Guida per l'utente

AWS Amplify Ospitare

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Amplify Ospitare: Guida per l'utente

Copyright © 2025 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 delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è AWS Amplify l'hosting? ... 1
Framework supportati ... 1
Funzionalità di Amplify Hosting ... 2
Nozioni di base su Amplify Hosting ... 2
Creazione di un backend ... 3
Prezzi di Amplify Hosting ... 3
Tutorial introduttivi ... 4
Distribuisci un'app Next.js ... 4
Passaggio 1: Connect un repository ... 4
Passaggio 2: Conferma le impostazioni di build ... 5
Fase 3: Distribuire l'applicazione ... 6
Fase 4: (Facoltativo) pulire le risorse ... 7
Aggiungi funzionalità alla tua app ... 7
Distribuisci un'app Nuxt.js ... 8
Distribuisci un'app Astro.js ... 8
Implementa un'app SvelteKit ... 11
Implementazione di applicazioni SSR ... 14
Next.js ... 15
supporto delle funzionalità Next.js ... 16
Distribuzione di un'applicazione SSR Next.js su Amplify ... 17
Migrazione di un'app SSR Next.js 11 al calcolo Amplify Hosting ... 21
Aggiungere la funzionalità SSR a un'app Next.js statica ... 23
Rendere le variabili di ambiente accessibili ai runtime lato server ... 25
Distribuzione di un'app Next.js in un monorepo ... 28
Nuxt.js ... 28
Astro.js ... 28
SvelteKit ... 29
Implementazione di un'app SSR su Amplify ... 29
Funzionalità supportate da SSR ... 31
Supporto della versione Node.js per le app Next.js ... 31
Ottimizzazione delle immagini per le app SSR ... 31
Amazon CloudWatch Logs per app SSR ... 32
Supporto SSR Amplify Next.js 11 ... 32
Risoluzione dei problemi relativi alle implementazioni SSR ... 40

Avanzato: adattatori open source ... 41
Specifiche di distribuzione ... 41
Implementazione di un server Express ... 66
Ottimizzazione delle immagini per gli autori del framework ... 72
Utilizzo di adattatori open source per qualsiasi framework SSR ... 80
Distribuzione di un sito Web statico da S3 ... 82
Distribuzione dalla console Amplify ... 83
Creazione di una policy bucket da implementare utilizzando il SDKs ... 84
Aggiornamento di un sito Web statico distribuito da un S3 bucket ... 86
Aggiornamento di un S3 distribuzione per utilizzare un bucket e un prefisso anziché un file.zip ... 86
Implementazione senza Git ... 88
Distribuzioni manuali trascina e rilascia ... 88
Distribuzione manuale di Amazon S3 o URL ... 89
Risoluzione dei problemi di accesso ai bucket Amazon S3 per le distribuzioni manuali ... 90
Crea impostazioni e configurazione ... 91
Configurazione delle impostazioni di compilazione ... 92
Riferimento alle specifiche della build ... 92
Modifica delle specifiche di costruzione ... 95
Impostazioni di Build di Monorepo ... 102
Personalizzazione dell'immagine di costruzione ... 108
Configurazione di un'immagine di build personalizzata per un'app ... 110
Utilizzo di versioni specifiche di pacchetti e dipendenze nell'immagine di build ... 110
Configurazione dell'istanza di compilazione ... 111
Comprendere i tipi di istanze di build ... 112
Configurazione del tipo di istanza di compilazione nella console Amplify ... 113
Configurazione della memoria heap di un'applicazione per utilizzare tipi di istanze di grandi dimensioni ... 115
Numero di webhook in arrivo ... 117
Creazione di notifiche ... 118
Impostazione della notifica e-mail ... 118
Connessione di un dominio personalizzato ... 119
Comprensione della terminology and concepts DNS ... 120
Terminologia DNS ... 120
Verifica DNS ... 121
Processo di attivazione del dominio personalizzato ... 121

Utilizzo dei certificati SSL/TLS ... 122
Aggiunta di un dominio personalizzato gestito da Amazon Route 53 ... 123
Aggiunta di un dominio personalizzato gestito da un provider DNS di terze parti ... 125
Aggiornamento dei record DNS per un dominio gestito da GoDaddy ... 130
Aggiornamento del certificato SSL/TLS per un dominio ... 134
Gestione dei sottodomini ... 135
Solo per aggiungere un sottodominio ... 135
Per aggiungere un sottodominio multilivello ... 135
Aggiunta o modifica di un sottodominio ... 136
Configurazione di sottodomini wildcard ... 136
Per aggiungere o eliminare un sottodominio wildcard ... 137
Configurazione di sottodomini automatici per un dominio personalizzato Amazon Route 53 ... 137
Anteprime Web con sottodomini ... 138
Risoluzione dei problemi dei domini personalizzati ... 138
Supporto firewall per siti ospitati ... 139
Abilita l' AWS WAF utilizzo della console ... 140
Rimuovi da un'app AWS WAF ... 144
Abilita l' AWS WAF utilizzo del CDK ... 145
In che modo Amplify si integra con AWS WAF ... 146
Politica delle risorse Web ACL di Amplify ... 147
Prezzi del firewall ... 147
Funzionalità di implementazioni nelle filiali ... 149
Flussi di lavoro in team con app complete Amplify Gen 2 ... 150
Flussi di lavoro in team con app complete Amplify Gen 1 ... 150
Flusso di lavoro del ramo feature ... 150
GitFlow flusso di lavoro ... 156
Per sandbox sviluppatore ... 157
Implementazioni di feature branch basate su pattern ... 158
Implementazioni di feature branch basate su pattern per un'app connessa a un dominio personalizzato ... 159
Generazione automatica in fase di compilazione della configurazione Amplify (solo app di prima generazione) ... 159
Build di backend condizionali (solo app di prima generazione) ... 161
Usa i backend Amplify tra le app (solo app di prima generazione) ... 162
Riutilizza i backend quando crei una nuova app ... 162
Riutilizza i backend quando connetti una filiale a un'app esistente ... 163

Modifica un frontend esistente in modo che punti a un backend diverso 164
Creazione di un backend 166
Crea un backend per un'app di seconda generazione 166
Crea un backend per un'app di prima generazione 166
Prerequisiti 166
Fase 1: Implementazione di un frontend 167
Fase 2: Creare un backend 168
Passaggio 3: Connect il backend al frontend 169
Passaggi successivi 171
Caratteristiche avanzate di implementazione 172
Rami della protezione tramite password 172
Anteprima pull 173
Abilita le anteprime web per le richieste pull 174
Accesso all'anteprima Web con sottodomini 176
End-to-end test 176
Aggiungere test Cypress a un'applicazione Amplify esistente 176
Disattivazione dei test per un'applicazione o un ramo Amplify 178
Tasto di distribuzione con un clic 179
Aggiungere il pulsante Deploy to Amplify Hosting a un repository o blog 180
Reindirizza e riscritture 181
Comprendere i reindirizzamenti supportati da Amplify 181
Comprendere l'ordine dei reindirizzamenti 182
Comprendere come Amplify inoltra i parametri di interrogazione 183
Creazione e modifica dei reindirizzamenti 183
Esempi di reindirizzamenti e riscritture 184
Reindirizzamenti e riscritture semplici 185
Reindirizzamenti per app Web a pagina singola (SPA) 188
Riscrittura inversa del proxy 189
Barre finali e pulizia URLs 190
Placeholder 190
Stringhe di query e parametri del percorso 191
Reindirizzamenti basati sulla regione 192
Utilizzo di espressioni con caratteri jolly nei reindirizzamenti e nelle riscritture 193
Variabili di ambiente 195
Riferimento alla variabile di ambiente Amplify 195
Variabili di ambiente del framework frontend 201

Impostazione delle variabili di ambiente ... 202
Crea un nuovo ambiente di backend con parametri di autenticazione per l'accesso tramite social ... 202
Gestione dei segreti ambientali ... 204
Utilizzo AWS Systems Manager per impostare segreti ambientali per un'applicazione Amplify Gen 1 ... 204
Accesso ai segreti ambientali per un'applicazione di prima generazione ... 205
Riferimento ai segreti dell'ambiente Amplify ... 205
Intestazioni personalizzate ... 206
Riferimento YAML ... 206
Impostazione di intestazioni personalizzate ... 207
Esempio di intestazioni personalizzate di sicurezza ... 209
Impostazione delle intestazioni personalizzate di Cache-Control ... 209
Migrazione delle intestazioni personalizzate ... 210
Intestazioni personalizzate Monorepo ... 212
Gestione della configurazione della cache ... 213
In che modo Amplify applica la configurazione della cache ... 215
Comprensione delle politiche di cache gestita di Amplify ... 216
Gestione dei cookie della chiave cache ... 219
Inclusione o esclusione dei cookie dalla chiave cache ... 220
Modifica della configurazione dei cookie della chiave cache per un'app ... 221
Utilizzo dell'intestazione Cache-Control per aumentare le prestazioni dell'app ... 222
Disallineamento ... 224
Configurazione della protezione da asimmetria ... 225
Come funziona la protezione dall'inclinazione ... 226
X-Amplify-Dpl esempio di intestazione ... 227
Monitoraggio delle applicazioni ... 229
CloudWatch metriche e allarmi ... 229
CloudWatch Metriche supportate ... 229
Accesso alle metriche CloudWatch ... 233
Creazione di allarmi CloudWatch ... 233
Accesso ai CloudWatch log per le app SSR ... 235
Log di accesso ... 235
Recupero dei log di accesso di un'app ... 236
Analisi dei log di accesso ... 237
Registrazione delle chiamate API Amplify utilizzando AWS CloudTrail ... 237

Amplify le informazioni in CloudTrail ... 238
Informazioni sulle voci dei file di registro di Amplify ... 239
Utilizzo dei ruoli IAM con le applicazioni ... 242
Aggiungere un ruolo di servizio per distribuire le risorse di backend ... 242
Creazione di un ruolo di servizio Amplify nella console IAM ... 243
Modifica della politica di fiducia di un ruolo di servizio per evitare la confusione del vice ... 244
Aggiungere un ruolo SSR Compute ... 244
Creazione di un ruolo SSR Compute nella console IAM ... 246
Aggiungere un ruolo IAM SSR Compute a un'app Amplify ... 248
Gestione della sicurezza dei ruoli IAM SSR Compute ... 249
Aggiungere un ruolo di servizio per accedere ai registri CloudWatch ... 250
Webhook unificati per repository Git ... 251
Nozioni di base su webhook unificati ... 251
Sicurezza ... 253
Identity and Access Management ... 253
Destinatari ... 254
Autenticazione con identità ... 255
Gestione dell'accesso con policy ... 258
Come funziona Amplify con IAM ... 261
Esempi di policy basate su identità ... 268
Policy gestite da AWS ... 271
Risoluzione dei problemi ... 285
Protezione dei dati ... 287
Crittografia a riposo ... 288
Crittografia in transito ... 288
Gestione delle chiavi di crittografia ... 289
Convalida della conformità ... 289
Sicurezza dell'infrastruttura ... 290
Registrazione di log e monitoraggio ... 291
Prevenzione del problema "confused deputy" tra servizi ... 292
Best practice di sicurezza ... 294
Utilizzo dei cookie con il dominio predefinito Amplify ... 294
Quote ... 295
Risoluzione dei problemi ... 298
Problemi generali ... 298
Codice di stato HTTP 429 (troppe richieste) ... 298

La console Amplify non mostra lo stato della build e l'ora dell'ultimo aggiornamento della mia app ... 299
Le anteprime Web non vengono create per le nuove richieste pull ... 300
La mia distribuzione manuale è bloccata con uno stato in sospeso nella console Amplify ... 300
Devo aggiornare la versione Node.js della mia applicazione ... 301
AL2Immagine di compilazione 023 ... 303
Voglio eseguire le funzioni Amplify con il runtime Python ... 303
Voglio eseguire comandi che richiedono i privilegi di superutente o root ... 304
Problemi di build ... 304
I nuovi commit sul mio repository non attivano le build di Amplify ... 304
Il nome del mio repository non è elencato nella console Amplify quando creo una nuova applicazione ... 305
La mia build fallisce con l'`Cannot find module aws-exports`errore (solo app di prima generazione) ... 305
Voglio sovrascrivere un timeout di compilazione ... 306
Domini personalizzati ... 306
Devo verificare che il mio CNAME si risolva ... 306
Il mio dominio ospitato presso una terza parte è bloccato nello stato In attesa di verifica ... 307
Il mio dominio ospitato con Amazon Route 53 è bloccato nello stato di verifica in sospeso ... 308
La mia app con sottodomini a più livelli è bloccata nello stato In attesa di verifica ... 309
Il mio provider DNS non supporta i record A con nomi di dominio completi ... 309
Ricevo un errore CNAMEAlready ExistsException ... 310
Ricevo un errore di verifica aggiuntiva richiesta ... 311
Ricevo un errore 404 sull'URL CloudFront ... 312
Ricevo errori nel certificato SSL o nel protocollo HTTPS quando visito il mio dominio ... 312
Rendering lato server (SSR) ... 313
Ho bisogno di aiuto per usare un adattatore di framework ... 314
I percorsi dell'API Edge causano il fallimento della mia build di Next.js ... 314
La rigenerazione statica incrementale su richiesta non funziona per la mia app ... 314
L'output di build della mia applicazione supera la dimensione massima consentita ... 314
La mia build fallisce con un errore di memoria esaurita ... 39
La dimensione della risposta HTTP della mia applicazione è troppo grande ... 317
Come posso misurare l'ora di avvio della mia app di elaborazione a livello locale? ... 39
Reindirizzamenti e riscritture ... 318
L'accesso è negato per determinati percorsi anche con la regola di reindirizzamento SPA. ... 318
Voglio configurare un proxy inverso su un'API ... 319

Caching 319
Voglio ridurre le dimensioni della cache di un'app 319
Voglio disabilitare la lettura dalla cache per un'app 320
Configurazione delle voci di GitHub accesso 320
Installazione e autorizzazione dell'app GitHub Amplify per una nuova distribuzione 321
Migrazione di un'OAuthapp esistente all'app Amplify GitHub 322
Configurazione dell'app GitHub Amplify per le implementazioni CLI AWS CloudFormation e SDK 323
Configurazione delle anteprime web con l'app Amplify GitHub 324
AWS Amplify Riferimento all'hosting 325
AWS CloudFormation supporto 325
AWS Command Line Interface supporto 325
Supporto per l'etichettatura delle risorse 325
API di hosting Amplify 325
Cronologia dei documenti 326
..... cccxlii

Benvenuto su AWS Amplify Hosting

Amplify Hosting fornisce un flusso di lavoro basato su Git per l'hosting di applicazioni web full-stack serverless con distribuzione continua. Amplify distribuy distribuisce la tua app sulla rete di distribuzione di contenuti (AWS CDN) globale. Questa guida per l'utente fornisce informazioni necessarie per iniziare a utilizzare Amplify Hosting.

Framework supportati

Amplify Hosting supporta molti framework SSR comuni, framework di applicazioni a pagina singola (SPA) e generatori di siti statici, inclusi i seguenti.

Framework SSR

- Next.js
- Nuxt
- Astrocon un adattatore comunitario
- SvelteKitcon un adattatore comunitario
- Qualsiasi framework SSR con un adattatore personalizzato

Framework SPA

- React
- Angular
- Vue.js
- Ionic
- Ember

Generatori di siti statici

- Eleventy
- Gatsby
- Hugo
- Jekyll

- VuePress

Funzionalità di Amplify Hosting

Filiali funzionali

Gestisci gli ambienti di produzione e staging per il frontend e il backend collegando nuove filiali.

Domini personalizzati

Connect l'applicazione a un dominio personalizzato.

Anteprime delle pull request

Visualizza l'anteprima delle modifiche durante le revisioni del codice.

End-to-end test

Migliora la qualità delle tue app con end-to-end i test.

Filiali protette da password

Proteggere con password l'app Web in modo da poter sviluppare nuove funzionalità senza renderle accessibili pubblicamente.

Reindirizza e riscrive

Imposta riscritture e reindirizzamenti per mantenere il posizionamento SEO e indirizzare il traffico in base ai requisiti dell'app client.

Implementazioni atomiche

Le distribuzioni Atomic eliminano le finestre di manutenzione garantendo che l'app Web venga aggiornata solo al termine dell'intera distribuzione. In questo modo si eliminano gli scenari in cui i file non vengono aggiornati correttamente.

Nozioni di base su Amplify Hosting

Per iniziare con Amplify Hosting, consulta il tutorial. Guida introduttiva alla distribuzione di un'app su Amplify Hosting Dopo aver completato il tutorial, saprai come connettere un'app web in un repository Git (GitHub, BitBucket GitLab, o AWS CodeCommit) e distribuirla su Amplify Hosting con distribuzione continua.

Creazione di un backend

AWS Amplify Gen 2 introduce TypeScript un'esperienza di sviluppo basata sul codice e incentrata sul codice per la definizione dei backend. Per sapere come usare Amplify Gen 2 per creare e connettere un backend alla tua app, consulta Build & connect backend nei documenti Amplify.

Per comprendere meglio l'approccio basato sul codice di Gen 2 Amplify, consulta l'Amplify Gen 2 Workshop sul sito web di Workshop Studio.AWS In questo tutorial completo, creerai un'applicazione serverless con React e Next.js e imparerai a usare le librerie Amplify Gen 2 Data and Auth e la libreria Amplify UI per aggiungere funzionalità all'applicazione.

Se stai cercando la documentazione per la creazione di backend per un'app di prima generazione, utilizzando la CLI e Amplify Studio, consulta il backend Build & connect nei documenti Amplify di prima generazione.

Prezzi di Amplify Hosting

AWS Amplify ti fa pagare solo per le risorse utilizzate. Per ulteriori informazioni, consultare AWS Amplify Prezzi.

Guida introduttiva alla distribuzione di un'app su Amplify Hosting

Per aiutarti a capire come funziona Amplify Hosting, i seguenti tutorial ti guidano attraverso la creazione e la distribuzione di applicazioni create utilizzando i comuni framework SSR supportati da Amplify.

Tutorial

- Distribuisci un'app Next.js su Amplify Hosting
- Distribuisci un'app Nuxt.js su Amplify Hosting
- Distribuisci un'app Astro.js su Amplify Hosting
- Distribuisci un' SvelteKit app su Amplify Hosting

Distribuisci un'app Next.js su Amplify Hosting

Questo tutorial illustra la creazione e la distribuzione di un'applicazione Next.js da un repository Git.

Prima di iniziare questo tutorial, completa i seguenti prerequisiti.

Registrati per un Account AWS

Se non sei già un AWS cliente, devi crearne uno Account AWS seguendo le istruzioni online. La registrazione ti consente di accedere ad Amplify e AWS ad altri servizi che puoi utilizzare con la tua applicazione.

Creazione di un'applicazione

Crea un'applicazione Next.js di base da utilizzare per questo tutorial, utilizzando le create-next-appistruzioni nella documentazione di Next.js.

Crea un repository Git

Amplify GitHub supporta, GitLab Bitbucket e. AWS CodeCommit Invia la tua `create-next-app` applicazione al tuo repository Git.

Passaggio 1: Connect un repository Git

In questo passaggio, connetti la tua applicazione Next.js in un repository Git ad Amplify Hosting.

Per connettere un'app in un repository Git

1. Apri la console Amplify.
2. Se stai distribuendo la tua prima app nella regione corrente, per impostazione predefinita inizierai dalla pagina del AWS Amplifyservizio.

   Scegli Distribuisci un'app nella parte superiore della pagina.
3. Nella pagina Inizia a creare con Amplify, scegli il tuo provider di repository Git, quindi scegli Avanti.

   Per gli GitHub archivi, Amplify utilizza la funzione App per autorizzare GitHub l'accesso ad Amplify. Per ulteriori informazioni sull'installazione e l'autorizzazione dell'App, consulta. GitHub Configurazione dell'accesso Amplify ai repository GitHub

   > **Note**
   >
   > Dopo aver autorizzato la console Amplify con Bitbucket GitLab, AWS CodeCommit oppure, Amplify recupera un token di accesso dal provider del repository, ma non lo archivia sui server. AWS Amplify accede al repository utilizzando chiavi di distribuzione installate solo in uno specifico repository.
4. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Seleziona il nome del repository da connettere.
   b. Seleziona il nome del ramo del repository da connettere.
   c. Scegli Next (Successivo).

Passaggio 2: Conferma le impostazioni di build

Amplify rileva automaticamente la sequenza di comandi di compilazione da eseguire per il ramo che stai distribuendo. In questo passaggio rivedi e confermi le impostazioni di build.

Per confermare le impostazioni di build per un'app

1. Nella pagina delle impostazioni dell'app, individua la sezione Impostazioni di creazione.

   Verifica che il comando Frontend build e la directory di output Build siano corretti. Per questa app di esempio Next.js, la directory di output Build è impostata su. `.next`

2. La procedura per aggiungere un ruolo di servizio varia a seconda che si desideri creare un nuovo ruolo o utilizzarne uno esistente.
   - Per creare un nuovo ruolo:
     - Scegli Crea e utilizza un nuovo ruolo di servizio.
   - Per utilizzare un ruolo esistente:
     a. Scegli Usa un ruolo esistente.
     b. Nell'elenco dei ruoli di servizio, seleziona il ruolo da utilizzare.
3. Scegli Next (Successivo).

Fase 3: Distribuire l'applicazione

In questa fase distribuisci la tua app nella rete AWS globale di distribuzione dei contenuti (CDN).

Per salvare e distribuire un'app

1. Nella pagina di revisione, verifica che i dettagli del repository e le impostazioni dell'app siano corretti.
2. Scegliere Save and deploy (Salva e distribuisci). La creazione del front-end richiede in genere da 1 a 2 minuti, ma può variare in base alle dimensioni dell'app.
3. Una volta completata la distribuzione, puoi visualizzare l'app utilizzando il link al dominio `amplifyapp.com` predefinito.

> **Note**
>
> Per aumentare la sicurezza delle tue applicazioni Amplify, il dominio amplifyapp.com è registrato nella Public Suffix List (PSL). Per una maggiore sicurezza, ti consigliamo di utilizzare i cookie con un `__Host-` prefisso se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Amplify. Questa pratica ti aiuterà a difendere il tuo dominio dai tentativi CSRF (cross-site request forgery). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Fase 4: (Facoltativo) pulire le risorse

Se non ti serve più l'app che hai distribuito per il tutorial, puoi eliminarla. In questo modo hai la certezza che non ti vengano addebitati costi per risorse che non stai utilizzando.

Per eliminare un'app

1. Dal menu delle impostazioni dell'app nel riquadro di navigazione, scegli Impostazioni generali.
2. Nella pagina delle impostazioni generali, scegli Elimina app.
3. Nella finestra di conferma, inserisci**`delete`**. Quindi scegli Elimina app.

Aggiungi funzionalità alla tua app

Ora che hai un'app distribuita su Amplify, puoi esplorare alcune delle seguenti funzionalità disponibili per la tua applicazione ospitata.

Variabili di ambiente

Le applicazioni spesso richiedono informazioni di configurazione in fase di esecuzione. Queste configurazioni possono essere dettagli di connessione al database, chiavi API o parametri. Le variabili di ambiente forniscono un modo per esporre queste configurazioni in fase di compilazione. Per ulteriori informazioni, consulta Variabili di ambiente.

Domini personalizzati

In questo tutorial, Amplify ospita la tua app per te sul dominio `amplifyapp.com` predefinito con un URL come. `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com` Quando colleghi la tua app a un dominio personalizzato, gli utenti vedono che la tua app è ospitata su un URL personalizzato, ad esempio. `https://www.example.com` Per ulteriori informazioni, consulta Configurazione di domini personalizzati.

Anteprime delle pull request

Le anteprime delle richieste pull web offrono ai team un modo per visualizzare in anteprima le modifiche apportate alle pull request (PRs) prima di unire il codice a un ramo di produzione o di integrazione. Per ulteriori informazioni, vedete Anteprime Web per le richieste pull.

Gestione di più ambienti

Per scoprire come Amplify funziona con feature branch GitFlow e flussi di lavoro per supportare più implementazioni, consulta Distribuzioni di feature branch e flussi di lavoro di team.

Distribuisci un'app Nuxt.js su Amplify Hosting

Utilizza le seguenti istruzioni per distribuire un'applicazione Nuxt.js su Amplify Hosting. Nuxt ha implementato un adattatore preimpostato utilizzando il server Nitro. Ciò consente di implementare un progetto Nuxt senza alcuna configurazione aggiuntiva.

Per distribuire un'app Nuxt su Amplify Hosting

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella pagina Tutte le app, scegli Crea nuova app.
3. Nella pagina Inizia a creare con Amplify, scegli il tuo provider di repository Git, quindi scegli Avanti.
4. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Seleziona il nome del repository da connettere.
   b. Seleziona il nome del ramo del repository da connettere.
   c. Scegli Next (Successivo).
5. Se desideri che Amplify sia in grado di inviare i log delle app ad CloudWatch Amazon Logs, devi abilitarlo esplicitamente nella console. Apri la sezione Impostazioni avanzate, quindi scegli Abilita i log delle app SSR nella sezione Distribuzione Server-Side Rendering (SSR).
6. Scegli Next (Successivo).
7. Nella pagina di revisione, scegli Salva e distribuisci.

Distribuisci un'app Astro.js su Amplify Hosting

Utilizza le seguenti istruzioni per distribuire un'applicazione Astro.js su Amplify Hosting. È possibile utilizzare un'applicazione esistente o creare un'applicazione iniziale utilizzando uno degli esempi ufficiali forniti da Astro. Per creare un'applicazione iniziale, consulta Utilizzare un tema o un modello di avvio nella documentazione di Astro.

Per implementare un sito Astro con SSR su Amplify Hosting, devi aggiungere un adattatore alla tua applicazione. Non gestiamo un adattatore di proprietà di Amplify per il framework Astro. Questo tutorial utilizza l'`astro-aws-amplify`adattatore creato da un membro della community. Questo adattatore è disponibile su github. com/alexnguyennz/astro-aws-amplify sul sito web. GitHub AWS non mantiene questo adattatore.

Per distribuire un'app Astro su Amplify Hosting

1. Sul tuo computer locale, vai all'applicazione Astro per distribuirla.
2. Per installare l'adattatore, apri una finestra di terminale ed esegui il seguente comando. Questo esempio utilizza il community adapter disponibile su github. com/alexnguyennz/astro-aws-amplify. Puoi sostituirlo *astro-aws-amplify* con il nome dell'adattatore che stai utilizzando.

```
npm install astro-aws-amplify
```

3. Nella cartella del progetto dell'app Astro, apri il `astro.config.mjs` file. Aggiorna il file per aggiungere l'adattatore. Il file dovrebbe avere il seguente aspetto.

```
import { defineConfig } from 'astro/config';
import mdx from '@astrojs/mdx';
import awsAmplify from 'astro-aws-amplify';

import sitemap from '@astrojs/sitemap';

// https://astro.build/config
export default defineConfig({
  site: 'https://example.com',
  integrations: [mdx(), sitemap()],
  adapter: awsAmplify(),
  output: 'server',
});
```

4. Conferma la modifica e invia il progetto al tuo repository Git.

   Ora sei pronto per distribuire la tua app Astro su Amplify.

5. Accedi AWS Management Console e apri la console Amplify.
6. Nella pagina Tutte le app, scegli Crea nuova app.
7. Nella pagina Inizia a creare con Amplify, scegli il tuo provider di repository Git, quindi scegli Avanti.
8. Nella pagina Aggiungi ramo del repository, procedi come segue:

   a. Seleziona il nome del repository da connettere.

   b. Seleziona il nome del ramo del repository da connettere.

   c. Scegli Next (Successivo).

9. Nella pagina delle impostazioni dell'app, individua la sezione Impostazioni di creazione. Per la directory di output di Build, immettete **`.amplify-hosting`**.
10. È inoltre necessario aggiornare i comandi di compilazione del frontend dell'app nelle specifiche di build. Per aprire le specifiche di build, scegliete Modifica file YML.
11. Nel `amplify.yml` file, individua la sezione dei comandi di compilazione del frontend. Inserisci **`mv node_modules ./.amplify-hosting/compute/default`**

    Il file delle impostazioni di build dovrebbe avere l'aspetto seguente.

```
version: 1
frontend:
    phases:
        preBuild:
            commands:
                - 'npm ci --cache .npm --prefer-offline'
        build:
            commands:
                - 'npm run build'
                - 'mv node_modules ./.amplify-hosting/compute/default'
    artifacts:
        baseDirectory: .amplify-hosting
        files:
            - '**/*'
    cache:
        paths:
            - '.npm/**/*'
```

12. Scegli Save (Salva).
13. Se desideri che Amplify sia in grado di inviare i log delle app ad CloudWatch Amazon Logs, devi abilitarlo esplicitamente nella console. Apri la sezione Impostazioni avanzate, quindi scegli Abilita i log delle app SSR nella sezione Distribuzione Server-Side Rendering (SSR).
14. Scegli Next (Successivo).
15. Nella pagina di revisione, scegli Salva e distribuisci.

Distribuisci un' SvelteKit app su Amplify Hosting

Utilizza le seguenti istruzioni per distribuire un' SvelteKit applicazione su Amplify Hosting. Puoi usare la tua applicazione o creare un'app iniziale. Per ulteriori informazioni, consulta Creazione di un progetto nella SvelteKit documentazione.

Per distribuire un' SvelteKit app con SSR su Amplify Hosting, devi aggiungere un adattatore al tuo progetto. Non gestiamo un adattatore di proprietà di Amplify per il framework. SvelteKit In questo esempio, utilizziamo il file `amplify-adapter` creato da un membro della community. L'adattatore è disponibile su github. com/gzimbron/amplify-adapter sul sito web. GitHub AWS non mantiene questo adattatore.

Per distribuire un' SvelteKit app su Amplify Hosting

1. Sul tuo computer locale, vai all' SvelteKit applicazione da distribuire.
2. Per installare l'adattatore, aprite una finestra di terminale ed eseguite il seguente comando. Questo esempio utilizza il community adapter disponibile su github. com/gzimbron/amplify-adattatore. Se utilizzi un Community Adapter diverso, sostituiscilo *`amplify-adapter`* con il nome del tuo adattatore.

```
npm install amplify-adapter
```

3. Nella cartella del progetto SvelteKit dell'app, apri il `svelte.config.js` file. Modifica il file per utilizzare `amplify-adapter` o sostituirlo *`'amplify-adapter'`* con il nome dell'adattatore. Il file dovrebbe avere l'aspetto seguente.

```
import adapter from 'amplify-adapter';
import { vitePreprocess } from '@sveltejs/vite-plugin-svelte';


/** @type {import('@sveltejs/kit').Config} */
const config = {
        // Consult https://kit.svelte.dev/docs/integrations#preprocessors
        // for more information about preprocessors
        preprocess: vitePreprocess(),

        kit: {
                // adapter-auto only supports some environments, see https://
kit.svelte.dev/docs/adapter-auto for a list.
```

```
                // If your environment is not supported, or you settled on a
 specific environment, switch out the adapter.
                // See https://kit.svelte.dev/docs/adapters for more information
 about adapters.
                adapter: adapter()
        }
};

export default config;
```

4. Conferma la modifica e invia l'applicazione al tuo repository Git.
5. Ora sei pronto per distribuire la tua SvelteKit app su Amplify.

   Accedi AWS Management Console e apri la console Amplify.
6. Nella pagina Tutte le app, scegli Crea nuova app.
7. Nella pagina Inizia a creare con Amplify, scegli il tuo provider di repository Git, quindi scegli Avanti.
8. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Seleziona il nome del repository da connettere.
   b. Seleziona il nome del ramo del repository da connettere.
   c. Scegli Next (Successivo).
9. Nella pagina delle impostazioni dell'app, individua la sezione Impostazioni di creazione. Per la directory di output di Build, immettete**`build`**.
10. È inoltre necessario aggiornare i comandi di compilazione del frontend dell'app nelle specifiche di build. Per aprire le specifiche di build, scegliete Modifica file YML.
11. Nel `amplify.yml` file, individua la sezione dei comandi di compilazione del frontend. Inserisci **`- cd build/compute/default/`** e. **`- npm i --production`**

    Il file delle impostazioni di build dovrebbe avere l'aspetto seguente.

```
version: 1
frontend:
    phases:
        preBuild:
            commands:
                - 'npm ci --cache .npm --prefer-offline'
        build:
            commands:
```

```
                - 'npm run build'
                - 'cd build/compute/default/'
                - 'npm i --production'

    artifacts:
        baseDirectory: build
        files:
            - '**/*'
    cache:
        paths:
            - '.npm/**/*'
```

12. Scegli Save (Salva).
13. Se desideri che Amplify sia in grado di inviare i log delle app ad CloudWatch Amazon Logs, devi abilitarlo esplicitamente nella console. Apri la sezione Impostazioni avanzate, quindi scegli Abilita i log delle app SSR nella sezione Distribuzione Server-Side Rendering (SSR).
14. Scegli Next (Successivo).
15. Nella pagina di revisione, scegli Salva e distribuisci.

Implementazione di applicazioni renderizzate lato server con Amplify Hosting

È possibile utilizzarla AWS Amplify per distribuire e ospitare app Web che utilizzano il rendering lato server (SSR). Amplify Hosting rileva automaticamente le applicazioni create utilizzando il framework Next.js e non è necessario eseguire alcuna configurazione manuale in. AWS Management Console

Amplify supporta anche qualsiasi framework SSR basato su Javascript con un adattatore di build open source che trasforma l'output di build di un'applicazione nella struttura di directory prevista da Amplify Hosting. Ad esempio, puoi distribuire app create con Nuxt, Astro e framework installando gli adattatori disponibili. SvelteKit

Gli utenti esperti possono utilizzare le specifiche di distribuzione per creare un adattatore di build o configurare uno script post-build.

Puoi implementare i seguenti framework su Amplify Hosting con una configurazione minima.

Next.js

- Amplify supporta le applicazioni Next.js 15 senza la necessità di un adattatore. Per iniziare, consulta Supporto Amplify per Next.js.

Nuxt.js

- Amplify supporta le implementazioni di applicazioni Nuxt.js con un adattatore preimpostato. Per iniziare, consulta Supporto Amplify per Nuxt.js.

Astro.js

- Amplify supporta le distribuzioni di applicazioni Astro.js con un adattatore comunitario. Per iniziare, consulta Supporto Amplify per Astro.js.

SvelteKit

- Amplify SvelteKit supporta l'implementazione di applicazioni con un adattatore comunitario. Per iniziare, consulta Supporto Amplify per SvelteKit.

Adattatori open source

- Usa un adattatore open source: per istruzioni sull'uso di qualsiasi adattatore non presente nell'elenco precedente, consulta. Utilizzo di adattatori open source per qualsiasi framework SSR
- Crea un adattatore di framework: gli autori del framework che desiderano integrare le funzionalità fornite da un framework possono utilizzare le specifiche di implementazione di

Amplify Hosting per configurare l'output della build in modo che sia conforme alla struttura prevista da Amplify. Per ulteriori informazioni, consulta Utilizzo della specifica di distribuzione di Amplify Hosting per configurare l'output della build.

- Configura uno script post-compilazione: puoi utilizzare le specifiche di distribuzione di Amplify Hosting per manipolare l'output della build secondo necessità per scenari specifici. Per ulteriori informazioni, consulta Utilizzo della specifica di distribuzione di Amplify Hosting per configurare l'output della build. Per vedere un esempio, consulta Distribuzione di un server Express utilizzando il manifesto di distribuzione.

Argomenti

- Supporto Amplify per Next.js
- Supporto Amplify per Nuxt.js
- Supporto Amplify per Astro.js
- Supporto Amplify per SvelteKit
- Implementazione di un'app SSR su Amplify
- Funzionalità supportate da SSR
- Risoluzione dei problemi relativi alle implementazioni SSR
- Avanzato: adattatori open source

Supporto Amplify per Next.js

Amplify supporta la distribuzione e l'hosting di app Web renderizzate sul lato server (SSR) create utilizzando Next.js. Next.js è un framework React con cui sviluppare. SPAs JavaScript Puoi distribuire app create con versioni di Next.js fino a Next.js 15, con funzionalità come l'ottimizzazione delle immagini e il middleware.

Gli sviluppatori possono utilizzare Next.js per combinare la generazione statica di siti (SSG) e SSR in un unico progetto. Le pagine SSG vengono prerenderizzate in fase di compilazione e le pagine SSR vengono prerenderizzate al momento della richiesta.

Il prerendering può migliorare le prestazioni e l'ottimizzazione dei motori di ricerca. Poiché Next.js esegue il prerendering di tutte le pagine sul server, il contenuto HTML di ogni pagina è pronto quando raggiunge il browser del client. Questo contenuto può anche essere caricato più velocemente. Tempi di caricamento più rapidi migliorano l'esperienza dell'utente finale con un sito Web e influiscono positivamente sul posizionamento SEO del sito. Il prerendering migliora anche la SEO, poiché

consente ai bot dei motori di ricerca di trovare e scansionare facilmente i contenuti HTML di un sito Web.

Next.js fornisce un supporto analitico integrato per misurare varie metriche delle prestazioni, come Time to first byte (TTFB) e First contentful paint (FCP). Per ulteriori informazioni su Next.js, consulta Guida introduttiva al sito Web Next.js.

supporto delle funzionalità Next.js

Amplify Hosting compute gestisce completamente il rendering lato server (SSR) per le app create con le versioni da 12 a 15 di Next.js.

Se hai distribuito un'app Next.js su Amplify prima del rilascio di Amplify Hosting compute nel novembre 2022, la tua app utilizza il precedente provider SSR di Amplify, Classic (solo Next.js 11). Amplify Hosting compute non supporta le app create utilizzando Next.js versione 11 o precedente. Ti consigliamo vivamente di migrare le tue app Next.js 11 al provider SSR gestito dal calcolo Amplify Hosting.

L'elenco seguente descrive le funzionalità specifiche supportate dal provider SSR di calcolo Amplify Hosting.

Funzionalità supportate

- Pagine renderizzate lato server (SSR)
- Pagine statiche
- Percorsi API
- Percorsi dinamici
- Cattura tutti i percorsi
- SSG (generazione statica)
- Rigenerazione statica incrementale (ISR)
- Routing di sottopercorsi internazionalizzato (i18n)
- Routing di domini internazionalizzato (i18n)
- Rilevamento automatico delle impostazioni locali internazionalizzato (i18n)
- Middleware
- Variabili di ambiente
- Ottimizzazione delle immagini

- Directory delle app Next.js 13

Caratteristiche non supportate

- Edge API Routes (il middleware Edge non è supportato)
- Rigenerazione statica incrementale su richiesta (ISR)
- Streaming di Next.js
- Esecuzione di middleware su risorse statiche e immagini ottimizzate
- Esecuzione del codice dopo una risposta con `unstable_after` (funzionalità sperimentale rilasciata con Next.js 15)

Immagini Next.js

La dimensione massima di output di un'immagine non può superare 4,3 MB. È possibile archiviare un file di immagine più grande da qualche parte e utilizzare il componente Next.js Image per ridimensionarlo e ottimizzarlo in un formato Webp o AVIF e quindi utilizzarlo in una dimensione più piccola.

Si noti che la documentazione di Next.js consiglia di installare il modulo di elaborazione delle immagini Sharp per consentire il corretto funzionamento dell'ottimizzazione delle immagini in produzione. Tuttavia, ciò non è necessario per le implementazioni Amplify. Amplify implementa automaticamente Sharp per te.

Distribuzione di un'applicazione SSR Next.js su Amplify

Per impostazione predefinita, Amplify distribuisce nuove app SSR utilizzando il servizio di elaborazione di Amplify Hosting con supporto per le versioni da 12 a 15 di Next.js. Amplify Hosting compute gestisce completamente le risorse necessarie per implementare un'app SSR. Le app SSR nel tuo account Amplify che hai distribuito prima del 17 novembre 2022 utilizzano il provider SSR Classic (solo Next.js 11).

Ti consigliamo vivamente di migrare le app utilizzando l'SSR Classic (solo Next.js 11) al provider SSR di elaborazione Amplify Hosting. Amplify non esegue migrazioni automatiche per te. È necessario migrare manualmente l'app e quindi avviare una nuova build per completare l'aggiornamento. Per istruzioni, consultare Migrazione di un'app SSR Next.js 11 al calcolo Amplify Hosting.

Utilizza le seguenti istruzioni per distribuire una nuova app SSR Next.js.

Per distribuire un'app SSR su Amplify utilizzando il provider SSR di calcolo Amplify Hosting

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella pagina Tutte le app, scegli Crea nuova app.
3. Nella pagina Inizia a creare con Amplify, scegli il tuo provider di repository Git, quindi scegli Avanti.
4. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Nell'elenco dei repository aggiornati di recente, seleziona il nome del repository da connettere.
   b. Nell'elenco Branch, seleziona il nome del ramo del repository da connettere.
   c. Scegli Next (Successivo).
5. L'app richiede un ruolo di servizio IAM che Amplify assume quando chiama altri servizi per tuo conto. Puoi consentire ad Amplify Hosting compute di creare automaticamente un ruolo di servizio per te oppure puoi specificare un ruolo che hai creato.
   - Per consentire ad Amplify di creare automaticamente un ruolo e associarlo alla tua app:
     - Scegli Crea e utilizza un nuovo ruolo di servizio.
   - Per allegare un ruolo di servizio creato in precedenza:
     a. Scegli Usa un ruolo di servizio esistente.
     b. Seleziona il ruolo da utilizzare dall'elenco.
6. Scegli Next (Successivo).
7. Nella pagina Revisione, scegli Salva e distribuisci.

Impostazioni del file Package.json

Quando distribuisci un'app Next.js, Amplify esamina lo script di compilazione dell'app nel file per determinare `package.json` il tipo di applicazione.

Di seguito è riportato un esempio dello script di compilazione per un'app Next.js. Lo script di compilazione `"next build"` indica che l'app supporta sia le pagine SSG che SSR. Questo script di compilazione viene utilizzato anche per le app SSG Next.js 14 o versioni successive.

```
"scripts": {
  "dev": "next dev",
```

```
  "build": "next build",
  "start": "next start"
},
```

Di seguito è riportato un esempio dello script di compilazione per un'app SSG Next.js 13 o precedente. Lo script di compilazione `"next build && next export"` indica che l'app supporta solo pagine SSG.

```
"scripts": {
  "dev": "next dev",
  "build": "next build && next export",
  "start": "next start"
},
```

Amplify le impostazioni di build per un'applicazione SSR Next.js

Dopo aver esaminato il `package.json` file dell'app, Amplify verifica le impostazioni di build dell'app. Puoi salvare le impostazioni di build nella console Amplify o in `amplify.yml` un file nella radice del tuo repository. Per ulteriori informazioni, consulta Configurazione delle impostazioni di build per un'applicazione Amplify.

Se Amplify rileva che stai distribuendo un'app SSR Next.js e non è presente `amplify.yml` alcun file, genera una specifica di build per l'app e la imposta su. `baseDirectory .next` Se stai distribuendo un'app in cui è presente un file, le impostazioni di build nel `amplify.yml` file sostituiscono tutte le impostazioni di build nella console. Pertanto, è necessario impostare manualmente il `baseDirectory` to `.next` nel file.

Di seguito è riportato un esempio delle impostazioni di build per un'app in cui `baseDirectory` è impostato su `.next`. Ciò indica che gli artefatti della build riguardano un'app Next.js che supporta pagine SSG e SSR.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
```

```
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Amplify le impostazioni di build per un'applicazione SSG Next.js 13 o precedente

Se Amplify rileva che stai distribuendo un'app SSG Next.js 13 o precedente, genera una specifica di build per l'app e la imposta su. `baseDirectory` `out` Se state distribuendo un'app in cui è presente un `amplify.yml` file, dovete impostarlo manualmente nel file. `baseDirectory` `out` La `out` directory è la cartella predefinita creata da Next.js per archiviare le risorse statiche esportate. Quando configuri le impostazioni delle specifiche di build dell'app, modifica il nome della `baseDirectory` cartella in modo che corrisponda alla configurazione dell'app.

Di seguito è riportato un esempio delle impostazioni di compilazione per un'app in cui `baseDirectory` è impostato `out` per indicare che gli artefatti di compilazione si riferiscono a un'app Next.js 13 o precedente che supporta solo pagine SSG.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: out
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Amplify le impostazioni di build per un'applicazione SSG Next.js 14 o successiva

Nella versione 14 di Next.js, il `next export` comando era obsoleto e sostituito con `output: 'export'` nel file per abilitare le esportazioni statiche. `next.config.js` Se stai distribuendo

un'applicazione Next.js 14 SSG solo nella console, Amplify genera una buildspec per l'app e la imposta su. `baseDirectory .next` Se state distribuendo un'app in cui è presente un `amplify.yml` file, dovete impostarlo manualmente nel file. `baseDirectory .next` Questa è la stessa `baseDirectory` impostazione utilizzata da Amplify per le applicazioni `WEB_COMPUTE` Next.js che supportano sia le pagine SSG che SSR.

Di seguito è riportato un esempio delle impostazioni di build per un'app Next.js 14 SSG solo con l'impostazione su. `baseDirectory .next`

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Migrazione di un'app SSR Next.js 11 al calcolo Amplify Hosting

Quando si distribuisce una nuova app Next.js, per impostazione predefinita Amplify utilizza la versione supportata più recente di Next.js. Attualmente, il provider SSR di calcolo Amplify Hosting supporta la versione 15 di Next.js.

La console Amplify rileva le app nel tuo account che sono state distribuite prima della versione di novembre 2022 del servizio di elaborazione Amplify Hosting con supporto completo per le versioni di Next.js dalla 12 alla 15. La console visualizza un banner informativo che identifica le app con filiali distribuite utilizzando il precedente provider SSR di Amplify, Classic (solo Next.js 11). Ti consigliamo vivamente di migrare le tue app al provider SSR di calcolo Amplify Hosting.

Se stai aggiornando l'applicazione Next.js 11 ospitata a Next.js 12 o versione successiva, potresti ricevere un `"target" property is no longer supported` errore quando viene attivata una distribuzione. In questo caso, è necessario migrare al calcolo di Amplify Hosting.

È necessario migrare manualmente l'app e tutte le sue filiali di produzione contemporaneamente. Un'app non può contenere sia i rami Classic (solo Next.js 11) che Next.js 12 o versioni successive.

Utilizza le seguenti istruzioni per migrare un'app al provider SSR di calcolo Amplify Hosting.

Per migrare un'app al provider SSR di calcolo Amplify Hosting

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app Next.js che desideri migrare.

> **Note**
>
> Prima di migrare un'app nella console Amplify, devi prima aggiornare il file package.json dell'app per utilizzare Next.js versione 12 o successiva.

3. Nel pannello di navigazione, scegli Impostazioni app, Generali.
4. Nella home page dell'app, la console visualizza un banner se l'app ha filiali distribuite utilizzando il provider SSR Classic (solo Next.js 11). Sul banner, scegli Migra.
5. Nella finestra di conferma della migrazione, seleziona le tre istruzioni e scegli Migra.
6. Amplify creerà e ridistribuirà la tua app per completare la migrazione.

Ripristino di una migrazione SSR

Quando distribuisci un'app Next.js, Amplify Hosting rileva le impostazioni nell'app e imposta il valore interno della piattaforma per l'app. Esistono tre valori di piattaforma validi. Un'app SSG è impostata sul valore `WEB` della piattaforma. Un'app SSR che utilizza Next.js versione 11 è impostata sul valore della piattaforma. `WEB_DYNAMIC` Un'app SSR Next.js 12 o successiva è impostata sul valore della piattaforma. `WEB_COMPUTE`

Quando esegui la migrazione di un'app utilizzando le istruzioni nella sezione precedente, Amplify modifica il valore della piattaforma della tua app da a. `WEB_DYNAMIC` `WEB_COMPUTE` Una volta completata la migrazione al calcolo di Amplify Hosting, non è possibile ripristinare la migrazione nella console. Per ripristinare la migrazione, è necessario utilizzare per ripristinare la piattaforma dell'app AWS Command Line Interface a. `WEB_DYNAMIC` Apri una finestra di terminale e inserisci il seguente comando, aggiornando l'ID dell'app e la regione con le tue informazioni uniche.

```
aws amplify update-app --app-id abcd1234 --platform WEB_DYNAMIC --region us-west-2
```

Aggiungere la funzionalità SSR a un'app Next.js statica

È possibile aggiungere funzionalità SSR a un'app Next.js statica (SSG) esistente distribuita con Amplify. Prima di iniziare il processo di conversione dell'app SSG in SSR, aggiorna l'app per utilizzare Next.js versione 12 o successiva e aggiungi la funzionalità SSR. Quindi dovrai eseguire i seguenti passaggi.

1. Usa il AWS Command Line Interface per cambiare il tipo di piattaforma dell'app.
2. Aggiungi un ruolo di servizio all'app.
3. Aggiorna la directory di output nelle impostazioni di build dell'app.
4. Aggiorna il `package.json` file dell'app per indicare che l'app utilizza SSR.

Aggiornamento della piattaforma

Esistono tre valori validi per il tipo di piattaforma. Un'app SSG è impostata sul tipo `WEB` di piattaforma. Un'app SSR che utilizza Next.js versione 11 è impostata sul tipo di piattaforma. `WEB_DYNAMIC` Per le app distribuite su Next.js 12 o versioni successive utilizzando SSR gestito da Amplify Hosting compute, il tipo di piattaforma è impostato su. `WEB_COMPUTE`

Quando hai distribuito la tua app come app SSG, Amplify ha impostato il tipo di piattaforma su. `WEB` Usa il AWS CLI per cambiare la piattaforma della tua app. `WEB_COMPUTE` Apri una finestra di terminale e inserisci il seguente comando, aggiornando il testo in rosso con l'ID e la regione dell'app univoci.

```
aws amplify update-app --app-id abcd1234 --platform WEB_COMPUTE --region us-west-2
```

Aggiungere un ruolo di servizio

Un ruolo di servizio è il ruolo AWS Identity and Access Management (IAM) che Amplify assume quando chiama altri servizi per tuo conto. Segui questi passaggi per aggiungere un ruolo di servizio a un'app SSG già distribuita con Amplify.

Per aggiungere un ruolo di servizio

1. Accedi AWS Management Console e apri la console Amplify.
2. Se non hai già creato un ruolo di servizio nel tuo account Amplify, consulta Aggiungere un ruolo di servizio per completare questo passaggio preliminare.

3. Scegli l'app statica Next.js a cui desideri aggiungere un ruolo di servizio.
4. Nel riquadro di navigazione, scegli Impostazioni app, Generali.
5. Nella pagina dei dettagli dell'app, scegli Modifica
6. Per Ruolo di servizio, scegli il nome di un ruolo di servizio esistente o il nome del ruolo di servizio creato nel passaggio 2.
7. Scegli Save (Salva).

Aggiornamento delle impostazioni di build

Prima di ridistribuire l'app con la funzionalità SSR, devi aggiornare le impostazioni di build dell'app su cui impostare la directory di output. `.next` Puoi modificare le impostazioni di build nella console Amplify o in `amplify.yml` un file archiviato nel tuo repository. Per ulteriori informazioni, consultare Configurazione delle impostazioni di build per un'applicazione Amplify.

Di seguito è riportato un esempio delle impostazioni di build per un'app in cui `baseDirectory` è impostato su. `.next`

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Aggiornamento del file package.json

Dopo aver aggiunto un ruolo di servizio e aggiornato le impostazioni di build, aggiorna il file dell'app. `package.json` Come nell'esempio seguente, imposta lo script di compilazione in modo `"next build"` che indichi che l'app Next.js supporta sia le pagine SSG che SSR.

```
"scripts": {
  "dev": "next dev",
  "build": "next build",
  "start": "next start"
},
```

Amplify rileva la modifica al file nel repository e ridistribuisce `package.json` l'app con funzionalità SSR.

Rendere le variabili di ambiente accessibili ai runtime lato server

Amplify Hosting supporta l'aggiunta di variabili di ambiente alle build dell'applicazione impostandole nella configurazione del progetto nella console Amplify.

Tuttavia, un componente del server Next.js non ha accesso a tali variabili di ambiente per impostazione predefinita. Questo comportamento è intenzionale per proteggere eventuali segreti archiviati nelle variabili di ambiente utilizzate dall'applicazione durante la fase di compilazione.

Per rendere accessibili variabili di ambiente specifiche a Next.js, puoi modificare il file delle specifiche della build Amplify per impostarle nei file di ambiente riconosciuti da Next.js. Ciò consente ad Amplify di caricare queste variabili di ambiente prima di creare l'applicazione.

> **Important**
>
> Ti consigliamo vivamente di non memorizzare credenziali, segreti o informazioni sensibili nelle variabili di ambiente poiché qualsiasi utente con accesso agli artefatti della distribuzione può leggerli.
> Per consentire alla funzione di calcolo SSR di accedere alle AWS risorse, consigliamo di utilizzare i ruoli IAM.

Il seguente esempio di specifica di build dimostra come aggiungere variabili di ambiente nella sezione dei comandi di compilazione.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
```

```
        - npm ci
    build:
      commands:
        - env | grep -e API_BASE_URL >> .env.production
        - env | grep -e NEXT_PUBLIC_ >> .env.production
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
      - .next/cache/**/*
```

In questo esempio, la sezione dei comandi di compilazione include due comandi che scrivono le variabili di ambiente nel `.env.production` file prima dell'esecuzione della build dell'applicazione. Amplify Hosting consente all'applicazione di accedere a queste variabili quando l'applicazione riceve traffico.

La riga seguente della sezione dei comandi di compilazione dell'esempio precedente mostra come prendere una variabile specifica dall'ambiente di compilazione e aggiungerla al file. `.env.production`

```
- env | grep -e API_BASE_URL -e APP_ENV >> .env.production
```

Se le variabili esistono nell'ambiente di compilazione, il `.env.production` file conterrà le seguenti variabili di ambiente.

```
API_BASE_URL=localhost
APP_ENV=dev
```

La riga seguente della sezione dei comandi di compilazione dell'esempio precedente mostra come aggiungere una variabile di ambiente con un prefisso specifico al file. `.env.production` In questo esempio, vengono aggiunte tutte le variabili con il prefisso`NEXT_PUBLIC_`.

```
- env | grep -e NEXT_PUBLIC_ >> .env.production
```

Se nell'ambiente di compilazione esistono più variabili con il `NEXT_PUBLIC_` prefisso, il `.env.production` file avrà un aspetto simile al seguente.

```
NEXT_PUBLIC_ANALYTICS_ID=abcdefghijk
NEXT_PUBLIC_GRAPHQL_ENDPOINT=uowelalsmlsadf
NEXT_PUBLIC_FEATURE_FLAG=true
```

Variabili di ambiente SSR per monorepos

Se state distribuendo un'app SSR in un monorepo e desiderate rendere accessibili variabili di ambiente specifiche a Next.js, dovete anteporre al file il prefisso root dell'applicazione. `.env.production` L'esempio seguente di specifica di build per un'app Next.js all'interno di un monorepo Nx mostra come aggiungere variabili di ambiente nella sezione dei comandi di compilazione.

```
version: 1
applications:
  - frontend:
      phases:
        preBuild:
          commands:
            - npm ci
        build:
          commands:
            - env | grep -e API_BASE_URL -e APP_ENV >> apps/app/.env.production
            - env | grep -e NEXT_PUBLIC_ >> apps/app/.env.production
            - npx nx build app
      artifacts:
        baseDirectory: dist/apps/app/.next
        files:
          - '**/*'
      cache:
        paths:
          - node_modules/**/*
      buildPath: /
    appRoot: apps/app
```

Le righe seguenti della sezione build commands dell'esempio precedente mostrano come prendere variabili specifiche dall'ambiente di compilazione e aggiungerle al `.env.production` file per un'app in un monorepo con la radice dell'applicazione. `apps/app`

```
- env | grep -e API_BASE_URL -e APP_ENV >> apps/app/.env.production
- env | grep -e NEXT_PUBLIC_ >> apps/app/.env.production
```

Distribuzione di un'app Next.js in un monorepo

Amplify supporta app in monorepo generici e app in monorepo create utilizzando npm workspace, pnpm workspace, Yarn workspace, Nx e Turborepo. Quando distribuisci la tua app, Amplify rileva automaticamente il framework di build monorepo che stai utilizzando. Amplify applica automaticamente le impostazioni di build per le app in un'area di lavoro npm, un'area di lavoro Yarn o Nx. Le app Turborepo e pnpm richiedono una configurazione aggiuntiva. Per ulteriori informazioni, consulta Configurazione delle impostazioni di build monorepo.

Per un esempio dettagliato di Nx, consulta il post del blog Share code between Next.js apps with Nx on AWS Amplify Hosting.

Supporto Amplify per Nuxt.js

Nuxt è un framework per la creazione di applicazioni web complete con Vue.js.

Adattatore

È possibile distribuire un'applicazione Nuxt.js su Amplify utilizzando un adattatore preimpostato con configurazione zero. Per ulteriori informazioni sull'adattatore, consulta la documentazione di Nuxt.

Tutorial

Per informazioni su come distribuire un'app Nuxt.js su Amplify, consulta. Distribuisci un'app Nuxt.js su Amplify Hosting

Demo

Per una dimostrazione video, vedi Nuxt Hosting With ZERO Configuration In Minutes (With) on. AWS YouTube

Supporto Amplify per Astro.js

Astro è un framework web per la creazione di applicazioni web basate sui contenuti.

Adattatore

È possibile distribuire un'applicazione Astro.js su Amplify utilizzando un community adapter. Non gestiamo un adattatore di proprietà di Amplify per il framework Astro. Tuttavia, un adattatore è

disponibile su github. com/alexnguyennz/astro-aws-amplify sul sito web. GitHub Questo adattatore è stato creato da un membro della community e non è gestito da. AWS

Tutorial

Per informazioni su come distribuire un'app Astro su Amplify, consulta. Distribuisci un'app Astro.js su Amplify Hosting

Demo

Per una dimostrazione video, vedi Come implementare un sito Web Astro AWS sul canale Amazon Web Services YouTube .

Supporto Amplify per SvelteKit

SvelteKit è un framework per la creazione di applicazioni web complete con Svelte.

Adattatore

È possibile distribuire un' SvelteKit applicazione su Amplify utilizzando un adattatore di comunità. Non gestiamo un adattatore di proprietà di Amplify per il framework. SvelteKit Tuttavia, un adattatore è disponibile su github. com/gzimbron/amplify-adapter sul sito web. GitHub Questo adattatore è stato creato da un membro della community e non è gestito da AWS.

Tutorial

Per informazioni su come distribuire un' SvelteKit app su Amplify, consulta. Distribuisci un' SvelteKit app su Amplify Hosting

Demo

Per una dimostrazione video, consulta Come distribuire un SvelteKit sito Web (con API) AWS sul canale Amazon Web Services YouTube .

Implementazione di un'app SSR su Amplify

Puoi utilizzare queste istruzioni per distribuire un'app creata con qualsiasi framework con un pacchetto di distribuzione conforme all'output di build previsto da Amplify. Se stai distribuendo un'applicazione Next.js, non è necessario alcun adattatore.

Se stai distribuendo un'app SSR che utilizza un adattatore framework, devi prima installare e configurare l'adattatore. Per istruzioni, consultare Utilizzo di adattatori open source per qualsiasi framework SSR.

Per distribuire un'app SSR su Amplify Hosting

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella pagina Tutte le app, scegli Crea nuova app.
3. Nella pagina Inizia a creare con Amplify, scegli il tuo provider di repository Git, quindi scegli Avanti.
4. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Seleziona il nome del repository da connettere.
   b. Seleziona il nome del ramo del repository da connettere.
   c. Scegli Next (Successivo).
5. Nella pagina delle impostazioni dell'app, Amplify rileva automaticamente le app SSR Next.js.

   Se stai distribuendo un'app SSR che utilizza un adattatore per un altro framework, devi abilitare esplicitamente Amazon Logs. CloudWatch Apri la sezione Impostazioni avanzate, quindi scegli Abilita i log delle app SSR nella sezione Distribuzione Server-Side Rendering (SSR).
6. L'app richiede un ruolo di servizio IAM che Amplify assume per fornire i log al tuo. Account AWS

   La procedura per aggiungere un ruolo di servizio varia a seconda che si desideri creare un nuovo ruolo o utilizzarne uno esistente.
   - Per creare un nuovo ruolo:
     - Scegli Crea e utilizza un nuovo ruolo di servizio.
   - Per utilizzare un ruolo esistente:
     a. Scegli Usa un ruolo esistente.
     b. Nell'elenco dei ruoli di servizio, seleziona il ruolo da utilizzare.
7. Scegli Next (Successivo).
8. Nella pagina Revisione, scegli Salva e distribuisci.

Funzionalità supportate da SSR

Questa sezione fornisce informazioni sul supporto di Amplify per le funzionalità SSR.

Amplify fornisce il supporto per la versione di Node.js corrispondente alla versione di Node.js utilizzata per creare l'app.

Amplify offre una funzione di ottimizzazione delle immagini integrata che supporta tutte le app SSR. Se non desideri utilizzare la funzione di ottimizzazione delle immagini predefinita, puoi implementare un caricatore di ottimizzazione delle immagini personalizzato.

Argomenti

- Supporto della versione Node.js per le app Next.js
- Ottimizzazione delle immagini per le app SSR
- Amazon CloudWatch Logs per app SSR
- Supporto SSR Amplify Next.js 11

Supporto della versione Node.js per le app Next.js

Quando Amplify crea e distribuisce un'app di calcolo Next.js, utilizza Node.js la versione di runtime che corrisponde alla versione Node.js principale utilizzata per creare l'app.

È possibile specificare la Node.js versione da utilizzare nella funzione Live package override nella console Amplify. Per ulteriori informazioni sulla configurazione degli aggiornamenti live dei pacchetti, consulta. Utilizzo di versioni specifiche di pacchetti e dipendenze nell'immagine di build È inoltre possibile specificare la Node.js versione utilizzando altri meccanismi, ad esempio nvm i comandi. Se non specifichi una versione, per impostazione predefinita Amplify utilizza la versione corrente utilizzata dal contenitore di build Amplify.

Ottimizzazione delle immagini per le app SSR

Amplify Hosting offre una funzionalità integrata di ottimizzazione delle immagini che supporta tutte le app SSR. Con l'ottimizzazione delle immagini di Amplify, puoi fornire immagini di alta qualità nel formato, nella dimensione e nella risoluzione corretti per il dispositivo che vi accede, mantenendo al contempo la dimensione del file più piccola possibile.

Attualmente, è possibile utilizzare il componente Next.js Image per ottimizzare le immagini su richiesta oppure è possibile implementare un caricatore di immagini personalizzato. Se utilizzi

Next.js 13 o versioni successive, non devi intraprendere ulteriori azioni per utilizzare la funzione di ottimizzazione delle immagini di Amplify. Se state implementando un caricatore di immagini personalizzato, consultate il seguente argomento Utilizzo di un caricatore di immagini personalizzato.

Utilizzo di un caricatore di immagini personalizzato

Se utilizzi un caricatore di immagini personalizzato, Amplify rileva il caricatore nel file dell'applicazione e non utilizza la funzione `next.config.js` di ottimizzazione delle immagini integrata. Per ulteriori informazioni sui caricatori personalizzati supportati da Next.js, consulta la documentazione delle immagini Next.js.

Amazon CloudWatch Logs per app SSR

Amplify invia informazioni sul tuo runtime SSR ad CloudWatch Amazon Logs nel tuo. Account AWS Quando distribuisci un'app SSR, l'app richiede un ruolo di servizio IAM che Amplify assume quando chiama altri servizi per tuo conto. Puoi consentire ad Amplify Hosting compute di creare automaticamente un ruolo di servizio per te oppure puoi specificare un ruolo che hai creato.

Se scegli di consentire ad Amplify di creare un ruolo IAM per te, il ruolo avrà già le autorizzazioni per creare log. CloudWatch Se crei il tuo ruolo IAM, dovrai aggiungere le seguenti autorizzazioni alla tua policy per consentire ad Amplify di accedere ad Amazon Logs. CloudWatch

```
logs:CreateLogStream
logs:CreateLogGroup
logs:DescribeLogGroups
logs:PutLogEvents
```

Per ulteriori informazioni sui ruoli di servizio, consulta Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend.

Supporto SSR Amplify Next.js 11

Se hai distribuito un'app Next.js su Amplify prima del rilascio di Amplify Hosting compute il 17 novembre 2022, la tua app utilizza il precedente provider SSR di Amplify, Classic (solo Next.js 11). La documentazione contenuta in questa sezione si applica solo alle app distribuite utilizzando il provider SSR Classic (solo Next.js 11).

> **Note**
>
> Ti consigliamo vivamente di migrare le tue app Next.js 11 al provider SSR gestito dal calcolo Amplify Hosting. Per ulteriori informazioni, consulta Migrazione di un'app SSR Next.js 11 al calcolo Amplify Hosting.

L'elenco seguente descrive le funzionalità specifiche supportate dal provider SSR Amplify Classic (solo Next.js 11).

Funzionalità supportate

- Pagine renderizzate lato server (SSR)
- Pagine statiche
- Percorsi API
- Percorsi dinamici
- Cattura tutti i percorsi
- SSG (generazione statica)
- Rigenerazione statica incrementale (ISR)
- Routing di sottopercorsi internazionalizzato (i18n)
- Variabili di ambiente

Caratteristiche non supportate

- Ottimizzazione delle immagini
- Rigenerazione statica incrementale su richiesta (ISR)
- Routing di domini internazionalizzato (i18n)
- Rilevamento automatico delle impostazioni locali internazionalizzato (i18n)
- Middleware
- Middleware Edge
- Percorsi dell'API Edge

Prezzi delle app SSR Next.js 11

Quando distribuisci l'app Next.js 11 SSR, Amplify crea risorse di backend aggiuntive nel tuo account, tra cui: AWS

- Un bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) che archivia le risorse per gli asset statici della tua app. Per informazioni sui costi di Amazon S3, consulta la pagina dei prezzi di Amazon S3.
- Una CloudFront distribuzione Amazon per servire l'app. Per informazioni sugli CloudFront addebiti, consulta la pagina CloudFront dei prezzi di Amazon.
- Quattro funzioni Lambda @Edge per personalizzare il contenuto fornito CloudFront .

AWS Identity and Access Management autorizzazioni per le app SSR Next.js 11

Amplify AWS Identity and Access Management richiede le autorizzazioni (IAM) per distribuire un'app SSR. Per le app SSR, Amplify distribuisce risorse come un bucket Amazon S3, una distribuzione CloudFront , funzioniLambda@Edge, una coda Amazon SQS (se si utilizza ISR) e ruoli IAM. Senza le autorizzazioni minime richieste, riceverai un errore quando tenti di distribuire la tua app SSR`Access Denied`. Per fornire ad Amplify le autorizzazioni richieste, è necessario specificare un ruolo di servizio.

Per creare un ruolo di servizio IAM che Amplify assume quando chiama altri servizi per tuo conto, consulta. Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend Queste istruzioni mostrano come creare un ruolo che alleghi la policy gestita. `AdministratorAccess-Amplify`

La policy `AdministratorAccess-Amplify` gestita fornisce l'accesso a più AWS servizi, incluse le azioni IAM, e deve essere considerata potente quanto la policy. `AdministratorAccess` Questa policy fornisce più autorizzazioni di quelle necessarie per distribuire l'app SSR.

Si consiglia di seguire la migliore pratica di concedere il minimo privilegio e ridurre le autorizzazioni concesse al ruolo di servizio. Invece di concedere le autorizzazioni di accesso di amministratore al tuo ruolo di servizio, puoi creare una politica IAM gestita dai clienti che conceda solo le autorizzazioni necessarie per distribuire la tua app SSR. Per istruzioni sulla creazione di una policy gestita dal cliente, consulta Creazione di policy IAM nella IAM User Guide.

Se crei una policy personalizzata, consulta il seguente elenco delle autorizzazioni minime richieste per implementare un'app SSR.

```
acm:DescribeCertificate
acm:DescribeCertificate
acm:ListCertificates
acm:RequestCertificate
cloudfront:CreateCloudFrontOriginAccessIdentity
cloudfront:CreateDistribution
cloudfront:CreateInvalidation
cloudfront:GetDistribution
cloudfront:GetDistributionConfig
cloudfront:ListCloudFrontOriginAccessIdentities
cloudfront:ListDistributions
cloudfront:ListDistributionsByLambdaFunction
cloudfront:ListDistributionsByWebACLId
cloudfront:ListFieldLevelEncryptionConfigs
cloudfront:ListFieldLevelEncryptionProfiles
cloudfront:ListInvalidations
cloudfront:ListPublicKeys
cloudfront:ListStreamingDistributions
cloudfront:UpdateDistribution
cloudfront:TagResource
cloudfront:UntagResource
cloudfront:ListTagsForResource
iam:AttachRolePolicy
iam:CreateRole
iam:CreateServiceLinkedRole
iam:GetRole
iam:PutRolePolicy
iam:PassRole
lambda:CreateFunction
lambda:EnableReplication
lambda:DeleteFunction
lambda:GetFunction
lambda:GetFunctionConfiguration
lambda:PublishVersion
lambda:UpdateFunctionCode
lambda:UpdateFunctionConfiguration
lambda:ListTags
lambda:TagResource
lambda:UntagResource
route53:ChangeResourceRecordSets
route53:ListHostedZonesByName
route53:ListResourceRecordSets
s3:CreateBucket
```

```
s3:GetAccelerateConfiguration
s3:GetObject
s3:ListBucket
s3:PutAccelerateConfiguration
s3:PutBucketPolicy
s3:PutObject
s3:PutBucketTagging
s3:GetBucketTagging
lambda:ListEventSourceMappings
lambda:CreateEventSourceMapping
iam:UpdateAssumeRolePolicy
iam:DeleteRolePolicy
sqs:CreateQueue           // SQS only needed if using ISR feature
sqs:DeleteQueue
sqs:GetQueueAttributes
sqs:SetQueueAttributes
amplify:GetApp
amplify:GetBranch
amplify:UpdateApp
amplify:UpdateBranch
```

Risoluzione dei problemi relativi alle implementazioni SSR di Next.js 11

Se riscontri problemi imprevisti durante la distribuzione di un'app SSR Classic (solo Next.js 11) con Amplify, consulta i seguenti argomenti per la risoluzione dei problemi.

Argomenti

- La directory di output della mia applicazione viene sovrascritta
- Ricevo un errore 404 dopo aver distribuito il mio sito SSR
- Nella mia applicazione manca la regola di riscrittura per CloudFront le distribuzioni SSR
- La mia applicazione è troppo grande per essere distribuita
- La mia build fallisce con un errore di memoria esaurita
- La mia applicazione ha filiali SSR e SSG
- La mia applicazione memorizza i file statici in una cartella con un percorso riservato
- La mia applicazione ha raggiunto un limite CloudFront
- Le funzioni Lambda @Edge vengono create nella regione Stati Uniti orientali (Virginia settentrionale)
- La mia applicazione Next.js utilizza funzionalità non supportate

- Le immagini nella mia applicazione Next.js non si caricano
- Regioni non supportate

La directory di output della mia applicazione viene sovrascritta

La directory di output per un'app Next.js distribuita con Amplify deve essere impostata su. `.next` Se la directory di output della tua app viene sovrascritta, controlla il file. `next.config.js` Per impostare la directory di output della build come predefinita`.next`, rimuovi la seguente riga dal file:

```
distDir: 'build'
```

Verifica che la directory di output sia impostata su `.next` nelle impostazioni di build. Per informazioni sulla visualizzazione delle impostazioni di build dell'app, consultaConfigurazione delle impostazioni di build per un'applicazione Amplify.

Di seguito è riportato un esempio delle impostazioni di build per un'app in cui `baseDirectory` è impostato su`.next`.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .next
    files:
      - '**/*'
  cache:
    paths:
      - node_modules/**/*
```

Ricevo un errore 404 dopo aver distribuito il mio sito SSR

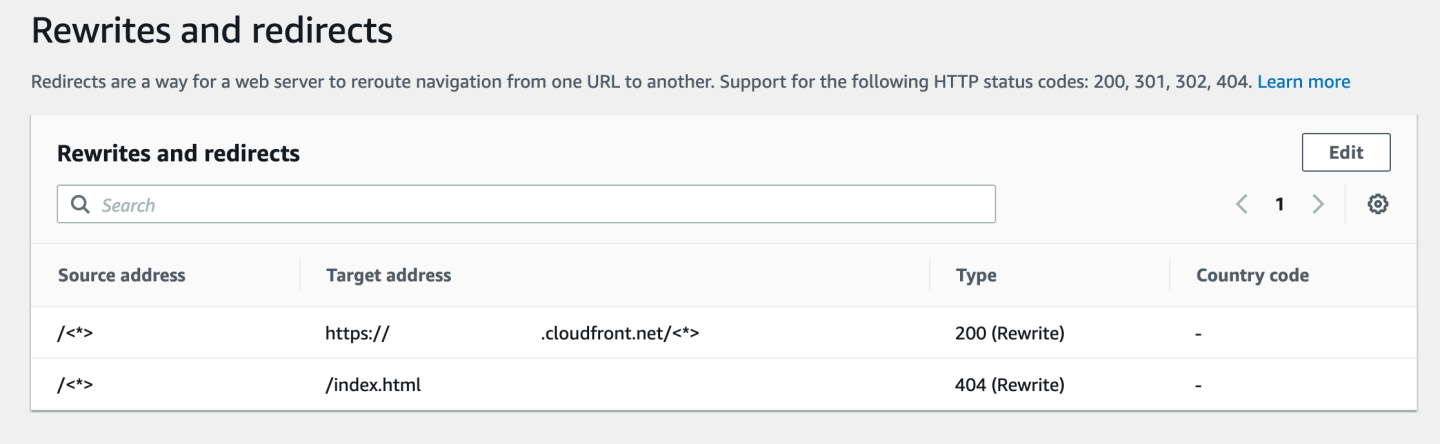
Se si verifica un errore 404 dopo la distribuzione del sito, il problema potrebbe essere causato dall'override della directory di output. Per controllare il `next.config.js` file e verificare la directory

di output della build corretta nelle specifiche di build dell'app, segui i passaggi dell'argomento precedente,. La directory di output della mia applicazione viene sovrascritta

Nella mia applicazione manca la regola di riscrittura per CloudFront le distribuzioni SSR

Quando distribuisci un'app SSR, Amplify crea una regola di riscrittura per le tue distribuzioni SSR. CloudFront Se non riesci ad accedere alla tua app in un browser web, verifica che la regola di CloudFront riscrittura esista per la tua app nella console Amplify. Se manca, puoi aggiungerla manualmente o ridistribuire l'app.

Per visualizzare o modificare le regole di riscrittura e reindirizzamento di un'app nella console Amplify, nel pannello di navigazione, scegli Impostazioni app, quindi Riscritture e reindirizzamenti. La schermata seguente mostra un esempio delle regole di riscrittura che Amplify crea per te quando distribuisci un'app SSR. Nota che in questo esempio esiste una regola di riscrittura. CloudFront



La mia applicazione è troppo grande per essere distribuita

Amplify limita la dimensione di una distribuzione SSR a 50 MB. Se provi a distribuire un'app SSR Next.js su Amplify e ricevi un `RequestEntityTooLargeException` errore, l'app è troppo grande per essere distribuita. Puoi provare a risolvere questo problema aggiungendo del codice di pulizia della cache al tuo file. `next.config.js`

Di seguito è riportato un esempio di codice contenuto nel `next.config.js` file che esegue la pulizia della cache.

```
module.exports = {
    webpack: (config, { buildId, dev, isServer, defaultLoaders, webpack }) => {
        config.optimization.splitChunks.cacheGroups = { }
        config.optimization.minimize = true;
        return config
      },
```

```
}
```

La mia build fallisce con un errore di memoria esaurita

Next.js consente di memorizzare nella cache gli elementi della build per migliorare le prestazioni nelle build successive. Inoltre, il AWS CodeBuild contenitore di Amplify comprime e carica questa cache su Amazon S3, per tuo conto, per migliorare le prestazioni di build successive. Ciò potrebbe causare il fallimento della compilazione con un errore di memoria esaurita.

Esegui le seguenti azioni per evitare che l'app superi il limite di memoria durante la fase di compilazione. Innanzitutto, rimuovi `.next/cache/**/*` dalla sezione cache.paths delle impostazioni di build. Quindi, rimuovi la variabile di `NODE_OPTIONS` ambiente dal file delle impostazioni di build. Invece, imposta la variabile di `NODE_OPTIONS` ambiente nella console Amplify per definire il limite massimo di memoria del nodo. Per ulteriori informazioni sull'impostazione delle variabili di ambiente utilizzando la console Amplify, vedere. Impostazione delle variabili di ambiente

Dopo aver apportato queste modifiche, riprova a eseguire la build. Se riesce, aggiungilo `.next/cache/**/*` nuovamente alla sezione cache.paths del file delle impostazioni di build.

Per ulteriori informazioni sulla configurazione della cache di Next.js per migliorare le prestazioni di compilazione, consulta AWS CodeBuild sul sito Web Next.js.

La mia applicazione ha filiali SSR e SSG

Non è possibile implementare un'app con filiali SSR e SSG. Se devi implementare sia filiali SSR che SSG, devi implementare un'app che utilizzi solo filiali SSR e un'altra app che utilizzi solo filiali SSG.

La mia applicazione memorizza i file statici in una cartella con un percorso riservato

Next.js può servire file statici da una cartella denominata `public` memorizzata nella directory principale del progetto. Quando distribuisci e ospiti un'app Next.js con Amplify, il tuo progetto non può includere cartelle con il percorso. `public/static` Amplify riserva `public/static` il percorso da utilizzare durante la distribuzione dell'app. Se l'app include questo percorso, è necessario rinominare la `static` cartella prima di distribuirla con Amplify.

La mia applicazione ha raggiunto un limite CloudFront

CloudFront le quote di servizio limitano l' AWS account a 25 distribuzioni con funzioni Lambda @Edge collegate. Se superi questa quota, puoi eliminare tutte le CloudFront distribuzioni inutilizzate dal tuo account o richiedere un aumento della quota. Per ulteriori informazioni, consulta Richiesta di un aumento di quota nella Guida per l'utente per Service Quotas.

Le funzioni Lambda @Edge vengono create nella regione Stati Uniti orientali (Virginia settentrionale)

Quando distribuisci un'app Next.js, Amplify crea funzioni Lambda @Edge per personalizzare il contenuto che distribuisce. CloudFront Le funzioni Lambda @Edge vengono create nella regione Stati Uniti orientali (Virginia settentrionale), non nella regione in cui viene distribuita l'app. Questa è una restrizione Lambda @Edge. Per ulteriori informazioni sulle funzioni Lambda @Edge, consulta Restrizioni sulle funzioni edge nella Amazon CloudFront Developer Guide.

La mia applicazione Next.js utilizza funzionalità non supportate

Le app distribuite con Amplify supportano le versioni principali di Next.js fino alla versione 11. Per un elenco dettagliato delle funzionalità di Next.js supportate e non supportate da Amplify, vedere. supported features

Quando si distribuisce una nuova app Next.js, Amplify utilizza la versione supportata più recente di Next.js per impostazione predefinita. Se disponi di un'app Next.js esistente che hai distribuito su Amplify con una versione precedente di Next.js, puoi migrare l'app al provider SSR di calcolo Amplify Hosting. Per istruzioni, consultare Migrazione di un'app SSR Next.js 11 al calcolo Amplify Hosting.

Le immagini nella mia applicazione Next.js non si caricano

Quando aggiungi immagini all'app Next.js utilizzando il `next/image` componente, la dimensione dell'immagine non può superare 1 MB. Quando distribuisci l'app su Amplify, le immagini di dimensioni superiori a 1 MB restituiranno un errore 503. Ciò è causato da un limite Lambda @Edge che limita la dimensione di una risposta generata da una funzione Lambda, inclusi header e body, a 1 MB.

Il limite di 1 MB si applica ad altri elementi dell'app, come file PDF e documenti.

Regioni non supportate

Amplify non supporta la distribuzione di app SSR Classic (solo Next.js 11) in tutte le AWS regioni in cui Amplify è disponibile. L'SSR classico (solo Next.js 11) non è supportato nelle seguenti regioni: Europa (Milano) eu-south-1, Medio Oriente (Bahrein) me-south-1 e Asia Pacifico (Hong Kong) ap-east-1.

Risoluzione dei problemi relativi alle implementazioni SSR

Se riscontri problemi imprevisti durante la distribuzione di un'app SSR con Amplify Hosting compute, consulta Risoluzione dei problemi relativi alle applicazioni renderizzate lato server il capitolo sulla risoluzione dei problemi di Amplify.

Avanzato: adattatori open source

Gli autori del framework possono utilizzare le specifiche di distribuzione basate sul file system per sviluppare adattatori di build open source personalizzati per i propri framework specifici. Questi adattatori trasformeranno l'output di build di un'app in un pacchetto di distribuzione conforme alla struttura di directory prevista da Amplify Hosting. Questo pacchetto di distribuzione includerà tutti i file e le risorse necessari per ospitare un'app, inclusa la configurazione di runtime, come le regole di routing.

Se non utilizzi un framework, puoi sviluppare la tua soluzione per generare un output di build che Amplify si aspetta.

Argomenti

- Utilizzo della specifica di distribuzione di Amplify Hosting per configurare l'output della build
- Distribuzione di un server Express utilizzando il manifesto di distribuzione
- Integrazione dell'ottimizzazione delle immagini per gli autori del framework
- Utilizzo di adattatori open source per qualsiasi framework SSR

Utilizzo della specifica di distribuzione di Amplify Hosting per configurare l'output della build

La specifica di distribuzione di Amplify Hosting è una specifica basata su file system che definisce la struttura di directory che facilita le distribuzioni su Amplify Hosting. Un framework può generare questa struttura di directory prevista come output del suo comando build, consentendo al framework di sfruttare le primitive di servizio di Amplify Hosting. Amplify Hosting comprende la struttura del pacchetto di distribuzione e lo distribuisce di conseguenza.

Per una dimostrazione video che spiega come utilizzare le specifiche di distribuzione, vedi Come ospitare qualsiasi sito Web utilizzando AWS Amplify il YouTube canale Amazon Web Services.

Di seguito è riportato un esempio della struttura di cartelle prevista da Amplify per il pacchetto di distribuzione. Ad alto livello, ha una cartella denominata`static`, una cartella denominata `compute` e un file manifesto di distribuzione denominato. `deploy-manifest.json`

```
.amplify-hosting/
```

```
### compute/
#   ### default/
#       ### chunks/
#       #   ### app/
#       #       ### _nuxt/
#       #       #   ### index-xxx.mjs
#       #       #   ### index-styles.xxx.js
#       #       ### server.mjs
#       ### node_modules/
#       ### server.js
### static/
#   ### css/
#   #   ### nuxt-google-fonts.css
#   ### fonts/
#   #   ### font.woff2
#   ### _nuxt/
#   #   ### builds/
#   #   #   ### latest.json
#   #   ### entry.xxx.js
#   ### favicon.ico
#   ### robots.txt
### deploy-manifest.json
```

Supporto primitivo Amplify SSR

La specifica di implementazione di Amplify Hosting definisce un contratto che si avvicina strettamente alle seguenti primitive.

Risorse statiche

Fornisce ai framework la possibilità di ospitare file statici.

Calcolo

Fornisce ai framework la possibilità di eseguire un server HTTP Node.js sulla porta 3000.

Ottimizzazione delle immagini

Fornisce ai framework un servizio per ottimizzare le immagini in fase di esecuzione.

Regole di routing

Fornisce ai framework un meccanismo per mappare i percorsi delle richieste in entrata verso obiettivi specifici.

La .amplify-hosting/static directory

È necessario inserire nella `.amplify-hosting/static` directory tutti i file statici accessibili al pubblico che devono essere serviti dall'URL dell'applicazione. I file all'interno di questa directory vengono serviti tramite la primitiva static assets.

I file statici sono accessibili nella radice (/) dell'URL dell'applicazione senza alcuna modifica al contenuto, al nome del file o all'estensione. Inoltre, le sottodirectory vengono mantenute nella struttura degli URL e vengono visualizzate prima del nome del file. Ad esempio, `.amplify-hosting/static/favicon.ico` verranno servite da `https://myAppId.amplify-hostingapp.com/favicon.ico` e `.amplify-hosting/static/_nuxt/main.js` verranno servite da `https://myAppId.amplify-hostingapp.com/_nuxt/main.js`

Se un framework supporta la possibilità di modificare il percorso di base dell'applicazione, deve anteporre il percorso di base agli asset statici all'interno della `.amplify-hosting/static` directory. Ad esempio, se il percorso di base è`/folder1/folder2`, l'output di build per una risorsa statica chiamata `main.css` sarà. `.amplify-hosting/static/folder1/folder2/main.css`

La .amplify-hosting/compute directory

Una singola risorsa di elaborazione è rappresentata da una singola sottodirectory denominata `default` contenuta all'interno della `.amplify-hosting/compute` directory. Il percorso è. `.amplify-hosting/compute/default` Questa risorsa di calcolo è mappata alla primitiva di calcolo di Amplify Hosting.

Il contenuto della `default` sottodirectory deve essere conforme alle seguenti regole.

- Un file deve esistere nella radice della `default` sottodirectory, per fungere da punto di ingresso alla risorsa di calcolo.
- Il file del punto di ingresso deve essere un modulo Node.js e deve avviare un server HTTP in ascolto sulla porta 3000.
- È possibile inserire altri file nella `default` sottodirectory e farvi riferimento dal codice contenuto nel file del punto di ingresso.
- Il contenuto della sottodirectory deve essere autonomo. Il codice nel modulo del punto di ingresso non può fare riferimento a nessun modulo al di fuori della sottodirectory. Nota che i framework possono raggruppare il loro server HTTP nel modo che preferiscono. Se il processo di calcolo può essere avviato con il `node server.js` comando, where `server.js is` is the name del file di ingresso, dall'interno della sottodirectory, Amplify considera la struttura della directory conforme alle specifiche di distribuzione.

Amplify Hosting raggruppa e distribuisce tutti i file all'interno della sottodirectory in una risorsa di `default` elaborazione fornita. A ogni risorsa di elaborazione vengono allocati 512 MB di storage temporaneo. Questo storage non è condiviso tra le istanze di esecuzione, ma è condiviso tra le chiamate successive all'interno della stessa istanza di esecuzione. Le istanze di esecuzione sono limitate a un tempo di esecuzione massimo di 15 minuti e l'unico percorso scrivibile all'interno dell'istanza di esecuzione è la directory. `/tmp` La dimensione compressa di ogni pacchetto di risorse di elaborazione non può superare i 220 MB. Ad esempio, la `.amplify/compute/default` sottodirectory non può superare i 220 MB quando è compressa.

Il file .amplify-hosting/deploy-manifest.json

Utilizzate il `deploy-manifest.json` file per archiviare i dettagli di configurazione e i metadati per una distribuzione. Come minimo, un `deploy-manifest.json` file deve includere un `version` attributo, l'`routes`attributo con un percorso generico specificato e l'`framework`attributo con i metadati del framework specificati.

La seguente definizione dell'oggetto illustra la configurazione per un manifesto di distribuzione.

```
type DeployManifest = {
  version: 1;
  routes: Route[];
  computeResources?: ComputeResource[];
  imageSettings?: ImageSettings;
  framework: FrameworkMetadata;
};
```

I seguenti argomenti descrivono i dettagli e l'utilizzo di ogni attributo nel manifesto di distribuzione.

Utilizzo dell'attributo version

L'`version`attributo definisce la versione della specifica di distribuzione che si sta implementando. Attualmente, l'unica versione per le specifiche di distribuzione di Amplify Hosting è la versione 1. Il seguente esempio JSON dimostra l'utilizzo dell'attributo. `version`

```
"version": 1
```

Utilizzo dell'attributo routes

L'`routes`attributo consente ai framework di sfruttare la primitiva delle regole di routing di Amplify Hosting. Le regole di routing forniscono un meccanismo per instradare i percorsi delle richieste

in entrata verso una destinazione specifica nel pacchetto di distribuzione. Le regole di routing determinano solo la destinazione di una richiesta in entrata e vengono applicate dopo che la richiesta è stata trasformata dalle regole di riscrittura e reindirizzamento. Per ulteriori informazioni su come Amplify Hosting gestisce le riscritture e i reindirizzamenti, consulta. Configurazione di reindirizzamenti e riscritture per un'applicazione Amplify

Le regole di routing non riscrivono o trasformano la richiesta. Se una richiesta in entrata corrisponde al modello di percorso di una rotta, la richiesta viene indirizzata così com'è alla destinazione della rotta.

Le regole di routing specificate nell'`routes`array devono essere conformi alle seguenti regole.

- È necessario specificare un percorso onnicomprensivo. Un percorso generico ha lo `/*` schema che corrisponde a tutte le richieste in arrivo.
- L'`routes`array può contenere un massimo di 25 elementi.
- È necessario specificare un `Static` percorso o un `Compute` percorso.
- Se si specifica un `Static` percorso, la `.amplify-hosting/static` directory deve esistere.
- Se si specifica una `Compute` rotta, la `.amplify-hosting/compute` directory deve esistere.
- Se si specifica un `ImageOptimization` percorso, è necessario specificare anche un `Compute` percorso. Ciò è necessario perché l'ottimizzazione delle immagini non è ancora supportata per applicazioni puramente statiche.

La seguente definizione dell'oggetto illustra la configurazione dell'`Route`oggetto.

```
type Route = {
  path: string;
  target: Target;
  fallback?: Target;
}
```

La tabella seguente descrive le proprietà dell'`Route`oggetto.

Chiave	Tipo	Campo obbligatorio	Descrizione
path	Stringa	Sì	Definisce uno schema che corrisponde ai percorsi delle richieste

Chiave	Tipo	Campo obbligatorio	Descrizione
			in entrata (esclusa querystring). La lunghezza massima del percorso è di 255 caratteri. Un percorso deve iniziare con la barra `/` in avanti. Un percorso può contenere uno qualsiasi dei seguenti caratteri: [A-Z], [a-z], [0-9], [_-.*$/~"'@: +]. Per la corrispon denza dei modelli, sono supportati solo i seguenti caratteri jolly: • `*`(corrisponde a 0 o più caratteri) • Il `/*` pattern è chiamato pattern generico e corrisponderà a tutte le richieste in arrivo.

Chiave	Tipo	Campo obbligatorio	Descrizione
target	Target	Sì	Un oggetto che definisce l'obiettivo verso cui indirizzare la richiesta corrispondente. Se viene specificata una `Compute` rotta, `ComputeResource` deve esistere una corrispondente. Se viene specificata una `ImageOptimization` rotta, `imageSettings` deve essere specificata anche questa.

Chiave	Tipo	Campo obbligatorio	Descrizione
riserva	Target	No	Un oggetto che definisce l'obiettivo su cui effettuare il fallback se il target originale restituisce un errore 404. Il `target` tipo e il `fallback` tipo non possono essere gli stessi per un percorso specificato. Ad esempio, il fallback from `Static` to non `Static` è consentito. I fallback sono supportati solo per le richieste GET che non hanno un corpo. Se nella richiesta è presente un corpo, verrà eliminato durante il fallback.

La seguente definizione dell'oggetto illustra la configurazione dell'oggetto. `Target`

```
type Target = {
  kind: TargetKind;
  src?: string;
  cacheControl?: string;
}
```

La tabella seguente descrive le proprietà dell'`Target`oggetto.

Chiave	Tipo	Campo obbligatorio	Descrizione
gentile	Tipo di bersaglio	Sì	E `enum` questo definisce il tipo di bersaglio. I valori validi sono `Static`, `Compute` e `ImageOptimization` .
src	Stringa	Sì per `Compute` No per altre primitive	Una stringa che specifica il nome della sottodirectory nel pacchetto di distribuzione che contiene il codice eseguibile della primitiva. Valido e richiesto solo per la primitiva Compute. Il valore deve puntare a una delle risorse di elaborazione presenti nel pacchetto di distribuzione. Attualmente, l'unico valore supportato per questo campo è. `default`
CacheControl	Stringa	No	Una stringa che specifica il valore dell'intestazione Cache-Control da applicare alla risposta. Valido solo per Static

Chiave	Tipo	Campo obbligatorio	Descrizione
			e per le primitive. ImageOptimization Il valore specificato viene sovrascritto dalle intestazioni personalizzate. Per ulteriori informazioni sulle intestazioni dei clienti di Amplify Hosting, consulta. Impostazione di intestazioni personalizzate per un'app Amplify Note Questa intestazione Cache-Control viene applicata solo alle risposte riuscite con un codice di stato impostato su 200 (OK).

La seguente definizione dell'oggetto illustra l'utilizzo dell'enumerazione. `TargetKind`

```
enum TargetKind {
  Static = "Static",
  Compute = "Compute",
  ImageOptimization = "ImageOptimization"
```

```
}
```

L'elenco seguente specifica i valori validi per l'enum. `TargetKind`

Statico

Indirizza le richieste alla primitiva degli asset statici.

Calcolo

Indirizza le richieste alla primitiva di calcolo.

ImageOptimization

Indirizza le richieste alla primitiva di ottimizzazione delle immagini.

Il seguente esempio JSON dimostra l'utilizzo dell'`routes`attributo con più regole di routing specificate.

```
"routes": [
    {
      "path": "/_nuxt/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/_nuxt/builds/meta/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/_nuxt/builds/*",
      "target": {
        "cacheControl": "public, max-age=1, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/_nuxt/*",
```

```
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ]
```

Per ulteriori informazioni sulla specificazione delle regole di routing nel manifesto di distribuzione, vedere Le migliori pratiche per la configurazione delle regole di routing

Utilizzo dell'attributo ComputeResources

L'`computeResources`attributo consente ai framework di fornire metadati sulle risorse di calcolo fornite. A ogni risorsa di elaborazione deve essere associata una route corrispondente.

La seguente definizione dell'oggetto illustra l'utilizzo dell'oggetto. `ComputeResource`

```
type ComputeResource = {
  name: string;
  runtime: ComputeRuntime;
  entrypoint: string;
};

type ComputeRuntime = 'nodejs16.x' | 'nodejs18.x' | 'nodejs20.x';
```

La tabella seguente descrive le proprietà dell'`ComputeResource`oggetto.

Chiave	Tipo	Campo obbligatorio	Descrizione
nome	Stringa	Sì	Speciifica il nome della risorsa di calcolo. Il nome deve corrispon dere al nome della sottodirectory all'inter no di. `.amplify-hosting/compute directory` Per la versione 1 della specifica di distribuz ione, l'unico valore valido è`default`.
runtime	ComputeRuntime	Sì	Definisce il runtime per la risorsa di calcolo fornita. I valori validi sono `nodejs16.x` , `nodejs18.x` e `nodejs20.x` .
punto di ingresso	Stringa	Sì	Speciifica il nome del file iniziale da cui verrà eseguito il codice per la risorsa di calcolo specifica ta. Il file deve trovarsi all'interno della sottodirectory che rappresenta una risorsa di calcolo.

Se hai una struttura di directory simile alla seguente.

```
.amplify-hosting
|---compute
|   |---default
|       |---index.js
```

Il codice JSON per l'`computeResource`attributo sarà simile al seguente.

```
"computeResources": [
    {
      "name": "default",
      "runtime": "nodejs16.x",
      "entrypoint": "index.js",
    }
  ]
```

Utilizzo dell'attributo imageSettings

L'`imageSettings`attributo consente ai framework di personalizzare il comportamento della primitiva di ottimizzazione delle immagini, che fornisce l'ottimizzazione su richiesta delle immagini in fase di esecuzione.

La seguente definizione dell'oggetto dimostra l'utilizzo dell'oggetto. `ImageSettings`

```
type ImageSettings = {
  sizes: number[];
  domains: string[];
  remotePatterns: RemotePattern[];
  formats: ImageFormat[];
  minumumCacheTTL: number;
  dangerouslyAllowSVG: boolean;
};

type ImageFormat = 'image/avif' | 'image/webp' | 'image/png' | 'image/jpeg';
```

La tabella seguente descrive le proprietà dell'`ImageSettings`oggetto.

Chiave	Tipo	Campo obbligatorio	Descrizione
dimensioni	Numero []	Sì	Una serie di larghezze di immagine supportate.
domains	Stringa []	Sì	Una serie di domini esterni consentiti che possono utilizzare l'ottimizzazione delle immagini. Lascia l'array vuoto per consentire solo al dominio di distribuzione di utilizzare l'ottimizzazione delle immagini.
Pattern remoti	RemotePattern[]	Sì	Una serie di pattern esterni consentiti che possono utilizzare l'ottimizzazione delle immagini. Simile ai domini, ma offre un maggiore controllo con le espressioni regolari (regex).
formati	ImageFormat[]	Sì	Una serie di formati di immagini di output consentiti.
CachetI minimo	Numero	Sì	La durata della cache in secondi per le immagini ottimizzate.
Pericolosamente consente SVG	Booleano	Sì	Consente l'immagine di input SVG.

Chiave	Tipo	Campo obbligatorio	Descrizione
			URLs Questa opzione è disattiva ta per impostazione predefinita per motivi di sicurezza.

La seguente definizione dell'oggetto illustra l'utilizzo dell'`RemotePattern`oggetto.

```
type RemotePattern = {
  protocol?: 'https';
  hostname: string;
  port?: string;
  pathname?: string;
}
```

La tabella seguente descrive le proprietà dell'`RemotePattern`oggetto.

Chiave	Tipo	Campo obbligatorio	Descrizione
protocol	Stringa	No	Il protocollo del pattern remoto consentito. L'unico valore valido è `https`.
hostname	Stringa	Sì	Il nome host del pattern remoto consentito. È possibile specifica re un valore letterale o un carattere jolly. Un singolo `*` corrispon de a un singolo sottodominio. Un `**` doppio corrisponde a

Chiave	Tipo	Campo obbligatorio	Descrizione
			un numero qualsiasi di sottodomini. Amplify non consente caratteri jolly generici in cui è specificato solo `**`.
port	Stringa	No	La porta del pattern remoto consentito.
percorso	Stringa	No	Il nome del percorso del pattern remoto consentito.

L'esempio seguente dimostra l'`imageSettings`attributo.

```
"imageSettings": {
    "sizes": [
      100,
      200
    ],
    "domains": [
      "example.com"
    ],
    "remotePatterns": [
      {
        "protocol": "https",
        "hostname": "example.com",
        "port": "",
        "pathname": "/**",
      }
    ],
    "formats": [
      "image/webp"
    ],
    "minumumCacheTTL": 60,
    "dangerouslyAllowSVG": false
  }
```

Utilizzo dell'attributo framework

Utilizzate l'`framework`attributo per specificare i metadati del framework.

La seguente definizione dell'oggetto illustra la configurazione dell'oggetto. `FrameworkMetadata`

```
type FrameworkMetadata = {
  name: string;
  version: string;
}
```

La tabella seguente descrive le proprietà dell'`FrameworkMetadata`oggetto.

Chiave	Tipo	Campo obbligatorio	Descrizione
nome	Stringa	Sì	Il nome del framework .
version	Stringa	Sì	La versione del framework. Deve essere una stringa di versioning semantico (semver) valida.

Le migliori pratiche per la configurazione delle regole di routing

Le regole di routing forniscono un meccanismo per instradare i percorsi delle richieste in entrata verso destinazioni specifiche del pacchetto di distribuzione. In un pacchetto di distribuzione, gli autori del framework possono inviare file nell'output di compilazione che vengono distribuiti a uno dei seguenti obiettivi:

- Risorse statiche primitive: i file sono contenuti nella directory. `.amplify-hosting/static`
- Primitiva di calcolo: i file sono contenuti nella directory. `.amplify-hosting/compute/default`

Gli autori del framework forniscono anche una serie di regole di routing nel file manifest di deploy. Ogni regola dell'array viene confrontata con la richiesta in entrata in ordine di attraversamento

sequenziale, finché non si verifica una corrispondenza. Quando esiste una regola di corrispondenza, la richiesta viene indirizzata alla destinazione specificata nella regola di corrispondenza. Facoltativamente, è possibile specificare un obiettivo di riserva per ogni regola. Se la destinazione originale restituisce un errore 404, la richiesta viene indirizzata alla destinazione di fallback.

Le specifiche di distribuzione richiedono che l'ultima regola nell'ordine di attraversamento sia una regola generale. Con il percorso viene specificata una regola generale. `/*` Se la richiesta in entrata non corrisponde a nessuna delle rotte precedenti nell'array delle regole di routing, la richiesta viene indirizzata al target della regola generale.

Per i framework SSR comeNuxt.js, l'obiettivo della regola generale deve essere la primitiva di calcolo. Questo perché le applicazioni SSR hanno pagine renderizzate lato server con percorsi che non sono prevedibili in fase di compilazione. Ad esempio, se un'Nuxt.jsapplicazione ha una pagina in `/blog/[slug]` cui si `[slug]` trova un parametro di percorso dinamico. L'obiettivo della regola generica è l'unico modo per indirizzare le richieste a queste pagine.

Al contrario, è possibile utilizzare schemi di percorso specifici per indirizzare percorsi noti in fase di compilazione. Ad esempio, Nuxt.js serve risorse statiche dal `/_nuxt` percorso. Ciò significa che il `/_nuxt/*` percorso può essere mirato da una regola di routing specifica che indirizza le richieste alla primitiva degli asset statici.

Routing delle cartelle pubbliche

La maggior parte dei framework SSR offre la possibilità di fornire risorse statiche mutabili da una cartella. `public` I file simili a `favicon.ico` e `robots.txt` sono in genere conservati all'interno della `public` cartella e vengono serviti dall'URL principale dell'applicazione. Ad esempio, il `favicon.ico` file viene fornito da`https://example.com/favicon.ico`. Nota che non esiste uno schema di percorso prevedibile per questi file. Sono quasi interamente dettati dal nome del file. L'unico modo per indirizzare i file all'interno della `public` cartella è utilizzare il percorso generico. Tuttavia, l'obiettivo generale della rotta deve essere la primitiva di calcolo.

Consigliamo uno dei seguenti approcci per la gestione della cartella. `public`

1. Utilizzate un modello di percorso per indirizzare i percorsi di richiesta che contengono estensioni di file. Ad esempio, puoi utilizzarlo `/*.*` per indirizzare tutti i percorsi di richiesta che contengono un'estensione di file.

   Nota che questo approccio può essere inaffidabile. Ad esempio, se all'interno della `public` cartella sono presenti file senza estensione, non vengono presi di mira da questa regola. Un altro

problema da tenere presente con questo approccio è che l'applicazione potrebbe avere pagine con punti nei nomi. Ad esempio, una pagina in `/blog/2021/01/01/hello.world` verrà scelta come target dalla `/*.*` regola. Questo non è l'ideale poiché la pagina non è una risorsa statica. Tuttavia, puoi aggiungere un obiettivo di fallback a questa regola per garantire che, quando si verifica un errore 404 dalla primitiva statica, la richiesta ritorni alla primitiva di calcolo.

```
{
    "path": "/*.*",
    "target": {
        "kind": "Static"
    },
    "fallback": {
        "kind": "Compute",
        "src": "default"
    }
}
```

2. Identifica i file nella `public` cartella in fase di compilazione ed emetti una regola di routing per ogni file. Questo approccio non è scalabile poiché esiste un limite di 25 regole imposto dalle specifiche di distribuzione.

```
{
    "path": "/favicon.ico",
    "target": {
        "kind": "Static"
    }
},
{
    "path": "/robots.txt",
    "target": {
        "kind": "Static"
    }
}
```

3. Consigliamo agli utenti del framework di archiviare tutte le risorse statiche mutabili all'interno di una sottocartella all'interno della cartella. `public`

   Nell'esempio seguente, l'utente può memorizzare tutte le risorse statiche mutabili all'interno della cartella. `public/assets` Quindi, è `/assets/*` possibile utilizzare una regola di routing con lo schema di percorso per indirizzare tutte le risorse statiche mutabili all'interno della cartella. `public/assets`

```
{
    "path": "/assets/*",
    "target": {
        "kind": "Static"
    }
}
```

4. Specificare un fallback statico per il percorso generico. Questo approccio presenta degli svantaggi descritti più dettagliatamente nella sezione successiva. Routing fallback generico

Routing fallback generico

Per i framework SSR, ad esempioNuxt.js, in cui viene specificata una route generica per la destinazione primitiva di calcolo, gli autori del framework potrebbero prendere in considerazione la possibilità di specificare un fallback statico per il percorso catch-all per risolvere il problema del routing delle cartelle. `public` Tuttavia, questo tipo di regola di routing interrompe le pagine 404 renderizzate sul lato server. Ad esempio, se l'utente finale visita una pagina che non esiste, l'applicazione esegue il rendering di una pagina 404 con un codice di stato 404. Tuttavia, se il percorso generico ha un fallback statico, la pagina 404 non viene renderizzata. La richiesta torna invece alla primitiva statica e finisce comunque con un codice di stato 404, ma la pagina 404 non viene renderizzata.

```
{
    "path": "/*",
    "target": {
        "kind": "Compute",
        "src": "default"
    },
    "fallback": {
        "kind": "Static"
    }
}
```

Routing del percorso di base

I framework che offrono la possibilità di modificare il percorso di base dell'applicazione dovrebbero anteporre il percorso di base agli asset statici all'interno della directory. `.amplify-hosting/static` Ad esempio, se il percorso di base è`/folder1/folder2`, lo sarà l'output della build

per una risorsa statica chiamata main.css. `.amplify-hosting/static/folder1/folder2/main.css`

Ciò significa che anche le regole di routing devono essere aggiornate per riflettere il percorso di base. Ad esempio, se il percorso di base è`/folder1/folder2`, la regola di routing per le risorse statiche nella `public` cartella sarà simile alla seguente.

```
{
    "path": "/folder1/folder2/*.*",
    "target": {
        "kind": "Static"
    }
}
```

Analogamente, anche le route lato server devono avere il percorso di base anteposto ad esse. Ad esempio, se il percorso base è`/folder1/folder2`, la regola di routing per il `/api` percorso sarà simile alla seguente.

```
{
    "path": "/folder1/folder2/api/*",
    "target": {
        "kind": "Compute",
        "src": "default"
    }
}
```

Tuttavia, il percorso base non deve essere anteposto al percorso generico. Ad esempio, se il percorso base è`/folder1/folder2`, il percorso generale rimarrà come segue.

```
{
    "path": "/*",
    "target": {
        "kind": "Compute",
        "src": "default"
    }
}
```

Esempi di percorsi Nuxt.js

Di seguito è riportato un `deploy-manifest.json` file di esempio per un'applicazione Nuxt che dimostra come specificare le regole di routing.

```
{
  "version": 1,
  "routes": [
    {
      "path": "/_nuxt/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/_nuxt/builds/meta/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/_nuxt/builds/*",
      "target": {
        "cacheControl": "public, max-age=1, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/_nuxt/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
```

```
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "entrypoint": "server.js",
      "runtime": "nodejs18.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
}
```

Di seguito è riportato un `deploy-manifest.json` file di esempio per Nuxt che dimostra come specificare le regole di routing, inclusi i percorsi di base.

```
{
  "version": 1,
  "routes": [
    {
      "path": "/base-path/_nuxt/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/base-path/_nuxt/builds/meta/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/_nuxt/builds/*",
      "target": {
        "cacheControl": "public, max-age=1, immutable",
```

```
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/_nuxt/*",
      "target": {
        "cacheControl": "public, max-age=31536000, immutable",
        "kind": "Static"
      }
    },
    {
      "path": "/base-path/*.*",
      "target": {
        "kind": "Static"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "entrypoint": "server.js",
      "runtime": "nodejs18.x"
    }
  ],
  "framework": {
    "name": "nuxt",
    "version": "3.8.1"
  }
}
```

Per ulteriori informazioni sull'utilizzo dell'`routes`attributo, vedere. Utilizzo dell'attributo routes

Distribuzione di un server Express utilizzando il manifesto di distribuzione

Questo esempio spiega come implementare un server Express di base utilizzando la specifica di distribuzione Amplify Hosting. È possibile sfruttare il manifesto di distribuzione fornito per specificare il routing, le risorse di calcolo e altre configurazioni.

Configura un server Express localmente prima di distribuirlo su Amplify Hosting

1. Crea una nuova directory per il tuo progetto e installa Express e Typescript.

```
mkdir express-app
cd express-app

# The following command will prompt you for information about your project
npm init

# Install express, typescript and types
npm install express --save
npm install typescript ts-node @types/node @types/express --save-dev
```

2. Aggiungi un `tsconfig.json` file alla radice del tuo progetto con i seguenti contenuti.

```
{
  "compilerOptions": {
    "target": "es6",
    "module": "commonjs",
    "outDir": "./dist",
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true
  },
  "include": ["src/**/*.ts"],
  "exclude": ["node_modules"]
}
```

3. Crea una directory denominata `src` nella radice del tuo progetto.
4. Crea un `index.ts` file nella `src` directory. Questo sarà il punto di accesso all'applicazione che avvia un server Express. Il server deve essere configurato per l'ascolto sulla porta 3000.

```
// src/index.ts
```

```
import express from 'express';

const app: express.Application = express();
const port = 3000;

app.use(express.text());

app.listen(port, () => {
  console.log(`server is listening on ${port}`);
});

// Homepage
app.get('/', (req: express.Request, res: express.Response) => {
  res.status(200).send("Hello World!");
});

// GET
app.get('/get', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-get-header", "get-header-value").send("get-response-
from-compute");
});

//POST
app.post('/post', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-post-header", "post-header-
value").send(req.body.toString());
});

//PUT
app.put('/put', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-put-header", "put-header-
value").send(req.body.toString());
});

//PATCH
app.patch('/patch', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-patch-header", "patch-header-
value").send(req.body.toString());
});

// Delete
app.delete('/delete', (req: express.Request, res: express.Response) => {
  res.status(200).header("x-delete-header", "delete-header-value").send();
```

```
});
```

5. Aggiungi i seguenti script al tuo `package.json` file.

```
"scripts": {
  "start": "ts-node src/index.ts",
  "build": "tsc",
  "serve": "node dist/index.js"
}
```

6. Crea una directory denominata `public` nella radice del tuo progetto. Quindi crea un file denominato `hello-world.txt` con i seguenti contenuti.

```
Hello world!
```

7. Aggiungi un `.gitignore` file alla radice del tuo progetto con i seguenti contenuti.

```
.amplify-hosting
dist
node_modules
```

Configurare il manifesto di distribuzione di Amplify

1. Crea un file denominato `deploy-manifest.json` nella directory principale del tuo progetto.
2. Copia e incolla il seguente manifesto nel tuo `deploy-manifest.json` file.

```
{
  "version": 1,
  "framework": { "name": "express", "version": "4.18.2" },
  "imageSettings": {
    "sizes": [
      100,
      200,
      1920
    ],
    "domains": [],
    "remotePatterns": [],
    "formats": [],
    "minimumCacheTTL": 60,
    "dangerouslyAllowSVG": false
  },
```

```
  "routes": [
    {
      "path": "/_amplify/image",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=3600, immutable"
      }
    },
    {
      "path": "/*.*",
      "target": {
        "kind": "Static",
        "cacheControl": "public, max-age=2"
      },
      "fallback": {
        "kind": "Compute",
        "src": "default"
      }
    },
    {
      "path": "/*",
      "target": {
        "kind": "Compute",
        "src": "default"
      }
    }
  ],
  "computeResources": [
    {
      "name": "default",
      "runtime": "nodejs18.x",
      "entrypoint": "index.js"
    }
  ]
}
```

Il manifesto descrive come Amplify Hosting dovrebbe gestire la distribuzione dell'applicazione. Le impostazioni principali sono le seguenti.

- versione: indica la versione della specifica di distribuzione che stai utilizzando.
- framework: modificalo per specificare la configurazione Express del server.

- ImageSettings: questa sezione è facoltativa per un Express server a meno che non si stia gestendo l'ottimizzazione delle immagini.
- percorsi: sono fondamentali per indirizzare il traffico verso le parti giuste dell'app. Il `"kind": "Compute"` percorso indirizza il traffico verso la logica del server.
- ComputeResources: utilizzate questa sezione per specificare il runtime e il punto di ingresso del Express server.

Successivamente, configura uno script di post-compilazione che sposti gli artefatti dell'applicazione integrata nel pacchetto di distribuzione. `.amplify-hosting` La struttura delle directory è in linea con le specifiche di distribuzione di Amplify Hosting.

Configura lo script di post-compilazione

1. Crea una directory denominata `bin` nella radice del tuo progetto.
2. Crea un file denominato `postbuild.sh` nella `bin` directory. Aggiungi i seguenti contenuti al file `postbuild.sh`.

```
#!/bin/bash

rm -rf ./.amplify-hosting

mkdir -p ./.amplify-hosting/compute

cp -r ./dist ./.amplify-hosting/compute/default
cp -r ./node_modules ./.amplify-hosting/compute/default/node_modules

cp -r public ./.amplify-hosting/static

cp deploy-manifest.json ./.amplify-hosting/deploy-manifest.json
```

3. Aggiungi uno `postbuild` script al tuo `package.json` file. Il file dovrebbe avere l'aspetto seguente.

```
"scripts": {
  "start": "ts-node src/index.ts",
  "build": "tsc",
  "serve": "node dist/index.js",
  "postbuild": "chmod +x bin/postbuild.sh && ./bin/postbuild.sh"
}
```

4. Esegui il comando seguente per creare la tua applicazione.

```
npm run build
```

5. (Facoltativo) Modifica i tuoi percorsi per Express. È possibile modificare i percorsi nel manifesto di distribuzione per adattarli al server Express. Ad esempio, se non hai risorse statiche nella `public` directory, potresti aver bisogno solo del percorso generico che `"path": "/*"` indirizza a Compute. Ciò dipenderà dalla configurazione del server.

La struttura finale delle cartelle dovrebbe essere simile alla seguente.

```
express-app/
### .amplify-hosting/
#   ### compute/
#   #   ### default/
#   #       ### node_modules/
#   #       ### index.js
#   ### static/
#   #   ### hello.txt
#   ### deploy-manifest.json
### bin/
#   ### .amplify-hosting/
#   #   ### compute/
#   #   #   ### default/
#   #   ### static/
#   ### postbuild.sh*
### dist/
#   ### index.js
### node_modules/
### public/
#   ### hello.txt
### src/
#   ### index.ts
### deploy-manifest.json
### package.json
### package-lock.json
### tsconfig.json
```

Implementa il tuo server

1. Invia il codice al tuo repository Git e poi distribuisci la tua app su Amplify Hosting.

2. Aggiorna le impostazioni di build in modo che punti a quanto segue`baseDirectory`. `.amplify-hosting` Durante la compilazione, Amplify rileverà il file manifest nella directory e distribuirà `.amplify-hosting` il server Express come configurato.

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        - nvm use 18
        - npm install
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: .amplify-hosting
    files:
      - '**/*'
```

3. Per verificare che la distribuzione sia avvenuta correttamente e che il server funzioni correttamente, visita l'app all'URL predefinito fornito da Amplify Hosting.

Integrazione dell'ottimizzazione delle immagini per gli autori del framework

Gli autori del framework possono integrare la funzionalità di ottimizzazione delle immagini di Amplify utilizzando le specifiche di implementazione di Amplify Hosting. Per abilitare l'ottimizzazione delle immagini, il manifesto di distribuzione deve contenere una regola di routing destinata al servizio di ottimizzazione delle immagini. L'esempio seguente mostra come configurare la regola di routing.

```
// .amplify-hosting/deploy-manifest.json

{
  "routes": [
    {
      "path": "/images/*",
      "target": {
        "kind": "ImageOptimization",
        "cacheControl": "public, max-age=31536000, immutable"
      }
    }
  ]
```

```
}
```

Per ulteriori informazioni sulla configurazione delle impostazioni di ottimizzazione delle immagini utilizzando le specifiche di distribuzione, vedere. Utilizzo della specifica di distribuzione di Amplify Hosting per configurare l'output della build

Comprendere l'API di ottimizzazione delle immagini

L'ottimizzazione delle immagini può essere richiamata in fase di esecuzione tramite l'URL di dominio di un'app Amplify, nel percorso definito dalla regola di routing.

```
GET https://{appDomainName}/{path}?{queryParams}
```

L'ottimizzazione delle immagini impone le seguenti regole sulle immagini.

- Amplify non può ottimizzare i formati GIF, APNG e SVG o convertirli in un altro formato.
- Le immagini SVG non vengono fornite a meno che l'impostazione non sia abilitata. `dangerouslyAllowSVG`
- La larghezza o l'altezza di un'immagine sorgente non può superare 11 MB o 9.000 pixel.
- Il limite di dimensione di un'immagine ottimizzata è di 4 MB.
- HTTPS è l'unico protocollo supportato per l'approvvigionamento di immagini con telecomando URLs.

Intestazioni HTTP

L'intestazione HTTP Accept request viene utilizzata per specificare i formati di immagine, espressi come tipi MIME, consentiti dal client (di solito un browser Web). Il servizio di ottimizzazione delle immagini tenterà di convertire l'immagine nel formato specificato. Il valore specificato per questa intestazione avrà una priorità maggiore rispetto al parametro di query di formato. Ad esempio, un valore valido per l'intestazione Accept è. `image/png, image/webp, */*` L'impostazione dei formati specificata nel manifesto di distribuzione di Amplify limiterà i formati a quelli presenti nell'elenco. Anche se l'intestazione Accept richiede un formato specifico, verrà ignorato se il formato non è nell'elenco dei formati consentiti.

Parametri della richiesta URI

La tabella seguente descrive i parametri di richiesta URI per l'ottimizzazione delle immagini.

Parametro di query	Tipo	Campo obbligatorio	Descrizione	Esempio
url	Stringa	Sì	Un percorso relativo o un URL assoluto dell'immagine sorgente. Per un URL remoto, è supportato solo il protocollo https. Il valore deve essere codificato in un URL.	`?url=https%3A%2F%2Fwww.example.com%2Fbuffalo.png`
width	Numero	Sì	La larghezza in pixel dell'immagine ottimizzata.	`?width=800`
height	Numero	No	L'altezza in pixel dell'immagine ottimizzata. Se non specificato, l'immagine verrà ridimensionata automaticamente in base alla larghezza.	`?height=600`

in forma	Valori Enum:`cover`,,`contain`
	`inside`
	`outside`
		No	Come viene ridimensionata l'immagine per adattarla alla larghezza e all'altezza specificate.	`?width=800&height=600&fit=cover`

Parametro di query	Tipo	Campo obbligatorio	Descrizione	Esempio
position	Valori Enum:`center`,,,`top` `right` `bottom` `left`	No	Una posizione da usare quando fit è `cover` o `contain`.	`?fit=contain&position=centre`
trim	Numero	No	Taglia i pixel da tutti i bordi che contengono valori simili al colore di sfondo specificato del pixel in alto a sinistra.	`?trim=50`
estendere	Oggetto	No	Aggiunge pixel ai bordi dell'immagine utilizzando il colore derivato dai pixel del bordo più vicini. Il formato è `{top}_{right}_{bottom}_{left}` dove ogni valore è il numero di pixel da aggiungere.	`?extend=10_0_5_0`

Parametro di query	Tipo	Campo obbligatorio	Descrizione	Esempio
estratto	Oggetto	No	Ritaglia l'immagine nel rettangolo specificato delimitato da alto, sinistra, larghezza e altezza. Il formato è {left} _ {top} _ {width} _ {right} dove ogni valore è il numero di pixel da ritagliare.	`?extract=10_0_5_0`
format	Stringa	No	Il formato di output desiderato per l'immagine ottimizzata.	`?format=webp`
quality	Numero	No	La qualità dell'immagine, da 1 a 100. Utilizzato solo per la conversione del formato dell'immagine.	`?quality=50`
rotate	Numero	No	Ruota l'immagine in base all'angolo specificato in numero di gradi.	`?rotate=45`

Parametro di query	Tipo	Campo obbligatorio	Descrizione	Esempio
capovolgere	Boolean	No	Riflette l'immagine verticalmente (dall'alto verso il basso) sull'asse x. Ciò si verifica sempre prima della rotazione, se presente.	`?flip`
fiasco	Boolean	No	Riflette l'immagine orizzontalmente (sinistra-destra) sull'asse y. Ciò si verifica sempre prima della rotazione, se presente.	`?flop`
affilare	Numero	No	La nitidezza migliora la definizione dei bordi dell'immagine. I valori validi sono compresi tra 0,000001 e 10.	`?sharpen=1`
median	Numero	No	Applica un filtro mediano. Questo rimuove il rumore o leviga i bordi di un'immagine.	`?sharpen=3`

Parametro di query	Tipo	Campo obbligatorio	Descrizione	Esempio
sfocato	Numero	No	Applica una sfocatura gaussiana del sigma specificato. I valori validi sono compresi tra 0,3 e 1.000.	`?blur=20`
gamma	Numero	No	Applica una correzione gamma per migliorare la luminosità percepita di un'immagine ridimensionata. Il valore deve essere compreso tra 1,0 e 3,0.	`?gamma=1`
negare	Boolean	No	Inverte i colori dell'immagine.	`?negate`
normalizzare	Boolean	No	Migliora il contrasto dell'immagine estendendone la luminanza per coprire un'intera gamma dinamica.	`?normalize`

Parametro di query	Tipo	Campo obbligatorio	Descrizione	Esempio
threshold	Numero	No	Sostituisce qualsiasi pixel dell'immagine con un pixel nero, se la sua intensità è inferiore alla soglia specificata. Oppure con un pixel bianco se è superiore alla soglia. I valori validi sono compresi tra 0 e 255.	`?threshold=155`
tinta	Stringa	No	Tinge l'immagine utilizzando l'RGB fornito preservando la luminanza dell'immagine.	`?tint=#7743CE`
in scala di grigi	Boolean	No	Trasforma l'immagine in scala di grigi (bianco e nero).	`?grayscale`

Codici di stato della risposta

L'elenco seguente descrive i codici di stato della risposta per l'ottimizzazione delle immagini.

Operazione riuscita: codice di stato HTTP 200

La richiesta è stata soddisfatta con successo.

BadRequest - Codice di stato HTTP 400

- Un parametro di query di input è stato specificato in modo errato.
- L'URL remoto non è elencato come consentito nell'`remotePatterns`impostazione.
- L'URL remoto non si risolve in un'immagine.
- La larghezza o l'altezza richieste non sono elencate come consentite nell'`sizes`impostazione.
- L'immagine richiesta è SVG ma l'`dangerouslyAllowSvg`impostazione è disabilitata.

Non trovato: codice di stato HTTP 404

L'immagine sorgente non è stata trovata.

Contenuto troppo grande: codice di stato HTTP 413

L'immagine sorgente o l'immagine ottimizzata superano la dimensione massima consentita in byte.

Informazioni sulla memorizzazione ottimizzata delle immagini nella cache

Amplify Hosting memorizza nella cache le immagini ottimizzate sulla nostra CDN in modo che le richieste successive alla stessa immagine, con gli stessi parametri di query, vengano servite dalla cache. Il Time to live (TTL) della cache è controllato dall'intestazione. `Cache-Control` L'elenco seguente descrive le opzioni per specificare l'intestazione. `Cache-Control`

- Utilizzo della `Cache-Control` chiave all'interno della regola di routing che mira all'ottimizzazione delle immagini.
- Utilizzo di intestazioni personalizzate definite nell'app Amplify.
- Per le immagini remote, l'`Cache-Control`intestazione restituita dall'immagine remota viene rispettata.

Quanto `minimumCacheTTL` specificato nelle impostazioni di ottimizzazione dell'immagine definisce il limite inferiore della Cache-Control max-age direttiva. Ad esempio, se l'URL di un'immagine remota risponde con un`Cache-Control s-max-age=10`, ma il valore di `minimumCacheTTL` è 60, viene utilizzato 60.

Utilizzo di adattatori open source per qualsiasi framework SSR

Puoi utilizzare qualsiasi adattatore di build del framework SSR creato per l'integrazione con Amplify Hosting. Ogni framework che offre un adattatore determina come l'adattatore è configurato e

connesso al processo di creazione. In genere, l'adattatore verrà installato come dipendenza di sviluppo di npm.

Dopo aver creato un'app con un framework, utilizza la documentazione del framework per scoprire come installare l'adattatore Amplify Hosting e configurarlo nel file di configurazione dell'applicazione.

Successivamente, crea un `amplify.yml` file nella directory principale del progetto. Nel `amplify.yml` file, impostalo nella directory `baseDirectory` di output di compilazione dell'applicazione. Il framework esegue l'adattatore durante il processo di compilazione per trasformare l'output nel pacchetto di distribuzione Amplify Hosting.

Il nome della directory di output della build può essere qualsiasi cosa, ma il nome del `.amplify-hosting` file ha un significato. Amplify cerca innanzitutto una directory definita come. `baseDirectory` Se esiste, Amplify cerca lì l'output della build. Se la directory non esiste, Amplify cerca l'output della build all'`.amplify-hosting`interno, anche se non è stato definito dal cliente.

Di seguito è riportato un esempio delle impostazioni di build per un'app. `baseDirectory`è impostato `.amplify-hosting` per indicare che l'output della build si trova nella `.amplify-hosting` cartella. Finché il contenuto della `.amplify-hosting` cartella corrisponde alle specifiche di distribuzione di Amplify Hosting, l'app verrà distribuita correttamente.

```
version: 1
frontend:
  preBuild:
    commands:
      - npm install
  build:
    commands:
      - npm run build
  artifacts:
    baseDirectory: .amplify-hosting
```

Dopo aver configurato l'app per utilizzare un adattatore framework, puoi distribuirla su Amplify Hosting. Per istruzioni dettagliate, consulta Implementazione di un'app SSR su Amplify

Distribuzione di un sito Web statico su Amplify da un bucket Amazon S3

Puoi utilizzare l'integrazione tra Amplify Hosting e Amazon S3 per ospitare contenuti statici di siti Web archiviati su S3 con pochi clic. L'implementazione su Amplify Hosting offre i seguenti vantaggi e funzionalità.

- Implementazione automatica sulla rete di distribuzione AWS dei contenuti (CDN) disponibile a livello globale basata su CloudFront
- Supporto HTTPS
- Collega facilmente il tuo sito Web a un dominio personalizzato utilizzando la console Amplify
- Porta i tuoi certificati SSL personalizzati
- Monitora il tuo sito web con log di accesso e metriche integrati CloudWatch
- Imposta la protezione tramite password per il tuo sito web
- Crea regole di reindirizzamento e riscrittura nella console Amplify

È possibile avviare il processo di distribuzione dalla console Amplify, AWS CLI il, o il. AWS SDKs Puoi eseguire la distribuzione su Amplify solo da un bucket Amazon S3 generico situato nel tuo account. Amplify non supporta più account S3 accesso al bucket.

Quando distribuisci un'applicazione da un bucket generico Amazon S3 ad Amplify Hosting AWS , i costi si basano sul modello di prezzo di Amplify. Per ulteriori informazioni, consulta AWS Amplify Prezzi.

> **Important**
>
> Amplify Hosting non è disponibile in tutti i paesi in cui è disponibile Regioni AWS Amazon S3. Per implementare un sito web statico su Hosting Amplify, il bucket Amazon S3 per uso generico contenente il sito web deve trovarsi in una Regione in cui è disponibile Amplify. Per l'elenco delle Regioni in cui è disponibile Amplify, consulta Endpoint Amplify in Riferimenti generali di Amazon Web Services.

Consulta i seguenti argomenti per scoprire come distribuire e aggiornare un sito Web statico da Amazon S3 a Amplify Hosting.

Argomenti

- Distribuzione di un sito Web statico da S3 utilizzo della console Amplify
- Creazione di una bucket policy da cui distribuire un sito Web statico S3 utilizzando il AWS SDKs
- Aggiornamento di un sito Web statico distribuito su Amplify da un S3 bucket
- Aggiornamento di un S3 distribuzione per utilizzare un bucket e un prefisso anziché un file.zip

Distribuzione di un sito Web statico da S3 utilizzo della console Amplify

Utilizza le seguenti istruzioni per distribuire un nuovo sito Web statico da un bucket generico Amazon S3 utilizzando la console Amplify.

Per distribuire un sito Web statico da un bucket generico Amazon S3 utilizzando la console Amplify

1. Accedi a AWS Management Console e apri la console Amplify all'indirizzo. https://console.aws.amazon.com/amplify/
2. Nella pagina Tutte le app, scegli Crea nuova app.
3. Nella pagina Inizia a creare con Amplify, scegli Deploy without Git.
4. Scegli Next (Successivo).
5. Nella pagina Avvia una distribuzione manuale, procedi come segue.
   a. Per Nome dell'app, inserisci il nome dell'app.
   b. Per Nome filiale, inserisci il nome del ramo da distribuire.
6. Per Metodo, scegli Amazon S3.
7. Per il S3 posizione degli oggetti da ospitare, scegliete Sfoglia. Seleziona il bucket generico Amazon S3 da utilizzare, quindi seleziona Scegli prefisso.
8. Scegliere Save and deploy (Salva e distribuisci).

Creazione di una bucket policy da cui distribuire un sito Web statico S3 utilizzando il AWS SDKs

Puoi utilizzare il AWS SDKs per distribuire un sito Web statico da Amazon S3 ad Amplify Hosting. Se distribuisci il tuo sito Web utilizzando un SDK, devi creare una tua policy bucket che conceda ad Amplify Hosting l'autorizzazione a recuperare gli oggetti nel tuo S3 secchio.

Per ulteriori informazioni sulla creazione di policy bucket, consulta Bucket policies per Amazon S3 nella Amazon Simple Storage Service User Guide.

L'esempio seguente di policy bucket concede ad Amplify Hosting le autorizzazioni per elencare i bucket e recuperare gli oggetti bucket per l'ID dell'applicazione Amplify e il ramo specificati. Account AWS

Per utilizzare questo esempio:

- Sostituiscilo *amzn-s3-demo-website-bucket/prefix* con il nome del bucket e del prefisso del tuo sito web.
- Sostituiscilo *111122223333* con il tuo ID. Account AWS
- Sostituisci *region-id* con Regione AWS quello in cui si trova l'applicazione Amplify, ad esempio. **us-east-1**
- Sostituisci *app_id* con l'ID dell'applicazione Amplify. Queste informazioni sono disponibili nella console Amplify.
- *branch_name*Sostituiscilo con il nome della filiale.

> **Note**
>
> Nella tua policy bucket, `aws:SourceArn` deve essere un ARN del ramo con codifica URL (codifica percentuale).

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "AllowAmplifyToListPrefix_appid_branch_prefix_",
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "amplify.amazonaws.com"
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::amzn-s3-demo-website-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn%3Aaws%3Aamplify%3Aregion-
id%3A111122223333%3Aapps%2Fapp_id%2Fbranches%2Fbranch_name",
                    "s3:prefix": ""
                }
            }
        },
        {
            "Sid": "AllowAmplifyToReadPrefix__appid_branch_prefix_",
            "Effect": "Allow",
            "Principal": {
                "Service": "amplify.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-website-bucket/prefix/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn%3Aaws%3Aamplify%3Aregion-
id%3A111122223333%3Aapps%2Fapp_id%2Fbranches%2Fbranch_name"
                }
            }
        },
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::amzn-s3-demo-website-bucket/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```

Aggiornamento di un sito Web statico distribuito su Amplify da un S3 bucket

Se aggiorni uno qualsiasi degli oggetti per un sito Web statico per scopi generici S3 bucket ospitato su Amplify, è necessario ridistribuire l'applicazione su Amplify Hosting per rendere effettive le modifiche. Amplify Hosting non rileva automaticamente le modifiche al S3 secchio. Ti consigliamo di utilizzare AWS Command Line Interface (CLI) per aggiornare il tuo sito Web.

Sincronizza gli aggiornamenti con S3

Dopo aver apportato modifiche ai file di progetto del tuo sito Web, utilizza il seguente comando s3 sync per sincronizzare le modifiche apportate alla directory di origine locale con il bucket Amazon S3 generico di destinazione. Per utilizzare questo esempio, *<source>* sostituiscilo con il nome della tua directory locale e *<target>* con il nome del tuo bucket Amazon S3.

```
aws s3 sync <source> <target>
```

Ridistribuisci il sito Web su Amplify Hosting

Utilizza il seguente comando amplify start-deployment per ridistribuire l'applicazione aggiornata in un bucket Amazon S3 su Amplify Hosting. Per usare questo esempio, *<app_id>* sostituiscilo con l'id della tua applicazione Amplify*<branch_name>*, con il nome della tua filiale e con il tuo *s3://amzn-s3-demo-website-bucket/prefix* S3 bucket e prefisso.

```
aws amplify start-deployment --app-id <app_id> --branch-name <branch_name> --source-
url s3://amzn-s3-demo-website-bucket/prefix --source-url-type BUCKET_PREFIX
```

Aggiornamento di un S3 distribuzione per utilizzare un bucket e un prefisso anziché un file.zip

Se disponi già di un sito Web statico esistente distribuito su Amplify Hosting da un file.zip in un bucket generico Amazon S3, puoi aggiornare la distribuzione dell'applicazione per utilizzare il nome e il prefisso del bucket che contengono gli oggetti da ospitare. Questo tipo di distribuzione elimina la necessità di caricare un file separato nel bucket che contiene il contenuto compresso dell'output della build.

Per migrare un sito Web statico da un file.zip al contenuto del bucket

1. Accedi a AWS Management Console e apri la console Amplify all'indirizzo. https://console.aws.amazon.com/amplify/
2. Nella pagina Tutte le app, scegli il nome dell'app distribuita manualmente di cui desideri migrare dall'utilizzo di un file.zip all'utilizzo diretto dei file dell'applicazione.
3. Nella pagina Panoramica dell'applicazione, scegli Distribuisci aggiornamenti.
4. Nella pagina Distribuisci aggiornamenti, per Metodo, scegli Amazon S3.
5. Per S3 posizione degli oggetti da ospitare, scegliete Sfoglia. Seleziona il bucket da usare, quindi seleziona Scegli prefisso.
6. Scegliere Save and deploy (Salva e distribuisci).

Distribuzione di un'applicazione su Amplify senza un repository Git

Le distribuzioni manuali ti consentono di pubblicare la tua app web con Amplify Hosting senza connettere un provider Git. Puoi trascinare e rilasciare una cartella compressa dal desktop e ospitare il tuo sito in pochi secondi. In alternativa, puoi fare riferimento agli asset in un bucket Amazon S3 o specificare un URL pubblico per la posizione in cui sono archiviati i tuoi file.

> **Note**
>
> Le distribuzioni manuali hanno un limite massimo di dimensione del file.zip di 5 GB a causa dei vincoli relativi alle operazioni di copia di Amazon S3. Se alcuni degli artefatti della build superano queste dimensioni, valuta la possibilità di suddividerli in archivi più piccoli o di utilizzare un metodo di distribuzione alternativo.

Per Amazon S3, puoi anche impostare AWS Lambda trigger per aggiornare il tuo sito ogni volta che vengono caricate nuove risorse. Per maggiori dettagli sulla configurazione di questo scenario, consulta il post del blog Distribuisci i file archiviati su Amazon S3, Dropbox o AWS Amplify il desktop sulla console.

Amplify Hosting non supporta le distribuzioni manuali per le app renderizzate lato server (SSR). Per ulteriori informazioni, consulta Implementazione di applicazioni renderizzate lato server con Amplify Hosting.

Distribuzioni manuali trascina e rilascia

Per distribuire manualmente un'app utilizzando il drag and drop

1. Accedi AWS Management Console e apri la console Amplify.
2. Nell'angolo in alto a destra, scegli Crea nuova app.
3. Nella pagina Inizia a creare con Amplify, scegli Deploy without Git. Quindi, seleziona Next (Successivo).
4. Nella pagina Avvia una distribuzione manuale, in Nome app, inserisci il nome della tua app.
5. Per Nome filiale, inserisci un nome significativo, ad esempio **`development`** o**`production`**.

6. Per Metodo, scegli Trascina e rilascia.
7. Trascina e rilascia una cartella dal desktop nella zona di rilascio o usa Scegli la cartella.zip per selezionare il file dal tuo computer. Il file che trascini o selezioni deve essere una cartella compressa contenente il contenuto dell'output della build.
8. Scegliere Save and deploy (Salva e distribuisci).

Distribuzione manuale di Amazon S3 o URL

> **Note**
>
> Se stai distribuendo un sito Web statico da S3, la procedura seguente richiede il caricamento di una cartella compressa con il contenuto dell'output della build sul S3 secchio. Ti consigliamo di implementare un sito Web statico direttamente da S3 utilizzando il nome e il prefisso del bucket. Per ulteriori informazioni su questo processo semplificato, vedere. Distribuzione di un sito Web statico su Amplify da un bucket Amazon S3

Per distribuire manualmente un'app da Amazon S3 o da un URL pubblico

1. Accedi AWS Management Console e apri la console Amplify.
2. Nell'angolo in alto a destra, scegli Crea nuova app.
3. Nella pagina Inizia a creare con Amplify, scegli Deploy without Git. Quindi, seleziona Next (Successivo).
4. Nella pagina Avvia una distribuzione manuale, in Nome app, inserisci il nome della tua app.
5. Per Nome filiale, inserisci un nome significativo, ad esempio **`development`** o**`production`**.
6. Per Metodo, scegli Amazon S3 o Any URL.
7. La procedura per caricare i file dipende dal metodo di caricamento.
   - Amazon S3
     a. Per S3 location of objects to host, scegli Sfoglia S3. Quindi, seleziona il nome del bucket Amazon S3 dall'elenco. Le liste di controllo degli accessi (ACLs) devono essere abilitate per il bucket selezionato. Per ulteriori informazioni, consulta Risoluzione dei problemi di accesso ai bucket Amazon S3 per le distribuzioni manuali.
     b. Seleziona il nome del file.zip da distribuire.
     c. Scegli Scegli il prefisso.

- Qualsiasi URL
  - Per Resource URL, inserisci l'URL del file.zip da distribuire.

8. Scegliere Save and deploy (Salva e distribuisci).

> **Note**
>
> Quando crei la cartella compressa, assicurati di comprimere il contenuto dell'output della build e non la cartella di livello superiore. Ad esempio, se l'output della build genera una cartella denominata «build» o «public», per prima cosa accedi a quella cartella, seleziona tutti i contenuti e comprimila da lì. Se non lo fai, vedrai un errore «Accesso negato» perché la directory principale del sito non verrà inizializzata correttamente.

Risoluzione dei problemi di accesso ai bucket Amazon S3 per le distribuzioni manuali

Quando crei un bucket Amazon S3, utilizzi l'impostazione Amazon S3 Object Ownership per controllare se le liste di controllo degli accessi (ACLs) sono abilitate o disabilitate per il bucket. Per distribuire manualmente un'app su Amplify da un bucket Amazon S3, è necessario abilitarla nel bucket ACLs .

Se ricevi un `AccessControlList` errore durante la distribuzione da un bucket Amazon S3, significa che il bucket è stato creato ACLs con l'opzione disabilitata e devi abilitarla nella console Amazon S3. Per istruzioni, consulta Setting Object Ownership su un bucket esistente nella Amazon Simple Storage Service User Guide.

Gestione della configurazione di build per un'applicazione Amplify

Puoi personalizzare le impostazioni e la configurazione di build per le tue implementazioni Amplify. Quando si distribuisce un'applicazione, Amplify rileva automaticamente il framework frontend e le impostazioni di build associate. Puoi personalizzare le impostazioni di build nelle specifiche di build dell'applicazione (buildspec) per aggiungere variabili di ambiente, eseguire comandi di build e specificare le dipendenze di compilazione.

L'immagine di build predefinita di Amplify include diversi pacchetti e dipendenze preinstallati, ma puoi anche utilizzare la funzionalità live degli aggiornamenti dei pacchetti per specificare una versione specifica o assicurarti che la versione più recente sia sempre installata. Se hai dipendenze specifiche che richiedono molto tempo per essere installate durante una build utilizzando il contenitore predefinito di Amplify, puoi creare la tua immagine di build personalizzata. Puoi anche personalizzare la dimensione dell'istanza di build per fornire alla distribuzione dell'applicazione le risorse di CPU, memoria e spazio su disco necessarie.

Le build vengono avviate automaticamente con ogni commit nel tuo repository Git e con ogni nuova implementazione. Puoi configurare la funzionalità dei webhook in entrata per avviare una build senza un commit nel tuo repository Git.

La funzionalità di notifica di compilazione consente di condividere informazioni con i membri del team sui successi e sugli errori delle build.

Argomenti

- Configurazione delle impostazioni di build per un'applicazione Amplify
- Personalizzazione dell'immagine di costruzione
- Configurazione dell'istanza di build per un'applicazione Amplify
- Creazione di un webhook in entrata per avviare una build
- Impostazione della notifica e-mail per le compilazioni

Configurazione delle impostazioni di build per un'applicazione Amplify

Quando distribuisci un'applicazione, Amplify rileva automaticamente il framework frontend e le impostazioni di build associate ispezionando il file dell'app nel tuo repository Git. `package.json` Per memorizzare le impostazioni di compilazione dell'app sono disponibili le seguenti opzioni:

- Salva le impostazioni di build nella console Amplify - La console Amplify rileva automaticamente le impostazioni di build e le salva in modo che siano accessibili dalla console Amplify. Amplify applica queste impostazioni a tutte le tue filiali a meno che non ci sia `amplify.yml` un file memorizzato nel tuo repository.
- Salva le impostazioni di build nel tuo repository: scarica il `amplify.yml` file e aggiungilo alla radice del tuo repository.

> **Note**
>
> Le impostazioni di build sono visibili nel menu Hosting della console Amplify solo quando un'app è configurata per la distribuzione continua e connessa a un repository git. Per istruzioni su questo tipo di distribuzione, consulta Guida introduttiva.

Riferimento alle specifiche della build

La specifica di build (buildspec) per un'applicazione Amplify è una raccolta di impostazioni YAML e comandi di build che Amplify utilizza per eseguire la build. L'elenco seguente descrive queste impostazioni e come vengono utilizzate.

version

Il numero di versione YAML di Amplify.

ApRoot

Il percorso all'interno del repository in cui risiede questa applicazione. Ignorato a meno che non vengano definite più applicazioni.

env

Aggiungi variabili di ambiente a questa sezione. Si possono aggiungere variabili d'ambiente anche dalla console.

backend

Esegui i comandi Amplify CLI per fornire un backend, aggiornare le funzioni Lambda o gli schemi GraphQL come parte della distribuzione continua.

frontend

Esegui i comandi di compilazione del frontend.

test

Esegui comandi durante una fase di test. Scopri come aggiungere test alla tua app.

fasi di compilazione

Il frontend, il backend e il test hanno tre fasi che rappresentano i comandi eseguiti durante ogni sequenza della build.

- PreBuild - Lo script PreBuild viene eseguito prima dell'inizio della compilazione effettiva, ma dopo che Amplify installa le dipendenze.
- build: i comandi di compilazione.
- PostBuild - Lo script post-build viene eseguito al termine della compilazione e Amplify ha copiato tutti gli artefatti necessari nella directory di output.

buildpath

Il percorso da utilizzare per eseguire la build. Amplify utilizza questo percorso per localizzare gli artefatti della tua build. Se non specifichi un percorso, Amplify utilizza la root dell'app monorepo, ad esempio. `apps/app`

artifacts>base-directory

La directory in cui esistono gli artefatti della compilazione.

artefatti > file

Specificate i file degli artefatti che desiderate distribuire. Inserisci `**/*` per includere tutti i file.

cache

Specifica le dipendenze in fase di compilazione, ad esempio la cartella node_modules. Durante la prima build, i percorsi forniti qui vengono memorizzati nella cache. Nelle build successive, Amplify ripristina la cache sugli stessi percorsi prima di eseguire i comandi.

Amplify considera tutti i percorsi di cache forniti come relativi alla radice del progetto. Tuttavia, Amplify non consente l'attraversamento al di fuori della radice del progetto. Ad esempio, se specifichi un percorso assoluto, la compilazione avrà esito positivo senza errori, ma il percorso non verrà memorizzato nella cache.

Riferimento alla sintassi YAML delle specifiche di build

Il seguente esempio di specifica di build dimostra la sintassi YAML di base.

```
version: 1
env:
  variables:
    key: value
backend:
  phases:
    preBuild:
      commands:
        - *enter command*
    build:
      commands:
        - *enter command*
    postBuild:
        commands:
        - *enter command*
frontend:
  buildpath:
  phases:
    preBuild:
      commands:
        - cd react-app
        - npm ci
    build:
      commands:
        - npm run build
  artifacts:
    files:
```

```
        - location
        - location
    discard-paths: yes
    baseDirectory: location
  cache:
    paths:
        - path # A cache path relative to the project root
        - path # Traversing outside of the project root is not allowed
test:
  phases:
    preTest:
      commands:
        - *enter command*
    test:
      commands:
        - *enter command*
    postTest:
      commands:
        - *enter command*
  artifacts:
    files:
        - location
        - location
    configFilePath: *location*
    baseDirectory: *location*
```

Modifica delle specifiche di build

Puoi personalizzare le impostazioni di build di un'applicazione modificando le specifiche di build (buildspec) nella console Amplify. Le impostazioni di build vengono applicate a tutti i rami dell'app, ad eccezione dei rami che hanno un `amplify.yml` file salvato nel repository Git.

Per modificare le impostazioni di build nella console Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegliere l'app per la quale modificare le impostazioni di compilazione.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Crea impostazioni.
4. Nella pagina delle impostazioni di build, nella sezione Specifiche di build dell'app, scegli Modifica.
5. Nella finestra Modifica le specifiche di build, inserisci gli aggiornamenti.

6. Scegli Save (Salva).

Puoi utilizzare gli esempi descritti nei seguenti argomenti per aggiornare le impostazioni di build per scenari specifici.

Argomenti

- Impostazione delle impostazioni di build specifiche del ramo con script
- Impostazione di un comando per accedere a una sottocartella
- Implementazione del backend con il front-end per un'app di prima generazione
- Impostazione della cartella di output
- Installazione di pacchetti come parte di una build
- Utilizzando un registro npm privato
- Installazione di pacchetti del sistema operativo
- Impostazione dell'archiviazione chiave-valore per ogni build
- Saltare la build per un commit
- Disattivazione delle build automatiche su ogni commit
- Configurazione della build e della distribuzione del frontend basato su diff
- Configurazione di build di backend basate su diff per un'app di prima generazione

Impostazione delle impostazioni di build specifiche del ramo con script

Puoi utilizzare lo scripting della shell Bash per specificare le impostazioni di compilazione specifiche per il ramo. Ad esempio, lo script seguente utilizza la variabile di ambiente di sistema $ AWS_BRANCH per eseguire un set di comandi se il nome del ramo è main e un set diverso di comandi se il nome del ramo è dev.

```
frontend:
  phases:
    build:
      commands:
        - if [ "${AWS_BRANCH}" = "main" ]; then echo "main branch"; fi
        - if [ "${AWS_BRANCH}" = "dev" ]; then echo "dev branch"; fi
```

Impostazione di un comando per accedere a una sottocartella

Per monorepos, gli utenti vogliono poter `cd` accedere a una cartella per eseguire la build. Dopo aver eseguito il `cd` comando, questo si applica a tutte le fasi della build, quindi non è necessario ripetere il comando in fasi separate.

```
version: 1
env:
  variables:
    key: value
frontend:
  phases:
    preBuild:
      commands:
        - cd react-app
        - npm ci
    build:
      commands:
        - npm run build
```

Implementazione del backend con il front-end per un'app di prima generazione

> **Note**
>
> Questa sezione si applica solo alle applicazioni Amplify Gen 1. Un backend di prima generazione viene creato utilizzando Amplify Studio e l'interfaccia a riga di comando (CLI) Amplify.

Il `amplifyPush` comando è uno script di supporto che ti aiuta con le implementazioni di backend. Le impostazioni di compilazione riportate di seguito determinano automaticamente l'ambiente back-end corretto da distribuire per il ramo corrente.

```
version: 1
env:
  variables:
    key: value
backend:
  phases:
    build:
      commands:
```

```
        - amplifyPush --simple
```

Impostazione della cartella di output

Le seguenti impostazioni di compilazione impostano la directory di output per la cartella pubblica.

```
frontend:
  phases:
    commands:
      build:
        - yarn run build
  artifacts:
    baseDirectory: public
```

Installazione di pacchetti come parte di una build

È possibile utilizzare i `yarn` comandi `npm` o per installare i pacchetti durante la compilazione.

```
frontend:
  phases:
    build:
      commands:
        - npm install -g <package>
        - <package> deploy
        - yarn run build
  artifacts:
    baseDirectory: public
```

Utilizzando un registro npm privato

Si possono aggiungere riferimenti a un registro privato nelle impostazioni di compilazione oppure come variabile d'ambiente.

```
build:
  phases:
    preBuild:
      commands:
        - npm config set <key> <value>
        - npm config set registry https://registry.npmjs.org
        - npm config set always-auth true
        - npm config set email hello@amplifyapp.com
        - yarn install
```

Installazione di pacchetti del sistema operativo

L'immagine AL2 023 di Amplify esegue il codice con un utente non privilegiato denominato. `amplify` Amplify concede a questo utente i privilegi per eseguire i comandi del sistema operativo utilizzando il comando Linux. `sudo` Se desideri installare pacchetti del sistema operativo per le dipendenze mancanti, puoi usare comandi come e with. `yum` `rpm` `sudo`

La sezione build di esempio seguente mostra la sintassi per l'installazione di un pacchetto del sistema operativo utilizzando il comando. `sudo`

```
build:
  phases:
    preBuild:
      commands:
        - sudo yum install -y <package>
```

Impostazione dell'archiviazione chiave-valore per ogni build

`envCache`Fornisce l'archiviazione dei valori chiave in fase di compilazione. I valori memorizzati in `envCache` possono essere modificati solo durante una build e possono essere riutilizzati nella build successiva. Utilizzando`envCache`, possiamo archiviare informazioni sull'ambiente distribuito e renderle disponibili al contenitore di build nelle build successive. A differenza dei valori memorizzati in`envCache`, le modifiche alle variabili di ambiente durante una build non vengono mantenute nelle build future.

Esempio di utilizzo:

```
envCache --set <key> <value>
envCache --get <key>
```

Saltare la build per un commit

Per saltare una compilazione automatica su un particolare commit, includi il testo [skip-cd] alla fine del messaggio di commit.

Disattivazione delle build automatiche su ogni commit

Puoi configurare Amplify per disattivare le build automatiche su ogni commit di codice. Per effettuare la configurazione, scegli Impostazioni app, Impostazioni Branch, quindi individua la sezione

Branches che elenca i rami collegati. Seleziona un ramo, quindi scegli Azioni, Disabilita la creazione automatica. I nuovi commit verso quel ramo non daranno più inizio a una nuova build.

Configurazione della build e della distribuzione del frontend basato su diff

Puoi configurare Amplify per utilizzare build di frontend basate su diff. Se abilitato, all'inizio di ogni build Amplify tenta di eseguire un diff sulla `appRoot` tua cartella o `/src/` sulla cartella per impostazione predefinita. Se Amplify non rileva alcuna differenza, salta i passaggi di compilazione, test (se configurato) e distribuzione del frontend e non aggiorna l'app ospitata.

Per configurare un frontend basato su diff, compila e distribuisci

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui configurare la creazione e la distribuzione del frontend basato su diff.
3. Nel pannello di navigazione, scegli Hosting, Variabili di ambiente.
4. Nella sezione Variabili di ambiente, scegli Gestisci variabili.
5. La procedura per configurare la variabile di ambiente varia a seconda che stiate abilitando o disabilitando la creazione e la distribuzione del frontend basato su diff.
   - Per abilitare la creazione e la distribuzione di frontend basati su differenze
     a. Nella sezione Gestisci variabili, sotto Variabile, inserisci. `AMPLIFY_DIFF_DEPLOY`
     b. In Valore, specifica `true`.
   - Per disabilitare la creazione e la distribuzione del frontend basato su diff
     - Esegui una di queste operazioni:
       - Nella sezione Gestisci variabili, individua. `AMPLIFY_DIFF_DEPLOY` In Valore, specifica `false`.
       - Rimuovere la variabile di `AMPLIFY_DIFF_DEPLOY` ambiente.
6. Scegli Save (Salva).

Facoltativamente, puoi impostare la variabile di `AMPLIFY_DIFF_DEPLOY_ROOT` ambiente in modo che sovrascriva il percorso predefinito con un percorso relativo alla radice del repository, ad esempio. `dist`

Configurazione di build di backend basate su diff per un'app di prima generazione

> **Note**
>
> Questa sezione si applica solo alle applicazioni Amplify Gen 1. Un backend di prima generazione viene creato utilizzando Amplify Studio e l'interfaccia a riga di comando (CLI) Amplify.

Puoi configurare Amplify Hosting per utilizzare build di backend basate su diff utilizzando la variabile di ambiente. `AMPLIFY_DIFF_BACKEND` Quando abiliti le build di backend basate su diff, all'inizio di ogni build Amplify tenta di eseguire un diff nella cartella del tuo repository. `amplify` Se Amplify non rileva alcuna differenza, salta la fase di creazione del backend e non aggiorna le risorse del backend. Se il progetto non ha una `amplify` cartella nel repository, Amplify ignora il valore della variabile di ambiente. `AMPLIFY_DIFF_BACKEND`

Se al momento hai dei comandi personalizzati specificati nelle impostazioni di compilazione della fase di backend, le build condizionali di backend non funzioneranno. Se desideri che questi comandi personalizzati vengano eseguiti, devi spostarli nella fase di frontend delle impostazioni di build nel file dell'app. `amplify.yml`

Per configurare build di backend basate su diff

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui configurare le build di backend basate su diff.
3. Nel pannello di navigazione, scegli Hosting, Variabili di ambiente.
4. Nella sezione Variabili di ambiente, scegli Gestisci variabili.
5. La procedura per configurare la variabile di ambiente varia a seconda che si stiano abilitando o disabilitando le build di backend basate su diff.
   - Per abilitare le build di backend basate su diff
     a. Nella sezione Gestisci variabili, sotto Variabile, inserisci. `AMPLIFY_DIFF_BACKEND`
     b. In Valore, specifica `true`.
   - Per disabilitare le build di backend basate su diff
     - Esegui una di queste operazioni:

- Nella sezione Gestisci variabili, individua. `AMPLIFY_DIFF_BACKEND` In Valore, specifica `false`.
- Rimuovere la variabile di `AMPLIFY_DIFF_BACKEND` ambiente.

6. Scegli Save (Salva).

Configurazione delle impostazioni di build monorepo

Quando si archiviano più progetti o microservizi in un unico repository, si parla di monorepo. Puoi utilizzare Amplify Hosting per distribuire applicazioni in un monorepo senza creare più configurazioni di build o configurazioni di filiale.

Amplify supporta app in monorepo generici e app in monorepo create utilizzando npm workspace, pnpm workspace, Yarn workspace, Nx e Turborepo. Quando distribuisci la tua app, Amplify rileva automaticamente lo strumento di creazione monorepo che stai utilizzando. Amplify applica automaticamente le impostazioni di build per le app in un'area di lavoro npm, un'area di lavoro Yarn o Nx. Le app Turborepo e pnpm richiedono una configurazione aggiuntiva. Per ulteriori informazioni, consulta Configurazione delle app Turborepo e pnpm monorepo.

Puoi salvare le impostazioni di build per un monorepo nella console Amplify oppure puoi scaricare il `amplify.yml` file e aggiungerlo alla radice del tuo repository. Amplify applica le impostazioni salvate nella console a tutte le tue filiali a meno che non trovi `amplify.yml` un file nel tuo repository. Quando un `amplify.yml` file è presente, le sue impostazioni sostituiscono tutte le impostazioni di build salvate nella console Amplify.

Riferimento alla sintassi YAML delle specifiche di build di Monorepo

La sintassi YAML per una specifica di build monorepo è diversa dalla sintassi YAML per un repository che contiene una singola applicazione. Per un monorepo, si dichiara ogni progetto in un elenco di applicazioni. È necessario fornire la seguente `appRoot` chiave aggiuntiva per ogni applicazione dichiarata nelle specifiche di build del monorepo:

AppRoot

La radice, all'interno del repository, da cui viene avviata l'applicazione. Questa chiave deve esistere e avere lo stesso valore della variabile di `AMPLIFY_MONOREPO_APP_ROOT` ambiente. Per istruzioni sull'impostazione di questa variabile di ambiente, vedereImpostazione della variabile di ambiente AMPLIFY_MONOREPO_APP_ROOT.

Il seguente esempio di specifica di build monorepo dimostra come dichiarare più applicazioni Amplify nello stesso repository. Le due app, e sono dichiarate nell'`react-app`elenco. `angular-app` `applications` La `appRoot` chiave di ogni app indica che l'app si trova nella cartella `apps` principale del repository.

L'`buildpath`attributo è impostato per `/` eseguire e creare l'app dalla radice del progetto monorepo. L'`baseDirectory`attributo è il percorso relativo di. `buildpath`

Sintassi YAML delle specifiche di build di Monorepo

```
version: 1
applications:
  - appRoot: apps/react-app
    env:
      variables:
        key: value
    backend:
      phases:
        preBuild:
          commands:
            - *enter command*
        build:
          commands:
            - *enter command*
        postBuild:
            commands:
            - *enter command*
    frontend:
      buildPath: / # Run install and build from the monorepo project root
      phases:
        preBuild:
          commands:
            - *enter command*
            - *enter command*
        build:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        discard-paths: yes
        baseDirectory: location
```

```
      cache:
        paths:
            - path
            - path
    test:
      phases:
        preTest:
          commands:
            - *enter command*
        test:
          commands:
            - *enter command*
        postTest:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        configFilePath: *location*
        baseDirectory: *location*
  - appRoot: apps/angular-app
    env:
      variables:
        key: value
    backend:
      phases:
        preBuild:
          commands:
            - *enter command*
        build:
          commands:
            - *enter command*
        postBuild:
            commands:
            - *enter command*
    frontend:
      phases:
        preBuild:
          commands:
            - *enter command*
            - *enter command*
        build:
          commands:
```

```
            - *enter command*
      artifacts:
        files:
            - location
            - location
        discard-paths: yes
        baseDirectory: location
      cache:
        paths:
            - path
            - path
    test:
      phases:
        preTest:
          commands:
            - *enter command*
        test:
          commands:
            - *enter command*
        postTest:
          commands:
            - *enter command*
      artifacts:
        files:
            - location
            - location
        configFilePath: *location*
        baseDirectory: *location*
```

Un'app che utilizza la seguente specifica di build di esempio verrà creata nella radice del progetto e gli elementi di compilazione si troveranno in. `/packages/nextjs-app/.next`

```
applications:
  - frontend:
      buildPath: '/'  # run install and build from monorepo project root
      phases:
        preBuild:
          commands:
            - npm install
        build:
          commands:
            - npm run build --workspace=nextjs-app
```

```
      artifacts:
        baseDirectory: packages/nextjs-app/.next
        files:
          - '**/*'
      cache:
        paths:
          - node_modules/**/*
    appRoot: packages/nextjs-app
```

Impostazione della variabile di ambiente AMPLIFY_MONOREPO_APP_ROOT

Quando distribuisci un'app archiviata in un monorepo, la variabile di `AMPLIFY_MONOREPO_APP_ROOT` ambiente dell'app deve avere lo stesso valore del percorso della radice dell'app, rispetto alla radice del tuo repository. Ad esempio, un monorepo denominato `ExampleMonorepo` con una cartella principale denominata`apps`, che contiene, `app1`app2, e `app3` ha la seguente struttura di directory:

```
ExampleMonorepo
  apps
    app1
    app2
    app3
```

In questo esempio, il valore della variabile di `AMPLIFY_MONOREPO_APP_ROOT` ambiente for `app1` è. `apps/app1`

Quando distribuisci un'app monorepo utilizzando la console Amplify, la console imposta automaticamente la variabile di `AMPLIFY_MONOREPO_APP_ROOT` ambiente utilizzando il valore specificato per il percorso alla radice dell'app. Tuttavia, se l'app monorepo esiste già in Amplify o viene distribuita utilizzando AWS CloudFormation, è necessario impostare manualmente la variabile di ambiente nella sezione Variabili di `AMPLIFY_MONOREPO_APP_ROOT` ambiente della console Amplify.

Impostazione automatica della variabile di ambiente AMPLIFY_MONOREPO_APP_ROOT durante la distribuzione

Le seguenti istruzioni mostrano come implementare un'app monorepo con la console Amplify. Amplify imposta automaticamente `AMPLIFY_MONOREPO_APP_ROOT` la variabile di ambiente utilizzando la cartella principale dell'app specificata nella console.

Per distribuire un'app monorepo con la console Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli Crea nuova app nell'angolo in alto a destra.
3. Nella pagina Inizia a creare con Amplify, scegli il tuo provider Git, quindi scegli Avanti.
4. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Scegliere il nome del repository dall'elenco.
   b. Scegliere il nome del ramo da utilizzare.
   c. Seleziona La mia app è un monorepo
   d. Inserisci il percorso della tua app nel tuo monorepo, ad esempio,. **`apps/app1`**
   e. Scegli Next (Successivo).
5. Nella pagina delle impostazioni dell'app, puoi utilizzare le impostazioni predefinite o personalizzare le impostazioni di build per la tua app. Nella sezione Variabili d'ambiente, Amplify `AMPLIFY_MONOREPO_APP_ROOT` imposta il percorso specificato nel passaggio 4d.
6. Scegli Next (Successivo).
7. Nella pagina Revisione, scegli Salva e distribuisci.

Impostazione della variabile di ambiente AMPLIFY_MONOREPO_APP_ROOT per un'app esistente

Utilizza le seguenti istruzioni per impostare manualmente la variabile di `AMPLIFY_MONOREPO_APP_ROOT` ambiente per un'app che è già distribuita su Amplify o che è stata creata utilizzando. CloudFormation

Per impostare la variabile di ambiente AMPLIFY_MONOREPO_APP_ROOT per un'app esistente

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli il nome dell'app per cui impostare la variabile di ambiente.
3. Nel riquadro di navigazione, scegli Hosting, quindi scegli Variabili di ambiente.
4. Nella pagina Variabili di ambiente, scegli Gestisci variabili.
5. Nella sezione Gestisci variabili, procedi come segue:
   a. Seleziona Add new (Aggiungi nuovo).
   b. Per Variabile, inserisci la chiave`AMPLIFY_MONOREPO_APP_ROOT`.
   c. Per Value, inserisci il percorso dell'app, ad esempio**`apps/app1`**.

d. Per Branch, per impostazione predefinita, Amplify applica la variabile di ambiente a tutti i rami.

6. Scegli Save (Salva).

Configurazione delle app Turborepo e pnpm monorepo

Gli strumenti di compilazione Turborepo e pnpm workspace monorepo ottengono informazioni di configurazione dai file. `.npmrc` Quando distribuisci un'app monorepo creata con uno di questi strumenti, devi avere un file nella directory principale del progetto. `.npmrc`

Nel `.npmrc` file, imposta il linker per l'installazione dei pacchetti Node su. `hoisted` Puoi copiare la seguente riga nel file.

```
node-linker=hoisted
```

Per ulteriori informazioni su *`.npmrc`* file e impostazioni, consulta pnpm .npmrc nella documentazione di pnpm.

Pnpm non è incluso nel contenitore di build predefinito di Amplify. Per le app pnpm workspace e Turborepo, devi aggiungere un comando per installare pnpm nella fase delle impostazioni di compilazione dell'app. `preBuild`

L'esempio seguente, estratto da una specifica di build, mostra una fase con un comando per installare pnpm. `preBuild`

```
version: 1
applications:
  - frontend:
      phases:
        preBuild:
          commands:
            - npm install -g pnpm
```

Personalizzazione dell'immagine di costruzione

Puoi utilizzare un'immagine di build personalizzata per fornire un ambiente di compilazione personalizzato per un'app Amplify. Se hai dipendenze specifiche che richiedono molto tempo per essere installate durante una build utilizzando il contenitore predefinito di Amplify, puoi creare la

tua immagine Docker e farvi riferimento durante una build. Le immagini possono essere ospitate su Amazon Elastic Container Registry Public.

Affinché un'immagine di build personalizzata funzioni come immagine di build Amplify, deve soddisfare i seguenti requisiti.

Requisiti relativi all'immagine di costruzione personalizzata

1. Una distribuzione Linux che supporta la GNU C Library (glibc), come Amazon Linux, compilata per l'architettura x86-64.
2. cURL: quando avviamo la tua immagine personalizzata, scarichiamo il nostro strumento di esecuzione della compilazione nel tuo container, pertanto è necessario che il cURL sia presente. Se manca questa dipendenza, la compilazione fallisce immediatamente senza alcun output poiché il nostro build-runner non è in grado di produrre alcun output.
3. Git: per clonare il tuo repository Git, è necessario che Git sia installato nell'immagine. Se manca questa dipendenza, la fase di clonazione del repository avrà esito negativo.
4. OpenSSH: per clonare in modo sicuro il tuo repository, richiediamo a OpenSSH di configurare temporaneamente la chiave SSH durante la compilazione. Il pacchetto OpenSSH fornisce i comandi necessari al build runner per eseguire questa operazione.
5. Bash e The Bourne Shell: queste due utilità vengono utilizzate per eseguire comandi in fase di compilazione. Se non sono installate, le tue build potrebbero fallire prima di iniziare.
6. Node.js+npm: il nostro build runner non installa Node. Invece, si basa sull'installazione di Node e NPM nell'immagine. Questa condizione è necessaria per le compilazioni che richiedono pacchetti NPM o comandi specifici di Node. Tuttavia, consigliamo vivamente di installarli perché, quando sono presenti, il build runner Amplify può utilizzare questi strumenti per migliorare l'esecuzione della build. La funzione di sovrascrittura dei pacchetti di Amplify utilizza NPM per installare il pacchetto Hugo-extended quando si imposta un override per Hugo.

I seguenti pacchetti non sono richiesti, ma è consigliabile.

1. NVM (Node Version Manager): Ti consigliamo di installare questo gestore di versioni se devi gestire diverse versioni diNode. Quando imposti un override, la funzionalità di sostituzione dei pacchetti di Amplify utilizza la funzionalità di sostituzione dei pacchetti NVM per modificare le versioni di Node.js prima di ogni build.
2. Wget: Amplify può utilizzare l'utilità per scaricare file durante Wget il processo di compilazione. Ti consigliamo di installarlo nella tua immagine personalizzata.

3. Tar: Amplify può utilizzare l'utilità per decomprimere i file scaricati durante Tar il processo di compilazione. Ti consigliamo di installarlo nella tua immagine personalizzata.

Configurazione di un'immagine di build personalizzata per un'app

Utilizzare la seguente procedura per configurare un'immagine personalizzata per un'applicazione nella console Amplify.

Per configurare un'immagine di build personalizzata ospitata in Amazon ECR

1. Consulta la Guida introduttiva alla guida per utenti pubblici di Amazon ECR per configurare un repository Amazon ECR Public con un'immagine Docker.
2. Accedi AWS Management Console e apri la console Amplify.
3. Scegliere l'app per la quale si desidera configurare un'immagine personalizzata della build.
4. Nel pannello di navigazione, scegli Hosting, Crea impostazioni.
5. Nella pagina delle impostazioni di creazione, nella sezione Impostazioni dell'immagine di creazione, scegli Modifica.
6. Nella pagina Modifica le impostazioni dell'immagine di costruzione, espandi il menu Crea immagine e scegli Immagine di creazione personalizzata.
7. Inserisci il nome del repository Amazon ECR Public che hai creato nel primo passaggio. Qui è ospitata l'immagine della tua build. Ad esempio, se il nome del tuo repository è ecr-examplerepo, devi inserire. **`public.ecr.aws/xxxxxxxx/ecr-examplerepo`**
8. Scegli Save (Salva).

Utilizzo di versioni specifiche di pacchetti e dipendenze nell'immagine di build

Gli aggiornamenti live dei pacchetti consentono di specificare le versioni dei pacchetti e delle dipendenze da utilizzare nell'immagine di build predefinita di Amplify. L'immagine di build predefinita include diversi pacchetti e dipendenze preinstallati (ad esempio Hugo, Amplify CLI, Yarn, ecc.). Con gli aggiornamenti live dei pacchetti puoi sovrascrivere la versione di queste dipendenze e specificare una versione specifica o assicurarti che sia sempre installata la versione più recente.

Se gli aggiornamenti live dei pacchetti sono abilitati, prima dell'esecuzione della build, il build runner aggiorna (o esegue il downgrade) delle dipendenze specificate. Ciò aumenta il tempo di compilazione

proporzionalmente al tempo necessario per aggiornare le dipendenze, ma il vantaggio è che puoi assicurarti che venga utilizzata la stessa versione di una dipendenza per creare la tua app.

> **Warning**
>
> L'impostazione della versione di Node.js sulla versione più recente causa il fallimento delle build. È invece necessario specificare una versione esatta di Node.js, ad esempio `1821.5`, ov`0.1.2`.

Per configurare gli aggiornamenti dei pacchetti in tempo reale

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegliere l'app per la quale si desidera configurare gli aggiornamenti dei pacchetti live.
3. Nel pannello di navigazione, scegli Hosting, Crea impostazioni.
4. Nella pagina delle impostazioni di creazione, nella sezione Impostazioni dell'immagine di creazione, scegli Modifica.
5. Nella pagina Modifica le impostazioni dell'immagine di costruzione, elenco Aggiornamenti dei pacchetti in tempo reale, scegli Aggiungi nuovo.
6. Per Package, seleziona la dipendenza da sostituire.
7. Per Versione, mantieni l'ultima versione predefinita o inserisci una versione specifica della dipendenza. Se utilizzi la versione più recente, la dipendenza verrà sempre aggiornata all'ultima versione disponibile.
8. Scegli Save (Salva).

Configurazione dell'istanza di build per un'applicazione Amplify

Amplify Hosting offre dimensioni di istanze di build configurabili che consentono di fornire all'istanza di build dell'applicazione le risorse di CPU, memoria e spazio su disco necessarie. Prima del rilascio di questa funzionalità, Amplify forniva una configurazione di istanze di build a dimensione fissa di 8 GiB di memoria e 4 v. CPUs

Amplify supporta tre tipi di istanze di buildStandard:Large, e. XLarge Se non specifichi un tipo di istanza, Amplify utilizza l'istanza predefinita. Standard È possibile configurare il tipo di istanza di compilazione per un'applicazione utilizzando la console Amplify, AWS CLI il, o il. SDKs

Il costo per ogni tipo di istanza di build viene calcolato per minuto di creazione. Per i dettagli sui prezzi, vedere Prezzi di AWS Amplify.

La tabella seguente descrive le specifiche di calcolo per ogni tipo di istanza build:

Tipo di istanza di compilazione	v CPUs	Memoria	Spazio su disco
Standard	4 v CPUs	8 GiB	128 GB
Large	8 v CPUs	16 GiB	128 GB
XLarge	36 g CPUs	72 GiB	256 GB

Argomenti

- Comprendere i tipi di istanze di build
- Configurazione del tipo di istanza di compilazione nella console Amplify
- Configurazione della memoria heap di un'applicazione per utilizzare tipi di istanze di grandi dimensioni

Comprendere i tipi di istanze di build

L'impostazione del tipo di istanza di compilazione è configurata a livello di applicazione e si estende a tutti i rami dell'applicazione. I seguenti dettagli chiave si applicano ai tipi di istanze di compilazione:

- Il tipo di istanza di build che configuri per un'applicazione si applica automaticamente ai branch creati automaticamente e alle anteprime delle pull request.
- La quota del servizio Concurrent jobs si applica a tutti i tipi di istanze di build presenti nel tuo. Account AWS Ad esempio, se il limite di Concurrent Job è cinque, puoi eseguire fino a un massimo di 5 build per tutti i tipi di istanze del tuo. Account AWS
- Il costo per ogni tipo di istanza di build viene calcolato per minuto di creazione. Il processo di allocazione dell'istanza di compilazione può richiedere un sovraccarico aggiuntivo prima dell'inizio della compilazione. Soprattutto per le istanze più grandi XLarge, la build potrebbe subire una latenza prima dell'inizio della compilazione, a causa di questo sovraccarico. Tuttavia, ti viene addebitato solo il tempo di compilazione effettivo, non il tempo generale.

È possibile configurare il tipo di istanza build quando si crea una nuova applicazione oppure si può aggiornare il tipo di istanza su un'applicazione esistente. Per istruzioni sulla configurazione di questa impostazione nella console Amplify, vedere. Configurazione del tipo di istanza di compilazione nella console Amplify È inoltre possibile aggiornare questa impostazione utilizzando la SDKs. Per ulteriori informazioni, consulta la pagina CreateAppe UpdateApp APIs nell'Amplify API Reference.

Se nel tuo account sono presenti applicazioni che sono state create prima del rilascio della funzionalità di compilazione personalizzabile del tipo di istanza, utilizzano il tipo di Standard istanza predefinito. Quando aggiorni il tipo di istanza di build per un'applicazione esistente, tutte le build in coda o in corso prima dell'aggiornamento utilizzeranno il tipo di istanza di build precedentemente configurato. Ad esempio, se disponi di un'applicazione esistente con il **`main`** ramo distribuito su Amplify e ne aggiorni il tipo di istanza di build da Standard a Large, tutte le nuove build avviate dal ramo utilizzeranno **`main`** il tipo di istanza di build Large. Tuttavia, tutte le build in corso al momento dell'aggiornamento del tipo di istanza di build continueranno a essere eseguite sull'istanza Standard.

Configurazione del tipo di istanza di compilazione nella console Amplify

Utilizza la procedura seguente per configurare il tipo di istanza build quando si crea una nuova applicazione Amplify.

Per configurare il tipo di istanza build per una nuova applicazione

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella pagina Tutte le app, scegli Crea nuova app.
3. Nella pagina Inizia a creare con Amplify, scegli il tuo provider di repository Git, quindi scegli Avanti.
4. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Nell'elenco dei repository aggiornati di recente, seleziona il nome del repository da connettere.
   b. Nell'elenco Branch, seleziona il nome del ramo del repository da connettere.
   c. Scegli Next (Successivo).
5. Nella pagina delle impostazioni dell'app, apri la sezione Impostazioni avanzate.
6. Per il tipo di istanza Build, scegli il tipo di istanza desiderato dall'elenco.
7. Se state distribuendo un'applicazione basata sul runtime Node.js, configurate la dimensione della memoria heap per utilizzare efficacemente un tipo di istanza di grandi dimensioni. Puoi farlo nella pagina delle impostazioni dell'app impostando una variabile di ambiente o aggiornando le

impostazioni di build. Per ulteriori informazioni, consulta Configurazione della memoria heap di un'applicazione per utilizzare tipi di istanze di grandi dimensioni .

- Imposta una variabile di ambiente
  a. Nella sezione Impostazioni avanzate, Variabili di ambiente, scegli Aggiungi nuovo.
  b. Per Key enter**`NODE_OPTIONS`**.
  c. In Valore, specifica `--max-old-space-size=`*`memory_size_in_mb`*. Sostituisci *`memory_size_in_mb`* con la dimensione della memoria heap desiderata in megabyte.
- Aggiornamento delle impostazioni di build
  a. Nella sezione Impostazioni di compilazione, scegli Modifica file YML.
  b. Aggiungere il comando seguente alla `preBuild` fase. Sostituisci *`memory_size_in_mb`* con la dimensione della memoria heap desiderata in megabyte.

```
export NODE_OPTIONS='--max-old-space-size=memory_size_in_mb'
```

  c. Scegli Save (Salva).

8. Scegli Avanti.
9. Nella pagina Revisione, scegli Salva e distribuisci.

Utilizza la procedura seguente per configurare il tipo di istanza build per un'applicazione Amplify esistente.

Per configurare il tipo di istanza di compilazione per un'applicazione esistente

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per la quale desideri configurare il tipo di istanza build.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Crea impostazioni.
4. Nella pagina delle impostazioni di creazione, nella sezione Impostazioni avanzate, scegli Modifica.
5. Nella pagina Modifica impostazioni, per il tipo di istanza Build, scegli il tipo di istanza desiderato dall'elenco.
6. Scegli Save (Salva). Questa modifica avrà effetto a partire dalla successiva distribuzione dell'applicazione.

7. (Facoltativo) Per distribuire immediatamente l'applicazione aggiornata, esegui le seguenti operazioni:

   a. Nel riquadro di navigazione seleziona Overview (Panoramica).
   b. Nella pagina di panoramica dell'applicazione, scegli il ramo da ridistribuire.
   c. Nella pagina Distribuzione, scegli una distribuzione, ad esempio la distribuzione più recente. Quindi, scegli Ridistribuisci questa versione. Inizierà una nuova distribuzione.
   d. Al termine della distribuzione, le impostazioni di build dell'applicazione mostreranno che il ramo utilizza il tipo di istanza di build aggiornato.

Configurazione della memoria heap di un'applicazione per utilizzare tipi di istanze di grandi dimensioni

Se state creando applicazioni a uso intensivo di memoria, utilizzate questa sezione per capire come configurare l'applicazione per l'utilizzo di tipi di istanze di grandi dimensioni. I linguaggi e i framework di programmazione spesso si basano sull'allocazione di memoria dinamica, nota anche come memoria heap, durante il runtime per gestire i requisiti di memoria delle applicazioni. La memoria heap viene richiesta dall'ambiente di runtime e allocata dal sistema operativo host. Per impostazione predefinita, gli ambienti di runtime impongono un limite massimo di dimensione dell'heap disponibile per l'applicazione. Ciò significa che non sarà disponibile memoria aggiuntiva per l'applicazione oltre la dimensione dell'heap, anche se il sistema operativo o il contenitore host dispone di una maggiore quantità di memoria disponibile.

Ad esempio, l'ambiente di runtime JavaScript Node.js v8 impone un limite di dimensione dell'heap predefinito che dipende da diversi fattori, inclusa la dimensione della memoria dell'host. Di conseguenza, Standard le istanze di Large compilazione hanno una dimensione dell'heap Node.js predefinita di 2096 MB e l'XLargeistanza ha una dimensione dell'heap predefinita di 4144 MB. Pertanto, la creazione di un'applicazione con un requisito di memoria di 6000 MB utilizzando la dimensione dell'heap Node.js predefinita su qualsiasi tipo di istanza di build Amplify comporterà una compilazione non riuscita a causa di un errore. out-of-memory

Per aggirare i limiti di memoria della dimensione dell'heap predefiniti Node.js, è possibile procedere in uno dei seguenti modi:

- Imposta la variabile di `NODE_OPTIONS` ambiente nell'applicazione Amplify sul valore. `--max-old-space-size=`*`memory_size_in_mb`* Per`memory_size_in_mb`, specifica la dimensione della memoria heap desiderata in megabyte.

Per istruzioni, consultare Impostazione delle variabili di ambiente.

- Aggiungi il seguente comando alla `preBuild` fase delle specifiche di build dell'applicazione Amplify.

```
export NODE_OPTIONS='--max-old-space-size=memory_size_in_mb'
```

Puoi aggiornare le specifiche di build nella console Amplify o nel file `amplify.yml` dell'applicazione nel repository del progetto. Per istruzioni, consultare Configurazione delle impostazioni di build per un'applicazione Amplify.

L'esempio seguente: la specifica di build Amplify imposta una dimensione della memoria heap Node.js a 7000 MB per la creazione di un'applicazione frontend React:

```
version: 1
frontend:
  phases:
    preBuild:
      commands:
        # Set the heap size to 7000 MB
        - export NODE_OPTIONS='--max-old-space-size=7000'
        # To check the heap size memory limit in MB
        - node -e "console.log('Total available heap size (MB):',
 v8.getHeapStatistics().heap_size_limit / 1024 / 1024)"
        - npm ci --cache .npm --prefer-offline
    build:
      commands:
        - npm run build
  artifacts:
    baseDirectory: build
    files:
      - '**/*'
  cache:
    paths:
      - .npm/**/*
```

Per utilizzare efficacemente tipi di istanze di grandi dimensioni, è importante disporre di una dimensione di memoria heap sufficiente. La configurazione di un heap di piccole dimensioni per un'applicazione che utilizza molta memoria potrebbe causare un errore di compilazione. I registri di compilazione dell'applicazione potrebbero non indicare direttamente un out-of-memory errore

poiché il runtime dell'applicazione può bloccarsi in modo imprevisto. La configurazione di un heap di dimensioni pari a quelle della memoria host potrebbe comportare lo scambio o l'interruzione di altri processi da parte del sistema operativo host e potenzialmente interrompere il processo di compilazione. Come riferimento, Node.js consiglia di impostare una dimensione massima dell'heap di 1536 MB su un computer con circa 2000 MB di memoria per lasciare una parte di memoria per altri usi.

La dimensione ottimale dell'heap dipende dalle esigenze dell'applicazione e dall'utilizzo delle risorse. Se riscontri out-of-memory errori, inizia con una dimensione dell'heap moderata e poi aumentala gradualmente secondo necessità. Come linea guida, consigliamo di iniziare con 6000 MB per un tipo di Standard istanza, 12000 MB per un tipo di Large istanza e 60000 MB per un tipo di istanza. XLarge

Creazione di un webhook in entrata per avviare una build

Configura un webhook in entrata nella console Amplify per avviare una build senza inserire codice nel tuo repository Git. Puoi utilizzare i webhook con strumenti CMS headless (come Contentful o GraphCMS) per avviare una build ogni volta che il contenuto cambia o per eseguire build giornaliere utilizzando servizi come Zapier.

Per creare un webhook in entrata

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegliere l'app per la quale si desidera creare un webhook.
3. Nel pannello di navigazione, scegli Hosting, quindi Crea impostazioni.
4. Nella pagina delle impostazioni di creazione, scorri verso il basso fino alla sezione Webhook in arrivo e scegli Crea webhook.
5. Nella finestra di dialogo Create webhook (Crea webhook), seguire questa procedura:
   a. Per il nome del webhook, inserite un nome per il webhook.
   b. Per creare Branch, seleziona il ramo da creare in base alle richieste di webhook in entrata.
   c. Scegli Crea webhook.
6. Nella sezione Webhook in entrata, esegui una delle seguenti operazioni:
   - Copia l'URL del webhook e forniscilo a uno strumento CMS headless o a un altro servizio per avviare le build.
   - Esegui il comando curl in una finestra di terminale per iniziare una nuova build.

Impostazione della notifica e-mail per le compilazioni

Puoi configurare notifiche e-mail per un' AWS Amplify app per avvisare le parti interessate o i membri del team quando una build ha esito positivo o negativo. Amplify Hosting crea un argomento Amazon Simple Notification Service (SNS) nel tuo account e lo utilizza per configurare le notifiche e-mail. Le notifiche possono essere configurate per applicarsi a tutte le filiali o a rami specifici di un'app Amplify.

Impostazione della notifica e-mail

Utilizza le seguenti procedure per configurare le notifiche e-mail per tutte le filiali o filiali specifiche di un'app Amplify.

Per configurare le notifiche e-mail per un'applicazione Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui desideri configurare le notifiche e-mail.
3. Nel pannello di navigazione, scegli Hosting, Crea notifiche. Nella pagina Crea notifiche, scegli Gestisci notifiche.
4. Nella pagina Gestisci notifiche, scegli Aggiungi nuovo.
5. Esegui una di queste operazioni:
   - Per inviare le notifiche per una singola filiale, per E-mail, inserisci l'indirizzo e-mail a cui desideri che vengano inviate le notifiche. Per Branch, seleziona il nome della filiale per cui inviare le notifiche.
   - Per inviare le notifiche a tutte le filiali collegate, per E-mail, inserisci l'indirizzo e-mail a cui desideri che vengano inviate le notifiche. Per Branch, scegli Tutte le filiali.
6. Scegli Save (Salva).

Connessione di un dominio personalizzato

Puoi connettere un'app che hai distribuito con Amplify Hosting a un dominio personalizzato. Quando utilizzi Amplify per distribuire la tua app web, Amplify la ospita per te sul dominio predefinito con un URL come. `amplifyapp.com` `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com` Quando colleghi la tua app a un dominio personalizzato, gli utenti vedono che la tua app è ospitata su un URL personalizzato, ad esempio. `https://www.example.com`

Puoi acquistare un dominio personalizzato tramite un registrar di domini accreditato come Amazon Route 53 o. GoDaddy Route 53 è il servizio Web DNS (Domain Name System) di Amazon. Per ulteriori informazioni sull'utilizzo di Route 53, consulta Cos'è Amazon Route 53. Per un elenco dei registrar di domini accreditati di terze parti, consulta la pagina Accredited Registrar Directory nel sito Web ICANN.

Quando si configura un dominio personalizzato, è possibile utilizzare il certificato gestito predefinito fornito da Amplify oppure è possibile utilizzare un certificato personalizzato. È possibile modificare il certificato in uso per il dominio in qualsiasi momento. Per informazioni dettagliate sulla gestione dei certificati, consultaUtilizzo dei certificati SSL/TLS.

Prima di procedere con la configurazione di un dominio personalizzato, verificare di aver soddisfatto i prerequisiti descritti di seguito.

- Possiedi un nome di dominio registrato.
- Hai un certificato emesso o importato in AWS Certificate Manager.
- Hai distribuito la tua app su Amplify Hosting.

  Per ulteriori informazioni sul completamento di questa fase, consultaGuida introduttiva alla distribuzione di un'app su Amplify Hosting.
- Hai una conoscenza di base dei domini e della terminologia DNS.

  Per ulteriori informazioni su domini e DNS, consulta. Comprensione della terminology and concepts DNS

Argomenti

- Comprensione della terminology and concepts DNS
- Utilizzo dei certificati SSL/TLS

- Aggiunta di un dominio personalizzato gestito da Amazon Route 53
- Aggiunta di un dominio personalizzato gestito da un provider DNS di terze parti
- Aggiornamento dei record DNS per un dominio gestito da GoDaddy
- Aggiornamento del certificato SSL/TLS per un dominio
- Gestione dei sottodomini
- Configurazione di sottodomini wildcard
- Configurazione di sottodomini automatici per un dominio personalizzato Amazon Route 53
- Risoluzione dei problemi dei domini personalizzati

Comprensione della terminology and concepts DNS

Se non conosci i termini e i concetti associati al Domain Name System (DNS), i seguenti argomenti possono aiutarti a comprendere le procedure per aggiungere domini personalizzati.

Terminologia DNS

Di seguito è riportato un elenco di termini comuni al DNS. Possono aiutarti a comprendere le procedure per aggiungere domini personalizzati.

CNAME

Un Canonical Record Name (CNAME) è un tipo di record DNS che maschera il dominio per un insieme di pagine Web e le fa apparire come se si trovassero altrove. Un CNAME indirizza un sottodominio a un nome di dominio completo (FQDN). Ad esempio, puoi creare un nuovo record CNAME per mappare il sottodominio www.example.com, dove www è il sottodominio, al dominio FQDN branch-name.d1m7bki6tdw1.cloudfront.net assegnato alla tua app nella console Amplify.

ANAME

Un record ANAME è come un record CNAME, ma a livello principale. Un ANAME indirizza la radice del dominio a un nome di dominio completo. Tale FQDN punta a un indirizzo IP.

Server dei nomi

Un name server è un server su Internet specializzato nella gestione di richieste riguardanti l'ubicazione dei vari servizi di un nome di dominio. Se configuri il tuo dominio in Amazon Route 53, al tuo dominio è già assegnato un elenco di name server.

Record NS

Un record NS rimanda ai name server che cercano i dettagli del tuo dominio.

Verifica DNS

Un Domain Name System (DNS) è come una rubrica telefonica che traduce i nomi di dominio leggibili dall'uomo in indirizzi IP compatibili con il computer. Quando si digita **`https://google.com`** in un browser, viene eseguita un'operazione di ricerca nel provider DNS per trovare l'indirizzo IP del server che ospita il sito Web.

I provider DNS contengono i record dei domini e gli indirizzi IP corrispondenti. I record DNS più utilizzati sono i record CNAME, ANAME e NS.

Amplify utilizza un record CNAME per verificare che tu sia il proprietario del tuo dominio personalizzato. Se ospiti il tuo dominio con Route 53, la verifica viene eseguita automaticamente per tuo conto. Tuttavia, se ospiti il tuo dominio presso un provider di terze parti GoDaddy, ad esempio, devi aggiornare manualmente le impostazioni DNS del dominio e aggiungere un nuovo record CNAME fornito da Amplify.

Processo di attivazione del dominio personalizzato

Quando colleghi l'app Amplify a un dominio personalizzato nella console Amplify, Amplify deve completare diversi passaggi prima di poter visualizzare l'app utilizzando il dominio personalizzato. L'elenco seguente descrive ogni fase del processo di configurazione e attivazione del dominio.

Creazione SSL/TLS

Se utilizzi un certificato gestito, AWS Amplify emette un certificato SSL/TLS per configurare un dominio personalizzato sicuro.

Configurazione e verifica SSL/TLS

Prima di emettere un certificato gestito, Amplify verifica che tu sia il proprietario del dominio. Per i domini gestiti da Amazon Route 53, Amplify aggiorna automaticamente il record di verifica DNS. Per i domini gestiti al di fuori di Route 53, devi aggiungere manualmente il record di verifica DNS fornito nella console Amplify al tuo dominio con un provider DNS di terze parti.

Se utilizzi un certificato personalizzato, sei responsabile della convalida della proprietà del dominio.

Attivazione del dominio

Il dominio è stato verificato con successo. Per i domini gestiti al di fuori di Route 53, devi aggiungere manualmente i record CNAME forniti nella console Amplify al tuo dominio con un provider DNS di terze parti.

Utilizzo dei certificati SSL/TLS

Un protocollo. SSL/TLS certificate is a digital document that allows web browsers to identify and establish encrypted network connections to web sites using the secure SSL/TLS Quando si configura un dominio personalizzato, è possibile utilizzare il certificato gestito predefinito fornito da Amplify oppure è possibile utilizzare un certificato personalizzato.

Con un certificato gestito, Amplify emette un certificato SSL/TLS per tutti i domini collegati alla tua app in modo che tutto il traffico sia protetto tramite HTTPS/2. Il certificato predefinito generato da AWS Certificate Manager (ACM) è valido per 13 mesi e si rinnova automaticamente finché l'app è ospitata con Amplify.

> **Warning**
>
> Amplify non può rinnovare il certificato se il record di verifica CNAME è stato modificato o eliminato nelle impostazioni DNS del tuo provider di dominio. È necessario eliminare e aggiungere nuovamente il dominio nella console Amplify.

Per utilizzare un certificato personalizzato, è necessario prima ottenere un certificato dall'autorità di certificazione di terze parti di propria scelta. Amplify Hosting supporta due tipi di certificati: RSA (Rivest-Shamir-Adleman) ed ECDSA (Elliptic Curve Digital Signature Algorithm). Ogni tipo di certificato deve rispettare i seguenti requisiti.

Certificati RSA

- Amplify Hosting supporta chiavi RSA a 1024, 2048, 3072 e 4096 bit.
- AWS Certificate Manager (ACM) emette certificati RSA con chiavi fino a 2048 bit.
- Per usare un certificato RSA a 3072 o 4096 bit, ottenerlo esternamente e importarlo in ACM. Sarà quindi disponibile per l'uso con Amplify Hosting.

Certificati ECDSA

- Amplify Hosting supporta chiavi a 256 bit.
- Usa la curva ellittica prime256v1 per ottenere un certificato ECDSA per Amplify Hosting.

Dopo aver ottenuto un certificato, importalo in. AWS Certificate Manager ACM è un servizio che consente il provisioning, la gestione e l'implementazione dei certificati SSL/TLS pubblici e privati con Servizi AWS le risorse connesse interne. Assicurati di richiedere o importare il certificato nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1).

Assicurati che il certificato personalizzato copra tutti i sottodomini che intendi aggiungere. Puoi usare un carattere jolly all'inizio del tuo nome di dominio per coprire più sottodomini. Ad esempio, se il tuo dominio è`example.com`, puoi includere il dominio wildcard. `*.example.com` Questo coprirà sottodomini come e. `product.example.com` `api.example.com`

Dopo che il certificato personalizzato sarà disponibile in ACM, potrai selezionarlo durante il processo di configurazione del dominio. Per istruzioni sull'importazione di certificati in AWS Certificate Manager, consulta Importazione di certificati in AWS Certificate Manager nella Guida per l'AWS Certificate Manager utente.

Se rinnovi o reimporti il certificato personalizzato in ACM, Amplify aggiorna i dati del certificato associati al tuo dominio personalizzato. Nel caso di certificati importati, ACM non gestisce automaticamente i rinnovi. Sei responsabile del rinnovo dei certificati personalizzati e della loro nuova importazione.

È possibile modificare il certificato in uso per un dominio in qualsiasi momento. Ad esempio, è possibile passare dal certificato gestito predefinito a un certificato personalizzato o passare da un certificato personalizzato a un certificato gestito. Inoltre, è possibile modificare il certificato personalizzato in uso con un altro certificato personalizzato. Per istruzioni sull'aggiornamento dei certificati, consulta Aggiornare il certificato SSL/TLS per un dominio.

Aggiunta di un dominio personalizzato gestito da Amazon Route 53

Amazon Route 53 è un servizio DNS altamente scalabile e disponibile. Per ulteriori informazioni, consulta Amazon Route 53 nella Guida per gli sviluppatori di Amazon Route 53. Se hai già un dominio Route 53, usa le seguenti istruzioni per connettere il tuo dominio personalizzato all'app Amplify.

Per aggiungere un dominio personalizzato gestito da Route 53

1. Accedi AWS Management Console e apri la console Amplify.

2. Scegli l'app che desideri connettere a un dominio personalizzato.
3. Nel pannello di navigazione, scegli Hosting, Domini personalizzati.
4. Nella pagina Domini personalizzati, scegli Aggiungi dominio.
5. Inserisci il nome del tuo dominio principale. Ad esempio, se il nome del tuo dominio è https://example.com, inserisci**`example.com`**.

   Quando inizi a digitare, nell'elenco vengono visualizzati tutti i domini root che già gestisci in Route 53. Puoi scegliere il dominio che desideri utilizzare dall'elenco. Se non possiedi già il dominio ed è disponibile, puoi acquistarlo in Amazon Route 53.
6. Dopo aver inserito il nome di dominio, scegli Configura dominio.
7. Per impostazione predefinita, Amplify crea automaticamente due voci di sottodominio per il tuo dominio. Ad esempio, se il tuo nome di dominio è example.com, vedrai i sottodomini https://www.example.come https://example.comun reindirizzamento impostato dal dominio root al sottodominio www.

   (Facoltativo) Puoi modificare la configurazione predefinita se desideri aggiungere solo sottodomini. Per modificare la configurazione predefinita, scegli Riscritture e reindirizzamenti dal pannello di navigazione, quindi configura il tuo dominio.
8. Scegli il certificato SSL/TLS da utilizzare. Puoi utilizzare il certificato gestito predefinito fornito da Amplify per te o un certificato di terze parti personalizzato in cui hai importato. AWS Certificate Manager
   - Utilizza il certificato gestito Amplify predefinito.
     - Scegli il certificato gestito Amplify.
   - Usa un certificato personalizzato di terze parti.
     a. Scegli un certificato SSL personalizzato.
     b. Selezionare il certificato da utilizzare dall'elenco.
9. Scegli Add domain (Aggiungi dominio).

> **Note**
>
> Possono essere necessarie fino a 24 ore prima che il DNS si propaghi e emetta il certificato. Per informazioni sulla risoluzione degli errori che si verificano, consulta. Risoluzione dei problemi dei domini personalizzati

Aggiunta di un dominio personalizzato gestito da un provider DNS di terze parti

Se non utilizzi Amazon Route 53 per gestire il tuo dominio, puoi aggiungere un dominio personalizzato gestito da un provider DNS di terze parti alla tua app distribuita con Amplify.

Se lo utilizzi GoDaddy, consulta le istruzioni specifiche the section called “Aggiornamento dei record DNS per un dominio gestito da GoDaddy” per questo provider.

Aggiunta di un dominio personalizzato gestito da un provider DNS di terze parti

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app a cui desideri aggiungere un dominio personalizzato.
3. Nel pannello di navigazione, scegli Hosting, Domini personalizzati.
4. Nella pagina Domini personalizzati, scegli Aggiungi dominio.
5. Inserisci il nome del tuo dominio principale. Ad esempio, se il nome del tuo dominio è https://example.com, inserisci**`example.com`**.
6. Amplify rileva che non stai utilizzando un dominio Route 53 e ti offre la possibilità di creare una zona ospitata in Route 53.
   - Per creare una zona ospitata in Route 53
     a. Scegli Crea zona ospitata sulla Route 53.
     b. Scegli Configura dominio.
     c. I server dei nomi delle zone ospitate vengono visualizzati nella console. Accedi al sito Web del tuo provider DNS e aggiungi i name server alle impostazioni DNS.
     d. Seleziona Ho aggiunto i server dei nomi sopra indicati al mio registro di dominio.
     e. Procedi al passaggio sette.
   - Per continuare con la configurazione manuale
     a. Scegli Configurazione manuale
     b. Scegli Configura dominio.
     c. Procedi al passaggio sette.
7. Per impostazione predefinita, Amplify crea automaticamente due voci di sottodominio per il tuo dominio. Ad esempio, se il tuo nome di dominio è example.com, vedrai i sottodomini https://

www.example.come https://example.comun reindirizzamento impostato dal dominio root al sottodominio www.

(Facoltativo) Puoi modificare la configurazione predefinita se desideri aggiungere solo sottodomini. Per modificare la configurazione predefinita, scegli Riscritture e reindirizzamenti dal pannello di navigazione e configura il tuo dominio.

8. Scegli il certificato SSL/TLS da utilizzare. Puoi utilizzare il certificato gestito predefinito fornito da Amplify per te o un certificato di terze parti personalizzato in cui hai importato. AWS Certificate Manager
   - Utilizza il certificato gestito Amplify predefinito.
     - Scegli il certificato gestito Amplify.
   - Usa un certificato personalizzato di terze parti.
     a. Scegli un certificato SSL personalizzato.
     b. Selezionare il certificato da utilizzare dall'elenco.
9. Scegli Add domain (Aggiungi dominio).
10. Se hai scelto Crea zona ospitata sulla Route 53 nel passaggio 6, procedi al passaggio 15.

    Se hai scelto la configurazione manuale, nel passaggio sei devi aggiornare i record DNS con il tuo provider di dominio di terze parti.

    Nel menu Azioni, scegli Visualizza record DNS. La schermata seguente mostra i record DNS visualizzati nella console.

DNS Records

Verify records in your domain registrar match these records.

Verification record

Hostname	Type	Data/URL
_39e1e8d7e0aedc8165cf52a176612124.testexample.com.	CNAME	_40404fb1d5a2a1bdec5b4ad98de4cfbb.mhbtsbpdnt.acm-validations.aws.

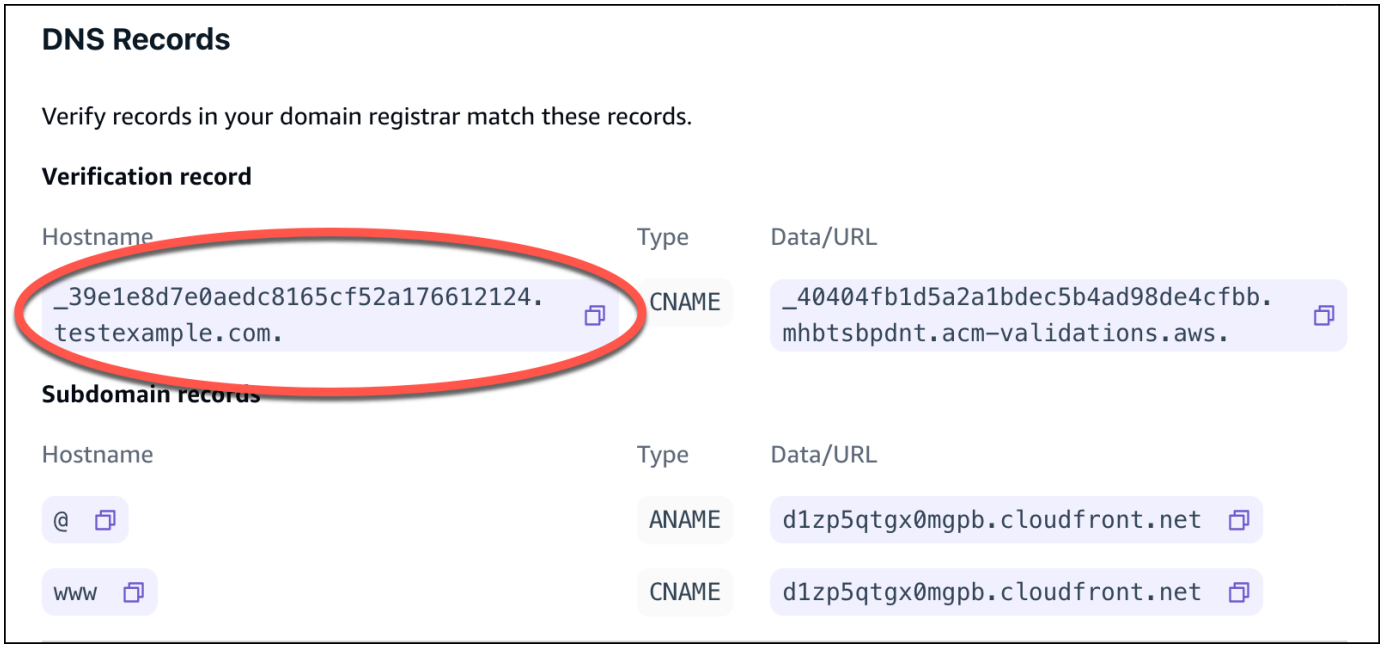
Subdomain records

Hostname	Type	Data/URL
@	ANAME	d1zp5qtgx0mgpb.cloudfront.net
www	CNAME	d1zp5qtgx0mgpb.cloudfront.net

11. Esegui una di queste operazioni:
    - Se lo stai usando GoDaddy, vai a. Aggiornamento dei record DNS per un dominio gestito da GoDaddy
    - Se utilizzi un provider DNS di terze parti diverso, vai al passaggio successivo di questa procedura.
12. Vai al sito web del tuo provider DNS, accedi al tuo account e individua le impostazioni di gestione DNS per il tuo dominio. Configurerai due record CNAME.
13. Configura il primo record CNAME per indirizzare il sottodominio verso il AWS server di convalida.

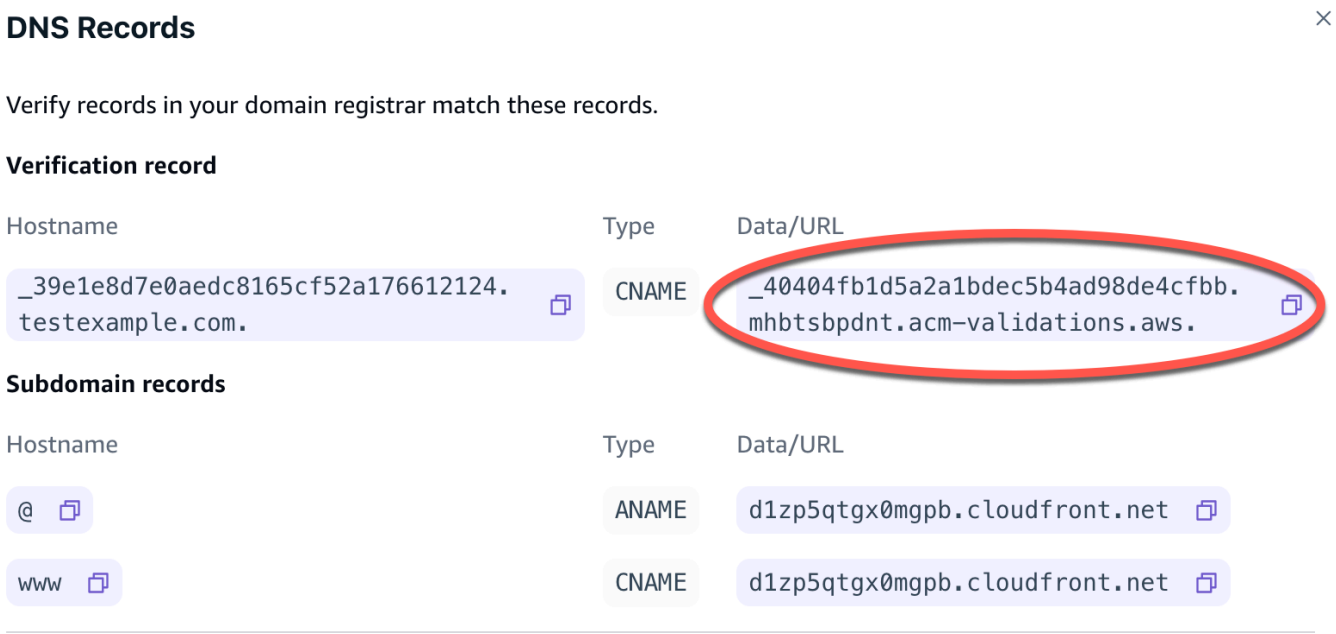
    Se la console Amplify visualizza un record DNS per la verifica della proprietà del sottodominio, ad esempio _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, inserisci solo il nome del sottodominio del record CNAME. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

    La schermata seguente mostra la posizione del registro di verifica da utilizzare.



Se la console Amplify visualizza un record del server di convalida ACM come _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, immettere il valore del record CNAME. **`_cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws`**

La schermata seguente mostra la posizione del registro di verifica ACM da utilizzare.



Amplify utilizza queste informazioni per verificare la proprietà del dominio e generare un certificato SSL/TLS per il dominio. Una volta che Amplify avrà convalidato la proprietà del tuo dominio, tutto il traffico verrà servito utilizzando HTTPS/2.

> **Note**
>
> Il certificato Amplify predefinito generato AWS Certificate Manager da (ACM) è valido per 13 mesi e si rinnova automaticamente finché l'app è ospitata con Amplify. Amplify non può rinnovare il certificato se il record di verifica CNAME è stato modificato o eliminato. È necessario eliminare e aggiungere nuovamente il dominio nella console Amplify.

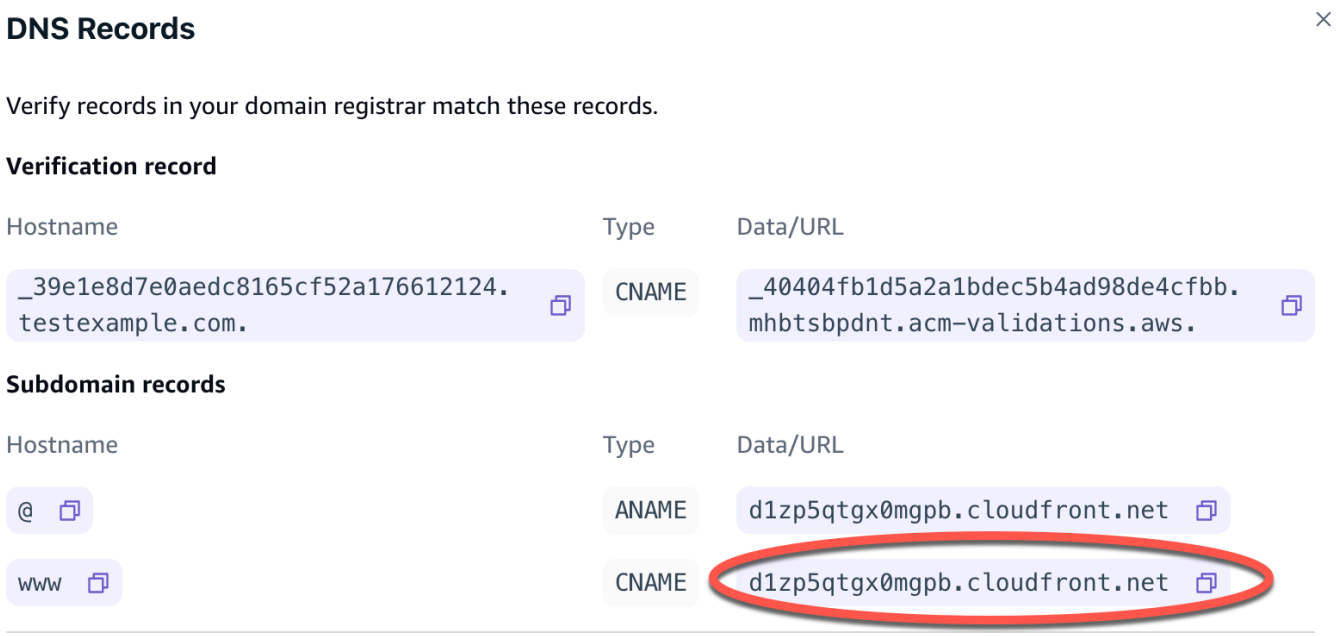
> **Important**
>
> È importante eseguire questo passaggio subito dopo aver aggiunto il dominio personalizzato nella console Amplify. L' AWS Certificate Manager (ACM) inizia immediatamente a tentare di verificare la proprietà. Nel tempo, i controlli diventano meno frequenti. Se aggiungi o aggiorni i record CNAME poche ore dopo aver creato l'app, ciò può far sì che l'app rimanga bloccata nello stato di verifica in sospeso.

14. Configura un secondo record CNAME per indirizzare i sottodomini al dominio Amplify. Ad esempio, se il sottodominio è www.example.com, inserisci www come nome del sottodominio.

    Se la console Amplify visualizza il dominio per la tua app come d111111abcdef8.cloudfront.net, inserisci il dominio Amplify. **`d111111abcdef8.cloudfront.net`**

    Se hai traffico di produzione, ti consigliamo di aggiornare questo record CNAME dopo che lo stato del dominio risulta DISPONIBILE nella console Amplify.

    La schermata seguente mostra la posizione del registro del nome di dominio da utilizzare.



15. Configura il record ANAME/ALIAS in modo che punti al dominio principale della tua app (ad esempio). https://example.com Un record ANAME indirizza la radice del dominio a un nome host. Se hai traffico di produzione, ti consigliamo di aggiornare il record ANAME dopo che lo stato del dominio risulta DISPONIBILE nella console. Per i provider DNS che non dispongono del supporto ANAME/ALIAS, consigliamo vivamente di migrare il DNS su Route 53. Per ulteriori informazioni, consulta Configurazione di Amazon Route 53 come servizio DNS.

> **Note**
>
> La verifica della proprietà del dominio e della propagazione DNS per i domini di terze parti può richiedere fino a 48 ore. Per informazioni sulla risoluzione degli errori che si verificano, consulta Risoluzione dei problemi relativi ai domini personalizzati.

Aggiornamento dei record DNS per un dominio gestito da GoDaddy

Se GoDaddy è il tuo provider DNS, usa le seguenti istruzioni per aggiornare i tuoi record DNS nell' GoDaddy interfaccia utente e completare la connessione dell'app Amplify al tuo dominio. GoDaddy

Per aggiungere un dominio personalizzato gestito da GoDaddy

1. Prima di poter aggiornare i record DNS con GoDaddy, completa i passaggi da uno a nove della procedurathe section called “Aggiunta di un dominio personalizzato gestito da un provider DNS di terze parti”.
2. Accedi al tuo GoDaddy account.
3. Nell'elenco dei domini, trova il dominio da aggiungere e scegli Gestisci DNS.
4. Nella pagina DNS, GoDaddy visualizza un elenco di record per il tuo dominio nella sezione Record DNS. È necessario aggiungere due nuovi record CNAME.
5. Crea il primo record CNAME per indirizzare i sottodomini al dominio Amplify.

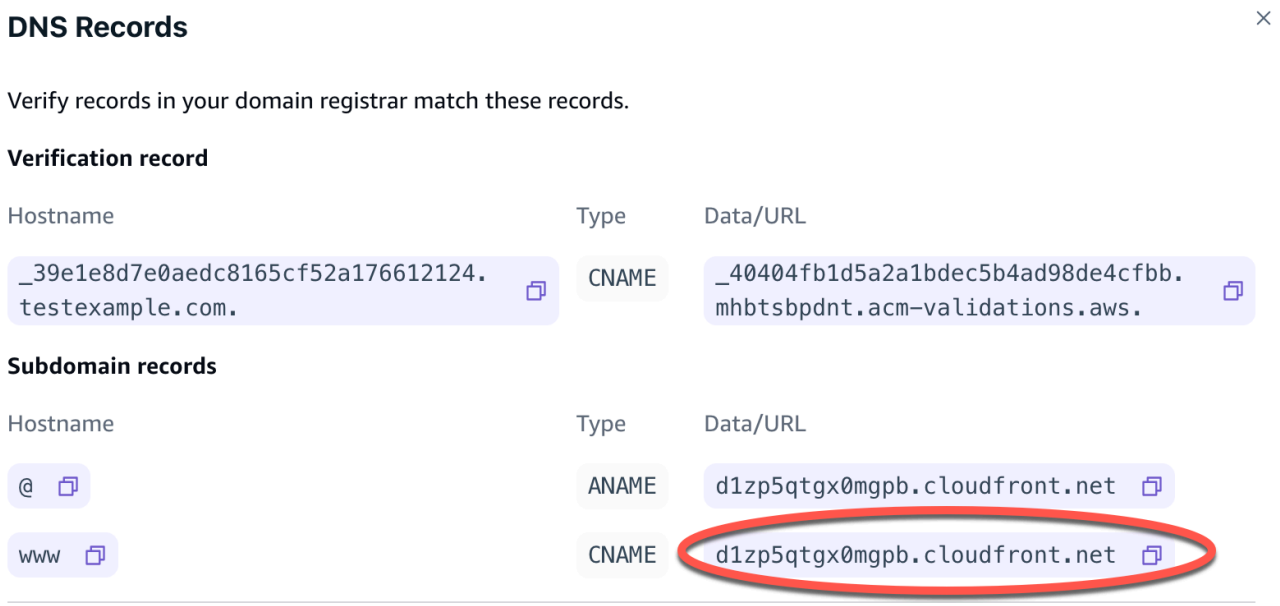
   a. Nella sezione Record DNS, scegli Aggiungi nuovo record.

   b. Per Tipo, scegli CNAME.

   c. Per Nome, inserisci solo il sottodominio. Ad esempio, se il sottodominio è www.example.com, inserisci www come Nome.

   d. Per Value, guarda i tuoi record DNS nella console Amplify e inserisci il valore. Se la console Amplify visualizza il dominio per la tua app come d111111abcdef8.cloudfront.net, inserisci Value. **`d111111abcdef8.cloudfront.net`**

   La schermata seguente mostra la posizione del registro del nome di dominio da utilizzare.



   e. Scegli Save (Salva).

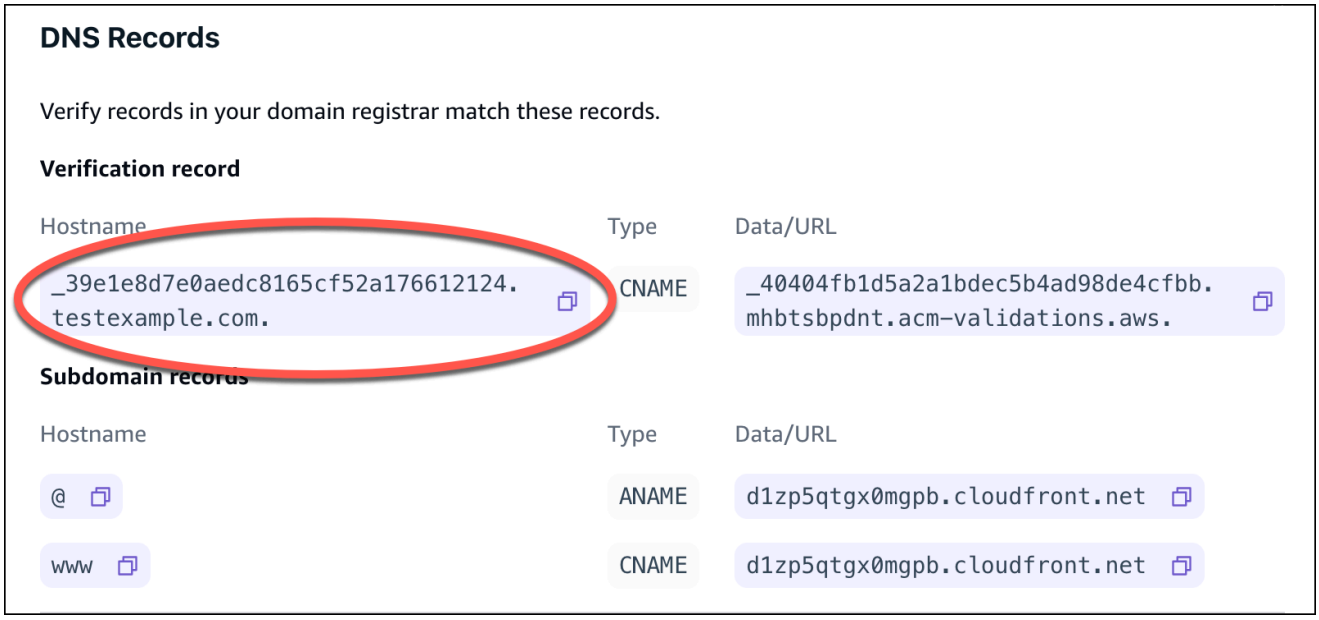
6. Crea il secondo record CNAME in modo che punti al server di convalida AWS Certificate Manager (ACM). Un singolo ACM convalidato genera un certificato SSL/TLS per il dominio.

   a. Per Tipo, scegli CNAME.

   b. Per Nome, inserisci il sottodominio.

      Ad esempio, se il record DNS nella console Amplify per la verifica della proprietà del sottodominio è _c3e2d7eaf1e656b73f46cd6980fdc0e.example.com, inserisci solo Nome. **`_c3e2d7eaf1e656b73f46cd6980fdc0e`**

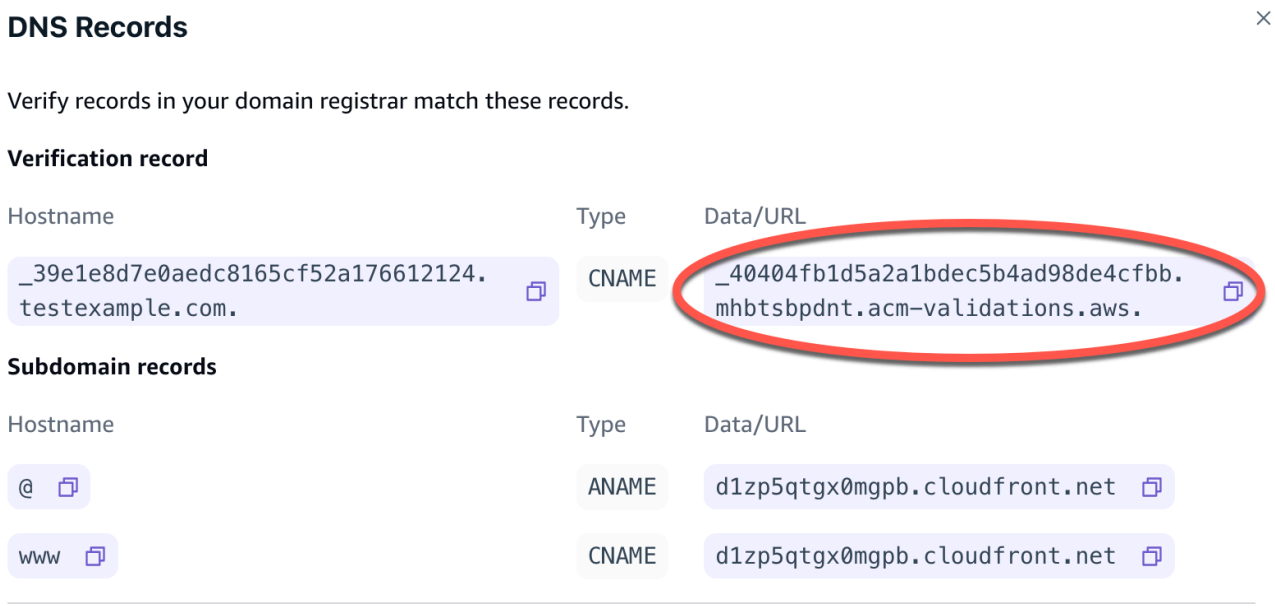
      La schermata seguente mostra la posizione del registro di verifica da utilizzare.



   c. Per Value, inserisci il certificato di convalida ACM.

      Ad esempio, se il server di convalida è _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws, inserisci _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws per Value.

      La schermata seguente mostra la posizione del registro di verifica ACM da utilizzare.



d. Scegli Save (Salva).

> **Note**
>
> Il certificato Amplify predefinito generato AWS Certificate Manager da (ACM) è valido per 13 mesi e si rinnova automaticamente finché l'app è ospitata con Amplify. Amplify non può rinnovare il certificato se il record di verifica CNAME è stato modificato o eliminato. È necessario eliminare e aggiungere nuovamente il dominio nella console Amplify.

7. Questa fase non è obbligatoria per i sottodomini. GoDaddy non supporta il ANAME/ALIAS records. For DNS providers that do not have ANAME/ALIAS supporto, consigliamo vivamente di migrare il DNS ad Amazon Route 53. Per ulteriori informazioni, consulta Configurazione di Amazon Route 53 come servizio DNS.

   Se desideri rimanere GoDaddy come provider e aggiornare il dominio principale, aggiungi Forwarding e configura un dominio successivo:

   a. Nella pagina DNS, individua il menu nella parte superiore della pagina e scegli Inoltro.
   b. Nella sezione Dominio, scegli Aggiungi inoltro.
   c. Scegli http://, quindi inserisci il nome del sottodominio a cui inoltrare (ad esempio, www.example.com) per l'URL di destinazione.
   d. Per Forward Type, scegliete Temporaneo (302).
   e. Scegliete, Salva.

Aggiornamento del certificato SSL/TLS per un dominio

È possibile modificare il certificato SSL/TLS utilizzato per un dominio in qualsiasi momento. Ad esempio, puoi passare dall'utilizzo di un certificato gestito all'utilizzo di un certificato personalizzato. Ciò è utile se desideri gestire il certificato e le relative notifiche di scadenza. Puoi anche modificare il certificato personalizzato utilizzato per il dominio. Le modifiche al certificato SSL non comporteranno tempi di inattività per il dominio attivo. Per ulteriori informazioni sui certificati, consulta Utilizzo dei certificati SSL/TLS.

Utilizzare la procedura seguente per aggiornare il tipo di certificato o il certificato personalizzato in uso per un dominio.

Per aggiornare il certificato di un dominio

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app che desideri aggiornare.
3. Nel pannello di navigazione, scegli Hosting, Domini personalizzati.
4. Nella pagina Domini personalizzati, scegli Configurazione del dominio.
5. Nella pagina dei dettagli del tuo dominio, individua la sezione Certificato SSL personalizzato. La procedura per aggiornare il certificato varia a seconda del tipo di modifica che desideri apportare.
   - Per passare da un certificato personalizzato al certificato gestito Amplify predefinito
     - Scegli il certificato gestito Amplify.
   - Per passare da un certificato gestito a un certificato personalizzato
     a. Scegli un certificato SSL personalizzato.
     b. Selezionare il certificato da utilizzare dall'elenco.
   - Per modificare un certificato personalizzato con un altro certificato personalizzato
     - Per il certificato SSL personalizzato, seleziona il nuovo certificato da utilizzare dall'elenco.
6. Scegli Save (Salva). I dettagli sullo stato del dominio indicheranno che Amplify ha avviato il processo di creazione SSL per un certificato gestito o il processo di configurazione per un certificato personalizzato.

Gestione dei sottodomini

Un sottodominio è la parte dell'URL che appare prima del nome di dominio. Ad esempio, www è il sottodominio di www.amazon.com e aws è il sottodominio di aws.amazon.com. Se disponi già di un sito Web di produzione, potresti voler collegare solo un sottodominio. I sottodomini possono anche essere multilivello, ad esempio beta.alpha.example.com ha il sottodominio multilivello beta.alpha.

Solo per aggiungere un sottodominio

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app a cui desideri aggiungere un sottodominio.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Domini personalizzati.
4. Nella pagina Domini personalizzati, scegli Aggiungi dominio.
5. Inserisci il nome del tuo dominio principale, quindi scegli Configura dominio. Ad esempio, se il nome del tuo dominio è https://example.com, inserisci example.com.
6. Scegli Exclude root e modifica il nome del sottodominio. Ad esempio, se il dominio è example.com, puoi modificarlo per aggiungere solo il sottodominio alpha.
7. Scegli Add domain (Aggiungi dominio).

Per aggiungere un sottodominio multilivello

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'applicazione a cui desideri aggiungere un sottodominio multilivello.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Domini personalizzati.
4. Nella pagina Domini personalizzati, scegli Aggiungi dominio.
5. Inserisci il nome di un dominio con un sottodominio, scegli Escludi root e modifica il sottodominio per aggiungere un nuovo livello.

   Ad esempio, se hai un dominio chiamato alpha.example.com e desideri creare un sottodominio multilivello beta.alpha.example.com, devi inserire beta come valore del sottodominio.
6. Scegli Add domain (Aggiungi dominio).

Aggiunta o modifica di un sottodominio

Dopo aver aggiunto un dominio personalizzato a un'app, puoi modificare un sottodominio esistente o aggiungere un nuovo sottodominio.

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui desideri gestire i sottodomini.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Domini personalizzati.
4. Nella pagina Domini personalizzati, scegli Configurazione dominio.
5. Nella sezione Sottodomini, puoi modificare i sottodomini esistenti secondo necessità.
6. (Facoltativo) Per aggiungere un nuovo sottodominio, scegli Aggiungi nuovo.
7. Scegli Save (Salva).

Configurazione di sottodomini wildcard

Amplify Hosting ora supporta i sottodomini wildcard. Un sottodominio wildcard è un sottodominio generico che consente di indirizzare sottodomini esistenti e non esistenti a un ramo specifico dell'applicazione. Quando usi un wildcard per associare tutti i sottodomini di un'app a un ramo specifico, puoi offrire gli stessi contenuti agli utenti dell'app in qualsiasi sottodominio ed evitare di configurare ogni sottodominio singolarmente.

Per creare un sottodominio con caratteri jolly, specifica un asterisco (*) come nome del sottodominio. Ad esempio, se specifichi il sottodominio wildcard `*.example.com` per un ramo specifico della tua app, qualsiasi URL che termina con example.com verrà indirizzato al ramo. In questo caso, le richieste `dev.example.com` e `prod.example.com` verranno indirizzate al sottodominio. `*.example.com`

Nota che Amplify supporta i sottodomini wildcard solo per un dominio personalizzato. Non puoi utilizzare questa funzionalità con il `amplifyapp.com` dominio predefinito.

I seguenti requisiti si applicano ai sottodomini wildcard:

- Il nome del sottodominio deve essere specificato solo con un asterisco (*).
- Non puoi utilizzare un carattere jolly per sostituire parte di un nome di sottodominio, come *domain.example.com.
- Non puoi sostituire un sottodominio nel mezzo di un nome di dominio, come ad esempio subdomain.*.example.com.

- Per impostazione predefinita, tutti i certificati forniti da Amplify coprono tutti i sottodomini di un dominio personalizzato.

Per aggiungere o eliminare un sottodominio wildcard

Dopo aver aggiunto un dominio personalizzato a un'app, puoi aggiungere un sottodominio wildcard per un ramo dell'app.

1. Accedi AWS Management Console e apri la console Amplify Hosting.
2. Scegli l'applicazione per la quale desideri gestire i sottodomini jolly.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Domini personalizzati.
4. Nella pagina Domini personalizzati, scegli Configurazione dominio.
5. Nella sezione Sottodomini, puoi aggiungere o eliminare sottodomini wildcard.
   - Per aggiungere un nuovo sottodominio con caratteri jolly
     a. Seleziona Add new (Aggiungi nuovo).
     b. Per il sottodominio, immettere un*****.
     c. Per il ramo dell'app, seleziona il nome di un ramo dall'elenco.
     d. Scegli Save (Salva).
   - Per eliminare un sottodominio wildcard
     a. Scegli Rimuovi accanto al nome del sottodominio. Il traffico verso un sottodominio non configurato in modo esplicito si interrompe e Amplify Hosting restituisce un codice di stato 404 a tali richieste.
     b. Scegli Save (Salva).

Configurazione di sottodomini automatici per un dominio personalizzato Amazon Route 53

Dopo che un'app è connessa a un dominio personalizzato in Route 53, Amplify consente di creare automaticamente sottodomini per le filiali appena connesse. Ad esempio, se colleghi il tuo ramo di sviluppo, Amplify può creare automaticamente dev.exampledomain.com. Quando elimini un ramo, tutti i sottodomini associati vengono eliminati automaticamente.

Per impostare la creazione automatica di sottodomini per le filiali appena connesse

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli un'app connessa a un dominio personalizzato gestito in Route 53.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Domini personalizzati.
4. Nella pagina Domini personalizzati, scegli Configurazione dominio.
5. Nella sezione Creazione automatica di sottodomini, attiva la funzionalità.

> **Note**
>
> Questa funzionalità è disponibile solo per i domini root, ad esempio exampledomain.com. La console Amplify non visualizza questa casella di controllo se il dominio è già un sottodominio, ad esempio dev.exampledomain.com.

Anteprime Web con sottodomini

Dopo aver abilitato la creazione automatica di sottodomini utilizzando le istruzioni precedenti, le anteprime web delle pull request dell'app saranno accessibili anche con i sottodomini creati automaticamente. Quando una pull request viene chiusa, il ramo e il sottodominio associati vengono eliminati automaticamente. Per ulteriori informazioni sull'impostazione delle anteprime Web per le richieste pull, consultaAnteprime Web per le richieste pull.

Risoluzione dei problemi dei domini personalizzati

Se riscontri problemi durante l'aggiunta di un dominio personalizzato a un'app nella AWS Amplify console, consulta Risoluzione dei problemi relativi ai domini personalizzati il capitolo sulla risoluzione dei problemi di Amplify. Se non trovi una soluzione al tuo problema, contatta. Supporto Per ulteriori informazioni, consulta Creazione di una richiesta di assistenza nella Guida per l'utente di Supporto AWS .

Supporto firewall per i siti ospitati da Amplify

Il supporto firewall per i siti ospitati da Amplify consente di proteggere le applicazioni Web con un'integrazione diretta con. AWS WAF AWS WAF consente di configurare una serie di regole, denominate Web Access Control List (Web ACL), che consentono, bloccano o monitorano (contano) le richieste Web in base a regole e condizioni di sicurezza Web personalizzabili da voi definite. Quando integri la tua app Amplify AWS WAF con, ottieni maggiore controllo e visibilità sul traffico HTTP accettato dalla tua app. Per saperne di più AWS WAF, consulta How AWS WAF Works nella AWS WAF Developer Guide.

Il supporto firewall è disponibile in tutti gli ambienti Regioni AWS in cui Amplify Hosting opera. Questa integrazione rientra in una risorsa AWS WAF globale, simile a. CloudFront Il Web ACLs può essere collegato a più app di Amplify Hosting, ma devono risiedere nella stessa regione.

Puoi utilizzarla AWS WAF per proteggere la tua app Amplify da exploit web comuni, come SQL injection e cross-site scripting. Questi potrebbero influire sulla disponibilità e sulle prestazioni dell'app, compromettere la sicurezza o consumare risorse eccessive. Ad esempio, puoi creare regole per consentire o bloccare le richieste provenienti da intervalli di indirizzi IP specifici, le richieste provenienti da blocchi CIDR, le richieste che provengono da un paese o un'area geografica specifici o le richieste che contengono codice SQL o script imprevisti.

Puoi anche creare regole che corrispondono a una stringa specificata o un modello di espressione regolare in intestazioni HTTP, metodo, stringa di query, URI e il corpo della richiesta (entro i primi 8 KB). Inoltre, puoi creare regole per bloccare gli eventi provenienti da agenti utente, bot e raccoglitori di contenuti specifici. Ad esempio, puoi usare le regole basate sulla frequenza per specificare il numero di richieste Web consentite da ogni IP client in un periodo di 5 minuti, costantemente aggiornato, finale.

Per ulteriori informazioni sui tipi di regole supportate e sulle AWS WAF funzionalità aggiuntive, consulta la Guida per gli AWS WAF sviluppatori e l'AWS WAF API Reference.

> **Important**
>
> La sicurezza è una responsabilità condivisa tra te AWS e te. AWS WAF non è la soluzione a tutti i problemi di sicurezza di Internet ed è necessario configurarla per soddisfare i propri obiettivi di sicurezza e conformità. Per aiutarti a capire come applicare il modello

di responsabilità condivisa durante l'utilizzo AWS WAF, consulta la sezione Sicurezza nell'utilizzo del AWS WAF servizio.

Argomenti

- AWS WAF Attivazione di un'applicazione Amplify in AWS Management Console
- Dissociare un ACL Web da un'applicazione Amplify
- AWS WAF Attivazione di un'applicazione Amplify utilizzando AWS CDK
- In che modo Amplify si integra con AWS WAF
- Prezzi del firewall per le applicazioni Amplify

AWS WAF Attivazione di un'applicazione Amplify in AWS Management Console

Puoi abilitare AWS WAF le protezioni per un'app Amplify nella console Amplify o nella console. AWS WAF

- Console Amplify: puoi abilitare le funzionalità Firewall per un'app Amplify esistente associando un ACL AWS WAF Web alla tua app nella console Amplify. Usa la protezione con un solo clic per creare un ACL web con regole preconfigurate che consideriamo la migliore pratica per la maggior parte delle app. Hai la possibilità di personalizzare l'accesso in base all'indirizzo IP e al Paese. Le istruzioni contenute in questa sezione descrivono l'impostazione delle protezioni con un clic.
- AWS WAF console: utilizza un ACL Web preconfigurato creato nella AWS WAF console o utilizzando. AWS WAF APIs Devi creare il web ACLs che desideri associare a un'app Amplify nella regione Global CloudFront (). I siti Web regionali ACLs potrebbero già esistere nel tuo Account AWS, ma non sono compatibili con Amplify. Per istruzioni introduttive, consulta Configurazione AWS WAF e relativi componenti nella Guida per gli AWS WAF sviluppatori.

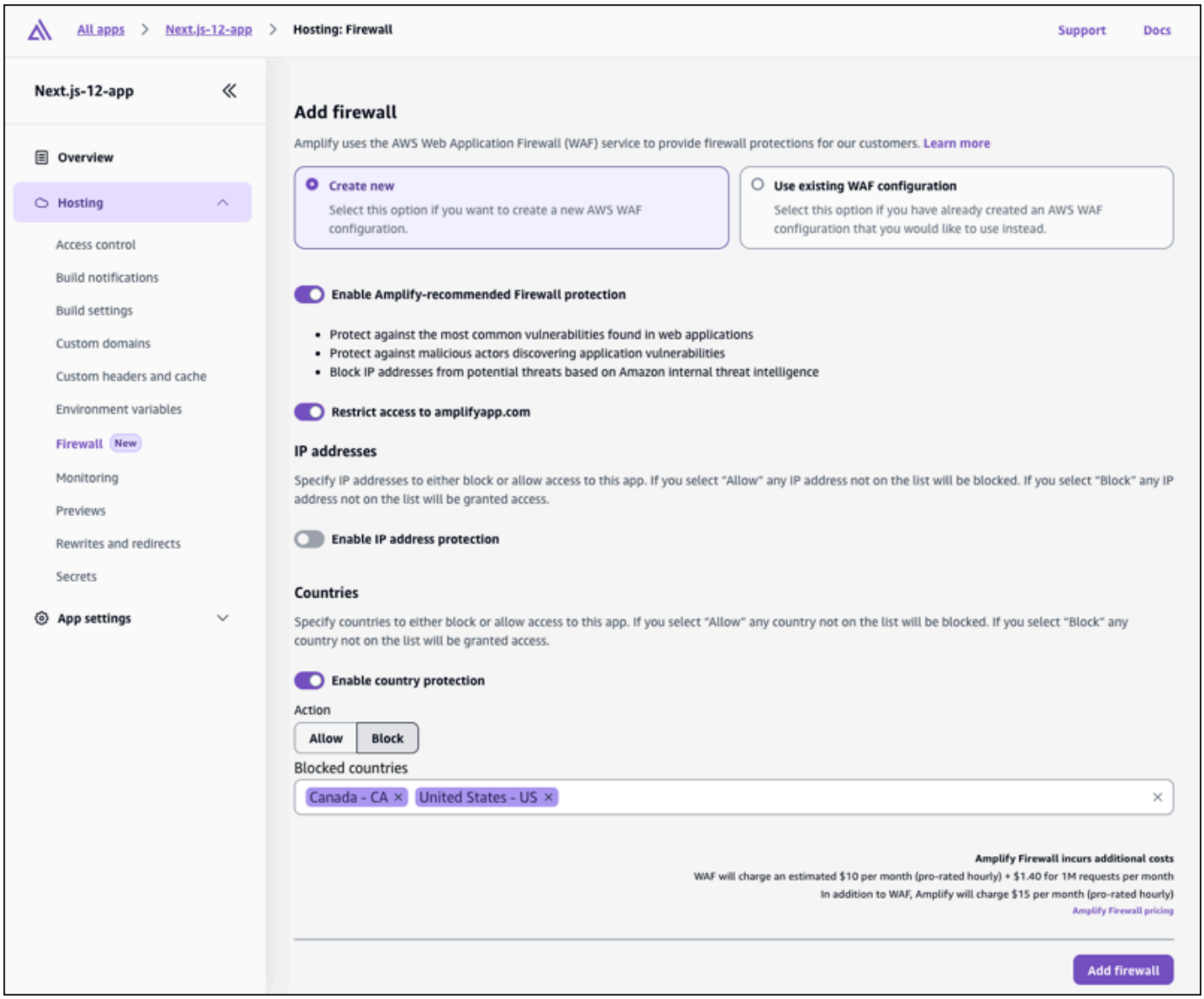
Usa la seguente procedura AWS WAF per abilitare un'app esistente nella console Amplify.

Abilita AWS WAF per un'app Amplify esistente

1. Accedi a AWS Management Console e apri la console Amplify all'indirizzo. https://console.aws.amazon.com/amplify/
2. Nella pagina Tutte le app, scegli il nome dell'app distribuita per abilitare la funzionalità Firewall.

3. Nel riquadro di navigazione, scegli Hosting, quindi scegli Firewall.

   La schermata seguente mostra come accedere alla pagina Aggiungi firewall nella console Amplify.



4. Nella pagina Aggiungi firewall, le tue azioni dipenderanno dal fatto che desideri creare una nuova AWS WAF configurazione o utilizzarne una esistente.

   - Crea una nuova AWS WAF configurazione.

     a. Scegli Crea nuova.

     b. Facoltativamente, abilita una delle seguenti configurazioni:

        i. Attiva la protezione firewall consigliata da Amplify.

ii. Attiva Limita l'accesso a amplifyapp.com per impedire l'accesso alla tua app sul dominio Amplify predefinito.

iii. Per gli indirizzi IP, attiva Abilita la protezione degli indirizzi IP.

   A. In Azione, scegli Consenti se desideri specificare gli indirizzi IP a cui accedere e tutti gli altri verranno bloccati. Scegli Blocca se desideri specificare gli indirizzi IP che verranno bloccati e tutti gli altri avranno accesso.

   B. Per la versione IP, seleziona IPV4o IPV6.

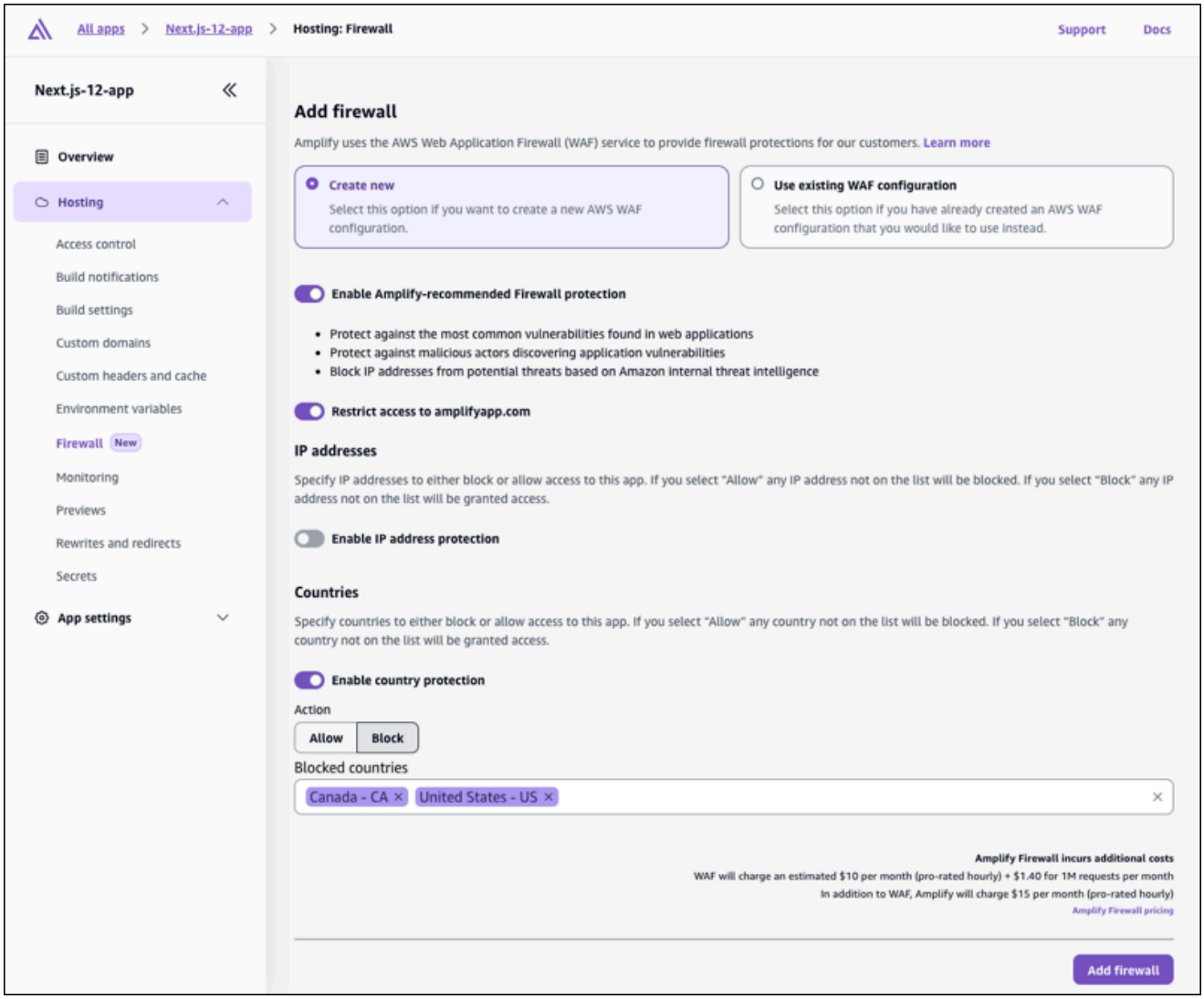
   C. Nella casella di testo Indirizzi IP, inserisci gli indirizzi IP consentiti o bloccati, uno per riga, in formato CIDR.

iv. Per i Paesi, attiva Abilita la protezione del paese.

   A. In Azione, scegli Consenti se desideri specificare i paesi che avranno accesso e tutti gli altri verranno bloccati. Scegli Blocca se desideri specificare i paesi che verranno bloccati e tutti gli altri avranno accesso.

   B. Per i Paesi, seleziona i Paesi consentiti o bloccati dall'elenco.

La schermata seguente mostra come abilitare una nuova AWS WAF configurazione per un'app.



- Usa una configurazione esistente AWS WAF .

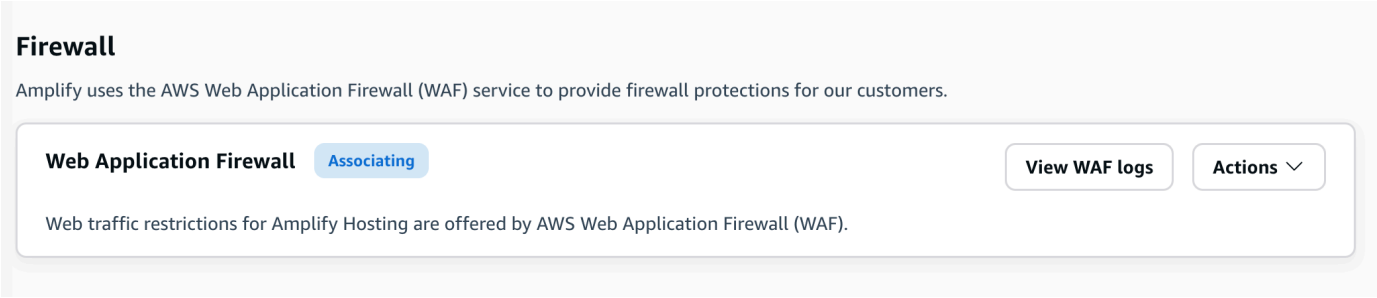
  a. Scegli Usa AWS WAF configurazione esistente.

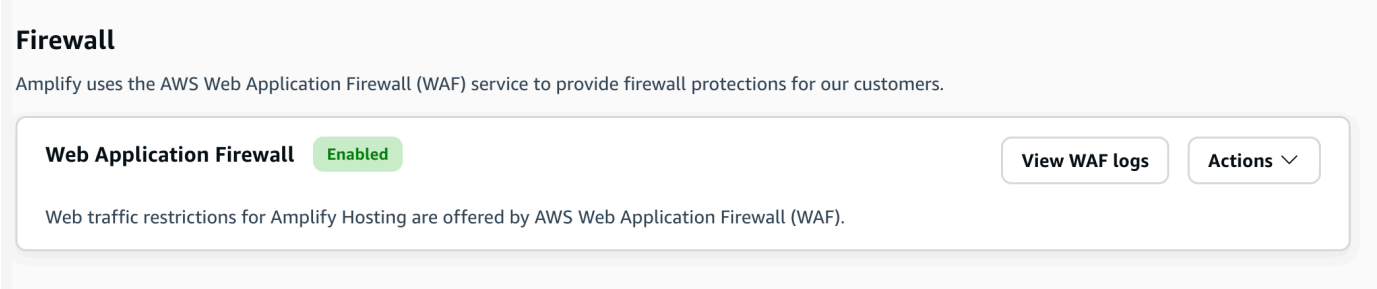
  b. Seleziona una configurazione salvata dall'elenco dei siti web ACLs AWS WAF nel tuo Account AWS. L'ACL web che associ alla tua app Amplify deve essere creato nella regione Global (). CloudFront I siti Web regionali ACLs potrebbero già esistere nel tuo Account AWS, ma non sono compatibili con Amplify.

5. Scegli Aggiungi firewall.

6. Nella pagina Firewall, viene visualizzato lo stato di associazione per indicare che le AWS WAF impostazioni vengono propagate. Una volta completato il processo, lo stato diventa Attivato.

   Le schermate seguenti mostrano lo stato di avanzamento del firewall nella console Amplify, indicando quando AWS WAF la configurazione è associata e abilitata.





Dissociare un ACL Web da un'applicazione Amplify

Non puoi eliminare un ACL web associato a un'app Amplify. È innanzitutto necessario dissociare l'ACL Web dall'app nella console Amplify. Quindi puoi eliminarlo nella console. AWS WAF

Per dissociare un ACL Web da un'app Amplify

1. Accedi a AWS Management Console e apri la console Amplify all'indirizzo. https://console.aws.amazon.com/amplify/
2. Nella pagina Tutte le app, scegli il nome dell'app da cui dissociare un ACL web.
3. Nel riquadro di navigazione, scegli Hosting, quindi scegli Firewall.
4. Nella pagina Firewall, scegli Azioni, quindi scegli Dissocia firewall.
5. Nella modalità di conferma, inserisci**`disassociate`**, quindi scegli Disassocia firewall.
6. Nella pagina Firewall, viene visualizzato lo stato Dissociazione per indicare che AWS WAF le impostazioni vengono propagate.

   Una volta completato il processo, è possibile eliminare l'ACL Web nella console. AWS WAF

AWS WAF Attivazione di un'applicazione Amplify utilizzando AWS CDK

È possibile utilizzare l'opzione AWS Cloud Development Kit (AWS CDK) AWS WAF per abilitare un'applicazione Amplify. Per ulteriori informazioni sull'utilizzo del CDK, consulta Cos'è il CDK? nella Guida per gli AWS Cloud Development Kit (AWS CDK) sviluppatori.

Il seguente esempio di TypeScript codice mostra come creare un' AWS CDK app con due stack CDK: uno per Amplify e uno per. AWS WAF Si noti che lo AWS WAF stack deve essere distribuito nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1). Lo stack di applicazioni Amplify può essere distribuito in una regione diversa. È necessario creare l'ACL Web che si desidera associare all'app Amplify nella regione Global (). CloudFront I siti Web regionali ACLs potrebbero già esistere nel tuo Account AWS, ma non sono compatibili con Amplify.

```
import * as cdk from "aws-cdk-lib";
import { Construct } from "constructs";
import * as wafv2 from "aws-cdk-lib/aws-wafv2";
import * as amplify from "aws-cdk-lib/aws-amplify";

interface WafStackProps extends cdk.StackProps {
  appArn: string;
}

export class AmplifyStack extends cdk.Stack {
  public readonly appArn: string;
  constructor(scope: Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);
    const amplifyApp = new amplify.CfnApp(this, "AmplifyApp", {
      name: "MyApp",
    });
    this.appArn = amplifyApp.attrArn;
  }
}

export class WAFStack extends cdk.Stack {
  constructor(scope: Construct, id: string, props: WafStackProps) {
    super(scope, id, props);
    const webAcl = new wafv2.CfnWebACL(this, "WebACL", {
      defaultAction: { allow: {} },
      scope: "CLOUDFRONT",
      rules: [
```

```
        // Add your own rules here.
      ],
      visibilityConfig: {
        cloudWatchMetricsEnabled: true,
        metricName: "my-metric-name",
        sampledRequestsEnabled: true,
      },
    });

    new wafv2.CfnWebACLAssociation(this, "WebACLAssociation", {
      resourceArn: props.appArn,
      webAclArn: webAcl.attrArn,
    });
  }
}

const app = new cdk.App();

// Create AmplifyStack in your desired Region.
const amplifyStack = new AmplifyStack(app, 'AmplifyStack', {
  env: { region: 'us-west-2' },
});

// Create WAFStack in IAD region, passing appArn from AmplifyStack.
new WAFStack(app, 'WAFStack', {
  env: { region: 'us-east-1' },
  crossRegionReferences: true,

  appArn: amplifyStack.appArn,  // Pass appArn from AmplifyStack.
});
```

In che modo Amplify si integra con AWS WAF

L'elenco seguente fornisce dettagli specifici su come è integrato il supporto Firewall AWS WAF e sui vincoli da considerare durante la creazione di siti Web ACLs e l'associazione alle app Amplify.

- Puoi abilitare AWS WAF qualsiasi tipo di app Amplify. Ciò include qualsiasi framework supportato, app renderizzate lato server (SSR) e siti completamente statici. AWS WAF è supportato per le app Amplify Gen 1 e Gen 2.

- Devi creare il web ACLs che desideri associare a un'app Amplify nella regione Global CloudFront (). I siti Web regionali ACLs potrebbero già esistere nel tuo Account AWS, ma non sono compatibili con Amplify.
- L'ACL web e l'app Amplify devono essere creati nello stesso ambiente. Account AWSÈ possibile utilizzarla AWS Firewall Manager per replicare AWS WAF le regole su più livelli Account AWS, per semplificare la gestione delle regole organizzative centralizzate e distribuite su più piattaforme. Account AWS Per ulteriori informazioni, consulta la sezione AWS Firewall Manager nella Guida per gli sviluppatori di AWS WAF .
- Puoi condividere lo stesso ACL web su più app Amplify nella stessa. Account AWS Tutte le app devono trovarsi nella stessa regione.
- Quando associ un ACL Web a un'app Amplify, l'ACL Web si collega a ogni ramo dell'app per impostazione predefinita. Quando crei nuove filiali, queste avranno l'ACL web.
- Quando associ un ACL web a un'app Amplify, questo viene automaticamente associato a tutti i domini dell'app. Tuttavia, puoi configurare regole che si applicano a un singolo nome di dominio utilizzando le regole di abbinamento Host-header.
- Non puoi eliminare un ACL web associato a un'app Amplify. Prima di eliminare un ACL web dalla AWS WAF console, devi dissociarlo dall'app.

Politica delle risorse Web ACL di Amplify

Per consentire ad Amplify di accedere all'ACL Web, durante l'associazione viene allegata una politica delle risorse all'ACL Web. Amplify costruisce automaticamente questa politica delle risorse, ma puoi visualizzarla utilizzando l'API. AWS WAFV2 GetPermissionPolicy Le seguenti autorizzazioni IAM sono necessarie per associare un ACL Web a un'app Amplify.

- amplify: ACL AssociateWeb
- wafv2: ACL AssociateWeb
- wafv2: PutPermissionPolicy
- wafv2: GetPermissionPolicy

Prezzi del firewall per le applicazioni Amplify

Il costo di implementazione AWS WAF su un'applicazione Amplify viene calcolato in base ai seguenti due componenti:

- AWS WAF utilizzo: ti verrà addebitato l' AWS WAF utilizzo in base al modello di prezzo. AWS WAF AWS WAF i costi si basano sugli elenchi di controllo degli accessi Web (Web ACLs) creati, sul numero di regole aggiunte per ACL Web e sul numero di richieste Web ricevute. Per i dettagli sui prezzi, vedere Prezzi di AWS WAF.
- Costo di integrazione di Amplify Hosting: è previsto un addebito di $15,00 al mese per app quando si collega un ACL Web a un'applicazione Amplify. Viene ripartito proporzionalmente su base oraria.

Distribuzioni del ramo feature e flussi di lavoro del team

Amplify Hosting è progettato per funzionare con feature branch e flussi di lavoro. GitFlow Amplify utilizza i branch Git per creare una nuova distribuzione ogni volta che connetti un nuovo ramo nel tuo repository. Dopo aver collegato la prima filiale, crei rami di funzionalità aggiuntive.

Per aggiungere un ramo a un'app

1. Scegli l'app a cui vuoi aggiungere un ramo.
2. Scegli Impostazioni app, quindi Impostazioni Branch.
3. Nella pagina delle impostazioni Branch, scegli Aggiungi filiale.
4. Seleziona un ramo dal tuo repository.
5. Scegli Aggiungi ramo.
6. Ridistribuisci la tua app.

Dopo aver aggiunto un ramo, l'app ha due distribuzioni disponibili nei domini predefiniti di Amplify, ad esempio e. https://main.appid.amplifyapp.comhttps://dev.appid.amplifyapp.com Questo può variare team-to-team, ma in genere la filiale principale tiene traccia del codice di rilascio ed è la filiale di produzione. Il ramo develop è utilizzato come ramo integrativo per testare le nuove funzionalità. Ciò consente ai beta tester di testare funzionalità inedite nell'implementazione della filiale di sviluppo, senza influire sugli utenti finali di produzione coinvolti nell'implementazione della filiale principale.

Argomenti

- Flussi di lavoro in team con app complete Amplify Gen 2
- Flussi di lavoro in team con app complete Amplify Gen 1
- Implementazioni di feature branch basate su pattern
- Generazione automatica in fase di compilazione della configurazione Amplify (solo app di prima generazione)
- Build di backend condizionali (solo app di prima generazione)
- Usa i backend Amplify tra le app (solo app di prima generazione)

Flussi di lavoro in team con app complete Amplify Gen 2

AWS Amplify Gen 2 introduce TypeScript un'esperienza di sviluppo basata sul codice per la definizione dei backend. Per ulteriori informazioni sui flussi di lavoro fullstack con le applicazioni Amplify Gen 2, consulta Flussi di lavoro Fullstack nei documenti Amplify.

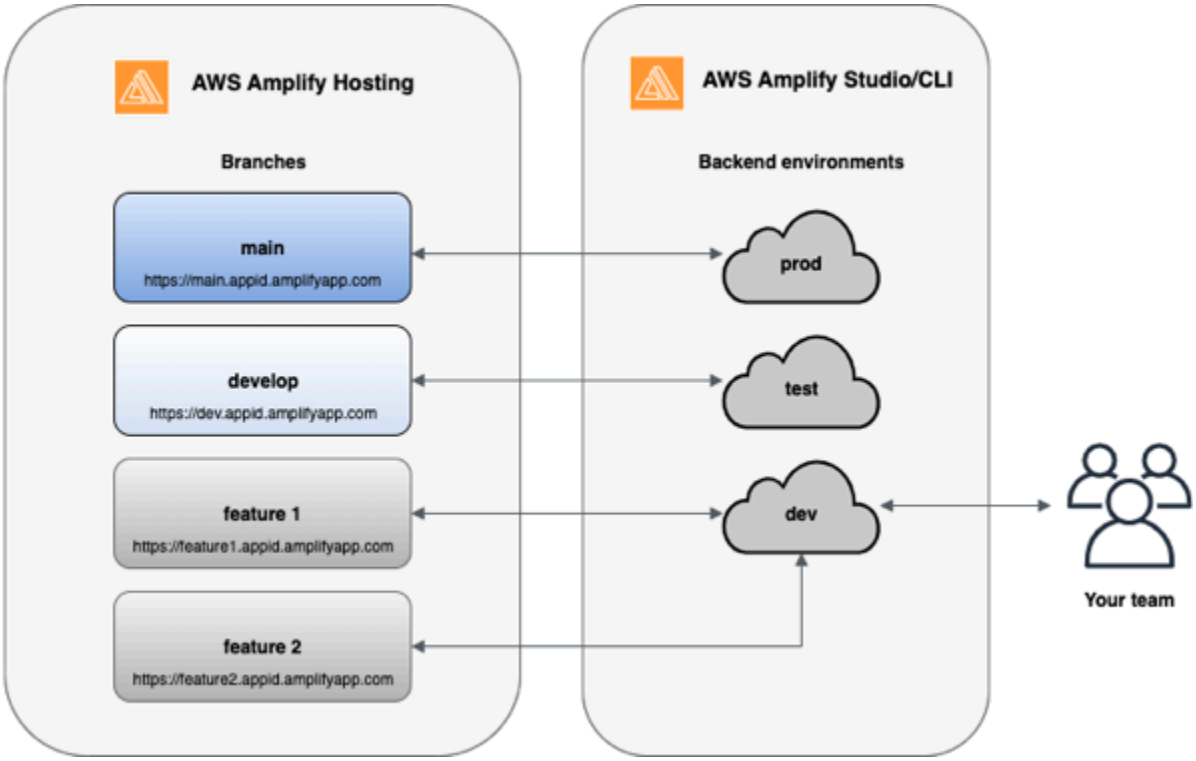
Flussi di lavoro in team con app complete Amplify Gen 1

Una distribuzione di feature branch consiste in un frontend e un ambiente backend opzionale. Il frontend è costruito e distribuito su una rete di distribuzione dei contenuti (CDN) globale, mentre il backend viene distribuito da Amplify Studio o dalla CLI di Amplify. AWS Per informazioni su come configurare questo scenario di distribuzione, consulta. Creazione di un backend per un'applicazione

Amplify Hosting implementa continuamente risorse di backend come le funzioni GraphQL e Lambda con le APIs tue implementazioni di feature branch. Puoi utilizzare i seguenti modelli di ramificazione per implementare il backend e il frontend con Amplify Hosting.

Flusso di lavoro del ramo feature

- Crea ambienti di backend di produzione, test e sviluppo con Amplify Studio o Amplify CLI.
- Mappa il backend prod sul ramo principale.
- Mappa il backend di test sul ramo di sviluppo.
- I membri del team possono utilizzare l'ambiente di backend di sviluppo per testare singoli rami di funzionalità.



1. Installare l'interfaccia a riga di comando di Amplify per inizializzare un nuovo progetto Amplify.

```
npm install -g @aws-amplify/cli
```

2. Inizializzare un ambiente di back-end prod per il progetto. Se non hai un progetto, creane uno usando strumenti di bootstrap come o Gatsby. create-react-app

```
create-react-app next-unicorn
cd next-unicorn
amplify init
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: prod
...
amplify push
```

3. Aggiungere gli ambienti di back-end test e dev.

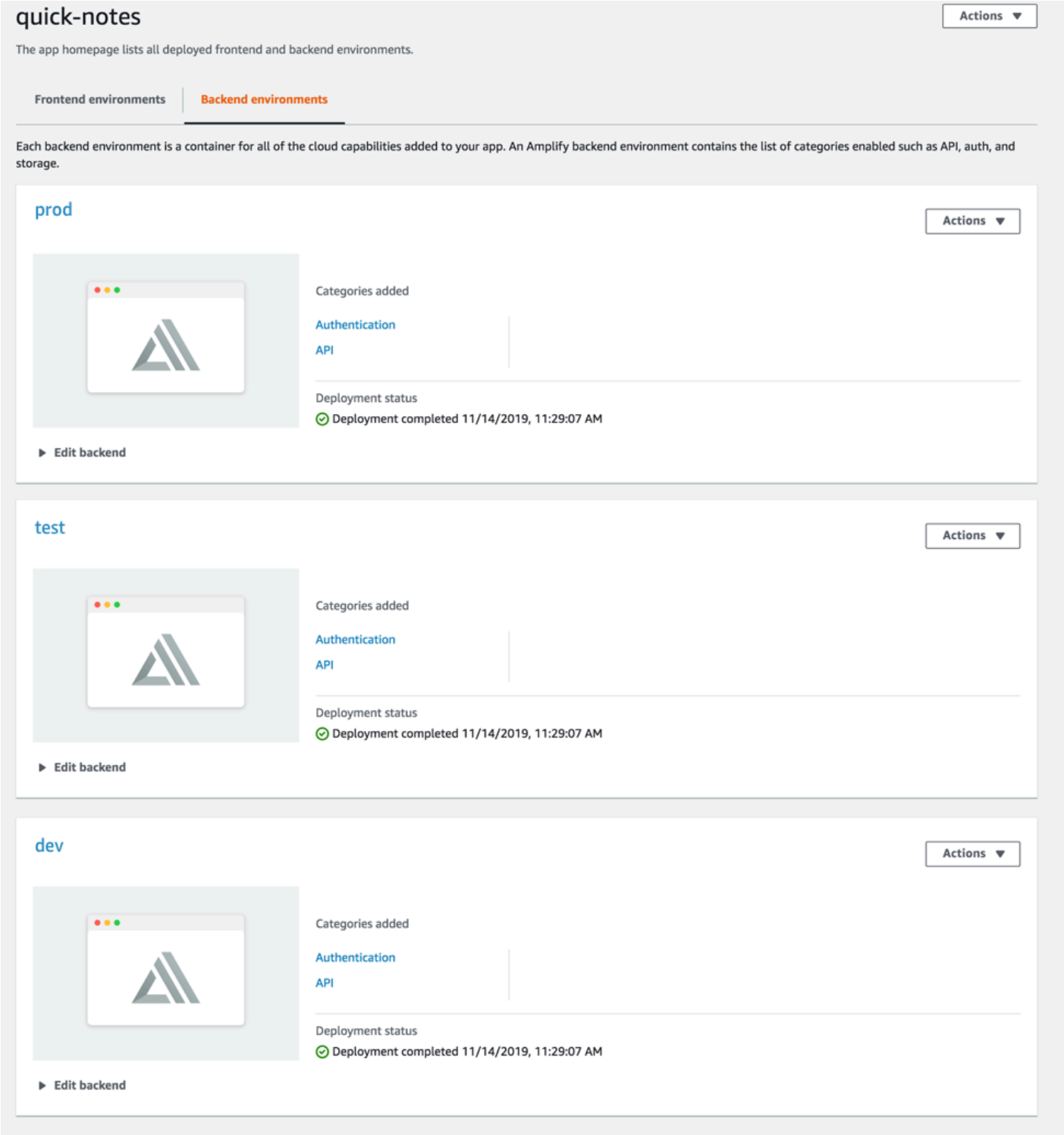
```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: test
...
amplify push
```

```
amplify env add
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: dev
...
amplify push
```

4. Invia il codice a un repository Git di tua scelta (in questo esempio supponiamo che tu abbia eseguito il push su main).

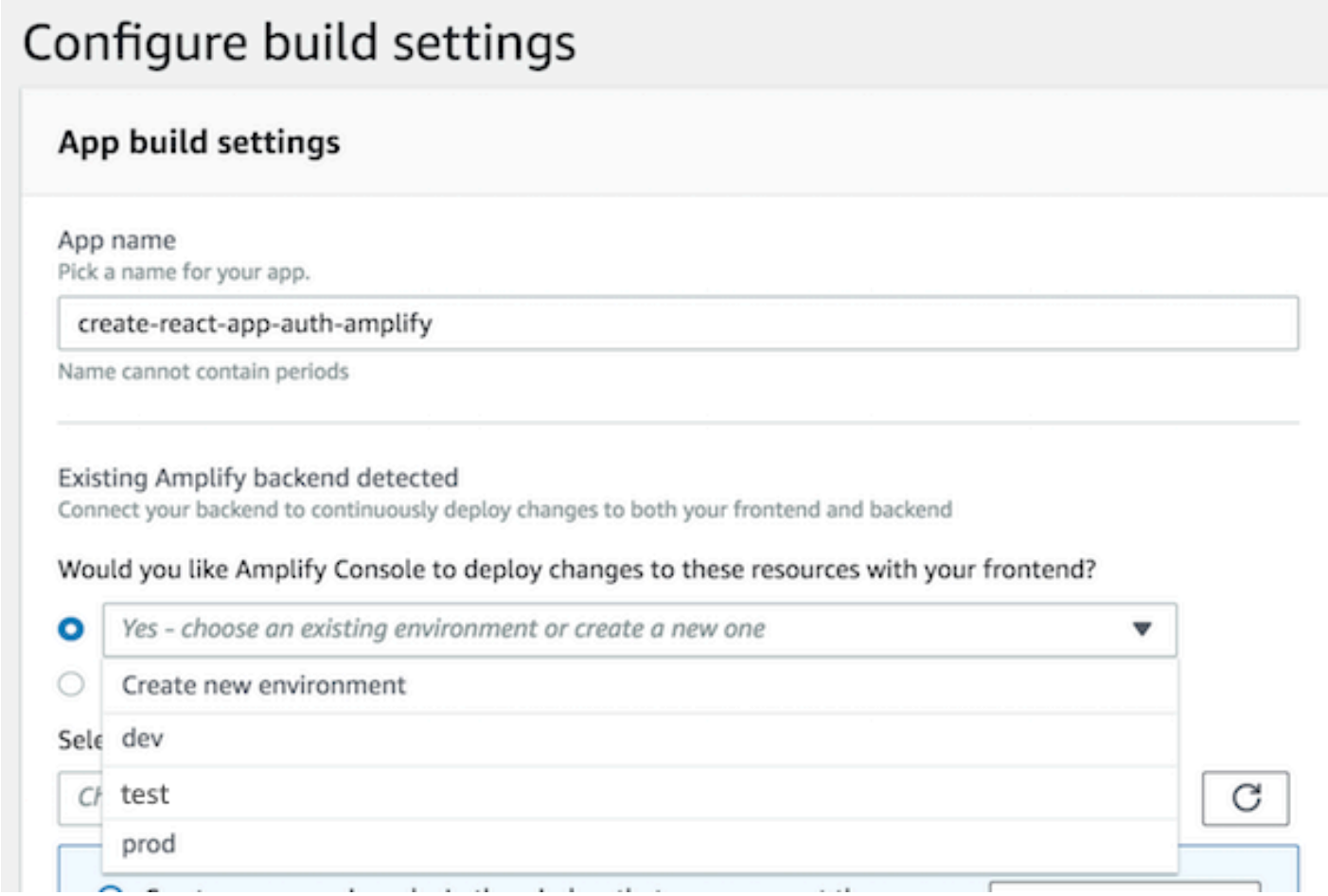
```
git commit -am 'Added dev, test, and prod environments'
git push origin main
```

5. Visita Amplify nel per vedere AWS Management Console il tuo ambiente di backend attuale. Sali di livello dal breadcrumb per visualizzare un elenco di tutti gli ambienti di backend creati nella scheda Ambienti di backend.

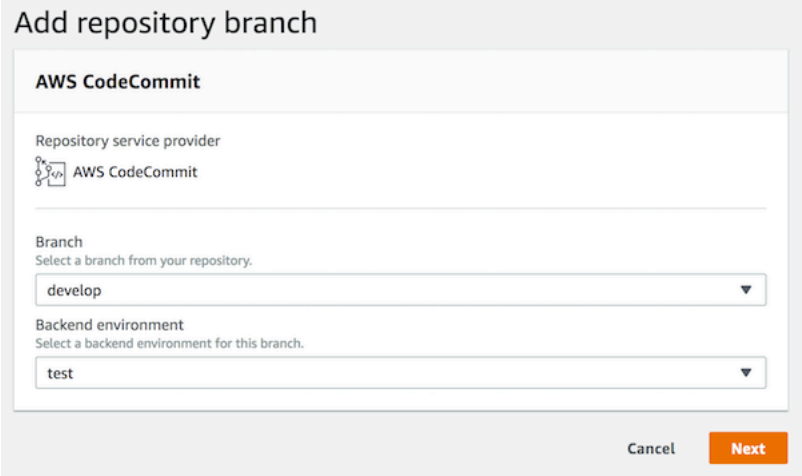


6. Passa alla scheda Ambienti frontend e collega il provider del repository e la filiale principale.
7. Nella pagina delle impostazioni di compilazione, seleziona un ambiente di backend esistente per configurare la distribuzione continua con il ramo principale. Scegli prod dall'elenco e concedi il ruolo di servizio ad Amplify. Scegliere Save and deploy (Salva e distribuisci). Una volta

completata la build, sarà disponibile una distribuzione della filiale principale all'indirizzo. https://main.appid.amplifyapp.com



8. Connect develop branch in Amplify (supponiamo che develop e main branch siano gli stessi a questo punto). Scegliere l'ambiente di back-end test.



9. Amplify è ora configurato. È possibile iniziare a lavorare su nuove funzionalità in un ramo feature. Aggiungere funzionalità di back-end utilizzando l'ambiente di back-end dev dalla workstation locale.

```
git checkout -b newinternet
amplify env checkout dev
amplify add api
...
amplify push
```

10. Dopo avere terminato di lavorare sulla funzionalità, eseguire il commit del codice e creare una richiesta di pull da rivedere internamente.

```
git commit -am 'Decentralized internet v0.1'
git push origin newinternet
```

11. Per vedere in anteprima come saranno le modifiche, vai alla console Amplify e collega il tuo ramo di funzionalità. Nota: se lo hai AWS CLI installato sul tuo sistema (non l'Amplify CLI), puoi collegare un ramo direttamente dal tuo terminale. È possibile reperire il proprio appid accedendo a App settings > General > AppARN (Impostazioni applicazione > Generali > ARNapp): arn:aws:amplify:<region>:<region>:apps/<appid>

```
aws amplify create-branch --app-id <appid> --branch-name <branchname>
aws amplify start-job --app-id <appid> --branch-name <branchname> --job-type RELEASE
```

12. La tua funzionalità sarà accessibile da condividerla con i tuoi https://newinternet.appid.amplifyapp.comcompagni di squadra. Se tutto appare corretto, unire PR al ramo develop.

```
git checkout develop
git merge newinternet
git push
```

13. Questo darà il via a una build che aggiornerà il backend e il frontend in Amplify con una distribuzione in filiale presso. https://dev.appid.amplifyapp.com È possibile condividere questo link con parti interessate interne affinché possano esaminare la nuova funzionalità.

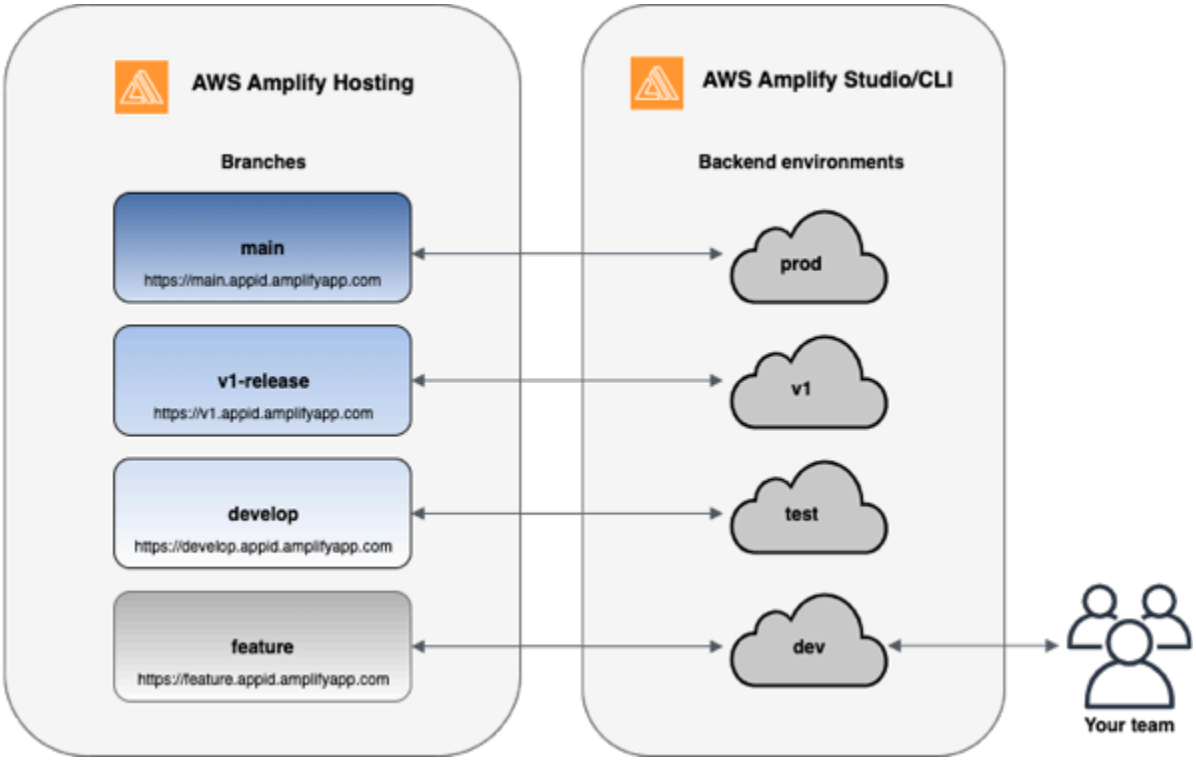
14. Elimina il tuo ramo di funzionalità da Git, Amplify e rimuovi l'ambiente di backend dal cloud (puoi sempre crearne uno nuovo basandoti su «amplify env checkout prod» ed eseguendo «amplify env add»).

```
git push origin --delete newinternet
aws amplify delete-branch --app-id <appid> --branch-name <branchname>
amplify env remove dev
```

GitFlow flusso di lavoro

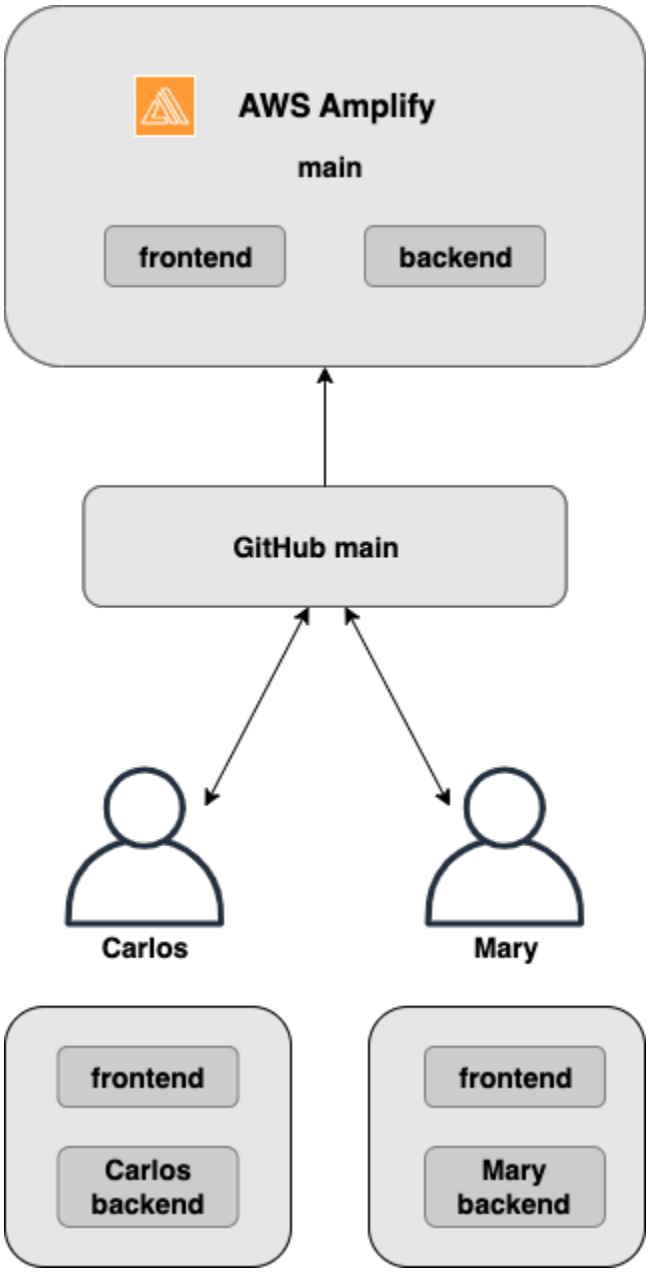
GitFlow utilizza due rami per registrare la cronologia del progetto. Il ramo principale tiene traccia solo del codice di rilascio e il ramo di sviluppo viene utilizzato come ramo di integrazione per nuove funzionalità. GitFlow semplifica lo sviluppo parallelo isolando il nuovo sviluppo dal lavoro completato. Il nuovo sviluppo (come le funzionalità e le correzioni di bug non urgenti) viene svolto nei rami feature. Quando lo sviluppatore è sicuro che il codice sia pronto per il rilascio, il ramo feature viene ricollegato al ramo develop delle integrazioni. Gli unici commit verso il ramo principale sono le fusioni tra le branch release e le branch hotfix (per correggere i bug di emergenza).

Il diagramma seguente mostra una configurazione consigliata con. GitFlow Puoi seguire lo stesso processo descritto nella sezione relativa al flusso di lavoro del ramo feature di cui sopra.



Per sandbox sviluppatore

- Ogni sviluppatore di un team crea un ambiente sandbox nel cloud che è separato dal computer locale. Ciò consente agli sviluppatori di lavorare in modo isolato gli uni dagli altri senza sovrascrivere le modifiche degli altri membri del team.
- Ogni filiale di Amplify ha il proprio backend. Ciò garantisce che Amplify utilizzi il repository Git come unica fonte di verità da cui distribuire le modifiche, anziché affidarsi agli sviluppatori del team per inviare manualmente il backend o il front-end alla produzione dai computer locali.



1. Installare l'interfaccia a riga di comando di Amplify per inizializzare un nuovo progetto Amplify.

```
npm install -g @aws-amplify/cli
```

2. Inizializza un ambiente di backend Mary per il tuo progetto. Se non hai un progetto, creane uno usando strumenti di bootstrap come create-react-app o Gatsby.

```
cd next-unicorn
amplify init
 ? Do you want to use an existing environment? (Y/n): n
 ? Enter a name for the environment: mary
...
amplify push
```

3. Invia il codice a un repository Git di tua scelta (in questo esempio supponiamo che tu abbia eseguito il push su main.

```
git commit -am 'Added mary sandbox'
git push origin main
```

4. Connect il repo > main ad Amplify.
5. La console Amplify rileverà gli ambienti di backend creati dalla CLI Amplify. Scegli Crea nuovo ambiente dal menu a discesa e concedi il ruolo di servizio ad Amplify. Scegliere Save and deploy (Salva e distribuisci). Una volta completata la build, avrai a disposizione una distribuzione della filiale principale https://main.appid.amplifyapp.comcon un nuovo ambiente di backend collegato alla filiale.
6. Connetti il ramo di sviluppo in Amplify (supponiamo che develop e main branch siano gli stessi a questo punto) e scegli Crea

Implementazioni di feature branch basate su pattern

Le distribuzioni di filiali basate su pattern consentono di distribuire automaticamente le filiali che corrispondono a uno schema specifico su Amplify. I team di prodotto che utilizzano feature branch o GitFlow flussi di lavoro per le loro versioni possono ora definire modelli come **`release**`** distribuire automaticamente i rami Git che iniziano con «release» su un URL condivisibile.

1. Scegli le impostazioni dell'app, quindi le impostazioni di Branch.
2. Nella pagina delle impostazioni di Branch, scegli Modifica.

3. Seleziona Rilevamento automatico del ramo per connettere automaticamente i rami ad Amplify che corrispondono a un set di pattern.
4. Nella casella Rilevamento automatico di Branch - pattern, inserisci i modelli per la distribuzione automatica dei rami.
   - **`*`**— Implementa tutte le filiali del tuo repository.
   - **`release*`**— Distribuisce tutti i rami che iniziano con la parola «rilascio».
   - **`release*/`**— Distribuisce tutti i rami che corrispondono a uno schema 'release /'.
   - Specificate più modelli in un elenco separato da virgole. Ad esempio **`release*, feature*`**.
5. Imposta la protezione automatica con password per tutte le filiali che vengono create automaticamente selezionando il controllo degli accessi con rilevamento automatico di Branch.
6. Per le applicazioni di prima generazione create con un backend Amplify, puoi scegliere di creare un nuovo ambiente per ogni filiale connessa o indirizzare tutte le filiali verso un backend esistente.
7. Seleziona Salva.

Implementazioni di feature branch basate su pattern per un'app connessa a un dominio personalizzato

Puoi utilizzare distribuzioni di branch basate su funzionalità basate su pattern per un'app connessa a un dominio personalizzato Amazon Route 53.

- Per istruzioni sulla configurazione di distribuzioni di feature branch basate su pattern, consulta Configurazione di sottodomini automatici per un dominio personalizzato Amazon Route 53
- Per istruzioni su come collegare un'app Amplify a un dominio personalizzato gestito in Route 53, vedi Aggiunta di un dominio personalizzato gestito da Amazon Route 53
- Per ulteriori informazioni sull'uso di Route 53, consulta What is Amazon Route 53.

Generazione automatica in fase di compilazione della configurazione Amplify (solo app di prima generazione)

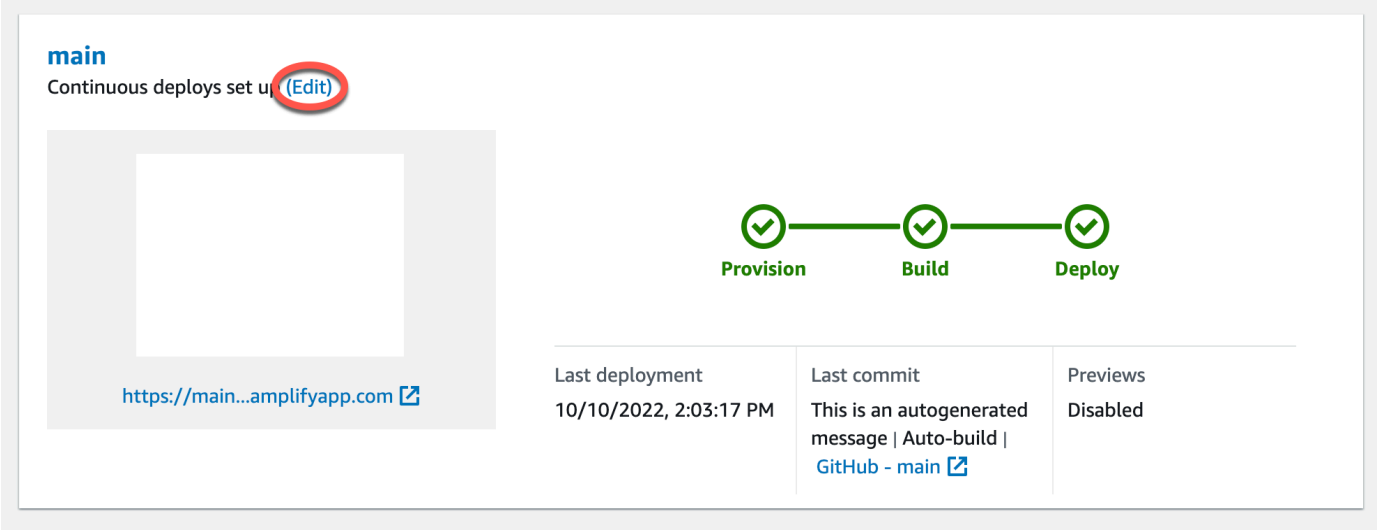
> **Note**
>
> Le informazioni in questa sezione si riferiscono solo alle app di prima generazione.
> Se desideri implementare automaticamente le modifiche al codice dell'infrastruttura e

dell'applicazione dai rami delle funzionalità per un'app di seconda generazione, consulta le implementazioni delle filiali Fullstack nei documenti di Amplify

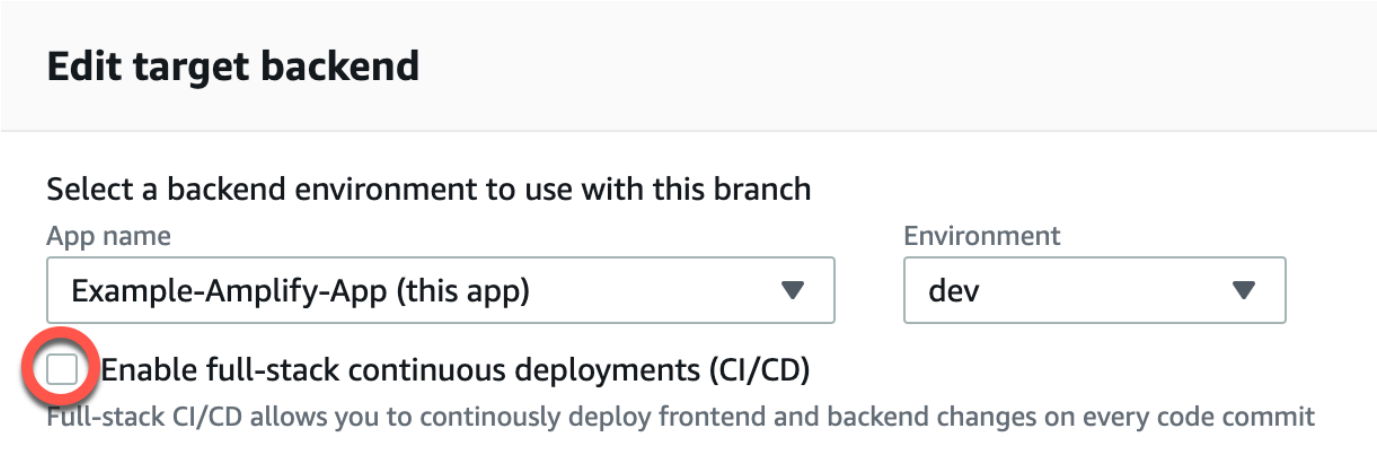
Amplify supporta la generazione automatica in fase di compilazione del file di configurazione Amplify per le app `aws-exports.js` di prima generazione. Disattivando le implementazioni CI/CD complete dello stack, consenti alla tua app di generare automaticamente il file e assicuri che non vengano apportati aggiornamenti al backend in fase di compilazione. `aws-exports.js`

Da **`aws-exports.js`** generare automaticamente in fase di compilazione

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app da modificare.
3. Scegli la scheda Ambienti di hosting.
4. Individua il ramo da modificare e scegli Modifica.



5. Nella pagina Modifica backend di destinazione, deseleziona Abilita le distribuzioni continue a stack completo (CI/CD) per disattivare la CI/CD full-stack per questo backend.



6. Seleziona un ruolo di servizio esistente per concedere ad Amplify le autorizzazioni necessarie per apportare modifiche al backend dell'app. Se devi creare un ruolo di servizio, scegli Crea nuovo ruolo. Per ulteriori informazioni sulla creazione di un ruolo del servizio, consulta Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend.
7. Seleziona Salva. Amplify applica queste modifiche la prossima volta che crei l'app.

Build di backend condizionali (solo app di prima generazione)

> **Note**
>
> Le informazioni in questa sezione si riferiscono solo alle app di prima generazione. Amplify Gen 2 introduce TypeScript un'esperienza di sviluppo basata sul codice. Pertanto, questa funzionalità non è necessaria per i backend di seconda generazione.

Amplify supporta build di backend condizionali su tutte le filiali in un'app di prima generazione. Per configurare build di backend condizionali, imposta la variabile di ambiente su. `AMPLIFY_DIFF_BACKEND` `true` L'abilitazione delle build condizionali di backend contribuirà a velocizzare le build in cui le modifiche vengono apportate solo al frontend.

Quando abiliti le build di backend basate su diff, all'inizio di ogni build, Amplify tenta di eseguire un diff nella cartella del tuo repository. `amplify` Se Amplify non rileva alcuna differenza, salta la fase di creazione del backend e non aggiorna le risorse del backend. Se il progetto non ha una `amplify` cartella nel repository, Amplify ignora il valore della variabile di ambiente. `AMPLIFY_DIFF_BACKEND`

Per istruzioni sull'impostazione della variabile di `AMPLIFY_DIFF_BACKEND` ambiente, consulta. Configurazione di build di backend basate su diff per un'app di prima generazione

Se al momento sono stati specificati comandi personalizzati nelle impostazioni di compilazione della fase di backend, le build condizionali di backend non funzioneranno. Se desideri che questi comandi personalizzati vengano eseguiti, devi spostarli nella fase di frontend delle impostazioni di build nel file dell'app. `amplify.yml` Per ulteriori informazioni sull'aggiornamento del `amplify.yml` file, consultaRiferimento alle specifiche della build.

Usa i backend Amplify tra le app (solo app di prima generazione)

> **Note**
>
> Le informazioni in questa sezione si riferiscono solo alle app di prima generazione. Se desideri condividere risorse di backend per un'app di seconda generazione, consulta Condividere risorse tra filiali nei documenti Amplify

Amplify ti consente di riutilizzare gli ambienti di backend esistenti in tutte le tue app di prima generazione in una determinata regione. Puoi farlo quando crei una nuova app, connetti una nuova filiale a un'app esistente o aggiorni un frontend esistente in modo che punti a un ambiente di backend diverso.

Riutilizza i backend quando crei una nuova app

Per riutilizzare un backend durante la creazione di una nuova app Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Per creare un nuovo backend da utilizzare per questo esempio, procedi come segue:
   a. Nel riquadro di navigazione, scegli Tutte le app.
   b. Scegli Nuova app, Crea un'app.
   c. Inserisci un nome per la tua app, ad esempio**`Example-Amplify-App`**.
   d. Scegli Conferma distribuzione.
3. Per connettere un frontend al nuovo backend, scegli la scheda Ambienti di hosting.
4. Scegli il tuo provider git, quindi scegli Connect branch.

5. Nella pagina Aggiungi ramo del repository, per Archivi aggiornati di recente, scegli il nome del repository. Per Branch, seleziona il ramo dal tuo repository per connetterti.
6. Nella pagina Build settings, procedi come segue:
   a. Per il nome dell'app, seleziona l'app da utilizzare per aggiungere un ambiente di backend. Puoi scegliere l'app corrente o qualsiasi altra app nella regione corrente.
   b. Per Ambiente, seleziona il nome dell'ambiente di backend da aggiungere. È possibile utilizzare un ambiente esistente o crearne uno nuovo.
   c. Per impostazione predefinita, lo stack completo CI/CD is turned off. Turning off full-stack CI/CD fa sì che l'app venga eseguita in modalità pull only. In fase di compilazione, Amplify genererà automaticamente solo `aws-exports.js` il file, senza modificare l'ambiente di backend.
   d. Seleziona un ruolo di servizio esistente per concedere ad Amplify le autorizzazioni necessarie per apportare modifiche al backend dell'app. Se devi creare un ruolo di servizio, scegli Crea nuovo ruolo. Per ulteriori informazioni sulla creazione di un ruolo del servizio, consulta Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend.
   e. Scegli Next (Successivo).
7. Scegliere Save and deploy (Salva e distribuisci).

Riutilizza i backend quando connetti una filiale a un'app esistente

Per riutilizzare un backend quando si collega una filiale a un'app Amplify esistente

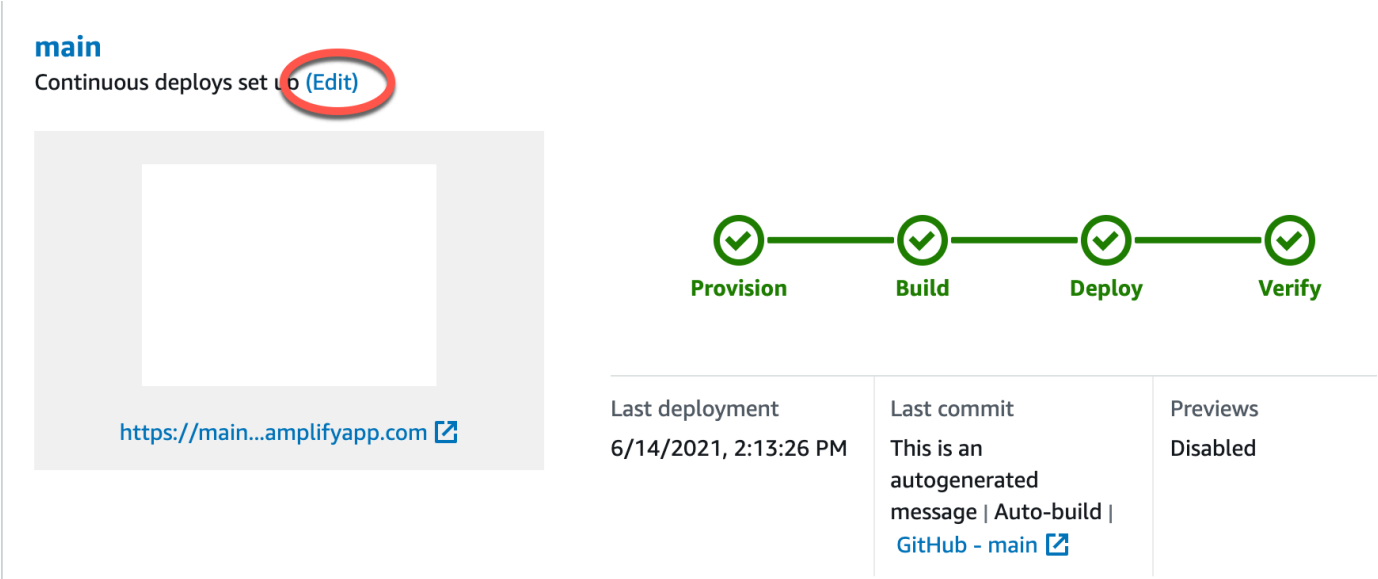
1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app a cui connettere una nuova filiale.
3. Nel riquadro di navigazione, scegli Impostazioni app, Generali.
4. Nella sezione Filiali, scegli Connetti una filiale.
5. Nella pagina Aggiungi ramo del repository, per Branch, seleziona il ramo dal tuo repository per connetterti.
6. Per il nome dell'app, seleziona l'app da utilizzare per aggiungere un ambiente di backend. Puoi scegliere l'app corrente o qualsiasi altra app nella regione corrente.
7. Per Ambiente, seleziona il nome dell'ambiente di backend da aggiungere. È possibile utilizzare un ambiente esistente o crearne uno nuovo.

8. Se devi configurare un ruolo di servizio per concedere ad Amplify le autorizzazioni necessarie per apportare modifiche al backend dell'app, la console ti chiederà di eseguire questa operazione. Per ulteriori informazioni sulla creazione di un ruolo del servizio, consulta Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend.
9. Per impostazione predefinita, lo stack completo CI/CD is turned off. Turning off full-stack CI/CD fa sì che l'app venga eseguita in modalità pull only. In fase di compilazione, Amplify genererà automaticamente solo `aws-exports.js` il file, senza modificare l'ambiente di backend.
10. Scegli Next (Successivo).
11. Scegliere Save and deploy (Salva e distribuisci).

Modifica un frontend esistente in modo che punti a un backend diverso

Per modificare un frontend, l'app Amplify in modo che punti a un backend diverso

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui modificare il backend.
3. Scegli la scheda Ambienti di hosting.
4. Individua il ramo da modificare e scegli Modifica.



5. Nella pagina Seleziona un ambiente di backend da usare con questo ramo, per Nome app, seleziona l'app di frontend per cui desideri modificare l'ambiente di backend. Puoi scegliere l'app corrente o qualsiasi altra app nella regione corrente.
6. Per Ambiente di backend, seleziona il nome dell'ambiente di backend da aggiungere.

7. Per impostazione predefinita, lo stack completo CI/CD is enabled. Uncheck this option to turn off full-stack CI/CD for this backend. Turning off full-stack CI/CD fa sì che l'app venga eseguita in modalità pull only. In fase di compilazione, Amplify genererà automaticamente solo `aws-exports.js` il file, senza modificare l'ambiente di backend.
8. Seleziona Salva. Amplify applica queste modifiche la prossima volta che crei l'app.

Creazione di un backend per un'applicazione

Con AWS Amplify puoi creare un'applicazione full stack con dati, autenticazione, archiviazione e hosting frontend da distribuire su. AWS

AWS Amplify Gen 2 introduce TypeScript un'esperienza di sviluppo basata sul codice per la definizione dei backend. Per sapere come usare Amplify Gen 2 per creare e connettere un backend alla tua app, consulta Build & connect backend nei documenti Amplify.

Se stai cercando la documentazione per creare un backend per un'app di prima generazione, utilizzando la CLI e Amplify Studio, consulta il backend Build & connect nei documenti Amplify di prima generazione.

Argomenti

- Crea un backend per un'app di seconda generazione
- Crea un backend per un'app di prima generazione

Crea un backend per un'app di seconda generazione

Per un tutorial che ti guida attraverso i passaggi per creare un'applicazione fullstack Amplify Gen 2 con TypeScript un backend basato, consulta Guida introduttiva nei documenti Amplify.

Crea un backend per un'app di prima generazione

In questo tutorial, configurerai un flusso di lavoro CI/CD completo con Amplify. Distribuirai un'app frontend su Amplify Hosting. Quindi creerai un backend usando Amplify Studio. Infine, collegherai il backend cloud all'app frontend.

Prerequisiti

Prima di iniziare questo tutorial, completa i seguenti prerequisiti.

Registrati per un Account AWS

Se non sei già un AWS cliente, devi crearne uno Account AWS seguendo le istruzioni online. La registrazione ti consente di accedere ad Amplify e AWS ad altri servizi che puoi utilizzare con la tua applicazione.

Crea un repository Git

Amplify GitHub supporta, GitLab Bitbucket e. AWS CodeCommit Invia la tua applicazione al tuo repository Git.

Installazione dell'interfaccia a riga di comando (CLI) Amplify

Per istruzioni, consulta Installare la CLI Amplify nella documentazione di Amplify Framework.

Fase 1: Implementazione di un frontend

Se hai un'app frontend esistente in un repository git che desideri utilizzare per questo esempio, puoi procedere con le istruzioni per la distribuzione di un'app frontend.

Se devi creare una nuova app frontend da utilizzare per questo esempio, puoi seguire le istruzioni Create React App nella documentazione di Create React App.

Per distribuire un'app frontend

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella pagina Tutte le app, scegli Nuova app, quindi Ospita app web nell'angolo in alto a destra.
3. Seleziona il tuo fornitore GitHub, Bitbucket o di AWS CodeCommit repository GitLab, quindi scegli Continua.
4. Amplify autorizza l'accesso al tuo repository git. Per gli GitHub archivi, Amplify ora utilizza la funzione Apps per autorizzare GitHub l'accesso ad Amplify.

   Per ulteriori informazioni sull'installazione e l'autorizzazione dell'App, consulta. GitHub Configurazione dell'accesso Amplify ai repository GitHub
5. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Nell'elenco dei repository aggiornati di recente, seleziona il nome del repository da connettere.
   b. Nell'elenco Branch, seleziona il nome del ramo del repository da connettere.
   c. Scegli Next (Successivo).
6. Nella pagina Configura le impostazioni di build, scegli Avanti.
7. Nella pagina Revisione, scegli Salva e distribuisci. Una volta completata la distribuzione, puoi visualizzare l'app nel dominio `amplifyapp.com` predefinito.

> **Note**
>
> Per aumentare la sicurezza delle tue applicazioni Amplify, il dominio amplifyapp.com è registrato nella Public Suffix List (PSL). Per una maggiore sicurezza, ti consigliamo di utilizzare i cookie con un `__Host-` prefisso se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Amplify. Questa pratica ti aiuterà a difendere il tuo dominio dai tentativi CSRF (cross-site request forgery). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Fase 2: Creare un backend

Ora che hai distribuito un'app frontend su Amplify Hosting, puoi creare un backend. Utilizza le seguenti istruzioni per creare un backend con un database semplice e un endpoint API GraphQL.

Per creare un backend

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella pagina Tutte le app, seleziona l'app che hai creato nel passaggio 1.
3. Nella home page dell'app, scegli la scheda Ambienti di backend, quindi scegli Inizia. Questo avvia il processo di configurazione per un ambiente di staging predefinito.
4. Al termine della configurazione, scegli Launch Studio per accedere all'ambiente di backend di staging in Amplify Studio.

Amplify Studio è un'interfaccia visiva per creare e gestire il backend e accelerare lo sviluppo dell'interfaccia utente frontend. Per ulteriori informazioni su Amplify Studio, consulta la documentazione di Amplify Studio.

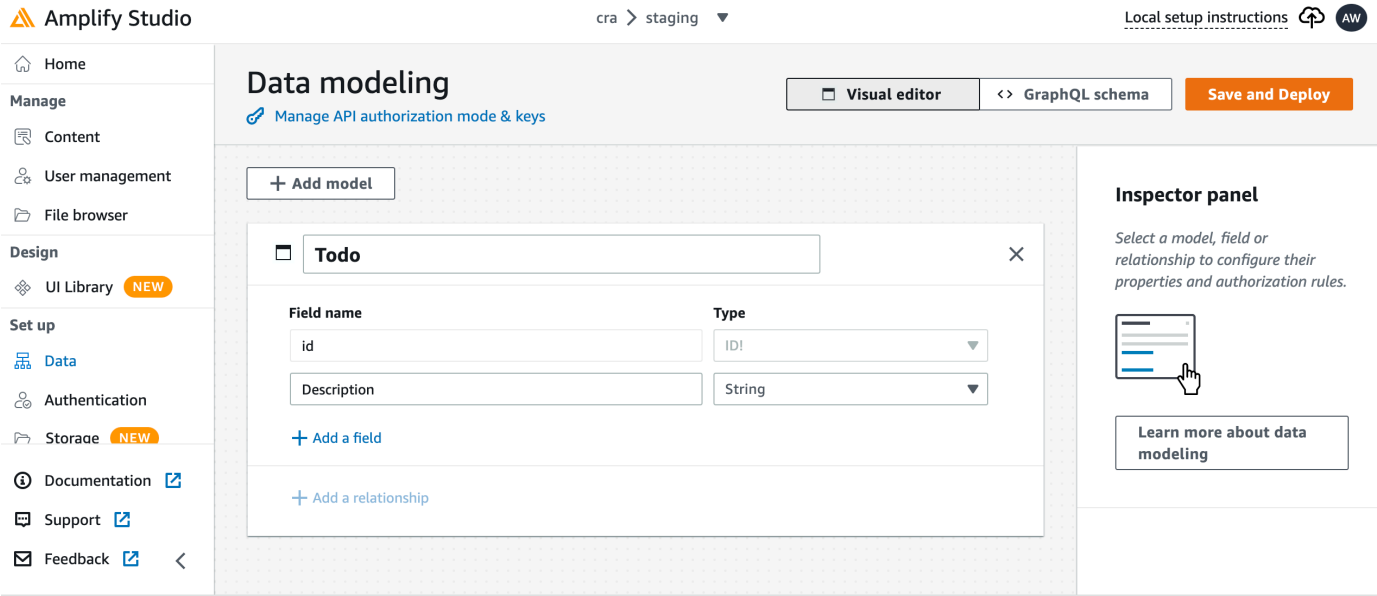
Usa le seguenti istruzioni per creare un database semplice utilizzando l'interfaccia visual backend builder di Amplify Studio.

Crea un modello di dati

1. Nella home page dell'ambiente di staging dell'app, scegli Crea modello di dati. Questo apre il designer del modello di dati.
2. Nella pagina di modellazione dei dati, scegli Aggiungi modello.
3. Per il titolo, inserisci**`Todo`**.

4. Scegli Aggiungi un campo.
5. Per Nome campo, inserisci**`Description`**.

   La schermata seguente è un esempio di come apparirà il modello di dati nel designer.



6. Scegli Salva e distribuisci.
7. Torna alla console Amplify Hosting e la distribuzione dell'ambiente di staging sarà in corso.

Durante l'implementazione, Amplify Studio crea tutte le risorse AWS necessarie nel backend, tra cui un'API AWS AppSync GraphQL per accedere ai dati e una tabella Amazon DynamoDB per ospitare gli elementi Todo. Amplify AWS CloudFormation utilizza per implementare il backend, il che consente di archiviare la definizione del backend come. infrastructure-as-code

Passaggio 3: Connect il backend al frontend

Ora che hai distribuito un frontend e creato un backend cloud che contiene un modello di dati, devi connetterli. Usa le seguenti istruzioni per trasferire la definizione del backend al tuo progetto di app locale con la CLI Amplify.

Per connettere un backend cloud a un frontend locale

1. Apri una finestra di terminale e vai alla directory principale del tuo progetto locale.
2. Esegui il seguente comando nella finestra del terminale, sostituendo il testo rosso con l'ID univoco dell'app e il nome dell'ambiente di backend per il tuo progetto.

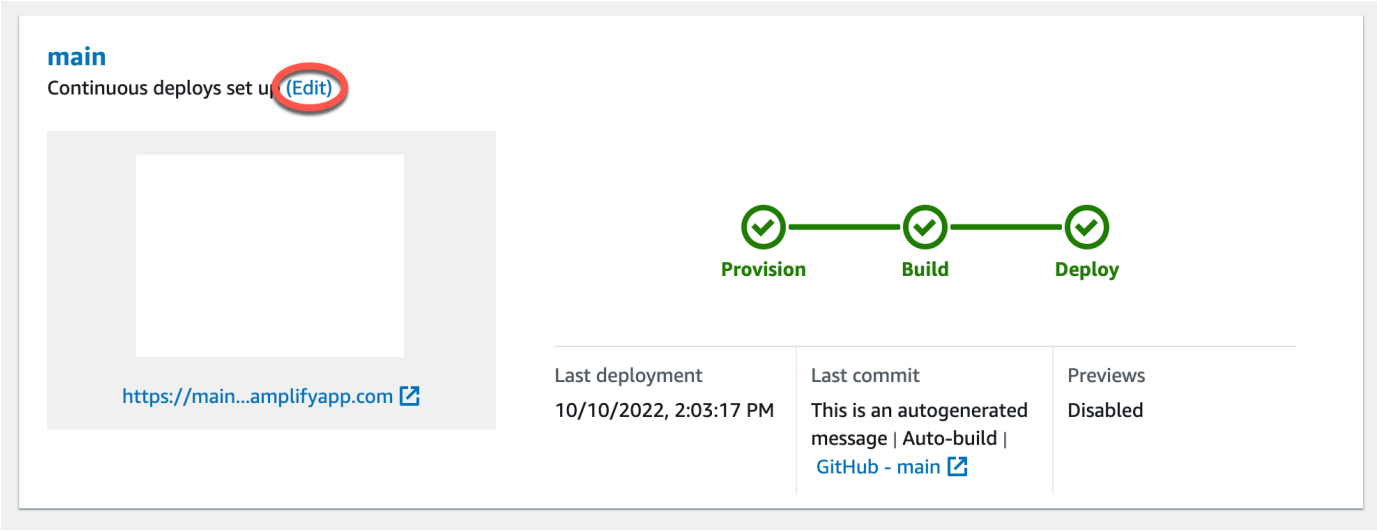
```
amplify pull --appId abcd1234 --envName staging
```

3. Segui le istruzioni nella finestra del terminale per completare la configurazione del progetto.

Ora puoi configurare il processo di compilazione per aggiungere il backend al flusso di lavoro di distribuzione continua. Usa le seguenti istruzioni per connettere un ramo frontend con un backend nella console Amplify Hosting.

Per connettere un ramo di app frontend e un backend cloud

1. Nella home page dell'app, scegli la scheda Ambienti di hosting.
2. Individua il ramo principale e scegli Modifica.



3. Nella finestra Modifica backend di destinazione, per Ambiente, seleziona il nome del backend da connettere. In questo esempio, scegli il backend di staging che hai creato nel passaggio 2.

   Per impostazione predefinita, lo stack completo CI/CD is enabled. Uncheck this option to turn off full-stack CI/CD for this backend. Turning off full-stack CI/CD fa sì che l'app venga eseguita in modalità pull only. In fase di compilazione, Amplify genererà automaticamente solo `aws-exports.js` il file, senza modificare l'ambiente di backend.

4. Successivamente, devi impostare un ruolo di servizio per concedere ad Amplify le autorizzazioni necessarie per apportare modifiche al backend dell'app. Puoi utilizzare un ruolo di servizio esistente o crearne uno nuovo. Per istruzioni, consulta Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend.

5. Dopo aver aggiunto un ruolo di servizio, torna alla finestra Modifica backend di destinazione e scegli Salva.
6. Per completare la connessione del backend di staging al ramo principale dell'app frontend, esegui una nuova build del progetto.

   Esegui una di queste operazioni:

   - Dal tuo repository git, invia del codice per avviare una build nella console Amplify.
   - Nella console Amplify, vai alla pagina dei dettagli della build dell'app e scegli Redeploy this version.

Passaggi successivi

Configura le distribuzioni delle feature branch

Segui il nostro flusso di lavoro consigliato per configurare implementazioni di feature branch con più ambienti di backend.

Crea un'interfaccia utente frontend in Amplify Studio

Usa Studio per creare l'interfaccia utente di frontend con un set di componenti dell' ready-to-useinterfaccia utente, quindi collegala al backend dell'app. Per ulteriori informazioni e tutorial, consulta la guida per l'utente di Amplify Studio nella documentazione di Amplify Framework.

Caratteristiche avanzate di implementazione

Questo capitolo tratta le funzionalità di distribuzione avanzate che migliorano il flusso di lavoro di Amplify Hosting. Queste funzionalità forniscono controlli e funzionalità aggiuntivi per aiutare i team a gestire le implementazioni in modo più efficace, garantire la qualità del codice e mantenere la sicurezza per tutto il ciclo di vita dello sviluppo.

Scopri come proteggere le tue funzionalità con l'autenticazione tramite password per limitare l'accesso a funzionalità inedite. Attiva le anteprime web per le richieste pull per rivedere le modifiche in anteprima unica URLs prima di unire il codice ai rami di produzione. Imposta end-to-end i test utilizzando il framework Cypress per catturare le regressioni prima di inviare il codice alla produzione. Sebbene la funzionalità del pulsante Deploy to Amplify non sia più disponibile, puoi comunque distribuire facilmente le applicazioni direttamente dal tuo repository utilizzando Amplify Hosting.

Argomenti

- Limitazione dell'accesso alle filiali di un'app Amplify
- Anteprime Web per le richieste pull
- Configurazione dei test end-to-end Cypress per l'applicazione Amplify
- Utilizzo del pulsante Deploy to Amplify per condividere un progetto GitHub

Limitazione dell'accesso alle filiali di un'app Amplify

Se stai lavorando su funzionalità inedite, puoi proteggere con password i rami delle funzionalità per limitare l'accesso a utenti specifici. Quando il controllo degli accessi è impostato su una filiale, agli utenti viene richiesto di inserire un nome utente e una password quando tentano di accedere all'URL della filiale.

È possibile impostare una password da applicare a una singola filiale o a livello globale a tutte le filiali connesse. Quando il controllo degli accessi è abilitato sia a livello di filiale che globale, la password a livello di filiale ha la precedenza su una password a livello globale (di applicazione).

Amplify limita le richieste non riuscite che tentano di accedere a risorse protette da password. Questo comportamento protegge le applicazioni dagli attacchi ai dizionari o da altri tentativi di lettura dei dati protetti dai controlli di accesso.

Utilizza la seguente procedura per impostare una password per limitare l'accesso alle filiali di un'app Amplify.

Per impostare le password sui rami delle funzionalità

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app su cui vuoi impostare le password del Feature Branch.
3. Nel riquadro di navigazione, scegli Hosting, quindi scegli Controllo degli accessi.
4. Nella sezione Impostazioni di controllo degli accessi, scegli Gestisci l'accesso.
5. Nella pagina Gestisci il controllo degli accessi, esegui una delle seguenti operazioni.
   - Per impostare un nome utente e una password validi per tutte le filiali connesse
     - Attiva Gestisci l'accesso per tutte le filiali. Ad esempio, se hai i rami main, dev e feature collegati, puoi applicare lo stesso nome utente e la stessa password a tutte le filiali.
   - Per impostare un nome utente e una password validi per una singola filiale
     a. Disattiva Gestisci l'accesso per tutte le filiali.
     b. Individua il ramo che vuoi gestire. Per le impostazioni di accesso, scegli Password limitata richiesta.
     c. In Nome utente, immetti un nome utente.
     d. Per Password immetti una password.
   - Scegli Save (Salva).
6. Se gestisci il controllo degli accessi per un'app renderizzata lato server (SSR), ridistribuisci l'app eseguendo una nuova build dal tuo repository Git. Questo passaggio è necessario per consentire ad Amplify di applicare le impostazioni di controllo degli accessi.

Anteprime Web per le richieste pull

Le anteprime Web offrono ai team di sviluppo e controllo qualità (QA) un modo per visualizzare in anteprima le modifiche apportate alle pull request (PRs) prima di unire il codice a un ramo di produzione o di integrazione. Le richieste pull ti consentono di comunicare agli altri le modifiche che hai inviato a una filiale in un repository. Dopo l'apertura di una pull request, puoi discutere ed esaminare le potenziali modifiche con i collaboratori e aggiungere commit di follow-up prima che le modifiche vengano unite al ramo base.

Un'anteprima web distribuisce ogni pull request inviata al tuo repository su un URL di anteprima unico, completamente diverso dall'URL utilizzato dal tuo sito principale. Per le app con ambienti di

backend forniti utilizzando l'Amplify CLI o Amplify Studio, ogni pull request (solo repository Git privati) crea un backend temporaneo che viene eliminato alla chiusura del PR.

Quando le anteprime web sono attivate per la tua app, ogni PR conta ai fini della quota Amplify di 50 filiali per app. Per evitare di superare questa quota, assicurati di chiudere la tua. PRs Per ulteriori informazioni sulle quote, consulta Quote del servizio Amplify Hosting.

> **Note**
>
> Attualmente, la variabile di `AWS_PULL_REQUEST_ID` ambiente non è disponibile quando viene utilizzata AWS CodeCommit come provider di repository.

Sicurezza dell'anteprima Web

Per motivi di sicurezza, puoi abilitare le anteprime web su tutte le app con repository privati, ma non su tutte le app con repository pubblici. Se il tuo repository Git è pubblico, puoi configurare le anteprime solo per le app che non richiedono un ruolo di servizio IAM. Ad esempio, le app con backend e le app distribuite sulla piattaforma di `WEB_COMPUTE` hosting richiedono un ruolo di servizio IAM. Pertanto, non puoi abilitare le anteprime web per questi tipi di app se il loro archivio è pubblico. Amplify applica questa restrizione per impedire a terze parti di inviare codice arbitrario da eseguire utilizzando le autorizzazioni dei ruoli IAM della tua app.

Quando le anteprime web sono abilitate per un'applicazione in un archivio pubblico, con un ruolo SSR Compute, devi gestire con attenzione le filiali che possono accedere al ruolo. Si consiglia di non utilizzare un ruolo a livello di app. Invece, dovresti assegnare un ruolo Compute a livello di filiale. Ciò consente di concedere le autorizzazioni solo alle filiali che richiedono l'accesso a risorse specifiche. Per ulteriori informazioni, consulta Aggiungere un ruolo SSR Compute per consentire l'accesso alle risorse AWS.

Abilita le anteprime web per le richieste pull

Per le app archiviate in un GitHub repository, le anteprime web utilizzano l'app Amplify per l'accesso ai repository. GitHub Se stai abilitando le anteprime web su un'app Amplify esistente che hai precedentemente distribuito da GitHub un repository OAuth utilizzando for access, devi prima migrare l'app per utilizzare l'app Amplify. GitHub Per le istruzioni sulla migrazione, Migrazione di un'OAuthapp esistente all'app Amplify GitHub consulta.

Per abilitare le anteprime web per le richieste pull

1. Scegli Hosting, quindi Anteprime.

> **Note**
>
> Le anteprime sono visibili nel menu delle impostazioni dell'app solo quando un'app è configurata per la distribuzione continua e connessa a un repository git. Per istruzioni su questo tipo di distribuzione, consulta Guida introduttiva al codice esistente.

2. Solo per i GitHub repository, procedi come segue per installare e autorizzare l'app Amplify nel tuo account GitHub :

   a. Nella finestra Installa GitHub app per abilitare le anteprime, scegli Installa app. GitHub

   b. Seleziona l' GitHub account in cui desideri configurare l'app Amplify. GitHub

   c. Si apre una pagina su GitHub.com per configurare le autorizzazioni di archiviazione per il tuo account.

   d. Esegui una di queste operazioni:

      - Per applicare l'installazione a tutti gli archivi, scegli Tutti gli archivi.
      - Per limitare l'installazione ai repository specifici selezionati, scegli Seleziona solo i repository. Assicurati di includere il repository per l'app per cui stai abilitando le anteprime web nei repository selezionati.

   e. Seleziona Salva

3. Dopo aver abilitato le anteprime per il tuo repository, torna alla console Amplify per abilitare le anteprime per rami specifici. Nella pagina Anteprime, seleziona un ramo dall'elenco e scegli Modifica impostazioni.

4. Nella pagina Gestisci le impostazioni di anteprima, attiva le anteprime delle richieste Pull. Quindi scegli Conferma.

5. Per le applicazioni fullstack, eseguire una delle seguenti operazioni:

   - Scegli, crea un nuovo ambiente di backend per ogni Pull Request. Questa opzione consente di testare le modifiche senza influire sulla produzione.
   - Scegli Indirizza tutte le richieste Pull per questo ramo a un ambiente esistente.

6. Scegli Conferma.

La prossima volta che invii una pull request per la filiale, Amplify crea e distribuisce il tuo PR su un URL di anteprima. Dopo la chiusura della pull request, l'URL di anteprima viene eliminato e qualsiasi ambiente di backend temporaneo collegato alla pull request viene eliminato. Solo per i GitHub repository, puoi accedere a un'anteprima dell'URL direttamente dalla pull request del tuo GitHub account.

Accesso all'anteprima Web con sottodomini

Le anteprime Web per le richieste pull sono accessibili con i sottodomini di un'app Amplify connessa a un dominio personalizzato gestito da Amazon Route 53. Quando la pull request viene chiusa, i rami e i sottodomini associati alla pull request vengono eliminati automaticamente. Questo è il comportamento predefinito per le anteprime web dopo aver configurato le distribuzioni di feature branch basate su pattern per la tua app. Per istruzioni sull'impostazione di sottodomini automatici, consulta. Configurazione di sottodomini automatici per un dominio personalizzato Amazon Route 53

Configurazione dei test end-to-end Cypress per l'applicazione Amplify

Puoi eseguire test end-to-end (E2E) nella fase di test dell'app Amplify per catturare le regressioni prima di inviare il codice alla produzione. La fase di test può essere configurata nella specifica di build YAML. Attualmente, puoi eseguire solo il framework di test Cypress durante una build.

Cypress è un framework di test JavaScript basato che consente di eseguire test E2E su un browser. Per un tutorial che dimostra come configurare i test E2E, consulta il post sul blog Running end-to-end Cypress tests for your fullstack CI/CD deployment with Amplify.

Aggiungere test Cypress a un'applicazione Amplify esistente

Puoi aggiungere test Cypress a un'app esistente aggiornando le impostazioni di build dell'app nella console Amplify. La specifica di build YAML contiene una raccolta di comandi di compilazione e impostazioni correlate che Amplify utilizza per eseguire la build. Usa questo `test` passaggio per eseguire qualsiasi comando di test in fase di compilazione. Per i test E2E, Amplify Hosting offre un'integrazione più profonda con Cypress che consente di generare un report dell'interfaccia utente per i test.

Di seguito sono elencate le impostazioni del test e il modo in cui vengono utilizzate.

Pretest

Installa le dipendenze necessarie per eseguire i test Cypress. Amplify Hosting utilizza mochawesome per generare un rapporto per visualizzare i risultati dei test e attendere la configurazione del server localhost durante la compilazione.

test

Esegui i comandi cypress per eseguire test utilizzando mochawesome.

PostTest

Il report mochawesome viene generato dall'output JSON. Nota che se stai usando Yarn, devi eseguire questo comando in modalità silenziosa per generare il report mochawesome. Per Yarn, puoi utilizzare il seguente comando.

```
yarn run --silent mochawesome-merge cypress/report/mochawesome-report/
mochawesome*.json > cypress/report/mochawesome.json
```

Artefatti > BaseDirectory

La directory da cui vengono eseguiti i test.

artefatti> configFilePath

I dati del rapporto di test generato.

artefatti > file

Gli artefatti generati (schermate e video) sono disponibili per il download.

Il seguente estratto di esempio da un `amplify.yml` file di specifiche di build mostra come aggiungere i test Cypress alla tua app.

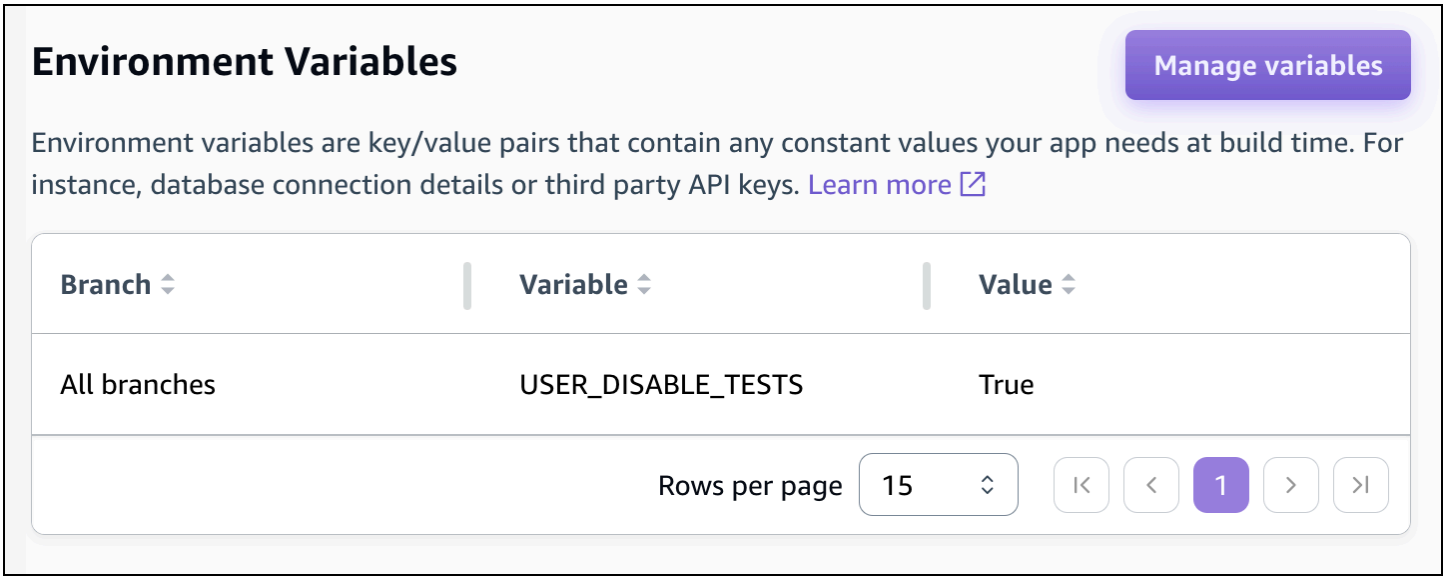
```
test:
  phases:
    preTest:
      commands:
        - npm ci
        - npm install -g pm2
        - npm install -g wait-on
        - npm install mocha mochawesome mochawesome-merge mochawesome-report-generator
        - pm2 start npm -- start
        - wait-on http://localhost:3000
```

```
    test:
      commands:
        - 'npx cypress run --reporter mochawesome --reporter-options
 "reportDir=cypress/report/mochawesome-
report,overwrite=false,html=false,json=true,timestamp=mmddyyyy_HHMMss"'
    postTest:
      commands:
        - npx mochawesome-merge cypress/report/mochawesome-report/mochawesome*.json >
 cypress/report/mochawesome.json
        - pm2 kill
  artifacts:
    baseDirectory: cypress
    configFilePath: '**/mochawesome.json'
    files:
      - '**/*.png'
      - '**/*.mp4'
```

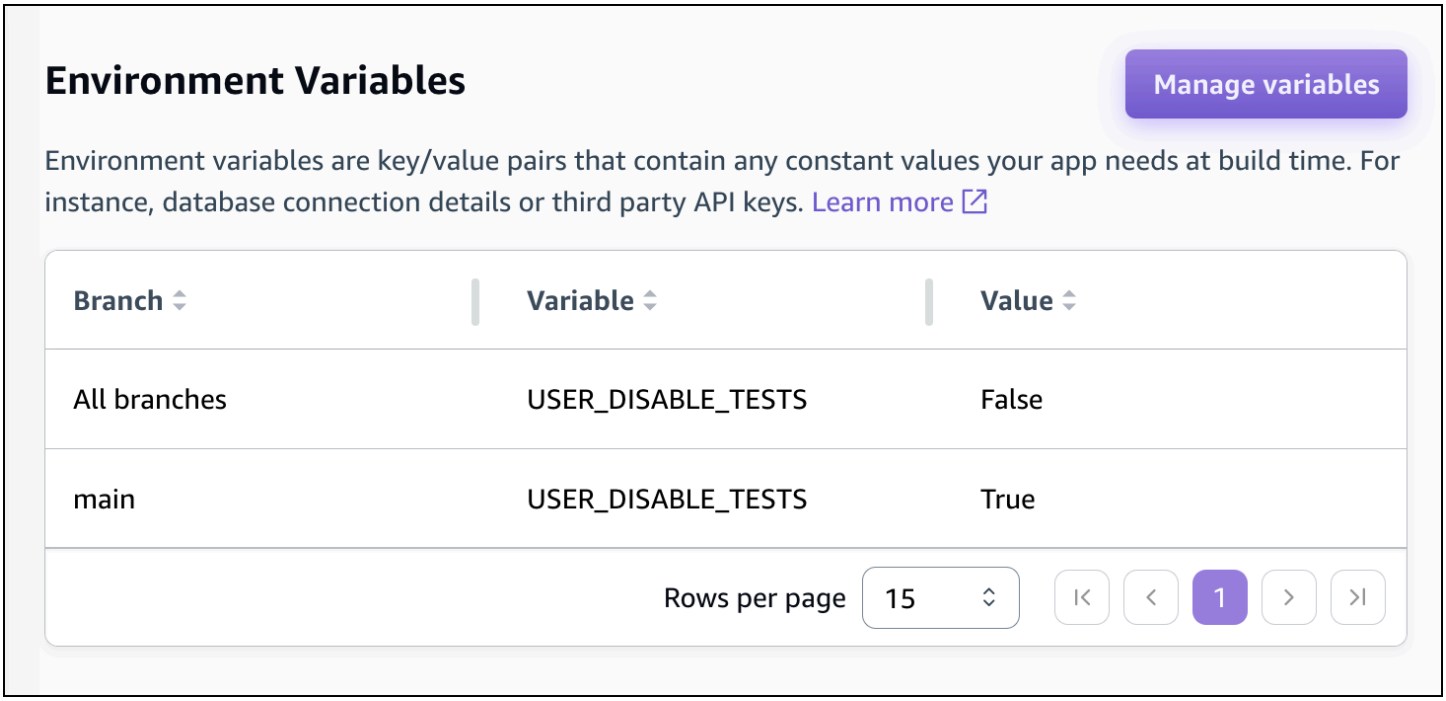
Disattivazione dei test per un'applicazione o un ramo Amplify

Dopo aver aggiunto la configurazione di test alle impostazioni di `amplify.yml` build, il `test` passaggio viene eseguito per ogni build, su ogni ramo. Se desideri disabilitare globalmente l'esecuzione dei test o eseguire solo test per rami specifici, puoi utilizzare la variabile di USER_DISABLE_TESTS ambiente senza modificare le impostazioni di build.

Per disabilitare globalmente i test per tutti i rami, aggiungi la variabile di USER_DISABLE_TESTS ambiente con un valore pari a `true` for all branch. La schermata seguente mostra la sezione Variabili di ambiente nella console Amplify con i test disabilitati per tutte le filiali.



Per disabilitare i test per un ramo specifico, aggiungi la variabile di USER_DISABLE_TESTS ambiente con un valore di `false` for all branch, quindi aggiungi un override per ogni ramo che desideri disabilitare con un valore di. `true` Nella schermata seguente, i test sono disabilitati sul ramo principale e abilitati per ogni altro ramo.



La disabilitazione dei test con questa variabile farà sì che la fase di test venga saltata del tutto durante la compilazione. Per riattivare i test, imposta questo valore su o elimina la variabile di `false` ambiente.

Utilizzo del pulsante Deploy to Amplify per condividere un progetto GitHub

> **Important**
>
> La distribuzione con un clic utilizzando il pulsante Deploy to Amplify Hosting non è più disponibile. Per eseguire la distribuzione da un repository, crea una nuova applicazione in Amplify Hosting. Per istruzioni, consultare Guida introduttiva alla distribuzione di un'app su Amplify Hosting.

Il pulsante Deploy to Amplify Hosting ti consente di GitHub condividere progetti pubblicamente o all'interno del tuo team. Di seguito è riportata un'immagine del pulsante:

Aggiungere il pulsante Deploy to Amplify Hosting a un repository o blog

Aggiungi il pulsante al file GitHub README.md, al post del blog o a qualsiasi altra pagina di markup che esegua il rendering HTML. Il pulsante ha i seguenti due componenti:

1. Un'immagine SVG situata all'URL `https://oneclick.amplifyapp.com/button.svg`
2. L'URL della console Amplify con un link al tuo repository. GitHub Puoi copiare l'URL del tuo repository, ad esempio`https://github.com/username/repository`, oppure puoi fornire un collegamento diretto a una cartella specifica, ad esempio. `https://github.com/username/repository/tree/branchname/folder` Amplify Hosting distribuirà il ramo predefinito nel tuo repository. Ulteriori rami possono essere connessi una volta connessa l'applicazione.

Usa l'esempio seguente per aggiungere il pulsante a un file markdown, come README.md. GitHub Sostituiscilo `https://github.com/username/repository` con l'URL del tuo repository.

```
[![amplifybutton](https://oneclick.amplifyapp.com/button.svg)](https://
console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/username/
repository)
```

Utilizzate l'esempio seguente per aggiungere il pulsante a qualsiasi documento HTML. Sostituiscilo `https://github.com/username/repository` con l'URL del tuo repository.

```
<a href="https://console.aws.amazon.com/amplify/home#/deploy?repo=https://github.com/
username/repository">
    <img src="https://oneclick.amplifyapp.com/button.svg" alt="Deploy to Amplify
 Hosting">
</a>
```

Configurazione di reindirizzamenti e riscritture per un'applicazione Amplify

I reindirizzamenti permettono a un server Web di reinstradare la navigazione da un URL a un altro. I motivi più comuni per utilizzare i reindirizzamenti includono la personalizzazione dell'aspetto di un URL, l'evitare collegamenti interrotti, lo spostamento della posizione di hosting di un'app o di un sito senza modificarne l'indirizzo e la modifica di un URL richiesto nel modulo richiesto da un'app Web.

Comprendere i reindirizzamenti supportati da Amplify

Amplify supporta i seguenti tipi di reindirizzamento nella console.

Permanent redirect (301) (Reindirizzamento permanente (301))

I reindirizzamenti 301 sono intesi per modifiche durature alla destinazione di un indirizzo Web. La cronologia di classificazione dei motori di ricerca per l'indirizzo originale è applicabile al nuovo indirizzo di destinazione. Il reindirizzamento si verifica lato client; una barra di navigazione del browser mostra l'indirizzo di destinazione dopo il reindirizzamento.

Le comuni motivazioni per l'utilizzo dei reindirizzamenti 301 includono:

- Per evitare un collegamento interrotto quando l'indirizzo di una pagina cambia.
- Per evitare un collegamento interrotto quando un utente inserisce un refuso prevedibile in un indirizzo.

Reindirizzamento temporaneo (302)

I reindirizzamenti 302 servono per modifiche temporanee alla destinazione di un indirizzo Web. La cronologia del posizionamento nei motori di ricerca dell'indirizzo originale non si applica al nuovo indirizzo di destinazione. Il reindirizzamento si verifica lato client; una barra di navigazione del browser mostra l'indirizzo di destinazione dopo il reindirizzamento.

Le comuni motivazioni per l'utilizzo dei reindirizzamenti 302 includono:

- Per fornire una destinazione di deviazione, mentre si tengono riparazioni sull'indirizzo originale.
- Fornire pagine di prova per il A/B confronto di un'interfaccia utente.

> **Note**
>
> Se la tua app restituisce una risposta 302 inaspettata, l'errore è probabilmente causato dalle modifiche che hai apportato al reindirizzamento dell'app e alla configurazione personalizzata dell'intestazione. Per risolvere il problema, verifica che le intestazioni personalizzate siano valide, quindi riattiva la regola di riscrittura 404 predefinita per l'app.

Rewrite (200)

I reindirizzamenti 200 (riscritture) servono per mostrare contenuti dall'indirizzo di destinazione, come se venissero forniti dall'indirizzo originale. La cronologia di classificazione dei motori di ricerca continua a essere applicata all'indirizzo originale. Il reindirizzamento si verifica lato server; una barra di navigazione del browser mostra l'indirizzo originale dopo il reindirizzamento. Le comuni motivazioni per l'utilizzo dei reindirizzamenti 200 includono:

- Per reindirizzare un intero sito in un nuovo percorso di hosting senza modificare l'indirizzo del sito.
- Per reindirizzare tutto il traffico in un'applicazione Web a singola pagina (SPA) alla sua pagina index.html per la gestione da parte di una funzione di router lato client.

Not Found (404) (Non trovato (404))

I reindirizzamenti 404 si verificano quando una richiesta punta a un indirizzo che non esiste. Viene visualizzata la pagina di destinazione di un 404 invece di quella richiesta. Le comuni motivazioni per un reindirizzamento 404 includono:

- Per evitare un messaggio collegamento interrotto quando un utente inserisce un URL sbagliato.
- Per far puntare le richieste a pagine inesistenti di un'applicazione Web sulla pagina index.html, per la gestione da una funzione router lato client.

Comprendere l'ordine dei reindirizzamenti

I reindirizzamenti vengono applicati dall'inizio dell'elenco verso il basso. Controllare che l'ordinamento abbia l'effetto inteso. Ad esempio, il seguente ordine di reindirizzamenti causa il reindirizzamento di tutte le richieste per uno specifico percorso sotto a /docs/ nello stesso percorso sotto a /documents/, tranne /docs/specific-filename.html che viene reindirizzato su /documents/different-filename.html:

```
/docs/specific-filename.html /documents/different-filename.html 301
/docs/<*> /documents/<*>
```

Il seguente ordine di reindirizzamenti ignora il reindirizzamento di specific-filename.html su different-filename.html:

```
/docs/<*> /documents/<*>
/docs/specific-filename.html /documents/different-filename.html 301
```

Comprendere come Amplify inoltra i parametri di interrogazione

Puoi utilizzare i parametri di query per un maggiore controllo sulle corrispondenze degli URL. Amplify inoltra tutti i parametri della query al percorso di destinazione per i reindirizzamenti 301 e 302, con le seguenti eccezioni:

- Se l'indirizzo originale include una stringa di query impostata su un valore specifico, Amplify non inoltra i parametri di query. In questo caso, il reindirizzamento si applica solo alle richieste all'URL di destinazione con il valore di query specificato.
- Se l'indirizzo di destinazione per la regola di corrispondenza ha parametri di query, i parametri di query non vengono inoltrati. Ad esempio, se l'indirizzo di destinazione per il reindirizzamento è`https://example-target.com?q=someParam`, i parametri di query non vengono trasmessi.

Creazione e modifica di reindirizzamenti nella console Amplify

Puoi creare e modificare i reindirizzamenti per un'applicazione nella console Amplify. Prima di iniziare, avrai bisogno delle seguenti informazioni sulle parti di un reindirizzamento.

Un indirizzo originale

L'indirizzo richiesto dall'utente.

Un indirizzo di destinazione

L'indirizzo che serve effettivamente il contenuto visualizzato dall'utente.

Un tipo di reindirizzamento

I tipi includono un reindirizzamento permanente (301), un reindirizzamento temporaneo (302), una riscrittura (200) o un reindirizzamento non trovato (404).

Un codice del paese di due lettere (opzionale)

Un valore che puoi includere per segmentare l'esperienza utente della tua app per area geografica.

Per creare un reindirizzamento nella console Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui desideri creare un reindirizzamento.
3. Nel riquadro di navigazione, scegli Hosting, quindi scegli Riscritture e reindirizzamenti.
4. Nella pagina Riscritture e reindirizzamenti, scegli Gestisci reindirizzamenti.
5. Aggiungi o aggiorna manualmente i reindirizzamenti nell'editor JSON di riscrittura e reindirizzamento.
   a. `Persource`, specifica l'indirizzo originale richiesto dall'utente.
   b. `Perstatus`, specifica il tipo di reindirizzamento.
   c. `Pertarget`, specifica l'indirizzo di destinazione che invia il contenuto all'utente.
   d. (Facoltativo) `Percondition`, inserite una condizione relativa al prefisso internazionale di due lettere.
6. Scegli Save (Salva).

Reindirizza e riscrive il riferimento di esempio

Questa sezione fornisce esempi per una serie di scenari di reindirizzamento comuni. Puoi usare questi esempi per comprendere la sintassi JSON per creare reindirizzamenti e riscritture personalizzati nell'editor JSON della console Amplify.

> **Note**
> La corrispondenza tra indirizzi e domini originali non fa distinzione tra maiuscole e minuscole.

Argomenti

- Reindirizzamenti e riscritture semplici
- Reindirizzamenti per app Web a pagina singola (SPA)

- Riscrittura inversa del proxy
- Barre finali e pulizia URLs
- Placeholder
- Stringhe di query e parametri del percorso
- Reindirizzamenti basati sulla regione
- Utilizzo di espressioni con caratteri jolly nei reindirizzamenti e nelle riscritture

Reindirizzamenti e riscritture semplici

È possibile utilizzare l'esempio seguente per reindirizzare in modo permanente una pagina specifica a un nuovo indirizzo.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/original.html`	`/destination.html`	`permanent redirect (301)`	

Formato JSON

```
[
  {
    "source": "/original.html",
    "status": "301",
    "target": "/destination.html",
    "condition": null
  }
]
```

È possibile utilizzare l'esempio seguente per reindirizzare qualsiasi percorso contenuto in una cartella verso lo stesso percorso in una cartella diversa.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

Formato JSON

```
[
  {
    "source": "/docs/<*>",
    "status": "301",
    "target": "/documents/<*>",
    "condition": null
  }
]
```

È possibile utilizzare l'esempio seguente per reindirizzare tutto il traffico verso index.html come riscrittura. In questo scenario, la riscrittura fa credere all'utente di aver raggiunto l'indirizzo originale.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/<*>`	`/index.html`	`rewrite (200)`	

Formato JSON

```
[
  {
    "source": "/<*>",
    "status": "200",
    "target": "/index.html",
    "condition": null
  }
]
```

È possibile utilizzare l'esempio seguente per utilizzare una riscrittura per modificare il sottodominio visualizzato all'utente.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`https://mydomain.com`	`https://www.mydomain.com`	`rewrite (200)`	

Formato JSON

```
[
  {
    "source": "https://mydomain.com",
    "status": "200", "target": "https://www.mydomain.com",
    "condition": null
  }
]
```

È possibile utilizzare l'esempio seguente per reindirizzare a un dominio diverso con un prefisso di percorso.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`https://mydomain.com`	`https://www.mydomain.com/documents`	`temporary redirect (302)`	

Formato JSON

```
[
  {
    "source": "https://mydomain.com",
    "status": "302",
    "target": "https://www.mydomain.com/documents/",
    "condition": null
  }
```

```
]
```

È possibile utilizzare l'esempio seguente per reindirizzare i percorsi di una cartella non trovata a una pagina 404 personalizzata.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
/<*>	/404.html	not found (404)	

Formato JSON

```
[
  {
    "source": "/<*>",
    "status": "404",
    "target": "/404.html",
    "condition": null
  }
]
```

Reindirizzamenti per app Web a pagina singola (SPA)

La maggior parte dei framework SPA supporta HTML5 history.pushState () per modificare la posizione del browser senza avviare una richiesta al server. Funziona per gli utenti che iniziano il proprio percorso dalla root (o da /index.html), ma fallisce per gli utenti che accedono direttamente a qualsiasi altra pagina.

L'esempio seguente utilizza le espressioni regolari per impostare una riscrittura di 200 per tutti i file in index.html, ad eccezione delle estensioni di file specificate nell'espressione regolare.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
</^[^.]+$|\.(?!(css|gif|ico|jpg|js|png|txt|svg|woff|	/index.html	200	

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`woff2|ttf|map|json|webp)$)([^.]+$)/>`			

Formato JSON

```
[
  {
    "source": "</^[^.]+$|\.(?!(css|gif|ico|jpg|js|png|txt|svg|woff|woff2|ttf|map|json|webp)$)([^.]+$)/>",
    "status": "200",
    "target": "/index.html",
    "condition": null
  }
]
```

Riscrittura inversa del proxy

L'esempio seguente utilizza una riscrittura del contenuto proxy da un'altra posizione in modo che all'utente appaia che il dominio non è cambiato. HTTPS è l'unico protocollo supportato per i proxy inversi.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/images/<*>`	`https://images.otherdomain.com/<*>`	`rewrite (200)`	

Formato JSON

```
[
  {
```

```
    "source": "/images/<*>",
    "status": "200",
    "target": "https://images.otherdomain.com/<*>",
    "condition": null
  }
]
```

Barre finali e pulizia URLs

Per creare strutture URL pulite, come about anziché about.html, i generatori di siti statici, come Hugo, generano directory per le pagine con index.html (/about/index.html). Amplify crea automaticamente URLs clean aggiungendo una barra finale quando necessario. La tabella seguente illustra diversi scenari:

Input utente dal browser	URL nella barra degli indirizzi	Documento fornito
`/about`	`/about`	`/about.html`
`/about (when about.html returns 404)`	`/about/`	`/about/index.html`
`/about/`	`/about/`	`/about/index.html`

Placeholder

È possibile utilizzare l'esempio seguente per reindirizzare i percorsi in una struttura di cartelle verso una struttura corrispondente in un'altra cartella.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/docs/<year>/<month>/<date>/<itemid>`	`/documents/<year>/<month>/<date>/<itemid>`	`permanent redirect (301)`	

Formato JSON

```
[
  {
    "source":  "/docs/<year>/<month>/<date>/<itemid>",
    "status": "301",
    "target": "/documents/<year>/<month>/<date>/<itemid>",
               "condition": null
   }
]
```

Stringhe di query e parametri del percorso

> **Warning**
>
> Non includere segreti, credenziali o dati sensibili URLs come parametri di percorso o di query. Questi valori sono visualizzabili in testo semplice nei log di accesso dell'applicazione Amplify.

È possibile utilizzare l'esempio seguente per reindirizzare un percorso a una cartella con un nome che corrisponde al valore di un elemento della stringa di query nell'indirizzo originale:

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/docs?id=<my-blog-id-value`	`/documents/<my-blog-post-id-value>`	`permanent redirect (301)`	

Formato JSON

```
[
  {
    "source": "/docs?id=<my-blog-id-value>",
    "status": "301",
    "target": "/documents/<my-blog-id-value>",
    "condition": null
  }
]
```

> **Note**
>
> Amplify inoltra tutti i parametri della stringa di query al percorso di destinazione per i reindirizzamenti 301 e 302. Tuttavia, se l'indirizzo originale include una stringa di query impostata su un valore specifico, come dimostrato in questo esempio, Amplify non inoltra i parametri di query. In questo caso, il reindirizzamento si applica solo alle richieste all'indirizzo di destinazione con il valore di query specificato. `id`

È possibile utilizzare l'esempio seguente per reindirizzare tutti i percorsi che non possono essere trovati a un determinato livello di una struttura di cartelle a index.html in una cartella specificata.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/documents/<folder>/<child-folder>/<grand-child-folder>`	`/documents/index.html`	`not found (404)`	

Formato JSON

```
[
  {
    "source": "/documents/<x>/<y>/<z>",
    "status": "404",
    "target": "/documents/index.html",
    "condition": null
  }
]
```

Reindirizzamenti basati sulla regione

È possibile utilizzare il seguente esempio per reindirizzare le richieste in base alla regione.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/documents`	`/documents/us/`	`temporary redirect (302)`	`<US>`

Formato JSON

```
[
  {
    "source": "/documents",
    "status": "302",
    "target": "/documents/us/",
    "condition": "<US>"
  }
]
```

Utilizzo di espressioni con caratteri jolly nei reindirizzamenti e nelle riscritture

È possibile utilizzare l'espressione con caratteri jolly`<*>`, nell'indirizzo originale per un reindirizzamento o una riscrittura. È necessario inserire l'espressione alla fine dell'indirizzo originale e deve essere univoca. Amplify ignora gli indirizzi originali che includono più di un'espressione con caratteri jolly o li utilizza in una posizione diversa.

Di seguito è riportato un esempio di reindirizzamento valido con un'espressione con caratteri jolly.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/docs/<*>`	`/documents/<*>`	`permanent redirect (301)`	

I due esempi seguenti mostrano reindirizzamenti non validi con espressioni con caratteri jolly.

Indirizzo originale	Indirizzo di destinazione	Tipo di reindirizzamento	Codice del paese
`/docs/<*>/content`	`/documents/<*>/content`	`permanent redirect (301)`	
`/docs/<*>/content/<*>`	`/documents/<*>/content/<*>`	`permanent redirect (301)`	

Utilizzo delle variabili di ambiente in un'applicazione Amplify

Le variabili di ambiente sono coppie chiave-valore che puoi aggiungere alle impostazioni dell'applicazione per renderle disponibili ad Amplify Hosting. Come best practice, è possibile utilizzare le variabili di ambiente per esporre i dati di configurazione dell'applicazione. Tutte le variabili di ambiente aggiunte sono crittografate per impedire accessi non autorizzati.

Amplify applica i seguenti vincoli alle variabili di ambiente create.

- Amplify non consente di creare nomi di variabili di ambiente con un prefisso. `AWS` Questo prefisso è riservato solo per uso interno di Amplify.
- Il valore di una variabile di ambiente non può superare i 5500 caratteri.

> **Important**
>
> Non utilizzare variabili di ambiente per memorizzare segreti. Per un'app di seconda generazione, utilizza la funzionalità di gestione segreta nella console Amplify. Per ulteriori informazioni, consulta Secrets and environment vars nella documentazione di Amplify. Per un'app di prima generazione, archivia i segreti in un ambiente segreto creato utilizzando Parameter Store. AWS Systems Manager Per ulteriori informazioni, consulta Gestione dei segreti ambientali.

Riferimento alla variabile di ambiente Amplify

Le seguenti variabili di ambiente sono accessibili per impostazione predefinita all'interno della console Amplify.

Nome della variabile	Descrizione	Valore di esempio
_BUILD_TIMEOUT	La durata del timeout di compilazione in minuti. Il valore minimo è 5. Il valore massimo è 120.	`30`

Nome della variabile	Descrizione	Valore di esempio
_LIVE_UPDATES	Lo strumento verrà aggiornato alla versione più recente.	`[{"name":"Amplify CLI","pkg":"@aws-amplify/cli","type":"npm","version":"latest"}]`
USER_DISABLE_TESTS	La fase di test viene saltata durante la compilazione. Puoi disabilitare i test per tutte le filiali o per quelle specifiche di un'app. Questa variabile di ambiente viene utilizzata per le app che eseguono test durante la fase di compilazione. Per ulteriori informazioni sull'impostazione di questa variabile, vedereDisattivazione dei test per un'applicazione o un ramo Amplify.	`true`
AWS_APP_ID	ID dell'applicazione della compilazione corrente	`abcd1234`
AWS_BRANCH	Nome del ramo della compilazione corrente	`main`, `develop`, `beta`, `v2.0`
AWS_BRANCH_ARN	La filiale Amazon Resource Name (ARN) della build corrente	`aws:arn:amplify:us-west-2:123456789012:appname/branch/...`
AWS_CLONE_URL	URL di clonazione utilizzato per recuperare i contenuti del repository git	`git@github.com:<user-name>/<repo-name>.git`

Nome della variabile	Descrizione	Valore di esempio
AWS_COMMIT_ID	L'ID di commit della build corrente «HEAD» per le ricostruzioni	`abcd1234`
AWS_JOB_ID	ID processo della compilazione corrente. Questo include una certa imbottitura di '0' in modo che abbia sempre la stessa lunghezza.	`0000000001`
AWS_PULL_REQUEST_ID	L'ID della pull request della build di anteprima web della pull request. Questa variabile di ambiente non è disponibile quando viene utilizzata AWS CodeCommit come provider di repository.	`1`
AWS_PULL_REQUEST_SOURCE_BRANCH	Il nome del feature branch per l'anteprima di una pull request inviata a un ramo dell'applicazione nella console Amplify.	`featureA`
AWS_PULL_REQUEST_DESTINATION_BRANCH	Il nome del ramo dell'applicazione nella console Amplify a cui viene inviata una richiesta pull del feature branch.	`main`
AMPLIFY_AMAZON_CLIENT_ID	L'ID client Amazon	`123456`

Nome della variabile	Descrizione	Valore di esempio
AMPLIFY_AMAZON_CLIENT_SECRET	Il segreto del cliente Amazon	`example123456`
AMPLIFY_FACEBOOK_CLIENT_ID	L'ID del client di Facebook	`123456`
AMPLIFY_FACEBOOK_CLIENT_SECRET	Il segreto del client di Facebook	`example123456`
AMPLIFY_GOOGLE_CLIENT_ID	L'ID del client Google	`123456`
AMPLIFY_GOOGLE_CLIENT_SECRET	Il segreto del client Google	`example123456`
AMPLIFY_DIFF_DEPLOY	Abilita o disabilita la distribuzione frontend basata su diff. Per ulteriori informazioni, consulta Configurazione della build e della distribuzione del frontend basato su diff.	`true`
AMPLIFY_DIFF_DEPLOY_ROOT	Il percorso da utilizzare per i confronti delle distribuzioni frontend basate su diff, rispetto alla radice del repository.	`dist`
AMPLIFY_DIFF_BACKEND	Abilita o disabilita le build di backend basate su diff. Questo vale solo per le app di prima generazione. Per ulteriori informazioni, consulta Configurazione di build di backend basate su diff per un'app di prima generazione	`true`

Nome della variabile	Descrizione	Valore di esempio
AMPLIFY_BACKEND_PULL_ONLY	Amplify gestisce questa variabile d'ambiente. Questo vale solo per le app di prima generazione. Per ulteriori informazioni, consulta Modifica un frontend esistente in modo che punti a un backend diverso	`true`
AMPLIFY_BACKEND_APP_ID	Amplify gestisce questa variabile d'ambiente. Questo vale solo per le app di prima generazione. Per ulteriori informazioni, consulta Modifica un frontend esistente in modo che punti a un backend diverso	`abcd1234`
AMPLIFY_SKIP_BACKEND_BUILD	Se non hai una sezione di backend nelle specifiche della build e desideri disabilitare le build di backend, imposta questa variabile di ambiente su. `true` Questo vale solo per le app di prima generazione.	`true`
AMPLIFY_ENABLE_DEBUG_OUTPUT	Imposta questa variabile su `true` per stampare una traccia dello stack nei log. Questo è utile per il debug degli errori di compilazione del backend.	`true`

Nome della variabile	Descrizione	Valore di esempio
AMPLIFY_MONOREPO_APP_ROOT	Il percorso da utilizzare per specificare la radice dell'app di un'app monorepo, relativa alla radice del repository.	`apps/react-app`
AMPLIFY_USERPOOL_ID	L'ID per il pool di utenti di Amazon Cognito importato per l'autenticazione	`us-west-2_example`
AMPLIFY_WEBCLIENT_ID	L'ID del client dell'app che deve essere utilizzato dalle applicazioni Web Il client dell'app deve essere configurato con l'accesso al pool di utenti di Amazon Cognito specificato dalla variabile di ambiente AMPLIFY_USERPOOL_ID.	`123456`
AMPLIFY_NATIVECLIENT_ID	L'ID del client dell'app che deve essere utilizzato dalle applicazioni native Il client dell'app deve essere configurato con l'accesso al pool di utenti di Amazon Cognito specificato dalla variabile di ambiente AMPLIFY_USERPOOL_ID.	`123456`
AMPLIFY_IDENTITYPOOL_ID	L'ID per il pool di identità di Amazon Cognito	`example-identitypool-id`

Nome della variabile	Descrizione	Valore di esempio
AMPLIFY_PERMISSIONS_BOUNDARY_ARN	L'ARN per la policy IAM da utilizzare come limite di autorizzazioni che si applica a tutti i ruoli IAM creati da Amplify.	`arn:aws:iam::123456789012:policy/example-policy`
AMPLIFY_DESTRUCTIVE_UPDATES	Imposta questa variabile di ambiente su true per consentire l'aggiornamento di un'API GraphQL con operazioni di schema che possono potenzialmente causare la perdita di dati.	`true`

> **Note**
>
> Le variabili `AMPLIFY_AMAZON_CLIENT_ID` di `AMPLIFY_AMAZON_CLIENT_SECRET` ambiente sono OAuth token, non una chiave di AWS accesso e una chiave segreta.

Variabili di ambiente del framework frontend

Se stai sviluppando la tua app con un framework frontend che supporta le proprie variabili di ambiente, è importante capire che queste non sono le stesse variabili di ambiente che configuri nella console Amplify. Ad esempio, React (prefisso REACT_APP) e Gatsby (prefisso GATSBY), consentono di creare variabili di ambiente di runtime che tali framework raggruppano automaticamente nella build di produzione del frontend. Per comprendere gli effetti dell'utilizzo di queste variabili di ambiente per memorizzare valori, consulta la documentazione del framework di frontend che stai utilizzando.

La memorizzazione di valori sensibili, come le chiavi API, all'interno di queste variabili di ambiente con prefisso del framework di frontend non è una buona pratica ed è altamente sconsigliata.

Impostazione delle variabili di ambiente

Usa le seguenti istruzioni per impostare le variabili di ambiente per un'applicazione nella console Amplify.

> **Note**
>
> Le variabili di ambiente sono visibili nel menu delle impostazioni dell'app della console Amplify solo quando un'app è configurata per la distribuzione continua e connessa a un repository git. Per istruzioni su questo tipo di distribuzione, consulta Guida introduttiva al codice esistente.

Per impostare le variabili di ambiente

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella console Amplify, scegli Hosting, quindi scegli Variabili di ambiente.
3. Nella pagina Variabili di ambiente, scegli Gestisci variabili.
4. Per Variabile, inserisci la tua chiave. Per Valore, inserisci il tuo valore. Per impostazione predefinita, Amplify applica le variabili di ambiente a tutti i rami, quindi non è necessario reinserire le variabili quando si collega un nuovo ramo.
5. (Facoltativo) Per personalizzare una variabile di ambiente specifica per un ramo, aggiungete un branch override come segue:
   a. Scegliete Azioni, quindi scegliete Aggiungi override variabile.
   b. A questo punto è stata impostato un set di variabili d'ambiente specifiche per il branch.
6. Scegli Save (Salva).

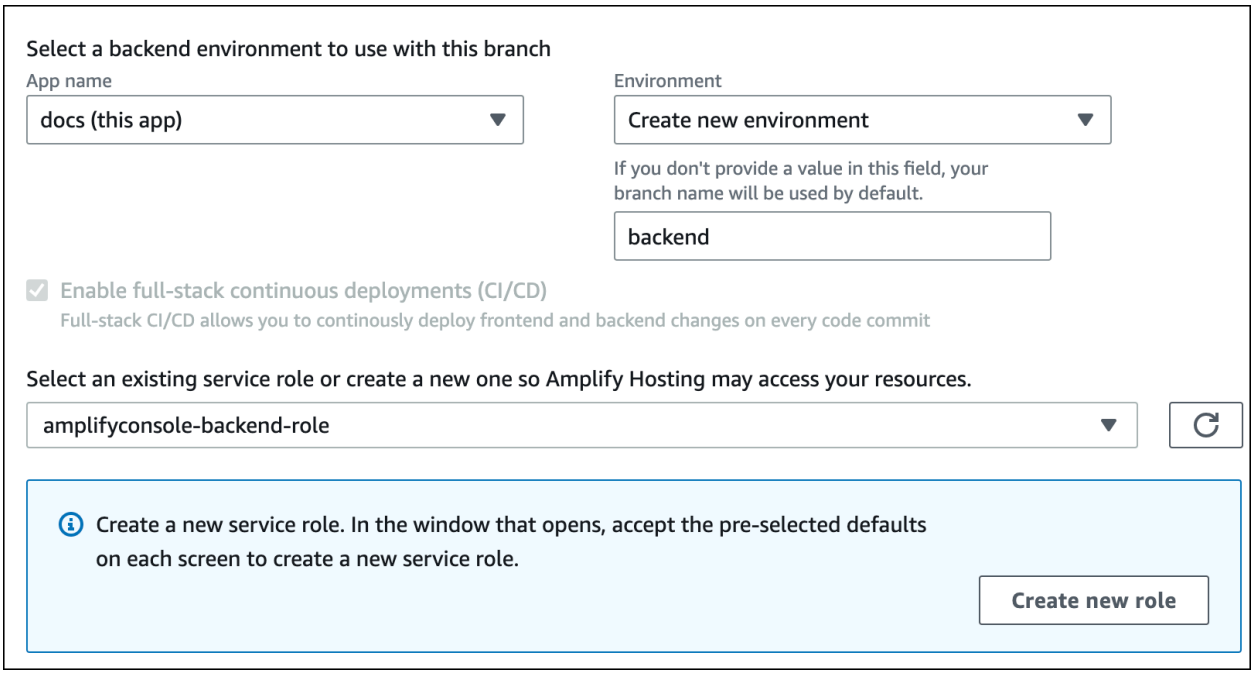
Crea un nuovo ambiente di backend con parametri di autenticazione per l'accesso tramite social

Per connettere una filiale a un'app

1. Accedi AWS Management Console e apri la console Amplify.
2. La procedura per connettere una filiale a un'app varia a seconda che si stia connettendo una filiale a una nuova app o a un'app esistente.

- Connessione di una filiale a una nuova app

  a. Nella pagina delle impostazioni di build, individua la sezione Seleziona un ambiente di backend da utilizzare con questo ramo. Per Ambiente, scegli Crea nuovo ambiente e inserisci il nome del tuo ambiente di backend. La schermata seguente mostra la sezione Seleziona un ambiente di backend da usare con questo ramo della pagina delle impostazioni di build con cui è stato **`backend`** inserito il nome dell'ambiente di backend.



  b. Espandi la sezione Impostazioni avanzate nella pagina delle impostazioni di Build e aggiungi le variabili di ambiente per le chiavi di accesso social. Ad esempio, **`AMPLIFY_FACEBOOK_CLIENT_SECRET`** è una variabile di ambiente valida. Per l'elenco delle variabili di ambiente del sistema Amplify disponibili per impostazione predefinita, vedere la tabella in. Riferimento alla variabile di ambiente Amplify

- Connessione di una filiale a un'app esistente

  a. Se stai connettendo una nuova filiale a un'app esistente, imposta le variabili di ambiente di accesso social prima di connettere la filiale. Nel pannello di navigazione, scegli Impostazioni app, Variabili di ambiente.

  b. Nella sezione Variabili di ambiente, scegli Gestisci variabili.

  c. Nella sezione Gestisci variabili, scegli Aggiungi variabile.

  d. Per Variabile (chiave), inserisci il tuo ID cliente. In Value, inserisci il segreto del tuo cliente.

e. Scegli, Salva.

Gestione dei segreti ambientali

Con il rilascio di Amplify Gen 2, il flusso di lavoro per i segreti ambientali è semplificato per centralizzare la gestione dei segreti e delle variabili di ambiente nella console Amplify. Per istruzioni sull'impostazione e l'accesso ai segreti per un'app Amplify Gen 2, consulta Segreti e variabili di ambiente nella documentazione di Amplify.

I segreti ambientali per un'app di prima generazione sono simili alle variabili di ambiente, ma sono coppie di valori chiave di AWS Systems Manager Parameter Store che possono essere crittografate. Alcuni valori devono essere crittografati, ad esempio la chiave privata Accedi con Apple per Amplify.

Utilizzo AWS Systems Manager per impostare segreti ambientali per un'applicazione Amplify Gen 1

Usa le seguenti istruzioni per impostare un ambiente segreto per un'app Amplify di prima generazione utilizzando la console. AWS Systems Manager

Per impostare un segreto ambientale

1. Accedi a AWS Management Console e apri la AWS Systems Manager console.
2. Nel riquadro di navigazione, scegli Gestione applicazioni, quindi scegli Parameter Store.
3. Nella pagina AWS Systems Manager Parameter Store, scegli Crea parametro.
4. Nella pagina Crea parametro, nella sezione Dettagli dei parametri, procedi come segue:
   a. Per Nome, immettete un parametro nel formato**`/amplify/{your_app_id}/{your_backend_environment_name}/{your_parameter_name}`**.
   b. In Type (Tipo) scegliere SecureString.
   c. Per la fonte della chiave KMS, scegli Il mio account corrente per utilizzare la chiave predefinita per il tuo account.
   d. In Value, inserisci il valore segreto da crittografare.
5. Scegli, Crea parametro.

> **Note**
>
> Amplify ha accesso solo alle chiavi contenute nella build `/amplify/{your_app_id}/{your_backend_environment_name}` dell'ambiente specifico. È necessario specificare l'impostazione predefinita AWS KMS key per consentire ad Amplify di decrittografare il valore.

Accesso ai segreti ambientali per un'applicazione di prima generazione

I segreti ambientali per un'applicazione di prima generazione vengono archiviati `process.env.secrets` come stringa JSON.

Riferimento ai segreti dell'ambiente Amplify

Specificate un parametro Systems Manager nel formato`/amplify/{your_app_id}/{your_backend_environment_name}/AMPLIFY_SIWA_CLIENT_ID`.

È possibile utilizzare i seguenti segreti ambientali, accessibili per impostazione predefinita all'interno della console Amplify.

Nome della variabile	Descrizione	Valore di esempio
AMPLIFY_SIWA_CLIENT_ID	L'accesso con l'ID client Apple	`com.yourapp.auth`
AMPLIFY_SIWA_TEAM_ID	L'accesso con l'ID del team Apple	`ABCD123`
AMPLIFY_SIWA_KEY_ID	L'ID Accedi con la chiave Apple	`ABCD123`
AMPLIFY_SIWA_PRIVATE_KEY	L'accesso con la chiave privata Apple	-----CHIAVE PRIVATA DI INIZIO----- ****..... -----CHIAVE PRIVATA DI FINE CORSO-----

Impostazione di intestazioni personalizzate per un'app Amplify

Le intestazioni HTTP personalizzate consentono di specificare le intestazioni per ogni risposta HTTP. Le intestazioni di risposta possono essere utilizzate per scopi di debug, sicurezza e informativi. Puoi specificare le intestazioni nella console Amplify oppure scaricando e modificando il file di un'app e salvandolo nella directory principale `customHttp.yml` del progetto. Per le procedure dettagliate, consulta Impostazione di intestazioni personalizzate.

In precedenza, le intestazioni HTTP personalizzate venivano specificate per un'app modificando le specifiche di build (buildspec) nella console Amplify o scaricando e aggiornando il `amplify.yml` file e salvandolo nella directory principale del progetto. Consigliamo vivamente di migrare le intestazioni personalizzate specificate in questo modo fuori da buildspec e dal file. `amplify.yml` Per istruzioni, consulta Migrazione delle intestazioni personalizzate fuori dalle specifiche di build e amplify.yml.

Argomenti

- Riferimento YAML per intestazioni personalizzate
- Impostazione di intestazioni personalizzate
- Migrazione delle intestazioni personalizzate fuori dalle specifiche di build e amplify.yml
- Requisiti per le intestazioni personalizzate di Monorepo

Riferimento YAML per intestazioni personalizzate

Specificate le intestazioni personalizzate utilizzando il seguente formato YAML:

```
customHeaders:
  - pattern: '*.json'
    headers:
    - key: 'custom-header-name-1'
      value: 'custom-header-value-1'
    - key: 'custom-header-name-2'
      value: 'custom-header-value-2'
  - pattern: '/path/*'
    headers:
    - key: 'custom-header-name-1'
      value: 'custom-header-value-2'
```

Per un monorepo, usa il seguente formato YAML:

```
applications:
  - appRoot: app1
    customHeaders:
    - pattern: '**/*'
      headers:
      - key: 'custom-header-name-1'
        value: 'custom-header-value-1'
  - appRoot: app2
    customHeaders:
    - pattern: '/path/*.json'
      headers:
      - key: 'custom-header-name-2'
        value: 'custom-header-value-2'
```

Quando aggiungi intestazioni personalizzate alla tua app, specificherai i tuoi valori per quanto segue:

pattern

Le intestazioni personalizzate vengono applicate a tutti i percorsi dei file URL che corrispondono al modello.

headers

Definisce le intestazioni che corrispondono al modello del file.

Chiave

Il nome dell'intestazione personalizzata.

value

Il valore dell'intestazione personalizzata.

Per ulteriori informazioni sulle intestazioni HTTP, consulta l'elenco delle intestazioni HTTP di Mozilla.

Impostazione di intestazioni personalizzate

Esistono due modi per specificare intestazioni HTTP personalizzate per un'app Amplify. Puoi specificare le intestazioni nella console Amplify oppure puoi specificare le intestazioni scaricando e modificando il file di un'app e salvandolo nella directory principale `customHttp.yml` del progetto.

Per impostare intestazioni personalizzate per un'app e salvarle nella console

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui impostare intestazioni personalizzate.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Intestazioni personalizzate.
4. Nella pagina Intestazioni personalizzate, scegli Modifica.
5. Nella finestra Modifica intestazioni personalizzate, inserisci le informazioni per le intestazioni personalizzate utilizzando il formato YAML dell'intestazione personalizzata.
   a. Per`pattern`, inserisci lo schema da abbinare.
   b. Per`key`, inserisci il nome dell'intestazione personalizzata.
   c. Per`value`, inserisci il valore dell'intestazione personalizzata.
6. Seleziona Salva.
7. Ridistribuisci l'app per applicare le nuove intestazioni personalizzate.
   - Per un'app CI/CD, vai alla filiale da distribuire e scegli Redeploy this version. Puoi anche eseguire una nuova build dal tuo repository Git.
   - Per un'app con distribuzione manuale, distribuisci nuovamente l'app nella console Amplify.

Per impostare intestazioni personalizzate per un'app e salvarle nella radice del tuo repository

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui impostare intestazioni personalizzate.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Intestazioni personalizzate.
4. Nella pagina Intestazioni personalizzate, scegli Scarica YML.
5. Apri il `customHttp.yml` file scaricato nell'editor di codice che preferisci e inserisci le informazioni per le intestazioni personalizzate utilizzando il formato YAML dell'intestazione personalizzata.
   a. Per`pattern`, inserisci lo schema da abbinare.
   b. Per`key`, inserisci il nome dell'intestazione personalizzata.
   c. Per`value`, inserisci il valore dell'intestazione personalizzata.
6. Salva il `customHttp.yml` file modificato nella directory principale del progetto. Se stai lavorando con un monorepo, salva il `customHttp.yml` file nella radice del tuo repository.
7. Ridistribuisci l'app per applicare le nuove intestazioni personalizzate.

- Per un'app CI/CD, esegui una nuova build dal tuo repository Git che includa il nuovo file. `customHttp.yml`
- Per un'app con distribuzione manuale, distribuisci nuovamente l'app nella console Amplify e includi il nuovo `customHttp.yml` file con gli artefatti che carichi.

> **Note**
>
> Le intestazioni personalizzate impostate nel `customHttp.yml` file e distribuite nella directory principale dell'app sostituiscono le intestazioni personalizzate definite nella sezione Intestazioni personalizzate della console Amplify.

Esempio di intestazioni personalizzate di sicurezza

Le intestazioni di sicurezza personalizzate consentono di far rispettare l'HTTPS, prevenire gli attacchi XSS e difendere il browser dal clickjacking. Utilizza la seguente sintassi YAML per applicare intestazioni di sicurezza personalizzate alla tua app.

```
customHeaders:
  - pattern: '**'
    headers:
      - key: 'Strict-Transport-Security'
        value: 'max-age=31536000; includeSubDomains'
      - key: 'X-Frame-Options'
        value: 'SAMEORIGIN'
      - key: 'X-XSS-Protection'
        value: '1; mode=block'
      - key: 'X-Content-Type-Options'
        value: 'nosniff'
      - key: 'Content-Security-Policy'
        value: "default-src 'self'"
```

Impostazione delle intestazioni personalizzate di Cache-Control

Le app ospitate con Amplify rispettano `Cache-Control` le intestazioni inviate dall'origine, a meno che non le sovrascriviate con intestazioni personalizzate da voi definite. Amplify applica solo le intestazioni personalizzate Cache-Control per risposte riuscite con un codice di stato. `200 OK` Ciò

impedisce che le risposte agli errori vengano memorizzate nella cache e inviate ad altri utenti che effettuano la stessa richiesta.

Puoi modificare manualmente la `s-maxage` direttiva per avere un maggiore controllo sulle prestazioni e sulla disponibilità di implementazione della tua app. Ad esempio, per aumentare il periodo di tempo in cui i contenuti rimangono memorizzati nella cache periferica, puoi aumentare manualmente il time to live (TTL) eseguendo l'aggiornamento `s-maxage` a un valore più lungo del valore predefinito di 600 secondi (10 minuti).

Per specificare un valore personalizzato per`s-maxage`, utilizzate il seguente formato YAML. Questo esempio mantiene il contenuto associato memorizzato nella cache periferica per 3600 secondi (un'ora).

```
customHeaders:
  - pattern: '/img/*'
    headers:
      - key: 'Cache-Control'
        value: 's-maxage=3600'
```

Per ulteriori informazioni sul controllo delle prestazioni delle applicazioni con le intestazioni, consulta. Utilizzo dell'intestazione Cache-Control per aumentare le prestazioni dell'app

Migrazione delle intestazioni personalizzate fuori dalle specifiche di build e amplify.yml

In precedenza, le intestazioni HTTP personalizzate venivano specificate per un'app modificando le specifiche di build nella console Amplify o scaricando e aggiornando `amplify.yml` il file e salvandolo nella directory principale del progetto. Si consiglia vivamente di migrare le intestazioni personalizzate dalle specifiche di build e dal file. `amplify.yml`

Specificate le intestazioni personalizzate nella sezione Intestazioni personalizzate della console Amplify o scaricando e modificando il file. `customHttp.yml`

Per migrare le intestazioni personalizzate archiviate nella console Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app su cui eseguire la migrazione personalizzata dell'header.
3. Nel pannello di navigazione, scegli Hosting, Crea impostazioni. Nella sezione delle specifiche di build dell'app, puoi rivedere le specifiche di build dell'app.

4. Scegli Scarica per salvare una copia delle specifiche di build correnti. Puoi fare riferimento a questa copia in un secondo momento se hai bisogno di ripristinare le impostazioni.
5. Una volta completato il download, scegli Modifica.
6. Prendi nota delle informazioni di intestazione personalizzate nel file, poiché le utilizzerai più avanti nel passaggio 9. Nella finestra Modifica, elimina tutte le intestazioni personalizzate dal file e scegli Salva.
7. Nel pannello di navigazione, scegli Hosting, Intestazioni personalizzate.
8. Nella pagina Intestazioni personalizzate, scegli Modifica.
9. Nella finestra Modifica intestazioni personalizzate, inserisci le informazioni relative alle intestazioni personalizzate che hai eliminato nel passaggio 6.
10. Seleziona Salva.
11. Ridistribuisci qualsiasi ramo a cui desideri applicare le nuove intestazioni personalizzate.

Per migrare le intestazioni personalizzate da amplify.yml a CustomHttp.yml

1. Passa al file attualmente distribuito nella directory principale dell'app. `amplify.yml`
2. Apri `amplify.yml` nell'editor di codice che preferisci.
3. Prendi nota delle informazioni di intestazione personalizzate nel file, poiché le utilizzerai più avanti nel passaggio 8. Eliminare le intestazioni personalizzate nel file. Salva e chiudi il file.
4. Accedi AWS Management Console e apri la console Amplify.
5. Scegli l'app per cui impostare intestazioni personalizzate.
6. Nel riquadro di navigazione, scegli Hosting, Intestazioni personalizzate.
7. Nella pagina Intestazioni personalizzate, scegli Scarica.
8. Apri il `customHttp.yml` file scaricato nell'editor di codice che preferisci e inserisci le informazioni per le intestazioni personalizzate da cui hai eliminato `amplify.yml` nel passaggio 3.
9. Salva il `customHttp.yml` file modificato nella directory principale del progetto. Se stai lavorando con un monorepo, salva il file nella radice del tuo repository.
10. Ridistribuisci l'app per applicare le nuove intestazioni personalizzate.
    - Per un'app CI/CD, esegui una nuova build dal tuo repository Git che includa il nuovo file. `customHttp.yml`
    - Per un'app con distribuzione manuale, distribuisci nuovamente l'app nella console Amplify e includi il nuovo `customHttp.yml` file con gli artefatti che carichi.

> **Note**
>
> Le intestazioni personalizzate impostate nel `customHttp.yml` file e distribuite nella directory principale dell'app sostituiscono le intestazioni personalizzate definite nella sezione Intestazioni personalizzate della console Amplify.

Requisiti per le intestazioni personalizzate di Monorepo

Quando specifichi intestazioni personalizzate per un'app in un monorepo, tieni presente i seguenti requisiti di configurazione:

- Esiste un formato YAML specifico per un monorepo. Per la sintassi corretta, vedere. Riferimento YAML per intestazioni personalizzate
- È possibile specificare intestazioni personalizzate per un'applicazione in un monorepo utilizzando la sezione Custom header della console Amplify. È necessario ridistribuire l'applicazione per applicare le nuove intestazioni personalizzate.
- In alternativa all'utilizzo della console, puoi specificare intestazioni personalizzate per un'app in un monorepo in un file. `customHttp.yml` È necessario salvare il `customHttp.yml` file nella radice del repository e quindi ridistribuire l'applicazione per applicare le nuove intestazioni personalizzate. Le intestazioni personalizzate specificate nel `customHttp.yml` file sovrascrivono le intestazioni personalizzate specificate utilizzando la sezione Intestazioni personalizzate della console Amplify.

Gestione della configurazione della cache per un'app

Amplify utilizza CloudFront Amazon per gestire la configurazione della memorizzazione nella cache per le applicazioni ospitate. A ciascuna app viene applicata una configurazione cache per ottimizzare le prestazioni.

Il 13 agosto 2024, Amplify ha rilasciato miglioramenti all'efficienza della memorizzazione nella cache per le applicazioni. Per ulteriori informazioni, consulta Miglioramenti della memorizzazione nella cache CDN per migliori prestazioni delle app con hosting. AWS Amplify

La tabella seguente riassume il supporto di Amplify per comportamenti specifici di memorizzazione nella cache prima e dopo il rilascio dei miglioramenti della memorizzazione nella cache.

Comportamento della cache	Supporto precedente	Con miglioramenti della memorizzazione nella cache
Puoi aggiungere intestazioni personalizzate per un'app nella console Amplify o in un file. `customHeaders.yaml` Una delle intestazioni che puoi sovrascrivere è. `Cache-Control` Per ulteriori informazioni, consulta Impostazione di intestazioni personalizzate per un'app Amplify.	Sì	Sì
Amplify rispetta le intestazioni definite in `customHeaders.yaml` un file e hanno `Cache-Control` la precedenza sulle impostazioni della cache predefinite di Amplify.	Sì	Sì

Comportamento della cache	Supporto precedente	Con miglioramenti della memorizzazione nella cache
Amplify rispetta le intestazioni impostate all'interno `Cache-Control` del framework di un'applicazione per le rotte dinamiche (ad esempio, le rotte SSR Next.js). Se nel file dell'app è impostata un'`Cache-Control` intestazione, questa ha la precedenza sulle impostazioni del `customHeaders.yaml` file. `next.config.js`	Sì	Sì
Ogni nuova distribuzione CI/CD dell'app cancella la cache.	Sì	Sì
Puoi attivare la modalità prestazioni per un'app.	Sì	No L'impostazione della modalità prestazioni non è più disponibile nella console Amplify. Tuttavia, puoi creare un'`Cache-Control` intestazione che imposta la direttiva `s-maxage`. Per istruzioni, consultare Utilizzo dell'intestazione Cache-Control per aumentare le prestazioni dell'app.

La tabella seguente elenca le modifiche ai valori predefiniti per impostazioni specifiche della cache.

Impostazione della cache	Valore predefinito precedente	Valore predefinito con miglioramenti della memorizzazione nella cache
Durata della cache per le risorse statiche	Due secondi	Un anno
Durata della cache per le risposte del proxy inverso	Due secondi	Zero secondi (nessuna memorizzazione nella cache)
Tempo massimo di vita (TTL)	Dieci minuti	Un anno

Per ulteriori informazioni su come Amplify determina la configurazione di memorizzazione nella cache da applicare a un'applicazione e istruzioni sulla gestione della configurazione delle chiavi della cache, consulta i seguenti argomenti.

Argomenti

- In che modo Amplify applica la configurazione della cache a un'app
- Gestione dei cookie della chiave cache
- Utilizzo dell'intestazione Cache-Control per aumentare le prestazioni dell'app

In che modo Amplify applica la configurazione della cache a un'app

Per gestire la memorizzazione nella cache della tua app, Amplify determina il tipo di contenuto che viene fornito esaminando il tipo di piattaforma dell'app e le regole di riscrittura. Per `Compute` le app, Amplify esamina anche le regole di routing nel manifesto di distribuzione.

> **Note**
>
> Il tipo di piattaforma dell'app viene impostato da Amplify Hosting durante la distribuzione. Un'app SSG (statica) è impostata sul tipo di piattaforma. `WEB` Un'app SSR (Next.js 12 o successiva) è impostata sul tipo di piattaforma. `WEB_COMPUTE`

Amplify identifica i seguenti quattro tipi di contenuti e applica la policy di cache gestita specificata.

Statico

Il contenuto fornito dalle app con la `WEB` piattaforma o i percorsi statici in un'app. `WEB_COMPUTE`

Questo contenuto utilizza la politica Amplify-StaticContent della cache.

Ottimizzazione delle immagini

Le immagini servite dai `ImageOptimization` percorsi in un'`WEB_COMPUTE`app.

Questo contenuto utilizza la politica Amplify-ImageOptimization della cache.

Calcolo

Il contenuto servito dai `Compute` percorsi in un'`WEB_COMPUTE`app. Ciò include tutti i contenuti renderizzati lato server (SSR).

Questo contenuto utilizza la politica Amplify-Default o Amplify-DefaultNoCookies cache a seconda del `cacheConfig.type` valore impostato su Amplify. `App`

Proxy inverso

Il contenuto servito dai percorsi che corrispondono a una regola personalizzata di riscrittura del proxy inverso. Per ulteriori informazioni sulla creazione di questa regola personalizzata, consulta il Riscrittura inversa del proxy capitolo Utilizzo dei reindirizzamenti.

Questo contenuto utilizza la politica Amplify-Default o Amplify-DefaultNoCookies cache a seconda del `cacheConfig.type` valore impostato su Amplify. `App`

Comprensione delle politiche di cache gestita di Amplify

Amplify utilizza le seguenti politiche di cache gestita predefinite per ottimizzare la configurazione predefinita della cache per le applicazioni ospitate.

- Amplify-Default
- Amplify-DefaultNoCookies
- Amplify-ImageOptimization
- Amplify-StaticContent

Impostazioni della politica di cache gestita di Amplify-Default

Visualizza questa politica nella console CloudFront

Questa policy è progettata per l'utilizzo con un'origine che è una Web App AWS Amplify.

Questa policy ha le seguenti impostazioni:

- TTL minimo = 0 secondi
- TTL massimo: 31536000 secondi (un anno)
- TTL di default = 0 secondi
- Intestazioni incluse nella chiave cache:
  - `Authorization`
  - `Accept`
  - `CloudFront-Viewer-Country`
  - `Host`
- Cookie inclusi nella chiave cache: tutti i cookie sono inclusi.
- Stringhe di query incluse nella chiave cache: tutte le stringhe di query sono incluse.
- Impostazione cache degli oggetti compressi: Gzip e Brotli abilitati.

Amplify: impostazioni della politica della cache gestita da Amplify DefaultNoCookies

Visualizza questa politica nella console CloudFront

Questa policy è progettata per l'utilizzo con un'origine che è una Web App AWS Amplify.

Questa policy ha le seguenti impostazioni:

- TTL minimo = 0 secondi
- TTL massimo: 31536000 secondi (un anno)
- TTL di default = 0 secondi
- Intestazioni incluse nella chiave cache:
  - `Authorization`
  - `Accept`
  - `CloudFront-Viewer-Country`
  - `Host`
- Cookie inclusi nella chiave cache: non sono inclusi cookie.

- Stringhe di query incluse nella chiave cache: tutte le stringhe di query sono incluse.
- Impostazione cache degli oggetti compressi: Gzip e Brotli abilitati.

Amplify: impostazioni della politica della cache gestita da Amplify ImageOptimization

Visualizza questa politica nella console CloudFront

Questa policy è progettata per l'utilizzo con un'origine che è una Web App AWS Amplify.

Questa policy ha le seguenti impostazioni:

- TTL minimo = 0 secondi
- TTL massimo: 31536000 secondi (un anno)
- TTL di default = 0 secondi
- Intestazioni incluse nella chiave cache:
  - `Authorization`
  - `Accept`
  - `Host`
- Cookie inclusi nella chiave cache: non sono inclusi cookie.
- Stringhe di query incluse nella chiave cache: tutte le stringhe di query sono incluse.
- Impostazione cache degli oggetti compressi: Gzip e Brotli abilitati.

Amplify: impostazioni della politica della cache gestita da Amplify StaticContent

Visualizza questa politica nella console CloudFront

Questa policy è progettata per l'utilizzo con un'origine che è una Web App AWS Amplify.

Questa policy ha le seguenti impostazioni:

- TTL minimo = 0 secondi
- TTL massimo: 31536000 secondi (un anno)
- TTL di default = 0 secondi
- Intestazioni incluse nella chiave cache:
  - `Authorization`

- `Host`
- Cookie inclusi nella chiave cache: non sono inclusi cookie.
- Stringhe di query incluse nella chiave della cache: non sono incluse stringhe di query.
- Impostazione cache degli oggetti compressi: Gzip e Brotli abilitati.

Gestione dei cookie della chiave cache

Quando distribuisci la tua app su Amplify, puoi scegliere se includere o escludere i cookie nella chiave cache. Nella console Amplify, questa impostazione è specificata nella pagina Custom headers and cache utilizzando l'interruttore delle impostazioni del tasto Cache. Per istruzioni, consultare Inclusione o esclusione dei cookie dalla chiave cache.

Includi i cookie nella chiave cache

Con questa impostazione, Amplify sceglie automaticamente una configurazione cache ottimale per l'app in base al tipo di contenuto che viene fornito. È necessario scegliere esplicitamente questo tipo di configurazione della cache.

Se si utilizza il SDKs o il AWS CLI, questa impostazione corrisponde `cacheConfig.type` all'impostazione `AMPLIFY_MANAGED` con `CreateApp` o `UpdateApp` APIs.

Escludi i cookie dalla chiave cache

Questa è la configurazione predefinita della cache. Questa configurazione della cache è simile alla `AMPLIFY_MANAGED` configurazione, tranne per il fatto che esclude tutti i cookie dalla chiave cache.

La scelta di escludere i cookie dalla chiave cache può comportare migliori prestazioni della cache. Tuttavia, prima di scegliere questa configurazione della cache, è importante considerare se l'app utilizza i cookie per fornire contenuti dinamici.

Se si utilizza il SDKs o il AWS CLI, questa impostazione corrisponde all'impostazione `cacheConfig.type` di to `AMPLIFY_MANAGED_NO_COOKIES` con `CreateApp` o `UpdateApp` APIs.

Per ulteriori informazioni sulla chiave della cache, consulta Understand the cache key nella Amazon CloudFront Developer Guide;.

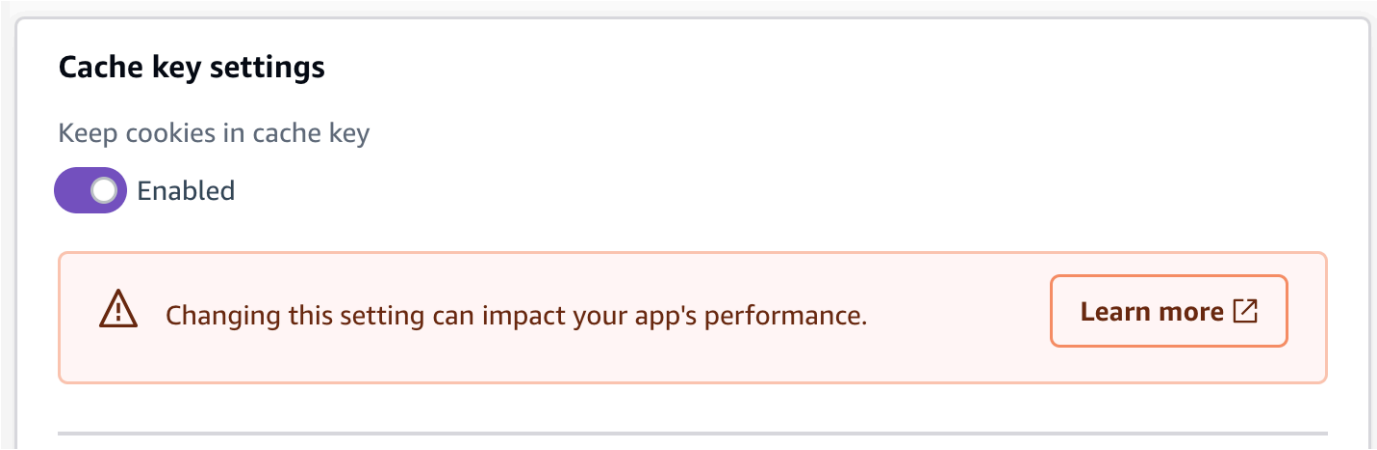
Inclusione o esclusione dei cookie dalla chiave cache

Puoi impostare la configurazione dei cookie della chiave cache per un'app nella console Amplify oppure SDKs. AWS CLI

Utilizza la procedura seguente per specificare se includere o escludere i cookie dalla chiave cache quando distribuisci una nuova app utilizzando la console Amplify.

Per impostare la configurazione dei cookie della chiave cache durante la distribuzione di un'app su Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella pagina Tutte le app, scegli Crea nuova app.
3. Nella pagina Inizia a creare con Amplify, scegli il tuo provider di repository Git, quindi scegli Avanti.
4. Nella pagina Aggiungi ramo del repository, procedi come segue:
   a. Seleziona il nome del repository da connettere.
   b. Seleziona il nome del ramo del repository da connettere.
   c. Scegli Next (Successivo).
5. Se l'app richiede un ruolo di servizio IAM, puoi consentire ad Amplify Hosting compute di creare automaticamente un ruolo di servizio per te oppure puoi specificare un ruolo che hai creato tu.
   - Per consentire ad Amplify di creare automaticamente un ruolo e associarlo alla tua app:
     - Scegli Crea e utilizza un nuovo ruolo di servizio.
   - Per allegare un ruolo di servizio creato in precedenza:
     a. Scegli Usa un ruolo di servizio esistente.
     b. Seleziona il ruolo da utilizzare dall'elenco.
6. Scegli Impostazioni avanzate, quindi individua la sezione Impostazioni chiave della cache.
7. Scegli Mantieni i cookie nella chiave cache o Rimuovi i cookie dalla chiave cache. La schermata seguente mostra l'interruttore delle impostazioni del tasto Cache nella console.



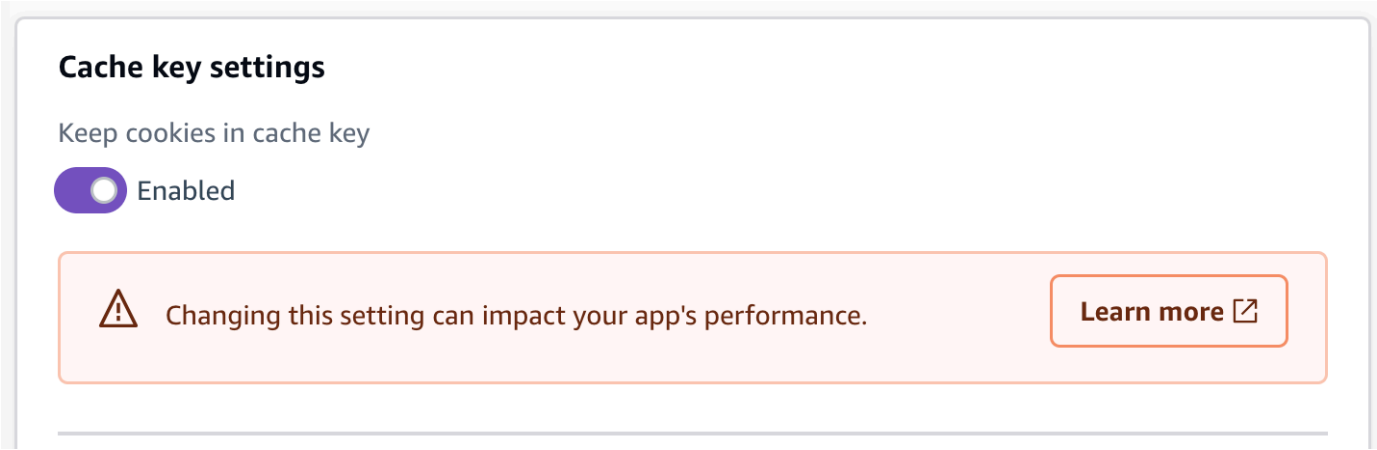
8. Scegli Next (Successivo).
9. Nella pagina Revisione, scegli Salva e distribuisci.

Modifica della configurazione dei cookie della chiave cache per un'app

Puoi modificare la configurazione dei cookie della chiave cache per un'app già distribuita su Amplify. Utilizza la seguente procedura per modificare se includere o escludere i cookie dalla chiave cache per un'app che utilizza la console Amplify.

Per modificare la configurazione dei cookie della chiave cache per un'app distribuita

1. Accedi AWS Management Console e apri la console Amplify.
2. Nella pagina Tutte le app, scegli l'applicazione che desideri aggiornare.
3. Nel pannello di navigazione, scegli Hosting, quindi scegli Intestazioni e cache personalizzate.
4. Nella pagina Intestazioni e cache personalizzate, individua la sezione Impostazioni della chiave cache e scegli Modifica.
5. Scegli Mantieni i cookie nella chiave cache o Rimuovi i cookie dalla chiave cache. La schermata seguente mostra l'interruttore delle impostazioni del tasto Cache nella console.



6. Scegli Save (Salva).

Utilizzo dell'intestazione Cache-Control per aumentare le prestazioni dell'app

L'architettura di hosting predefinita di Amplify ottimizza l'equilibrio tra prestazioni di hosting e disponibilità dell'implementazione. Per la maggior parte dei clienti, consigliamo di utilizzare l'architettura predefinita.

Se hai bisogno di un controllo più preciso sulle prestazioni di un'app, puoi impostare manualmente l'`Cache-Control`intestazione HTTP per ottimizzare le prestazioni di hosting mantenendo i contenuti memorizzati nella cache all'estremità della rete di distribuzione dei contenuti (CDN) per un intervallo più lungo.

Le `Cache-Control` intestazioni `max-age` e le `s-maxage` direttive HTTP influiscono sulla durata della memorizzazione nella cache dei contenuti dell'app. La `max-age` direttiva indica al browser per quanto tempo (in secondi) desiderate che il contenuto rimanga nella cache prima di essere aggiornato dal server di origine. La `s-maxage` direttiva sostituisce `max-age` e consente di specificare per quanto tempo (in secondi) il contenuto deve rimanere sull'edge CDN prima che venga aggiornato dal server di origine.

Le app ospitate con Amplify rispettano `Cache-Control` le intestazioni inviate dall'origine, a meno che non le sovrascriviate con intestazioni personalizzate da voi definite. Amplify `Cache-Control` applica solo intestazioni personalizzate per risposte di successo con un codice di stato. `200 OK` Ciò impedisce che le risposte agli errori vengano memorizzate nella cache e inviate ad altri utenti che effettuano la stessa richiesta.

Puoi modificare manualmente la `s-maxage` direttiva per avere un maggiore controllo sulle prestazioni e sulla disponibilità di implementazione della tua app. Ad esempio, per modificare il periodo di tempo in cui i contenuti rimangono memorizzati nella cache periferica, puoi impostare manualmente il time to live (TTL) eseguendo l'aggiornamento `s-maxage` a un valore diverso dal valore predefinito 31536000 secondi (un anno).

Puoi definire intestazioni personalizzate per un'app nella sezione Intestazioni personalizzate della console Amplify. Per un esempio del formato, vedi. YAML Impostazione delle intestazioni personalizzate di Cache-Control

Utilizzate la seguente procedura per impostare la `s-maxage` direttiva in modo da mantenere i contenuti memorizzati nella cache della rete CDN per 24 ore.

Per impostare un'intestazione personalizzata Cache-Control

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui impostare intestazioni personalizzate.
3. Nel pannello di navigazione, scegli Hosting, Intestazioni personalizzate.
4. Nella pagina Intestazioni personalizzate, scegli Modifica.
5. Nella finestra Modifica intestazioni personalizzate, inserisci le informazioni per l'intestazione personalizzata come segue:
   a. Per`pattern`, inserisci **`**/*`** per tutti i percorsi.
   b. In `key`, immettere **`Cache-Control`**.
   c. In `value`, immettere **`s-maxage=86400`**.
6. Scegli Save (Salva).
7. Ridistribuisci l'app per applicare la nuova intestazione personalizzata.

Protezione antiinclinazione per le implementazioni Amplify

La protezione dall'inclinazione dell'implementazione è disponibile per le applicazioni Amplify per eliminare i problemi di distorsione della versione tra client e server nelle applicazioni Web. Quando applichi la protezione da inclinazione a un'applicazione Amplify, puoi assicurarti che i tuoi client interagiscano sempre con la versione corretta degli asset lato server, indipendentemente dal momento in cui si verifica una distribuzione.

L'inclinazione della versione è una sfida comune per gli sviluppatori web. Si verifica quando un browser Web esegue una versione obsoleta di un'applicazione e il server ne esegue una nuova. Questa discrepanza può causare comportamenti imprevedibili, errori e un'esperienza negativa per l'utente dell'applicazione. La funzionalità di protezione dalla distorsione dell'implementazione Amplify associa i client in esecuzione sui browser Web a una distribuzione specifica. Ciò garantisce che Amplify serva sempre le risorse per quella particolare implementazione, mantenendo il client e il server sincronizzati.

La funzione di protezione dall'inclinazione di Amplify può ridurre gli errori per gli utenti dell'applicazione quando si rilasciano nuove distribuzioni. Può anche migliorare l'esperienza degli sviluppatori riducendo il tempo dedicato alla gestione dei problemi di compatibilità con le versioni precedenti e successive.

Dettagli della funzionalità di protezione asimmetrica:

Tipi di applicazione supportati

È possibile aggiungere la protezione dall'inclinazione alle applicazioni statiche e SSR create con qualsiasi framework supportato da Amplify. Le applicazioni possono essere distribuite da un repository Git o da una distribuzione manuale.

Non è possibile aggiungere una protezione dall'inclinazione a un'applicazione distribuita `WEB_DYNAMIC` sulla piattaforma (Next.js versione 11 o precedente).

Durata

Per le applicazioni statiche, Amplify offre una settimana di implementazioni. Per le applicazioni SSR, garantiamo una protezione dall'inclinazione per un massimo di otto implementazioni precedenti.

Costo

L'aggiunta di una protezione da asimmetria a un'applicazione non prevede costi aggiuntivi.

Considerazione delle prestazioni

Quando la protezione da inclinazione è abilitata per un'applicazione, Amplify deve aggiornare le configurazioni della cache CDN. Pertanto, dovresti aspettarti che la prima implementazione dopo aver abilitato la protezione da skew richieda fino a dieci minuti.

Argomenti

- Configurazione della protezione dall'inclinazione dell'implementazione per un'applicazione Amplify
- Come funziona la protezione dall'inclinazione

Configurazione della protezione dall'inclinazione dell'implementazione per un'applicazione Amplify

È possibile aggiungere o rimuovere la protezione dall'inclinazione della distribuzione per un'applicazione utilizzando la console Amplify, il, AWS Command Line Interface o il. SDKs La funzionalità viene applicata a livello di filiale. Solo le nuove distribuzioni, effettuate dopo aver abilitato la protezione dall'inclinazione per una filiale, saranno protette dall'inclinazione.

Per aggiungere o rimuovere la protezione dall'inclinazione della distribuzione utilizzando AWS CLI o SDKs, utilizza i campi e. `CreateBranch.enableSkewProtection` `UpdateBranch.enableSkewProtection` Per ulteriori informazioni, consulta CreateBranche UpdateBranchnella documentazione di riferimento dell'API Amplify.

Se desideri rimuovere una distribuzione specifica in modo che non venga più fornita, utilizza l'`DeleteJob`API. Per ulteriori informazioni, consulta la DeleteJobdocumentazione di riferimento dell'API Amplify.

Al momento, puoi abilitare la protezione dall'inclinazione solo su un'applicazione già distribuita su Amplify Hosting. Usa le seguenti istruzioni per aggiungere una protezione da inclinazione a un ramo utilizzando la console Amplify.

Abilita la protezione dall'inclinazione per il ramo di un'applicazione Amplify

1. Accedi a AWS Management Console e apri la console Amplify all'indirizzo. https://console.aws.amazon.com/amplify/
2. Nella pagina Tutte le app, scegli il nome dell'app distribuita su cui attivare la protezione da inclinazione.

3. Nel pannello di navigazione, scegli Impostazioni app, quindi scegli Impostazioni Branch.
4. Nella sezione Succursali scegli il nome del ramo da aggiornare.
5. Nel menu Azioni, scegli Abilita protezione dall'inclinazione.
6. Nella finestra di conferma scegli Conferma. La protezione da asimmetria è ora abilitata per il ramo.
7. Ridistribuisci il ramo dell'applicazione. Solo le distribuzioni effettuate dopo l'attivazione della protezione da inclinazione sono protette dall'inclinazione.

Utilizza le seguenti istruzioni per rimuovere la protezione da inclinazione da un ramo di un'applicazione utilizzando la console Amplify.

Rimuovere la protezione da asimmetria da un ramo di un'applicazione Amplify

1. Accedi a AWS Management Console e apri la console Amplify all'indirizzo. https://console.aws.amazon.com/amplify/
2. Nella pagina Tutte le app, scegli il nome dell'app distribuita da cui rimuovere la protezione da inclinazione.
3. Nel pannello di navigazione, scegli Impostazioni app, quindi scegli Impostazioni Branch.
4. Nella sezione Succursali scegli il nome del ramo da aggiornare.
5. Nel menu Azioni, scegli Disabilita la protezione dall'inclinazione. La protezione dall'inclinazione è ora disabilitata per la filiale e verranno pubblicati solo i contenuti più recenti.

Come funziona la protezione dall'inclinazione

Nella maggior parte dei casi, il comportamento predefinito del cookie _dpl soddisferà le tue esigenze di protezione dagli skew. Tuttavia, nei seguenti scenari avanzati, la protezione dall'inclinazione è abilitata meglio utilizzando i parametri `X-Amplify-Dpl` header e query. `dpl`

- Caricamento del sito Web in più schede del browser contemporaneamente
- Utilizzo di addetti all'assistenza

Amplify valuta la richiesta in arrivo nel seguente ordine quando determina il contenuto da fornire al client:

1. **`X-Amplify-Dpl`**intestazione: le applicazioni possono utilizzare questa intestazione per indirizzare le richieste a una specifica distribuzione Amplify. Questa intestazione della richiesta può essere impostata utilizzando il valore di. `process.env.AWS_AMPLIFY_DEPLOYMENT_ID`
2. **`dpl`**parametro di query: le applicazioni Next.js imposteranno automaticamente il parametro di query _dpl per le richieste agli asset con impronte digitali (file con estensione js e css).
3. _dpl cookie: è l'impostazione predefinita per tutte le applicazioni protette da skew. Per un browser specifico, lo stesso cookie viene inviato per ogni scheda o istanza del browser che interagisce con un dominio.

   Tieni presente che se in diverse schede del browser sono caricate versioni diverse di un sito Web, il cookie _dpl viene condiviso da tutte le schede. In questo scenario, non è possibile ottenere una protezione totale dall'inclinazione con il cookie _dpl e dovresti prendere in considerazione l'utilizzo dell'intestazione per la protezione dall'inclinazione. `X-Amplify-Dpl`

X-Amplify-Dpl esempio di intestazione

L'esempio seguente mostra il codice per una pagina SSR Next.js che accede alla protezione dall'inclinazione tramite l'intestazione. `X-Amplify-Dpl` La pagina esegue il rendering del contenuto in base a uno dei suoi percorsi api. La distribuzione da servire al percorso api viene specificata utilizzando l'`X-Amplify-Dpl`intestazione, che è impostata sul valore di. `process.env.AWS_AMPLIFY_DEPLOYMENT_ID`

```
import { useEffect, useState } from 'react';

export default function MyPage({deploymentId}) {
    const [data, setData] = useState(null);

    useEffect(() => {
        fetch('/api/hello', {
            headers: {
                'X-Amplify-Dpl': process.env.AWS_AMPLIFY_DEPLOYMENT_ID
            },
        })
        .then(res => res.json())
        .then(data => setData(data))
        .catch(error => console.error("error", error))
    }, []);

    return <div>
```

```
        {data ? JSON.stringify(data) : "Loading ... " }
    </div>
}
```

Monitoraggio di un'applicazione Amplify

AWS Amplify offre le seguenti funzionalità per il monitoraggio delle applicazioni ospitate:

- CloudWatch metriche: Amplify emette parametri tramite CloudWatch Amazon che puoi utilizzare per monitorare il traffico, gli errori, il trasferimento dei dati e la latenza per le tue applicazioni.
- Registri di accesso: Amplify fornisce registri di accesso con informazioni dettagliate sulle richieste fatte all'applicazione.
- CloudTrail registrazione: Amplify è integrato AWS CloudTrail e fornisce una registrazione delle azioni intraprese da un utente, ruolo o servizio in Amplify. AWS È possibile visualizzare questi eventi nella console. CloudTrail

Argomenti

- Monitoraggio di un'applicazione Amplify con Amazon CloudWatch
- Recupero e analisi dei log di accesso per un'applicazione Amplify
- Registrazione delle chiamate API Amplify utilizzando AWS CloudTrail

Monitoraggio di un'applicazione Amplify con Amazon CloudWatch

AWS Amplify è integrato con Amazon CloudWatch, consentendoti di monitorare i parametri per le tue applicazioni Amplify quasi in tempo reale e di creare allarmi che inviano notifiche quando una metrica supera una soglia impostata. Per ulteriori informazioni su come funziona il CloudWatch servizio, consulta la Amazon CloudWatch User Guide.

CloudWatch Metriche supportate

Amplify supporta CloudWatch sette metriche nel namespace per il monitoraggio `AWS/AmplifyHosting` del traffico, degli errori, del trasferimento dei dati, della latenza e dei token di richiesta per le tue app. Queste metriche sono aggregate a intervalli di un minuto. CloudWatch le metriche di monitoraggio sono gratuite e non influiscono sulle quote di servizio. CloudWatch

La tabella seguente descrive ogni metrica supportata ed elenca le statistiche più rilevanti. Non tutte le statistiche disponibili sono applicabili a ogni metrica.

Parametro	Descrizione
Richieste	Il numero totale di richieste di spettatori ricevute dalla tua app. La statistica più rilevante è`Sum`. Utilizza la `Sum` statistica per ottenere il numero totale di richieste.
BytesDownloaded	La quantità totale di dati trasferiti dall'app (scaricati), espressa in byte `GETHEAD`, dagli utenti per e dalle richieste. `OPTIONS` La statistica più rilevante è. `Sum`
BytesUploaded	La quantità totale di dati trasferiti nell'app (caricati) in byte per qualsiasi richiesta, incluse le intestazioni. Amplify non ti addebita alcun costo per i dati caricati nelle tue applicazioni. La statistica più rilevante è. `Sum`
4xxErrors	Il numero di richieste che hanno restituito un errore nell'intervallo del codice di stato HTTP 400-499. La statistica più rilevante è. `Sum` Usa la `Sum` statistica per ottenere il numero totale di occorrenze di questi errori.
5xxErrors	Il numero di richieste che hanno restituito un errore nell'intervallo del codice di stato HTTP 500-599. La statistica più rilevante è. `Sum` Usa la `Sum` statistica per ottenere il numero totale di occorrenze di questi errori.

Parametro	Descrizione
Latenza	Il tempo necessario per arrivare al primo byte, in secondi. Questo è il tempo totale tra il momento in cui Amplify Hosting riceve una richiesta e il momento in cui restituisce una risposta alla rete. Ciò non include la latenza di rete rilevata quando una risposta raggiunge il dispositivo dello spettatore. Le statistiche più rilevanti sono `AverageMaximum`,`Minimum`,`p10`,`p50`, `p90p95`, `ep100`. Utilizza la `Average` statistica per valutare le latenze previste.

Parametro	Descrizione
TokensConsumed	I token di richiesta utilizzati dalla tua app. La `Sum` statistica rappresenta il consumo totale di token di richiesta. Puoi confrontare questa statistica con la tua attuale quota di `Request tokens per second` servizi per determina re se è necessario richiedere un aumento della quota per evitare potenziali limitazioni durante future manifestazioni ad alto traffico. La `Average` statistica rappresenta il consumo di token di richiesta nelle ore normali e di punta. Un maggiore consumo di token comporta in genere un tempo di trasferimento del primo byte (TTFB) più lungo. Pertanto, puoi utilizzar e questa statistica per valutare la latenza dell'applicazione. Se la latenza è scarsa, potete migliorare il downstream APIs per ridurre il consumo di token ed evitare le limitazioni che si possono verificare quando il consumo di token supera la quota di servizio dell'applicazione. `Request tokens per second` Per ulteriori informazioni sulla quota di servizio, consulta. `Request tokens per second` Quote del servizio Amplify Hosting

Amplify fornisce le seguenti dimensioni metriche. CloudWatch

Dimensione	Descrizione
App	I dati metrici sono forniti dall'app.
Account AWS	I dati metrici vengono forniti in tutte le app di. Account AWS

Accesso alle metriche CloudWatch

Puoi accedere alle CloudWatch metriche direttamente dalla console Amplify utilizzando la procedura seguente.

> **Note**
>
> Puoi anche accedere alle CloudWatch metriche all'indirizzo. AWS Management Console https://console.aws.amazon.com/cloudwatch/

Per accedere alle metriche nella console Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui desideri visualizzare le metriche.
3. Nel riquadro di navigazione, scegli Monitoraggio, quindi scegli Metriche.

Creazione di allarmi CloudWatch

Puoi creare CloudWatch allarmi nella console Amplify che inviano notifiche quando vengono soddisfatti criteri specifici. Un allarme controlla una singola CloudWatch metrica e invia una notifica Amazon Simple Notification Service quando la metrica supera la soglia per un determinato numero di periodi di valutazione.

Puoi creare allarmi più avanzati che utilizzano espressioni matematiche metriche nella console o utilizzando il. CloudWatch CloudWatch APIs Ad esempio, puoi creare un allarme che ti avvisa quando la percentuale 4xxErrors supera il 15% per tre periodi consecutivi. Per ulteriori informazioni, consulta Creazione di un CloudWatch allarme basato su un'espressione matematica metrica nella Amazon CloudWatch User Guide.

CloudWatch Il prezzo standard si applica agli allarmi. Per ulteriori informazioni, consulta i CloudWatchprezzi di Amazon.

Utilizzare la procedura seguente per creare un allarme nella console Amplify.

Per creare un CloudWatch allarme per una metrica Amplify

1. Accedi AWS Management Console e apri la console Amplify.

2. Scegli l'app su cui vuoi impostare una sveglia.
3. Nel pannello di navigazione, scegli Monitoraggio, quindi scegli Allarmi.
4. Nella pagina Allarmi, scegli Crea allarme.
5. Nella finestra Crea allarme, configura la sveglia come segue:
   a. Per Metric, scegli il nome della metrica da monitorare dall'elenco.
   b. In Nome dell'allarme, inserisci un nome significativo per l'avviso. Ad esempio, se stai monitorando le richieste, puoi assegnare un nome all'allarme**`HighTraffic`**. Il nome deve contenere solo caratteri ASCII.
   c. Per Configurare le notifiche, esegui una delle seguenti operazioni:
      - i. Scegli Nuovo per configurare un nuovo argomento Amazon SNS.
        ii. Per Indirizzo e-mail, inserisci l'indirizzo e-mail del destinatario delle notifiche.
        iii. Scegli Aggiungi nuovo indirizzo email per aggiungere altri destinatari.
      - i. Scegli Existing per riutilizzare un argomento di Amazon SNS.
        ii. Per l'argomento SNS, seleziona il nome di un argomento Amazon SNS esistente dall'elenco.
   d. Per Whenever the Statistic of Metric, imposta le condizioni per l'allarme come segue:
      i. Specificate se la metrica deve essere maggiore, minore o uguale al valore di soglia.
      ii. Specificare il valore della soglia.
      iii. Specificate il numero di periodi di valutazione consecutivi che devono trovarsi nello stato di allarme per richiamare l'allarme.
      iv. Specificare la durata del periodo di valutazione.
   e. Scegli Conferma.

> **Note**
>
> Ogni destinatario Amazon SNS specificato riceve un'e-mail di conferma da AWS Notifications. L'e-mail contiene un link che il destinatario deve seguire per confermare l'iscrizione e ricevere notifiche.

Accesso ai CloudWatch log per le app SSR

Amplify invia informazioni sul tuo runtime SSR ad CloudWatch Amazon Logs nel tuo. Account AWS Quando distribuisci un'app SSR su Amplify Hosting Compute, l'app richiede un ruolo di servizio IAM che Amplify assume quando chiama altri servizi per tuo conto. Puoi consentire ad Amplify Hosting compute di creare automaticamente un ruolo di servizio per te oppure puoi specificare un ruolo che hai creato.

Se scegli di consentire ad Amplify di creare un ruolo IAM per te, il ruolo avrà già le autorizzazioni per creare log. CloudWatch Se crei il tuo ruolo IAM, dovrai aggiungere le seguenti autorizzazioni alla tua policy per consentire ad Amplify di accedere ad Amazon Logs. CloudWatch

```
logs:CreateLogStream
logs:CreateLogGroup
logs:DescribeLogGroups
logs:PutLogEvents
```

Per ulteriori informazioni sull'aggiunta di un ruolo di servizio, consulta. Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend Per ulteriori informazioni sulla distribuzione di app renderizzate lato server, consulta. Implementazione di applicazioni renderizzate lato server con Amplify Hosting

È possibile visualizzare i registri di calcolo di Amplify Hosting per un'applicazione SSR nella console o nella CloudWatch console Amplify. Usa le seguenti istruzioni per visualizzare i log nella console Amplify.

Per visualizzare CloudWatch i log di un'applicazione SSR nella console Amplify

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app SSR per visualizzare i CloudWatch log.
3. Nel pannello di navigazione, scegli Monitoraggio, quindi scegli Hosting compute logs.
4. Nella pagina Hosting compute logs, cerca e seleziona un gruppo di CloudWatch log per un ramo specifico.

Recupero e analisi dei log di accesso per un'applicazione Amplify

Amplify archivia i log di accesso per tutte le app ospitate in Amplify. I log di accesso contengono informazioni sulle richieste effettuate alle app ospitate. Amplify conserva tutti i log di accesso per

un'app fino a quando non elimini l'app. Tutti i log di accesso per un'app sono disponibili nella console Amplify. Tuttavia, ogni singola richiesta di log di accesso è limitata a un periodo di due settimane specificato dall'utente.

> **Warning**
>
> Non includere segreti, credenziali o dati sensibili URLs come parametri di percorso o di query. Questi valori sono visualizzabili in testo semplice nei log di accesso dell'applicazione Amplify.

Amplify non CloudFront riutilizza mai le distribuzioni tra clienti. Amplify CloudFront crea le distribuzioni in anticipo in modo da non dover attendere la creazione di CloudFront una distribuzione quando si distribuisce una nuova app. Prima che queste distribuzioni vengano assegnate a un'app Amplify, potrebbero ricevere traffico dai bot. Tuttavia, sono configurate per rispondere sempre come Non trovate prima di essere assegnate. Se i registri di accesso dell'app contengono voci relative a un periodo di tempo precedente alla creazione dell'app, tali voci sono correlate a questa attività.

> **Important**
>
> Ti consigliamo di utilizzare i log per comprendere la natura delle richieste per il tuo contenuto e non come resoconto completo di tutte le richieste. Amplify fornisce i log di accesso con la massima diligenza possibile. È possibile che la voce di log per una specifica richiesta venga distribuita molto tempo dopo l'elaborazione effettiva della richiesta e, in rari casi, che non venga distribuita affatto. Quando una voce di registro viene omessa dai log di accesso, il numero di voci nei log di accesso non corrisponderà all'utilizzo che appare nei report di fatturazione e utilizzo. AWS

Recupero dei log di accesso di un'app

Usa la seguente procedura per recuperare i log di accesso per un'app Amplify.

Per visualizzare i log di accesso

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app per cui desideri visualizzare i registri di accesso.
3. Nel riquadro di navigazione, scegli Monitoraggio, quindi scegli Registri di accesso.

4. Scegli Modifica intervallo di tempo.
5. Nella finestra Modifica intervallo di tempo, procedi come segue.
   a. Per Data di inizio, specifica il primo giorno dell'intervallo di due settimane per cui recuperare i log.
   b. Per Ora di inizio, scegli l'ora del primo giorno per iniziare il recupero del registro.
   c. Scegli Conferma.
6. La console Amplify visualizza i registri per l'intervallo di tempo specificato nella sezione Registri di accesso. Scegli Scarica per salvare i log in formato CSV.

Analisi dei log di accesso

Per analizzare i log di accesso, puoi archiviare i file CSV in un bucket Amazon S3. Un modo per analizzare i log di accesso consiste nell'utilizzare Athena. Athena è un servizio di interrogazione interattivo che può aiutarti ad analizzare i dati per AWS i servizi. Puoi seguire le step-by-step istruzioni riportate qui per creare una tabella. Una volta creata la tabella, puoi interrogare i dati come segue.

```
SELECT SUM(bytes) AS total_bytes
FROM logs
WHERE "date" BETWEEN DATE '2018-06-09' AND DATE '2018-06-11'
LIMIT 100;
```

Registrazione delle chiamate API Amplify utilizzando AWS CloudTrail

AWS Amplify è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in Amplify. CloudTrail acquisisce tutte le chiamate API per Amplify come eventi. Le chiamate acquisite includono chiamate dalla console Amplify e chiamate in codice alle operazioni dell'API Amplify. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Amplify. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni CloudTrail raccolte, è possibile determinare la richiesta che è stata effettuata ad Amplify, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

Amplify le informazioni in CloudTrail

CloudTrail è abilitato per impostazione predefinita sul tuo AWS account. Quando si verifica un'attività in Amplify, tale attività viene registrata in CloudTrail un evento insieme ad AWS altri eventi di servizio nella cronologia degli eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS . Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi nella Guida per l'AWS CloudTrail utente.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amplify, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le Regioni AWS . Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consultare gli argomenti seguenti nella Guida per l'utente di AWS CloudTrail :

- Creare un percorso per il tuo account AWS
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le operazioni di Amplify vengono registrate e sono CloudTrail documentate nel Console API Reference,AWS Amplify nell'Amplify Admin UI API Referencee nell'AWS Amplify UI Builder API Reference. Ad esempio, le chiamate alle operazioni e generano voci nei file di `CreateApp` registro`DeleteApp`. `DeleteBackendEnvironment` CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- La richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- La richiesta è stata effettuata con credenziali di sicurezza temporanee per un ruolo o un utente federato.
- La richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, vedete l'elemento CloudTrail userIdentity nella Guida per l'AWS CloudTrail utente.

Informazioni sulle voci dei file di registro di Amplify

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra il funzionamento di AWS Amplify Console API Reference. `ListApps`

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::444455556666:user/Mary_Major",
        "accountId": "444455556666",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Mary_Major",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-01-12T05:48:10Z"
            }
        }
    },
    "eventTime": "2021-01-12T06:47:29Z",
    "eventSource": "amplify.amazonaws.com",
    "eventName": "ListApps",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.898
 Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.275-b01
 java/1.8.0_275 vendor/Oracle_Corporation",
    "requestParameters": {
```

```
        "maxResults": "100"
    },
    "responseElements": null,
    "requestID": "1c026d0b-3397-405a-95aa-aa43aexample",
    "eventID": "c5fca3fb-d148-4fa1-ba22-5fa63example",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "444455556666"
}
```

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'operazione AWS Amplify Admin UI API Reference. `ListBackendJobs`

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::444455556666:user/Mary_Major",
        "accountId": "444455556666",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Mary_Major",
        "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-01-13T00:47:25Z"
            }
        }
    },
    "eventTime": "2021-01-13T01:15:43Z",
    "eventSource": "amplifybackend.amazonaws.com",
    "eventName": "ListBackendJobs",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.255",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.898
 Linux/4.9.230-0.1.ac.223.84.332.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.275-b01
 java/1.8.0_275 vendor/Oracle_Corporation",
    "requestParameters": {
        "appId": "d23mv2oexample",
```

```
        "backendEnvironmentName": "staging"
    },
    "responseElements": {
        "jobs": [
            {
                "appId": "d23mv2oexample",
                "backendEnvironmentName": "staging",
                "jobId": "ed63e9b2-dd1b-4bf2-895b-3d5dcexample",
                "operation": "CreateBackendAuth",
                "status": "COMPLETED",
                "createTime": "1610499932490",
                "updateTime": "1610500140053"
            },
            {
                "appId": "d23mv2oexample",
                "backendEnvironmentName": "staging",
                "jobId": "06904b10-a795-49c1-92b7-185dfexample",
                "operation": "CreateBackend",
                "status": "COMPLETED",
                "createTime": "1610499657938",
                "updateTime": "1610499704458"
            }
        ],
        "appId": "d23mv2oexample",
        "backendEnvironmentName": "staging"
    },
    "requestID": "7adfabd6-98d5-4b11-bd39-c7deaexample",
    "eventID": "68769310-c96c-4789-a6bb-68b52example",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "444455556666"
}
```

Utilizzo dei ruoli IAM con le applicazioni Amplify

Un ruolo IAM è un'identità IAM con autorizzazioni specifiche. Le autorizzazioni del ruolo determinano ciò che l'identità può e non può fare. AWS Puoi creare ruoli IAM nel tuo Account AWS e utilizzarli per delegare le autorizzazioni ad Amplify Hosting. Per saperne di più sui ruoli, consulta i ruoli IAM nella IAM User Guide.

Puoi utilizzare i seguenti tipi di ruoli IAM per concedere ad Amplify Hosting le autorizzazioni necessarie per eseguire azioni per tuo conto o eseguire codice di calcolo che accede ad altre risorse. AWS

Ruolo di servizio IAM

Amplify assume questo ruolo per eseguire azioni per tuo conto. Questo ruolo è richiesto per le applicazioni con risorse di backend.

Ruolo IAM SSR Compute

Consente a un'applicazione renderizzata lato server (SSR) di accedere in modo sicuro a risorse specifiche. AWS

CloudWatch Ruolo IAM SSR Logs

Quando distribuisci un'app SSR, l'app richiede un ruolo di servizio IAM che Amplify assume per consentire ad Amplify di accedere ad Amazon Logs. CloudWatch

Argomenti

- Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend
- Aggiungere un ruolo SSR Compute per consentire l'accesso alle risorse AWS
- Aggiungere un ruolo di servizio con autorizzazioni per accedere ai registri CloudWatch

Aggiungere un ruolo di servizio con autorizzazioni per distribuire risorse di backend

Amplify richiede le autorizzazioni per distribuire risorse di backend con il front-end. Per eseguire questa operazione, è possibile utilizzare un ruolo di servizio. Un ruolo di servizio è il ruolo AWS

Identity and Access Management (IAM) che fornisce ad Amplify Hosting le autorizzazioni per distribuire, creare e gestire i backend per tuo conto.

Quando crei una nuova app che richiede un ruolo di servizio IAM, puoi consentire ad Amplify Hosting di creare automaticamente un ruolo di servizio per te oppure puoi selezionare un ruolo IAM che hai già creato. In questa sezione, imparerai come creare un ruolo del servizio Amplify con autorizzazioni amministrative dell'account e che consenta esplicitamente l'accesso diretto alle risorse richieste dalle applicazioni Amplify per distribuire, creare e gestire i backend.

Creazione di un ruolo di servizio Amplify nella console IAM

Per creare un ruolo di servizio

1. Apri la console IAM e scegli Ruoli dalla barra di navigazione a sinistra, quindi scegli Crea ruolo.
2. Nella sezione Seleziona un'identità attendibile scegli Servizio AWS . Per Use Case, seleziona Amplify - Backend Deployment, quindi scegli Avanti.
3. Nella pagina Add permissions (Aggiungi autorizzazioni), scegli Next (Successivo).
4. Nella pagina Nome, visualizzazione e creazione, per Nome del ruolo inserisci un nome significativo, ad esempio. **`AmplifyConsoleServiceRole-AmplifyRole`**
5. Accetta tutte le impostazioni predefinite e scegli Crea ruolo.
6. Torna alla console Amplify per assegnare il ruolo alla tua app.
   - Se stai distribuendo una nuova app, procedi come segue:
     a. Aggiorna l'elenco dei ruoli di servizio.
     b. Seleziona il ruolo che hai appena creato. Per questo esempio, dovrebbe assomigliare a AmplifyConsoleServiceRole- AmplifyRole.
     c. Scegli Avanti e segui i passaggi per completare la distribuzione dell'app.
   - Se hai un'app esistente, procedi come segue:
     a. Nel riquadro di navigazione, scegli Impostazioni app, quindi scegli Ruoli IAM.
     b. Nella pagina dei ruoli IAM, nella sezione Service role, scegli Modifica.
     c. Nella pagina del ruolo di servizio, seleziona il ruolo appena creato dall'elenco dei ruoli di servizio.
     d. Scegli Save (Salva).
7. Amplify ora dispone delle autorizzazioni per distribuire risorse di backend per la tua app.

Modifica della politica di fiducia di un ruolo di servizio per evitare la confusione del vice

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Per ulteriori informazioni, consulta Prevenzione del problema "confused deputy" tra servizi.

Attualmente, la politica di fiducia predefinita per il ruolo di `Amplify-Backend Deployment` servizio applica le chiavi relative alle condizioni di contesto `aws:SourceAccount` globale per evitare che il `aws:SourceArn` sostituto sia confuso. Tuttavia, se in precedenza hai creato un `Amplify-Backend Deployment` ruolo nel tuo account, puoi aggiornare la politica di fiducia del ruolo aggiungendo queste condizioni per evitare che il sostituto sia confuso.

Usa l'esempio seguente per limitare l'accesso alle app del tuo account. Sostituisci la regione e l'ID dell'applicazione nell'esempio con le tue informazioni.

```
"Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
```

Per istruzioni su come modificare la politica di fiducia per un ruolo utilizzando la AWS Management Console, consulta Modifying a role (console) nella IAM User Guide.

Aggiungere un ruolo SSR Compute per consentire l'accesso alle risorse AWS

Questa integrazione consente di assegnare un ruolo IAM al servizio Amplify SSR Compute per consentire all'applicazione renderizzata lato server (SSR) di accedere in modo sicuro a risorse specifiche in base alle autorizzazioni del ruolo. AWS Ad esempio, puoi consentire alle funzioni di calcolo SSR della tua app di accedere in modo sicuro ad altri AWS servizi o risorse, come Amazon Bedrock un bucket Amazon S3, in base alle autorizzazioni definite nel ruolo IAM assegnato.

Il ruolo IAM SSR Compute fornisce credenziali temporanee, eliminando la necessità di codificare credenziali di sicurezza di lunga durata nelle variabili di ambiente. L'utilizzo del ruolo IAM SSR Compute è in linea con le migliori pratiche di AWS sicurezza che prevedono la concessione di autorizzazioni con privilegi minimi e l'utilizzo di credenziali a breve termine quando possibile.

Le istruzioni riportate più avanti in questa sezione descrivono come creare una policy con autorizzazioni personalizzate e allegare la policy a un ruolo. Quando crei il ruolo, devi allegare una politica di fiducia personalizzata che dia ad Amplify il permesso di assumere il ruolo. Se la relazione di fiducia non è definita correttamente, riceverai un errore quando tenti di aggiungere il ruolo. La seguente politica di fiducia personalizzata concede ad Amplify il permesso di assumere il ruolo.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Puoi associare un ruolo IAM nel tuo Account AWS a un'applicazione SSR esistente utilizzando la console AWS SDKs Amplify o il. AWS CLI Il ruolo assegnato viene automaticamente associato al servizio di calcolo Amplify SSR, concedendogli le autorizzazioni specificate per accedere ad altre risorse. AWS Man mano che le esigenze dell'applicazione cambiano nel tempo, puoi modificare il ruolo IAM associato senza ridistribuire l'applicazione. Ciò offre flessibilità e riduce i tempi di inattività delle applicazioni.

> **Important**
>
> L'utente è responsabile della configurazione dell'applicazione per soddisfare gli obiettivi di sicurezza e conformità. Ciò include la gestione del ruolo SSR Compute, che deve essere configurato in modo da disporre del set minimo di autorizzazioni necessario per supportare il caso d'uso. Per ulteriori informazioni, consulta Gestione della sicurezza dei ruoli IAM SSR Compute.

Creazione di un ruolo SSR Compute nella console IAM

Prima di poter collegare un ruolo IAM SSR Compute a un'applicazione Amplify, il ruolo deve già esistere nella tua. Account AWS In questa sezione, imparerai come creare una policy IAM e collegarla a un ruolo che Amplify può assumere per accedere a risorse specifiche. AWS

Ti consigliamo di seguire la AWS best practice di concessione delle autorizzazioni con privilegi minimi durante la creazione di un ruolo IAM. Il ruolo IAM SSR Compute viene chiamato solo dalle funzioni di calcolo SSR e pertanto dovrebbe concedere solo le autorizzazioni necessarie per eseguire il codice.

È possibile utilizzare AWS Management Console, AWS CLI o SDKs per creare politiche in IAM. Per ulteriori informazioni, consulta Definire le autorizzazioni IAM personalizzate con le politiche gestite dai clienti nella Guida per l'utente IAM.

Le seguenti istruzioni mostrano come utilizzare la console IAM per creare una policy IAM che definisce le autorizzazioni da concedere al servizio Amplify Compute.

Per utilizzare l'editor di policy JSON della console IAM per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Scegliere Create Policy (Crea policy).
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Digitare o incollare un documento di policy JSON.
6. Una volta terminata l'aggiunta delle autorizzazioni alla policy, scegli Successivo.
7. Nella pagina Verifica e crea, digita i valori per Nome policy e Descrizione (facoltativa) per la policy che si sta creando. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.

8. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato una policy, utilizza le seguenti istruzioni per collegarla a un ruolo IAM.

Per creare un ruolo che conceda le autorizzazioni Amplify a risorse specifiche AWS

1. Accedi AWS Management Console e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione della console, selezionare Roles (Ruoli) e Crea ruolo.
3. Scegli il tipo di ruolo Custom trust policy (Policy di attendibilità personalizzata).
4. Nella sezione Politica di fiducia personalizzata, inserisci la politica di fiducia personalizzata per il ruolo. È necessaria una politica di fiducia per i ruoli e definisce i principi di cui ti fidi per assumere il ruolo.

   Copia e incolla la seguente politica di fiducia per concedere al servizio Amplify l'autorizzazione ad assumere questo ruolo.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "amplify.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

5. Risolvi eventuali avvisi di sicurezza, errori o avvertenze generali generati durante la convalida delle policy, quindi scegli Successivo.
6. Nella pagina Aggiungi autorizzazioni, cerca il nome della politica che hai creato nella procedura precedente e selezionala. Quindi scegli Successivo.

7. In Role name, (Nome ruolo), inserisci un nome. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **`PRODROLE`** che **`prodrole`**. Poiché altre AWS risorse potrebbero fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo che è stato creato.
8. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
9. (Facoltativo) Scegli Modifica nelle sezioni Fase 1: seleziona le entità attendibili o Fase 2: aggiungi autorizzazioni per modificare la policy personalizzata e le autorizzazioni per il ruolo.
10. Rivedere il ruolo e scegliere Crea ruolo.

Aggiungere un ruolo IAM SSR Compute a un'app Amplify

Dopo aver creato un ruolo IAM nel tuo Account AWS, puoi associarlo a un'app nella console Amplify.

Per aggiungere un ruolo SSR Compute a un'app nella console Amplify

1. Accedi a AWS Management Console e apri la console Amplify all'indirizzo. https://console.aws.amazon.com/amplify/
2. Nella pagina Tutte le app, scegli il nome dell'app a cui aggiungere un ruolo Compute.
3. Nel riquadro di navigazione, scegli Impostazioni app, quindi scegli ruoli IAM.
4. Nella sezione Compute role, scegli Modifica.
5. Nell'elenco Ruolo predefinito, cerca il nome del ruolo che desideri allegare e selezionalo. Per questo esempio, puoi scegliere il nome del ruolo creato nella procedura precedente. Per impostazione predefinita, il ruolo selezionato verrà associato a tutti i rami dell'app.

   Se la relazione di fiducia del ruolo non è definita correttamente, verrà visualizzato un errore e non sarà possibile aggiungere il ruolo.
6. (facoltativo) Se l'applicazione si trova in un archivio pubblico e utilizza la creazione automatica di branch o ha le anteprime web per le richieste pull abilitate, non è consigliabile utilizzare un ruolo a livello di app. Collega invece il ruolo Compute solo alle filiali che richiedono l'accesso a risorse specifiche. Per ignorare il comportamento predefinito a livello di app e assegnare un ruolo a un ramo specifico, procedi come segue:

   a. Per Branch, selezionate il nome del ramo da usare.

   b. Per Compute role, seleziona il nome del ruolo da associare al ramo.
7. Scegli, Salva.

Gestione della sicurezza dei ruoli IAM SSR Compute

La sicurezza è una responsabilità condivisa tra te AWS e te. L'utente è responsabile della configurazione dell'applicazione per soddisfare gli obiettivi di sicurezza e conformità. Ciò include la gestione del ruolo SSR Compute, che deve essere configurato in modo da disporre del set minimo di autorizzazioni necessario per supportare il caso d'uso. Le credenziali per il ruolo SSR Compute specificato sono immediatamente disponibili nel runtime della funzione SSR. Se il codice SSR espone queste credenziali, intenzionalmente, a causa di un bug o permettendo l'esecuzione di codice in modalità remota (RCE), un utente non autorizzato può accedere al ruolo SSR e alle relative autorizzazioni.

Quando un'applicazione in un archivio pubblico utilizza un ruolo SSR Compute e la creazione automatica di branch o anteprime web per le richieste pull, è necessario gestire con attenzione le filiali che possono accedere al ruolo. Ti consigliamo di non utilizzare un ruolo a livello di app. Invece, dovresti assegnare un ruolo Compute a livello di filiale. Ciò consente di concedere le autorizzazioni solo alle filiali che richiedono l'accesso a risorse specifiche.

Se le credenziali del tuo ruolo sono esposte, esegui le seguenti azioni per rimuovere tutti gli accessi alle credenziali del ruolo.

1. Revoca tutte le sessioni

   Per istruzioni su come revocare immediatamente tutte le autorizzazioni relative alle credenziali del ruolo, consulta Revoca delle credenziali di sicurezza temporanee del ruolo IAM.
2. Eliminare il ruolo dalla console Amplify

   Questa azione ha effetto immediato. Non è necessario ridistribuire l'applicazione.

Per eliminare un ruolo Compute nella console Amplify

1. Accedi a AWS Management Console e apri la console Amplify all'indirizzo. https://console.aws.amazon.com/amplify/
2. Nella pagina Tutte le app, scegli il nome dell'app da cui rimuovere il ruolo Compute.
3. Nel riquadro di navigazione, scegli Impostazioni app, quindi scegli ruoli IAM.
4. Nella sezione Compute role, scegli Modifica.
5. Per eliminare il ruolo predefinito, scegli la X a destra del nome del ruolo.
6. Scegli Save (Salva).

Aggiungere un ruolo di servizio con autorizzazioni per accedere ai registri CloudWatch

Amplify invia informazioni sul tuo runtime SSR ad CloudWatch Amazon Logs nel tuo. Account AWS Quando distribuisci un'app SSR, l'app richiede un ruolo di servizio IAM che Amplify assume quando chiama altri servizi per tuo conto. Puoi consentire ad Amplify Hosting compute di creare automaticamente un ruolo di servizio per te oppure puoi specificare un ruolo che hai creato.

Se scegli di consentire ad Amplify di creare un ruolo IAM per te, il ruolo avrà già le autorizzazioni per creare log. CloudWatch Se crei il tuo ruolo IAM, dovrai aggiungere le seguenti autorizzazioni alla tua policy per consentire ad Amplify di accedere ad Amazon Logs. CloudWatch

```
logs:CreateLogStream
logs:CreateLogGroup
logs:DescribeLogGroups
logs:PutLogEvents
```

Webhook unificati per repository Git

Amplify Hosting utilizza i webhook per avviare automaticamente una build dopo un nuovo commit nel tuo repository Git. La funzionalità webhook unificata migliora le integrazioni di Amplify con i provider Git e consente di connettere più applicazioni Amplify a un unico repository. Con i webhook unificati, Amplify ora utilizza un singolo webhook per regione per tutte le applicazioni associate nel tuo repository. Ad esempio, se il tuo repository è connesso ad applicazioni nelle regioni Stati Uniti orientali (Virginia settentrionale) e Stati Uniti occidentali (Oregon), avrai due webhook unificati.

Prima di questa versione, Amplify creava un nuovo webhook per ogni app associata a un repository. Se disponi di più app in un unico repository, potresti raggiungere i limiti dei webhook imposti dai singoli provider Git e impedirti di aggiungere altre app. Ciò era particolarmente impegnativo per i team che lavoravano in monorepos, dove più progetti erano presenti in un unico repository.

I webhook unificati offrono i seguenti vantaggi:

- Supera i limiti dei webhook del provider Git: puoi connettere tutte le app Amplify di cui hai bisogno a un unico repository.
- Supporto monorepo migliorato: hai più flessibilità ed efficienza quando lavori con monorepos, in cui più progetti condividono un unico repository.
- Gestione semplificata: la gestione di più app Amplify con un unico webhook del repository riduce la complessità e i potenziali punti di errore.
- Migliore integrazione del flusso di lavoro: puoi utilizzare i webhook assegnati dal tuo provider Git per altri flussi di lavoro essenziali nel processo di sviluppo.

Nozioni di base su webhook unificati

Creare una nuova app

Quando distribuisci una nuova applicazione su Amplify Hosting da un repository Git, la funzionalità webhook unificata viene implementata automaticamente per il tuo repository. Per istruzioni sulla creazione di una nuova applicazione, consultaGuida introduttiva alla distribuzione di un'app su Amplify Hosting.

Aggiornamento di un'app esistente

Per le applicazioni Amplify esistenti, è necessario ricollegare il repository Git all'applicazione per sostituire i webhook esistenti con un webhook unificato. Se hai già raggiunto il numero massimo di

webhook consentito dal tuo provider Git, la migrazione al webhook unificato potrebbe non riuscire. In questo caso, rimuovi manualmente almeno un webhook esistente prima di riconnetterti.

È possibile avere più applicazioni in un repository distribuite in regioni diverse. AWS Poiché le operazioni di Amplify sono basate sulla regione, la migrazione a un webhook unificato avviene solo per i webhook nella regione in cui hai ricollegato l'app Amplify. Di conseguenza, nel tuo repository potresti vedere sia webhook basati sugli ID delle applicazioni che webhook unificati basati sulla regione.

Utilizza le seguenti istruzioni per migrare un'app Amplify esistente su un webhook unificato.

Per migrare un'app Amplify esistente su un webhook unificato

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app che desideri migrare su un webhook unificato.
3. Nel pannello di navigazione, scegli Impostazioni app, quindi scegli Impostazioni Branch.
4. Nella pagina delle impostazioni di Branch, scegli Reconnetti repository.
5. Per verificare la corretta migrazione al webhook unificato, accedi alle impostazioni del webhook nel tuo repository Git. Dovresti vedere un singolo URL del webhook nel formato. `https://amplify-webhooks.`*`Region`*`.amazonaws.com/`*`git-provider`*

Sicurezza in Amplify

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS Amplify, consulta AWS Servizi nell'ambito del programma di conformitàAWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi Amplify. I seguenti argomenti mostrano come configurare Amplify per soddisfare i tuoi obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amplify.

Argomenti

- Identity and Access Management per Amplify
- Protezione dei dati in Amplify
- Convalida della conformità per AWS Amplify
- Sicurezza dell'infrastruttura in AWS Amplify
- Registrazione e monitoraggio degli eventi di sicurezza in Amplify
- Prevenzione del problema "confused deputy" tra servizi
- Le migliori pratiche di sicurezza per Amplify

Identity and Access Management per Amplify

AWS Identity and Access Management (IAM) è un programma Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse Amplify. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso con policy
- Come funziona Amplify con IAM
- Esempi di policy basate sull'identità per Amplify
- AWS politiche gestite per AWS Amplify
- Risoluzione dei problemi relativi all'identità e all'accesso di Amplify

Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amplify.

Utente del servizio: se utilizzi il servizio Amplify per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzioni di Amplify per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzione di Amplify, consulta. Risoluzione dei problemi relativi all'identità e all'accesso di Amplify

Amministratore del servizio: se sei responsabile delle risorse Amplify presso la tua azienda, probabilmente hai pieno accesso ad Amplify. È tuo compito determinare a quali funzionalità e risorse di Amplify devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con Amplify, consulta. Come funziona Amplify con IAM

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere politiche per gestire l'accesso ad Amplify. Per visualizzare esempi di policy basate

sull'identità di Amplify che puoi utilizzare in IAM, consulta. Esempi di policy basate sull'identità per Amplify

Autenticazione con identità

L'autenticazione è il modo in cui accedi utilizzando le tue credenziali di identità. AWS Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo consigliato per la firma delle richieste, consulta Signature Version 4 AWS per le richieste API nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA)AWS in IAM nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root

può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali

temporanee. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi passare da un ruolo utente a un ruolo IAM (console). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Create a role for a third-party identity provider (federation) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.

- Sessioni di accesso inoltrato (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.
- Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.
- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. Questa soluzione è preferibile alla memorizzazione delle chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella IAM User Guide.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo del servizio (SCPs): SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più di proprietà dell' Account AWS azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS Per ulteriori informazioni su Organizations and SCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.
- Politiche di controllo delle risorse (RCPs): RCPs sono politiche JSON che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le politiche IAM allegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations e RCPs, incluso un elenco di

Servizi AWS tale supporto RCPs, vedere Resource control policies (RCPs) nella Guida per l'AWS Organizations utente.

- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta la logica di valutazione delle policy nella IAM User Guide.

Come funziona Amplify con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amplify, scopri quali funzionalità IAM sono disponibili per l'uso con Amplify.

Funzionalità IAM che puoi utilizzare con Amplify

Funzionalità IAM	Supporto Amplify
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
ACLs	No
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì

Funzionalità IAM	Supporto Amplify
Inoltro delle sessioni di accesso (FAS)	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una visione di alto livello di come Amplify e AWS altri servizi funzionano con la maggior parte delle funzionalità IAM, AWS consulta i servizi che funzionano con IAM nella IAM User Guide.

Politiche basate sull'identità per Amplify

Supporta le policy basate su identità: sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate sull'identità per Amplify

Per visualizzare esempi di politiche basate sull'identità di Amplify, vedere. Esempi di policy basate sull'identità per Amplify

Politiche basate sulle risorse all'interno di Amplify

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket

Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Azioni politiche per Amplify

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per un elenco delle azioni Amplify, consulta Azioni definite AWS Amplify da nel Service Authorization Reference.

Le azioni politiche in Amplify utilizzano il seguente prefisso prima dell'azione:

```
amplify
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "amplify:action1",
      "amplify:action2"
         ]
```

Per visualizzare esempi di politiche basate sull'identità di Amplify, vedere. Esempi di policy basate sull'identità per Amplify

Risorse politiche per Amplify

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per un elenco dei tipi di risorse Amplify e ARNs relativi, vedere Tipi di risorse definiti AWS Amplify da nel Service Authorization Reference. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta la sezione Operazioni definite da AWS Amplify.

Per visualizzare esempi di politiche basate sull'identità di Amplify, vedere. Esempi di policy basate sull'identità per Amplify

Chiavi relative alle condizioni delle politiche per Amplify

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per un elenco delle chiavi di condizione Amplify, consulta Condition keys nel Service Authorization AWS Amplify Reference. Per sapere con quali azioni e risorse è possibile utilizzare una chiave di condizione, consulta Azioni definite da. AWS Amplify

Per visualizzare esempi di politiche basate sull'identità di Amplify, vedere. Esempi di policy basate sull'identità per Amplify

Elenchi di controllo degli accessi (ACLs) in Amplify

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Controllo degli accessi basato sugli attributi (ABAC) con Amplify

Supporta ABAC (tag nelle policy): parzialmente

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In, questi attributi sono chiamati AWS tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con Amplify

Supporta le credenziali temporanee: sì

Alcune Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-On (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Passaggio da un ruolo utente a un ruolo IAM (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Sessioni di accesso diretto per Amplify

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per Amplify

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità Amplify. Modifica i ruoli di servizio solo quando Amplify fornisce indicazioni in tal senso.

Ruoli collegati ai servizi per Amplify

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per i dettagli sulla creazione o la gestione di ruoli collegati ai servizi, consulta i AWS servizi che funzionano con IAM nella IAM User Guide. Trova un servizio nella tabella che include un `Yes`

nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il link Sì per visualizzare la documentazione sui ruoli collegati ai servizi per quel servizio.

Esempi di policy basate sull'identità per Amplify

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare le risorse Amplify. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente IAM.

Per i dettagli sulle azioni e sui tipi di risorse definiti da Amplify, incluso il formato di per ogni tipo di ARNs risorsa, vedere Azioni, risorse e chiavi di condizione nel Service Authorization AWS Amplify Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console Amplify
- Consentire agli utenti di visualizzare le loro autorizzazioni

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse Amplify nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni

che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.

- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Protezione dell'accesso API con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console Amplify

Per accedere alla AWS Amplify console, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amplify presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso o l' AWS CLI API. AWS Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Con il rilascio di Amplify Studio, l'eliminazione di un'app o di un backend richiede entrambe le autorizzazioni. `amplify` `amplifybackend` Se una policy IAM fornisce solo `amplify` autorizzazioni, un utente riceve un errore di autorizzazione quando tenta di eliminare un'app. Se sei un amministratore che scrive policy, determina le autorizzazioni corrette da concedere agli utenti che devono eseguire azioni di eliminazione.

Per garantire che gli utenti e i ruoli possano ancora utilizzare la console Amplify, collega anche la policy Amplify o gestita alle `ConsoleAccess` entità `ReadOnly` AWS . Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l'API o. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
```

```
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politiche gestite per AWS Amplify

Una politica AWS gestita è una politica autonoma creata e amministrata da. AWS AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

Policy gestita da AWS: AdministratorAccess -Amplify

È possibile allegare la policy `AdministratorAccess-Amplify` alle identità IAM. Amplify attribuisce inoltre questa politica a un ruolo di servizio che consente ad Amplify di eseguire azioni per conto dell'utente.

Quando si distribuisce un backend nella console Amplify, è necessario creare un `Amplify-Backend Deployment` ruolo di servizio utilizzato da Amplify per creare e gestire le risorse. AWS IAM associa la policy gestita al ruolo di servizio. `AdministratorAccess-Amplify` `Amplify-Backend Deployment`

Questa politica concede le autorizzazioni amministrative dell'account, consentendo esplicitamente l'accesso diretto alle risorse richieste dalle applicazioni Amplify per creare e gestire i backend.

Dettagli dell'autorizzazione

Questa policy fornisce l'accesso a più servizi, incluse le azioni AWS IAM. Queste azioni consentono alle identità con questa policy di essere utilizzate AWS Identity and Access Management per creare altre identità con qualsiasi autorizzazione. Ciò consente l'aumento delle autorizzazioni e questa politica deve essere considerata efficace quanto la politica. `AdministratorAccess`

Questa politica concede l'autorizzazione all'`iam:PassRole`azione per tutte le risorse. Ciò è necessario per supportare la configurazione dei pool di utenti di Amazon Cognito.

Per visualizzare le autorizzazioni per questa politica, vedere AdministratorAccess-Amplify nel Managed Policy Reference.AWS

AWS politica gestita: AmplifyBackendDeployFullAccess

È possibile allegare la policy `AmplifyBackendDeployFullAccess` alle identità IAM.

Questa politica concede ad Amplify le autorizzazioni di accesso completo per distribuire le risorse di backend Amplify utilizzando il. AWS Cloud Development Kit (AWS CDK) Le autorizzazioni vengono trasferite ai ruoli che dispongono delle autorizzazioni politiche necessarie. AWS CDK `AdministratorAccess`

Dettagli dell'autorizzazione

Questa politica include le autorizzazioni per eseguire le seguenti operazioni.

- `Amplify`— Recupera i metadati sulle applicazioni distribuite.
- `AWS CloudFormation`— Crea, aggiorna ed elimina gli stack gestiti da Amplify.
- `SSM`— Crea, aggiorna ed elimina l'archivio dei parametri e i parametri SSM gestiti da Amplify. `String` `SecureString`
- `AWS AppSync`— Aggiorna e recupera AWS AppSync lo schema, il resolver e le risorse funzionali. Lo scopo è supportare la funzionalità di hotswapping della sandbox di seconda generazione.
- `Lambda`— Aggiorna e recupera la configurazione per le funzioni gestite da Amplify. Lo scopo è supportare la funzionalità di hotswapping della sandbox di seconda generazione.

  Recupera i tag di una funzione Lambda. Lo scopo è supportare le funzioni Lambda definite dai clienti.

- `Amazon S3`— Recupera le risorse di distribuzione di Amplify.
- `AWS Security Token Service`— Consente alla AWS Cloud Development Kit (AWS CDK) CLI di assumere il ruolo di distribuzione.
- `Amazon RDS`— Leggi i metadati di istanze DB, cluster e proxy.
- `Amazon EC2`— Leggi le informazioni sulla zona di disponibilità per una sottorete.
- `CloudWatch Logs`— Recupera i log per la funzione Lambda di un cliente. Lo scopo è consentire a un ambiente sandbox di sviluppo su cloud Amplify di trasmettere i log di una funzione Lambda al terminale di un cliente.

Per vedere le autorizzazioni per questa policy, consulta AmplifyBackendDeployFullAccess nella Guida di riferimento sulle policy gestite da AWS .

Amplify gli aggiornamenti alle policy gestite AWS

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amplify da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per gli avvisi automatici sulle modifiche apportate alla pagina, iscriviti al feed RSS alla pagina Cronologia dei documenti per AWS Amplify.

Modifica	Descrizione	Data
AmplifyBackendDeployFullAccess: aggiornamento a una policy esistente	Aggiungi l'accesso in lettura alla `logs:FilterLogEvents` risorsa per consentire ad Amplify di trasmettere i log dalle funzioni in cui è stato creato un gruppo di log personalizzato. Questa è un'estensione della capacità esistente di trasmettere in streaming i log di una funzione Lambda.	14 novembre 2024
AmplifyBackendDeployFullAccess: aggiornamento a una policy esistente	Aggiungi l'accesso in lettura alle `logs:FilterLogEvents` risorse `lambda:Li`	18 luglio 2024

Modifica	Descrizione	Data
	`stTags` e per supportare le funzioni Lambda definite dai clienti. Queste autorizzazioni consentono a un ambiente sandbox di sviluppo cloud Amplify di trasmettere i log di una funzione Lambda al terminale di un cliente.	
AmplifyBackendDeployFullAccess: aggiornamento a una policy esistente	Aggiungi l'accesso in lettura alla `arn:aws:ssm:*:*:parameter/cdk-bootstrap/*` risorsa per consentire ad Amplify di rilevare la versione bootstrap CDK nell'account di un cliente.	31 maggio 2024

Modifica	Descrizione	Data
AmplifyBackendDeployFullAccess: aggiornamento a una policy esistente	Aggiungi una nuova dichiarazione `AmplifyDiscoverRDSVpcConfig` politica con autorizzazioni di EC2 sola lettura di Amazon RDS e Amazon in base alle condizioni delle risorse e dell'account. Queste autorizzazioni supportano il comando Amplify Gen `npx amplify generate schema-from-database` 2 che consente ai clienti di generare uno schema di dati Typescript da un database SQL esistente. Aggiungi`rds:DescribeDBProxies` ,,, e le `rds:DescribeDBInstances` autorizzazioni. `rds:DescribeDBClusters` `rds:DescribeDBSubnetGroups` `ec2:DescribeSubnets` Il `npx amplify generate schema-from-database` comando richiede queste autorizzazioni per verificare se un host DB specificato è ospitato in Amazon RDS e generare automaticamente la configurazione Amazon VPC necessaria per fornire le altre risorse necessarie e per configurare un' AWS	17 aprile 2024

Modifica	Descrizione	Data
	AppSync API supportata da un database SQL.	
AmplifyBackendDeployFullAccess: aggiornamento a una policy esistente	Aggiungi l'azione `cloudformation:DeleteStack` politica per supportare l'eliminazione dello stack quando viene chiamata l'`DeleteBranch` API. Aggiungi l'azione `lambda:GetFunction` politica per supportare le funzioni di hotswap. Aggiungi l'azione `lambda:UpdateFunctionConfiguration` politica per supportare gli aggiornamenti alla funzione Lambda.	5 aprile 2024
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi le autorizzazioni `cloudformation:TagResource` e le `cloudformation:UnTagResource` autorizzazioni a cui supportare le chiamate. AWS CloudFormation APIs	4 aprile 2024

Modifica	Descrizione	Data
AmplifyBackendDeployFullAccess: aggiornamento a una policy esistente	Aggiungi l'azione `lambda:InvokeFunction` politica per supportare l' AWS Cloud Development Kit (AWS CDK) hotswap. AWS CDK Effettua chiamate dirette a una funzione Lambda per eseguire l'hotswap degli asset Amazon S3. Aggiungi l'azione `lambda:UpdateFunctionCode` politica per supportare le funzioni di hotswapping.	02 gennaio 2024
AmplifyBackendDeployFullAccess: aggiornamento a una policy esistente	Aggiungi azioni politiche a supporto dell'`UpdateApiKey` operazione. Ciò è necessario per consentire una corretta distribuzione dell'app dopo l'uscita e il riavvio della sandbox senza eliminare risorse.	17 novembre 2023
AmplifyBackendDeployFullAccess: aggiornamento a una policy esistente	Aggiungi l'`amplify:GetBackendEnvironment` autorizzazione per supportare la distribuzione dell'app Amplify.	6 novembre 2023
AmplifyBackendDeployFullAccess: nuova policy	Amplify ha aggiunto una nuova policy con le autorizzazioni minime richieste per distribuire le risorse di backend Amplify.	8 ottobre 2023

Modifica	Descrizione	Data
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi l'`ecr:DescribeRepositories` autorizzazione richiesta dall'interfaccia CLI (Command Line Interface) di Amplify.	1 giugno 2023

Modifica	Descrizione	Data
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi un'azione politica per supportare la rimozione dei tag da una risorsa. AWS AppSync Aggiungi un'azione politica per supportare la risorsa Amazon Polly. Aggiungi un'azione politica per supportare l'aggiorn amento della configurazione del OpenSearch dominio. Aggiungi un'azione politica per supportare la rimozione di tag da un AWS Identity and Access Management ruolo. Aggiungi un'azione politica per supportare la rimozione di tag da una risorsa Amazon DynamoDB. Aggiungi le `cloudfron t:GetCloudFrontOri ginAccessIdentityC onfig` autorizzazioni `cloudfront:GetClou dFrontOriginAccess Identity` e al blocco di istruzioni per `CLISDKCalls` supportare i flussi di lavoro di pubblicazione e hosting di Amplify.	24 febbraio 2023

Modifica	Descrizione	Data
	Aggiungi l'`s3:PutBucketPublicAccessBlock` autorizzazione al blocco di `CLIManageviaCFNPolicy` istruzioni per consentire il AWS CLI supporto della best practice di sicurezza di Amazon S3 di abilitare la funzionalità Amazon S3 Block Public Access su bucket interni. Aggiungi l'`cloudformation:DescribeStacks` autorizzazione al blocco di istruzioni per supportare `CLISDKCalls` il recupero degli AWS CloudFormation stack dei clienti in caso di nuovi tentativi nel processore di backend Amplify per evitare di duplicare le esecuzioni se uno stack è in fase di aggiornamento. `CLICloudformationPolicy` Aggiungi l'`cloudformation:ListStacks` autorizzazione al blocco di istruzioni. Questa autorizzazione è necessaria per supportare pienamente l' CloudFormation DescribeStacks azione.	

Modifica	Descrizione	Data
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi azioni politiche per consentire alla funzional ità di rendering lato server Amplify di inviare le metriche CloudWatch delle applicazioni a un cliente. Account AWS	30 agosto 2022
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi azioni politiche per bloccare l'accesso pubblico al bucket Amazon S3 di distribuz ione Amplify.	27 aprile 2022
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi un'azione per consentire ai clienti di eliminare le app renderizz ate sul lato server (SSR). Ciò consente inoltre di eliminare correttamente la CloudFront distribuzione corrispondente. Aggiungi un'azione per consentire ai clienti di specificare una funzione Lambda diversa per gestire gli eventi da un'origine di eventi esistente utilizzando la CLI Amplify. Con queste modifiche, AWS Lambda sarà in grado di eseguire l'azione. UpdateEventSourceMapping	17 aprile 2022
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi un'azione politica per abilitare le azioni di Amplify UI Builder su tutte le risorse.	2 dicembre 2021

Modifica	Descrizione	Data
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi azioni politiche per supportare la funzionalità di autenticazione Amazon Cognito che utilizza provider di identità social. Aggiungi un'azione politica per supportare i livelli Lambda. Aggiungi un'azione politica per supportare la categoria Amplify Storage.	8 novembre 2021

Modifica	Descrizione	Data
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Aggiungi azioni Amazon Lex per supportare la categoria Amplify Interactions. Aggiungi le azioni Amazon Rekognition per supportare la categoria Amplify Predictions. Aggiungi un'azione Amazon Cognito per supportare la configurazione MFA sui pool di utenti di Amazon Cognito. Aggiungi CloudFormation azioni al supporto. AWS CloudFormation StackSets Aggiungi azioni Amazon Location Service per supportar e la categoria Amplify Geo. Aggiungi un'azione Lambda per supportare i layer Lambda in Amplify. Aggiungi azioni di CloudWatc h registro per supportare gli eventi. CloudWatch Aggiungi azioni Amazon S3 per supportare la categoria Amplify Storage. Aggiungi azioni politiche per supportare le app renderizzate lato server (SSR).	27 settembre 2021

Modifica	Descrizione	Data
AdministratorAccess-Amplify: aggiornamento a una politica esistente	Consolida tutte le azioni Amplify in un'unica azione. `amplify:*` Aggiungi un'azione Amazon S3 per supportare la crittogra fia dei bucket Amazon S3 dei clienti. Aggiungi azioni limite di autorizzazione IAM per supportare le app Amplify con limiti di autorizzazione abilitati. Aggiungi azioni Amazon SNS per supportare la visualizz azione dei numeri di telefono di origine e la visualizzazione, la creazione, la verifica e l'eliminazione dei numeri di telefono di destinazione. Amplify Studio: aggiungi Amazon Cognito AWS Lambda, IAM e azioni politiche per abilitare la gestione dei backend nella console Amplify AWS CloudFormation e Amplify Studio. Aggiungi una dichiarazione di policy AWS Systems Manager (SSM) per gestire i segreti dell'ambiente Amplify.	28 luglio 2021

Modifica	Descrizione	Data
	Aggiungi un' AWS CloudFormation `ListResources` azione per supportare i layer Lambda per le app Amplify.	
Amplify ha iniziato a tracciare le modifiche	Amplify ha iniziato a tenere traccia delle modifiche per AWS le sue politiche gestite.	28 luglio 2021

Risoluzione dei problemi relativi all'identità e all'accesso di Amplify

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con Amplify e IAM.

Argomenti

- Non sono autorizzato a eseguire un'azione in Amplify
- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire a persone esterne al mio AWS account di accedere alle mie risorse Amplify

Non sono autorizzato a eseguire un'azione in Amplify

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `amplify:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplify:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `amplify:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Con il rilascio di Amplify Studio, l'eliminazione di un'app o di un backend richiede entrambe le autorizzazioni. `amplify` `amplifybackend` Se un amministratore ha scritto una policy IAM che fornisce solo `amplify` autorizzazioni, riceverai un errore di autorizzazione quando tenti di eliminare un'app.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` IAM tenta di utilizzare la console per eliminare una *`example-amplify-app`* risorsa fittizia ma non dispone delle autorizzazioni. `amplifybackend:`*`RemoveAllBackends`*

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 amplifybackend;:RemoveAllBackends on resource: example-amplify-app
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa *`example-amplify-app`* utilizzando l'azione `amplifybackend:`*`RemoveAllBackends`*.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di trasferire un ruolo ad Amplify.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in Amplify. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne al mio AWS account di accedere alle mie risorse Amplify

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amplify supporta queste funzionalità, consulta. Come funziona Amplify con IAM
- Per scoprire come fornire l'accesso alle tue risorse su tutto Account AWS ciò che possiedi, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Protezione dei dati in Amplify

AWS Amplify è conforme al modello di responsabilità AWS condivisa Modello di responsabilità , che include regolamenti e linee guida per la protezione dei dati. AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i AWS servizi. AWS mantiene il controllo sui dati ospitati su questa infrastruttura, compresi i controlli di configurazione di sicurezza per la gestione dei contenuti e dei dati personali dei clienti. AWS i clienti e i partner APN, che agiscono in qualità di titolari o incaricati del trattamento dei dati, sono responsabili di tutti i dati personali che inseriscono nel AWS Cloud.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In

questo modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere il proprio lavoro. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno AWS dei servizi.
- Utilizza i servizi di sicurezza gestiti avanzati, ad esempio Amazon Macie, che aiutano a individuare e proteggere i dati personali archiviati in Amazon S3.

Ti consigliamo di non inserire mai informazioni identificative sensibili, ad esempio i numeri di account dei clienti, in campi a formato libero, ad esempio un campo Name (Nome). Ciò include quando lavori con Amplify o AWS altri servizi utilizzando la console, l'API AWS CLI o. AWS SDKs Tutti i dati che inserisci in Amplify o in altri servizi potrebbero essere raccolti per essere inclusi nei registri di diagnostica. Quando fornisci un URL a un server esterno, non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta a tale server.

Per ulteriori informazioni sulla protezione dei dati, consulta il post del blog AWS Modello di responsabilità condivisa e GDPR su AWS Security Blog.

Crittografia a riposo

La crittografia dei dati inattivi si riferisce alla protezione dei dati da accessi non autorizzati crittografando i dati durante l'archiviazione. Amplify crittografa gli artefatti di build di un'app per impostazione predefinita utilizzando AWS KMS keys Amazon S3 che sono gestiti da. AWS Key Management Service

Amplify utilizza CloudFront Amazon per offrire la tua app ai tuoi clienti. CloudFront utilizza SSDs che sono crittografati per i punti di presenza di edge location (POPs) e volumi EBS crittografati per le cache Edge regionali (). RECs Il codice e la configurazione della funzione in CloudFront Functions sono sempre archiviati in un formato crittografato SSDs sulla posizione POPs periferica crittografata e in altre posizioni di archiviazione utilizzate da. CloudFront

Crittografia in transito

La crittografia in transito si riferisce alla protezione dei dati da qualsiasi intercettazione mentre si spostano tra gli endpoint di comunicazione. Amplify Hosting fornisce la crittografia per i dati in transito

per impostazione predefinita. Tutte le comunicazioni tra i clienti e Amplify e tra Amplify e le sue dipendenze a valle sono protette tramite connessioni TLS firmate utilizzando il processo di firma Signature Version 4. Tutti gli endpoint di Amplify Hosting utilizzano certificati SHA-256 gestiti da. AWS Private Certificate AuthorityPer ulteriori informazioni, consulta il processo di firma della versione 4 di Signature e What is. AWS Private Certificate Authority

Gestione delle chiavi di crittografia

AWS Key Management Service (KMS) è un servizio gestito per la creazione e il controllo delle AWS KMS keys chiavi di crittografia utilizzate per crittografare i dati dei clienti. AWS Amplify genera e gestisce chiavi crittografiche per crittografare i dati per conto dei clienti. Non ci sono chiavi di crittografia da gestire.

Convalida della conformità per AWS Amplify

I revisori esterni valutano la sicurezza e la conformità nell' AWS Amplify ambito di più programmi di AWS conformità. Questi includono SOC, PCI, ISO, HIPAA, MTCS, C5, K-ISMS, ENS High, OSPAR, HITRUST CSF e FINMA.

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta la sezione Scope by Compliance Program Servizi AWS in Scope by Compliance ProgramServizi AWS e che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Governance e conformità per la sicurezza: queste guide all'implementazione di soluzioni illustrano considerazioni relative all'architettura e i passaggi per implementare le funzionalità di sicurezza e conformità.
- Riferimenti sui servizi conformi ai requisiti HIPAA: elenca i servizi HIPAA idonei. Non tutti Servizi AWS sono idonei alla normativa HIPAA.
- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.

- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- Valutazione delle risorse con regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, come lo standard PCI DSS, soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente l' AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Sicurezza dell'infrastruttura in AWS Amplify

In quanto servizio gestito, AWS Amplify è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amplify attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Registrazione e monitoraggio degli eventi di sicurezza in Amplify

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amplify e delle altre soluzioni. AWS AWS fornisce i seguenti strumenti di monitoraggio per guardare Amplify, segnalare quando qualcosa non va e intraprendere azioni automatiche quando appropriato:

- Amazon CloudWatch monitora in tempo reale AWS le tue risorse e le applicazioni su AWS cui esegui. Puoi raccogliere e tenere traccia dei parametri, creare dashboard personalizzati e impostare allarmi che ti avvisano o intraprendono azioni quando una determinata metrica raggiunge una soglia da te specificata. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue istanze Amazon Elastic Compute Cloud EC2 (Amazon) e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni sull'utilizzo di CloudWatch metriche e allarmi con Amplify, consulta. Monitoraggio di un'applicazione Amplify
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di registro da EC2 istanze Amazon e altre fonti. AWS CloudTrail CloudWatch I log possono monitorare le informazioni contenute nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.
- AWS CloudTrailacquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo AWS account e invia i file di log a un bucket Amazon Simple Storage Service (Amazon S3) da te specificato. Puoi identificare quali utenti e account hanno effettuato la chiamata AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta Registrazione delle chiamate API Amplify utilizzando AWS CloudTrail.
- Amazon EventBridge è un servizio di bus eventi senza server che semplifica la connessione delle applicazioni con dati provenienti da una varietà di fonti. EventBridge fornisce un flusso di dati in tempo reale dalle tue applicazioni, applicazioni Software-as-a-Service (SaaS) e AWS servizi e indirizza tali dati verso destinazioni come. AWS Lambda Ciò consente di monitorare gli eventi che si verificano nei servizi e creare architetture basate sugli eventi. Per ulteriori informazioni, consulta la Amazon EventBridge User Guide.

Prevenzione del problema "confused deputy" tra servizi

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel AWS, l'impersonificazione tra servizi può portare al confuso problema del vice. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account.

Si consiglia di utilizzare `aws:SourceArn`le chiavi di contesto della condizione `aws:SourceAccount`globale nelle politiche delle risorse per limitare le autorizzazioni che AWS Amplify forniscono un altro servizio alla risorsa. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` devono utilizzare lo stesso ID account nella stessa istruzione di policy.

Il valore di `aws:SourceArn` deve essere l'ARN della filiale dell'app Amplify. Specificate questo valore nel formato. `arn:`*`Partition`*`:amplify:`*`Region`*`:`*`Account`*`:apps/`*`AppId`*`/branches/`*`BranchName`*

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, è necessario utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (`*`) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

L'esempio seguente mostra una politica di fiducia dei ruoli che puoi applicare per limitare l'accesso a qualsiasi app Amplify nel tuo account e prevenire il problema del confuso vice. Per utilizzare questa politica, sostituisci il testo in corsivo rosso nella politica di esempio con le tue informazioni.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
```

```
    "Effect": "Allow",
    "Principal": {
      "Service": [
          "amplify.me-south-1.amazonaws.com",
          "amplify.eu-south-1.amazonaws.com",
          "amplify.ap-east-1.amazonaws.com",
          "amplifybackend.amazonaws.com",
          "amplify.amazonaws.com"
        ]
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

L'esempio seguente mostra una politica di fiducia dei ruoli che puoi applicare per limitare l'accesso a una determinata app Amplify nel tuo account e prevenire il problema del confuso vice. Per utilizzare questa politica, sostituisci il testo in corsivo rosso nella politica di esempio con le tue informazioni.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": [
          "amplify.me-south-1.amazonaws.com",
          "amplify.eu-south-1.amazonaws.com",
          "amplify.ap-east-1.amazonaws.com",
          "amplifybackend.amazonaws.com",
          "amplify.amazonaws.com"
        ]
```

```
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:amplify:us-east-1:123456789012:apps/d123456789/
branches/*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Le migliori pratiche di sicurezza per Amplify

Amplify offre una serie di funzionalità di sicurezza da considerare durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per il tuo ambiente, considerale come consigli utili più che prescrizioni.

Utilizzo dei cookie con il dominio predefinito Amplify

Quando usi Amplify per distribuire un'app web, Amplify la ospita per te sul dominio predefinito. `amplifyapp.com` Puoi visualizzare la tua app su un URL formattato come. `https://branch-name.d1m7bkiki6tdw1.amplifyapp.com`

Per aumentare la sicurezza delle tue applicazioni Amplify, il dominio amplifyapp.com è registrato nella Public Suffix List (PSL). Per una maggiore sicurezza, ti consigliamo di utilizzare i cookie con un `__Host-` prefisso se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Amplify. Questa pratica ti aiuterà a difendere il tuo dominio dai tentativi CSRF (cross-site request forgery). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Quote del servizio Amplify Hosting

Di seguito sono riportate le quote di servizio per l'hosting. AWS Amplify Le quote di servizio (precedentemente denominate limiti) sono il numero massimo di risorse o operazioni di servizio per l'utente. Account AWS

Le nuove applicazioni Account AWS prevedono quote ridotte per app e lavori simultanei. AWS aumenta automaticamente queste quote in base all'utilizzo. È possibile anche richiedere un aumento delle quote.

La console Service Quotas fornisce informazioni sulle quote per il tuo account. È possibile utilizzare la console Service Quotas per visualizzare le quote di default e richiedere aumenti delle quote per le quote regolabili. Per ulteriori informazioni, consulta Richiesta di un aumento di quota nella Guida per l'utente di Service Quotas.

Nome	Predefinita	Adattabile	Descrizione
App	Ogni regione supportata: 25	Sì	Il numero massimo di app che puoi creare in AWS Amplify Console in questo account nella regione corrente.
Diramazioni per app	Ogni Regione supportata: 50	No	Il numero massimo di diramazioni per app che è possibile creare in questo account nella regione corrente
Dimensioni artefatto di build	Ogni regione supportata: 5 GB	No	La dimensione massima (in GB) di un artefatto di build di un'app. Un artefatto di build viene distribuito da Amplify AWS Console dopo una build.

Nome	Predefinita	Adattabile	Descrizione
Dimensioni artefatto di cache	Ogni regione supportata: 5 GB	No	La dimensione massima (in GB) di un artefatto della cache.
Processi simultanei	Ogni regione supportata: 5	Sì	Il numero massimo di processi simultanei che puoi creare in questo account nella regione corrente.
Domini per app	Ogni regione supportata: 5	Sì	Il numero massimo di domini per app che è possibile creare in questo account nella regione corrente
Dimensioni artefatto di cache dell'ambiente	Ogni regione supportata: 5 GB	No	La dimensione massima (in GB) dell'artefatto della cache dell'ambiente.
Dimensioni file ZIP di distribuzione manuale	Ogni regione supportata: 5 GB	No	La dimensione massima (in GB) di un file ZIP di distribuzione manuale.
Numero massimo di creazioni di app all'ora	Ogni regione supportata: 25	No	Il numero massimo di app che puoi creare in AWS Amplify Console all'ora in questo account nella regione corrente.

Nome	Predefinita	Adattare	Descrizione
Richiedi token al secondo	Ogni regione supportata: 20.000	Sì	Il numero massimo di token di richiesta al secondo per un'app. Amplify Hosting alloca i token alle richieste in base alla quantità di risorse (tempo di elaborazione e trasferimento dati) che consumano.
Sottodomini per dominio	Ogni Regione supportata: 50	No	Il numero massimo di sottodomini per dominio che è possibile creare in questo account nella regione corrente
Webhook per app	Ogni Regione supportata: 50	Sì	Il numero massimo di webhook per app che è possibile creare in questo account nella regione corrente

Per ulteriori informazioni sulle quote del servizio Amplify, AWS Amplify vedere endpoint e quote nel. Riferimenti generali di AWS

Risoluzione dei problemi di Amplify

Se si verificano errori o problemi di distribuzione durante l'utilizzo di Amplify Hosting, consulta gli argomenti in questa sezione.

Argomenti

- Risoluzione dei problemi generali
- Risoluzione dei problemi relativi all'immagine di build di Amazon Linux 2023
- Risoluzione dei problemi di build
- Risoluzione dei problemi relativi ai domini personalizzati
- Risoluzione dei problemi relativi alle applicazioni renderizzate lato server
- Risoluzione dei problemi relativi a reindirizzamenti e riscritture
- Risoluzione dei problemi di caching
- Configurazione dell'accesso Amplify ai repository GitHub

Risoluzione dei problemi generali

Le informazioni seguenti possono essere utili per risolvere problemi generali con Amplify.

Argomenti

- Codice di stato HTTP 429 (troppe richieste)
- La console Amplify non mostra lo stato della build e l'ora dell'ultimo aggiornamento della mia app
- Le anteprime Web non vengono create per le nuove richieste pull
- La mia distribuzione manuale è bloccata con uno stato in sospeso nella console Amplify
- Devo aggiornare la versione Node.js della mia applicazione

Codice di stato HTTP 429 (troppe richieste)

Amplify controlla il numero di richieste al secondo (RPS) al tuo sito Web in base al tempo di elaborazione e al trasferimento dei dati consumati dalle richieste in arrivo. Se l'applicazione restituisce un codice di stato HTTP 429, le richieste in arrivo superano il tempo di elaborazione e il trasferimento dei dati assegnati all'applicazione. Questo limite di applicazioni è gestito dalla quota di

servizio di Amplify. `REQUEST_TOKENS_PER_SECOND` Per ulteriori informazioni sulle quote, consulta Quote del servizio Amplify Hosting.

Per risolvere questo problema, ti consigliamo di ottimizzare l'applicazione per ridurre la durata delle richieste e il trasferimento dei dati per aumentare l'RPS dell'app. Ad esempio, con gli stessi 20.000 token, una pagina SSR altamente ottimizzata che risponde entro 100 millisecondi può supportare un RPS più elevato rispetto a una pagina con una latenza superiore a 200 millisecondi.

Analogamente, un'applicazione che restituisce una dimensione di risposta di 1 MB consumerà più token rispetto a un'applicazione che restituisce una dimensione di risposta di 250 KB.

Ti consigliamo inoltre di sfruttare la CloudFront cache di Amazon configurando Cache-Control intestazioni che massimizzino il tempo di conservazione di una determinata risposta nella cache. Le richieste inviate dalla CloudFront cache non vengono conteggiate ai fini del limite di velocità. Ogni CloudFront distribuzione può gestire fino a 250.000 richieste al secondo, consentendoti di scalare molto l'app utilizzando la cache. Per ulteriori informazioni sulla CloudFront cache, consulta Optimizing caching and availability nella Amazon CloudFront Developer Guide.

La console Amplify non mostra lo stato della build e l'ora dell'ultimo aggiornamento della mia app

Quando accedi alla pagina Tutte le app nella console Amplify, viene visualizzato un riquadro per ciascuna delle app nella regione corrente. Se non vedi lo stato della build, ad esempio Distribuito, e l'ora dell'ultimo aggiornamento visualizzati per un'app, all'app non è associato uno `Production` stage branch.

Per elencare le app nella console, Amplify utilizza l'API. `ListApps` Amplify utilizza `ProductionBranch.status` l'attributo per visualizzare lo stato della build e l'attributo per visualizzare `ProductionBranch.lastDeployTime` l'ora dell'ultimo aggiornamento. Per ulteriori informazioni su questa API, consulta la ProductionBranchdocumentazione dell'API Amplify Hosting.

Usa le seguenti istruzioni per associare uno `Production` stage al ramo della tua app.

1. Accedi alla console Amplify.
2. Nella pagina Tutte le app, scegli l'app che desideri aggiornare.
3. Nel pannello di navigazione scegli Impostazioni app, quindi Impostazioni Branch.
4. Nella sezione Impostazioni Branch, scegli Modifica.

5. Per il ramo di produzione, scegli il nome del ramo che desideri utilizzare.
6. Scegli Save (Salva).
7. Torna alla pagina Tutte le app. Ora dovrebbero essere visualizzati lo stato della build e l'ora dell'ultimo aggiornamento dell'app.

Le anteprime Web non vengono create per le nuove richieste pull

La funzionalità di anteprima Web consente di visualizzare in anteprima le modifiche apportate alle richieste pull prima di unirle in un ramo di integrazione. Un'anteprima web distribuisce ogni richiesta pull inviata al tuo repository su un URL di anteprima univoco diverso dall'URL utilizzato dal tuo sito principale.

Se hai attivato le anteprime web per la tua app, ma non vengono create per essere utilizzate come nuove PRs, verifica se una delle seguenti cause è la causa del problema.

1. Verifica se la tua app ha raggiunto la quota massima `Branches per app` di servizio. Per ulteriori informazioni sulle quote, consulta Quote del servizio Amplify Hosting.

   Per rimanere entro la quota predefinita di 50 filiali per app, valuta la possibilità di abilitare l'eliminazione automatica delle filiali nella tua app. Questo ti impedirà di accumulare filiali nel tuo account che non esistono più nel tuo repository.
2. Se utilizzi un GitHub archivio pubblico e alla tua app Amplify è associato un ruolo di servizio IAM, Amplify non crea anteprime per motivi di sicurezza. Ad esempio, le app con backend e le app distribuite sulla piattaforma di hosting richiedono un ruolo di servizio IAM. `WEB_COMPUTE` Pertanto, non puoi abilitare le anteprime web per questi tipi di app se il loro archivio è pubblico.

   Per consentire il funzionamento delle anteprime Web per la tua app, puoi annullare l'associazione del ruolo di servizio (se l'app non ha un backend o non è un'`WEB_COMPUTE`app) oppure puoi rendere privato il repository. GitHub

La mia distribuzione manuale è bloccata con uno stato in sospeso nella console Amplify

Le distribuzioni manuali ti consentono di pubblicare la tua app web con Amplify Hosting senza connettere un provider Git. È possibile utilizzare una delle seguenti quattro opzioni di distribuzione.

1. Trascina e rilascia la cartella dell'applicazione nella console Amplify.

2. Trascina e rilascia un file.zip (che contiene gli artefatti di compilazione del tuo sito) nella console Amplify.
3. Carica un file.zip (che contiene gli elementi di compilazione del tuo sito) in un bucket Amazon S3 e collega il bucket a un'app nella console Amplify.
4. Usa un URL pubblico che punti a un file.zip (che contiene gli artefatti di build del tuo sito) nella console Amplify.

Siamo consapevoli dei problemi relativi alla funzionalità drag a drop quando si utilizza una cartella di applicazioni per una distribuzione manuale nella console Amplify. Queste implementazioni possono non riuscire per i seguenti motivi.

- Si verificano problemi transitori di rete.
- Durante il caricamento viene apportata una modifica locale ai file.
- La sessione del browser tenta di caricare contemporaneamente una grande quantità di risorse statiche.

Sebbene lavoriamo per migliorare l'affidabilità dei nostri caricamenti drag and drop, ti consigliamo di utilizzare un file.zip invece di trascinare le cartelle dell'applicazione.

Consigliamo vivamente di caricare un file.zip in un bucket Amazon S3, in quanto ciò evita il caricamento di file dalla console Amplify e offre una maggiore affidabilità per le distribuzioni manuali. L'integrazione di Amplify Per ulteriori informazioni, consulta Distribuzione di un sito Web statico su Amplify da un bucket Amazon S3.

Devo aggiornare la versione Node.js della mia applicazione

Il supporto di Amplify per le app che utilizzano le versioni 16 e 18 di Node.js termina il 15 settembre 2025. Le applicazioni già distribuite continueranno a funzionare. Tuttavia, dopo questa data, non sarà possibile distribuire gli aggiornamenti dell'applicazione finché non si esegue l'aggiornamento alla versione 20 o successiva di Node.js.

Se utilizzi l'immagine di build di Amazon Linux 2023, la versione 20 di Node.js è supportata per impostazione predefinita. A partire dal 15 settembre 2025, l'immagine AL2 023 supporterà automaticamente Node.js 22 e cambierà la versione Node.js predefinita da 18 a 22.

Amazon Linux 2 (AL2) non supporta automaticamente Node.js versione 20 o successiva. Se stai attualmente utilizzando AL2, ti consigliamo di passare a AL2 023. Puoi modificare l'immagine di

build nella console Amplify. Puoi anche utilizzare un'immagine di build personalizzata che supporti la versione di Node.js specificata.

Prima di eseguire l'aggiornamento, consigliamo di testare l'applicazione in una nuova filiale per verificare che funzioni correttamente.

Opzioni di aggiornamento

Console Amplify

È possibile utilizzare la funzionalità di aggiornamento live dei pacchetti nella console Amplify per specificare la versione di Node.js da utilizzare. Per istruzioni, consultare Utilizzo di versioni specifiche di pacchetti e dipendenze nell'immagine di build.

Immagine di build personalizzata

Se stai usando un'immagine di build personalizzata e NVM è installato sull'immagine, puoi aggiungerla `nvm install 20` al tuo Dockerfile. Per ulteriori informazioni sui requisiti e le istruzioni di configurazione per un'immagine di build personalizzata, consulta. Personalizzazione dell'immagine di costruzione

Creazione di impostazioni

È possibile specificare la versione di Node.js da utilizzare nelle impostazioni di `amplify.yml` compilazione dell'applicazione aggiungendo il `nvm use` comando alla sezione dei comandi PreBuild. Per istruzioni sull'aggiornamento delle impostazioni di build di un'applicazione, consultaConfigurazione delle impostazioni di build per un'applicazione Amplify.

L'esempio seguente dimostra come personalizzare le impostazioni di build per impostare la versione predefinita di Node.js su Node.js 18 e aggiornarla alla versione Node.js 20 su un ramo di test denominato`node-20`.

```
frontend:
  phases:
    preBuild:
      commands:
        - nvm use 18
        - if [ "${AWS_BRANCH}" = "node-20" ]; then nvm use 20; fi
```

> **Warning**
>
> Tieni presente che `preBuild` i comandi vengono eseguiti dopo gli aggiornamenti live dei pacchetti. La versione di Node.js specificata dal `nvm use` comando sovrascriverà la versione Node.js impostata dagli aggiornamenti dei pacchetti live.

Risoluzione dei problemi relativi all'immagine di build di Amazon Linux 2023

Le informazioni seguenti possono essere utili per risolvere i problemi con l'immagine di build di Amazon Linux 2023 (AL2023).

Argomenti

- Voglio eseguire le funzioni Amplify con il runtime Python
- Voglio eseguire comandi che richiedono i privilegi di superutente o root

Voglio eseguire le funzioni Amplify con il runtime Python

Amplify Hosting ora utilizza l'immagine di build di Amazon Linux 2023 per impostazione predefinita quando distribuisci una nuova applicazione. AL2023 è preinstallato con le versioni Python 3.8, 3.9, 3.10 e 3.11.

Per la retrocompatibilità con l'immagine di Amazon Linux 2, l'immagine di build AL2 023 ha collegamenti simbolici per le versioni precedenti di Python preinstallati.

Per impostazione predefinita, la versione 3.10 di Python viene utilizzata a livello globale. Per creare le tue funzioni utilizzando una versione specifica di Python, esegui i seguenti comandi nel file delle specifiche di build dell'applicazione.

```
version: 1
backend:
  phases:
    build:
      commands:
        # use a python version globally
        - pyenv global 3.11
        # verify python version
```

```
        - python --version
        # install pipenv
        - pip install --user pipenv
        # add to path
        - export PATH=$PATH:/root/.local/bin
        # verify pipenv version
        - pipenv --version
        - amplifyPush --simple
```

Voglio eseguire comandi che richiedono i privilegi di superutente o root

Se utilizzi l'immagine di build di Amazon Linux 2023 e ricevi un errore durante l'esecuzione di comandi di sistema che richiedono privilegi di superutente o root, devi eseguire questi comandi utilizzando il comando Linux`sudo`. Ad esempio, se ricevi un errore durante l'esecuzione`yum install -y gcc`, usa. `sudo yum install -y gcc`

L'immagine di build di Amazon Linux 2 utilizzava l'utente root, ma l'immagine AL2 023 di Amplify esegue il codice con un utente personalizzato. `amplify` Amplify concede a questo utente i privilegi per eseguire comandi utilizzando il comando Linux. `sudo` È consigliabile utilizzarlo per i comandi che richiedono i privilegi `sudo` di superutente.

Risoluzione dei problemi di build

Se si verificano problemi durante la creazione o la creazione di un'applicazione Amplify, consulta gli argomenti in questa sezione per ricevere assistenza.

Argomenti

- I nuovi commit sul mio repository non attivano le build di Amplify
- Il nome del mio repository non è elencato nella console Amplify quando creo una nuova applicazione
- La mia build fallisce con l'Cannot find module aws-exportserrore (solo app di prima generazione)
- Voglio sovrascrivere un timeout di compilazione

I nuovi commit sul mio repository non attivano le build di Amplify

Se i nuovi commit sul tuo repository Git non attivano le build Amplify, verifica che il webhook sia ancora presente nel tuo repository. Se è presente, controlla la cronologia delle richieste di webhook

per vedere se ci sono errori. Amplify ha un limite di dimensione del payload di 256 KB per i webhook in entrata. Se invii un commit al tuo repository che contiene un gran numero di file modificati, potresti superare questo limite e impedire l'attivazione delle build.

Il nome del mio repository non è elencato nella console Amplify quando creo una nuova applicazione

Quando crei una nuova applicazione nella console Amplify, puoi scegliere tra gli archivi disponibili della tua organizzazione nella pagina Aggiungi repository e branch. Il tuo repository di destinazione potrebbe non essere visualizzato nell'elenco se non è stato aggiornato di recente. Ciò potrebbe verificarsi se l'organizzazione dispone di un numero elevato di repository. Per risolvere questo problema, invia un commit al repository, quindi aggiorna l'elenco dei repository nella console. Ciò dovrebbe far sì che il repository venga visualizzato.

La mia build fallisce con l'**`Cannot find module aws-exports`**errore (solo app di prima generazione)

Se l'app non riesce a trovare il `aws-exports.js` file durante una compilazione, viene restituito il seguente errore.

```
TS2307: Cannot find module 'aws-exports'
```

L'interfaccia a riga di comando (CLI) di Amplify genera il file durante la compilazione `aws-exports.js` del backend. Per risolvere questo errore, è necessario creare un `aws-exports.js` file da utilizzare nella build. Aggiungi il seguente codice alle specifiche di build per creare il file:

```
backend:
  phases:
    build:
      commands:
        - "# Execute Amplify CLI with the helper script"
        - amplifyPush --simple
```

Per un esempio completo delle impostazioni delle specifiche di build per un'app Amplify, consulta. Riferimento alla sintassi YAML delle specifiche di build

Voglio sovrascrivere un timeout di compilazione

Il timeout di compilazione predefinito è di 30 minuti. Puoi sovrascrivere il timeout di compilazione predefinito utilizzando la `_BUILD_TIMEOUT` variabile di ambiente. Il timeout minimo di build è di 5 minuti. Il timeout massimo di build è di 120 minuti.

Per istruzioni sull'impostazione di una variabile di ambiente per un'app nella console Amplify, consulta. Impostazione delle variabili di ambiente

Risoluzione dei problemi relativi ai domini personalizzati

Se si verificano problemi durante il collegamento di un dominio personalizzato all'applicazione Amplify, consulta gli argomenti in questa sezione per ricevere assistenza.

Se non trovi una soluzione al tuo problema qui, contatta. Supporto Per ulteriori informazioni, consulta Creazione di una richiesta di assistenza nella Guida per l'utente di Supporto AWS .

Argomenti

- Devo verificare che il mio CNAME si risolva
- Il mio dominio ospitato presso una terza parte è bloccato nello stato In attesa di verifica
- Il mio dominio ospitato con Amazon Route 53 è bloccato nello stato di verifica in sospeso
- La mia app con sottodomini a più livelli è bloccata nello stato In attesa di verifica
- Il mio provider DNS non supporta i record A con nomi di dominio completi
- Ricevo un errore CNAMEAlready ExistsException
- Ricevo un errore di verifica aggiuntiva richiesta
- Ricevo un errore 404 sull'URL CloudFront
- Ricevo errori nel certificato SSL o nel protocollo HTTPS quando visito il mio dominio

Devo verificare che il mio CNAME si risolva

1. Dopo aver aggiornato i tuoi record DNS con il tuo provider di dominio di terze parti, puoi utilizzare uno strumento come dig o un sito Web gratuito come https://www.whatsmydns.net/ per verificare che il record CNAME si stia risolvendo correttamente. La schermata seguente mostra come usare whatsmydns.net per controllare il record CNAME per il dominio www.example.com.

2. Scegli Cerca e whatsmydns.net mostrerà i risultati del tuo CNAME. La schermata seguente è un esempio di un elenco di risultati che verificano che il CNAME si risolva correttamente in un URL di cloudfront.net.

Il mio dominio ospitato presso una terza parte è bloccato nello stato In attesa di verifica

1. Se il tuo dominio personalizzato è bloccato nello stato In attesa di verifica, verifica che i tuoi CNAME record siano in corso di risoluzione. Per istruzioni su come eseguire questa operazione, consulta l'argomento precedente sulla risoluzione dei problemi, How do I verify that my CNAME resolves.
2. Se i CNAME record non vengono risolti, verifica che la CNAME voce esista nelle impostazioni DNS con il tuo provider di dominio.

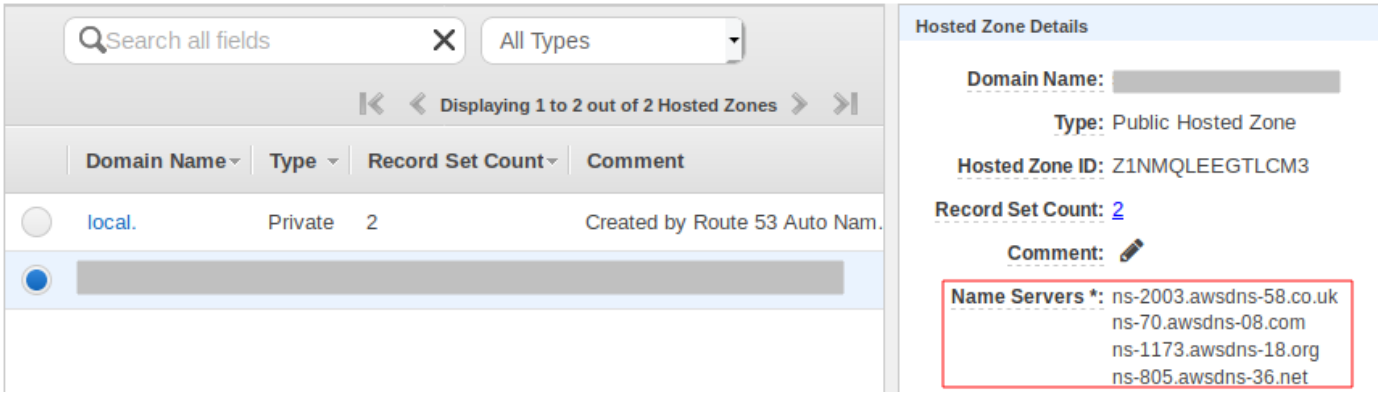
> **Important**
>
> È importante aggiornare i CNAME record non appena si crea il dominio personalizzato. Dopo aver creato l'app nella console Amplify, CNAME il record viene controllato ogni pochi minuti per determinare se si risolve. Se non si risolve dopo un'ora, il controllo viene effettuato ogni poche ore, il che può comportare un ritardo nella preparazione del dominio per l'uso. Se hai aggiunto o aggiornato i tuoi CNAME record poche ore dopo aver creato l'app, questa è la causa più probabile che l'app rimanga bloccata nello stato di verifica in sospeso.

3. Se hai verificato l'esistenza del CNAME record, potrebbe esserci un problema con il tuo provider DNS. Puoi contattare il provider DNS per diagnosticare il motivo per cui la verifica DNS non CNAME si risolve oppure puoi migrare il tuo DNS su Route 53. Per ulteriori informazioni, consulta Rendere Amazon Route 53 il servizio DNS per un dominio esistente.

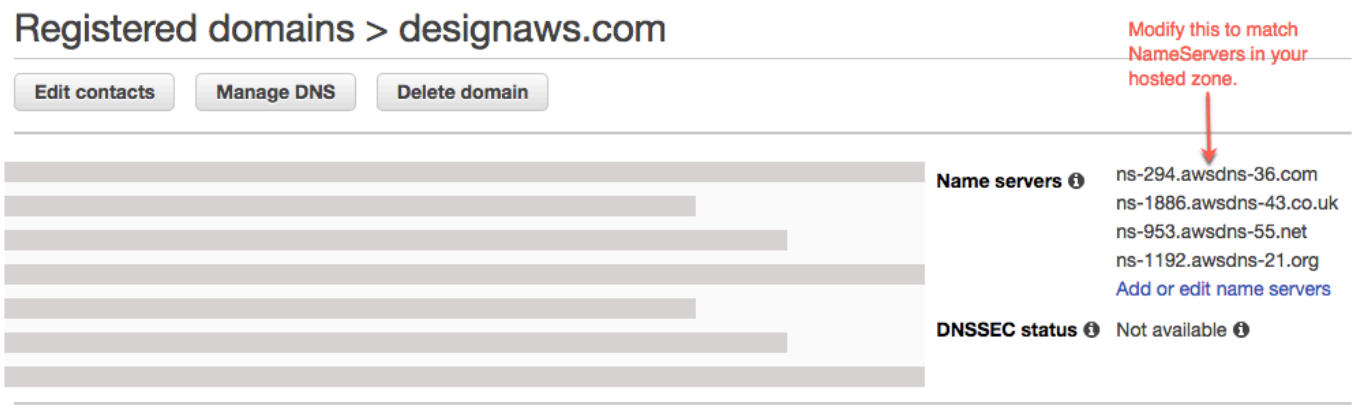
Il mio dominio ospitato con Amazon Route 53 è bloccato nello stato di verifica in sospeso

Se hai trasferito il tuo dominio su Amazon Route 53, è possibile che il dominio abbia server di nomi diversi da quelli emessi da Amplify al momento della creazione dell'app. Per diagnosticare la causa dell'errore, attenersi alla seguente procedura.

1. Accedi alla console Amazon Route 53
2. Nel riquadro di navigazione, scegli Hosted Zones (Zone ospitate) e scegli il nome del dominio a cui stai collegando.
3. Registra i valori del name server dalla sezione Hosted Zone Details. Questi valori sono necessari per completare il passaggio successivo. La seguente schermata della console Route 53 mostra la posizione dei valori del name server nell'angolo inferiore destro.



4. Nel riquadro di navigazione seleziona Registered domains (Domini registrati). Verifica che i name server visualizzati nella sezione Domini registrati corrispondano ai valori dei name server registrati nel passaggio precedente nella sezione Dettagli della zona ospitata. Se non corrispondono, modifica i valori del name server in modo che corrispondano ai valori della tua Hosted Zone. La seguente schermata della console Route 53 mostra la posizione dei valori del name server sul lato destro.



5. Se il problema persiste, contatta Supporto. Per ulteriori informazioni, consulta Creazione di una richiesta di assistenza nella Guida per l'utente di Supporto AWS .

La mia app con sottodomini a più livelli è bloccata nello stato In attesa di verifica

Se un'app con sottodomini a più livelli è bloccata nello stato In attesa di verifica quando si connette a un provider DNS di terze parti, potrebbe esserci un problema con il formato dei tuoi record DNS. Alcuni provider DNS aggiungono automaticamente i suffissi del dominio di secondo livello (SLD) e del dominio di primo livello (TLD) ai tuoi record. Se specifichi anche il dominio nel formato che include SLD e TLD, ciò può causare un problema di verifica del dominio.

Quando colleghi un dominio, prova innanzitutto a specificare il nome di dominio utilizzando il formato completo fornito da Amplify, ad esempio. `_hash.docs.backend.example.com` Se la configurazione SSL rimane bloccata nello stato di verifica in sospeso, prova a rimuovere il TLD e l'SLD dai record. Ad esempio, se il formato completo è, specifica. `_hash.docs.backend.example.com` `_hash.docs.backend` Attendere da 15 a 30 minuti per consentire la propagazione dei record. Quindi utilizza uno strumento come MX Toolbox per verificare se il processo di verifica funziona.

Il mio provider DNS non supporta i record A con nomi di dominio completi

Alcuni provider DNS non supportano i record A con nome di dominio completo (FQDN) come `example.cloudfront.net` Ad esempio, Cloudflare A records può solo scrivere IPv4 indirizzi e non supporta. FQDNs Per ovviare a questa limitazione, consigliamo di utilizzare CNAME i record anziché A records nella configurazione. DNS

Ad esempio, la DNS configurazione seguente utilizza unA record.

```
A     | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Modificalo nella seguente DNS configurazione per utilizzare solo CNAME i record.

```
CNAME | @ | ***.cloudfront.net
CNAME | www | ***.cloudfront.net
```

Questa soluzione alternativa ti consente di indirizzare correttamente il tuo dominio apex (@ record) a servizi come CloudFront, evitando al contempo la limitazione IPv4 -only del sistema di A records Cloudflare.

Ricevo un errore CNAMEAlready ExistsException

Se ricevi un CNAMEAlreadyExistsExceptionerrore, significa che uno dei nomi host che hai provato a connettere (un sottodominio o il dominio apex) è già distribuito su un'altra distribuzione Amazon. CloudFront L'origine dell'errore dipende dai tuoi attuali provider di hosting e DNS.

Un CNAME alias, ad esempio `example.com` o, `sub.example.com` può essere associato solo a una singola CloudFront distribuzione alla volta. CNAMEAlreadyExistsExceptionIndica che il dominio è già associato a un'altra CloudFront distribuzione, all'interno della stessa Account AWS o potenzialmente in un account diverso. Il dominio deve essere dissociato dalla CloudFront distribuzione precedente prima che la nuova distribuzione creata da Amplify Hosting funzioni. Potrebbe essere necessario controllare più di un account se tu o la tua organizzazione possedete più di un account. Account AWS

Effettuare le seguenti operazioni per diagnosticare la causa dell'CNAMEAlreadyExistsExceptionerrore.

1. Accedi alla CloudFront console Amazon e verifica di non avere questo dominio distribuito su un'altra distribuzione. È possibile collegare un singolo CNAME record a una CloudFront distribuzione alla volta.
2. Se in precedenza hai distribuito il dominio su una CloudFront distribuzione, devi rimuoverlo.
   a. Dal menu di navigazione a sinistra, scegli
   b. Seleziona il nome della distribuzione da modificare.
   c. Scegli la scheda Generale. Nella sezione Settings (Impostazioni), scegli Edit (Modifica).

   d. Rimuovi il nome di dominio dal nome di dominio alternativo (CNAME). Quindi scegli Salva le modifiche.

3. Verifica che non esista nessun'altra CloudFront distribuzione che utilizza questo dominio nella versione corrente Account AWS o in un'altra Account AWS. Se non interrompe i servizi attualmente in esecuzione, prova a eliminare e ricreare la zona ospitata.
4. Verifica se questo dominio è collegato a un'altra app Amplify di tua proprietà. In questo caso, accertati che non stai tentando di riutilizzare uno dei nomi host. Se lo utilizzi `www.example.com` per un'altra app, non puoi utilizzarlo `www.example.com` con l'app a cui ti stai attualmente connettendo. Puoi usare altri sottodomini, ad esempio. `blog.example.com`
5. Se questo dominio è stato collegato correttamente a un'altra app e poi eliminato nell'ultima ora, riprova dopo almeno un'ora. Se dopo 6 ore vedi ancora questa eccezione, contatta Supporto. Per ulteriori informazioni, consulta Creazione di una richiesta di assistenza nella Guida per l'utente di Supporto AWS .
6. Se gestisci il tuo dominio tramite Route 53, assicurati di ripulire tutte le zone CNAME o ALIAS i record ospitati che rimandano alla vecchia CloudFront distribuzione.
7. Dopo aver completato i passaggi precedenti, rimuovi il dominio personalizzato da Amplify Hosting e ricomincia con il flusso di lavoro per connettere un dominio personalizzato nella console Amplify.

Ricevo un errore di verifica aggiuntiva richiesta

Se ricevi un errore di verifica aggiuntiva richiesta, significa che AWS Certificate Manager (ACM) richiede informazioni aggiuntive per elaborare questa richiesta di certificato. Ciò può accadere come misura di protezione contro le frodi, ad esempio quando il dominio si colloca all'interno dei migliori 1000 siti web di Alexa. Per fornire queste informazioni, usa il Centro di supporto per contattare Supporto. Se non disponi di un piano di supporto, pubblica un nuovo thread nel forum di discussione di ACM.

> **Note**
>
> Non puoi richiedere un certificato per i nomi di dominio di proprietà di Amazon, ad esempio quelli che finiscono con amazonaws.com, cloudfront.net o elasticbeanstalk.com.

Ricevo un errore 404 sull'URL CloudFront

Per servire il traffico, Amplify Hosting punta a CloudFront un URL tramite un record CNAME. Durante il processo di connessione di un'app a un dominio personalizzato, la console Amplify visualizza l'URL CloudFront dell'app. Tuttavia, non è possibile accedere direttamente all'applicazione utilizzando questo CloudFront URL. Restituisce un errore 404. L'applicazione si risolve solo utilizzando l'URL dell'app Amplify (ad esempio) o il dominio personalizzato (ad esempio). `https://main.d5udybEXAMPLE.amplifyapp.com` `www.example.com`

Amplify deve indirizzare le richieste al ramo distribuito corretto e utilizza l'hostname per farlo. Ad esempio, puoi configurare il dominio `www.example.com` che punta al ramo principale di un'app, ma anche configurare `dev.example.com` che punti al ramo di sviluppo della stessa app. Pertanto, è necessario visitare l'applicazione in base ai sottodomini configurati in modo che Amplify possa indirizzare le richieste di conseguenza.

Ricevo errori nel certificato SSL o nel protocollo HTTPS quando visito il mio dominio

Se disponi di record DNS di Certificate Authority Authorization (CAA) configurati con il tuo provider DNS di terze parti, AWS Certificate Manager (ACM) potrebbe non essere in grado di aggiornare o riemettere i certificati intermedi per il certificato SSL del tuo dominio personalizzato. Per risolvere questo problema, devi aggiungere un record CAA per considerare attendibile almeno uno dei domini dell'autorità di certificazione di Amazon. La procedura seguente descrive i passaggi da eseguire.

Per aggiungere un record CAA per considerare attendibile un'autorità di certificazione Amazon

1. Configura un record CAA con il tuo provider di dominio per considerare attendibile almeno uno dei domini di autorità di certificazione di Amazon. Per ulteriori informazioni sulla configurazione del record CAA, consulta Problemi di autorizzazione dell'autorità di certificazione (CAA) nella Guida per l'utente.AWS Certificate Manager
2. Utilizza uno dei seguenti metodi per aggiornare il tuo certificato SSL:
   - Aggiorna manualmente utilizzando la console Amplify.

> **Note**
>
> Questo metodo causerà tempi di inattività per il tuo dominio personalizzato.

a. Accedi AWS Management Console e apri la console Amplify.

b. Scegli l'app a cui desideri aggiungere un record CAA.

c. Nel riquadro di navigazione, scegli Impostazioni app, Gestione del dominio.

d. Nella pagina di gestione del dominio, elimina il dominio personalizzato.

e. Connetti nuovamente la tua app al dominio personalizzato. Questo processo emette un nuovo certificato SSL e i relativi certificati intermedi possono ora essere gestiti da ACM.

   Per ricollegare l'app al dominio personalizzato, utilizza una delle seguenti procedure che corrisponde al provider di dominio che stai utilizzando.

   - Aggiunta di un dominio personalizzato gestito da Amazon Route 53.
   - Aggiunta di un dominio personalizzato gestito da un provider DNS di terze parti.
   - Aggiornamento dei record DNS per un dominio gestito da GoDaddy.

- Contattaci Supporto per richiedere la riemissione del certificato SSL.

Risoluzione dei problemi relativi alle applicazioni renderizzate lato server

Se riscontri problemi imprevisti durante la distribuzione di un'app SSR con Amplify Hosting compute, consulta i seguenti argomenti per la risoluzione dei problemi. Se non trovi una soluzione al tuo problema qui, consulta la guida alla risoluzione dei problemi di calcolo web SSR nell'archivio Amplify Hosting Issues. GitHub

Argomenti

- Ho bisogno di aiuto per usare un adattatore di framework
- I percorsi dell'API Edge causano il fallimento della mia build di Next.js
- La rigenerazione statica incrementale su richiesta non funziona per la mia app
- L'output di build della mia applicazione supera la dimensione massima consentita
- La mia build fallisce con un errore di memoria esaurita
- La dimensione della risposta HTTP della mia applicazione è troppo grande
- Come posso misurare l'ora di avvio della mia app di elaborazione a livello locale?

Ho bisogno di aiuto per usare un adattatore di framework

Se riscontri problemi durante la distribuzione di un'app SSR che utilizza un adattatore di framework, consulta. Utilizzo di adattatori open source per qualsiasi framework SSR

I percorsi dell'API Edge causano il fallimento della mia build di Next.js

Attualmente, Amplify non supporta Next.js Edge API Routes. È necessario utilizzare sistemi non edge APIs e middleware per ospitare l'app con Amplify.

La rigenerazione statica incrementale su richiesta non funziona per la mia app

A partire dalla versione 12.2.0, Next.js supporta la rigenerazione statica incrementale (ISR) per eliminare manualmente la cache Next.js per una pagina specifica. Tuttavia, Amplify attualmente non supporta l'ISR su richiesta. Se la tua app utilizza la riconvalida su richiesta di Next.js, questa funzionalità non funzionerà quando distribuisci l'app su Amplify.

L'output di build della mia applicazione supera la dimensione massima consentita

Attualmente, la dimensione massima di output di build supportata da Amplify per le app SSR è di 220 MB. Se ricevi un messaggio di errore che indica che la dimensione dell'output di build della tua app supera la dimensione massima consentita, devi prendere provvedimenti per ridurla.

Per ridurre le dimensioni dell'output di compilazione di un'app, puoi ispezionare gli artefatti di build dell'app e identificare eventuali dipendenze di grandi dimensioni da aggiornare o rimuovere. Per prima cosa, scarica gli elementi di build sul tuo computer locale. Quindi, controlla la dimensione delle directory. Ad esempio, la `node_modules` directory potrebbe contenere file binari come `@swc` e a `@esbuild` cui fanno riferimento i file di runtime del server Next.js. Poiché questi file binari non sono necessari in fase di esecuzione, è possibile eliminarli dopo la compilazione.

Utilizza le seguenti istruzioni per scaricare l'output della build di un'app e controllare le dimensioni delle directory utilizzando (AWS Command Line Interface CLI).

Per scaricare e controllare l'output della build di un'app Next.js

1. Apri una finestra del terminale ed esegui il comando seguente. Modifica l'ID dell'app, il nome del ramo e l'ID del lavoro con le tue informazioni. Per l'ID del lavoro, usa il numero di build della build fallita su cui stai indagando.

```
aws amplify get-job --app-id abcd1234 --branch-name main --job-id 2
```

2. Nell'output del terminale, individua l'URL degli artefatti predefiniti nella sezione,,. `job steps stepName: "BUILD"` L'URL è evidenziato in rosso nell'output di esempio seguente.

```
"job": {
    "summary": {
        "jobArn": "arn:aws:amplify:us-west-2:111122223333:apps/abcd1234/main/
jobs/0000000002",
        "jobId": "2",
        "commitId": "HEAD",
        "commitTime": "2024-02-08T21:54:42.398000+00:00",
        "startTime": "2024-02-08T21:54:42.674000+00:00",
        "status": "SUCCEED",
        "endTime": "2024-02-08T22:03:58.071000+00:00"
    },
    "steps": [
        {
            "stepName": "BUILD",
            "startTime": "2024-02-08T21:54:42.693000+00:00",
            "status": "SUCCEED",
            "endTime": "2024-02-08T22:03:30.897000+00:00",
            "logUrl": "https://aws-amplify-prod-us-west-2-artifacts.s3.us-
west-2.amazonaws.com/abcd1234/main/0000000002/BUILD/log.txt?X-Amz-Security-
Token=IQoJb3JpZ2luX2V...Example
```

3. Copia e incolla l'URL in una finestra del browser. Un `artifacts.zip` file viene scaricato sul tuo computer locale. Questo è il risultato della tua build.
4. Esegui il comando du disk usage per controllare la dimensione delle directory. Il comando di esempio seguente restituisce la dimensione delle directory `compute` and`static`.

```
du -csh compute static
```

Di seguito è riportato un esempio di output con informazioni sulle dimensioni per le `static` directory `compute` and.

```
 29M    compute
3.8M    static
 33M    total
```

5. Aprire la `compute` directory e individuarla. `node_modules` Controlla le dipendenze dei file che puoi aggiornare o rimuovere per ridurre le dimensioni della cartella.
6. Se la tua app include file binari che non sono necessari in fase di esecuzione, eliminali dopo la compilazione aggiungendo i seguenti comandi alla sezione build del file dell'`amplify.yml`app.

```
- rm -f node_modules/@swc/core-linux-x64-gnu/swc.linux-x64-gnu.node
- rm -f node_modules/@swc/core-linux-x64-musl/swc.linux-x64-musl.node
```

Di seguito è riportato un esempio della sezione dei comandi di compilazione di un `amplify.yml` file con questi comandi aggiunti dopo l'esecuzione di una build di produzione.

```
frontend:
  phases:
    build:
      commands:
         -npm run build

         // After running a production build, delete the files
         - rm -f node_modules/@swc/core-linux-x64-gnu/swc.linux-x64-gnu.node
         - rm -f node_modules/@swc/core-linux-x64-musl/swc.linux-x64-musl.node
```

La mia build fallisce con un errore di memoria esaurita

Next.js consente di memorizzare nella cache gli elementi della build per migliorare le prestazioni nelle build successive. Inoltre, il AWS CodeBuild contenitore di Amplify comprime e carica questa cache su Amazon S3, per tuo conto, per migliorare le prestazioni di build successive. Ciò potrebbe causare il fallimento della compilazione con un errore di memoria esaurita.

Esegui le seguenti azioni per evitare che l'app superi il limite di memoria durante la fase di compilazione. Innanzitutto, rimuovi `.next/cache/**/*` dalla sezione cache.paths delle impostazioni di build. Quindi, rimuovi la variabile di `NODE_OPTIONS` ambiente dal file delle impostazioni di build. Invece, imposta la variabile di `NODE_OPTIONS` ambiente nella console Amplify per definire il limite massimo di memoria del nodo. Per ulteriori informazioni sull'impostazione delle variabili di ambiente utilizzando la console Amplify Impostazione delle variabili di ambiente

Dopo aver apportato queste modifiche, prova di nuovo con la build. Se riesce, aggiungilo `.next/cache/**/*` nuovamente alla sezione cache.paths del file delle impostazioni di build.

Per ulteriori informazioni sulla configurazione della cache di Next.js per migliorare le prestazioni di compilazione, consulta AWS CodeBuild sul sito Web Next.js.

La dimensione della risposta HTTP della mia applicazione è troppo grande

Attualmente, la dimensione massima di risposta supportata da Amplify per le app Next.js 12 e successive che utilizzano la piattaforma Web Compute è di 5,72 MB. Le risposte oltre tale limite restituiscono 504 errori senza contenuto ai client.

Come posso misurare l'ora di avvio della mia app di elaborazione a livello locale?

Utilizza le seguenti istruzioni per determinare l'ora di inizializzazione/avvio locale dell'app Compute Next.js 12 o versione successiva. Puoi confrontare le prestazioni della tua app a livello locale con quelle di Amplify Hosting e utilizzare i risultati per migliorare le prestazioni della tua app.

Per misurare il tempo di inizializzazione di un'app Next.js Compute a livello locale

1. Apri il `next.config.js` file dell'app e imposta l'`output`opzione `standalone` come segue.

```
** @type {import('next').NextConfig} */
const nextConfig = {
  // Other options
  output: "standalone",
};

module.exports = nextConfig;
```

2. Apri una finestra del terminale ed esegui il comando seguente per creare l'app.

```
next build
```

3. Esegui il seguente comando per copiare la `.next/static` cartella in`.next/standalone/.next/static`.

```
cp -r .next/static .next/standalone/.next/static
```

4. Esegui il comando seguente per copiare la `public` cartella in`.next/standalone/public`.

```
cp -r public .next/standalone/public
```

5. Esegui il comando seguente per avviare il server Next.js.

```
node .next/standalone/server.js
```

6. Notate quanto tempo occorre tra l'esecuzione del comando nel passaggio 5 e l'avvio del server. Quando il server è in ascolto su una porta, dovrebbe stampare il seguente messaggio.

```
Listening on port 3000
```

7. Nota quanto tempo impiega il caricamento di tutti gli altri moduli dopo l'avvio del server nel passaggio 6. Ad esempio, il caricamento di librerie `bugsnag` richiede 10-12 secondi. Dopo il caricamento, verrà visualizzato il messaggio `[bugsnag] loaded` di conferma.
8. Aggiungi insieme le durate temporali del passaggio 6 e del passaggio 7. Questo risultato è l'ora di inizializzazione/avvio locale dell'app Compute.

Risoluzione dei problemi relativi a reindirizzamenti e riscritture

Se si verificano problemi durante l'impostazione di reindirizzamenti e riscritture per un'applicazione Amplify, consulta gli argomenti in questa sezione per ricevere assistenza.

Argomenti

- L'accesso è negato per determinati percorsi anche con la regola di reindirizzamento SPA.
- Voglio configurare un proxy inverso su un'API

L'accesso è negato per determinati percorsi anche con la regola di reindirizzamento SPA.

Se ricevi un errore di accesso negato per determinati percorsi con una regola di reindirizzamento SPA, `baseDirectory` potrebbe non essere impostato correttamente nelle impostazioni di configurazione dell'app. Ad esempio, se il frontend dell'app è integrato `build` nella directory, anche le impostazioni di build devono puntare alla `build` directory. Il seguente esempio di specifica di build dimostra questa impostazione.

```
frontend:
```

```
  artifacts:
    baseDirectory: build
    files:
        - "**/*"
```

Per un esempio completo delle impostazioni delle specifiche di build per un'app Amplify, consulta Riferimento alla sintassi YAML delle specifiche di build

Voglio configurare un proxy inverso su un'API

È possibile utilizzare il seguente codice JSON per configurare un reverse proxy su un endpoint dinamico.

```
[
  {
    "source": "/documents/<*>",
    "target": "https://otherdomain/resource/<*>",
    "status": "200",
    "condition": null
  }
]
```

Per un esempio di base sulla creazione di un proxy inverso per la tua app Amplify su un'API di terze parti, consulta. Riscrittura inversa del proxy

Risoluzione dei problemi di caching

Se riscontri problemi di cache per un'applicazione Amplify

Argomenti

- Voglio ridurre le dimensioni della cache di un'app
- Voglio disabilitare la lettura dalla cache per un'app

Voglio ridurre le dimensioni della cache di un'app

Se utilizzi la cache, potresti memorizzare nella cache file intermedi che non vengono ripuliti tra le build. La memorizzazione nella cache di questi file usati raramente aumenterà le dimensioni della cache. Per evitare che ciò accada, puoi escludere cartelle specifiche dalla cache utilizzando la `!` direttiva nella `cache` sezione delle specifiche di build dell'app.

Il seguente esempio di impostazioni di build mostra come utilizzare la `!` direttiva per specificare una cartella che non si desidera memorizzare nella cache.

```
cache:
  paths:
    - node_modules/**/*
    - "!node_modules/path/not/to/cache"
```

Quando si inserisce nella cache la `node_modules` cartella, `node_modules/.cache` viene omessa per impostazione predefinita.

Per un esempio completo delle impostazioni delle specifiche di build per un'app Amplify, consulta Riferimento alla sintassi YAML delle specifiche di build

Voglio disabilitare la lettura dalla cache per un'app

Se desideri disabilitare la lettura dalla cache di un'app, rimuovi la sezione cache dalle specifiche di build dell'app.

Configurazione dell'accesso Amplify ai repository GitHub

Amplify ora utilizza la funzione Apps per autorizzare GitHub l'accesso in sola lettura di Amplify ai repository. GitHub Con l'app GitHub Amplify, le autorizzazioni sono più ottimizzate, consentendoti di concedere ad Amplify l'accesso solo ai repository che specifichi. Per saperne di più sulle app, consulta Informazioni sulle app sul sito web GitHub. GitHub GitHub

Quando connetti una nuova app archiviata in un GitHub repository, per impostazione predefinita Amplify utilizza GitHub l'app per accedere al repository. Tuttavia, le app Amplify esistenti che hai collegato in precedenza GitHub dai OAuth repository vengono utilizzate per l'accesso. CI/CD continuerà a funzionare per queste app, ma ti consigliamo vivamente di migrarle per utilizzare la nuova app Amplify. GitHub

Quando si distribuisce una nuova app o si migra un'app esistente utilizzando la console Amplify, si viene automaticamente indirizzati alla posizione di installazione dell'app Amplify. GitHub Per accedere manualmente alla pagina di installazione dell'app, apri un browser Web e accedi all'app per regione. Usa il formato`https://github.com/apps/aws-amplify-`*`REGION`*, sostituendolo *`REGION`* con la regione in cui distribuirai l'app Amplify. Ad esempio, per installare l'app GitHub Amplify nella regione Stati Uniti occidentali (Oregon), vai a -2. https://github.com/apps/ aws-amplify-us-west

Argomenti

- Installazione e autorizzazione dell'app GitHub Amplify per una nuova distribuzione
- Migrazione di un'OAuthapp esistente all'app Amplify GitHub
- Configurazione dell'app GitHub Amplify per le implementazioni CLI AWS CloudFormation e SDK
- Configurazione delle anteprime web con l'app Amplify GitHub

Installazione e autorizzazione dell'app GitHub Amplify per una nuova distribuzione

Quando distribuisci una nuova app su Amplify da codice esistente in GitHub un repository, usa le seguenti istruzioni per installare e autorizzare l'app. GitHub

Per installare e autorizzare l'app Amplify GitHub

1. Accedi AWS Management Console e apri la console Amplify.
2. Dalla pagina Tutte le app, scegli Nuova app, quindi Host web app.
3. Nella pagina Inizia a usare Amplify Hosting, GitHubscegli, quindi scegli Continua.
4. Se è la prima volta che connetti un GitHub repository, nel tuo browser si apre una nuova pagina su GitHub .com, che richiede l'autorizzazione per l'autorizzazione AWS Amplify all'accesso al tuo account. GitHub Seleziona Authorize (Autorizza).
5. Successivamente, devi installare l'app Amplify nel tuo GitHub account. GitHub Si apre una pagina su GitHub.com che richiede l'autorizzazione all'installazione e all'autorizzazione nel tuo account. AWS Amplify GitHub
6. Seleziona l' GitHub account in cui desideri installare l'app Amplify. GitHub
7. Esegui una di queste operazioni:
   - Per applicare l'installazione a tutti i repository, scegli Tutti gli archivi.
   - Per limitare l'installazione ai repository specifici selezionati, scegli Seleziona solo i repository. Assicurati di includere il repository per l'app di cui stai migrando nei repository selezionati.
8. Scegli Installa e autorizza.
9. Verrai reindirizzato alla pagina Aggiungi ramo del repository per la tua app nella console Amplify.
10. Nell'elenco dei repository aggiornati di recente, seleziona il nome del repository a cui connetterti.
11. Nell'elenco Branch, seleziona il nome del ramo del repository da connettere.

12. Scegli Next (Successivo).
13. Nella pagina Configura le impostazioni di build, scegli Avanti.
14. Nella pagina Revisione, scegli Salva e distribuisci.

Migrazione di un'OAuthapp esistente all'app Amplify GitHub

Le app Amplify esistenti che hai collegato in precedenza GitHub dai repository vengono utilizzate per l'accesso ai repository OAuth. Si consiglia fortemente di effettuare la migrazione di queste applicazioni per usare l'applicazione Amplify. GitHub

Utilizza le seguenti istruzioni per migrare un'app ed eliminare il OAuth webhook corrispondente dal tuo account. GitHub Tieni presente che la procedura per la migrazione varia a seconda che l'app GitHub Amplify sia già installata. Dopo aver migrato la prima app e aver installato e autorizzato l' GitHub app, devi solo aggiornare le autorizzazioni del repository per le successive migrazioni dell'app.

Per migrare un'app dall'app all'app OAuth GitHub

1. Accedi AWS Management Console e apri la console Amplify.
2. Scegli l'app che desideri migrare.
3. Nella pagina delle informazioni dell'app, individua il messaggio blu Migra alla nostra GitHub app e scegli Avvia migrazione.
4. Nella pagina Installa e autorizza GitHub l'app, scegli Configura GitHub app.
5. Si apre una nuova pagina nel browser su GitHub .com, che richiede l'autorizzazione per l'autorizzazione AWS Amplify all'accesso al tuo account. GitHub Seleziona Authorize (Autorizza).
6. Seleziona l' GitHub account in cui desideri installare l'app Amplify. GitHub
7. Esegui una di queste operazioni:
   - Per applicare l'installazione a tutti i repository, scegli Tutti gli archivi.
   - Per limitare l'installazione ai repository specifici selezionati, scegli Seleziona solo i repository. Assicurati di includere il repository per l'app di cui stai migrando nei repository selezionati.
8. Scegli Installa e autorizza.
9. Verrai reindirizzato alla pagina Installa e autorizza GitHub l'app per la tua app nella console Amplify. Se GitHub l'autorizzazione ha esito positivo, visualizzerai un messaggio di esito positivo. Scegli Avanti.

10. Nella pagina Installazione completa, scegli Installazione completa. Questo passaggio elimina il webhook esistente, ne crea uno nuovo e completa la migrazione.

Configurazione dell'app GitHub Amplify per le implementazioni CLI AWS CloudFormation e SDK

Le app Amplify esistenti che hai collegato in precedenza GitHub dai repository vengono utilizzate per l'accesso ai repository OAuth. Ciò può includere le applicazioni distribuite utilizzando l'interfaccia a riga di comando (CLI) o. AWS CloudFormation SDKs Si consiglia fortemente di effettuare la migrazione di queste applicazioni per usare la nuova applicazione Amplify. GitHub La migrazione deve essere eseguita nella console Amplify in. AWS Management Console Per istruzioni, consultare Migrazione di un'OAuthapp esistente all'app Amplify GitHub .

È possibile utilizzare l' AWS CloudFormation Amplify CLI SDKs e distribuire una nuova app Amplify che utilizza l'app per l'accesso ai repository. GitHub Questo processo richiede che tu installi prima l'app GitHub Amplify nel tuo account. GitHub Successivamente, dovrai generare un token di accesso personale nel tuo GitHub account. Infine, distribuisci l'app e specifica il token di accesso personale.

Installa l'app GitHub Amplify nel tuo account

1. Apri un browser Web e vai alla posizione di installazione dell'app GitHub Amplify nella regione in cui distribuirai AWS l'app.

   Usa il formato`https://github.com/apps/aws-amplify-`*`REGION`*`/installations/new`, sostituendolo *`REGION`* con il tuo input. Ad esempio, se installi la tua app nella regione Stati Uniti occidentali (Oregon), specifica`https://github.com/apps/aws-amplify-us-west-2/installations/new`.
2. Seleziona l' GitHub account in cui desideri installare l'app Amplify. GitHub
3. Esegui una di queste operazioni:
   - Per applicare l'installazione a tutti i repository, scegli Tutti gli archivi.
   - Per limitare l'installazione ai repository specifici selezionati, scegli Seleziona solo i repository. Assicurati di includere il repository per l'app di cui stai migrando nei repository selezionati.
4. Scegli Installa.

Genera un token di accesso personale nel tuo account GitHub

1. Accedi al tuo GitHub account.
2. Nell'angolo in alto a destra, individua la foto del profilo e scegli Impostazioni dal menu.
3. Dal menu di navigazione a sinistra, scegli Impostazioni sviluppatore.
4. Nella pagina GitHub App, nel menu di navigazione a sinistra, scegli Token di accesso personali.
5. Nella pagina Token di accesso personali, scegli Genera nuovo token.
6. Nella pagina Nuovo token di accesso personale, in Nota inserisci un nome descrittivo per il token.
7. Nella sezione Seleziona gli ambiti, seleziona admin:repo_hook.
8. Scegli Generate token (Genera token).
9. Copia e salva il token di accesso personale. Dovrai fornirlo quando distribuisci un'app Amplify con la CLI o il. AWS CloudFormation SDKs

Dopo aver installato l'app GitHub Amplify nel GitHub tuo account e aver generato un token di accesso personale, puoi distribuire una nuova app con la CLI di Amplify o il. AWS CloudFormation SDKs Utilizzare il `accessToken` campo per specificare il token di accesso personale creato nella procedura precedente. Per ulteriori informazioni, consulta il riferimento CreateAppall'API Amplify AWS::Amplify::Appe la Guida per l'utente.AWS CloudFormation

Il seguente comando CLI distribuisce una nuova app Amplify che utilizza l'app per l'accesso al repository. GitHub Sostituisci e *`myapp-using-githubapp`* con *`https://github.com/Myaccount/react-app`* le tue informazioni. *`MY_TOKEN`*

```
aws amplify create-app --name myapp-using-githubapp --repository https://github.com/
Myaccount/react-app --access-token MY_TOKEN
```

Configurazione delle anteprime web con l'app Amplify GitHub

Un'anteprima web distribuisce ogni pull request (PR) inviata al tuo GitHub repository in un URL di anteprima unico. Le anteprime ora utilizzano l'app GitHub Amplify per accedere al tuo repository. GitHub Per istruzioni sull'installazione e l'autorizzazione dell' GitHub App per le anteprime web, consulta. Abilita le anteprime web per le richieste pull

AWS Amplify Riferimento all'hosting

Utilizza gli argomenti di questa sezione per trovare materiale di riferimento dettagliato per. AWS Amplify

Argomenti

- AWS CloudFormation supporto
- AWS Command Line Interface supporto
- Supporto per l'etichettatura delle risorse
- API di hosting Amplify

AWS CloudFormation supporto

Utilizza AWS CloudFormation i modelli per effettuare il provisioning delle risorse Amplify, abilitando implementazioni di app Web ripetibili e affidabili. AWS CloudFormation fornisce un linguaggio comune per descrivere e fornire tutte le risorse dell'infrastruttura nel tuo ambiente cloud e semplifica l'implementazione su più AWS account e/o regioni con solo un paio di clic.

Per Amplify Hosting, consulta la documentazione di Amplify. CloudFormation Per Amplify Studio, consulta la documentazione di Amplify UI Builder. CloudFormation

AWS Command Line Interface supporto

Usa AWS Command Line Interface per creare app Amplify a livello di codice dalla riga di comando. Per informazioni, consulta la documentazione.AWS CLI

Supporto per l'etichettatura delle risorse

Puoi usare il AWS Command Line Interface per etichettare le risorse Amplify. Per ulteriori informazioni, consulta la documentazione di AWS CLI tag-resource.

API di hosting Amplify

Questo riferimento fornisce descrizioni delle azioni e dei tipi di dati per l'API Amplify Hosting. Per ulteriori informazioni, consulta la documentazione di riferimento dell'API Amplify.

Cronologia dei documenti per AWS Amplify

Nella tabella seguente vengono descritte le modifiche importanti apportate alla documentazione dall'ultima versione di AWS Amplify.

- Ultimo aggiornamento della documentazione: 28 maggio 2025

Modifica	Descrizione	Data
Capitolo aggiornato sulle impostazioni e sulla configura zione di Build	È stato aggiornato il Gestione della configurazione di build per un'applicazione Amplify capitolo per descrivere la nuova funzionalità configura bile di compilazione del tipo di istanza che consente di scegliere un tipo di istanza che fornisca all'applicazione le risorse di CPU, memoria e spazio su disco necessarie.	28 maggio 2025
Capitolo firewall aggiornato	È stato aggiornato il Supporto firewall per i siti ospitati da Amplify capitolo per descriver e la disponibilità generale (GA) dell'integrazione di Amplify con AWS WAF, inclusa la funzionalità GA e la struttura dei prezzi.	26 marzo 2025
Nuovo capitolo sulla protezion e Skew	È stato aggiunto il Protezion e antiinclinazione per le implementazioni Amplify capitolo per descrivere la funzionalità di protezione dall'inclinazione che elimina i	10 marzo 2025

Modifica	Descrizione	Data
	problemi di distorsione della versione tra client e server nelle applicazioni web Amplify.	
Capitolo Webhooks aggiornato	È stato aggiunto l'Webhook unificati per repository Gitargomento per descriver e la funzionalità dei webhook unificati che utilizza un webhook completo per tutte le applicazioni Amplify associate a un singolo repository Git.	10 marzo 2025
Novità: aggiunta di un ruolo SSR Compute per consentir e l'accesso alle risorse. Argomento AWS	È stato aggiunto l'Aggiunger e un ruolo SSR Compute per consentire l'accesso alle risorse AWSargomento per descrivere come creare e associare un ruolo Amplify SSR Compute a un'app per consentire al servizio Amplify Compute di accedere alle risorse. AWS	17 febbraio 2025
Nuovo capitolo «Utilizzo AWS WAF per proteggere le app Amplify»	È stato aggiunto il Supporto firewall per i siti ospitati da Amplify capitolo per descriver e l'integrazione di Amplify con AWS WAF (in anteprima) che consente di protegger e le applicazioni Web con un elenco di controllo degli accessi Web (Web ACL).	18 dicembre 2024

Modifica	Descrizione	Data
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	14 novembre 2024
Supporto Amplify aggiornato per l'argomento Next.js	È stato aggiornato l'Supporto Amplify per Next.jsargomento per descrivere il supporto di Amplify per la versione 15 di Next.js.	6 novembre 2024
Novità: capitolo sulla distribuzione di un sito web statico su Amplify da Amazon S3	È stato aggiunto il Distribuzione di un sito Web statico su Amplify da un bucket Amazon S3 capitolo per descrivere la nuova integrazione di Amplify con Amazon S3 che consente di ospitare contenuti statici di siti Web archiviati S3 in pochi clic.	16 ottobre 2024
Nuovo capitolo sulla gestione della configurazione della cache	È stato aggiunto il Gestione della configurazione della cache per un'app capitolo per descrivere il comportamento di memorizzazione nella cache predefinito di Amplify e come applica le politiche di gestione della cache ai contenuti.	13 agosto 2024

Modifica	Descrizione	Data
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	18 luglio 2024
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	31 maggio 2024
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	17 aprile 2024
Nozioni di base aggiornate	È stato aggiornato il Guida introduttiva alla distribuzione di un'app su Amplify Hosting capitolo per utilizzare un'applic azione di esempio Next.js nel tutorial.	12 aprile 2024
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	5 aprile 2024

Modifica	Descrizione	Data
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	4 aprile 2024
Nuovo capitolo sulla risoluzio ne dei problemi	È stato aggiunto il Risoluzio ne dei problemi di Amplify capitolo per descrivere come risolvere i problemi riscontrati con le applicazioni distribuite su Amplify Hosting.	2 aprile 2024
Nuovo supporto per i certificati SSL/TLS ad hoc	È stato aggiunto l'Utilizzo dei certificati SSL/TLSargomento al Connessione di un dominio personalizzato capitolo per descrivere il supporto di Amplify per i certificati SSL/TLS personalizzati durante la connessione di un'app a un dominio personalizzato.	20 febbraio 2024
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	2 gennaio 2024

Modifica	Descrizione	Data
Nuovo supporto per i framework SSR	È stato aggiornato l'Implementazione di applicazioni renderizzate lato server con Amplify Hostingargomento per descrivere il supporto Amplify per qualsiasi framework SSR basato su JavaScript con un adattatore open source.	19 novembre 2023
Nuovo supporto per il lancio della funzionalità di ottimizzazione delle immagini	È stato aggiunto l'Ottimizzazione delle immagini per le app SSRargomento per descrivere il supporto integrato per l'ottimizzazione delle immagini per le app renderizzate lato server.	19 novembre 2023
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	17 novembre 2023
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	6 novembre 2023

Modifica	Descrizione	Data
Nuovo argomento relativo ai sottodomini wildcard	È stato aggiunto l'Configurazione di sottodomini wildcardargomento per descrivere il supporto per i sottodomini wildcard nei domini personalizzati.	6 novembre 2023
Nuove policy gestite da	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere la nuova politica AmplifyBackendDeployFullAccess AWS gestita per Amplify.	8 ottobre 2023
Lancio della nuova funzionalità di supporto per i framework monorepo	È stato aggiornato l'Configurazione delle impostazioni di build monorepoargomento per descrivere il supporto per la distribuzione di app in monorepos create utilizzando npm workspace, pnpm workspace, Yarn workspace, Nx e Turborepo.	19 giugno 2023
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	1 giugno 2023

Modifica	Descrizione	Data
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	24 febbraio 2023
Capitolo sul rendering lato server aggiornato	È stato aggiornato il Implementazione di applicazioni renderizzate lato server con Amplify Hosting capitolo per descrivere le recenti modifiche al supporto di Amplify per le versioni 12 e 13 di Next.js.	17 novembre 2022
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	30 agosto 2022
Argomento delle policy gestite	È stato aggiornato l'Creazione di un backend per un'applicazioneargomento per descrivere come implementare un backend utilizzando Amplify Studio.	23 agosto 2022

Modifica	Descrizione	Data
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	27 aprile 2022
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	17 aprile 2022
Lancio della nuova funzionalità GitHub dell'app	È stato aggiunto l'Configurazione dell'accesso Amplify ai repository GitHubargomento per descrivere la nuova GitHub app per autorizzare l'accesso di Amplify al tuo repository. GitHub	5 aprile 2022
Lancio della nuova funzionalità Amplify Studio	L'Benvenuto su AWS Amplify Hostingargomento è stato aggiornato per descrivere gli aggiornamenti di Amplify Studio che forniscono un visual designer per creare componenti dell'interfaccia utente che è possibile connettere ai dati di backend.	2 dicembre 2021

Modifica	Descrizione	Data
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify per supportare Amplify Studio.	2 dicembre 2021
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	8 novembre 2021
Argomento delle policy gestite	È stato aggiornato l'AWS politiche gestite per AWS Amplifyargomento per descrivere le recenti modifiche alle politiche AWS gestite per Amplify.	27 settembre 2021
Argomento delle policy gestite	È stato aggiunto l'AWS politiche gestite per AWS Amplifyargomento per descrivere le policy AWS gestite da per Amplify e modifiche recenti a tali policy.	28 luglio 2021

Modifica	Descrizione	Data
Capitolo aggiornato sul rendering lato server	È stato aggiornato il Implementazione di applicazioni renderizzate lato server con Amplify Hosting capitolo per descrivere il nuovo supporto per la versione 10 di Next.js. x. x e Next.js versione 11.	22 luglio 2021
Aggiornato il capitolo Configurazione delle impostazioni di build	È stato aggiunto l'Configurazione delle impostazioni di build monorepoargomento per descrivere come configurare le impostazioni di build e la nuova variabile di `AMPLIFY_MONOREPO_APP_ROOT` ambiente durante la distribuzione di un'app monorepo con Amplify.	20 luglio 2021

Modifica	Descrizione	Data
Capitolo aggiornato sulle distribuzioni delle filiali Feature	È stato aggiunto l'Generazione automatica in fase di compilazione della configurazione Amplify (solo app di prima generazione)argomento per descrivere come generare automaticamente il file in fase di compilazione. `aws-exports.js` È stato aggiunto l'Build di backend condizionali (solo app di prima generazione)argomento per descriver e come abilitare le build condizionali di backend. È stato aggiunto l'Usa i backend Amplify tra le app (solo app di prima generazione)argomento per descrivere come riutilizzare i backend esistenti quando si crea una nuova app, si collega un nuovo ramo a un'app esistente o si aggiorna un frontend esistente in modo che punti a un ambiente di backend diverso.	30 giugno 2021
Capitolo sulla sicurezza aggiornato	È stato aggiunto l'Protezione dei dati in Amplifyargomento per descrivere come applicare il modello di responsabilità condivisa e come Amplify utilizza la crittografia per proteggere i dati a riposo e in transito.	3 giugno 2021

Modifica	Descrizione	Data
Nuovo supporto per il lancio della funzionalità SSR	È stato aggiunto il Implementazione di applicazioni renderizzate lato server con Amplify Hosting capitolo per descrivere il supporto di Amplify per le app Web che utilizzano il rendering lato server (SSR) e vengono create con Next.js.	18 maggio 2021
Nuovo capitolo sulla sicurezza	È stato aggiunto il Sicurezza in Amplify capitolo per descriver e come applicare il modello di responsabilità condivisa quando si utilizza Amplify e la modalità di configurazione di Amplify per soddisfare i propri obiettivi di sicurezza e conformità.	26 marzo 2021
Argomento delle build ad hoc aggiornate	È stato aggiornato l'argomento Immagini di build personalizzate e aggiornamenti live dei pacchetti per descrivere come configurare un'immagine di build personalizzata ospitata in Amazon Elastic Container Registry Public.	12 marzo 2021
Argomento monitoraggio aggiornato	È stato aggiornato l'argomento Monitoraggio per descriver e come accedere ai dati dei CloudWatch parametri di Amazon e impostare allarmi.	2 febbraio 2021

Modifica	Descrizione	Data
Nuovo argomento CloudTrail sulla registrazione	Sono state aggiunte le chiamate all'API Logging Amplify AWS CloudTrai l utilizzando l'argomen to per descrivere AWS CloudTrail come acquisisce e registra tutte le azioni API per AWS Amplify Console API Reference e Admin UI API Reference. AWS Amplify	2 febbraio 2021
Lancio della nuova funzional ità dell'interfaccia utente di amministrazione	È stato aggiornato l'Benvenuto su AWS Amplify Hostingargomento per descrivere la nuova interfacc ia utente di amministrazione che fornisce un'interfaccia visiva per gli sviluppatori web e mobili di frontend per creare e gestire i backend delle app all'esterno di. AWS Management Console	1 dicembre 2020
Lancio di una nuova funzional ità in modalità performance	È stato aggiornato l'argomen to Gestione delle prestazio ni delle app per descrivere come abilitare la modalità a prestazioni per ottimizza re prestazioni di hosting più rapide.	4 novembre 2020

Modifica	Descrizione	Data
È stato aggiornato l'argomento delle intestazioni personalizzate	È stato aggiornato l'argomento Intestazioni personalizzate per descrivere come definire intestazioni personalizzate per un'app Amplify utilizzando la console o modificando un file YML.	28 ottobre 2020
Avvio della nuova funzionalità di sottodomini automatici	È stato aggiunto l'argomento Configurazione di sottodomini automatici per un dominio personalizzato Route 53 per descrivere come utilizzare le distribuzioni di feature branch basate su pattern per un'app connessa a un dominio personalizzato Amazon Route 53. È stato aggiunto l'argomento Accesso all'anteprima Web con sottodomini per descrivere come configurare le anteprime Web delle richieste pull in modo che siano accessibili con i sottodomini.	20 giugno 2020
Nuovo argomento sulle notifiche	È stato aggiunto l'argomento Notifiche per descrivere come configurare le notifiche e-mail per un'app Amplify per avvisare le parti interessate o i membri del team quando una build ha successo o fallisce.	20 giugno 2020

Modifica	Descrizione	Data
Aggiornato l'argomento sui domini personalizzati	È stato aggiornato l'Connessione di un dominio personalizzatoargomento per migliorare le procedure per l'aggiunta di domini personalizzati in Amazon Route 53 e Google Domains. GoDaddy Questo aggiornamento include anche nuove informazioni sulla risoluzione dei problemi per la configurazione di domini personalizzati.	12 maggio 2020
AWS Amplify versione	In questa release è stata introdotta Amplify.	26 novembre 2018

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.