



Whitepaper da AWS

# Práticas recomendadas para o WordPress na AWS



# Práticas recomendadas para o WordPress na AWS: Whitepaper da AWS

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

As marcas comerciais e o visual comercial da Amazon não podem ser usados em conexão com nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa causar confusão entre os clientes ou que deprecie ou desacredite a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, conectados ou patrocinados pela Amazon.

---

# Table of Contents

Resumo .....	1
Resumo .....	1
Introdução .....	2
Implantação simples .....	3
Considerações .....	3
Abordagens disponíveis .....	3
Amazon Lightsail .....	4
Selecionar um plano de preços do Amazon Lightsail .....	4
Instalação do WordPress .....	5
Recuperação após falhas .....	5
Melhoria da performance e da eficiência de custo .....	7
Acelerar a entrega de conteúdo .....	7
Descarregamento de conteúdo estático .....	8
Conteúdo dinâmico .....	8
Armazenamento em cache de banco de dados .....	10
Armazenamento em cache de bytecode .....	10
Implantação elástica .....	12
Arquitetura de referência .....	12
Escalabilidade da camada da Web .....	14
Camada da Web sem estado .....	15
Armazenamento compartilhado (Amazon S3 e Amazon EFS) .....	16
Camada de dados (Amazon Aurora e Amazon ElastiCache) .....	17
WordPress High Availability by Bitnami no Quick Start da AWS .....	18
Conclusão .....	21
Colaboradores .....	22
Revisões do documento .....	23
Apêndice A: configuração do CloudFront .....	24
Origens e comportamentos .....	24
Criação de distribuição do CloudFront .....	24
Apêndice B: instalação e configuração de plugins .....	28
Plugin AWS for WordPress .....	28
Instalação e configuração de plugins .....	28
Amazon CloudFront e AWS Certificate Manager .....	30
Traduzir e vocalizar o conteúdo .....	30

---

Podcasts com o Amazon Pollycast .....	33
Ler o conteúdo por meio de dispositivos Amazon Alexa .....	33
Configuração de conteúdo estático .....	34
Criação de usuário do IAM .....	35
Criação de bucket do Amazon S3 .....	35
Criação de uma origem estática .....	36
Apêndice C: backup e recuperação .....	38
Apêndice D: implantar novos plugins e temas .....	40
Avisos .....	41

# Práticas recomendadas para o WordPress na AWS

Data de publicação: 19 de outubro de 2021 ([Revisões do documento](#))

## Resumo

Este whitepaper fornece aos administradores de sistema orientações específicas sobre os conceitos básicos do WordPress na Amazon Web Services (AWS) e como melhorar a eficiência de custos da implantação e a experiência do usuário final. Também descreve a arquitetura de referência que aborda os requisitos comuns de escalabilidade e de alta disponibilidade.

# Introdução

O WordPress é uma ferramenta de blogging e um sistema de gerenciamento de conteúdo (CMS) de código aberto baseado em PHP e MySQL, usado em diversos tipos de sites, de blogs pessoais a sites de alto volume de tráfego.

Quando a primeira versão do WordPress foi lançada em 2003, ela não foi criada considerando as infraestruturas modernas baseadas na nuvem escaláveis e elásticas. Com o trabalho da comunidade do WordPress e o lançamento de diversos módulos do WordPress, os recursos dessa solução de CMS se expandem constantemente. Hoje, é possível criar uma arquitetura do WordPress que aproveite muitos dos benefícios da Nuvem AWS.

# Implantação simples

Para blogs ou sites de tráfego baixo sem requisitos rigorosos de alta disponibilidade, pode ser adequada uma implantação simples de um único servidor. Essa implantação não é a arquitetura mais resiliente ou escalável, mas é a forma mais rápida e econômica de fazer o site funcionar bem.

## Tópicos

- [Considerações](#)
- [Abordagens disponíveis](#)
- [Amazon Lightsail](#)

## Considerações

Esta discussão começa com uma única implantação de servidor Web. Há momentos de você querer expandi-lo, como:

- A máquina virtual em que o site do WordPress está implantado é um ponto de falha único. Um problema com essa instância gera uma perda de serviço do site.
- A escalabilidade de recursos para melhorar a performance só pode ser alcançada pela “escalabilidade vertical”; ou seja, aumentando o tamanho da máquina virtual que executa o site do WordPress.

## Abordagens disponíveis

A AWS tem diversas opções para provisionar máquinas virtuais. Há três maneiras diferentes de hospedar seu próprio site do WordPress na AWS:

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

O [Amazon Lightsail](#) é um serviço que permite executar rapidamente um servidor virtual privado (uma instância do Lightsail) para hospedar um site do WordPress. O Lightsail é a maneira mais fácil de começar se você não precisa de tipos de instância altamente configuráveis ou acesso a recursos avançados de redes.

O [Amazon EC2](#) é um serviço da Web que fornece capacidade computacional redimensionável para que você possa executar um servidor virtual em minutos. O Amazon EC2 oferece mais opções de configuração e gerenciamento em comparação ao Lightsail, o que é desejável em arquiteturas mais avançadas. Você tem acesso administrativo às instâncias do EC2 e pode instalar qualquer pacote de software de sua preferência, inclusive o WordPress.

O [AWS Marketplace](#) é uma loja online onde você pode encontrar, comprar e implantar rapidamente um software executado na AWS. Você pode usar a implantação do 1-Click para executar imagens pré-configuradas do WordPress diretamente no Amazon EC2 em sua própria conta da AWS, em questão de minutos. Existem vários fornecedores do Marketplace que oferecem instâncias do WordPress prontas para execução.

Este whitepaper aborda a opção do Lightsail como implementação recomendada para um site do WordPress de servidor único.

## Amazon Lightsail

O Lightsail é a forma mais fácil de começar a usar a AWS para desenvolvedores, pequenas empresas, estudantes e outros usuários que precisam de uma solução simples de servidor privado virtual (VPS).

O serviço abstrai do usuário muitos dos elementos complexos do gerenciamento de infraestrutura. Portanto, ele é um ponto de partida ideal para quando você tem menos experiência com infraestrutura ou precisa se concentrar em executar seu site, e um produto simplificado é ideal para suas necessidades.

Com o Amazon Lightsail, é possível escolher o sistema operacional Windows ou Linux/Unix e aplicações Web populares, como o WordPress, e implantá-los com um único clique em modelos pré-configurados.

Conforme suas necessidades aumentam, você tem mais capacidade de avançar as barreiras iniciais e se conectar a serviços adicionais de banco de dados da AWS, armazenamento de objetos, armazenamento em cache e distribuição de conteúdo.

## Selecionar um plano de preços do Amazon Lightsail

Um [plano do Lightsail](#) define o custo mensal dos recursos do Lightsail que você usa para hospedar o site do WordPress. Há vários planos disponíveis que abrangem uma variedade de casos de uso, com níveis variados de recursos de CPU, memória, armazenamento em unidade de estado sólido



(SSD) e transferência de dados. Se o site é complexo, será necessário uma instância maior, com mais recursos. Isso acontece pela migração do servidor para um plano maior que [use o console da Web](#) ou conforme descrito na [documentação da CLI do Amazon Lightsail](#).

## Instalação do WordPress

O Lightsail conta com modelos para aplicações comumente usadas, como o WordPress. Esse modelo é um bom ponto de partida para executar seu próprio site do WordPress, pois ele vem pré-instalado com a maioria dos programas de software necessários. Você pode instalar outros programas de software ou personalizar a configuração do software usando o terminal no navegador ou seu próprio cliente SSH, ou pela interface Web de administração do WordPress.

O Amazon Lightsail tem uma parceria com o produto Pro Sites da GoDaddy que visa ajudar clientes do WordPress a gerenciar facilmente as instâncias sem custo adicional. Os servidores virtuais do WordPress no Lightsail são pré-configurados e otimizados para proporcionar performance e segurança rapidamente, permitindo que você coloque em operação um site do WordPress com rapidez e facilidade. Os clientes que executam várias instâncias do WordPress acham as atividades de atualização, manutenção e gerenciamento de todos os sites demoradas e desafiadoras. Com essa integração, é possível gerenciar facilmente as várias instâncias do WordPress em minutos com apenas alguns cliques.

Para obter mais informações sobre o gerenciamento do WordPress no Lightsail, consulte [Conceitos básicos sobre o uso do WordPress na instância do Amazon Lightsail](#). Recomendamos tirar um snapshot da instância quando terminar de personalizar o site do WordPress.

Um [snapshot](#) é uma forma de criar uma imagem de backup da instância do Lightsail. Ele é uma cópia do disco do sistema e também armazena a configuração original da máquina (ou seja, memória, CPU, tamanho do disco e taxa de transferência de dados). Os snapshots podem ser usados para reverter a uma configuração correta conhecida depois de uma implantação ou atualização que não deu certo.

Esse snapshot permite que você recupere o servidor, se necessário, mas também execute novas instâncias com as mesmas personalizações.

## Recuperação após falhas

Servidor Web único é um ponto de falha exclusivo, por isso você precisa saber se os dados do site estão copiados. O mecanismo de snapshot descrito anteriormente também pode ser usado para essa finalidade. Para a recuperação após falha, você pode restaurar uma nova instância do snapshot

mais recente. Para reduzir a quantidade de dados que pode ser perdida durante uma restauração, seus snapshots devem ser os mais recentes possíveis.

Para minimizar o potencial de perda de dados, verifique se os snapshots estão sendo tirados com regularidade. É possível programar snapshots automáticos das instâncias Linux/Unix do Lightsail. Para obter as etapas, consulte [Ativação ou desativação de snapshots automáticos para instâncias ou discos no Amazon Lightsail](#).

A AWS recomenda que você use um IP estático: um endereço IP público fixo que seja dedicado à conta do Lightsail. Se você precisar substituir a instância por outra, poderá reatribuir o IP estático para a nova instância. Dessa forma, você não precisa reconfigurar todos os sistemas externos (como registros DNS) para apontar para um novo endereço IP cada vez que desejar substituir sua instância.

# Melhoria da performance e da eficiência de custo

Você pode querer expandir sua implantação de servidor único. Neste caso, é necessário considerar opções para melhorar a performance do site. Antes de migrar para uma implantação escalável e de vários servidores (abordaremos isso posteriormente neste artigo), há muitas eficiências de performance e custo que podem ser aplicadas. Essas são boas práticas que devem ser seguidas em todos os casos, mesmo se for migrar para uma arquitetura com vários servidores.

As seções a seguir apresentam diversas opções capazes de melhorar aspectos de performance e escalabilidade do site do WordPress. Algumas podem ser aplicadas a uma implantação de servidor único, enquanto outras aproveitam a escalabilidade de diversos servidores. Muitas dessas modificações exigem o uso de um ou mais plugins do WordPress. Embora haja várias opções disponíveis, o [W3 Total Cache](#) é uma opção popular que combina muitas dessas modificações em um único plugin.

## Tópicos

- [Acelerar a entrega de conteúdo](#)
- [Armazenamento em cache de banco de dados](#)
- [Armazenamento em cache de bytecode](#)

## Acelerar a entrega de conteúdo

Qualquer site do WordPress precisa entregar uma mistura de conteúdo estático e dinâmico. O conteúdo estático inclui imagens, arquivos do JavaScript ou folhas de estilo. O conteúdo dinâmico inclui tudo o que é gerado do lado do servidor que use código PHP do WordPress, como, por exemplo, elementos do site que sejam gerados do banco de dados ou personalizados para cada visualizador.

Um aspecto importante da experiência do usuário final é a latência de rede envolvida na entrega do conteúdo anterior para usuários do mundo todo. Acelerar a entrega do conteúdo anterior melhora a experiência do usuário final, principalmente de usuários distribuídos geograficamente pelo globo. Isso pode ser feito com uma Rede de entrega de conteúdo (CDN), como o Amazon CloudFront.

O [Amazon CloudFront](#) é um serviço da Web que oferece uma maneira fácil e econômica de distribuir conteúdo com baixa latência e altas velocidades de transferência de dados por meio de vários locais da borda em todo o mundo. As solicitações do visualizador são encaminhadas automaticamente

para um [local da borda](#) adequado do CloudFront a fim de reduzir a latência. Se o conteúdo puder ser armazenado em cache (por alguns segundos, minutos ou até dias) e já estiver armazenado em um local da borda específico, o CloudFront o entregará imediatamente. Se o conteúdo não for armazenado em cache, ou não estiver no local da borda, o CloudFront o recuperará de uma ou mais fontes de confiabilidade, referidas como origens (neste caso, a instância do Lightsail) na configuração do CloudFront. Essa recuperação acontece em conexões de rede otimizadas, que funcionam para agilizar a entrega de conteúdo do seu site. Além de aprimorar a experiência do usuário final, o modelo abordado também reduz a carga nos servidores de origem e tem potencial para gerar uma boa economia de custo.

## Descarregamento de conteúdo estático

Inclui CSS, JavaScript e arquivos de imagem – tanto o s que fazem parte de seus temas do WordPress como os arquivos de mídia carregados por administradores de conteúdo. Todos esses arquivos podem ser armazenados no Amazon Simple Storage Service (Amazon S3) com o uso de um plugin, como o W3 Total Cache, e entregues aos usuários de maneira escalável e altamente disponível. O [Amazon S3](#) oferece uma infraestrutura de armazenamento de dados altamente escalável, confiável e de baixa latência com baixo custo, acessível por meio de APIs REST. O Amazon S3 armazena os objetos de forma redundante, não apenas em vários dispositivos, mas também em várias instalações em uma região AWS, proporcionando níveis excepcionalmente altos de durabilidade.

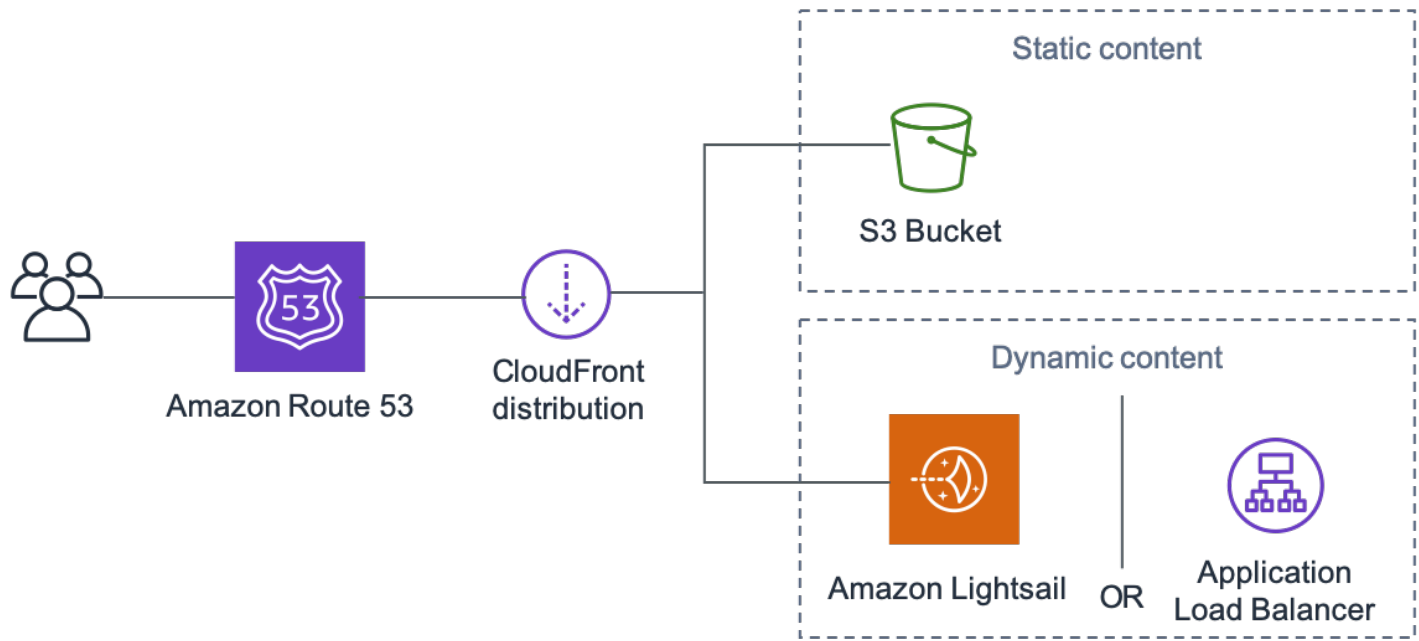
Isso produz o efeito colateral positivo por descarregar essa workload de sua instância do Lightsail e permitir um foco sobre a geração de conteúdo dinâmico. Isso reduz a carga no servidor e é uma etapa importante para a criação de uma arquitetura sem estado (um pré-requisito antes de implementar a escalabilidade automática).

Depois você pode configurar o Amazon S3 como origem do CloudFront para melhorar a entrega desses ativos estáticos a usuários do mundo todo. Embora o WordPress não seja integrado ao Amazon S3 e ao CloudFront imediatamente, diversos plugins oferecem suporte a esses serviços (por exemplo, W3 Total Cache).

## Conteúdo dinâmico

O conteúdo dinâmico inclui o resultado de scripts PHP do WordPress do lado do servidor. O conteúdo dinâmico também pode ser entregue pelo CloudFront com a configuração do site do WordPress como origem. Como o conteúdo dinâmico inclui conteúdo personalizado, é necessário configurar o CloudFront para encaminhar determinados cookies HTTP e cabeçalhos HTTP como

parte de uma solicitação ao servidor de origem personalizado. O CloudFront usa os valores de cookie encaminhados como parte da chave que identifica um objeto exclusivo em seu cache. Para garantir que você maximize a eficiência do caching, configure o CloudFront para encaminhar somente os cookies HTTP e cabeçalhos HTTP que realmente variam o conteúdo (não cookies que são usados somente no lado do cliente ou por aplicações de terceiros, como análises da Web, por exemplo).



### Entrega de todo o site pelo Amazon CloudFront

A figura anterior inclui duas origens: uma para conteúdo estático e outra para conteúdo dinâmico. Para obter detalhes de implementação, consulte o [Apêndice A: configuração do CloudFront](#) e o [Apêndice B: instalação e configuração de plugins](#).

O CloudFront usa cabeçalhos de controle de cache padrão para identificar se e por quanto tempo ele deve armazenar respostas HTTP específicas. Os mesmos cabeçalhos de controle de cache são usados por navegadores da Web para resolver quando e por quanto tempo armazenar conteúdo localmente para melhor experiência do usuário final (por exemplo, um arquivo `.css` que já foi baixado não será baixado novamente toda vez que um visitante recorrente visualizar uma página). É possível configurar cabeçalhos de controle de cache no nível do servidor Web (por exemplo, por meio de arquivos `.htaccess` ou modificações do arquivo `httpd.conf`) ou instalar um plugin do WordPress (por exemplo, W3 Total Cache) a fim de ditar como esses cabeçalhos são definidos para conteúdo estático e dinâmico.

## Armazenamento em cache de banco de dados

O armazenamento em cache do banco de dados pode reduzir bastante a latência e aumentar a taxa de transferência de workloads de aplicações com muita leitura, como o WordPress. A performance da aplicação é aprimorada com o armazenamento de partes de dados acessadas com frequência na memória para acesso de baixa latência (por exemplo, os resultados das consultas de bancos de dados com alto consumo de E/S). Quando uma grande porcentagem das consultas é disponibilizada pelo cache, a quantidade de consultas que precisam atingir o banco de dados é reduzida, resultando em um custo menor, associado à execução do banco de dados.

Embora o WordPress tenha recursos de cache limitados prontos para uso, há uma variedade de plugins compatíveis com a integração com o [Memcached](#), um sistema de cache de objetos de memória amplamente adotado. O plugin W3 Total Cache é um bom exemplo.

Nos cenários mais simples, você instala o Memcached em seu servidor Web e captura os resultados como um novo snapshot. Neste caso, você é responsável pelas tarefas administrativas associadas à execução de um cache.

Outra opção é aproveitar um managed service, como o [Amazon ElastiCache](#), e evitar essa carga operacional. O ElastiCache facilita a implantação, a operação e o dimensionamento de um cache em memória distribuído na nuvem. Você encontra informações sobre como se conectar aos nós de cluster do ElastiCache na [documentação do Amazon ElastiCache](#).

Caso você use o Lightsail e queira acessar um cluster do ElastiCache privadamente em sua conta da AWS, use o peering de VPC. Para obter instruções sobre como ativar o peering de VPC, consulte [Configurar o peering da Amazon VPC para trabalhar com recursos da AWS fora do Amazon Lightsail](#).

## Armazenamento em cache de bytecode

Sempre que um script PHP é executado, ele é analisado e compilado. Com o uso de um cache de bytecode PHP, o resultado da compilação de PHP é armazenado em RAM para que o mesmo script não seja compilado várias vezes. Isso reduz a sobrecarga relacionada à execução de scripts PHP, gerando melhor performance e menos requisitos de CPU.

Um cache de bytecode pode ser instalado em qualquer instância do Lightsail que hospede o WordPress e pode reduzir muito sua carga. Para o PHP 5.5 e posteriores, a AWS recomenda o uso do [OPcache](#), uma extensão em pacote com essa versão do PHP.

Observe que o OPcache é ativado por padrão no modelo do Bitnami WordPress Lightsail, portanto, nenhuma ação adicional é necessária.

# Implantação elástica

Há muitos cenários em que uma implantação de servidor único não é suficiente para seu site. Nessas situações, é necessário uma arquitetura escalável de vários servidores.

## Tópicos

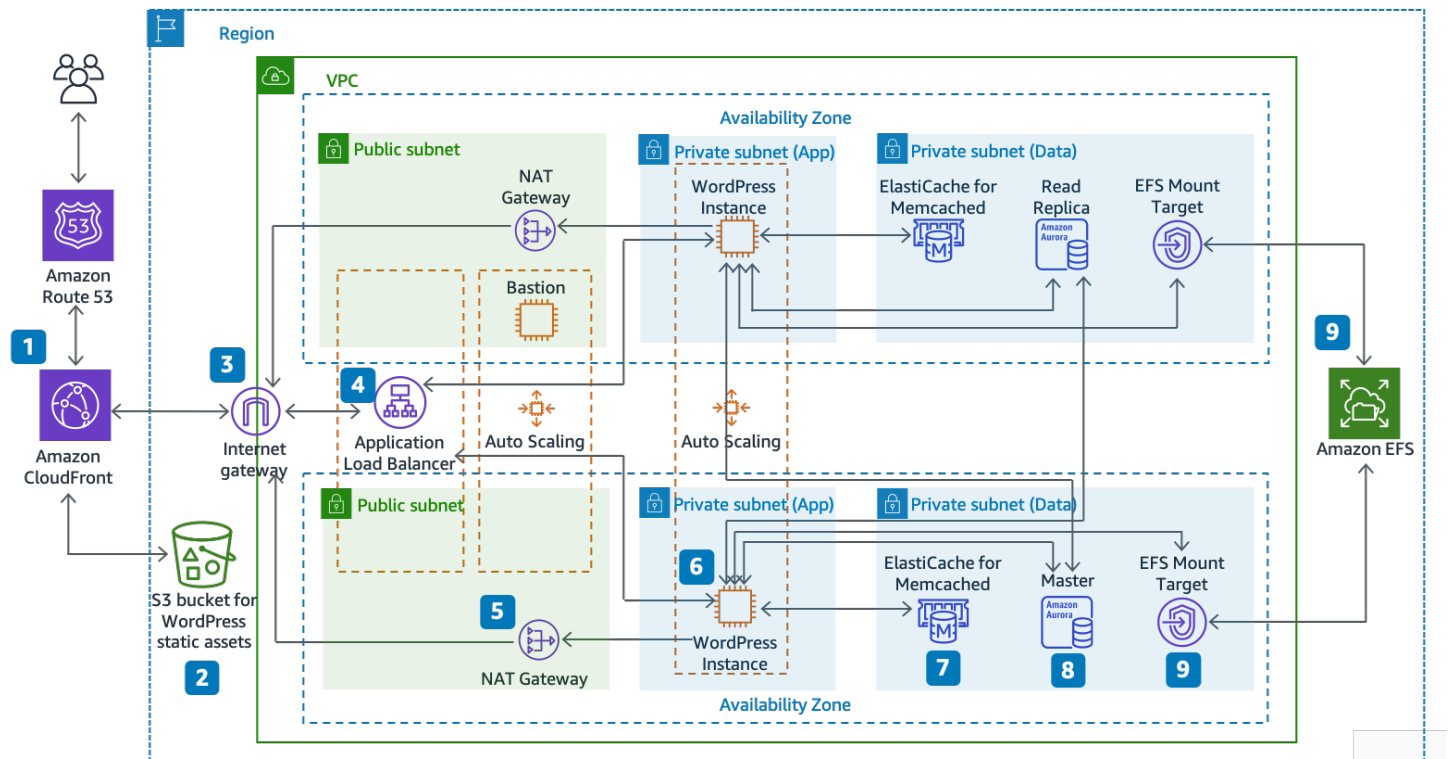
- [Arquitetura de referência](#)
- [Escalabilidade da camada da Web](#)
- [Camada da Web sem estado](#)
- [WordPress High Availability by Bitnami no Quick Start da AWS](#)

## Arquitetura de referência

A [Hosting WordPress on AWS reference architecture](#) (Arquitetura de referência para hospedar o WordPress na AWS) disponível no GitHub descreve as práticas recomendadas para a implantação do WordPress na AWS e inclui um conjunto de modelos AWS CloudFormation para que você comece rapidamente. A arquitetura a seguir é baseada nessa arquitetura de referência. O restante desta seção analisa os motivos por trás das opções de arquitetura.

A AMI baseada no GitHub foi alterada do Amazon Linux1 para o Amazon Linux2 em julho de 2021. No entanto, os modelos de implantação no S3 ainda não foram alterados. É recomendável usar modelos no GitHub se houver um problema para implantar a arquitetura de referência com modelos no S3.





## Arquitetura de referência para hospedar o WordPress na AWS

### Componentes da arquitetura

A arquitetura de referência ilustra uma completa implantação de práticas recomendadas para um site do WordPress na AWS.

- Ela começa com o cache de borda no Amazon CloudFront (1) a fim de armazenar conteúdo perto dos usuários finais para entrega mais rápida.
- O CloudFront extrai conteúdo de um bucket do S3 (2) e conteúdo dinâmico de um Application Load Balancer (4) na frente das instâncias da Web.
- As instâncias da Web são executadas em um grupo do Auto Scaling de instâncias do Amazon EC2 (6).
- Um cluster do ElastiCache (7) armazena dados consultados frequentemente para acelerar as respostas.

Uma instância do Amazon Aurora MySQL (8) hospeda o banco de dados do WordPress.

- As instâncias do EC2 do WordPress acessam dados do WordPress compartilhados em um sistema de arquivos do Amazon EFS por um destino de montagem do EFS (9) em cada zona de disponibilidade.

- Um Internet Gateway (3) permite a comunicação entre recursos na VPC e na Internet.
- Gateways NAT (5) em cada zona de disponibilidade ativam instâncias do EC2 em sub-redes privadas (aplicação e dados) para acessar a Internet.

Na Amazon VPC, há dois tipos de sub-redes: pública (sub-rede pública) e privada (sub-rede de aplicação e sub-rede de dados). Os recursos implantados nas sub-redes públicas recebem um endereço IP público e ficam publicamente visíveis na Internet. O Application Load Balancer (4) e um host Bastion da administração são implantados aqui. Os recursos implantados nas sub-redes privadas recebem apenas um endereço IP privado, por isso não ficam publicamente visíveis na Internet, o que melhora a segurança desses recursos. As instâncias do servidor Web do WordPress (6), as instâncias de cluster do ElastiCache (7), as instâncias de banco de dados do Aurora MySQL (8) e os destinos de montagem do EFS (9) são todos implantados em sub-redes privadas.

O restante desta seção aborda cada uma dessas considerações detalhadamente.

## Escalabilidade da camada da Web

Para transformar sua arquitetura de servidor único em uma arquitetura escalável de vários servidores, é necessário usar cinco componentes principais:

- Instâncias do Amazon EC2
- Imagens Amazon Machine Images (AMIs)
- Balanceadores de carga
- Escalabilidade automática
- Health checks

A AWS fornece uma variedade grande de tipos de instâncias do EC2 para que você escolha a melhor configuração de servidor para performance e custo. De um modo geral, o tipo de instância otimizado para computação (por exemplo, C4) pode ser uma boa opção para um servidor Web do WordPress. É possível implantar as instâncias em várias zonas de disponibilidade dentro de uma região AWS para aumentar a confiabilidade da arquitetura toda.

Como você tem controle total de sua instância do EC2, é possível fazer login com acesso root para instalar e configurar todos os componentes de software necessários para executar um site do WordPress. Feito isso, você pode salvar essa configuração como uma AMI, que pode em seguida usar para executar novas instâncias com todas as personalizações que você fez.

Para distribuir solicitações do usuário final a diversos nós de servidor Web, você precisa de uma solução de balanceamento de carga. A AWS fornece esse recurso por meio do [Elastic Load Balancing](#), um serviço altamente disponível que distribui tráfego para várias instâncias do EC2. Como seu site está veiculando conteúdo para os usuários via HTTP ou HTTPS, recomendamos que use o Application Load Balancer, um balanceador de carga na camada da aplicação com roteamento de conteúdo e capacidade de executar diversos sites do WordPress em domínios diferentes, se necessário.

O Elastic Load Balancing aceita a distribuição de solicitações em diversas zonas de disponibilidade na região AWS. Você ainda pode configurar uma health check para que o Application Load Balancer pare automaticamente de enviar tráfego para instâncias individuais que falharam (por exemplo, por um problema de hardware ou falha no software). A AWS recomenda usar a página de login do administrador do WordPress (`/wp-login.php`) para a health check, pois ela confirma se um servidor Web está funcionando e configurado para entregar arquivos PHP corretamente.

Você pode optar por criar uma página de health check personalizada que confira outros recursos dependentes, como banco de dados e recursos de cache. Para obter mais informações, consulte [Health checks para os grupos de destino](#), no Guia do Application Load Balancer.

Elasticidade é uma característica importante da Nuvem AWS. É possível executar maior capacidade computacional (por exemplo, servidores Web) quando precisar e executar menos quando não precisar. O [AWS Auto Scaling](#) é um serviço da AWS que ajuda a automatizar esse provisionamento a fim de aumentar ou diminuir a capacidade do Amazon EC2 de acordo com as condições definidas, sem necessidade de intervenção manual. Você pode configurar o AWS Auto Scaling de modo que o número de instâncias do EC2 em uso aumente facilmente durante os picos de demanda para manter a performance e diminua automaticamente quando o tráfego diminui, para minimizar os custos.

O Elastic Load Balancing também aceita a adição e a remoção dinâmicas de hosts do Amazon EC2 na alternância do balanceamento de carga. O próprio Elastic Load Balancing também aumenta e diminui dinamicamente a capacidade de balanceamento de carga para se ajustar às demandas de tráfego, sem intervenção manual.

## Camada da Web sem estado

Para aproveitar os diversos servidores Web na configuração de escalabilidade automática, a camada da Web deve ser “sem estado”. Uma aplicação sem estado não precisa de conhecimento de interações anteriores e não armazena informações de sessão. No caso do WordPress, isso significa que todos os usuários finais recebem a mesma resposta, independentemente de qual servidor Web

processou sua solicitação. Uma aplicação sem estado pode ter dimensionamento horizontal, pois qualquer solicitação pode ser entregue por qualquer recurso de computação disponível (ou seja, instâncias do servidor Web). Quando essa capacidade deixa de ser necessária, qualquer recurso individual pode ser encerrado com segurança (depois de executar tarefas que foram drenadas). Esses recursos não precisam ter conhecimento da presença de seus pares, pois eles só precisam de um jeito de receber a workload.

Quando se trata de armazenamento físico de dados da sessão do usuário, o núcleo do WordPress é completamente sem estado, pois conta com cookies armazenados no navegador da Web do cliente. O armazenamento de sessão não é uma preocupação, a menos que você tenha instalado qualquer código personalizado (por exemplo, um plugin do WordPress) que conte com sessões de PHP nativas.

No entanto, o WordPress foi originalmente projetado para executar em um servidor único. Como resultado, ele armazena alguns dados no sistema de arquivos local do servidor. Ao executar o WordPress em uma configuração de vários servidores, isso gera um problema, pois há uma inconsistência entre os servidores da Web. Por exemplo, se um usuário fizer upload de uma nova imagem, ela será armazenada somente em um dos servidores.

Isso demonstra por que precisamos aprimorar a configuração de execução padrão do WordPress para mover dados importantes para um armazenamento compartilhado. A arquitetura de prática recomendada tem um banco de dados como uma camada separada fora do servidor Web e usa o armazenamento compartilhado para armazenar uploads, temas e plugins do usuário.

## Armazenamento compartilhado (Amazon S3 e Amazon EFS)

Por padrão, o WordPress armazena uploads do usuário no sistema de arquivos local, portanto não é sem estado. Portanto, precisamos mover a instalação do WordPress e todas as personalizações do usuário (como configuração, plugins, temas e uploads gerados pelo usuário) para uma plataforma de dados compartilhados a fim de ajudar a reduzir a carga nos servidores Web e tornar a camada da Web sem estado.

O [Amazon Elastic File System](#) (Amazon EFS) fornece sistemas de arquivos de rede escaláveis para uso com instâncias do EC2. Os sistemas de arquivos do Amazon EFS são distribuídos em um número irrestrito de servidores de armazenamento, possibilitando que sistemas de arquivos cresçam de forma elástica e permitindo o acesso paralelo em grande escala das instâncias do EC2. O design distribuído do Amazon EFS evita gargalos e restrições inerentes a servidores de arquivos tradicionais.

Com a migração de todo o diretório de instalação do WordPress para um sistema de arquivos do EFS e a montagem dele em cada instância do EC2 durante a inicialização, o site do WordPress e todos os seus dados são automaticamente armazenados em um sistema de arquivos distribuídos que não depende de qualquer instância do EC2, tornando a camada da Web completamente sem estado. O benefício dessa arquitetura é que você não precisa instalar plugins e temas em cada execução da instância, e ainda pode acelerar muito a instalação e a recuperação das instâncias do WordPress. Também é mais fácil implantar alterações em plugins e temas no WordPress, conforme definido na seção [Considerações de implantação](#) deste documento.

Para garantir a melhor performance do site ao executar um sistema de arquivos do EFS, confira as configurações recomendadas do Amazon EFS e do OPcache na [AWS Reference Architecture for WordPress](#) (Arquitetura de referência da AWS para o WordPress).

Você também tem a opção de descarregar todos os ativos estáticos, como imagens, CSS e arquivos JavaScript, em um bucket do S3 com caching do CloudFront na frente. O mecanismo para fazer isso em uma arquitetura de vários servidores é exatamente o mesmo para uma arquitetura de servidor único, conforme abordado na seção [Conteúdo estático](#) deste whitepaper. Os benefícios são os mesmos da arquitetura de servidor único: você pode descarregar o trabalho associado à entrega dos ativos estáticos no Amazon S3 e no CloudFront, assim seus servidores Web se concentram em gerar conteúdo dinâmico somente e entregam mais solicitações de usuário por servidor Web.

## Camada de dados (Amazon Aurora e Amazon ElastiCache)

Com a instalação do WordPress armazenada em um sistema de arquivos de rede compartilhado, distribuído e escalável e com os ativos estáticos sendo entregues pelo Amazon S3, você pode concentrar sua atenção no componente stateful restante: o banco de dados. Assim como acontece com a camada de armazenamento, o banco de dados não deve depender de um único servidor, portanto, não pode ser hospedado em um dos servidores Web. Por isso, hospedamos o banco de dados do WordPress no Amazon Aurora.

O [Amazon Aurora](#) é um banco de dados relacional compatível com MySQL e PostgreSQL projetado para a nuvem, que combina a performance e a disponibilidade de bancos de dados comerciais avançados com a simplicidade e a economia de bancos de dados de código aberto. O Aurora MySQL aumenta a performance e a disponibilidade do MySQL integrando totalmente o mecanismo de banco de dados com um sistema de armazenamento distribuído especificamente, com o respaldo do SSD. Ele tem tolerância a falhas e autocura, replica seis cópias de seus dados entre três zonas de disponibilidade, foi projetado para uma disponibilidade de mais de 99,99% e copia continuamente seus dados no Amazon S3. O Amazon Aurora foi projetado para detectar automaticamente falhas

de banco de dados e reiniciar sem necessidade de recuperação de falhas ou de recriar o cache do banco de dados.

O Amazon Aurora fornece vários [tipos de instâncias](#) para atender a diferentes perfis de aplicações, inclusive instâncias otimizadas para memória e com capacidade de intermitência. Para melhorar a performance do banco de dados, é possível selecionar um tipo de instância grande para fornecer mais recursos de CPU e memória.

O Amazon Aurora gerencia automaticamente o failover entre a instância primária e as [réplicas do Aurora](#), assim a aplicação retoma as operações do banco de dados o quanto antes, sem intervenção administrativa manual. O failover costuma ter duração inferior a 30 segundos.

Depois que você criar pelo menos uma réplica do Aurora, conecte sua instância primária usando o endpoint do cluster para que a aplicação reproduza o failover automaticamente caso a instância primária falhe. É possível criar até 15 réplicas de leitura de baixa latência em três zonas de disponibilidade.

Conforme seu banco de dados dimensiona, o cache também precisa ser dimensionado. Conforme abordado na seção [Armazenamento em cache de banco de dados](#), o ElastiCache conta com recursos para escalar o cache em diversos nós em um cluster do ElastiCache e em diversas zonas de disponibilidade em uma região a fim de melhorar a disponibilidade. Conforme você escala o cluster do ElastiCache, configure o plugin de armazenamento em cache para se conectar usando o endpoint de configuração, assim o WordPress pode usar novos nós de cluster conforme são adicionados e parar de usar os antigos à medida que são removidos. Também é necessário configurar servidores Web para usar o [ElastiCache Cluster Client for PHP](#) e atualizar a AMI para armazenar essa alteração.

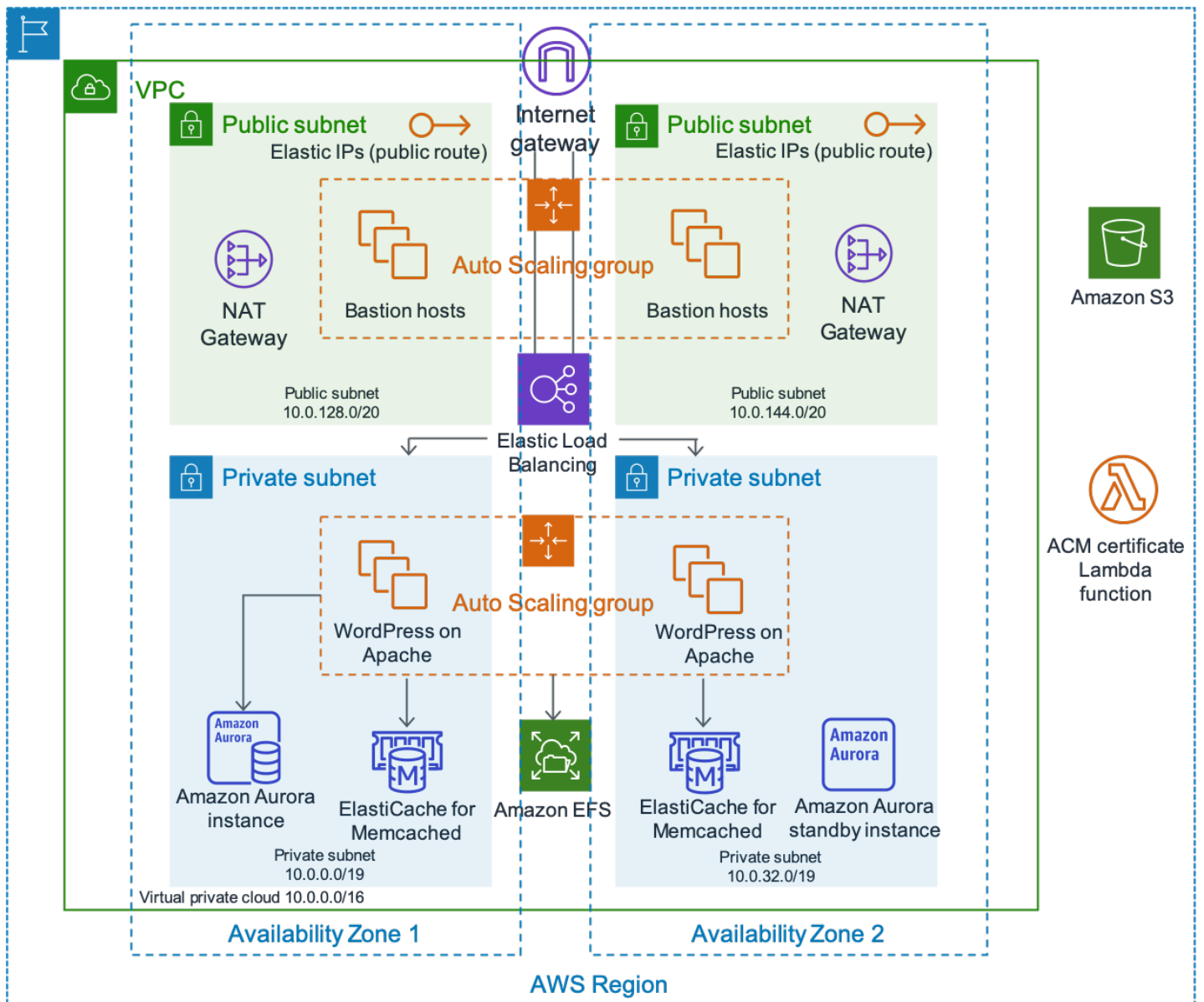
## WordPress High Availability by Bitnami no Quick Start da AWS

Os Quick Starts são criados por arquitetos de soluções e parceiros da AWS para ajudar você a implantar tecnologias populares na AWS, de acordo com as práticas recomendadas de segurança e a alta disponibilidade da AWS. Esses aceleradores transformam centenas de procedimentos manuais em poucas etapas, o que permite que você crie rapidamente um ambiente de produção e comece a usá-lo imediatamente. Cada Quick Start inclui modelos do AWS CloudFormation que automatizam a implantação, além de um guia que aborda a arquitetura e fornece instruções passo a passo de implantação.

O [WordPress High Availability by Bitnami no Quick Start da AWS](#) configura o seguinte ambiente configurável na AWS:

- Uma arquitetura altamente disponível que abrange duas Zonas de disponibilidade.\*
- Uma Virtual Private Cloud (VPC) configurada com sub-redes públicas e privadas de acordo com as práticas recomendadas da AWS. Isso fornece a infraestrutura de rede para sua implantação.\*
- Um internet gateway para fornecer acesso à Internet. Esse gateway é usado pelos bastion hosts para enviar e receber tráfego.\*
- Nas sub-redes públicas, gateways NAT gerenciados para permitir acesso de saída à Internet para recursos nas sub-redes privadas.\*
- Nas sub-redes públicas, bastion hosts do Linux em um grupo do Auto Scaling para permitir acesso de entrada Secure Shell (SSH) às instâncias do EC2 em sub-redes públicas e privadas.\*
- Elastic Load Balancing para distribuir solicitações HTTP e HTTPS por meio de múltiplas instâncias do WordPress.
- Em sub-redes privadas, instâncias do EC2 que hospedam a aplicação do WordPress no Apache. Essas instâncias são fornecidas em um grupo do Auto Scaling para garantir a alta disponibilidade.
- Em sub-redes privadas, as instâncias de banco de dados do Amazon Aurora administradas pelo Amazon Relational Database Service (Amazon RDS).
- Nas sub-redes privadas, o Amazon Elastic File System (Amazon EFS) para compartilhar ativos (como plugins, temas e imagens) em instâncias do WordPress.
- Nas sub-redes privadas, nós do Amazon ElastiCache for Memcached para armazenar em cache consultas do banco de dados.

\*O modelo que implanta o Quick Start em uma VPC existente ignora as tarefas marcadas por asteriscos e solicita sua configuração de VPC existente.



## Arquitetura do WordPress High Availability by Bitnami

Uma descrição detalhada da implantação do WordPress High Availability by Bitnami na AWS não faz parte do escopo deste documento. Para configuração e opções, consulte [WordPress High Availability by Bitnami na AWS](#).



## Conclusão

A AWS tem muitas opções de arquitetura para executar o WordPress. A opção mais simples é uma instalação de servidor único para sites de tráfego baixo. Para sites mais avançados, os administradores podem adicionar diversas outras opções, cada uma representando uma melhoria incremental em termos de disponibilidade e escalabilidade. Os administradores podem selecionar os recursos que se associam melhor com seus requisitos e seu orçamento.

# Colaboradores

Os colaboradores desse documento incluem:

- Paul Lewis, arquiteto de soluções da Amazon Web Services
- Ronan Guilfoyle, arquiteto de soluções da Amazon Web Services
- Andreas Chatzakis, gerente de arquitetura de soluções da Amazon Web Services
- Jibril Touzi, gerente de conta técnico da Amazon Web Services
- Hakmin Kim, arquiteto de soluções de parceiros de migração da Amazon Web Services

## Revisões do documento

Para ser notificado sobre atualizações deste whitepaper, inscreva-se no RSS feed.

update-history-change	update-history-description	update-history-date
<a href="#">Whitepaper atualizado</a>	Atualizado para modificar a arquitetura de referência e o plugin AWS for WordPress.	19 de outubro de 2021
<a href="#">Whitepaper atualizado</a>	Atualizado para incluir novas abordagens de implantação e o plugin AWS for WordPress.	30 de outubro de 2019
<a href="#">Whitepaper atualizado</a>	Atualizado para esclarecer o sistema de mensagens do produto do Amazon Aurora.	1 de fevereiro de 2018
<a href="#">Whitepaper atualizado</a>	Atualizado para incluir serviços da AWS lançados desde a primeira publicação.	1º de dezembro de 2017
<a href="#">Publicação inicial</a>	Primeira publicação.	1º de dezembro de 2014

# Apêndice A: configuração do CloudFront

Para obter a performance e a eficiência ideais ao usar o Amazon CloudFront com seu site do WordPress, é importante configurar o site corretamente para os diferentes tipos de conteúdo disponibilizados.

Tópicos

- [Origens e comportamentos](#)
- [Criação de distribuição do CloudFront](#)

## Origens e comportamentos

Uma [origem](#) é um local em que o CloudFront envia solicitações de conteúdo que ele distribui por meio dos locais da borda. Dependendo da implementação, é possível ter uma ou duas origens. Uma para conteúdo dinâmico (a instância do Lightsail na [opção de implantação de servidor único](#) ou o Application Load Balancer na [opção de implantação elástica](#)) usando uma origem personalizada. E uma segunda origem para direcionar o CloudFront para o conteúdo estático. Um bucket do S3, de acordo com a [arquitetura de referência](#) anterior. Ao usar o Amazon S3 como origem da distribuição, é necessário usar uma [política de bucket](#) para que o conteúdo seja acessível publicamente.

Os [comportamentos](#) permitem definir regras que controlam como o CloudFront armazena o conteúdo em cache e, por sua vez, determinam a eficácia do cache. Os comportamentos permitem controlar o protocolo e os métodos HTTP pelos quais o site pode ser acessado. Eles também permitem que você controle se passará cabeçalhos HTTP, cookies ou strings de consulta para o back-end (nesse caso, quais são eles). Os comportamentos se aplicam a padrões de caminhos de URL específicos.

## Criação de distribuição do CloudFront

Crie uma distribuição na Web do CloudFront seguindo a distribuição; a origem e o comportamento padrão criados automaticamente serão usados para conteúdo dinâmico. Crie quatro comportamentos adicionais para personalizar ainda mais a forma como as solicitações estáticas e dinâmicas são tratadas. A tabela a seguir resume as propriedades de configuração dos cinco comportamentos. Também é possível ignorar esta configuração manual e usar o plugin AWS for WordPress abordado no [Apêndice B: instalação e configuração de plugins](#). Esta é a maneira mais fácil de configurar o CloudFront para acelerar o site do WordPress.

Tabela 1: resumo das propriedades de configuração dos comportamentos do CloudFront

Propriedade	Estático	Dinâmico (administrador)	Dinâmico (front-end)
Caminhos (comportamentos)	wp-content/* wp-includes/*	wp-admin/* wp-login.php	padrão (*)
Protocolos	HTTP e HTTPS	Redirecionar para HTTPS	HTTP e HTTPS
Métodos HTTP	GET, HEAD	ALL	ALL
Cabeçalhos 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-*
Query strings	YES (invalidação)	SIM	SIM

Para o comportamento padrão, a AWS recomenda a seguinte configuração:

- Ative a Política de protocolo de origem como Match Viewer, de modo que, se os visualizadores se conectarem ao CloudFront usando HTTPS, o CloudFront se conectará à origem usando HTTPS também, obtendo uma criptografia de ponta a ponta. Observe que isso requer a instalação de um certificado SSL confiável no balanceador de carga. Para obter mais detalhes, consulte [Exigir HTTPS para comunicação entre o CloudFront e a origem personalizada](#).
- Ative todos os métodos HTTP, pois as partes dinâmicas do site precisam das solicitações GET e POST (por exemplo, para aceitar POST nos formulários de envio de comentários).
- Encaminhe somente os cookies que apresentam resultados do WordPress diferentes; por exemplo `>wordpress_*`, `wp-settings-*` e `comment_*`. Caso tenha instalado plugins que dependam de cookies que não estão na lista, será necessário aumentá-la.
- Encaminhe somente os cabeçalhos HTTP que afetam o resultado do WordPress, por exemplo `Host`, `CloudFront-Forwarded-Proto`, `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` e `CloudFront-is-Tablet-Viewer`:
  - `Host` permite que vários sites do WordPress sejam hospedados na mesma origem.
  - `CloudFront-Forwarded-Proto` permite que diferentes versões de páginas sejam armazenadas em cache, dependendo se elas são acessadas via HTTP ou HTTPS.
  - `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` e `CloudFront-is-Tablet-Viewer` permitem personalizar o resultado dos temas com base no tipo de dispositivo do usuário final.
- Encaminhe todas as strings de consulta para o cache, de acordo com os respectivos valores, pois o WordPress depende delas. Essas strings também podem ser usadas para invalidar objetos em cache.

Se deseja veicular seu site com um nome de domínio personalizado (ou seja, diferente de `*.cloudfront.net`), insira os URIs apropriados em Alternate Domain Names (Nomes de domínio alternativos) nas Distribution Settings (Configurações da distribuição). Neste caso, também é necessário um certificado SSL do nome de domínio personalizado. É possível [solicitar](#) certificados SSL por meio do AWS Certificate Manager e configurá-los em uma distribuição do CloudFront.

Agora, crie mais dois comportamentos de cache para conteúdo dinâmico: um para a página de login (padrão de caminho: `wp-login.php`) e outro para o painel de administrador (padrão de caminho: `wp-admin/*`). Esses dois comportamentos têm exatamente as mesmas configurações, como segue:

- Aplique uma política de protocolo de visualização de HTTPS somente.

- Ative todos os métodos HTTP.
- Armazene com base em todos os cabeçalhos HTTP.
- Encaminhe todos os cookies.
- Encaminhe e armazene com base em todas as strings de consulta.

O motivo por trás dessa configuração é que esta seção do site é altamente personalizada e, geralmente, tem apenas alguns usuários, por isso a eficiência do cache não é a maior preocupação. O objetivo é manter a configuração simples a fim de garantir máxima compatibilidade com todos os plugins instalados, transferindo todos os cookies e cabeçalhos para a origem.

O [plugin AWS for WordPress](#) abordado no Apêndice B cria automaticamente uma distribuição do CloudFront que atende à configuração anterior.

Por padrão, o WordPress armazena tudo localmente no servidor Web, que é o armazenamento em bloco (Amazon EBS) para [implantação de servidor único](#) e o armazenamento de arquivos (Amazon EFS) para [implantação elástica](#). Além de reduzir os custos de armazenamento e transferência de dados, a transferência de ativos estáticos para o Amazon S3 oferece escalabilidade, disponibilidade de dados, segurança e performance. Existem vários plugins que facilitam a transferência de conteúdo estático para o Amazon S3; um deles é o [W3 Total Cache](#), também abordado no [Apêndice B: instalação e configuração de plugins](#).

# Apêndice B: instalação e configuração de plugins

## Tópicos

- [Plugin AWS for WordPress](#)
- [Configuração de conteúdo estático](#)

## Plugin AWS for WordPress

O plugin AWS for WordPress é o único plugin do WordPress gravado e mantido ativamente pela AWS. Ele permite que os clientes configurem facilmente o [Amazon CloudFront](#) e o [AWS Certificate Manager \(ACM\)](#) em sites do WordPress para melhorar a performance e a segurança. O plugin usa os serviços do [Amazon Machine Learning](#) para traduzir conteúdo para um ou mais idiomas, produz versões de áudio de cada tradução e lê sites do WordPress por meio de dispositivos Amazon Alexa.

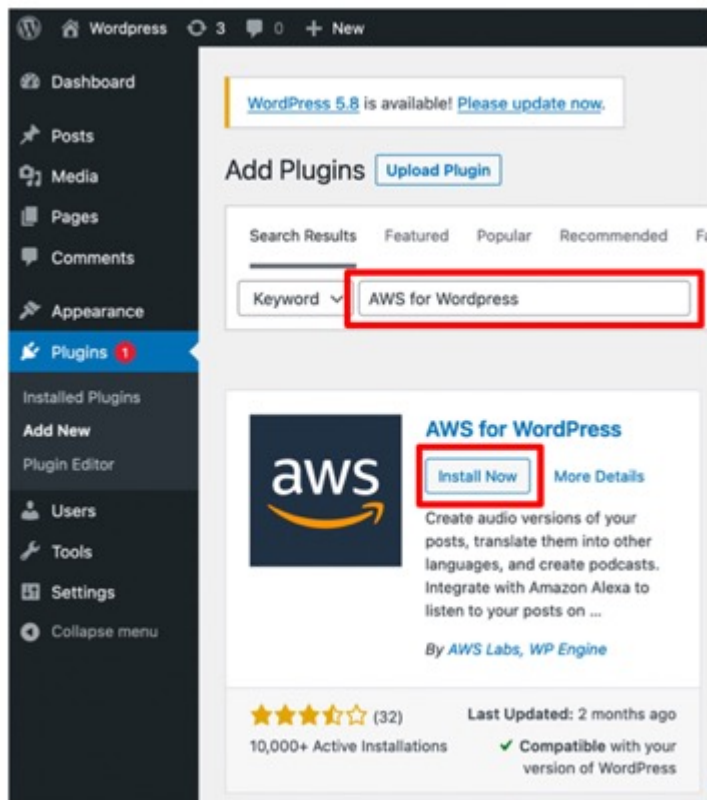
O plugin já está instalado no [WordPress High Availability by Bitnami no Quick Start da AWS](#).

## Instalação e configuração de plugins

Como instalar o plugin:

1. Para usar o plugin AWS for WordPress, é necessário criar um usuário do IAM para o plugin. Um usuário do IAM é uma pessoa ou uma aplicação em uma conta da AWS que tem permissão para fazer chamadas de API para serviços da AWS.
2. É necessário uma função do [AWS Identity and Access Management](#) (IAM) ou um usuário do IAM para controlar a autenticação e a autorização da sua conta da AWS. Para impedir que usuários não autorizados obtenham essas permissões, proteja as credenciais do usuário do IAM. Trate a chave de acesso secreta como uma senha; guarde-a em um local seguro e não a compartilhe com ninguém. Assim como com uma senha, [alterne a chave de acesso periodicamente](#). Se a chave de acesso secreta vazar acidentalmente, [exclua-a imediatamente](#). É possível criar uma nova chave de acesso para usar com o plugin AWS for WordPress.
3. No menu Plugins do painel de administrador do WordPress, pesquise AWS for WordPress e escolha Install Now (Instalar agora).





4. Se a instalação do plugin não estiver funcionando, pode haver um problema de permissão do usuário. Conecte-se ao servidor Web do WordPress e complete as instruções a seguir para resolver o problema.
  - a. Abra o arquivo `wp-config.php` no diretório de instalação do WordPress e grave o código a seguir no final do arquivo `wp-config.php`:

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

- b. Execute o comando a seguir para conceder permissão de gravação:

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

Aviso: manter a permissão de gravação como 777 é arriscado. Se a permissão for mantida como 777, qualquer pessoa poderá editar ou excluir essa pasta. Altere a permissão de gravação para 755 ou inferior após concluir o trabalho do plugin.

- c. Se a arquitetura de referência for usada, o diretório de instalação do WordPress será ``/var/www/wordpress/<site directory>``.

Uma descrição detalhada de todas as configurações do AWS for WordPress não faz parte do escopo deste documento. Para configuração e opções, consulte [Conceitos básicos do plugin AWS for WordPress](#).

## Amazon CloudFront e AWS Certificate Manager

Para configurar o CloudFront e o AWS Certificate Manager:

1. No menu de plugins, escolha CloudFront e insira os seguintes parâmetros:
  - Nome de domínio de origem: domínio DNS do servidor de origem HTTP em que o CloudFront obtém o conteúdo do site (como `example.com`).
  - Nome de domínio alternativo (CNAME): nome de domínio que os visitantes usam para a experiência acelerada do site. A AWS recomenda usar “www” na frente do domínio (como `www.example.com`).
2. Escolha Initiate Setup (Iniciar configuração) para iniciar a configuração.

O plugin solicita automaticamente um certificado SSL para o CNAME via ACM. Assim que validar o token do ACM [ao atualizar os registros DNS](#) com as entradas CNAME, o plugin criará uma distribuição do CloudFront que atenda às práticas recomendadas definidas no [Apêndice A](#).

### Note

O plugin AWS for WordPress requer HTTPS para comunicação entre o CloudFront e a origem personalizada. Certifique-se de que a origem tenha um certificado SSL válido para o nome de domínio de origem. Para mais informações, consulte [Exigir HTTPS na comunicação entre o CloudFront e a origem personalizada](#).

## Traduzir e vocalizar o conteúdo

O plugin AWS for WordPress permite traduzir texto automaticamente em diferentes idiomas e converter o conteúdo escrito em formatos de áudio multilíngue. Esses recursos são desenvolvidos pelos serviços do Amazon Machine Learning.

O [Amazon Polly](#) é um serviço que transforma texto em falas realistas. Com dezenas de vozes em vários idiomas, é possível selecionar a voz ideal e criar aplicações com recursos de fala interessantes que funcionam em vários países diferentes. Use o plugin para criar arquivos de áudio

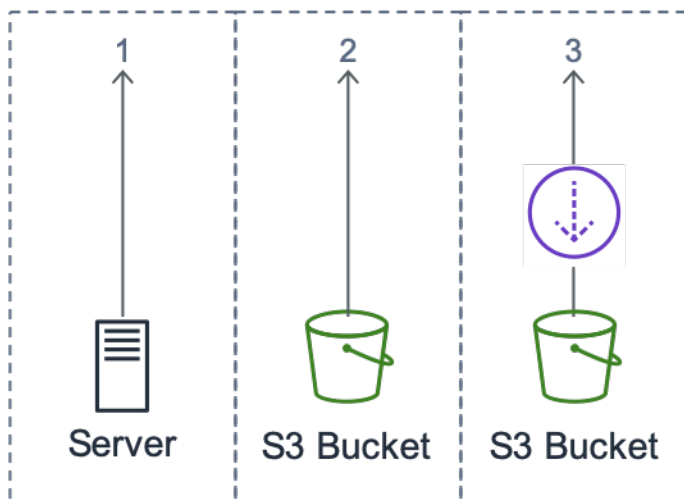
em qualquer uma das vozes e idiomas compatíveis com o Amazon Polly. Os visitantes podem transmitir o áudio conforme necessário usando reprodutores de áudio em linha e aplicativos móveis.

Por padrão, o plugin armazena os novos arquivos de áudio em seu servidor Web. Você pode optar por armazenar os arquivos no Amazon S3 ou no Amazon CloudFront. Os usuários terão a mesma experiência de áudio, independentemente de onde você armazenar seus arquivos de áudio. Somente o local da transmissão é alterado:

- Para arquivos de áudio armazenados no servidor do WordPress, os arquivos são transmitidos diretamente do servidor.
- Para arquivos armazenados em um bucket do S3, os arquivos são transmitidos do bucket.
- Se você usa o CloudFront, os arquivos são armazenados no Amazon S3 e transmitidos com o CloudFront.



Listening...

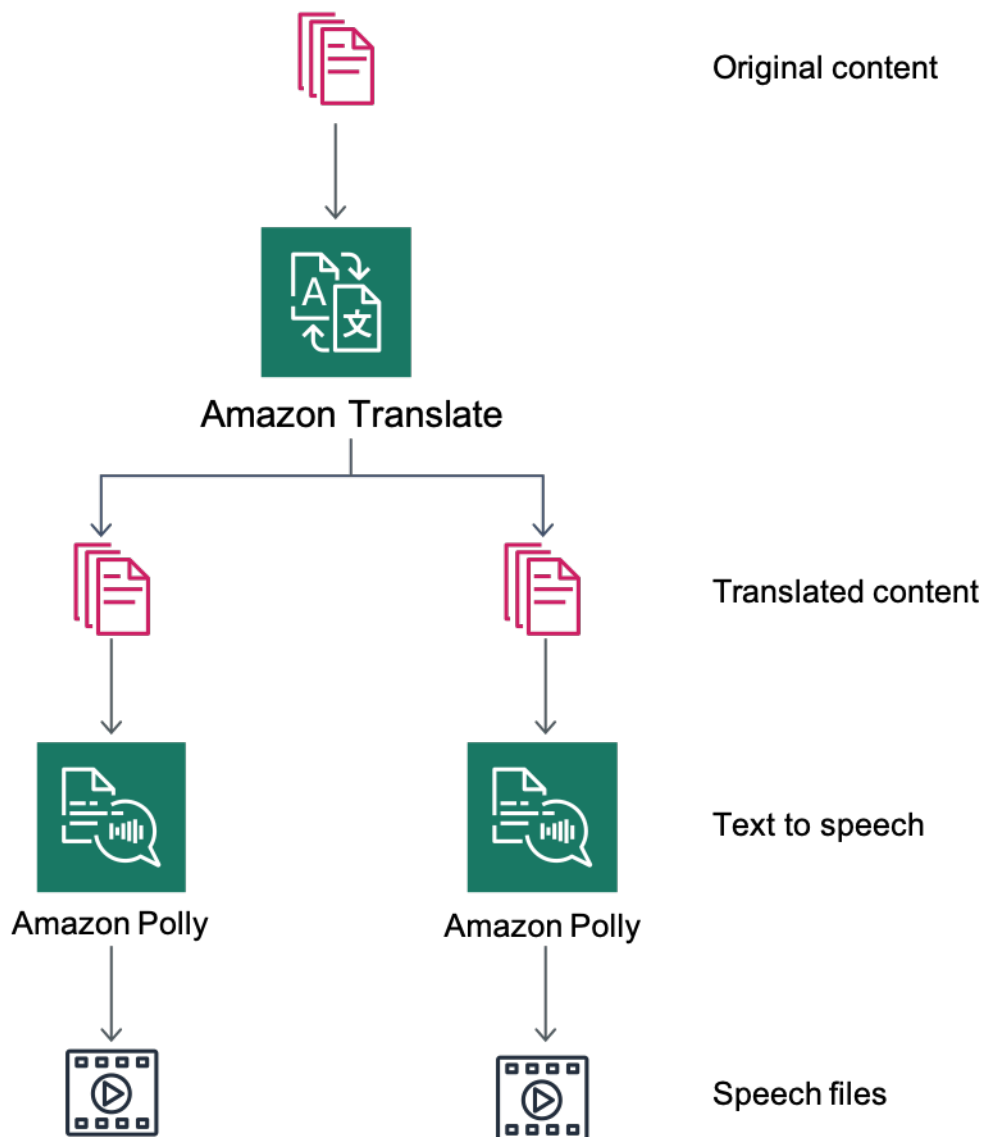


## Local de transmissão

O [Amazon Translate](#) é um serviço de tradução de máquina que oferece tradução de idiomas rápida, acessível e de alta qualidade. A disponibilização de conteúdo multilíngue representa uma grande oportunidade para os proprietários de sites. Embora o inglês seja o idioma dominante na Web, os falantes nativos do idioma representam apenas 26% do público online total.

Ao oferecer versões em áudio e por escrito de seu conteúdo do WordPress em vários idiomas, é possível atender às necessidades de um público internacional ainda maior. É possível configurar o plugin para fazer o seguinte:

- Traduza automaticamente para diferentes idiomas e crie gravações de áudio de cada tradução para novos conteúdos após a publicação ou opte por traduzir e criar gravações para publicações individuais.
- Traduza para diferentes idiomas e crie gravações de áudio para cada tradução do conteúdo arquivado.
- Usar o feed RSS do Amazon Pollycast para disponibilizar conteúdo de áudio em formato de podcast.



## Visão geral da tradução de conteúdo e conversão de texto em fala

### Podcasts com o Amazon Pollycast

Com os feeds do Amazon Pollycast, seus visitantes podem ouvir o conteúdo de áudio usando aplicações de podcast padrão. Os feeds do Pollycast em conformidade com a especificação RSS 2.0 fornecem os dados XML necessários para agregar podcasts por meio de aplicativos móveis de podcast populares, como iTunes e diretórios de podcast.

Ao instalar o plugin AWS for WordPress, você encontrará a opção de ativar a geração de feed XML na guia de configuração do podcast. Você também encontrará a opção de configurar várias propriedades opcionais. Depois de ativar a funcionalidade, você receberá um link para o feed.

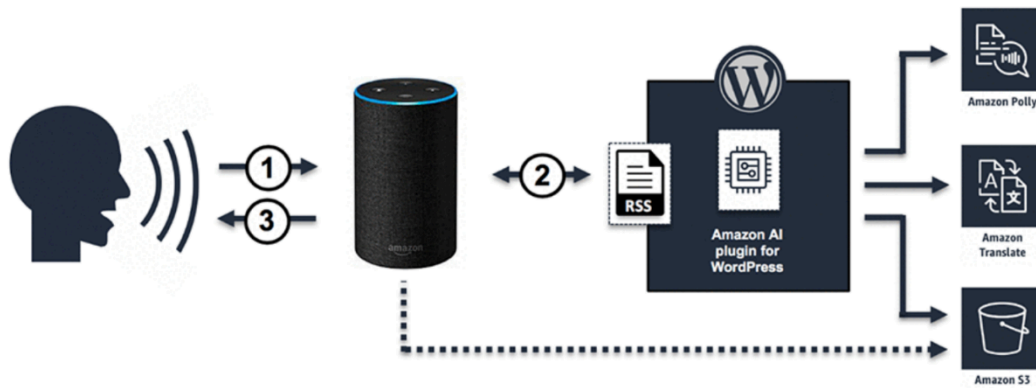
### Ler o conteúdo por meio de dispositivos Amazon Alexa

É possível estender sites e blogs do WordPress por meio de dispositivos Alexa. Isso abre novas possibilidades para os criadores e autores de sites atingirem um público ainda mais amplo. Também faz com que as pessoas ouçam seus blogs favoritos de forma mais fácil, simplesmente pedindo à Alexa que os leia.

Para expor o site do WordPress à Alexa, é necessário ativar:

- O plugin AWS for WordPress.
- As funcionalidades de conversão de texto em fala e do Amazon Pollycast. Essas funcionalidades geram um feed RSS no site do WordPress que é utilizado pelo Amazon Alexa.
- O Amazon S3 como armazenamento padrão para arquivos em conversão de texto em fala. É importante que o site use uma conexão HTTPS segura para expor o feed à Alexa.

O diagrama a seguir apresenta o fluxo de interações e componentes necessários para expor o site por meio da Alexa.



Fluxo de interações necessárias para expor sites do WordPress por meio da Alexa

1. O usuário invoca uma nova skill da Alexa, por exemplo, dizendo: Alexa, peça ao Demo Blog a atualização mais recente. A skill em si é criada usando um dos Alexa Skill Blueprints. Isso permite que você exponha a skill por meio de dispositivos Alexa, mesmo que não tenha um conhecimento técnico profundo.
2. A skill da Alexa analisa a chamada e o feed RSS que foi gerado pelo plugin AWS for WordPress e retorna o link para a versão de áudio do artigo mais recente.
3. Com base no link fornecido pelo feed, a Alexa lê o artigo reproduzindo o arquivo de áudio salvo no Amazon S3.

Consulte a [página do plugin](#) no Marketplace WordPress a fim de obter um guia de instruções detalhado para instalar e configurar o plugin e suas funcionalidades.

## Configuração de conteúdo estático

Por padrão, o WordPress armazena tudo localmente no servidor Web, que é o armazenamento em bloco (Amazon EBS) para [implantação de servidor único](#) e o armazenamento de arquivos (Amazon EFS) para [implantação elástica](#). Além de reduzir os custos de armazenamento e transferência de dados, a transferência de ativos estáticos para o Amazon S3 oferece escalabilidade, disponibilidade de dados, segurança e performance.

Neste exemplo, o plugin W3 Total Cache (W3TC) é usado para armazenar ativos estáticos no Amazon S3. Entretanto, há outros plugins disponíveis com capacidade parecida. Se quiser usar uma alternativa, poderá ajustar as etapas a seguir adequadamente. As etapas se referem somente a recursos ou configurações relevantes a este exemplo. Uma descrição detalhada de todas as

configurações vai além do escopo deste documento. Para mais informações, consulte a [página do plugin W3 Total Cache](#) em [wordpress.org](http://wordpress.org).

## Criação de usuário do IAM

É necessário criar um usuário AWS Identity and Access Management (IAM) para o plugin do WordPress a fim de armazenar ativos estáticos no Amazon S3. Para ver as etapas, consulte [Como criar um usuário do IAM na sua conta da AWS](#).

Observação: as funções do IAM são um jeito melhor de gerenciar o acesso a recursos da AWS, mas, no momento da gravação, o plugin W3 Total Cache não aceita [funções do IAM](#).

Anote as credenciais de segurança do usuário e as armazene de forma segura, pois você precisará delas depois.

## Criação de bucket do Amazon S3

1. Primeiro, crie um bucket do Amazon S3 na região AWS de sua escolha. Para ver as etapas, consulte [Como criar um bucket](#). Ative a hospedagem estática de sites para o bucket seguindo o [Tutorial: configurar um site estático no Amazon S3](#).
2. Crie uma política do IAM para fornecer ao usuário do IAM criado anteriormente acesso ao bucket do S3 especificado e anexe a política ao usuário do IAM. Para obter as etapas de criação da política a seguir, consulte [Gerenciar políticas do 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. Instale e ative o plugin W3TC pelo painel do administrador do WordPress.
4. Acesse a seção Configurações gerais do plugin e verifique se o Browser Cache (Cache do navegador) e a CDN estão ativados.
5. Na lista suspensa na configuração da CDN, escolha Origin Push: Amazon CloudFront (essa opção tem o Amazon S3 como origem).
6. Acesse a seção Browser Cache (Cache do navegador) da configuração do plugin e ative os cabeçalhos expires, cache control e entity tag (ETag).
7. Ative a opção Prevent caching of objects after settings change (Impedir o armazenamento em cache dos objetos após a alteração das configurações), assim uma nova string de consulta é gerada e anexada a objetos sempre que as configurações forem alteradas.
8. Acesse a seção CDN da configuração do plugin e insira as credenciais de segurança do usuário do IAM que você criou anteriormente, além do nome do bucket do S3.
9. Se você estiver disponibilizando o site por um URL do CloudFront, insira o nome de domínio de distribuição na caixa relevante. Senão, insira um ou mais CNAMEs para os nomes de domínio personalizados.
10. Por fim, exporte a biblioteca de mídia e faça upload dos wp-includes, arquivos de temas e arquivos personalizados no Amazon S3 usando o plugin W3TC. Essas funções de upload estão disponíveis na seção Geral da página de configuração da CDN.

## Criação de uma origem estática

Agora que os arquivos estáticos estão armazenados no Amazon S3, volte para a configuração do CloudFront no console do CloudFront e configure o Amazon S3 como origem de conteúdo estático. Para isso, adicione uma segunda origem direcionada para o bucket do S3 que você criou exatamente para isso. Depois crie dois outros comportamentos de cache, um para cada uma das duas pastas (wp-content e wp-includes), que devem usar a origem do S3 em vez da origem padrão do conteúdo dinâmico. Configure os dois do mesmo jeito:

- Disponibilize somente solicitações HTTP GET.



- O Amazon S3 não varia seu resultado com base em cookies ou cabeçalhos HTTP, por isso você pode aprimorar a eficiência do cache deixando de encaminhá-los para a origem pelo CloudFront.
- Apesar de esses comportamentos entregarem somente conteúdo estático (que não aceita parâmetros), você encaminhará strings de consulta para a origem. Isso é para que você possa usar strings de consulta como identificadores de versão para invalidar instantaneamente, por exemplo, arquivos CSS mais antigos ao implantar novas versões. Consulte o [Guia do desenvolvedor do Amazon CloudFront](#) para obter mais informações.

#### Note

Depois de adicionar os comportamentos de origem estática à distribuição do CloudFront, confira a ordem para garantir que os comportamentos `wp-admin/*` e `wp-login.php` tenham maior precedência que os comportamentos do conteúdo estático. Caso contrário, você pode ver um comportamento estranho ao acessar o painel de administrador.

## Apêndice C: backup e recuperação

A recuperação pós-falha na AWS é mais rápida e mais fácil em comparação com ambientes de hospedagem tradicionais. Por exemplo, você pode executar uma instância de substituição em minutos como resposta a uma falha no hardware, ou usar o failover automatizado em muitos dos nossos managed services para negar o impacto de uma reinicialização devido à manutenção de rotina.

Entretanto, você ainda precisa garantir a cópia dos dados certos para conseguir recuperá-los. Para restabelecer a disponibilidade de um site do WordPress, recupere os seguintes componentes:

- Sistema operacional (OS) e instalação e configuração de serviços (Apache, MySQL e assim por diante)
- Código de aplicação e configuração do WordPress
- Temas e plugins do WordPress
- Uploads (por exemplo, arquivos de mídia para publicações)
- Conteúdo do banco de dados (publicações, comentários e assim por diante)

A AWS oferece uma variedade de métodos para fazer backup e restaurar seus dados e ativos de aplicações Web.

Este whitepaper apresentou anteriormente o uso de snapshots do Lightsail para proteger todos os dados armazenados no armazenamento local da instância. Se o site do WordPress for executado apenas na instância do Lightsail, os snapshots regulares do Lightsail deverão ser suficientes para que você recupere o site do WordPress na íntegra. Mas você ainda perderá todas as alterações no site, pois o último snapshot seria tirado se você fizesse a restauração por um snapshot.

Em uma implantação de vários servidores, é necessário copiar cada um dos componentes abordados anteriormente, usando diferentes mecanismos. Cada componente pode ter um requisito diferente de frequência de backup, por exemplo, a instalação e a configuração do OS e do WordPress mudam com muito menos frequência que o conteúdo gerado pelo usuário, por isso podem ser copiados com menos frequência sem perder dados no caso de uma recuperação.

Para fazer o backup da instalação e configuração do OS e serviços, além do código e da configuração da aplicação do WordPress, você pode criar uma AMI de uma instância do EC2 configurada corretamente. As AMIs podem agir com duas finalidades: como backup do estado da instância e como modelo ao executar novas instâncias.

Para fazer o backup do código e da configuração da aplicação do WordPress, é necessário usar as AMIs e os backups do Aurora.

Para fazer o backup de temas e plugins do WordPress instalados no site, faça o backup do bucket do Amazon S3 ou do sistema de arquivos do Amazon EFS onde eles estão armazenados.

- Para temas e plugins armazenados em um bucket do S3, é possível ativar a [Replicação entre regiões](#) para que todos os objetos carregados no bucket principal sejam replicados automaticamente para o bucket de backup em outra região AWS. A replicação entre regiões exige que o [versionamento](#) esteja ativado nos buckets de origem e de destino, o que fornece uma camada adicional de proteção e permite reverter para uma versão anterior de qualquer objeto específico no bucket.
- Para temas e plugins armazenados em um sistema de arquivos do EFS, é possível criar um AWS Data Pipeline para copiar dados do sistema de arquivos do EFS de produção para outro sistema de arquivos do EFS, conforme descrito na página de documentação [Fazer backup dos sistemas de arquivos do Amazon EFS](#). Também é possível fazer backup de um sistema de arquivos do EFS usando qualquer aplicação de backup já conhecida.
- Para fazer backup de uploads do usuário, siga as etapas descritas anteriormente para temas e plugins do WordPress.
- Para fazer backup do conteúdo do banco de dados, é necessário usar o [backup do Aurora](#). O Aurora faz backup do volume do cluster automaticamente e retém os dados de restauração durante o período de retenção de backup. Os backups do Aurora são contínuos e incrementais para que você possa restaurar rapidamente para qualquer ponto dentro do período de retenção de backup. Quando os dados do backup estão sendo gravados, não há nenhum impacto em relação à performance ou à interrupção de serviço do banco de dados. Você pode especificar um período de retenção de backup entre 1 e 35 dias. Você ainda pode criar [snapshots de bancos de dados manuais](#), que são mantidos até que você os exclua. Snapshots de bancos de dados manuais são úteis para backups e arquivamentos de longo prazo.

## Apêndice D: implantar novos plugins e temas

Poucos sites permanecem estáticos. Na maioria dos casos, você adicionará periodicamente temas e plugins do WordPress disponíveis publicamente ou atualiza para uma versão mais recente do WordPress. Em outros casos, você desenvolverá seus próprios temas e plugins do zero.

Sempre que você faz uma alteração estrutural na instalação do WordPress, existe certo risco de apresentar problemas não previstos. Pelo menos, faça um backup do código da aplicação, da configuração e do banco de dados antes de aplicar uma alteração relevante (como instalar um novo plugin). Para sites comerciais ou de outro valor, teste essas alterações em um ambiente de preparação separado primeiro. Com a AWS, é fácil replicar a configuração do ambiente de produção e executar todo o processo de implantação de maneira segura. Depois de terminar os testes, você pode simplesmente encerrar o ambiente de teste e parar de pagar os recursos. Posteriormente, este whitepaper aborda algumas considerações específicas do WordPress.

Alguns plugins gravam informações de configuração na tabela de banco de dados `wp_options` (ou apresentam alterações no esquema do banco de dados), enquanto outros criam arquivos de configuração no diretório de instalação do WordPress. Como movemos o banco de dados e o armazenamento para plataformas compartilhadas, essas alterações ficam imediatamente disponíveis a todas as instâncias em execução, sem qualquer trabalho da sua parte.

Ao implantar novos temas no WordPress, é necessário um pouco mais de trabalho. Se você só utiliza o Amazon EFS para armazenar todos os arquivos de instalação do WordPress, novos temas serão imediatamente disponibilizados a todas as instâncias em execução. Entretanto, se você está descarregando conteúdo estático no Amazon S3, será necessário processar uma cópia dele no local correto do bucket. Plugins como o W3 Total Cache são um jeito de iniciar manualmente essa tarefa. Como alternativa, você pode automatizar essa etapa como parte do processo de compilação.

Como os ativos de temas podem ser armazenados no CloudFront e no navegador, você precisa de um jeito de invalidar versões mais antigas quando for implantar as alterações. A melhor maneira de isso acontecer é incluindo algum tipo de identificador de versão em seu objeto. Esse identificador pode ser uma string de consulta com carimbo de data/hora ou uma string randômica. Se você usar o plug-in W3 Total Cache, pode atualizar uma string de consulta de mídia que esteja anexada a URLs dos arquivos de mídia.

# Avisos

Os clientes são responsáveis por fazer sua própria avaliação independente das informações neste documento. Este documento é: (a) fornecido apenas para fins informativos, (b) representa as ofertas e práticas de produtos atuais da AWS, que estão sujeitas a alterações sem aviso prévio e (c) não cria nenhum compromisso ou garantia da AWS e suas afiliadas, fornecedores ou licenciadores. Os produtos ou serviços da AWS são fornecidos no “estado em que se encontram”, sem garantias, declarações ou condições de qualquer tipo, explícitas ou implícitas. As responsabilidades e obrigações da AWS com seus clientes são regidas por contratos da AWS, e este documento não modifica nem faz parte de nenhum contrato entre a AWS e seus clientes.

© 2021, Amazon Web Services, Inc. ou suas afiliadas. Todos os direitos reservados.