
WordPress : bonnes pratiques sur AWS

Architecture de référence des
sites web évolutifs alimentés
par WordPress

Andreas Chatzakis

Décembre 2014



(Veuillez consulter <http://aws.amazon.com/whitepapers> pour obtenir la dernière version de ce document.)

Table des matières

Table des matières	2
Résumé	2
Introduction	3
Mise en route (serveur unique)	3
Installation de WordPress	3
Sélection du type et la taille d'instance appropriés	3
Récupération après défaillance	4
Amélioration des performances et rentabilité	6
Montée en puissance	9
Séparation des couches web et base de données	9
Disponibilité de la couche données	11
Evolutivité et disponibilité de la couche web	13
Dimensionnement de la couche données	15
Considérations sur le développement et le déploiement	17
Réduction du risque de modification	17
Conception réactive	18
Récapitulatif	19

Résumé

WordPress est un outil de blog et un système de gestion de contenu (CMS) en open source, basé sur PHP et MySQL. Il est utilisé pour alimenter n'importe quel site, du blog personnel aux sites web à trafic élevé. Amazon Web Services (AWS) est destiné à fournir une infrastructure fiable, évolutive, sûre et ultra performante, conçue pour les applications web les plus exigeantes.

Le présent livre blanc fournit aux administrateurs système les instructions spécifiques sur la façon de démarrer avec WordPress sur AWS et d'améliorer la rentabilité du déploiement aussi bien que l'expérience de l'utilisateur final. Il présente aussi une architecture de référence qui prend en compte les exigences communes en matière d'évolutivité et de haute disponibilité.

Introduction

La première version de WordPress lancée en 2003 n'avait pas été créée dans l'esprit des infrastructures modernes, élastiques et évolutives basées sur le cloud. Grâce au travail de la communauté WordPress et au lancement de divers modules WordPress, les capacités de cette solution CMS se développent sans cesse. Il est désormais possible de créer une architecture WordPress qui tire parti des nombreux avantages de la plateforme AWS.

Mise en route (serveur unique)

Pour les blocs à faible trafic ou les sites web sans exigences strictes de haute disponibilité, le simple déploiement d'un seul serveur peut convenir. [Amazon Elastic Compute Cloud](#) (Amazon EC2) est un service web qui offre une capacité de calcul redimensionnable, ce qui vous permet de lancer un serveur virtuel en quelques minutes.¹

Installation de WordPress

Comme vous avez le contrôle complet de votre instance Amazon EC2, vous pouvez vous connecter avec l'accès racine pour installer et configurer tous les composants logiciels requis pour exécuter un site web WordPress. Une fois que vous avez terminé, vous pouvez enregistrer cette configuration comme AMI (Amazon Machine Image) et utiliser ce dernier pour lancer de nouvelles instances avec toutes les personnalisations que vous avez effectuées.

Une solution plus simple pour démarrer consiste à utiliser [AWS Marketplace](#), magasin en ligne qui aide les clients à rechercher, à acheter et à démarrer rapidement grâce à un large éventail de solutions logicielles.² Les clients peuvent utiliser le déploiement 1-Click d'AWS Marketplace pour lancer rapidement une instance Amazon EC2 basée sur un AMI publiquement disponible. Une gamme d'offres [WordPress est](#) disponible et permet aux utilisateurs de lancer un serveur virtuel préinstallé avec une base de données, un serveur web et le code de l'application WordPress.³

Sélection du type et la taille d'instance appropriés

Amazon EC2 propose une large sélection de [types d'instance](#) optimisés pour convenir à différents cas d'utilisation.⁴ Les types d'instance se composent de différentes combinaisons d'UC, de mémoire, de stockage et de capacité réseau, et vous offrent la souplesse nécessaire pour choisir le mélange approprié de ressources pour vos applications. Chaque type d'instance inclut une ou plusieurs tailles d'instance, ce qui vous permet de dimensionner vos ressources en fonction des exigences de votre charge de travail cible.

¹ <http://aws.amazon.com/ec2/>

² <https://aws.amazon.com/marketplace/>

³ https://aws.amazon.com/marketplace/search/results/ref=gtw_navgno_search_box?searchTerms=wordpress&search

⁴ <http://aws.amazon.com/ec2/instance-types/>

Taille d'instance initiale

S'il s'agit d'un nouveau projet, vous devez d'abord procéder à quelques hypothèses sur le type et la taille d'instance dont vous avez besoin. En règle générale les instances T2 offrent un choix économique pour les sites web à faible trafic qui n'utilisent qu'irrégulièrement la totalité de l'UC, mais ont besoin à l'occasion de transmettre en mode rafale. Les types d'instance M3 fournissent plus de capacité et un équilibre entre les ressources de calcul, de mémoire et de réseau. Ils constituent ainsi un choix idéal pour les sites web plus exigeants. Les deux considérations ci-dessus constituent les points de départ recommandés, mais le choix optimal dépend, entre autres, des modèles de trafic, de la complexité du site web et des ressources requises par les plug-ins installés. La supervision des performances de l'hôte peut vous aider à trouver le type et la taille d'instance optimaux pour votre site web WordPress.

Dimensionnement vertical (monter en puissance)

Au fur et à mesure que le trafic croît, vous pouvez augmenter la taille de votre instance pour gérer la charge additionnelle. Lors de l'utilisation d'une instance Amazon EC2 [soutenue par Amazon Elastic Block Store](#) (Amazon EBS), vous pouvez augmenter la taille de votre instance en arrêtant simplement l'instance, en modifiant son type d'instance type et en redémarrant.⁵ Veuillez noter que les données stockées sur le [stockage d'instance éphémère](#) sont perdues lors de ce processus, comme expliqué dans le *Guide de l'utilisateur Amazon EC2 pour Linux*.⁶

Récupération après défaillance

La récupération après défaillance d'une instance est plus rapide que dans les environnements d'hébergement traditionnels. Vous pouvez lancer une instance de remplacement en quelques minutes et utiliser une grande variété de fonctions qui aident à réduire les interruptions, même dans l'hypothèse d'un seul serveur.

Stockage et sauvegardes fiables

Pour rétablir la disponibilité d'un site web WordPress, vous devez pouvoir récupérer les composants suivants :

- Système d'exploitation, installation des services et configuration (Apache, MySQL, etc.)
- Code applicatif WordPress et configuration
- Thèmes et plug-ins WordPress
- Chargements (fichiers multimédias pour les publications)
- Contenu de base de données (publications, commentaires, etc.)

Comme recensé dans le livre blanc, [Approches de sauvegarde et de récupération avec Amazon Web Services](#), AWS offre une grande diversité de méthodes pour sauvegarder et restaurer les données et les ressources de votre application web :⁷

⁵ <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ComponentsAMIs.html#storage-for-the-root-device>

⁶ <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>

⁷ http://media.amazonwebservices.com/AWS_Backup_Recovery.pdf

Les volumes Amazon EBS offrent un stockage durable au niveau bloc à utiliser avec les instances Amazon EC2 (machines virtuelles). Les données de volume Amazon EBS sont répliquées sur plusieurs serveurs pour éviter la perte de données résultant de la défaillance d'un seul composant. Un volume EBS se comporte comme un périphérique de stockage en mode bloc brut et non formaté, que vous pouvez utiliser comme disque dur virtuel. Vous pouvez créer un système de fichiers par-dessus un volume EBS et l'attacher à votre instance Amazon EC2 pour fournir un stockage permanent des fichiers de base de données et multimédias WordPress. Les volumes Amazon EBS fournissent un stockage NAS hors instance qui persiste indépendamment de la durée d'exécution d'une instance Amazon EC2. Si une instance échoue, vous pouvez lancer une instance de remplacement Amazon EC2 et lui attacher le volume Amazon EBS. Cette opération peut même être entièrement automatisée à l'aide d'Auto Scaling (pour remplacer l'instance) et d'un script d'initialisation (pour attacher le volume Amazon EBS à la nouvelle instance).

A titre d'exemple, dans le scénario d'installation d'un seul serveur décrit précédemment, vous pouviez configurer votre système de telle sorte que le code défini par l'utilisateur, les ressources (les plug-ins, les thèmes et les chargements situés sous le dossier `wp-content`, par exemple) et le contenu de base de données (`/var/lib/mysql`, par exemple) soient stockés sur un volume Amazon EBS.

Les volumes Amazon EBS ont un taux annuel de défaillance (AFR) inférieur à celui des disques durs classiques, mais il est toujours plus sûr de suivre les bonnes pratiques de sauvegarde. De plus, il existe toujours le risque d'une erreur humaine (suppression de fichiers importants par un administrateur de site WordPress, par exemple). D'où l'importance de définir et de tester votre stratégie de sauvegarde.

Le service de stockage AWS le plus approprié pour stocker durablement les sauvegardes est [Amazon Simple Storage Service](#) (Amazon S3).⁸ Amazon S3 propose aux développeurs de logiciels une infrastructure de stockage des données hautement évolutive, fiable et à latence faible, à coût très bas et accessible via les API de service web REST et SOAP. Amazon S3 stocke de façon redondante vos objets non seulement sur plusieurs périphériques, mais aussi sur plusieurs équipements d'une région Amazon S3, ce qui offre une durabilité encore supérieure à celle que propose Amazon EBS. Une grande variété de plug-ins WordPress est disponible pour les sauvegardes planifiées et manuelles sur Amazon S3.⁹

Une méthode plus économique de sauvegarde des données stockées sur les volumes Amazon EBS consiste à utiliser la fonctionnalité d'instantané. Cette fonction crée une copie de sauvegarde d'un volume Amazon EBS à un instant donné qui est alors stocké dans Amazon S3. Les instantanés Amazon EBS sont stockés de façon incrémentielle : seuls les blocs qui ont été enregistrés depuis votre dernier instantané sont enregistrés et seuls les blocs modifiés vous sont facturés.

Enfin, il importe de noter que les AMI sont également stockés sur Amazon S3, ce qui offre un stockage durable de la configuration de base de vos serveurs.

⁸ <http://aws.amazon.com/s3/>

⁹ Les exemples incluent Updraft Plus (<https://wordpress.org/plugins/updraftplus/>), BackWPup Free (<https://wordpress.org/plugins/backwpup/>) et BackUpWordPress (<https://wordpress.org/plugins/backupwordpress/>). Notez que ni AWS ni l'auteur n'ont testé ou soutenu ces plug-ins ou tout autre plug-in spécifique.

Adressage IP et DNS

7 Les adresses IP élastiques sont des adresses IP statiques conçues pour le cloud computing dynamique et vous permettent de masquer les défaillances d'une instance ou d'une zone de disponibilité. Au lieu d'attendre que DNS ne se propage sur la totalité de vos clients, Amazon EC2 vous permet de résoudre les problèmes liés à votre instance ou à votre logiciel en remappant votre adresse IP élastique sur une instance de remplacement.

De plus, vous pouvez utiliser [Amazon Route 53](#), solution d'hébergement DNS hautement disponible d'Amazon, pour pointer votre nom de domaine vers l'adresse IP de l'instance qui héberge votre site web.¹⁰

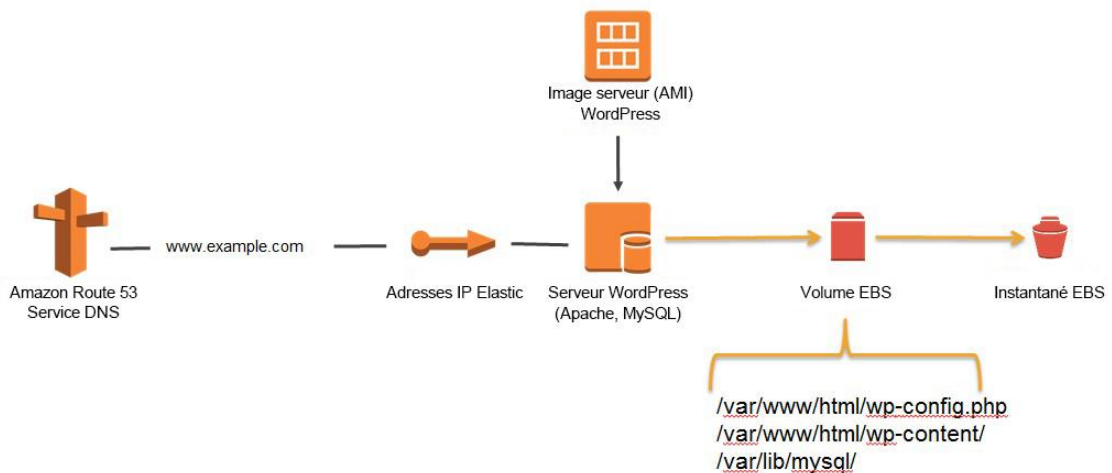


Figure 1 : Déploiement à une seule instance

Amélioration des performances et rentabilité

Même si la configuration ci-dessus peut, dans une certaine mesure, évoluer verticalement en utilisant un type d'instance aux spécifications supérieures, vous pouvez facilement augmenter les performances, réduire les coûts et améliorer l'expérience utilisateur final à l'aide de quelques modifications. Un grand nombre de ces modifications nécessite que vous utilisiez un ou plusieurs plug-ins WordPress. Bien que diverses options soient disponibles, [W3 Total Cache](#) constitue un choix répandu, qui regroupe la plupart de ces modifications en un seul plug-in.¹¹

¹⁰ <http://aws.amazon.com/route53/>

¹¹ <https://wordpress.org/plugins/w3-total-cache/>

Mise en cache navigateur et périphérique

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 via le code PHP WordPress (les éléments de votre site qui sont chargés depuis la base de données ou même personnalisés pour chaque visionneuse, par exemple). Un aspect important de l'expérience de l'utilisateur final est la latence réseau impliquée lors de la remise du contenu aux utilisateurs à travers le monde.

[Amazon CloudFront](#) est un service web qui offre aux entreprises et aux développeurs d'applications web une solution simple et économique pour répartir leurs contenus à latence faible et à vitesse de transfert de données élevée sur plusieurs emplacements périphériques à travers le monde.¹² Les demandes des visionneuses sont automatiquement acheminées vers un emplacement périphérique Amazon CloudFront adapté afin de réduire la latence. Si le contenu peut être mis en cache (pendant quelques secondes, minutes ou jours) et qu'il est déjà stocké dans un emplacement périphérique particulier, CloudFront le remet immédiatement. Si le contenu ne doit pas être mis en cache, qu'il a expiré ou qu'il ne se trouve pas encore dans cet emplacement périphérique, CloudFront l'extrait de l'origine configurée comme source de la version définitive du contenu. Cette extraction a lieu sur les connexions réseau optimisées, lesquelles œuvrent à accélérer la remise du contenu sur votre site web. En dehors d'améliorer l'expérience utilisateur finale, le modèle ci-dessus réduit aussi la charge sur vos serveurs d'origine et a le potentiel nécessaire pour créer des économies significatives.

Vous pouvez créer une distribution CloudFront, la mapper sur le nom de domaine de votre site web et configurer deux origines comme sources du contenu. Vous configurez les règles sous la forme de comportements CloudFront qui définissent l'origine à utiliser en fonction de modèles de chemin d'accès spécifiques.

Contenu statique

Ce contenu inclut CSS, JavaScript et les fichiers image (ceux qui font partie de vos thèmes WordPress ou les fichiers multimédias chargés par les administrateurs de contenu). Tous ces fichiers peuvent être stockés dans Amazon S3 (comme décrit précédemment dans [Stockage et sauvegardes fiables](#)). En dehors du stockage sécurisé pour le contenu privé comme les sauvegardes, vous pouvez configurer Amazon S3 de manière à servir le contenu public d'une manière évolutive et hautement disponible.

Il en résulte l'effet secondaire positif de décharger cette charge de travail de votre serveur web Amazon EC2 et de pouvoir se concentrer sur la génération du contenu dynamique. La charge sur le serveur s'en trouve ainsi réduite, mais, plus loin dans le présent document, nous verrons comment il en résulte aussi une architecture sans état (et pourquoi il s'agit d'un prérequis avant de pouvoir implémenter Auto Scaling).

Vous pouvez par la suite configurer Amazon S3 comme origine de CloudFront afin d'améliorer la remise de ces ressources statiques aux utilisateurs à travers le monde. Même si WordPress n'est pas intégré à Amazon S3 et CloudFront dès la première utilisation, une grande variété de plug-ins ajoutent cette prise en charge ([W3 Total Cache](#), par exemple).

¹² <http://aws.amazon.com/cloudfront/>

Contenu dynamique

Ce contenu inclut la sortie des scripts PHP WordPress côté serveur et peut aussi être remis via CloudFront en configurant le serveur web Amazon EC2 comme origine. Comme il inclut aussi le contenu personnalisé, vous devez configurer CloudFront pour acheminer certains cookies HTTP et en-têtes HTTP comme partie d'une demande de votre serveur d'origine personnalisé. CloudFront utilise les valeurs de cookie acheminées comme parties de la clé qui identifie un objet unique de son cache. Pour garantir que vous optimisez l'efficacité de la mise en cache, vous devez configurer CloudFront pour n'acheminer que les cookies HTTP et en-têtes HTTP qui modifient réellement le contenu (pas les cookies qui ne sont utilisés que côté client ou par des applications tierces, telles que les analyses web par exemple).

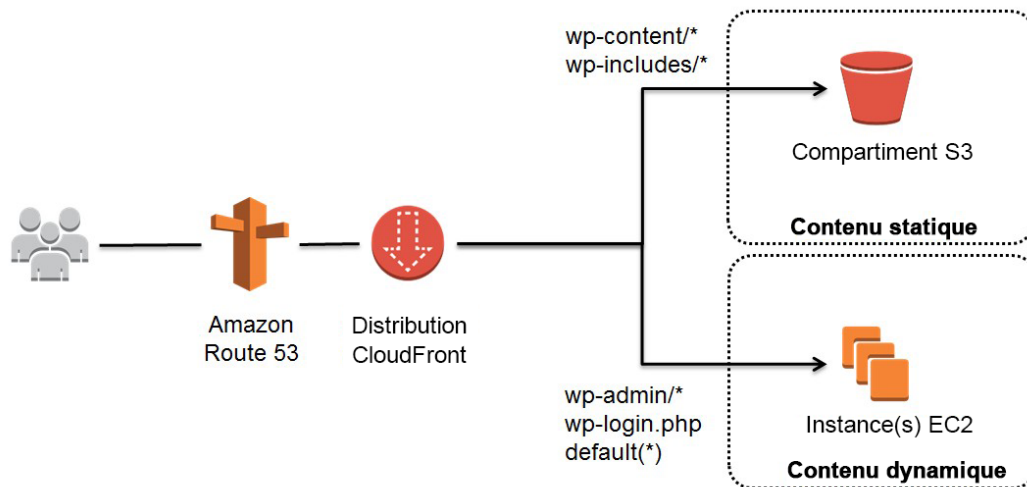


Figure 2 : Diffusion de site complet via CloudFront

Amazon CloudFront utilise les en-têtes standard de contrôle de cache pour identifier s'il doit mettre en cache des réponses HTTP spécifiques, et pendant combien de temps. Les mêmes en-têtes de contrôle de cache sont aussi utilisés par les navigateurs web pour décider quand et pour combien de temps le contenu doit être mis en cache localement afin que l'utilisateur final bénéficie d'une expérience encore plus optimale. (Par exemple, un fichier `.css` qui est déjà téléchargé ne sera pas à nouveau téléchargé chaque fois qu'un visiteur revient sur une page qu'il a déjà consultée.) Vous pouvez configurer cet aspect au niveau du serveur web (via les fichiers `.htaccess` ou les modifications du fichier `httpd.conf`) ou installer un plug-in WordPress ([W3 Total Cache](#), par exemple) pour indiquer comment ces en-têtes sont définis pour le contenu statique et pour le contenu dynamique.

Mise en cache de base de données

La mise en cache de base de données peut réduire sensiblement la latence et augmenter le débit des charges de travail d'application intenses en lecture, telles que WordPress. Les performances des applications sont améliorées en stockant les données fréquemment accédées en mémoire à des fins d'accès à faible latence (les résultats de requêtes de base de données gourmandes en E/S). Quand un pourcentage important de requêtes sont servies depuis le cache, le nombre de requêtes qui doivent accéder à la base de données est réduit, d'où un coût moindre associé au dimensionnement de la base de données.

Même WordPress possède des capacités de mise en cache prêtes à l'utilisation limitées, une grande variété de plug-ins prennent en charge l'intégration à Memcached, un système de mise en cache des objets en mémoire largement adopté. Le plug-in [W3 Total Cache](#) en constitue un bon exemple.

Vous pouvez installer Memcached sur une instance Amazon EC2. Dans les scénarios les plus simples, vous installez Memcached sur votre serveur web et capturez le résultat comme nouvel AMI. Dans ce cas, vous êtes responsable des tâches administratives associées à l'exécution d'un cache.

Une autre option consiste à tirer parti d'[Amazon ElastiCache](#)¹³ et à éviter cette charge opérationnelle. Il s'agit d'un service géré qui facilite le déploiement, le fonctionnement et la mise à l'échelle d'un cache en mémoire distribué dans le cloud. L'un des moteurs de mise en cache pris en charge par ElastiCache est Memcached, qui, en tant que tel, peut être utilisé avec WordPress et un plug-in Memcached WordPress approprié sans autre personnalisation supplémentaire.

Mise en cache du bytecode

Chaque fois qu'un script PHP est exécuté, il est analysé et compilé. En utilisant un cache de bytecode PHP, la sortie de la compilation PHP est stockée dans la RAM, de telle sorte que le même script ne doit pas être recompilé indéfiniment. Le traitement lié à l'exécution de scripts PHP s'en trouve réduit et il en résulte de meilleures performances et des exigences d'UC moindres.

Un cache du bytecode peut être installé sur toute instance EC2 qui héberge WordPress et peut grandement réduire sa charge. Un cache du bytecode PHP répandu est [Alternative PHP Cache](#) ou APC,¹⁴ mais pour PHP 5.5 et version ultérieure, nous recommandons l'utilisation d'[OPcache](#) qui est une extension groupée avec cette version PHP.¹⁵

Montée en puissance

Même si un déploiement à une seule instance peut suffire dans certains cas, tout site web WordPress qui répond à un objectif professionnel ou autre important nécessite une architecture hautement disponible et aux meilleures performances. Avec AWS, vous lancez un tel environnement en un temps très bref.

Séparation des couches web et base de données

La première étape consiste à séparer les couches web et base de données (DB). En utilisant deux serveurs distincts, nous pouvons accroître la capacité de l'implémentation, mais bénéficier aussi d'un plus grand contrôle sur les caractéristiques de la configuration de chaque couche (web et base de données).

¹³ <http://aws.amazon.com/elasticache/>

¹⁴ <http://php.net/apc>

¹⁵ <http://php.net/manual/en/book.opcache.php>

Un serveur web et un serveur de base de données répondent généralement à des exigences différentes en termes d'UC, de mémoire, de stockage et de capacité de mise en réseau. Le scénario à un serveur fonctionne bien avec les types d'instance à usage général (T2, M3, par exemple), mais comme AWS fournit une plus grande variété de types d'instance Amazon EC2, vous pouvez optimiser la configuration serveur de chaque charge de travail à des fins de performance et de coût. Généralement parlant, l'instance C3 optimisée pour le calcul peut être un bon choix pour un serveur web WordPress, tandis qu'une instance R3 optimisée pour la mémoire peut se traduire par des performances de base de données supérieures en raison de la mémoire RAM disponible et de la façon dont elle peut être utilisée : par le moteur de stockage InnoDB, par exemple, pour son [pool de mémoires tampons](#).¹⁶

Lors du déplacement de la base de données vers sa propre infrastructure distincte, le fichier de configuration WordPress (`wp-config.php`) du serveur web doit simplement être reconfiguré pour pointer vers le nom d'hôte de l'instance de base de données distincte. Cette instance de base de données peut être une instance Amazon EC2 que vous gérez par vous-même et où vous avez installé MySQL. Dans ce cas, vous êtes responsable des tâches administratives associées à l'exécution d'un serveur de base de données. Une meilleure solution consiste à utiliser [Amazon Relational Database Service](#) (RDS).¹⁷

Amazon RDS est un service géré qui vous donne accès aux capacités d'un moteur de base de données MySQL tout en gérant les tâches d'administration de base de données gourmandes en temps ; vous pouvez ainsi vous concentrer sur vos applications et sur votre activité. Amazon RDS applique automatiquement les correctifs logiciels à la base de données, sauvegarde votre base de données, stocke les sauvegardes pendant une période définie par l'utilisateur et prend en charge la restauration à un instant dans le passé. Vous pouvez aussi dimensionner les ressources de calcul et ou la capacité de stockage de votre base de données à l'aide d'un seul appel d'API.

¹⁶ <http://dev.mysql.com/doc/refman/5.6/en/innodb-buffer-pool.html>

¹⁷ <http://aws.amazon.com/rds/>

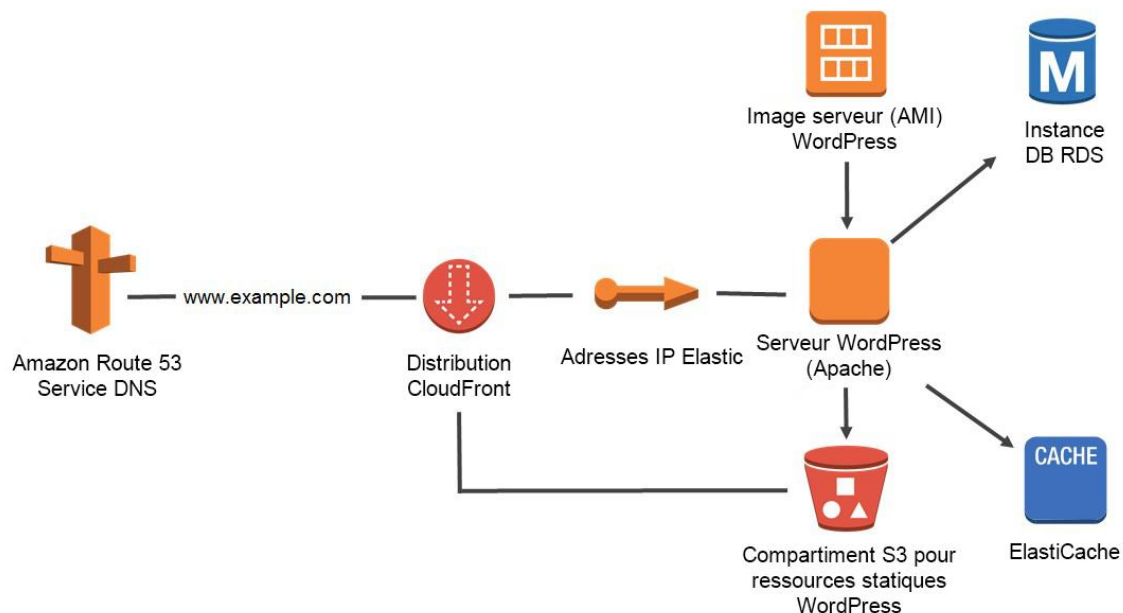


Figure 3 : Séparation des trois charges de travail (web, base de données et cache)

Disponibilité de la couche données

Les déploiements multi-AZ Amazon RDS fournissent une disponibilité et une durabilité améliorées des instances de base de données, ce qui en fait une solution naturelle pour les charges de travail de base de données en production. Lorsque vous provisionnez une instance DB multi-AZ, Amazon RDS crée automatiquement une instance DB principale et réplique de façon synchrone les données sur une instance de secours d'une autre zone de disponibilité (AZ). Les zones de disponibilité sont des emplacements physiquement distincts au sein d'une région AWS donnée. Elles fournissent une connectivité réseau à faible latence à d'autres zones de disponibilité de la même région. Chaque zone de disponibilité exécute sa propre infrastructure indépendante et physiquement distincte et est conçue pour être hautement fiable. En cas de défaillance de l'infrastructure, Amazon RDS bascule automatiquement vers la copie de secours afin que vous puissiez reprendre les opérations de base de données aussitôt le basculement terminé. Comme le point de terminaison de votre instance DB demeure le même après un basculement, votre application peut reprendre l'opération de base de données sans qu'une intervention manuelle administrative ne soit nécessaire.

En ce qui concerne la couche Memcached, ElastiCache remplace automatiquement les nœuds individuels en cas de défaillance. Pour améliorer encore plus la fiabilité du cluster, vous pouvez également tirer parti du modèle flexible de placement des nœuds pour ElastiCache. Avec cette configuration, les clusters de cache de Memcached de deux nœuds ou plus peuvent couvrir plusieurs zones de disponibilité d'une région.

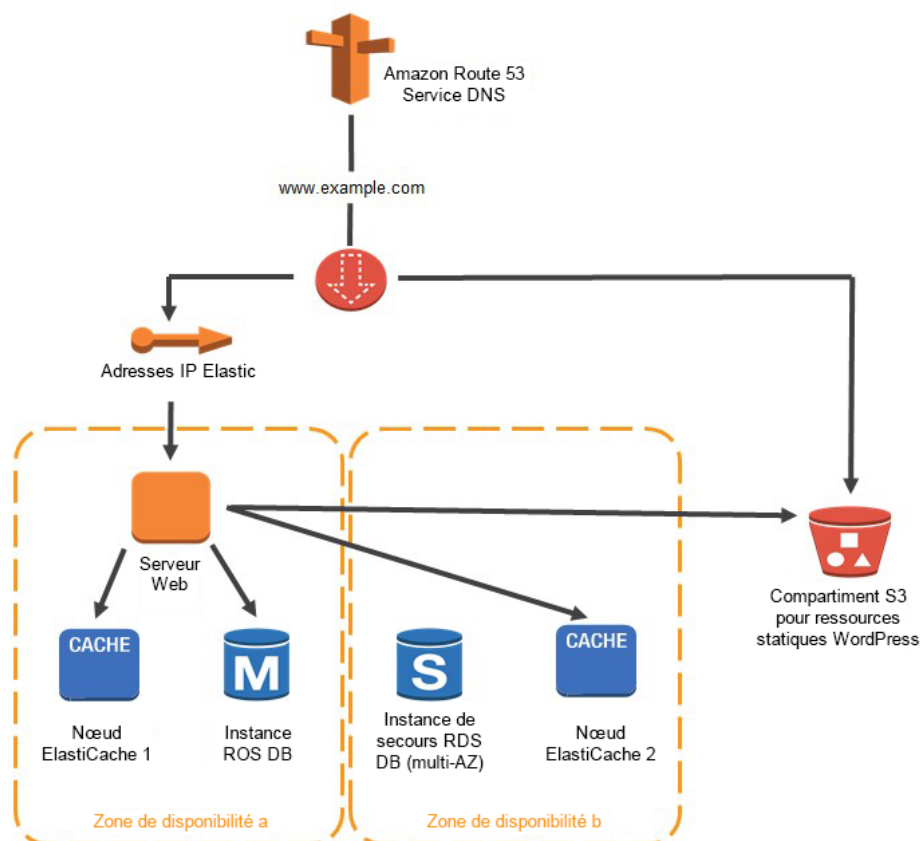


Figure 4 : Couche de données hautement disponible

Remarque sur Amazon RDS et la fonction multi-site de WordPress

WordPress propose une fonctionnalité multi-site avec laquelle vous pouvez alimenter un réseau de sites à partir d'une seule installation WordPress. Dans ce mode, tandis que certaines des tables de base de données demeurent partagées, chacun de ces sites ajoute aussi ses propres tables distinctes à la base de données. Si vous prévoyez d'alimenter des centaines de sites web depuis une seule instance de base de données, sachez que la bonne pratique pour les instances Amazon RDS MySQL consiste à créer plus de 10 000 tables à l'aide d'IOPS provisionnées ou de 1 000 tables à l'aide d'un stockage standard. Un grand nombre de tables augmente considérablement le temps de récupération après un basculement ou un incident de la base de données. Si vous devez réellement créer plus de tables que recommandé, définissez le paramètre `innodb_file_per_table` avec la valeur 0. Pour plus d'informations, consultez [Utilisation des espaces de table InnoDB pour améliorer les temps de récupération sur incident](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.MySQL.CommonDBATasks.html#Appendix.MySQL.CommonDBATasks.Tables)¹⁸ et [Utilisation des groupes de paramètres DB](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithParamGroups.html)¹⁹ dans le *Guide utilisateur Amazon RDS*.

¹⁸ <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.MySQL.CommonDBATasks.html#Appendix.MySQL.CommonDBATasks.Tables>

¹⁹ http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithParamGroups.html

Evolutivité et disponibilité de la couche web

Une fois que vous avez créé une fondation solide avec une couche données hautement disponible, la préoccupation suivante concerne la disponibilité et l'évolutivité de votre capacité de service web.

Haute disponibilité pour la couche web

Pour prendre en charge la haute disponibilité pour les serveurs web qui alimentent un site web WordPress, vous devez déployer plus d'un nœud de telle sorte que votre application puisse supporter une défaillance du serveur web. Tout comme ce qui est décrit dans [Disponibilité de la couche données](#), vous pouvez déployer ces instances dans les zones de disponibilité distinctes d'une région pour augmenter la fiabilité de l'architecture globale.

Pour répartir les demandes des utilisateurs finaux sur plusieurs nœuds de serveur web, vous avez besoin d'une solution d'équilibrage de charge. AWS fournit cette capacité via [Elastic Load Balancing](#), service hautement disponible qui répartir le trafic sur plusieurs instances Amazon EC2.²⁰

Elastic Load Balancing prend en charge la distribution des demandes entre plusieurs zones de disponibilité au sein d'une région AWS. Vous pouvez aussi configurer une vérification de l'état de telle sorte que le répartiteur de charge Elastic Load Balancing arrête automatiquement l'envoi du trafic aux instances qui ont défailli (suite à un incident matériel ou un problème logiciel, par exemple). Pour plus d'informations, consultez [Vérification de l'état](#) dans le *Guide du développeur Elastic Load Balancing*.²¹

Evolutivité de la couche web

Autre caractéristique majeure de la plateforme AWS, son élasticité. Vous pouvez lancer une plus grande capacité de calcul (serveurs web, par exemple) quand vous en avez besoin ou une capacité moindre dans le cas contraire. Comme vous ne payez que pour ce que vous utilisez et qu'un sur-approvisionnement n'est pas nécessaire, vous pouvez optimiser les coûts. Vous réduisez le risque de sous-provisionner dans la mesure où vous n'avez pas besoin de deviner la capacité requise exacte dont vous avez besoin aux heures de pic.

²⁰ <http://aws.amazon.com/elasticloadbalancing/>

²¹ <http://docs.aws.amazon.com/ElasticLoadBalancing/latest/DeveloperGuide/TerminologyandKeyConcepts.html#healthcheck>

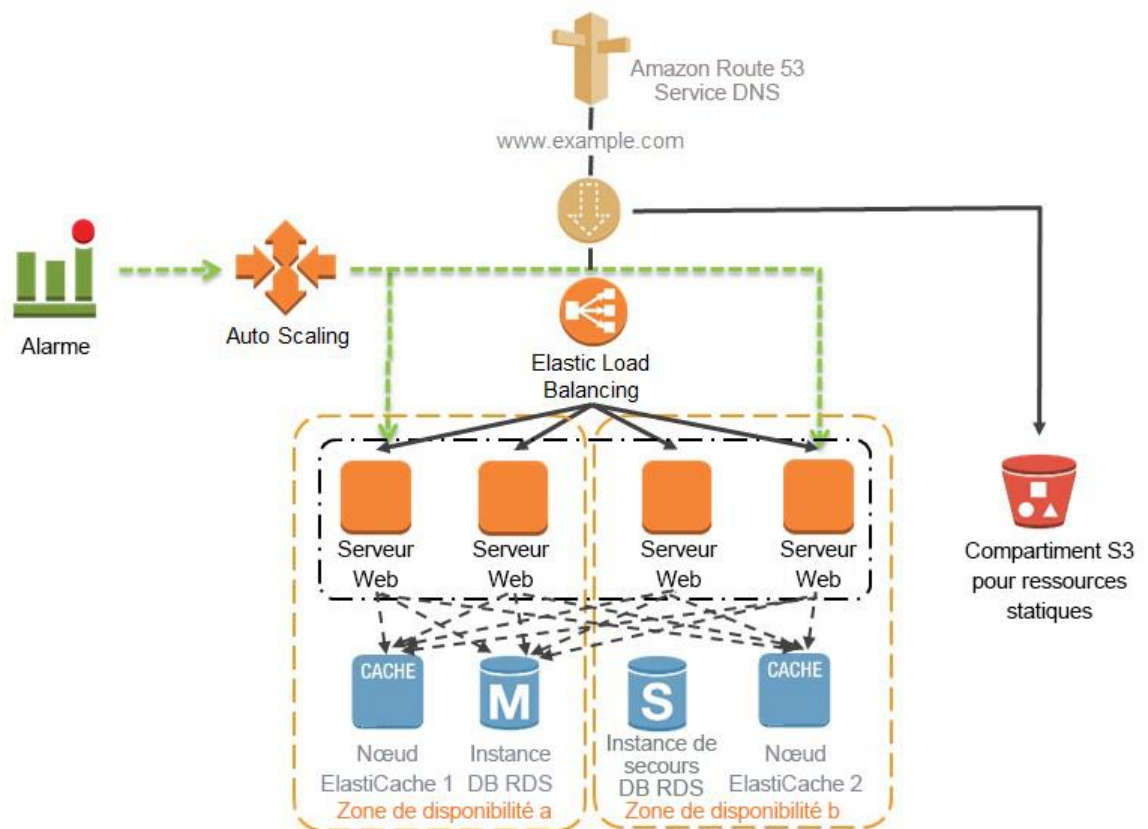


Figure 5 : Haute disponibilité et évolutivité de la couche web

[Auto Scaling](#) est un service AWS qui vous aide à automatiser ce provisionnement pour augmenter ou réduire la capacité Amazon EC2 selon les conditions que vous définissez sans qu'une intervention manuelle ne soit nécessaire.²² Vous pouvez configurer Auto Scaling de telle sorte que le nombre d'instances Amazon EC2 que vous utilisez augmente de façon transparente pendant les pics pour maintenir les performances et diminue automatiquement lorsque le trafic se réduit de façon à minimiser les coûts.

Le service Elastic Load Balancing prend en charge l'ajout et la suppression dynamiques des hôtes Amazon EC2 depuis la rotation de l'équilibrage de charge. Le service Elastic Load Balancing peut aussi augmenter et réduire dynamiquement la capacité d'équilibrage de charge pour s'adapter à la demande en terme de trafic, et ce sans intervention manuelle.

Une architecture sans état

Pour tirer parti de plusieurs serveurs web dans une configuration Auto Scaling, votre couche web doit être sans état. Cela signifie que les données qui doivent être conservées pendant plus d'une seule demande HTTP ne sont pas stockées sur les serveurs web. La raison en est que les instances existantes prennent automatiquement fin quand la configuration Auto Scaling indique qu'une partie de la capacité provisionnée n'est plus nécessaire. Lorsque cela se produit, les données qui n'ont pas été stockées en dehors de cette instance sont perdues.

²² <http://aws.amazon.com/autoscaling/>

En stockant tout type de données côté client (les cookies, par exemple) ou dans un stockage durable partagé (une base de données Amazon RDS multi-AZ ou Amazon S3, par exemple) au lieu du système de fichiers local, vous avez la possibilité de lancer simplement plusieurs instances Amazon EC2 lorsque vous devez augmenter la capacité et de mettre fin à l'une d'entre elles lorsque vous avez un surplus de capacité. Vous évitez aussi d'avoir à configurer tout mécanisme complexe pour synchroniser les données entre plusieurs serveurs web. De plus, la défaillance d'une seule instance de serveur web (suite à un problème matériel, par exemple) n'impacte pas l'intégrité de vos données, car les données cruciales ne sont pas stockées localement. Pour la plupart des applications web, cet aspect se répartit en deux zones : d'un côté, les sessions utilisateur, de l'autre, les données générées par l'utilisateur et autres chargements.

Sessions utilisateur

Contrairement à la plupart des autres applications web, le cœur de WordPress est entièrement sans état en ce qui concerne le stockage des données de session. Il s'appuie à la place sur les cookies. Par exemple, les utilisateurs connectés sont identifiés via un cookie d'authentification dont la validité est contrôlée par rapport à un enregistrement stocké dans la base de données. En tant que tel, un stockage de session ne constitue pas un problème à moins que vous n'ayez installé un code personnalisé (un plug-in WordPress, par exemple) qui repose à la place sur les sessions PHP natives. Dans ce cas, Amazon DynamoDB est la meilleure option pour un stockage rapide, durable et évolutif des sessions PHP. Pour plus d'informations, consultez l'article de Jeremy Lindblom, [Scalable Session Handling with DynamoDB](#).²³

Données générées par l'utilisateur et autres chargements

Par défaut, comme WordPress stocke les chargements utilisateur sur le système de fichiers, il s'agit d'une zone où WordPress ne possède pas une architecture sans état. Nous avons déjà expliqué comment l'utilisation d'un plug-in pour stocker ces fichiers dans Amazon S3 et le service depuis CloudFront aide à réduire la charge serveur et à améliorer l'expérience utilisateur final. Un autre avantage est que la fonction est déplacée vers une couche distincte et que la couche web est rendue sans état.

Dimensionnement de la couche données

Même avec la mise en cache, il se peut que vous ayez besoin d'une capacité de base de données supérieure. Si tel est le cas, vous pouvez facilement basculer vers un type d'instance Amazon RDS plus grand et/ou utiliser le stockage IOPS provisionnées pour des performances disque supérieures et cohérentes. Pour plus d'informations, consultez [Classe d'instance DB](#)²⁴ et [Stockage IOPS provisionnées](#)²⁵ dans le *Guide de l'utilisateur Amazon Relational Database*.

²³ <http://aws.amazon.com/blogs/aws/scalable-session-handling-in-php-using-amazon-dynamodb/>

²⁴ <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.DBInstanceClass.html>

²⁵ http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Storage.html#USER_PIOPS

Comme les déploiements WordPress sont généralement des charges de travail à lecture intensive, une solution d'évolution plus efficace consiste à tirer parti de la réplication MySQL en lançant un ou plusieurs réplicas en lecture Amazon RDS. Il s'agit de copies en lecture de votre base de données qui facilitent le dimensionnement au-delà de la capacité de déploiement d'une seule base de données. Consultez [Utilisation des réplicas en lecture PostgreSQL et MySQL](#) dans la documentation Amazon RDS.²⁶

Prêt à l'emploi, WordPress n'est pas conçu pour tirer parti de plusieurs instances de base de données et, par conséquent, vous devez l'étendre avec un plug-in adapté. Par exemple, le [plug-in HyperDB](#)²⁷ de WordPress, qui remplace la classe de base de données WordPress standard et ajoute la possibilité d'utiliser plusieurs instances de base de données.²⁸

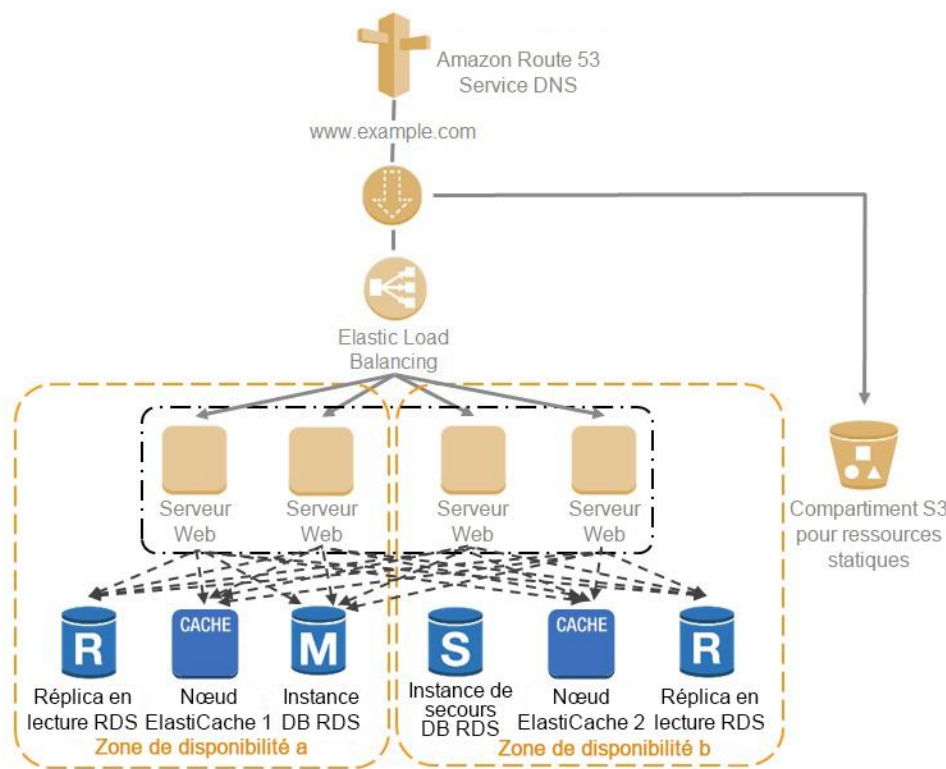


Figure 6 : Dimensionnement de la base de données avec réplicas en lecture RDS

²⁶ http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html

²⁷ <https://wordpress.org/plugins/hyperdb/>

²⁸ A la date de rédaction de ce document, HyperDB et le W3 Total Cache mentionné précédemment ne sont pas compatibles entre eux à la première utilisation.

Considérations sur le développement et le déploiement

Peu de sites web demeurent statiques. Dans la plupart des cas, vous ajouterez des thèmes ou des plug-ins WordPress publiquement disponibles ou procéderez à une mise à niveau vers une version WordPress plus récente, et ce régulièrement. Dans d'autres cas, vous développerez intégralement vos propres plug-ins et thèmes personnalisés.

Réduction du risque de modification

Chaque fois que vous procédez à une modification structurelle de votre installation WordPress, le risque existe que vous introduisiez des problèmes imprévus. A tout le moins, vous devez effectuer une sauvegarde du code de votre application, de votre configuration et de votre base de données avant d'appliquer une modification importante (installation d'un nouveau plug-in, par exemple). Pour les sites web ayant une valeur commerciale ou autre, vous devrez certainement tester d'abord ces modifications sur un environnement intermédiaire distinct. Avec AWS, il est très facile de répliquer la configuration de votre environnement de production et d'exécuter la totalité du processus de déploiement de manière sécurisée. Une fois que vous avez terminé vos tests, vous pouvez simplement défaire votre environnement de test et cesser de payer ces ressources. Avant de débattre de certaines considérations propres à WordPress, si vous souhaitez plus d'informations sur les bonnes pratiques de développement et de test sur AWS, consultez le livre blanc [Développement et test sur Amazon Web Services](http://aws.amazon.com/whitepapers/dev-test-on-aws/).²⁹

Déploiement de nouveaux plug-ins

En raison de la façon dont l'installation du plug-in WordPress se déroule, une certaine planification est souvent requise. A l'activation, certains plug-ins WordPress écrivent les informations de configuration sur la table de base de données wp_options (ou introduisent les modifications du schéma DB). Certains plug-ins créent les fichiers de configuration dans le système de fichiers local. Dans un environnement multi-serveur, cela peut constituer un défi parce que ces fichiers ne sont créés que pour une seule des instances en cours d'exécution. Une solution simple de ce problème consiste à exécuter d'abord l'installation du plug-in sur un serveur intermédiaire. Toutes les modifications du système de fichiers devront être fusionnées avec le code de votre application et introduites comme nouvelle version du code de l'application. Vous utiliserez ensuite votre méthode de déploiement standard pour vous assurer que ces modifications sont envoyées sur tous vos serveurs web.

Déploiement des modifications de thème

Lors du déploiement des modifications de thème, si vous utilisez Amazon S3 pour le stockage des ressources (JavaScript, feuille de style et fichiers image), une procédure est nécessaire pour les copier sur l'emplacement de compartiment approprié. Les plug-ins comme W3 Total Cache offrent un moyen pour que vous démarriez manuellement cette tâche. Une autre solution consiste à optimiser cette étape comme partie du processus d'élaboration.

²⁹ <http://aws.amazon.com/whitepapers/dev-test-on-aws/>

Comme ces ressources peuvent être mises en cache sur CloudFront et le navigateur, vous avez besoin d'un moyen d'invalider les versions plus anciennes lorsque vous déployez les modifications. Le meilleur moyen d'y parvenir consiste à inclure une sorte d'identificateur de version dans votre objet. Cet identificateur peut être une chaîne de requête avec un horodatage ou une chaîne aléatoire. Si vous utilisez le plug-in W3 Total Cache, vous pouvez mettre à jour une chaîne de requête de média que vous ajoutez à l'URL des fichiers multimédias.

Conception réactive

Tandis que l'utilisation des smartphones et des tablettes ne cesse de croître, il est important que vous optimisiez l'expérience de consultation de site web pour une grande variété d'appareils. Vous pouvez y parvenir en combinant différentes solutions :

- S'appuyer sur la logique client (requêtes de média CSS3, disposition fluide, etc.) pour adapter la présentation sans modifier le contenu HTML lui-même.
- Utiliser un plug-in pour basculer le thème WordPress actif sur l'appareil de l'utilisateur.
- Utiliser la logique conditionnelle au sein des fichiers modèles du thème WordPress

Même si vous pouvez améliorer l'expérience utilisateur simplement avec la logique côté client, vous disposez d'un bien plus grand contrôle quand vous pouvez réellement modifier la sortie HTML côté serveur. Vous pouvez utiliser l'en-tête HTTP User-Agent pour identifier et différencier les appareils bureau et les appareils mobiles, et pour fournir un contenu adapté et approprié à chacun d'eux. Si vous utilisez CloudFront pour servir un contenu dynamique, vous pouvez à la place utiliser ses en-têtes de détection de type d'appareil (CloudFront est une visionneuse pour appareils mobiles, CloudFront est une visionneuse pour ordinateurs de bureau et CloudFront est une visionneuse pour tablettes). Une bonne approche consiste à utiliser ces en-têtes de niveau supérieur qui varient beaucoup moins quant à leurs valeurs possibles au lieu de repasser l'en-tête User-Agent à l'origine. Il en résulte un meilleur taux de réussite du cache et une réduction du nombre de demandes qui accèdent à vos serveur d'origine.

L'exemple suivant illustre un extrait d'un fichier modèle WordPress, qui utilise les en-têtes CloudFront pour retourner de façon conditionnelle une image dimensionnée différemment en fonction du type d'appareil. Cela réduit l'utilisation de la bande passante superflue et améliore la vitesse de chargement de la page :

```
else

<div class="entry-thumbnail">
<?php if
($_SERVER['HTTP_CLOUDFRONT_IS_DESKTOP_VIEWER']=='true')
    {the_post_thumbnail('large');}

else if
($_SERVER['HTTP_CLOUDFRONT_IS_TABLET_VIEWER']=='true')
    {the_post_thumbnail('medium');}

else if
($_SERVER['HTTP_CLOUDFRONT_IS_MOBILE_VIEWER']=='true')
    {the_post_thumbnail('thumbnail');}

    {the_post_thumbnail('large');}?>

</div> <?php endif; ?>
```

Récapitulatif

AWS propose de nombreuses options d'architecture pour l'exécution de WordPress. L'option la plus simple est l'installation d'un seul serveur pour les sites web à faible trafic. Pour les sites web avancés, les administrateurs de site peuvent ajouter plusieurs autres options, chacune représentant une amélioration incrémentielle en termes de disponibilité et d'évolutivité. Les administrateurs peuvent sélectionner les fonctionnalités qui correspondent le plus étroitement à leurs exigences et à leur budget. Pour plus de détails sur la façon d'implémenter une telle architecture sur AWS, reportez-vous au livre blanc [Déploiement de WordPress avec AWS Elastic Beanstalk](#).