



Whitepaper AWS

# Best practice per WordPress su AWS



---

# Best practice per WordPress su AWS: Whitepaper AWS

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in qualsiasi modo che possa causare confusione tra i clienti o in qualsiasi modo che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

---

# Table of Contents

Riassunto .....	1
Riassunto .....	1
Introduzione .....	2
Semplicità di implementazione .....	3
Considerazioni .....	3
Approcci disponibili .....	3
Amazon Lightsail .....	4
Selezione di un piano tariffario Amazon Lightsail .....	4
Installazione di WordPress .....	5
Ripristino dopo errori .....	5
Miglioramento delle prestazioni e ottimizzazione dei costi .....	7
Accelerazione della distribuzione di contenuti .....	7
Offload di contenuti statici .....	8
Contenuti dinamici .....	8
Caching di database .....	9
Caching bytecode .....	10
Distribuzione elastica .....	11
Architettura di riferimento .....	11
Dimensionamento del livello Web .....	13
Livello Web senza stato .....	14
Archiviazione condivisa (Amazon S3 e Amazon EFS) .....	15
Livello dati (Amazon Aurora e Amazon ElastiCache) .....	16
WordPress high availability by Bitnami su AWS Quick Start .....	17
Conclusione .....	20
Collaboratori .....	21
Revisioni del documento .....	22
Appendice A: Configurazione di CloudFront .....	23
Origini e comportamenti .....	23
Creazione di una distribuzione CloudFront .....	23
Appendice B: Installazione e configurazione di plug-in .....	27
Plug-in AWS per WordPress .....	27
Installazione e configurazione dei plug-in .....	27
Amazon CloudFront e AWS Certificate Manager .....	29
Traduzione e conversione in vocale dei contenuti .....	29

---

Riproduzione di podcast con Amazon Pollycast .....	32
Lettura dei contenuti tramite i dispositivi Amazon Alexa .....	32
Configurazione statica dei contenuti .....	33
Creazione di utenti IAM .....	33
Creazione di un bucket Amazon S3 .....	34
Creazione di origini statiche .....	35
Appendice C: Backup e ripristino .....	37
Appendice D: Implementazione di nuovi plug-in e temi .....	39
Avvisi .....	40

# Best practice per WordPress su AWS

Data di pubblicazione: 19 ottobre 2021 ([Revisioni del documento](#))

## Riassunto

Questo white paper offre agli amministratori di sistema linee guida specifiche su come iniziare a utilizzare WordPress su Amazon Web Services (AWS), oltre che su come ridurre i costi dell'implementazione e migliorare l'esperienza dell'utente finale. Definisce inoltre un'architettura di riferimento in grado di soddisfare i più comuni requisiti di scalabilità e di disponibilità elevata.

# Introduzione

WordPress è uno strumento open source di creazione di blog e un sistema di gestione dei contenuti (CMS) basato su PHP e MySQL adatto per creare sia blog personali sia siti Web a traffico elevato.

La prima versione di WordPress è stata rilasciata nel 2003, per cui non è stata creata secondo i criteri moderni di elasticità e scalabilità propri delle infrastrutture cloud. Grazie al lavoro della comunità WordPress e al rilascio di diversi moduli, le capacità di questa soluzione CMS sono in continua espansione. Oggi è possibile creare un'architettura WordPress in grado di sfruttare numerosi vantaggi di AWS Cloud.

# Semplicità di implementazione

Per la creazione di blog o siti Web a basso traffico senza rigorosi requisiti di alta disponibilità, potrebbe essere adatta una semplice implementazione a singolo server. Questa implementazione non rappresenta l'architettura più resiliente o scalabile, ma è il modo più rapido ed economico per rendere funzionante il proprio sito Web.

## Argomenti

- [Considerazioni](#)
- [Approcci disponibili](#)
- [Amazon Lightsail](#)

## Considerazioni

Questa discussione inizia con un'implementazione a singolo server Web. In alcuni casi, un'implementazione di questo tipo potrebbe non essere più sufficiente a soddisfare i propri requisiti di crescita, ad esempio:

- La macchina virtuale su cui viene implementato il sito Web WordPress rappresenta un singolo punto di errore. Un problema con questa istanza causa una perdita di servizio per il sito Web.
- È possibile ottenere il ridimensionamento delle risorse per migliorare le prestazioni solo con il "ridimensionamento verticale", ovvero aumentando le dimensioni della macchina virtuale su cui è in esecuzione il sito Web WordPress.

## Approcci disponibili

AWS offre diverse opzioni per il provisioning di macchine virtuali. Sono tre i modi principali in cui è possibile ospitare il sito Web WordPress su AWS:

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

[Amazon Lightsail](#) è un servizio che consente di avviare rapidamente un server virtuale privato (un'istanza Lightsail) per ospitare un sito Web WordPress. Lightsail è il modo più semplice per iniziare

se non occorrono tipi di istanze altamente configurabili o non è necessario accedere a funzionalità di rete avanzate.

[Amazon EC2](#) è un servizio Web che fornisce capacità di elaborazione ridimensionabile in modo da poter avviare un server virtuale in pochi minuti. Amazon EC2 offre più opzioni di configurazione e gestione rispetto a Lightsail, per cui è un'opzione adatta per architetture più avanzate. Offre l'accesso amministrativo alle istanze EC2 e consente di installare qualsiasi pacchetto software desiderato, incluso WordPress.

[Marketplace AWS](#) è un negozio online in cui è possibile trovare, acquistare e implementare rapidamente software in esecuzione su AWS. È possibile utilizzare l'implementazione 1-Click per avviare immagini WordPress preconfigurate direttamente su Amazon EC2 nel proprio account AWS in pochi minuti. Nel Marketplace sono presenti numerosi fornitori che offrono istanze WordPress pronte per essere eseguite.

In questo white paper viene discussa l'opzione Lightsail come implementazione consigliata per un sito Web WordPress a server singolo.

## Amazon Lightsail

Per gli sviluppatori, le piccole aziende, gli studenti e altri utenti che necessitano di una soluzione semplice per server virtuali privati (VPS), Lightsail è il modo più semplice per iniziare a utilizzare AWS.

Il servizio astrae per l'utente numerosi degli elementi più complessi della gestione dell'infrastruttura. Pertanto, rappresenta un punto di partenza ideale per chi ha meno esperienza con l'infrastruttura o deve concentrarsi sull'esecuzione del proprio sito Web e un prodotto semplificato è sufficiente per soddisfare le sue esigenze.

Amazon Lightsail consente di scegliere il sistema operativo Windows o Linux/Unix e le applicazioni Web più diffuse, tra cui WordPress, e implementarli con un solo clic da modelli preconfigurati.

Quando le proprie esigenze aumentano, è possibile utilizzare ulteriori servizi di database, archiviazione di oggetti, chaching e distribuzione di contenuti di AWS.

## Selezione di un piano tariffario Amazon Lightsail

Un [piano Lightsail](#) definisce il costo mensile delle risorse Lightsail utilizzate per ospitare il sito Web WordPress. Sono disponibili piani per vari casi d'uso, con diversi livelli di risorse della CPU, memoria,



archiviazione su unità a stato solido (SSD) e trasferimento dati. Nel caso di un sito Web complesso, potrebbe essere necessaria un'istanza di dimensioni maggiori dotata di più risorse. Per ottenerla, occorre eseguire la migrazione del server a un piano più grande [utilizzando la console Web](#) o come descritto nella [documentazione di Amazon Lightsail CLI](#).

## Installazione di WordPress

Lightsail fornisce modelli per applicazioni di uso comune come WordPress. Questo modello è un ottimo punto di partenza per la gestione del sito Web WordPress, in quanto include già preinstallata la maggior parte del software necessario. È possibile installare software aggiuntivo o personalizzare la configurazione del software utilizzando il terminale nel browser o il proprio client SSH, oppure tramite l'interfaccia Web di amministrazione di WordPress.

Amazon Lightsail ha siglato una partnership con il prodotto Pro Sites di GoDaddy, che supporta i clienti di WordPress nella gestione delle istanze in modo semplice e gratuito. I server virtuali WordPress di Lightsail sono preconfigurati e ottimizzati per offrire rapidità e sicurezza, permettendo, in modo semplice, di creare in modo immediato un sito WordPress perfettamente funzionante. I clienti che utilizzano più istanze di WordPress trovano che l'aggiornamento, la manutenzione e la gestione di tutti i loro siti siano estremamente complicati e richiedano troppo tempo. Questa integrazione consente di gestire facilmente più istanze di WordPress in pochi minuti e con pochi clic.

Ulteriori informazioni sulla gestione di WordPress su Lightsail sono disponibili nel documento [Nozioni di base sull'utilizzo di WordPress dalla tua istanza Amazon Lightsail](#). Una volta completata la personalizzazione del sito Web WordPress, è consigliabile eseguire una snapshot dell'istanza.

Una [snapshot](#) rappresenta un modo per creare un'immagine di backup della propria istanza Lightsail. È una copia del disco di sistema e memorizza anche la configurazione del computer originale (ovvero memoria, CPU, dimensione del disco e velocità di trasferimento dei dati). È possibile utilizzare uno snapshot per ripristinare una configurazione funzionante dopo un'implementazione o un aggiornamento contenente errori.

Questo snapshot consente di ripristinare il server, se necessario, oltre che avviare nuove istanze con le stesse personalizzazioni.

## Ripristino dopo errori

Un singolo server Web rappresenta un singolo punto di errore, per cui occorre assicurarsi di sottoporre a backup i dati del proprio sito Web. A tale scopo è possibile utilizzare le snapshot con le modalità descritte in precedenza. In caso di errori, è possibile ripristinare una nuova istanza da quella

più recente. Per ridurre la quantità di dati che potrebbero andare perduti durante un ripristino, occorre utilizzare gli snapshot più recenti possibile.

Per ridurre al minimo il rischio di perdita di dati, eseguire regolarmente snapshot. È possibile programmare l'esecuzione automatica di snapshot delle istanze Lightsail Linux/Unix. La procedura è disponibile nel documento [Abilitazione o disabilitazione di snapshot automatici per istanze o dischi in Amazon Lightsail](#).

AWS consiglia di utilizzare un IP statico, ovvero un indirizzo IP fisso e pubblico dedicato al proprio account Lightsail. Se si decide di sostituire l'istanza, è possibile riassegnare l'IP statico alla nuova istanza. In questo modo, non occorre riconfigurare eventuali sistemi esterni (come i record DNS) da puntare a un nuovo IP ogni volta che si desidera sostituire l'istanza.

# Miglioramento delle prestazioni e ottimizzazione dei costi

Qualora l'implementazione a server singolo non sia più sufficiente a soddisfare le esigenze di crescita, potrebbe essere necessario valutare opzioni per migliorare le prestazioni del sito Web. Prima di passare a un'implementazione scalabile multi-server (discussa più avanti in questo documento), è possibile applicare una serie di efficienze in termini di prestazioni e costi. Si tratta di pratiche che dovrebbero essere seguite comunque, anche se si passa a un'architettura multi-server.

Le sezioni seguenti illustrano diverse opzioni che possono migliorare aspetti delle prestazioni e della scalabilità di un sito Web WordPress. Alcune di esse possono essere applicate a una distribuzione a server singolo, mentre altre sfruttano la scalabilità di più server. Molte di queste modifiche richiedono l'utilizzo di uno o più plug-in di WordPress. Tra le varie opzioni disponibili, [W3 Total Cache](#) è una scelta popolare che combina numerose di queste modifiche in un unico plug-in.

## Argomenti

- [Accelerazione della distribuzione di contenuti](#)
- [Caching di database](#)
- [Caching bytecode](#)

## Accelerazione della distribuzione di contenuti

Qualsiasi sito Web WordPress deve fornire contenuti sia statici che dinamici. I contenuti statici includono immagini, file JavaScript o fogli di stile. I contenuti dinamici includono tutto ciò che viene generato sul lato server utilizzando il codice PHP di WordPress, ad esempio, elementi del proprio sito generati dal database o personalizzati per ciascun visualizzatore.

Un aspetto importante dell'esperienza dell'utente finale è la latenza di rete dovuta alla distribuzione dei contenuti precedenti agli utenti di tutto il mondo. L'accelerazione della distribuzione dei contenuti precedenti migliora l'esperienza degli utenti finali, in particolare di quelli dislocati nel mondo. Questo risultato può essere ottenuto con una rete per la distribuzione di contenuti (CDN, Content Delivery Network) come Amazon CloudFront.

[Amazon CloudFront](#) è un servizio Web che offre un modo semplice ed economico per distribuire contenuti con bassa latenza e velocità di trasferimento dati elevate in diverse posizioni edge nel mondo. Le richieste dei visualizzatori vengono instradate automaticamente in una [posizione edge](#) di CloudFront adatta per ridurre la latenza. Se il contenuto può essere memorizzato nella cache

(per pochi secondi, minuti o addirittura giorni) ed è già archiviato in una particolare posizione edge, CloudFront lo distribuisce immediatamente. Se il contenuto non deve essere memorizzato nella cache, è scaduto o non si trova in quella posizione edge, CloudFront lo recupera da una o più fonti di attendibilità, indicate come origini (in questo caso, l'istanza Lightsail) nella configurazione di CloudFront. Tale recupero avviene tramite connessioni di rete ottimizzate, che accelerano la distribuzione di contenuti sul sito Web. Oltre a migliorare l'esperienza dell'utente finale, il modello discusso riduce anche il carico sui server di origine ed è potenzialmente in grado di offrire notevoli risparmi sui costi.

## Offload di contenuti statici

Sono inclusi file CSS, JavaScript e immagini, sia che si tratti di temi WordPress o di file multimediali caricati dagli amministratori dei contenuti. Tutti questi file possono essere archiviati in Amazon Simple Storage Service (Amazon S3) utilizzando un plug-in come W3 Total Cache e possono essere forniti agli utenti in modo scalabile e con disponibilità elevata. [Amazon S3](#) offre un'infrastruttura di storage dei dati economica, altamente scalabile, affidabile e a bassa latenza, accessibile tramite API REST. Amazon S3 archivia in modo ridondante gli oggetti non solo su più dispositivi, ma anche su più strutture all'interno di una regione AWS, offrendo così livelli di durata eccezionalmente elevati.

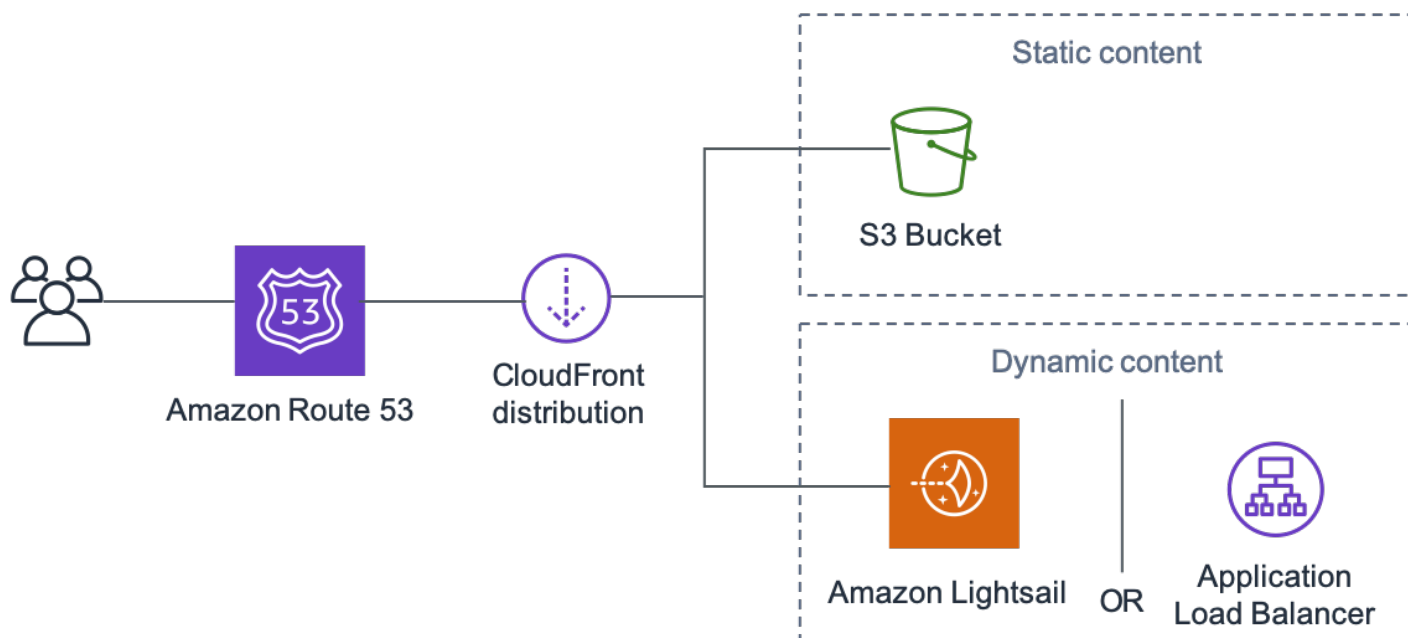
Ciò ha come effetto secondario positivo l'offloading del carico di lavoro dall'istanza Lightsail, che a sua volta consente di dedicare maggiore attenzione alla generazione dinamica di contenuti. Questo riduce il carico sul server e rappresenta un importante passo avanti verso la creazione di un'architettura senza stato (necessaria per l'implementazione del ridimensionamento automatico).

In seguito è possibile configurare Amazon S3 come origine per CloudFront per migliorare la distribuzione di tali risorse statiche agli utenti di tutto il mondo. Sebbene WordPress non sia integrato con Amazon S3 e CloudFront, sono disponibili diversi plug-in che aggiungono il supporto di questi servizi (ad esempio, W3 Total Cache).

## Contenuti dinamici

I contenuti dinamici includono l'output di script PHP WordPress lato server. I contenuti dinamici possono essere forniti anche tramite CloudFront configurando il sito Web WordPress come origine. Poiché i contenuti dinamici includono contenuti personalizzati, è necessario configurare CloudFront per l'inoltro di determinati cookie HTTP e specifiche intestazioni HTTP nell'ambito di una richiesta al server di origine personalizzato. CloudFront utilizza i valori dei cookie inoltrati come parte della chiave che identifica un oggetto univoco nella sua cache. Per ottimizzare l'efficienza del caching, è necessario configurare CloudFront in modo che inoltri solo i cookie HTTP e le intestazioni HTTP

che cambiano realmente il contenuto (non i cookie che vengono utilizzati solo sul lato client o da applicazioni di terze parti, ad esempio per l'analisi dei dati Web).



## Distribuzione dell'intero sito Web tramite Amazon CloudFront

La figura precedente include due origini: una per i contenuti statici e l'altra per i contenuti dinamici. Maggiori informazioni sull'implementazione sono disponibili nell'[Appendice A: Configurazione di CloudFront](#) e nell'[Appendice B: Installazione e configurazione di plug-in](#).

CloudFront utilizza intestazioni di controllo della cache standard per identificare se e per quanto tempo deve memorizzare nella cache risposte HTTP specifiche. Le stesse intestazioni di controllo della cache vengono utilizzate anche dai browser Web per decidere quando e per quanto tempo memorizzare i contenuti localmente nella cache, in modo da offrire all'utente finale un'esperienza ottimale (ad esempio, un file `.css` già scaricato non verrà scaricato nuovamente ogni volta che un visitatore non nuovo visualizza una pagina). È possibile configurare le intestazioni di controllo della cache a livello del server Web (ad esempio, tramite file `.htaccess` o modifiche del file `httpd.conf`) o installare un plug-in di WordPress (ad esempio, W3 Total Cache) per stabilire in che modo tali intestazioni vengono impostate per i contenuti statici e dinamici.

## Caching di database

Il caching di database può ridurre significativamente la latenza e aumentare il throughput per i carichi di lavoro di applicazioni con numerose operazioni in lettura come WordPress. Le prestazioni delle applicazioni vengono migliorate grazie all'accesso a bassa latenza consentito dall'archiviazione nella

memoria dei dati utilizzati di frequente (ad esempio, i risultati delle query del database che richiedono numerose operazioni di I/O). Quando una percentuale elevata delle query viene fornita dalla cache, il numero di query che devono raggiungere il database è inferiore, con conseguente riduzione dei costi associati all'esecuzione del database.

Sebbene WordPress abbia capacità di caching limitate, vari plug-in consentono di aggiungere il supporto dell'integrazione con [Memcached](#), un sistema di caching degli oggetti in memoria ampiamente diffuso. Il plug-in W3 Total Cache ne è un buon esempio.

Negli scenari più semplici, è sufficiente installare Memcached sul server Web e acquisire il risultato sotto forma di nuova snapshot. In questo caso, l'utente è responsabile delle attività amministrative associate all'esecuzione di una cache.

Un'altra opzione è sfruttare un servizio gestito come [Amazon ElastiCache](#) ed evitare questo onere operativo. ElastiCache rende semplice implementare, utilizzare e dimensionare una cache in memoria distribuita nel cloud. Informazioni su come connettersi ai nodi del cluster ElastiCache sono disponibili nella [documentazione di Amazon ElastiCache](#).

Se si utilizza Lightsail e si desidera accedere privatamente a un cluster ElastiCache nel proprio account AWS, è possibile farlo utilizzando il peering VPC. Le istruzioni su come abilitare il peering VPC sono disponibili in [Configura il peering di Amazon VPC in modo che funzioni con risorse AWS esterne ad Amazon Lightsail](#).

## Caching bytecode

Ogni volta che viene eseguito, uno script PHP viene analizzato e compilato. Usando una cache bytecode PHP, l'output della compilazione PHP viene memorizzato nella RAM in modo che lo stesso script non debba essere compilato più volte. Ciò riduce il sovraccarico relativo all'esecuzione di script PHP, con conseguente miglioramento delle prestazioni e riduzione dei requisiti della CPU.

Una cache bytecode può essere installata su qualsiasi istanza Lightsail che ospita WordPress e può ridurne notevolmente il carico. Per PHP 5.5 e versioni successive, AWS consiglia l'uso di [OPcache](#), un'estensione offerta in bundle con quella versione di PHP.

Nota: OPcache è abilitato per impostazione predefinita nel modello Bitnami WordPress Lightsail, per cui non è necessario intraprendere ulteriori azioni.

## Distribuzione elastica

Sono numerosi gli scenari in cui una distribuzione a server singolo potrebbe non essere sufficiente per il proprio sito Web. In queste situazioni, è necessaria un'architettura multi-server e scalabile.

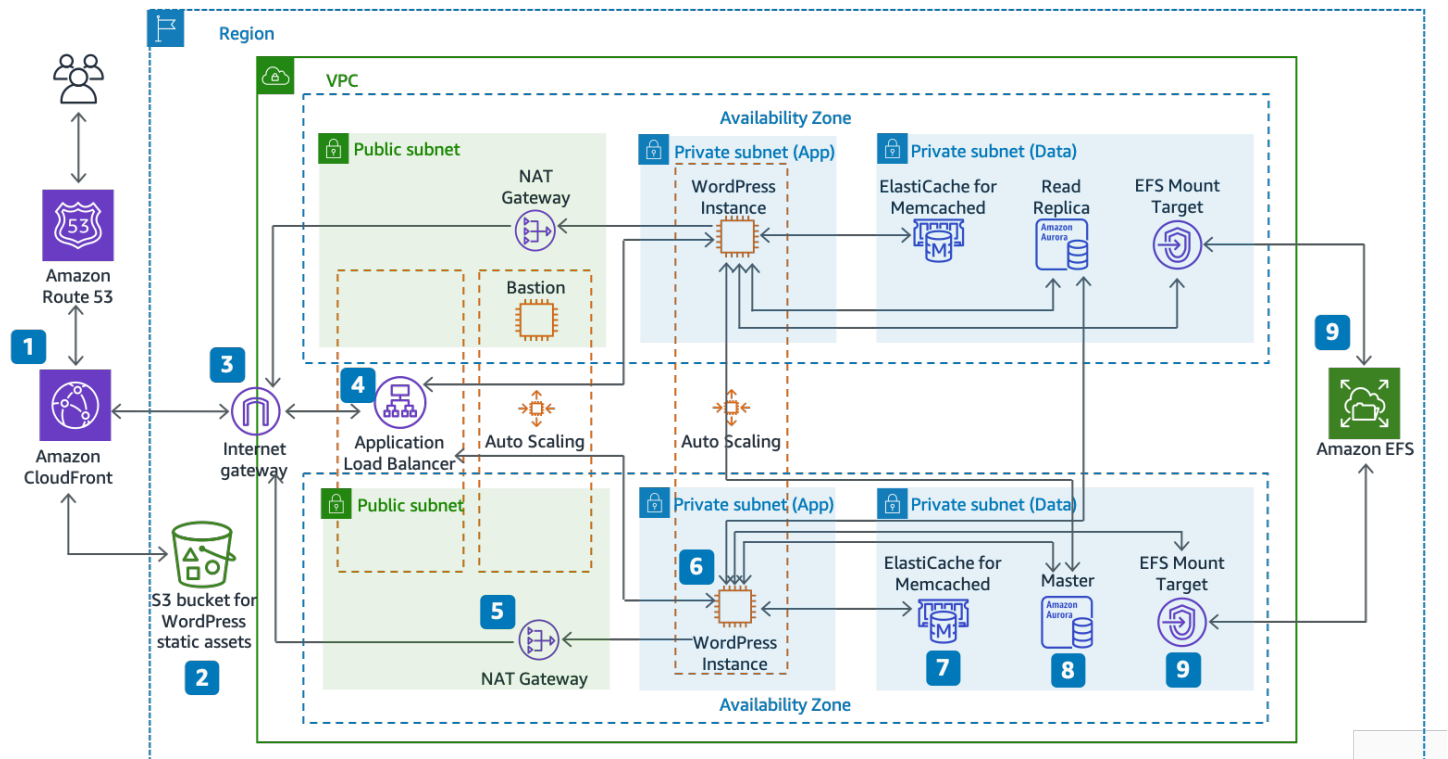
### Argomenti

- [Architettura di riferimento](#)
- [Dimensionamento del livello Web](#)
- [Livello Web senza stato](#)
- [WordPress high availability by Bitnami su AWS Quick Start](#)

## Architettura di riferimento

L'[architettura di riferimento per l'hosting di WordPress su AWS](#) disponibile su GitHub indica le best practice per l'implementazione di WordPress su AWS e include una serie di modelli AWS CloudFormation per renderla subito operativa. La seguente architettura si basa su tale architettura di riferimento. Nel resto di questa sezione esamineremo le ragioni alla base delle scelte dell'architettura.

L'AMI nel GitHub è stata modificata da Amazon Linux1 ad Amazon Linux2 a luglio 2021. Tuttavia, i modelli di implementazione su S3 non sono stati ancora modificati. Si consiglia di utilizzare i modelli su GitHub in caso di problemi con l'implementazione dell'architettura di riferimento con i modelli su S3.



## Architettura di riferimento per l'hosting di WordPress su AWS

### Componenti dell'architettura

L'architettura di riferimento mostra l'implementazione completa delle best practice per un sito Web WordPress su AWS.

- Inizia con la memorizzazione nella cache perimetrale in Amazon CloudFront (1) dei contenuti vicini agli utenti finali, per consentirne una distribuzione più rapida.
- CloudFront estrae il contenuto statico da un bucket S3 (2) e il contenuto dinamico da un Application Load Balancer (4) davanti alle istanze Web.
- Le istanze Web vengono eseguite in un gruppo Auto Scaling di istanze Amazon EC2 (6).
- Un cluster ElastiCache (7) memorizza nella cache i dati più frequentemente oggetto di query, in modo da velocizzare i tempi di risposta.

Un'istanza MySQL di Amazon Aurora (8) ospita il database di WordPress.

- Le istanze EC2 di WordPress accedono ai dati condivisi di WordPress su un file system Amazon EFS tramite un target di montaggio di EFS (9) in ciascuna zona di disponibilità.
- Un gateway Internet (3) consente la comunicazione tra le risorse all'interno del VPC e su Internet.



- I gateway NAT (5) in ciascuna zona di disponibilità consentono alle istanze EC2 in sottoreti private (app e dati) di accedere a Internet.

All'interno di Amazon VPC sono presenti due tipi di sottoreti: pubbliche (sottoreti pubbliche) e private (sottoreti di app e sottoreti di dati). Le risorse implementate nelle sottoreti pubbliche riceveranno un indirizzo IP pubblico e saranno visibili pubblicamente su Internet. Qui vengono implementati l'Application Load Balancer (4) e un host Bastion per l'amministrazione. Le risorse implementate nelle sottoreti private ricevono solo un indirizzo IP privato, pertanto non sono visibili pubblicamente su Internet e garantiscono un livello di sicurezza maggiore. Le istanze del server Web WordPress (6), le istanze cluster ElastiCache (7), le istanze del database Aurora MySQL (8) e i target di montaggio di EFS (9) sono tutti implementati in sottoreti private.

Nel resto di questa sezione tratteremo ciascuna di queste considerazioni in modo più dettagliato.

## Dimensionamento del livello Web

Per trasformare l'architettura a server singolo in un'architettura scalabile multi-server, sono necessari cinque componenti principali:

- Istanze Amazon EC2
- Amazon Machine Image (AMI)
- Sistemi di bilanciamento del carico
- Scalabilità automatica
- Controlli dell'integrità

AWS offre un'ampia gamma di tipi di istanze EC2 per garantire la migliore configurazione di server in termini di prestazioni e costi. In generale, il tipo di istanza ottimizzato per il calcolo (ad esempio, C4) può essere una buona scelta per un server Web WordPress. Consente l'implementazione delle istanze su più zone di disponibilità all'interno di una regione AWS per aumentare l'affidabilità dell'architettura complessiva.

Grazie al controllo completo dell'istanza EC2, è possibile eseguire l'accesso root per installare e configurare tutti i componenti software necessari per l'esecuzione di un sito Web WordPress. Una volta terminato, è possibile salvare la configurazione come AMI e utilizzarla per avviare nuove istanze con le relative personalizzazioni.

Per distribuire le richieste degli utenti finali a più nodi del server Web, è necessaria una soluzione di bilanciamento del carico. AWS fornisce questa funzionalità tramite [Elastic Load Balancing](#), un servizio ad alta disponibilità che distribuisce il traffico a più istanze EC2. Poiché il tuo sito Web fornisce contenuti agli utenti tramite HTTP o HTTPS, è consigliabile utilizzare Application Load Balancer, un sistema di bilanciamento del carico a livello di applicazione con routing dei contenuti e la possibilità di eseguire più siti Web WordPress su domini diversi, se necessario.

Elastic Load Balancing supporta la distribuzione di richieste su più zone di disponibilità all'interno di una regione AWS. È inoltre possibile configurare un controllo dell'integrità, in modo che Application Load Balancer interrompa automaticamente l'invio di traffico alle singole istanze che hanno generato errori (ad esempio, a causa di un problema hardware o di un arresto anomalo del software). AWS consiglia di utilizzare la pagina di accesso dell'amministratore di WordPress (`/wp-login.php`) per il controllo dell'integrità, che consente di verificare che il server Web sia in esecuzione e che sia configurato correttamente per fornire file PHP.

È possibile creare una pagina di controllo dell'integrità personalizzata per la verifica di altre risorse dipendenti, come quelle del database e della cache. Per ulteriori informazioni, consultare la sezione [Controlli dello stato per i gruppi target](#) nella guida relativa all'Application Load Balancer.

L'elasticità è una caratteristica fondamentale di AWS Cloud. All'occorrenza è possibile utilizzare una maggiore capacità di calcolo (ad esempio, mediante più server Web) o ridurla. [AWS Auto Scaling](#) è un servizio AWS che aiuta ad automatizzare il provisioning per dimensionare la capacità di Amazon EC2 in base a condizioni definite senza interventi manuali. È possibile configurare AWS Auto Scaling in modo che il numero di istanze EC2 utilizzate aumenti durante i picchi della domanda per mantenere il livello di prestazioni desiderato e diminuisca automaticamente nei periodi di traffico meno intenso, in modo da contenere al massimo i costi.

Elastic Load Balancing supporta inoltre l'aggiunta e la rimozione dinamica di host Amazon EC2 dalla rotazione del bilanciamento del carico. Lo stesso Elastic Load Balancing aumenta e riduce in modo dinamico la capacità di bilanciamento del carico in base alle esigenze del traffico senza alcun intervento manuale.

## Livello Web senza stato

Per sfruttare più server Web in una configurazione di scalabilità automatica, il livello Web deve essere senza stato. Un'applicazione senza stato non richiede alcuna conoscenza delle interazioni precedenti e non memorizza informazioni sulla sessione. Nel caso di WordPress, ciò significa che tutti gli utenti finali ricevono la stessa risposta, indipendentemente dal server Web che ha elaborato

la loro richiesta. Un'applicazione senza stato può ridimensionarsi orizzontalmente, poiché qualsiasi richiesta può essere soddisfatta da una qualunque delle risorse di calcolo (o istanze di server Web) disponibili. Quando tale capacità non è più necessaria, ogni risorsa può essere terminata in modo sicuro una volta esaurite le attività in esecuzione. Queste risorse non devono rilevare la presenza di peer: occorre soltanto un modo per distribuire loro il carico di lavoro.

Quando si tratta di archiviare i dati delle sessioni degli utenti, il core di WordPress è del tutto senza stato, poiché si basa su cookie che vengono memorizzati nel browser Web del client. L'archiviazione delle sessioni non dovrebbe creare particolari problemi, a meno che non sia stato installato codice personalizzato (ad esempio, un plug-in di WordPress) basato su sessioni PHP native.

WordPress è stato originariamente progettato per l'esecuzione su un singolo server. Di conseguenza, memorizza alcuni dati sul file system locale del server. L'esecuzione di WordPress in una configurazione multi-server può rappresentare un problema perché crea un'incongruenza tra i server Web. Ad esempio, se un utente carica una nuova immagine, questa viene archiviata solo su uno dei server.

Questo è uno dei motivi per cui è necessario migliorare la configurazione di esecuzione predefinita di WordPress spostando i dati importanti nell'archiviazione condivisa. L'architettura per l'implementazione delle best practice include un database come livello separato all'esterno del server Web e utilizza l'archiviazione condivisa per memorizzare temi, plug-in e caricamenti degli utenti.

## Archiviazione condivisa (Amazon S3 e Amazon EFS)

Per impostazione predefinita, WordPress archivia i caricamenti degli utenti sul file system locale, per cui non è senza stato. È quindi necessario trasferire l'installazione di WordPress e tutte le personalizzazioni dell'utente (ad esempio, configurazioni, plug-in, temi e caricamenti generati dall'utente) in una piattaforma di dati condivisa per contribuire a ridurre il carico sui server Web e per rendere il livello Web senza stato.

[Amazon Elastic File System](#) (Amazon EFS) fornisce file system di rete scalabili da utilizzare con le istanze EC2. I file system di Amazon EFS vengono distribuiti su un numero illimitato di server di archiviazione, perciò i file system possono crescere in modo elastico e consentono un numero estremamente elevato di accessi in parallelo da parte delle istanze EC2. La natura distribuita di Amazon EFS permette di evitare i colli di bottiglia e le limitazioni proprie dei file server tradizionali.

Trasferendo l'intera directory di installazione di WordPress su un file system EFS e montandola in ciascuna delle istanze EC2 all'avvio, il tuo sito WordPress e tutti i relativi dati vengono automaticamente archiviati su un file system distribuito che non dipende da nessuna istanza EC2,

rendendo il livello Web del tutto senza stato. Il vantaggio di questa architettura è che non richiede l'installazione di plug-in e temi ad ogni avvio di nuove istanze; inoltre, consente di velocizzare notevolmente l'installazione e il ripristino delle istanze di WordPress e semplifica l'implementazione di modifiche dei plug-in e dei temi in WordPress, come descritto nella sezione [Considerazioni sull'implementazione](#) di questo documento.

Per garantire prestazioni ottimali del sito Web quando viene eseguito da un file system EFS, occorre controllare le impostazioni di configurazione consigliate per Amazon EFS e OPcache nell'[architettura di riferimento di AWS per WordPress](#).

È anche possibile scaricare tutte le risorse statiche, come immagini, CSS e file JavaScript, in un bucket S3 con davanti il caching CloudFront. Il meccanismo per eseguire questa operazione in un'architettura multi-server è esattamente lo stesso di un'architettura a server singolo discusso nella sezione [Contenuti statici](#) di questo white paper. I vantaggi sono gli stessi dell'architettura a server singolo: è possibile scaricare il lavoro associato alla fornitura delle risorse statiche su Amazon S3 e CloudFront, consentendo ai server Web di dedicarsi esclusivamente alla generazione di contenuti dinamici e di soddisfare un maggior numero di richieste degli utenti per server Web.

## Livello dati (Amazon Aurora e Amazon ElastiCache)

Con l'installazione di WordPress archiviata su un file system di rete distribuito, scalabile e condiviso e le risorse statiche fornite da Amazon S3, è possibile concentrare l'attenzione sul componente con stato rimanente: il database. Come per il livello archiviazione, il database non dovrebbe dipendere da un singolo server, per cui non può essere ospitato su uno dei server Web. Il database di WordPress deve invece essere ospitato su Amazon Aurora.

[Amazon Aurora](#) è un database relazionale compatibile con MySQL e PostgreSQL creato per il cloud che unisce le prestazioni e la disponibilità dei database commerciali di fascia alta alla semplicità e al costo ridotto dei database open source. Aurora MySQL potenzia le prestazioni e la disponibilità di MySQL mediante la stretta integrazione tra il motore di database e un sistema di archiviazione distribuito appositamente progettato supportato da SSD. È tollerante ai guasti e si ripara automaticamente, replica sei copie dei dati in tre zone di disponibilità, è progettato per garantire una disponibilità superiore al 99,99% ed esegue continuamente il backup dei dati in Amazon S3. Amazon Aurora è stato sviluppato in modo da rilevare automaticamente gli arresti anomali dei database e riavviarsi automaticamente senza la necessità di alcuna operazione di ripristino o di creazione di una nuova cache del database.

Amazon Aurora offre vari [tipi di istanze](#) per adattarsi a diversi profili applicativi, tra cui istanze ottimizzate per la memoria ed espandibili. Per migliorare le prestazioni del database, è possibile selezionare un tipo di istanza di grandi dimensioni che fornisca più risorse di CPU e memoria.

Amazon Aurora gestisce automaticamente il failover tra l'istanza primaria e le [repliche di Aurora](#), in modo che le applicazioni possano ripristinare le operazioni del database il più rapidamente possibile senza interventi manuali di tipo amministrativo. Il failover richiede in genere meno di 30 secondi.

Dopo aver creato almeno una replica di Aurora, occorre collegarsi all'istanza primaria utilizzando l'endpoint cluster per consentire il failover automatico dell'applicazione in caso di errore dell'istanza primaria. È possibile creare fino a 15 repliche di lettura a bassa latenza in tre zone di disponibilità.

Man mano che il database si ridimensiona, anche la cache del database deve scalare. Come discusso in precedenza nella sezione [Caching di database](#), per garantire una maggiore disponibilità, ElastiCache offre funzionalità per ridimensionare la cache su più nodi in un cluster ElastiCache e su più zone di disponibilità in una regione. Quando si ridimensiona il cluster ElastiCache, occorre assicurarsi di configurare il plug-in di caching per la connessione utilizzando l'endpoint di configurazione, in modo che WordPress possa utilizzare i nuovi nodi del cluster man mano che vengono aggiunti e smetta di utilizzare quelli che vengono rimossi. È inoltre necessario configurare i server Web per l'utilizzo del [client del cluster ElastiCache per PHP](#) e aggiornare l'AMI per archiviare questa modifica.

## WordPress high availability by Bitnami su AWS Quick Start

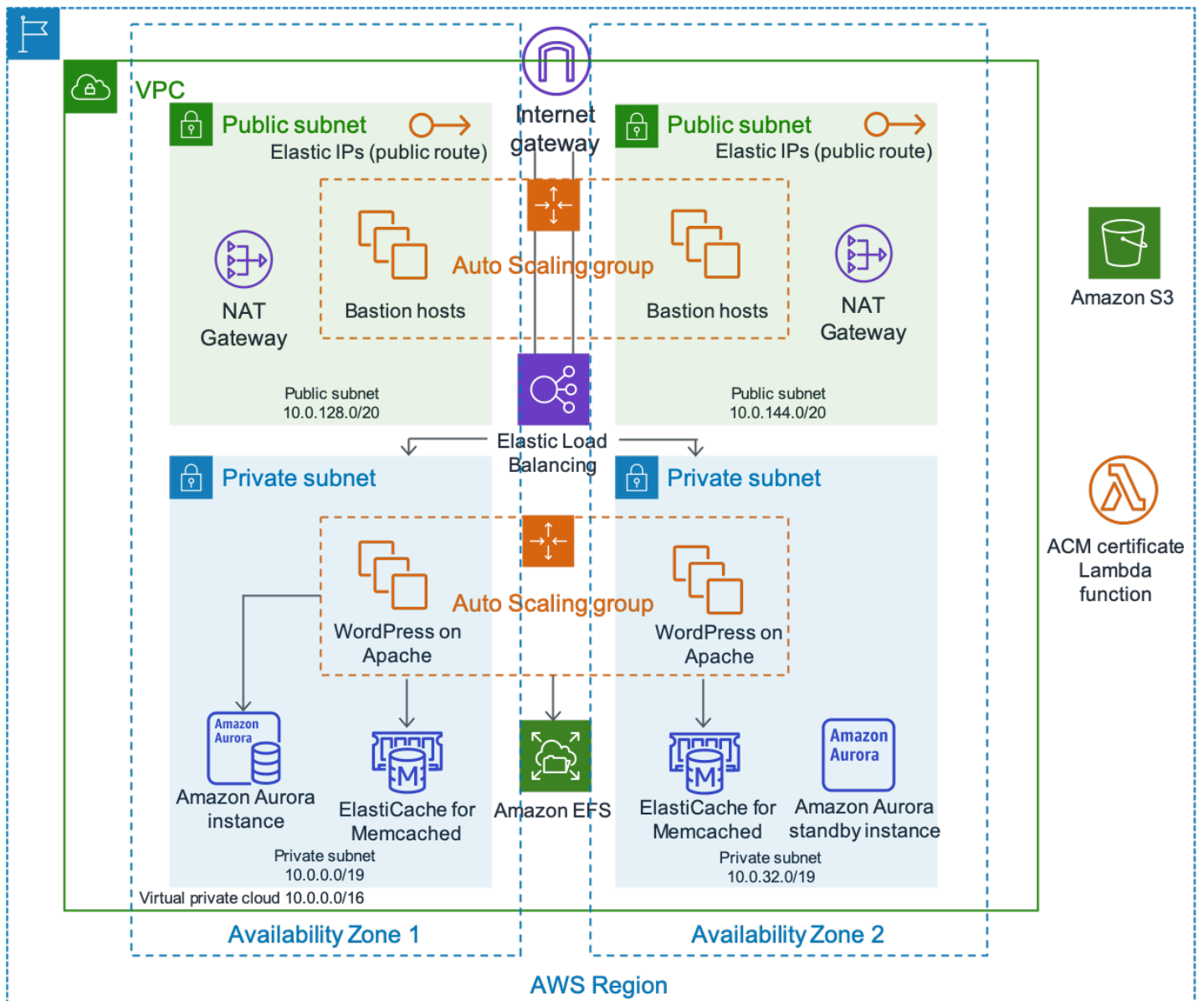
I Quick Start sono creati da partner e Solutions Architect di AWS per aiutare a implementare le tecnologie più utilizzate in AWS, secondo le best practice di sicurezza e disponibilità elevata. Questi acceleratori riducono centinaia di procedure manuali a pochi passaggi, consentendo la creazione rapida di un ambiente di produzione subito pronto per essere utilizzato. Ciascun Quick Start include modelli di AWS CloudFormation che permettono di automatizzare l'implementazione, oltre a una guida che descrive l'architettura e fornisce istruzioni dettagliate di tale implementazione.

[WordPress High Availability by Bitnami su AWS Quick Start](#) consente l'impostazione del seguente ambiente configurabile su AWS:

- Un'architettura ad alta disponibilità che si estende su due zone di disponibilità.\*
- Un cloud privato virtuale (VPC, Virtual Private Cloud) configurato con sottoreti pubbliche e private in conformità con le best practice di AWS. Questo ambiente offre l'infrastruttura di rete per l'implementazione.\*

- Un gateway Internet per l'accesso a Internet, utilizzato dai bastion host per inviare e ricevere traffico.\*
- Nelle sottoreti pubbliche, gateway NAT gestiti per consentire accesso a Internet in uscita per le risorse nelle sottoreti private.\*
- Nelle sottoreti pubbliche, un bastion host di Linux in un gruppo Auto Scaling per consentire accesso SSH (Secure Shell) in entrata alle istanze EC2 all'interno delle sottoreti pubbliche e private.\*
- Elastic Load Balancing per distribuire le richieste HTTP e HTTPS su più istanze WordPress.
- Nelle sottoreti private, istanze EC2 che ospitano l'applicazione WordPress su Apache. Queste istanze sono assegnate a un gruppo Auto Scaling per garantire una disponibilità elevata.
- Nelle sottoreti private, istanze di database Amazon Aurora amministrato da Amazon Relational Database Service (Amazon RDS).
- Nelle sottoreti private, Amazon Elastic File System (Amazon EFS) per condividere risorse (ad esempio, plug-in, temi e immagini) fra le istanze WordPress.
- Nelle sottoreti private, nodi di Amazon ElastiCache per Memcached per eseguire il caching delle query di database.

\* Il modello implementato da Quick Start su un VPC esistente non include le attività contrassegnate da un asterisco e richiede l'uso della configurazione del VPC esistente.



## Architettura di WordPress High Availability by Bitnami

Una descrizione dettagliata dell'implementazione di WordPress High Availability by Bitnami su AWS esula dall'ambito di questo documento. Per la configurazione e le opzioni disponibili, consultare [WordPress High Availability by Bitnami su AWS](#).

## Conclusione

AWS presenta numerose opzioni in termini di architettura per l'esecuzione di WordPress. L'opzione più semplice è l'installazione di un singolo server per siti Web con traffico ridotto. Per i siti Web più avanzati, gli amministratori del sito possono aggiungere diverse altre opzioni, ognuna delle quali rappresenta un miglioramento incrementale in termini di disponibilità e scalabilità. Gli amministratori possono selezionare le funzionalità più adatte alle loro esigenze e al loro budget.



# Collaboratori

I collaboratori di questo documento includono:

- Paul Lewis, Solutions Architect, Amazon Web Services
- Ronan Guilfoyle, Solutions Architect, Amazon Web Services
- Andreas Chatzakis, Solutions Architect Manager, Amazon Web Services
- Jibril Touzi, Technical Account Manager, Amazon Web Services
- Hakmin Kim, Migration Partner Solutions Architect, Amazon Web Services

## Revisioni del documento

Per ricevere una notifica sugli aggiornamenti di questo whitepaper, iscriversi al feed RSS.

update-history-change	update-history-description	update-history-date
<a href="#">Whitepaper aggiornato</a>	Aggiornamento alle sezioni Architettura di riferimento e Plug-in AWS per WordPress.	19 ottobre 2021
<a href="#">Whitepaper aggiornato</a>	Aggiornamento che include nuovi approcci di implementazione e il plug-in AWS per WordPress.	30 ottobre 2019
<a href="#">Whitepaper aggiornato</a>	Aggiornamento che offre delucidazioni sui messaggi relativi al prodotto Amazon Aurora.	1 febbraio 2018
<a href="#">Whitepaper aggiornato</a>	Aggiornamento che include i servizi AWS pubblicati dalla prima pubblicazione.	1 dicembre 2017
<a href="#">Pubblicazione iniziale</a>	Prima pubblicazione.	1 dicembre 2014

## Appendice A: Configurazione di CloudFront

Per ottenere prestazioni ed efficienza ottimali quando si utilizza Amazon CloudFront con il proprio sito Web WordPress, è importante configurare correttamente il sito Web per i diversi tipi di contenuti offerti.

### Argomenti

- [Origini e comportamenti](#)
- [Creazione di una distribuzione CloudFront](#)

## Origini e comportamenti

Un'[origine](#) è una posizione in cui CloudFront invia richieste di contenuti che distribuisce tramite posizioni edge. A seconda dell'implementazione, è possibile avere una o due origini: una per i contenuti dinamici (l'istanza Lightsail nell'[opzione di distribuzione a server singolo](#) o l'Application Load Balancer nell'[opzione di distribuzione elastica](#)) utilizzando un'origine personalizzata, oppure una seconda origine a cui indirizzare CloudFront per i contenuti statici. Nell'[architettura di riferimento](#) precedente, si tratta di un bucket S3. Quando si utilizza Amazon S3 come origine per una distribuzione, è necessario utilizzare una [policy per bucket](#) che consenta di rendere il contenuto accessibile pubblicamente.

I [comportamenti](#) consentono di impostare regole per gestire il modo in cui CloudFront memorizza i contenuti nella cache e, a loro volta, determinano l'efficacia della cache. I comportamenti permettono di controllare il protocollo e i metodi HTTP con cui è accessibile il sito Web. Consentono inoltre di stabilire se migrare intestazioni HTTP, cookie o stringhe di query al backend e quali di essi migrare. I comportamenti si applicano a modelli di percorsi URL specifici.

## Creazione di una distribuzione CloudFront

Creare una distribuzione Web CloudFront seguendo la distribuzione, l'origine e il comportamento predefiniti creati automaticamente per i contenuti dinamici. Creare quattro comportamenti aggiuntivi per personalizzare ulteriormente il modo in cui vengono trattate le richieste statiche e dinamiche. Nella tabella seguente vengono riepilogate le proprietà di configurazione per i cinque comportamenti. È anche possibile saltare questa configurazione manuale e utilizzare il plug-in AWS per WordPress trattato nell'[Appendice B: Installazione e configurazione di plug-in](#): è il modo più semplice per configurare CloudFront per accelerare il sito WordPress.

Tabella 1: Riepilogo delle proprietà di configurazione per i comportamenti di CloudFront

Proprietà	Statico	Dinamico (amministratore)	Dinamico (frontale)
Percorsi (comportamenti)	wp-content/* wp-includes/*	wp-admin/* wp-login.php	Predefinito (*)
Protocolli	HTTP e HTTPS	Reindirizzamento a HTTPS	HTTP e HTTPS
Metodi HTTP	GET, HEAD	ALL	ALL
Intestazioni HTTP	NONE	ALL	Host CloudFront-Forwarded-Proto CloudFront-Is-Mobile-Viewer CloudFront-Is-Tablet-Viewer CloudFront-Is-Desktop-Viewer
Cookie	NONE	ALL	comment_* wordpress_* wp-settings-*
Stringhe di query	YES (invalidamento)	Sì	Sì

Per il comportamento predefinito, AWS consiglia la seguente configurazione:

- Consentire la policy relativa al protocollo di origine per Match Viewer (Corrispondenza visualizzatore), in modo che se i visualizzatori si connettono a CloudFront tramite HTTPS, CloudFront si connette anche all'origine dell'utente tramite HTTPS, garantendo una crittografia end-to-end. Nota: è richiesta l'installazione di un certificato SSL attendibile sul sistema di bilanciamento del carico. Maggiori dettagli sono disponibili nel documento [Richiesta di una connessione HTTPS per la comunicazione tra CloudFront e il server di origine personalizzato](#).
- Consentire tutti i metodi HTTP, dal momento che le parti dinamiche del sito Web hanno bisogno sia di richieste GET che POST (ad esempio, per supportare POST per i moduli per l'invio di commenti).
- Inoltrare solo i cookie che modificano l'output di WordPress, ad esempio `>wordpress_*`, `wp-settings-*` e `comment_*`. È necessario estendere tale elenco se sono stati installati plug-in che dipendono da altri cookie non presenti in elenco.
- Inoltrare solo le intestazioni HTTP che influiscono sull'output di WordPress, ad esempio `Host`, `CloudFront-Forwarded-Proto`, `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` e `CloudFront-is-Tablet-Viewer`:
  - `Host` consente a più siti Web WordPress di essere ospitati sulla stessa origine.
  - `CloudFront-Forwarded-Proto` consente di memorizzare nella cache diverse versioni delle pagine a seconda del fatto che per l'accesso si utilizzi HTTP o HTTPS.
  - `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer` e `CloudFront-is-Tablet-Viewer` consentono di personalizzare l'output dei temi in base al tipo di dispositivo dell'utente finale.
- Inoltrare tutte le stringhe di query alla cache in base ai loro valori, perché WordPress dipende da questi; possono anche essere utilizzati per invalidare gli oggetti memorizzati nella cache.

Se si desidera utilizzare il sito Web con un nome di dominio personalizzato (ovvero, diverso da `*.cloudfront.net`), inserire gli URI appropriati in Alternate Domain Names (Nomi di dominio alternativi) in Distribution Settings (Impostazioni di distribuzione). In questo caso, è necessario anche un certificato SSL per il nome di dominio personalizzato. È possibile [richiedere](#) certificati SSL tramite AWS Certificate Manager e configurarli in base a una distribuzione CloudFront.

Ora, creare altri due comportamenti della cache per i contenuti dinamici: uno per la pagina di accesso (modello di percorso: `wp-login.php`) e uno per la dashboard di amministrazione (modello percorso: `wp-admin/*`). Questi due comportamenti hanno esattamente le stesse impostazioni, ovvero:

- Applicazione della policy solo HTTPS per il protocollo del visualizzatore.

- Tutti i metodi HTTP consentiti.
- Cache basata su tutte le intestazioni HTTP.
- Inoltro di tutti i cookie
- Inoltro e caching eseguiti in base a tutte le stringhe di query.

Il motivo alla base di questa configurazione è che questa sezione del sito Web include un livello di personalizzazione elevato e, in genere, ha solo pochi utenti, per cui l'efficienza del caching non è una priorità. L'obiettivo è quello di mantenere la configurazione semplice per garantire la massima compatibilità con qualsiasi plug-in installato migrando tutti i cookie e le intestazioni nell'origine.

Il [plug-in AWS per WordPress](#) discusso nell'Appendice B crea automaticamente una distribuzione CloudFront che soddisfa i requisiti della configurazione precedente.

Per impostazione predefinita, WordPress esegue la memorizzazione in locale sul server Web, più in particolare nello storage a blocchi (Amazon EBS) per la [distribuzione a server singolo](#) e nello storage di file (Amazon EFS) per la [distribuzione elastica](#). Oltre a ridurre i costi di storage e trasferimento dei dati, la migrazione di asset statici su Amazon S3 offre un migliore livello di scalabilità, disponibilità dei dati, sicurezza e prestazioni. Esistono diversi plug-in che semplificano la migrazione di contenuti statici su Amazon S3; uno di questi è [W3 Total Cache](#), anch'esso trattato nell'[Appendice B: Installazione e configurazione di plug-in](#).

## Appendice B: Installazione e configurazione di plug-in

### Argomenti

- [Plug-in AWS per WordPress](#)
- [Configurazione statica dei contenuti](#)

## Plug-in AWS per WordPress

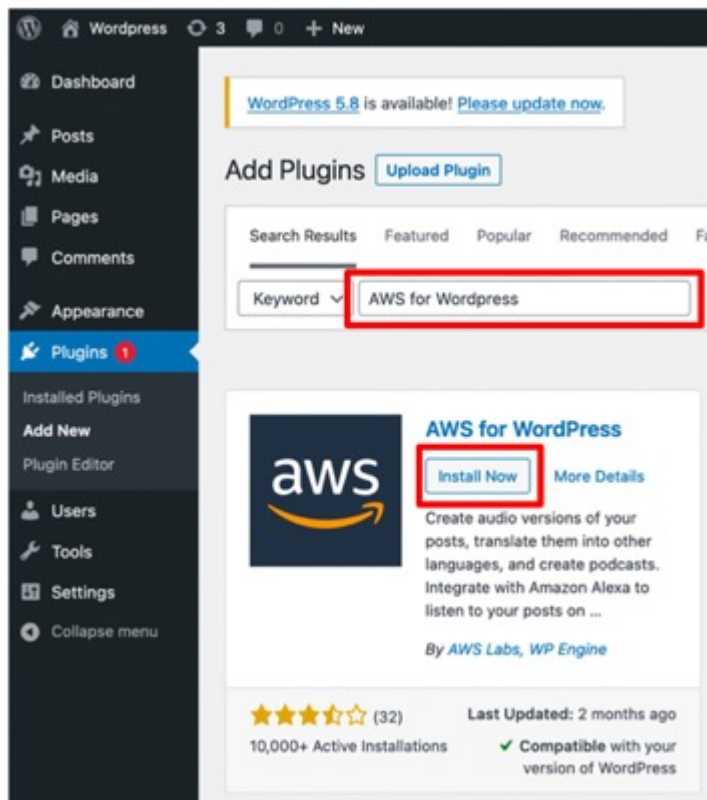
Il plug-in AWS per WordPress è l'unico plug-in WordPress scritto e gestito attivamente da AWS. Consente ai clienti di configurare in modo semplice [Amazon CloudFront](#) e [AWS Certificate Manager \(ACM\)](#) nei siti Web WordPress per migliorare le prestazioni e la sicurezza. Il plug-in utilizza i servizi di [Amazon Machine Learning](#) per tradurre i contenuti in una o più lingue, produce versioni audio di ciascuna traduzione e legge i siti Web WordPress tramite i dispositivi Amazon Alexa.

Il plug-in è già installato in [WordPress High Availability by Bitnami su AWS Quick Start](#).

## Installazione e configurazione dei plug-in

Per installare il plug-in:

1. Per utilizzare il plug-in AWS per WordPress, è necessario creare un utente IAM per il plug-in. Un utente IAM è una persona o un'applicazione con un account AWS che dispone dell'autorizzazione per effettuare chiamate API ai servizi AWS.
2. È necessario un ruolo [AWS Identity and Access Management](#) (IAM) o un utente IAM per controllare l'autenticazione e l'autorizzazione di un account AWS. Per impedire agli utenti non autorizzati di ottenere queste autorizzazioni, proteggere le credenziali dell'utente IAM. Considerare la chiave di accesso segreta come una password; memorizzarla in un luogo sicuro e non condividerla con nessuno. Come una password, [ruotare periodicamente la chiave di accesso](#). Se la chiave di accesso segreta viene accidentalmente divulgata, [eliminarla immediatamente](#). Quindi è possibile creare una nuova chiave di accesso da utilizzare con il plug-in AWS per WordPress.
3. Nel menu Plugins (Plug-in) del pannello di amministrazione di WordPress, cercare AWS for WordPress (AWS per WordPress) e scegliere Install Now (Installa ora).



4. Se l'installazione del plug-in non funziona, potrebbe esserci un problema di autorizzazione dell'utente. Connettersi al server Web WordPress e completare la seguente procedura per risolvere il problema.
  - a. Aprire il file `wp-config.php` nella directory di installazione di WordPress e scrivere il seguente codice alla fine del file `wp-config.php`:

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

- b. Avviare il seguente comando per concedere l'autorizzazione di scrittura:

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

Attenzione: mantenere come autorizzazione di scrittura 777 è rischioso, in quanto chiunque può modificare o eliminare questa cartella. Modificare l'autorizzazione di scrittura in 755 o un valore inferiore una volta terminato di utilizzare il plug-in.

- c. Se viene utilizzata l'architettura di riferimento, la directory di installazione di WordPress è “`/var/www/wordpress/<site directory>`”.



Una descrizione dettagliata di tutte le impostazioni di AWS per WordPress esula dall'ambito di questo documento. Per la configurazione e le opzioni, consultare [Nozioni di base sul plug-in AWS per WordPress](#).

## Amazon CloudFront e AWS Certificate Manager

Per configurare CloudFront e AWS Certificate Manager:

1. Nel menu del plug-in, scegliere CloudFront e inserire i seguenti parametri:
  - Nome di dominio di origine: dominio DNS del server di origine HTTP in cui CloudFront ottiene i contenuti del sito Web dell'utente (ad esempio, `example.com`).
  - Nome di dominio alternativo (CNAME): nome di dominio che i visitatori utilizzano per ottenere un'esperienza del sito Web accelerata. AWS consiglia di utilizzare "www" davanti al dominio (ad esempio, `www.example.com`).
2. Scegliere Initiate Setup (Avvia configurazione) per avviare la configurazione.

Il plug-in richiede automaticamente un certificato SSL per il CNAME tramite ACM; una volta convalidato il token ACM [aggiornando i record DNS](#) con le voci CNAME, il plug-in creerà una distribuzione CloudFront che soddisfa le best practice definite nell'[Appendice A](#).

### Note

Il plug-in AWS per WordPress richiede il protocollo HTTPS per la comunicazione tra CloudFront e l'origine personalizzata dell'utente. Assicurarsi che l'origine includa un certificato SSL valido per il nome di dominio di origine. Per maggiori informazioni, consultare [Richiesta di una connessione HTTPS per la comunicazione tra CloudFront e il server di origine personalizzato](#).

## Traduzione e conversione in vocale dei contenuti

Il plug-in AWS per WordPress è in grado di tradurre automaticamente il testo in più lingue e di convertire il contenuto scritto in formati audio multilingue. Queste funzionalità sono basate sui servizi Amazon Machine Learning.

[Amazon Polly](#) è un servizio che converte il testo in voce naturale. È possibile selezionare la voce più adatta tra le decine disponibili e creare applicazioni interattive dotate di funzionalità vocali che

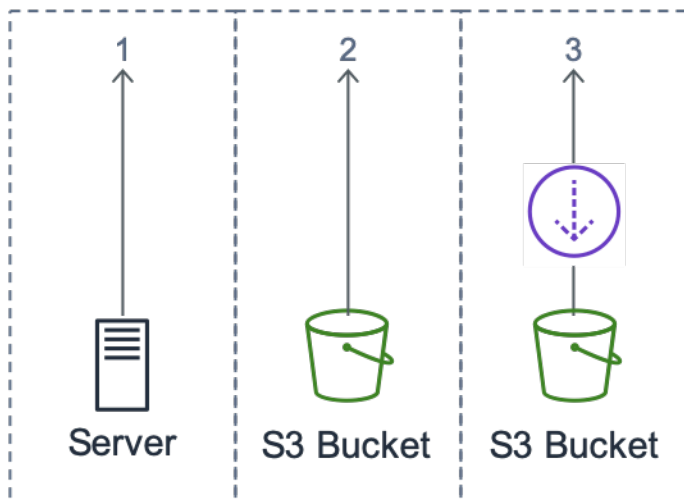
possono essere utilizzate in diversi Paesi. Utilizzare il plug-in per creare file audio in una qualsiasi delle voci e delle lingue supportate da Amazon Polly. I visitatori possono trasmettere l'audio a loro piacimento utilizzando lettori audio inline e applicazioni per dispositivi mobili.

Per impostazione predefinita, il plug-in archivia i nuovi file audio sul server Web dell'utente. È possibile scegliere di archiviare i file su Amazon S3 o su Amazon CloudFront. L'esperienza di ascolto degli utenti non dipende da dove vengono archiviati i file audio; cambia solo la zona di trasmissione:

- Se archiviati sul server WordPress, i file audio vengono trasmessi direttamente dal server.
- Se archiviati in un bucket S3, i file vengono trasmessi dal bucket.
- Se si utilizza CloudFront, i file vengono archiviati su Amazon S3 e trasmessi con CloudFront.



Listening...

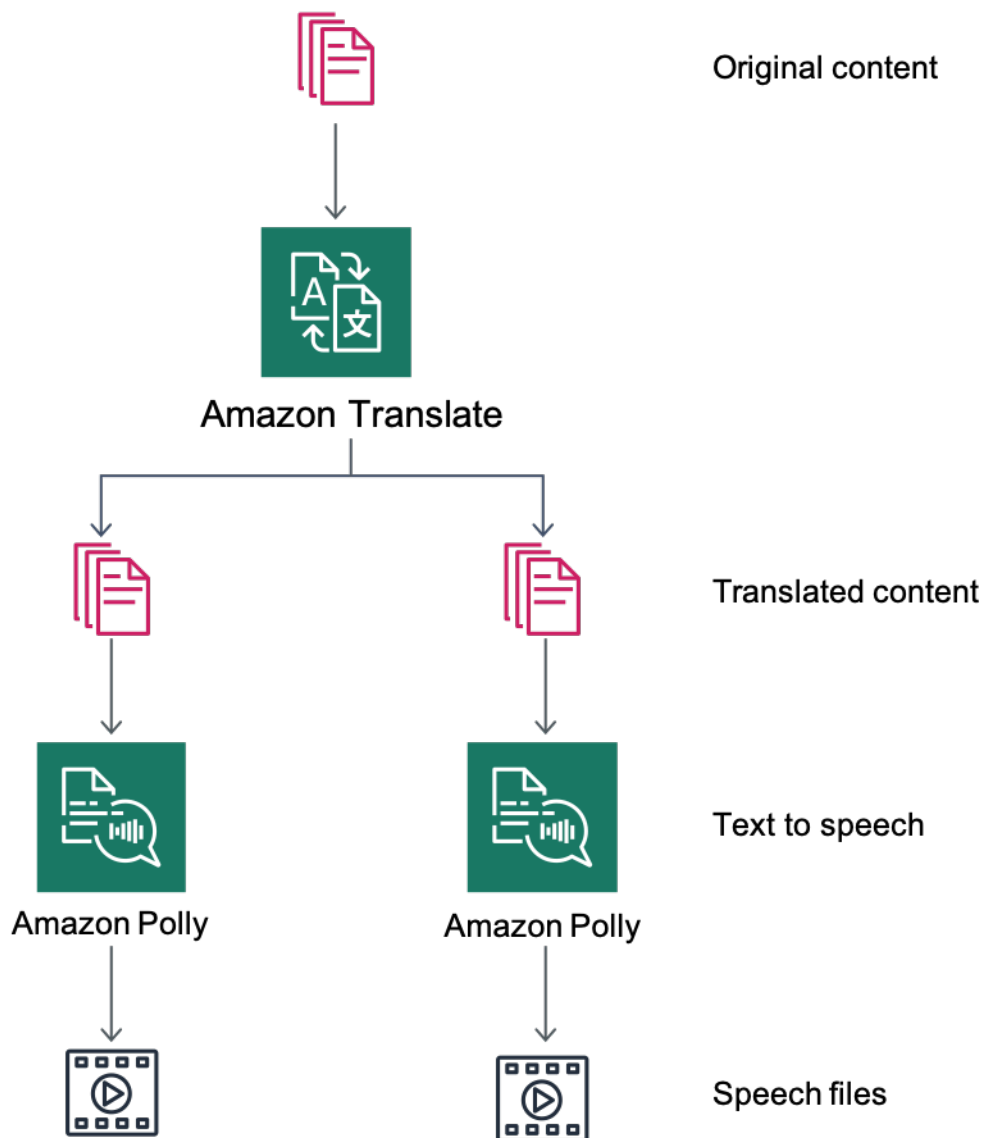


Posizione della trasmissione

[Amazon Translate](#) è un servizio di traduzione automatica neurale che genera traduzioni rapide, di alta qualità e a costi contenuti. Fornire contenuti multilingue rappresenta un'enorme opportunità per i proprietari dei siti. Anche se l'inglese resta la lingua dominante nel Web, gli utenti di madrelingua inglese sono appena il 26% del pubblico online totale.

Offrendo le versioni scritte e audio dei contenuti WordPress in più lingue, è possibile soddisfare le esigenze di un pubblico internazionale più ampio. È possibile configurare il plug-in in modo da poter effettuare le operazioni seguenti:

- Tradurre automaticamente i contenuti in diverse lingue e creare registrazioni audio di ogni traduzione di nuovi contenuti al momento della pubblicazione, oppure tradurre e creare registrazioni di singoli post
- Tradurre contenuti in diverse lingue e creare registrazioni audio di ogni traduzione dei contenuti archiviati
- Utilizzare i feed RSS di Amazon Pollycast per realizzare podcast del contenuto audio



Panoramica sulla traduzione dei contenuti e sulla sintesi vocale

## Riproduzione di podcast con Amazon Pollycast

Con i feed di Amazon Pollycast, i visitatori possono ascoltare i contenuti audio utilizzando applicazioni di podcast standard. I feed Pollycast compatibili con RSS 2.0 forniscono i dati XML necessari per l'aggregazione dei podcast per applicazioni podcast per dispositivi mobili popolari, quali ad esempio iTunes, e directory di podcast.

Quando si installa il plug-in AWS per WordPress, l'opzione per abilitare la generazione di feed XML si trova nella scheda Configurazione podcast. Lì sarà possibile trovare anche l'opzione per configurare più proprietà opzionali. Una volta abilitata la funzionalità, si riceverà un link al feed.

## Lettura dei contenuti tramite i dispositivi Amazon Alexa

È possibile estendere la portata dei siti Web e dei blog di WordPress tramite i dispositivi Alexa. Ciò offre ai creatori e gli autori di siti Web nuove possibilità per raggiungere un pubblico ancora più ampio. Inoltre, consente agli utenti di ascoltare i propri blog preferiti semplicemente chiedendo ad Alexa di leggerli.

Per consentire ad Alexa di accedere al sito Web WordPress, occorre abilitare:

- Plug-in AWS per WordPress.
- Le funzionalità di sintesi vocale e di Amazon Pollycast, che generano un feed RSS sul sito WordPress che viene utilizzato da Amazon Alexa.
- Amazon S3 come spazio di archiviazione predefinito per i file in sintesi vocale; è importante che il sito Web utilizzi una connessione HTTPS sicura per consentire ad Alexa di accedere al feed.

Il seguente diagramma presenta il flusso di interazioni e componenti necessari per consentire ad Alexa di accedere al sito Web.



## Flusso di interazioni necessarie per consentire ad Alexa di accedere ai siti Web WordPress

1. L'utente richiama una nuova competenza Alexa, ad esempio dicendo: Alexa, richiedi a Demo Blog l'ultimo aggiornamento. La skill viene creata utilizzando uno dei progetti di Alexa Skill. Ciò consente ai dispositivi Alexa di accedere alla skill dell'utente anche se quest'ultimo non ha conoscenze tecniche approfondite.
2. La skill Alexa analizza la chiamata e il feed RSS generati dal plug-in AWS per WordPress, quindi restituisce il link alla versione audio dell'ultimo articolo.
3. Sulla base del link fornito dal feed, Alexa legge l'articolo riproducendo il file audio salvato su Amazon S3.

Nella [pagina dei plug-in](#) sul marketplace di WordPress è disponibile una guida dettagliata sull'installazione e la configurazione del plug-in e sulle sue funzionalità.

## Configurazione statica dei contenuti

Per impostazione predefinita, WordPress esegue la memorizzazione in locale sul server Web, più in particolare nello storage a blocchi (Amazon EBS) per la [distribuzione a server singolo](#) e nello storage di file (Amazon EFS) per la [distribuzione elastica](#). Oltre a ridurre i costi di storage e trasferimento dei dati, la migrazione di asset statici su Amazon S3 offre un migliore livello di scalabilità, disponibilità dei dati, sicurezza e prestazioni.

In questo esempio, il plug-in W3 Total Cache (W3TC) viene utilizzato per archiviare risorse statiche su Amazon S3. Sono tuttavia disponibili altri plug-in con funzionalità simili. Se si desidera utilizzare un plug-in diverso, adattare la procedura qui descritta di conseguenza. I passaggi illustrati si applicano solo a caratteristiche o impostazioni relative a questo esempio. Una descrizione dettagliata di tutte le impostazioni esula dall'ambito di questo documento. Consultare la [pagina del plug-in W3 Total Cache](#) su [wordpress.org](http://wordpress.org) per maggiori informazioni.

## Creazione di utenti IAM

Per archiviare risorse statiche su Amazon S3 è necessario creare un utente AWS Identity and Access Management (IAM) per il plug-in WordPress. Per la procedura pertinente, consultare [Creare un utente IAM nell'account AWS](#).

Nota: i ruoli IAM consentono una gestione migliore dell'accesso alle risorse AWS, ma al momento in cui è stato redatto questo documento, il plug-in W3 Total Cache non supporta [ruoli IAM](#).

Si consiglia di prendere nota delle credenziali di sicurezza dell'utente e di archivarle in modo sicuro, in quanto saranno necessarie in un secondo momento.

## Creazione di un bucket Amazon S3

1. Innanzitutto, occorre creare un bucket Amazon S3 nella regione AWS desiderata. Per la procedura, consultare [Creazione di un bucket](#). Abilitare l'hosting di siti Web statici per il bucket seguendo il [tutorial Configurazione di un sito Web statico su Amazon S3](#).
2. Creare una policy IAM per fornire all'utente IAM creato in precedenza l'accesso al bucket S3 specificato e associare la policy all'utente IAM. La procedura per creare la seguente policy è disponibile in [Gestione di policy 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. Installare e attivare il plug-in W3TC dal pannello di amministrazione di WordPress.
4. Andare alla sezione General Settings (Impostazioni generali) della configurazione del plug-in e assicurarsi che sia la cache del browser che la rete CDN siano abilitate.
5. Dall'elenco a discesa relativo alla configurazione della rete CDN, scegliere Origin Push: Amazon CloudFront (Push origine: Amazon CloudFront), che ha come origine Amazon S3.

6. Andare alla sezione Browser Cache (Cache del browser) della configurazione del plug-in e abilitare le intestazioni expires (scadenza), cache control (controllo cache) ed entity tag (ETag) (tag entità) (ETag).
7. Attivare inoltre l'opzione Prevent caching of objects after settings change (Impedire il caching di oggetti dopo la modifica delle impostazioni) in modo che una nuova stringa di query venga generata e aggiunta agli oggetti ogni volta che vengono modificate impostazioni.
8. Andare alla sezione CDN della configurazione del plug-in e inserire le credenziali di sicurezza dell'utente IAM creato in precedenza, oltre al nome del bucket S3.
9. Se il sito Web è gestito tramite l'URL CloudFront, inserire il nome del dominio di distribuzione nella casella pertinente. In caso contrario, inserire uno o più CNAME per i propri nomi di dominio personalizzati.
10. Infine, esportare la libreria multimediale e caricare i wp-include, i file dei temi e i file personalizzati su Amazon S3 utilizzando il plug-in W3TC. Queste funzioni di caricamento sono disponibili nella sezione General (Generale) della pagina di configurazione della rete CDN.

## Creazione di origini statiche

Ora che i file statici sono archiviati su Amazon S3, occorre tornare alla configurazione di CloudFront nella console CloudFront e configurare Amazon S3 come origine per i contenuti statici. Per farlo, aggiungere una seconda origine che punta al bucket S3 creato a tale scopo. Quindi, creare altri due comportamenti della cache, uno per ciascuna delle due cartelle (wp-content e wp-includes) che dovrebbero utilizzare l'origine S3 anziché l'origine predefinita per i contenuti dinamici. Configurare entrambi nello stesso modo:

- Solo richieste HTTP GET.
- Amazon S3 non modifica il proprio output in base a cookie o intestazioni HTTP, per cui non inoltrarli all'origine tramite CloudFront consente di migliorare l'efficienza del caching.
- Sebbene questi comportamenti si applichino solo a contenuti statici (che non accettano parametri), occorrerà inoltrare le stringhe di query all'origine. In questo modo, sarà possibile utilizzare stringhe di query come identificatori di versione per invalidare istantaneamente, ad esempio, file CSS meno recenti quando si distribuiscono nuove versioni. Per ulteriori informazioni, consultare la [Guida per sviluppatori di CloudFront](#).

**Note**

Dopo aver aggiunto i comportamenti di origine statica alla distribuzione CloudFront, controllare l'ordine per assicurarsi che i comportamenti `wp-admin/*` e `wp-login.php` abbiano una priorità maggiore rispetto ai comportamenti per i contenuti statici. In caso contrario, potrebbe verificarsi un comportamento anomalo quando si accede al pannello di amministrazione.



## Appendice C: Backup e ripristino

Il ripristino dopo un guasto in AWS è più rapido e semplice rispetto agli ambienti di hosting tradizionali. Ad esempio, è possibile avviare un'istanza sostitutiva in pochi minuti dopo un guasto hardware oppure utilizzare il failover automatico in molti dei nostri servizi gestiti per annullare l'impatto di un riavvio dovuto a manutenzione ordinaria.

Tuttavia, per il ripristino corretto dei dati è comunque necessario eseguirne il backup. Per ripristinare la disponibilità di un sito Web WordPress, è necessario avere a disposizione i seguenti componenti:

- Installazione e configurazione del sistema operativo (OS) e dei servizi (Apache, MySQL e così via)
- Codice e configurazione dell'applicazione WordPress
- Temi e plug-in di WordPress
- Caricamenti (ad esempio, file multimediali per i post)
- Contenuti del database (post, commenti e così via)

AWS offre vari metodi per il backup e il ripristino dei dati e delle risorse delle applicazioni Web.

In una sezione precedente di questo white paper si è discusso dell'uso delle snapshot di Lightsail per proteggere tutti i dati archiviati nello storage locale dell'istanza. Se il sito Web WordPress viene eseguito solo dall'istanza Lightsail, le comuni snapshot di Lightsail dovrebbero essere sufficienti per ripristinarlo nella sua interezza. Tuttavia, l'esecuzione del ripristino da una snapshot comporterà la perdita delle modifiche applicate al sito Web dall'ultima snapshot.

In una distribuzione multi-server, è necessario eseguire il backup di ciascuno dei componenti discussi in precedenza utilizzando meccanismi diversi. Ogni componente può avere requisiti diversi in termini di frequenza di backup; ad esempio, l'installazione e la configurazione del sistema operativo e di WordPress variano con una frequenza decisamente minore rispetto ai contenuti generati dall'utente, pertanto possono essere sottoposti a backup meno frequentemente senza comportare la perdita di dati in caso di ripristino.

Per eseguire il backup dell'installazione e della configurazione del sistema operativo e dei servizi, nonché del codice e della configurazione dell'applicazione WordPress, è possibile creare un'AMI di un'istanza EC2 configurata correttamente. Le AMI hanno una duplice funzione: possono essere utilizzate come backup dello stato dell'istanza e fungere da modello durante l'avvio nuove istanze.

Per eseguire il backup del codice e della configurazione dell'applicazione WordPress, è necessario utilizzare sia le AMI che i backup di Aurora.

Per eseguire il backup dei temi e dei plug-in di WordPress installati sul proprio sito Web, è necessario eseguire il backup del bucket Amazon S3 o del file system Amazon EFS in cui sono archiviati.

- Per i temi e i plug-in archiviati in un bucket S3 è possibile abilitare la [replica tra regioni](#), in modo che tutti gli oggetti caricati nel bucket primario vengano automaticamente replicati nel bucket di backup in un'altra regione AWS. La replica tra regioni richiede che il [controllo delle versioni](#) sia abilitato sui bucket di origine e destinazione, il che fornisce un ulteriore livello di protezione e consente di ripristinare una versione precedente di un determinato oggetto nel bucket.
- Per i temi e i plug-in archiviati in un file system EFS, è possibile creare un AWS Data Pipeline per copiare i dati dal file system EFS di produzione a un altro file system EFS, come descritto nella sezione [Eseguire il backup dei file system Amazon EFS](#) della documentazione. È inoltre possibile eseguire il backup di un file system EFS utilizzando l'applicazione di backup desiderata.
- Per eseguire il backup dei caricamenti degli utenti, è necessario seguire la procedura per il backup dei temi e dei plug-in di WordPress descritta in precedenza.
- Per eseguire il backup del contenuto del database è necessario effettuare il [backup tramite Aurora](#). Aurora esegue automaticamente il backup del volume del cluster e conserva i dati di ripristino per l'intero tempo di conservazione dei backup. I backup di Aurora sono continui e incrementali, pertanto è possibile eseguire rapidamente il ripristino a un momento qualsiasi del tempo di conservazione dei backup. Durante la scrittura dei dati di backup, non si verifica alcun impatto sulle prestazioni o interruzione del funzionamento del servizio del database. È possibile impostare il tempo di conservazione del backup da 1 a 35 giorni. È inoltre possibile creare [snapshot manuali del database](#), che vengono mantenute finché non vengono eliminate. Le snapshot manuali del database sono utili per i backup e l'archiviazione a lungo termine.

## Appendice D: Implementazione di nuovi plug-in e temi

Pochi siti Web rimangono statici. Nella maggior parte dei casi, verranno aggiunti periodicamente temi e plug-in di WordPress disponibili pubblicamente o verrà eseguito l'aggiornamento a una versione più recente di WordPress. In altri casi, verranno sviluppati da zero temi e plug-in personalizzati.

Ogni volta che si apporta una modifica strutturale alla propria installazione di WordPress si corre il rischio di introdurre problemi imprevisti. Effettuare almeno un backup del codice, della configurazione e del database dell'applicazione prima di apportare qualsiasi modifica significativa (come installare un nuovo plug-in). Per i siti Web di valore aziendale o di altro tipo, testare prima tali modifiche in un ambiente di gestione separato. AWS consente di replicare la configurazione dell'ambiente di produzione e di eseguire l'intero processo di implementazione in modo semplice e sicuro. Dopo aver terminato i test, è possibile semplicemente eliminare l'ambiente di test e smettere di pagare per tali risorse. In una sezione successiva di questo white paper verranno prese in esame alcune considerazioni specifiche di WordPress.

Alcuni plug-in scrivono informazioni di configurazione nella tabella del database `wp_options` (o introducono modifiche allo schema del database), mentre altri creano file di configurazione nella directory di installazione di WordPress. Dal momento che il database e lo storage sono stati trasferiti su piattaforme condivise, queste modifiche diventano immediatamente disponibili per tutte le istanze in esecuzione senza ulteriori interventi da parte dell'utente.

Se vengono implementati nuovi temi in WordPress, potrebbero essere necessarie ulteriori operazioni. Se si utilizza Amazon EFS solo per archiviare tutti i file di installazione di WordPress, i nuovi temi diventeranno immediatamente disponibili per tutte le istanze in esecuzione. Se invece si scaricano contenuti statici su Amazon S3, sarà necessario elaborarne una copia nella posizione del bucket corretta. Plug-in come W3 Total Cache consentono di avviare manualmente tale operazione. In alternativa, è possibile automatizzare questo passaggio all'interno di un processo di creazione.

Poiché le risorse dei temi possono essere memorizzate nella cache su CloudFront e sul browser, è necessario invalidare le versioni precedenti quando si implementano modifiche. A tale scopo, è possibile includere una sorta di identificatore di versione nell'oggetto, che può essere una stringa di query con un indicatore di data e ora o una stringa casuale. Se si utilizza il plug-in W3 Total Cache, è possibile aggiornare una stringa di query che viene aggiunta agli URL dei file multimediali.

# Avvisi

I clienti sono responsabili della propria valutazione autonoma delle informazioni contenute in questo documento. Questo documento: (a) è solo a scopo informativo, (b) mostra le offerte e le pratiche attuali dei prodotti AWS soggette a modifiche senza preavviso, e (c) non crea alcun impegno o garanzia da parte di AWS e dei suoi affiliati, fornitori o licenziatari. I prodotti o servizi AWS sono forniti "così come sono" senza garanzie, dichiarazioni o condizioni di alcun tipo, sia esplicite che implicite. Le responsabilità e gli obblighi di AWS verso i propri clienti sono disciplinati dagli accordi AWS e il presente documento non fa parte né modifica alcun accordo tra AWS e i suoi clienti.

© 2021, Amazon Web Services, Inc., o sue affiliate. Tutti i diritti riservati.