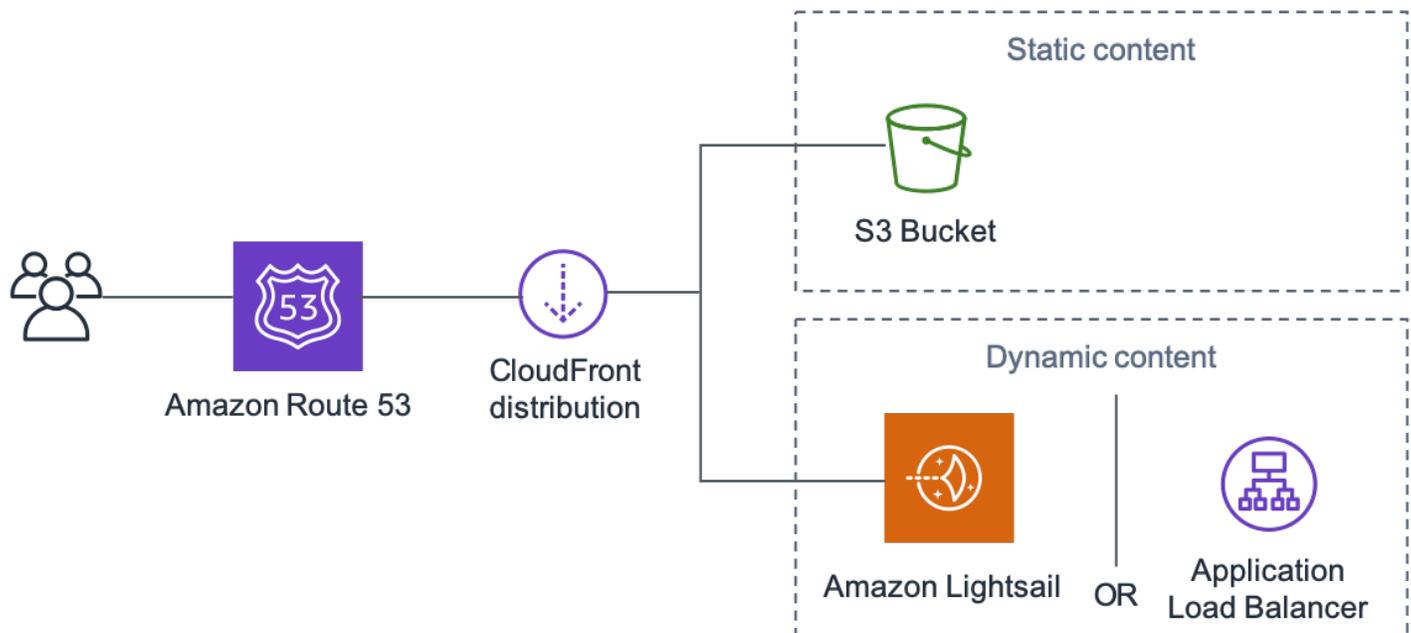
Cela a pour effet positif de délester votre instance Lightsail de cette charge de travail et de la laisser se concentrer sur la génération de contenu dynamique. Cela réduit la charge sur le serveur et constitue une étape importante vers la création d'une architecture sans état (condition préalable à la mise en œuvre de la scalabilité automatique).

Vous pouvez ensuite configurer Amazon S3 en tant qu'origine pour CloudFront afin d'améliorer la livraison de ces ressources statiques aux utilisateurs du monde entier. Bien que WordPress ne soit pas pré-intégré à Amazon S3 et CloudFront, plusieurs plugins permettent de prendre en charge ces services (par exemple, W3 Total Cache).

## Contenu dynamique

Le contenu dynamique inclut la sortie de scripts PHP WordPress côté serveur. Le contenu dynamique peut également être diffusé via CloudFront en configurant le site web WordPress en

tant qu'origine. Étant donné que le contenu dynamique inclut du contenu personnalisé, vous devez configurer CloudFront pour transférer certains cookies HTTP et en-têtes HTTP dans le cadre d'une demande envoyée à votre serveur d'origine personnalisé. CloudFront utilise les valeurs de cookie transférées dans la clé qui identifie un objet unique dans son cache. Pour optimiser l'efficacité de la mise en cache, vous devez configurer CloudFront pour ne transférer que les cookies HTTP et les en-têtes HTTP qui modifient réellement le contenu (et non les cookies qui ne sont utilisés que du côté client ou par des applications tierces, par exemple, pour l'analyse web).



### Livraison de l'ensemble du site web via Amazon CloudFront

La figure précédente inclut deux origines : l'une pour le contenu statique et l'autre pour le contenu dynamique. Pour plus d'informations sur l'implémentation, reportez-vous à l'[Annexe A : Configuration de CloudFront](#) et à l'[Annexe B : Installation et configuration des plugins](#).

CloudFront utilise des en-têtes de contrôle de cache standard pour déterminer s'il doit mettre en cache des réponses HTTP spécifiques et pendant combien de temps il doit le faire. Ces mêmes en-têtes de contrôle de cache sont également utilisés par les navigateurs web pour définir quand et pendant combien de temps le contenu doit être mis en cache localement pour une expérience utilisateur final plus optimale (par exemple, un fichier `.css` déjà téléchargé ne sera pas téléchargé de nouveau chaque fois qu'un visiteur récurrent consulte une page). Vous pouvez configurer les en-têtes de contrôle de cache au niveau du serveur web (par exemple, via des fichiers `.htaccess` ou des modifications du fichier `httpd.conf`) ou installer un plugin WordPress (par exemple, W3 Total Cache) pour dicter la façon dont ces en-têtes sont définis pour le contenu statique et le contenu dynamique.

## Mise en cache des base de données

La mise en cache des bases de données peut réduire considérablement la latence et augmenter le débit des applications à forte charge de lecture comme WordPress. Les performances des applications sont améliorées en stockant en mémoire les données fréquemment consultées pour un accès à faible latence (par exemple, les résultats de requêtes de base de données à forte intensité d'E/S). Lorsqu'un pourcentage élevé de requêtes est traité à partir du cache, le nombre de requêtes qui doivent atteindre la base de données est réduit, ce qui se traduit par une diminution des coûts associés au fonctionnement de la base de données.

Bien que WordPress dispose de capacités de mise en cache limitées à la base, divers plugins prennent en charge l'intégration avec [Memcached](#), un système de mise en cache d'objets mémoire largement répandu. Le plugin W3 Total Cache en est un bon exemple.

Dans les scénarios les plus simples, vous installez Memcached sur votre serveur web et vous capturez le résultat sous la forme d'un nouvel instantané. Dans ce cas, vous êtes responsable des tâches administratives associées à l'exécution d'un cache.

Une autre option consiste à utiliser un service géré tel qu'[Amazon ElastiCache](#) et à éviter cette charge opérationnelle. ElastiCache facilite le déploiement, l'exploitation et la mise à l'échelle d'un cache en mémoire distribué dans le cloud. Vous trouverez des informations sur la façon de vous connecter à vos nœuds de cluster ElastiCache dans la [documentation Amazon ElastiCache](#).

Si vous utilisez Lightsail et souhaitez accéder à un cluster ElastiCache de votre compte AWS en privé, vous pouvez le faire en utilisant l'appariage de VPC. Pour obtenir des instructions sur l'activation de l'appariage de VPC, consultez le document [Configurer l'appariage de VPC Amazon pour travailler avec les ressources AWS en dehors d'Amazon Lightsail](#).

## Mise en cache du bytecode

Chaque fois qu'un script PHP est exécuté, il est analysé et compilé. Si vous utilisez un cache de bytecode PHP, le résultat de la compilation PHP est stocké dans la RAM afin que le même script n'ait pas à être compilé encore et encore. Cela réduit la surcharge liée à l'exécution des scripts PHP, et permet d'obtenir de meilleures performances et une réduction des besoins en CPU.

Un cache de bytecode peut être installé sur n'importe quelle instance Lightsail hébergeant WordPress et peut réduire considérablement sa charge. Pour PHP 5.5 et versions ultérieures, AWS recommande d'utiliser [OPcache](#), une extension fournie avec cette version de PHP.

Notez qu'OPcache est activé par défaut dans le modèle WordPress Lightsail Bitnami, donc aucune autre action n'est requise.

# Déploiement Elastic

Il existe de nombreux scénarios dans lesquels un déploiement sur un seul serveur peut ne pas être suffisant pour votre site web. Dans ces situations, vous avez besoin d'une architecture évolutive multi-serveurs.

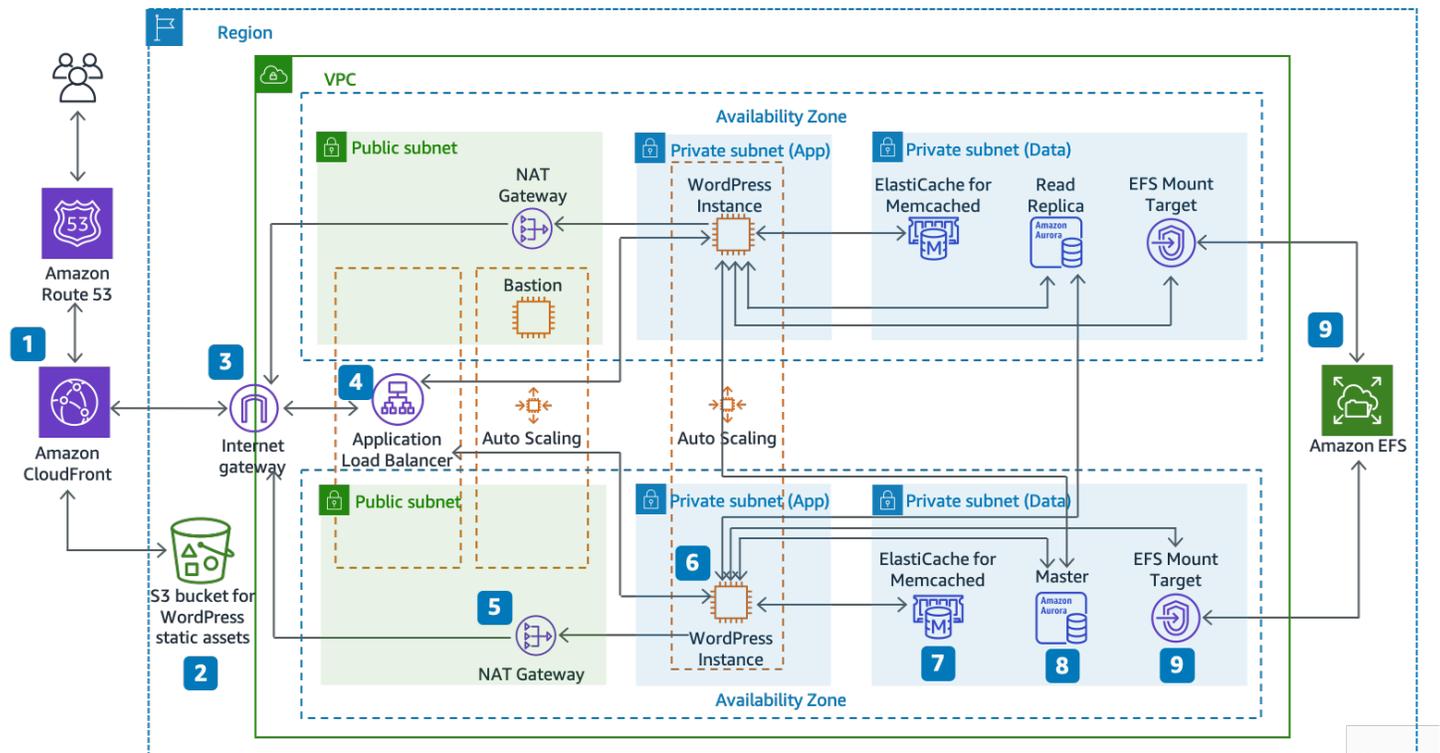
## Rubriques

- [Architecture de référence](#)
- [Mise à l'échelle du niveau web](#)
- [Niveau web sans état](#)
- [WordPress High Availability by Bitnami sur AWS Quick Start](#)

## Architecture de référence

Le document [Hosting WordPress on AWS reference architecture](#), disponible sur GitHub, décrit les bonnes pratiques de déploiement de WordPress sur AWS et comprend un ensemble de modèles AWS CloudFormation vous permettant d'être rapidement opérationnel. L'architecture suivante est basée sur cette architecture de référence. Le reste de cette section passe en revue les raisons qui sous-tendent ces choix architecturaux.

L'AMI basée dans GitHub est passée d'Amazon Linux1 à Amazon Linux2 en juillet 2021. Cependant, les modèles de déploiement sur S3 n'ont pas encore été modifiés. Il est recommandé d'utiliser des modèles sur GitHub en cas de problème de déploiement de l'architecture de référence avec des modèles sur S3.



## Architecture de référence pour l'hébergement de WordPress sur AWS

### Composants de l'architecture

L'architecture de référence illustre le déploiement complet des bonnes pratiques pour un site web WordPress sur AWS.

- Cela commence par une mise en cache en périphérie dans Amazon CloudFront (1), qui permet de mettre en cache le contenu à proximité des utilisateurs finaux pour une diffusion plus rapide.
- CloudFront extrait le contenu statique d'un compartiment S3 (2) et le contenu dynamique d'un Application Load Balancer (4) devant les instances web.
- Les instances web s'exécutent dans un groupe Auto Scaling d'instances Amazon EC2 (6).
- Un cluster ElastiCache (7) met en cache les données fréquemment demandées pour accélérer les réponses.

Une instance MySQL Amazon Aurora (8) héberge la base de données WordPress.

- Les instances EC2 de WordPress accèdent aux données WordPress partagées sur un système de fichiers Amazon EFS via une cible de montage EFS (9) dans chaque zone de disponibilité.
- Une passerelle Internet (3) permet la communication entre les ressources de votre VPC et Internet.

- Les passerelles NAT (5) de chaque zone de disponibilité permettent aux instances EC2 des sous-réseaux privés (application et données) d'accéder à Internet.

Dans le VPC Amazon, il existe deux types de sous-réseaux : public (sous-réseau public) et privé (sous-réseau d'application et sous-réseau de données). Les ressources déployées dans les sous-réseaux publics recevront une adresse IP publique et seront visibles publiquement sur Internet. L'Application Load Balancer (4) et un hôte Bastion pour l'administration sont déployés ici. Les ressources déployées dans les sous-réseaux privés reçoivent uniquement une adresse IP privée et ne sont donc pas visibles publiquement sur Internet, ce qui améliore la sécurité de ces ressources. Les instances de serveur web WordPress (6), les instances de cluster ElastiCache (7), les instances de base de données Aurora MySQL (8) et les cibles de montage EFS (9) sont toutes déployées dans des sous-réseaux privés.

Le reste de cette section explique chacune de ces considérations en détail.

## Mise à l'échelle du niveau web

Pour faire évoluer votre architecture mono-serveur vers une architecture évolutive multi-serveurs, vous devez utiliser cinq composants clés :

- Instances Amazon EC2
- Amazon Machine Images (AMI)
- Équilibreurs de charge
- Scalabilité automatique
- Surveillance de l'état

AWS fournit une grande variété de types d'instances EC2 afin que vous puissiez choisir la meilleure configuration de serveur en termes de performances et de coûts. De manière générale, le type d'instance optimisé pour le calcul (par exemple, C4) peut être un bon choix pour un serveur web WordPress. Vous pouvez déployer vos instances sur plusieurs zones de disponibilité au sein d'une région AWS afin d'augmenter la fiabilité de l'architecture globale.

Comme vous avez le contrôle total de votre instance EC2, vous pouvez vous connecter avec un accès root pour installer et configurer tous les composants logiciels nécessaires au fonctionnement d'un site web WordPress. Une fois que vous avez terminé, vous pouvez enregistrer cette

configuration en tant qu'AMI, que vous pouvez ensuite utiliser pour lancer de nouvelles instances avec toutes les personnalisations que vous avez apportées.

Pour distribuer les demandes des utilisateurs finaux à plusieurs nœuds de serveur web, vous avez besoin d'une solution de répartition de charge. AWS fournit cette fonctionnalité par le biais d'[Elastic Load Balancing](#), un service à haute disponibilité qui distribue le trafic vers plusieurs instances EC2. Étant donné que votre site web diffuse du contenu à vos utilisateurs via HTTP ou HTTPS, nous vous recommandons d'utiliser l'Application Load balancer, un équilibreur de charge de couche d'application avec un routage de contenu et la possibilité d'exécuter plusieurs sites web WordPress sur différents domaines, si nécessaire.

Elastic Load Balancing prend en charge la distribution des demandes dans plusieurs zones de disponibilité au sein d'une région AWS. Vous pouvez également configurer une surveillance de l'état afin que l'Application Load Balancer arrête automatiquement d'envoyer du trafic aux instances individuelles qui ont échoué (par exemple, en raison d'un problème matériel ou d'une panne logicielle). AWS recommande d'utiliser la page de connexion de l'administrateur WordPress (`/wp-login.php`) pour la surveillance de l'état, car cette page vérifie à la fois que le serveur web est en cours d'exécution et que le serveur web est configuré pour traiter correctement les fichiers PHP.

Vous pouvez choisir de créer une page de surveillance de l'état personnalisée qui vérifie d'autres ressources dépendantes, telles que les ressources de base de données et de cache. Pour plus d'informations, reportez-vous à la section [Vérifications de l'état de vos groupes cible](#) dans le Guide Application Load Balancer.

L'élasticité est l'une des principales caractéristiques du cloud AWS. Vous pouvez lancer plus de capacité de calcul (par exemple, des serveurs web) lorsque vous en avez besoin et en exécuter moins lorsque vous n'en avez pas besoin. [AWS Auto Scaling](#) est un service AWS qui vous aide à automatiser cet approvisionnement afin de mettre à l'échelle votre capacité Amazon EC2 à la hausse ou à la baisse selon les conditions que vous définissez, sans avoir besoin d'une intervention manuelle. Vous pouvez configurer AWS Auto Scaling afin que le nombre d'instances EC2 que vous utilisez augmente de manière transparente pendant les pics de demande pour maintenir les performances et diminue automatiquement lorsque le trafic diminue, afin de réduire les coûts.

Elastic Load Balancing prend également en charge l'ajout et la suppression dynamiques d'hôtes Amazon EC2 dans la rotation d'équilibrage de charge. Elastic Load Balancing augmente et diminue lui-même de manière dynamique la capacité d'équilibrage de charge pour s'adapter aux exigences du trafic sans intervention manuelle.

## Niveau web sans état

Pour pouvoir exploiter plusieurs serveurs web dans une configuration de scalabilité automatique, votre niveau web doit être sans état. Une application sans état est une application qui n'a pas besoin de connaître les interactions précédentes et qui ne stocke aucune information de session. Dans le cas de WordPress, cela signifie que tous les utilisateurs finaux reçoivent la même réponse, quel que soit le serveur web ayant traité leur demande. Une application sans état peut être mise à l'échelle horizontalement, car toute demande peut être traitée par n'importe laquelle des ressources de calcul disponibles (c'est-à-dire des instances de serveur web). Lorsque cette capacité n'est plus requise, toute ressource individuelle peut être résiliée en toute sécurité (une fois les tâches en cours d'exécution épuisées). Ces ressources n'ont pas besoin d'être conscientes de la présence de leurs homologues ; il suffit de trouver un moyen de répartir la charge de travail sur ces dernières.

En ce qui concerne le stockage des données de session utilisateur, le noyau de WordPress est complètement sans état, car il repose sur des cookies qui sont stockés dans le navigateur web du client. Le stockage de session n'est pas un problème, sauf si vous avez installé un code personnalisé (par exemple, un plugin WordPress) qui repose plutôt sur des sessions PHP natives.

Cependant, WordPress a été initialement conçu pour fonctionner sur un seul serveur. Par conséquent, certaines données sont stockées sur le système de fichiers local du serveur. Lorsque vous exécutez WordPress dans une configuration multi-serveurs, cela crée un problème car il existe une incohérence entre les serveurs web. Par exemple, si un utilisateur charge une nouvelle image, elle n'est stockée que sur l'un des serveurs.

Cela montre pourquoi nous devons améliorer la configuration d'exécution par défaut de WordPress pour déplacer des données importantes vers un stockage partagé. L'architecture sur laquelle reposent les bonnes pratiques dispose d'une base de données sous forme de couche distincte en dehors du serveur web et utilise le stockage partagé pour stocker les chargements, les thèmes et les plugins des utilisateurs.

## Stockage partagé (Amazon S3 et Amazon EFS)

Par défaut, WordPress stocke les chargements des utilisateurs sur le système de fichiers local et n'est donc pas sans état. Par conséquent, nous devons déplacer l'installation de WordPress et toutes les personnalisations utilisateur (telles que la configuration, les plugins, les thèmes et les chargements générés par les utilisateurs) vers une plateforme de données partagée afin de réduire la charge sur les serveurs web et de faire passer le niveau web au mode sans état.

[Amazon Elastic File System](#) (Amazon EFS) fournit des systèmes de fichiers réseau évolutifs à utiliser avec les instances EC2. Les systèmes de fichiers Amazon EFS sont distribués entre un nombre illimité de serveurs de stockage, ce qui permet aux systèmes de fichiers d'opérer une mise à l'échelle élastique et offre un accès parallèle massif depuis les instances EC2. La conception distribuée d'Amazon EFS évite les goulots d'étranglement et les contraintes inhérentes aux serveurs de fichiers traditionnels.

En déplaçant l'intégralité du répertoire d'installation de WordPress sur un système de fichiers EFS et en le montant dans chacune de vos instances EC2 au démarrage, votre site WordPress et toutes ses données sont automatiquement stockés sur un système de fichiers distribué qui ne dépend d'aucune instance EC2, ce qui rend votre niveau web complètement sans état. L'avantage de cette architecture est que vous n'avez pas besoin d'installer de plugins et de thèmes à chaque lancement de nouvelle instance, et vous pouvez accélérer considérablement l'installation et la récupération des instances WordPress. Il est également plus facile de déployer les modifications apportées aux plugins et aux thèmes dans WordPress, comme indiqué dans la section [Considérations relatives au déploiement](#) de ce document.

Pour garantir des performances optimales pour votre site web lorsqu'il est exécuté à partir d'un système de fichiers EFS, vérifiez les paramètres de configuration recommandés pour Amazon EFS et OPcache dans le document [AWS Reference Architecture for WordPress](#).

Vous avez également la possibilité de télécharger toutes les ressources statiques, telles que les fichiers image, CSS et JavaScript, vers un compartiment S3 avec la mise en cache CloudFront au premier plan. Le mécanisme pour effectuer cette opération dans une architecture multi-serveurs est exactement le même que pour une architecture à serveur unique, comme indiqué dans la section [Contenu statique](#) de ce livre blanc. Les avantages sont les mêmes que dans l'architecture à serveur unique : vous pouvez décharger le travail associé à la diffusion de vos ressources statiques vers Amazon S3 et CloudFront, ce qui permet à vos serveurs web de se concentrer uniquement sur la génération de contenu dynamique et de répondre à davantage de demandes d'utilisateurs par serveur web.

## Niveau données (Amazon Aurora et Amazon ElastiCache)

L'installation de WordPress étant stockée sur un système de fichiers réseau distribué, évolutif et partagé, et les ressources statiques étant mises à disposition à partir d'Amazon S3, vous pouvez vous concentrer sur le dernier composant avec état : la base de données. Comme pour le niveau stockage, la base de données ne doit pas dépendre d'un seul serveur ; elle ne peut donc pas être

hébergée sur l'un des serveurs web. Vous devez héberger la base de données WordPress sur Amazon Aurora.

[Amazon Aurora](#) est une base de données relationnelle compatible avec MySQL et PostgreSQL, créée pour le cloud, qui associe les performances et la disponibilité des bases de données commerciales haut de gamme à la simplicité et à la rentabilité des bases de données open source. Aurora MySQL augmente les performances et la disponibilité de MySQL en intégrant étroitement le moteur de base de données à un système de stockage distribué spécialement conçu, soutenu par une couche SSD. Il est tolérant aux pannes et auto-réparateur, réplique six copies de vos données dans trois zones de disponibilité, est conçu pour une disponibilité supérieure à 99,99 % et sauvegarde en permanence vos données dans Amazon S3. Amazon Aurora est conçu pour détecter automatiquement toute panne de la base de données et redémarrer sans effectuer de reprise sur incident ni recréer le cache de la base de données.

Amazon Aurora fournit un certain nombre de [types d'instances](#) adaptés à différents profils d'application, en particulier des instances optimisées pour la mémoire et des instances pouvant être éclatées. Pour améliorer les performances de votre base de données, vous pouvez sélectionner un type d'instance de grande taille afin de fournir davantage de ressources CPU et mémoire.

Amazon Aurora gère automatiquement le basculement entre l'instance principale et les [réplicas Aurora](#) afin que vos applications puissent reprendre les opérations de base de données le plus rapidement possible sans intervention administrative manuelle. Le basculement prend généralement moins de 30 secondes.

Après avoir créé au moins un réplica Aurora, connectez-vous à votre instance principale à l'aide du point de terminaison du cluster pour permettre à votre application de basculer automatiquement en cas d'échec de l'instance principale. Vous pouvez créer jusqu'à 15 réplicas en lecture à faible latence dans trois zones de disponibilité.

À mesure que votre base de données évolue, le cache de votre base de données doit également évoluer. Comme indiqué précédemment dans la section [Mise en cache de base de données](#), ElastiCache propose des fonctionnalités permettant de mettre à l'échelle le cache sur plusieurs nœuds d'un cluster ElastiCache et sur plusieurs zones de disponibilité d'une région afin d'améliorer la disponibilité. Lors de la mise à l'échelle de votre cluster ElastiCache, veillez à configurer votre plugin de mise en cache pour qu'il se connecte à l'aide du point de terminaison de configuration afin que WordPress puisse utiliser les nouveaux nœuds de cluster au fur et à mesure de leur ajout, et arrêter d'utiliser les anciens nœuds de cluster. Vous devez également configurer vos serveurs web pour utiliser [ElastiCache Cluster Client for PHP](#) et mettre à jour votre AMI pour stocker cette modification.

# WordPress High Availability by Bitnami sur AWS Quick Start

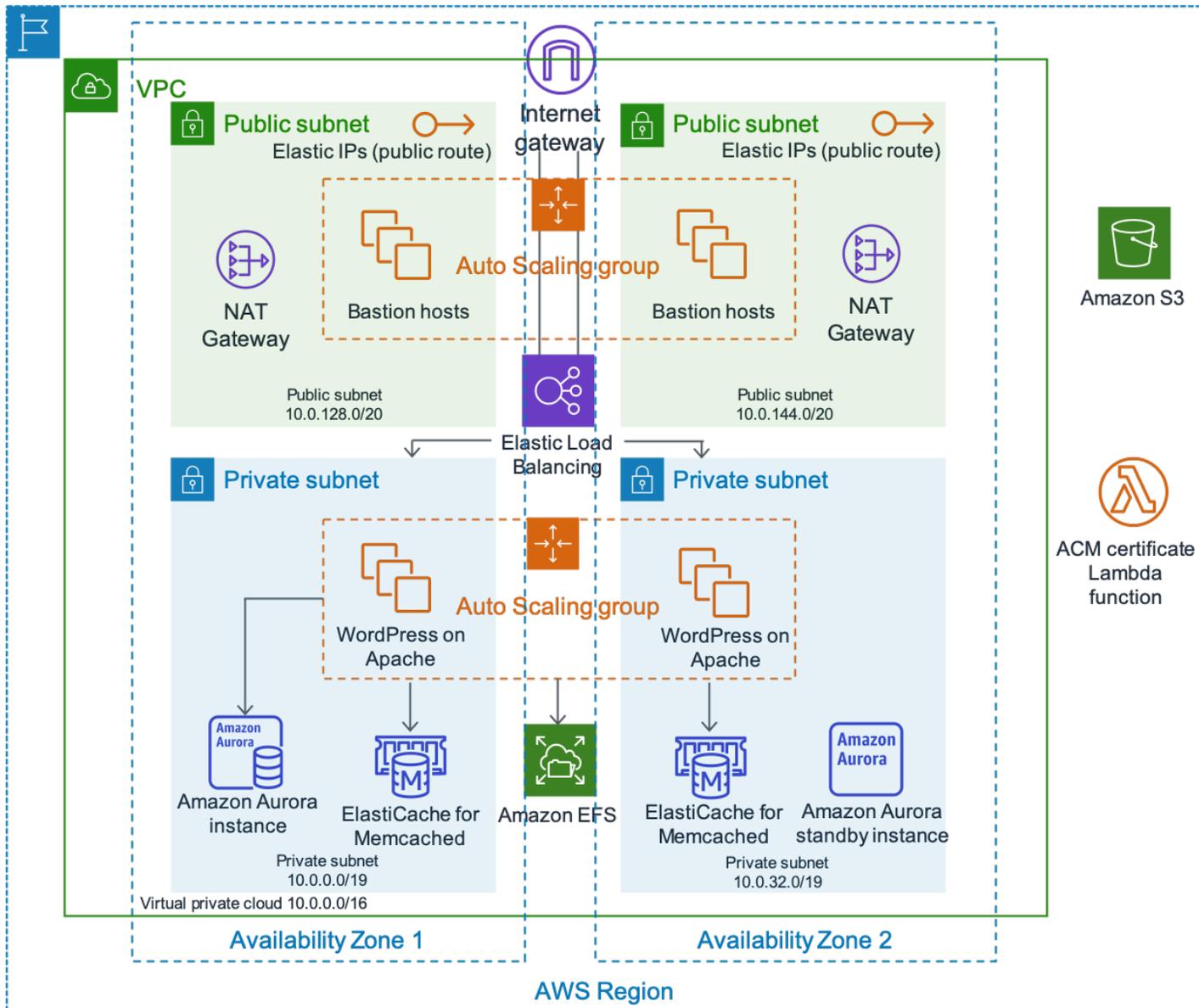
Les Quick Starts sont conçus par les architectes de solutions et partenaires AWS pour vous aider à déployer des technologies connues sur AWS en vous basant sur les bonnes pratiques d'AWS pour en assurer la sécurité et la haute disponibilité. Ces accélérateurs permettent de passer de centaines de procédures manuelles à seulement quelques étapes et vous permettent de créer rapidement votre environnement de production et de commencer à l'utiliser immédiatement. Chaque Quick Start comprend des modèles AWS CloudFormation qui automatisent le déploiement, ainsi qu'un guide qui explique l'architecture et indique les instructions de déploiement étape par étape.

[WordPress High Availability by Bitnami sur AWS Quick Start](#) met en place l'environnement configurable suivant sur AWS :

- Une architecture hautement disponible qui s'étend sur deux zones de disponibilité.\*
- Un Virtual Private Cloud (VPC) configuré avec des sous-réseaux publics et privés conformément aux bonnes pratiques AWS. Vous disposez ainsi de l'infrastructure réseau nécessaire à votre déploiement.\*
- Une passerelle Internet pour permettre l'accès à Internet. Cette passerelle est utilisée par les hôtes bastion pour envoyer et recevoir le trafic.\*
- Dans les sous-réseaux publics, des passerelles NAT gérées pour autoriser l'accès Internet sortant pour les ressources des sous-réseaux privés.\*
- Dans les sous-réseaux publics, les hôtes bastion Linux dans un groupe Auto Scaling pour permettre un accès Secure Shell (SSH) entrant aux instances EC2 des sous-réseaux publics et privés.\*
- Elastic Load Balancing pour distribuer les requêtes HTTP et HTTPS sur plusieurs instances WordPress.
- Dans les sous-réseaux privés, les instances EC2 qui hébergent l'application WordPress sur Apache. Ces instances sont allouées dans un groupe Auto Scaling afin de garantir une haute disponibilité.
- Dans les sous-réseaux privés, les instances de base de données Amazon Aurora administrées par Amazon Relational Database Service (Amazon RDS).
- Dans les sous-réseaux privés, Amazon Elastic File System (Amazon EFS) pour partager des ressources, par exemple des plugins, des thèmes et des images, sur plusieurs instances WordPress.

- Dans les sous-réseaux privés, les nœuds Amazon ElastiCache for Memcached pour la mise en cache de requêtes de base de données.

\* Le modèle qui déploie le Quick Start dans un VPC existant ignore les tâches marquées par des astérisques et vous invite à indiquer votre configuration de VPC existante.



### Architecture WordPress High Availability by Bitnami

La description détaillée du déploiement de WordPress High Availability by Bitnami sur AWS dépasse le cadre de ce document. Pour connaître la configuration et les options, reportez-vous à la section [WordPress High Availability by Bitnami sur AWS](#).

## Conclusion

AWS présente de nombreuses options d'architecture pour l'exécution de WordPress. L'option la plus simple consiste à installer un serveur unique pour les sites web à faible trafic. Pour les sites web plus avancés, les administrateurs de sites peuvent ajouter plusieurs autres options, chacune représentant une amélioration progressive en termes de disponibilité et de capacité de mise à l'échelle. Les administrateurs peuvent sélectionner les fonctionnalités qui correspondent le mieux à leurs exigences et à leur budget.

# Participants

Ont contribué à la préparation du présent document :

- Paul Lewis, architecte de solutions, Amazon Web Services
- Ronan Guilfoyle, architecte de solutions, Amazon Web Services
- Andreas Chatzakis, responsable des architectes de solutions, Amazon Web Services
- Jibril Touzi, gestionnaire des comptes techniques, Amazon Web Services
- Hakmin Kim, architecte de solutions partenaires de migration, Amazon Web Services

# Révisions du document

Pour être informé des mises à jour de ce livre blanc, abonnez-vous au flux RSS.

update-history-change	update-history-description	update-history-date
<a href="#">Livre blanc mis à jour</a>	Mise à jour pour modifier l'architecture de référence et le plugin AWS for WordPress.	19 octobre 2021
<a href="#">Livre blanc mis à jour</a>	Mise à jour pour inclure les nouvelles approches de déploiement et le plugin AWS for WordPress.	30 octobre 2019
<a href="#">Livre blanc mis à jour</a>	Mise à jour pour clarifier les messages relatifs au produit Amazon Aurora.	1er février 2018
<a href="#">Livre blanc mis à jour</a>	Mise à jour pour inclure les services AWS lancés depuis la première publication.	1er décembre 2017
<a href="#">Publication initiale</a>	Première publication.	1er décembre 2014

# Annexe A : Configuration de CloudFront

Pour obtenir des performances et une efficacité optimales lors de l'utilisation d'Amazon CloudFront avec votre site web WordPress, il est important de configurer le site web correctement pour les différents types de contenu diffusés.

Rubriques

- [Origines et comportements](#)
- [Création d'une distribution CloudFront](#)

## Origines et comportements

Une [origine](#) est un emplacement où CloudFront envoie des demandes de contenu qu'il distribue via les emplacements périphériques. Selon votre implémentation, vous pouvez avoir une ou deux origines. Une pour le contenu dynamique (l'instance Lightsail dans l'[option de déploiement à serveur unique](#), ou l'Application Load Balancer dans l'[option de déploiement élastique](#) utilisant une origine personnalisée. Vous pouvez avoir une deuxième origine vers laquelle diriger CloudFront pour votre contenu statique. Dans l'[architecture de référence](#) précédente, il s'agit d'un compartiment S3. Lorsque vous utilisez Amazon S3 comme origine pour votre distribution, vous devez utiliser une [politique de compartiment](#) pour rendre le contenu accessible au public.

Les [comportements](#) vous permettent de définir des règles qui régissent la manière dont CloudFront met en cache votre contenu et, par conséquent, de déterminer l'efficacité du cache. Les comportements vous permettent de contrôler le protocole et les méthodes HTTP grâce auxquels votre site web est accessible. Ils vous permettent également de définir si vous souhaitez transmettre des en-têtes HTTP, des cookies ou des chaînes de requête à votre backend (et, le cas échéant, lesquels). Les comportements s'appliquent à des modèles de chemin d'URL spécifiques.

## Création d'une distribution CloudFront

Créez une distribution web CloudFront en suivant la distribution. L'origine et le comportement par défaut créés automatiquement seront utilisés pour le contenu dynamique. Créez quatre comportements supplémentaires pour personnaliser davantage la façon dont les demandes statiques et dynamiques sont traitées. Le tableau suivant récapitule les propriétés de configuration des cinq comportements. Vous pouvez également ignorer cette configuration manuelle et utiliser le plugin

AWS for WordPress décrit dans l'[Annexe B : Installation et configuration des plugins](#). Il s'agit du moyen le plus simple de configurer CloudFront pour accélérer votre site WordPress.

Tableau 1 : Résumé des propriétés de configuration pour les comportements CloudFront

Propriété	Statique	Dynamique (admin)	Dynamique (appli frontale)
Chemins (Comportements)	wp-content/* wp-includes/*	wp-admin/* wp-login.php	par défaut (*)
Protocoles	HTTP et HTTPS	Redirection vers HTTPS	HTTP et HTTPS
Méthodes HTTP	GET, HEAD	ALL	ALL
En-têtes HTTP	NONE	ALL	Host CloudFront-Forwarded-Proto CloudFront-Is-Mobile-Viewer CloudFront-Is-Tablet-Viewer CloudFront-Is-Desktop-Viewer
Cookies	NONE	ALL	comment_* wordpress_* wp-settings-*
Chaînes de requête	OUI (invalidation)	OUI	OUI

Pour le comportement par défaut, AWS recommande la configuration suivante :

- Faites en sorte que la stratégie de protocole d'origine corresponde à celle de l'utilisateur, de sorte que si les utilisateurs se connectent à CloudFront à l'aide du protocole HTTPS, CloudFront se connecte également à votre origine via HTTPS, réalisant ainsi un chiffrement de bout en bout. Notez que cela nécessite l'installation d'un certificat SSL approuvé sur l'équilibreur de charge. Pour plus d'informations, reportez-vous à la section [Exigence du protocole HTTPS pour les communications entre CloudFront et votre origine personnalisée](#).
- Autorisez toutes les méthodes HTTP puisque les parties dynamiques du site web nécessitent à la fois des requêtes GET et POST (par exemple, pour prendre en charge l'opération POST pour les formulaires de soumission de commentaires).
- Transférez uniquement les cookies qui modifient la sortie de WordPress ; par exemple `>wordpress_*`, `wp-settings-*` et `comment_*`. Vous devez étendre cette liste si vous avez installé des plugins qui dépendent d'autres cookies ne figurant pas dans la liste.
- Transférez uniquement les en-têtes HTTP qui affectent la sortie de WordPress, par exemple `Host`, `CloudFront-Forwarded-Proto`, `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` et `CloudFront-is-Tablet-Viewer` :
  - `Host` permet d'héberger plusieurs sites web WordPress sur la même origine.
  - `CloudFront-Forwarded-Proto` permet la mise en cache de différentes versions de pages selon qu'elles sont accessibles via HTTP ou HTTPS.
  - `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` et `CloudFront-is-Tablet-Viewer` vous permettent de personnaliser la sortie de vos thèmes en fonction du type d'appareil de l'utilisateur final.
- Transférez toutes les chaînes de requête vers le cache en fonction de leurs valeurs, car WordPress s'appuie sur celles-ci. Elles peuvent également être utilisées pour invalider les objets mis en cache.

Si vous souhaitez diffuser votre site web sous un nom de domaine personnalisé (c'est-à-dire autre que `*.cloudfront.net`), entrez les URI appropriés sous Alternate Domain Names (Noms de domaine alternatifs) dans les paramètres de distribution. Dans ce cas, vous avez également besoin d'un certificat SSL pour votre nom de domaine personnalisé. Vous pouvez [demander](#) des certificats SSL via AWS Certificate Manager et les configurer par rapport à une distribution CloudFront.

À présent, créez deux autres comportements de cache pour le contenu dynamique : un pour la page de connexion (modèle de chemin : `wp-login.php`) et un pour le tableau de bord d'administration (modèle de chemin : `wp-admin/*`). Ces deux comportements ont exactement les mêmes paramètres, comme suit :

- Appliquez une stratégie de protocole d'utilisateur HTTPS uniquement.
- Autorisez toutes les méthodes HTTP.
- Mettez en cache en fonction de tous les en-têtes HTTP.
- Transférez tous les cookies.
- Transférez et mettez en cache en fonction de toutes les chaînes de requête.

Cette configuration s'explique par le fait que cette section du site web est hautement personnalisée et ne compte généralement que quelques utilisateurs. L'efficacité de la mise en cache n'est donc pas une préoccupation majeure. L'objectif est de simplifier la configuration afin d'assurer une compatibilité maximale avec tous les plugins installés en transmettant tous les cookies et en-têtes à l'origine.

Le [plugin AWS for WordPress](#) décrit dans l'Annexe B crée automatiquement une distribution CloudFront qui répond à la configuration précédente.

Par défaut, WordPress stocke tout localement sur le serveur web, à savoir le stockage par blocs (Amazon EBS) pour le [déploiement sur un seul serveur](#) et le stockage par fichiers (Amazon EFS) pour le [déploiement élastique](#). En plus de réduire les coûts de stockage et de transfert de données, le déplacement des ressources statiques vers Amazon S3 offre capacité de mise à l'échelle, disponibilité des données, sécurité et performances. Il existe plusieurs plugins qui facilitent le déplacement de contenu statique vers Amazon S3 ; l'un d'entre eux est [W3 Total Cache](#), également abordé dans l'[Annexe B : Installation et configuration des plugins](#).

# Annexe B : Installation et configuration des plugins

## Rubriques

- [Plugin AWS for WordPress](#)
- [Configuration du contenu statique](#)

## Plugin AWS for WordPress

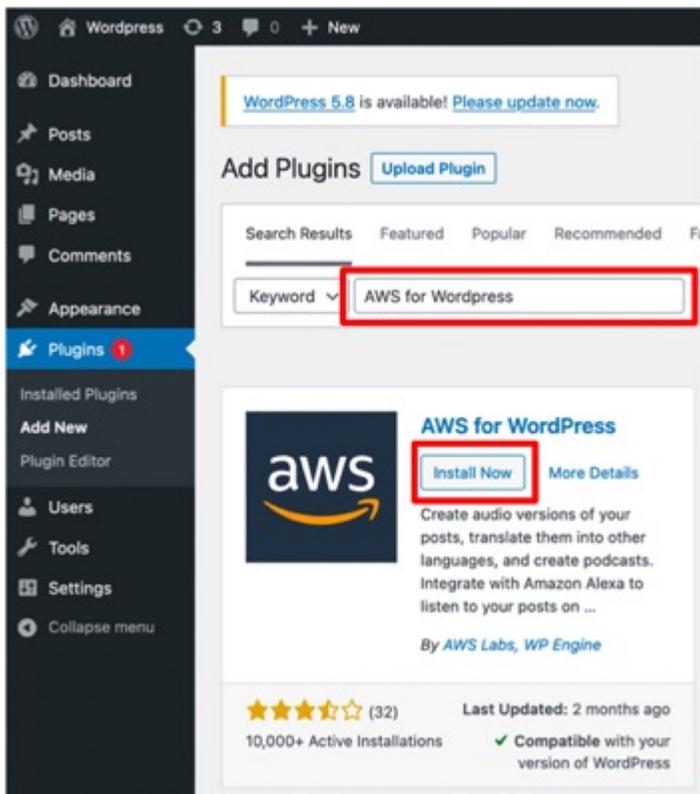
Le plugin AWS for WordPress est le seul plugin WordPress écrit et activement géré par AWS. Il permet aux clients de configurer facilement [Amazon CloudFront](#) et [AWS Certificate Manager \(ACM\)](#) sur des sites web WordPress afin d'en améliorer les performances et la sécurité. Le plugin utilise les services [Amazon Machine Learning](#) pour traduire du contenu dans une ou plusieurs langues, produit des versions audio de chaque traduction et lit les sites web WordPress via les appareils Amazon Alexa.

Le plugin est déjà installé dans [WordPress High Availability by Bitnami sur AWS Quick Start](#).

## Installation et configuration des plugins

Pour installer le plugin :

1. Pour utiliser le plugin AWS for WordPress, vous devez créer un utilisateur IAM pour ce dernier. Un utilisateur IAM est une personne ou une application sous un compte AWS qui a l'autorisation d'effectuer des appels d'API vers des services AWS.
2. Vous avez besoin d'un rôle [AWS Identity and Access Management](#) (IAM) ou d'un utilisateur IAM pour contrôler l'authentification et l'autorisation de votre compte AWS. Pour empêcher les utilisateurs non autorisés d'obtenir ces autorisations, protégez les informations d'identification de l'utilisateur IAM. Traitez la clé d'accès secrète comme un mot de passe ; stockez-la dans un endroit sûr et ne la partagez avec personne. Comme pour un mot de passe, [effectuez une rotation périodique de la clé d'accès](#). Si la clé d'accès secrète est divulguée par erreur, [supprimez-la immédiatement](#). Vous pouvez ensuite créer une nouvelle clé d'accès à utiliser avec le plugin AWS for WordPress.
3. Dans le menu Plugins du panneau d'administration de WordPress, recherchez AWS for WordPress et choisissez Install Now (Installer maintenant).



4. Si l'installation du plugin ne fonctionne pas, il y a peut-être un problème d'autorisation utilisateur. Connectez-vous au serveur web WordPress et suivez les instructions suivantes pour résoudre le problème.
- Ouvrez le fichier `wp-config.php` dans le répertoire d'installation de WordPress et recopiez le code suivant à la fin du fichier `wp-config.php` :

```
define('FS_METHOD', 'direct');
```

- Lancez la commande suivante pour accorder l'autorisation d'écriture :

```
chmod 777 <WordPress install directory>/wp-content
```

Avertissement : Il est risqué de conserver l'autorisation d'écriture 777. Si l'autorisation 777 est conservée, n'importe qui peut modifier ou supprimer ce dossier. Remplacez l'autorisation d'écriture par 755 ou moins après avoir terminé l'utilisation du plugin.

- Si vous utilisez l'architecture de référence, le répertoire d'installation de WordPress est « `/var/www/wordpress/<site directory>` ».

La description détaillée de tous les paramètres d'AWS for WordPress dépasse le cadre de ce document. Pour connaître la configuration et les options, reportez-vous à la section [Démarez avec le plugin AWS for WordPress](#).

## Amazon CloudFront et AWS Certificate Manager

Pour configurer CloudFront et AWS Certificate Manager :

1. Dans le menu du plugin, choisissez CloudFront et entrez les paramètres suivants :
  - Origin domain name (Nom du domaine d'origine) : domaine DNS du serveur d'origine HTTP à partir duquel CloudFront extrait le contenu de votre site web (tel que `example.com`).
  - Alternate domain name (Autres noms de domaine (CNAME)) : nom de domaine que vos visiteurs utilisent pour une expérience de site web accélérée. AWS recommande d'utiliser « `www` » devant le domaine (par exemple, `www.example.com`).
2. Choisissez Initiate Setup (Lancer la configuration) pour démarrer la configuration.

Le plugin demande automatiquement un certificat SSL pour le CNAME via ACM. Une fois que vous avez validé le jeton ACM en [mettant à jour les enregistrements DNS](#) avec les entrées CNAME, le plugin crée une distribution CloudFront qui respecte les bonnes pratiques définies dans l'[Annexe A](#).

### Note

Le plugin AWS for WordPress requiert le protocole HTTPS pour la communication entre CloudFront et votre origine personnalisée. Assurez-vous que votre origine possède un certificat SSL valide pour le nom du domaine d'origine. Pour plus d'informations, reportez-vous au document [Exigence du protocole HTTPS pour les communications entre CloudFront et votre origine personnalisée](#).

## Traduire et vocaliser votre contenu

Le plugin AWS for WordPress vous permet de traduire automatiquement du texte dans différentes langues et de convertir le contenu écrit en formats audio multilingues. Ces fonctionnalités sont optimisées par les services Amazon Machine Learning.

[Amazon Polly](#) est un service qui convertit le texte en un enregistrement audio réaliste. Avec des dizaines de voix dans différentes langues, vous pouvez sélectionner la voix idéale et créer des

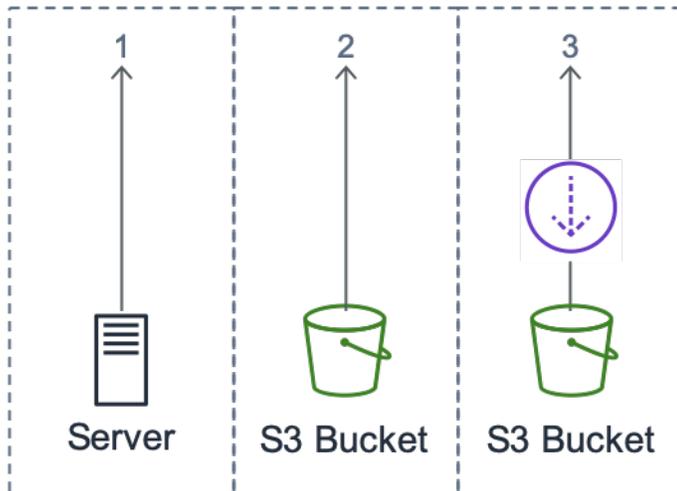
applications vocales qui fonctionnent dans de nombreux pays différents. Utilisez le plugin pour créer des fichiers audio dans l'une des voix et l'une des langues prises en charge par Amazon Polly. Vos visiteurs peuvent diffuser le contenu audio à leur convenance à l'aide de lecteurs audio en ligne et d'applications mobiles.

Par défaut, le plugin stocke les nouveaux fichiers audio sur votre serveur web. Vous pouvez choisir de stocker les fichiers sur Amazon S3 ou sur Amazon CloudFront. Les utilisateurs bénéficient de la même expérience d'écoute, peu importe l'emplacement de stockage de vos fichiers audio. Seul l'emplacement de diffusion change :

- Les fichiers audio stockés sur le serveur WordPress sont diffusés directement à partir du serveur.
- Les fichiers stockés dans un compartiment S3 sont diffusés depuis le compartiment.
- Si vous utilisez CloudFront, les fichiers sont stockés sur Amazon S3 et sont diffusés avec CloudFront.



Listening...

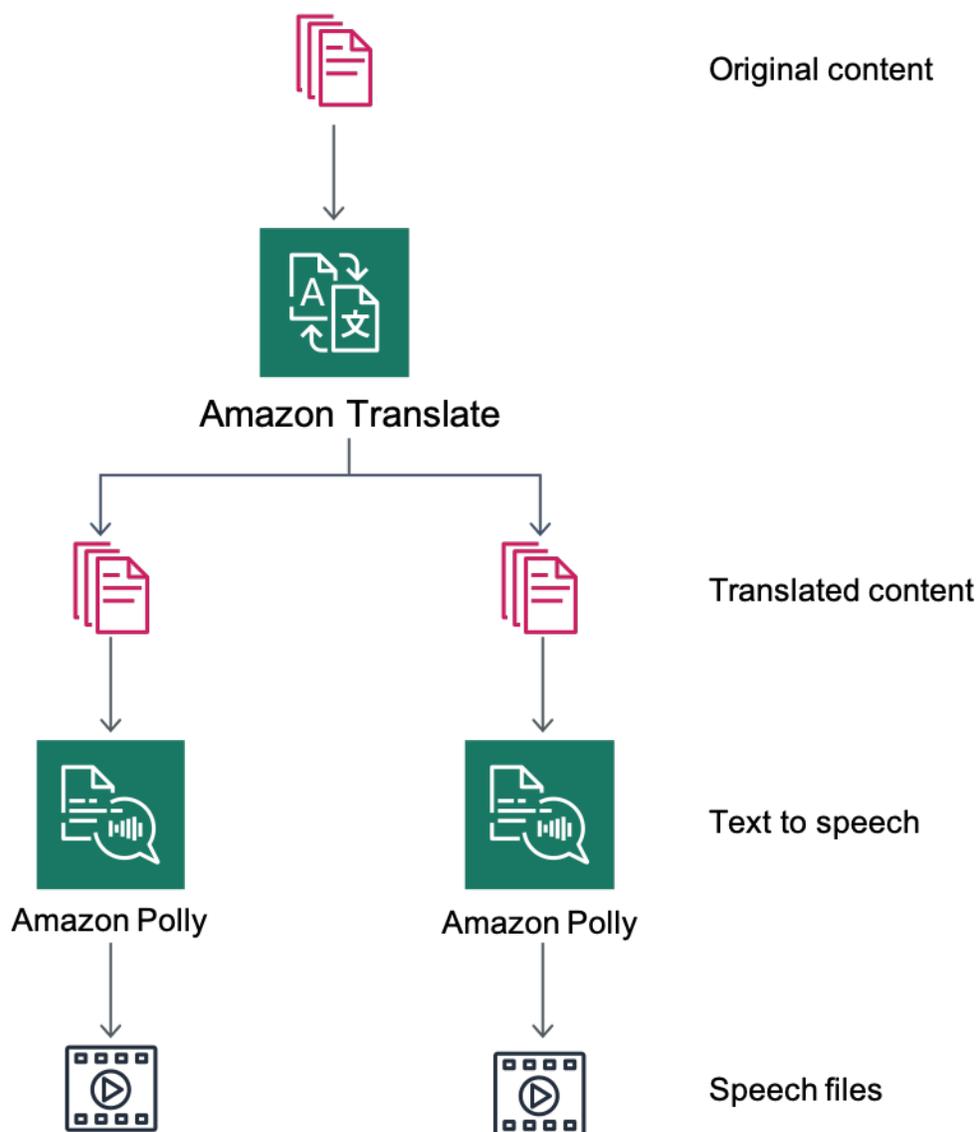


## Emplacement de diffusion

[Amazon Translate](#) est un service de traduction automatique qui offre une traduction linguistique rapide, de haute qualité et abordable. Pour les propriétaires de site, proposer du contenu multilingue représente une formidable opportunité. Si l'anglais est la langue qui prédomine sur le web, les anglophones natifs ne représentent que 26 % du total des lecteurs en ligne.

En proposant des versions écrites et audio de votre contenu WordPress dans de multiples langues, vous répondez aux besoins d'un public d'internautes international plus large. Vous pouvez configurer le plugin pour effectuer les opérations suivantes :

- Traduire automatiquement dans différentes langues et créer des enregistrements audio de chaque traduction pour le nouveau contenu lors de la publication, ou choisir de traduire et de créer des enregistrements pour des publications individuelles
- Traduire dans différentes langues et créer des enregistrements audio pour chaque traduction de votre contenu archivé
- Utiliser le flux RSS Amazon Pollycast pour diffuser du contenu audio



## Aperçu de la traduction du contenu et de la synthèse vocale

### Podcasting avec Amazon Pollycast

Les flux Amazon Pollycast permettent à vos visiteurs d'écouter votre contenu audio à l'aide des applications de podcast standard. Les flux Pollycast compatibles RSS 2.0 fournissent les données XML nécessaires à l'agrégation des podcasts par les applications de podcast mobiles, telles qu'iTunes, et les répertoires de podcast populaires.

Lorsque vous installez le plugin AWS for WordPress, vous découvrez une option permettant d'activer la génération de flux XML dans l'onglet de configuration Podcast. Vous avez également la possibilité de configurer plusieurs propriétés facultatives. Après avoir activé la fonctionnalité, vous recevez un lien vers le flux.

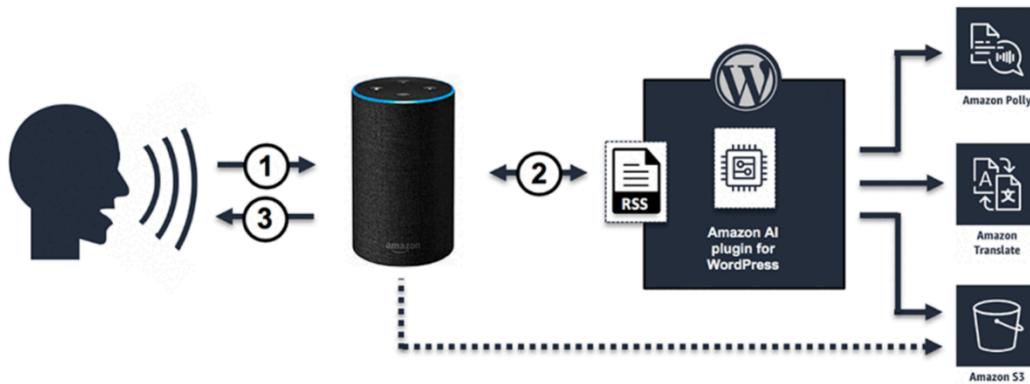
### Lecture de votre contenu via des appareils Amazon Alexa

Vous pouvez étendre les sites web et les blogs WordPress via des appareils Alexa. Cela offre de nouvelles possibilités aux créateurs et aux auteurs de sites web pour atteindre un public encore plus large. Cela permet également aux internautes d'écouter plus facilement leurs blogs préférés en demandant simplement à Alexa de les lire.

Pour exposer le site web WordPress sur Alexa, vous devez activer :

- Le plugin AWS for WordPress.
- Les fonctionnalités de synthèse vocale et Amazon Pollycast. Ces fonctionnalités génèrent sur votre site WordPress un flux RSS qui est consommé par Amazon Alexa.
- Amazon S3 étant le stockage par défaut de vos fichiers de synthèse vocale, il est important que votre site web utilise une connexion HTTPS sécurisée pour exposer son flux sur Alexa.

Le diagramme suivant présente le flux d'interactions et de composants nécessaires pour exposer votre site web via Alexa.



## Flux d'interactions nécessaires pour exposer les sites web WordPress via Alexa

1. L'utilisateur invoque une nouvelle compétence (skill) Alexa, par exemple en disant : Alexa, demande au blog de démonstration la dernière mise à jour. La compétence elle-même est créée à l'aide de l'un des plans Alexa Skill Blueprints. Cela vous permet d'exposer votre compétence via des appareils Alexa même si vous n'avez pas de connaissances techniques approfondies.
2. La compétence Alexa analyse l'appel et le flux RSS générés par le plugin AWS for WordPress, puis renvoie le lien vers la version audio du dernier article.
3. Sur la base du lien fourni par le flux, Alexa lit l'article en lisant le fichier audio enregistré sur Amazon S3.

Consultez la [page des plugins](#) sur le marketplace WordPress pour obtenir un guide détaillé étape par étape de l'installation et de la configuration du plugin et de ses fonctionnalités.

## Configuration du contenu statique

Par défaut, WordPress stocke tout localement sur le serveur web, à savoir le stockage par blocs (Amazon EBS) pour le [déploiement sur un seul serveur](#) et le stockage par fichiers (Amazon EFS) pour le [déploiement élastique](#). En plus de réduire les coûts de stockage et de transfert de données, le déplacement des ressources statiques vers Amazon S3 offre capacité de mise à l'échelle, disponibilité des données, sécurité et performances.

Dans cet exemple, le plugin W3 Total Cache (W3TC) est utilisé pour stocker des ressources statiques sur Amazon S3. Cependant, il existe d'autres plugins disponibles ayant des fonctionnalités similaires. Si vous souhaitez utiliser une alternative, vous pouvez adapter les étapes suivantes en conséquence. Ces étapes ne concernent que les fonctionnalités ou paramètres pertinents pour

cet exemple. La description détaillée de tous les paramètres dépasse le cadre de ce document. Reportez-vous à la [page du plugin W3 Total Cache](#) sur [wordpress.org](http://wordpress.org) pour plus d'informations.

## Création d'un utilisateur IAM

Vous devez créer un utilisateur AWS Identity and Access Management (IAM) pour que le plugin WordPress puisse stocker des ressources statiques dans Amazon S3. Pour connaître les étapes, reportez-vous au document [Création d'un utilisateur IAM dans votre compte AWS](#).

Remarque : les rôles IAM offrent un meilleur moyen de gérer l'accès aux ressources AWS, mais à l'heure où nous rédigeons cet article, le plugin W3 Total Cache ne prend pas en charge les [rôles IAM](#).

Prenez note des informations d'identification de sécurité de l'utilisateur et stockez-les de manière sécurisée. Vous en aurez besoin ultérieurement.

## Création d'un compartiment Amazon S3

1. Commencez par créer un compartiment Amazon S3 dans la région AWS de votre choix. Pour connaître les étapes, reportez-vous au document [Création d'un compartiment](#). Activez l'hébergement de site web statique pour le compartiment en suivant les instructions du [Didacticiel : Configuration d'un site web statique sur Amazon S3](#).
2. Créez une stratégie IAM pour fournir à l'utilisateur IAM créé précédemment un accès au compartiment S3 spécifié, puis attachez la politique à l'utilisateur IAM. Pour savoir comment créer la stratégie suivante, reportez-vous au document [Gestion des stratégies IAM](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1389783689000",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
```

```
        "arn:aws:s3:::wp-demo",
        "arn:aws:s3:::wp-demo/*"
    ]
}
]
```

3. Installez et activez le plugin W3TC depuis le panneau d'administration de WordPress.
4. Accédez à la section General Settings (Paramètres généraux) de la configuration du plugin et assurez-vous que les options Browser Cache (Cache du navigateur) et CDN sont activées.
5. Dans la liste déroulante de la configuration CDN, choisissez Origin Push: Amazon CloudFront (Push d'origine : Amazon CloudFront) (cette option a pour origine Amazon S3).
6. Accédez à la section Browser Cache (Cache du navigateur) de la configuration du plugin et activez les en-têtes expires (expire), cache control (contrôle de cache) et entity tag (étiquette d'entité) (ETag).
7. Activez également l'option Prevent caching of objects after settings change (Empêcher la mise en cache des objets après la modification des paramètres) afin qu'une nouvelle chaîne de requête soit générée et ajoutée aux objets chaque fois que des paramètres sont modifiés.
8. Accédez à la section CDN de la configuration du plugin et entrez les informations d'identification de sécurité de l'utilisateur IAM que vous avez créé précédemment, ainsi que le nom du compartiment S3.
9. Si vous diffusez votre site web via l'URL CloudFront, saisissez le nom du domaine de distribution dans la zone appropriée. Sinon, saisissez un ou plusieurs CNAME pour vos noms de domaine personnalisés.
10. Enfin, exportez la bibliothèque multimédia et chargez les fichiers wp-includes, les fichiers de thème et les fichiers personnalisés dans Amazon S3 à l'aide du plugin W3TC. Ces fonctions de chargement sont disponibles dans la section General (Général) de la page de configuration CDN.

## Création d'une origine statique

Maintenant que les fichiers statiques sont stockés sur Amazon S3, revenez à la configuration CloudFront dans la console CloudFront et configurez Amazon S3 en tant qu'origine du contenu statique. Pour ce faire, ajoutez une deuxième origine pointant vers le compartiment S3 que vous avez créé à cet effet. Créez ensuite deux autres comportements de cache, un pour chacun des deux dossiers (wp-content et wp-includes) qui doivent utiliser l'origine S3 au lieu de l'origine par défaut pour le contenu dynamique. Configurez les deux de la même manière :

- Traitez uniquement les requêtes HTTP GET.
- Amazon S3 ne modifie pas sa sortie en fonction des cookies ou des en-têtes HTTP. Vous pouvez donc améliorer l'efficacité de la mise en cache en ne les transférant pas à l'origine via CloudFront.
- Bien que ces comportements ne diffusent que du contenu statique (qui n'accepte aucun paramètre), vous allez transférer les chaînes de requête à l'origine. Vous pouvez ainsi utiliser des chaînes de requête comme identifiants de version pour invalider instantanément, par exemple, les anciens fichiers CSS lors du déploiement de nouvelles versions. Pour plus d'informations, reportez-vous au [Guide du développeur Amazon CloudFront](#).

#### Note

Après avoir ajouté les comportements d'origine du contenu statique à votre distribution CloudFront, vérifiez l'ordre pour vous assurer que les comportements de `wp-admin/*` et `wp-login.php` ont la priorité sur ceux du contenu statique. Sinon, vous risquez de voir un comportement étrange lorsque vous accédez à votre panneau d'administration.

## Annexe C : Sauvegarde et restauration

La reprise sur incident est plus rapide et plus facile à réaliser dans AWS que dans les environnements d'hébergement traditionnels. Par exemple, vous pouvez lancer une instance de remplacement en quelques minutes en réponse à une défaillance matérielle ou utiliser le basculement automatique dans bon nombre de nos services managés pour annuler l'impact d'un redémarrage dû à une maintenance de routine.

Cependant, vous devez toujours vous assurer d'avoir sauvegardé les bonnes données pour pouvoir les récupérer avec succès. Pour rétablir la disponibilité d'un site web WordPress, vous devez être capable de récupérer les composants suivants :

- Installation et configuration du système d'exploitation (OS) et des services (Apache, MySQL, etc.)
- Code et configuration de l'application WordPress
- Thèmes et plug-ins WordPress
- Chargements (par exemple, fichiers multimédias pour des publications)
- Contenu de la base de données (publications, commentaires, etc.)

AWS propose différentes méthodes pour sauvegarder et restaurer les données et les ressources de vos applications web.

Ce livre blanc a abordé précédemment l'utilisation des instantanés Lightsail pour protéger toutes les données stockées sur le stockage local de l'instance. Si votre site web WordPress ne fonctionne que sur l'instance Lightsail, des instantanés Lightsail réguliers devraient suffire pour que vous puissiez récupérer votre site web WordPress dans son intégralité. Toutefois, vous perdrez toujours toutes les modifications appliquées à votre site web depuis la prise du dernier instantané si vous effectuez une restauration à partir d'un instantané.

Dans un déploiement multiserveur, vous devez sauvegarder chacun des composants décrits précédemment à l'aide de différents mécanismes. Chaque composant peut avoir des exigences différentes en matière de fréquence de sauvegarde, par exemple, l'installation et la configuration du système d'exploitation et de WordPress changeront beaucoup moins fréquemment que le contenu généré par les utilisateurs et, par conséquent, pourront être sauvegardés moins fréquemment sans perdre de données en cas de récupération.

Pour sauvegarder l'installation et la configuration du système d'exploitation et des services, ainsi que le code et la configuration de l'application WordPress, vous pouvez créer une AMI d'une instance

EC2 correctement configurée. Les AMI peuvent avoir deux objectifs : servir de sauvegarde de l'état de l'instance et servir de modèle lors du lancement de nouvelles instances.

Pour sauvegarder le code et la configuration de l'application WordPress, vous devez utiliser des AMI et également des sauvegardes Aurora.

Pour sauvegarder les thèmes et plugins WordPress installés sur votre site web, sauvegardez le compartiment Amazon S3 ou le système de fichiers Amazon EFS sur lequel ils sont stockés.

- Pour les thèmes et les plugins stockés dans un compartiment S3, vous pouvez activer la [réplication entre régions](#) afin que tous les objets chargés dans votre compartiment principal soient automatiquement répliqués dans votre compartiment de sauvegarde dans une autre région AWS. La réplication entre régions nécessite que la [gestion des versions](#) soit activée à la fois sur vos compartiments source et de destination. Vous bénéficiez alors d'une couche de protection supplémentaire et pouvez revenir à une version précédente d'un objet donné de votre compartiment.
- Pour les thèmes et les plugins stockés sur un système de fichiers EFS, vous pouvez créer un AWS Data Pipeline pour copier les données de votre système de fichiers EFS de production vers un autre système de fichiers EFS, comme indiqué dans la page de documentation [Sauvegarde de vos systèmes de fichiers Amazon EFS](#). Vous pouvez également sauvegarder un système de fichiers EFS à l'aide de n'importe quelle application de sauvegarde que vous utilisez déjà.
- Pour sauvegarder les chargements des utilisateurs, vous devez suivre les étapes décrites précédemment pour la sauvegarde des thèmes et plugins WordPress.
- Pour sauvegarder le contenu de la base de données, vous devez utiliser la [sauvegarde Aurora](#). Aurora sauvegarde automatiquement votre volume de cluster et conserve les données de restauration pendant la durée de la période de rétention des sauvegardes. Comme les sauvegardes Aurora sont continues et progressives, vous pouvez rapidement procéder à une restauration à un point quelconque de la période de rétention des sauvegardes. Aucun impact sur les performances ni interruption du service de base de données ne se produit lors de l'écriture des données de sauvegarde. Vous pouvez définir la période de rétention des sauvegardes sur une période de 1 à 35 jours. Vous pouvez également créer des [instantanés de base de données manuels](#), qui sont conservés jusqu'à ce que vous les supprimiez. Les instantanés de base de données manuels sont utiles pour les sauvegardes et l'archivage à long terme.

## Annexe D : Déploiement de nouveaux plugins et thèmes

Peu de sites web restent statiques. Dans la plupart des cas, vous ajouterez périodiquement des thèmes et des plugins WordPress accessibles au public ou mettez à niveau vers une version WordPress plus récente. Dans d'autres cas, vous développerez vos propres thèmes et plugins personnalisés à partir de zéro.

Chaque fois que vous apportez un changement structurel à votre installation WordPress, il existe un certain risque d'introduire des problèmes imprévus. Au minimum, faites une sauvegarde du code de votre application, de la configuration et de la base de données avant d'appliquer tout changement important (comme l'installation d'un nouveau plugin). Pour les sites web commerciaux ou autres, testez d'abord ces modifications dans un environnement intermédiaire distinct. Avec AWS, il est facile de répliquer la configuration de votre environnement de production et d'exécuter l'ensemble du processus de déploiement en toute sécurité. Une fois vos tests terminés, vous pouvez simplement supprimer votre environnement de test et cesser de payer pour ces ressources. Ce livre blanc abordera ultérieurement certaines considérations spécifiques à WordPress.

Certains plugins écrivent des informations de configuration dans la table de base de données `wp_options` (ou introduisent des modifications de schéma de base de données), tandis que d'autres créent des fichiers de configuration dans le répertoire d'installation de WordPress. Étant donné que nous avons déplacé la base de données et le stockage vers des plateformes partagées, ces modifications sont immédiatement disponibles sur toutes vos instances en cours d'exécution, sans aucun effort supplémentaire de votre part.

Lors du déploiement de nouveaux thèmes dans WordPress, quelques efforts supplémentaires pourront être nécessaires. Si vous utilisez Amazon EFS uniquement pour stocker tous vos fichiers d'installation WordPress, les nouveaux thèmes seront immédiatement disponibles sur toutes les instances en cours d'exécution. Toutefois, si vous déchargez du contenu statique vers Amazon S3, vous devez en effectuer une copie sur le bon emplacement de compartiment. Les plugins tels que W3 Total Cache vous permettent de lancer manuellement cette tâche. Vous pouvez également automatiser cette étape dans le cadre d'un processus de génération.

Étant donné que les ressources de thème peuvent être mises en cache sur CloudFront et dans le navigateur, vous devez invalider les anciennes versions lorsque vous déployez des modifications. La meilleure façon d'y parvenir est d'inclure une sorte d'identifiant de version dans votre objet. Cet identifiant peut être une chaîne de requête avec un horodatage ou une chaîne aléatoire. Si vous utilisez le plugin W3 Total Cache, vous pouvez mettre à jour une chaîne de requête multimédia qui est ajoutée aux URL des fichiers multimédias.

## Mentions légales

Les clients sont responsables de leur propre évaluation indépendante des informations contenues dans ce document. Le présent document : (a) est fourni à titre informatif uniquement, (b) représente les offres et pratiques actuelles de produits AWS, qui sont susceptibles d'être modifiées sans préavis, et (c) ne crée aucun engagement ou assurance de la part d'AWS et de ses affiliés, fournisseurs ou concédants de licences. Les produits ou services AWS sont fournis « en l'état » sans garantie, représentation ou condition, de quelque nature que ce soit, explicite ou implicite. Les responsabilités et obligations d'AWS envers ses clients sont déterminées par les contrats AWS, et le présent document ne fait pas partie d'un contrat entre AWS et ses clients, ni le modifie.

© 2021, Amazon Web Services, Inc. ou ses sociétés apparentées. Tous droits réservés.