Guida per gli sviluppatori

AWS App Runner

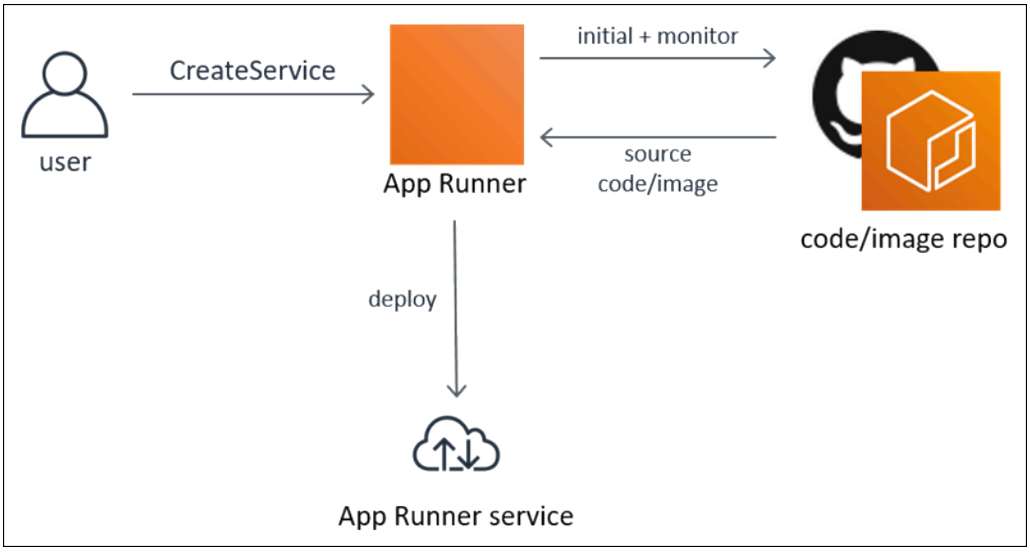
AWS App Runner: Guida per gli sviluppatori

Table of Contents

Che cosa è AWS App Runner?

AWS App Runnerè unAWSche fornisce un modo rapido, semplice e conveniente per distribuire dal codice sorgente o da un'immagine contenitore direttamente a un'applicazione Web scalabile e sicura nelAWSCloud. Non è necessario apprendere nuove tecnologie, decidere quale servizio di elaborazione utilizzare o sapere come eseguire il provisioning e la configurazioneAWSrisorse AWS.

App Runner si connette direttamente al tuo repository di codice o immagini. Fornisce una pipeline di integrazione e distribuzione automatica con operazioni completamente gestite, prestazioni elevate, scalabilità e sicurezza.



Per chi è App Runner?

Se sei unsviluppatore, è possibile utilizzare App Runner per semplificare il processo di distribuzione di una nuova versione del repository di codice o immagini.

PerTeam operativi, App Runner abilita le distribuzioni automatiche ogni volta che un commit viene inviato al repository del codice o una nuova versione dell'immagine contenitore viene spinta nel repository delle immagini.

Accesso a App Runner

È possibile definire e configurare le distribuzioni del servizio App Runner utilizzando una delle seguenti interfacce:

- Console di App Runner: fornisce un'interfaccia web per la gestione dei servizi App Runner.
- API di App Runner: fornisce un'API RESTful per l'esecuzione delle azioni di App Runner. Per ulteriori informazioni, consulta la Documentazione di riferimento delle API AWS App Runner.
- AWSInterfaccia a riga di comando (AWS CLI)— Forniscono comandi per un'ampia gamma diAWSServizi, incluso Amazon VPC, ed è supportata su Windows, macOS e Linux. Per ulteriori informazioni, consulta AWS Command Line Interface.
- AWSSDK: offre interfacce API specifiche per ogni lingua e si occupano di molti dettagli della connessione, ad esempio il calcolo delle firme, la gestione dei tentativi di richiesta e degli errori. Per ulteriori informazioni, consulta SDK AWS.

Prezzi per App Runner

App Runner offre un modo conveniente per eseguire l'applicazione. Paghi solo per le risorse consumate dal tuo servizio App Runner. Il servizio si riduce a un numero inferiore di istanze di calcolo quando il traffico delle richieste è più lento. È possibile controllare le impostazioni di scalabilità: il numero più basso e più alto di istanze con provisioning e il carico più alto gestito da un'istanza.

Per ulteriori informazioni sul dimensionamento automatico di App Runner, consultathe section called “Auto Scaling”: .

Per informazioni sui prezzi, consulta Prezzi di AWS App Runner.

Fasi successive

Ulteriori informazioni su come iniziare a utilizzare App Runner nei seguenti argomenti:

- Configurazione: completa i passaggi prerequisiti per l'utilizzo di App Runner.
- Nozioni di base: distribuisci la tua prima applicazione su App Runner.

Configurazione per App Runner

Se sei un nuovoAWS, completare i prerequisiti di installazione elencati in questa pagina prima di iniziare a utilizzareAWS App Runner: .

Per queste procedure di installazione, è possibile utilizzare ilAWS Identity and Access Management(IAM) servizio. Per informazioni complete su IAM, consulta i seguenti materiali di riferimento:

- AWS Identity and Access Management (IAM)
- IAM User Guide

Creazione di un Account AWS

Quando ti iscrivi conAWS, si ottiene un numero di account con accesso a tutti i servizi cheAWSofferte, tra cuiAWS App Runner: .

Se già disponi di unAccount AWSPassare al prerequisito successivo.

Se non disponi di unAWSPer crearne uno, completa le fasi seguenti.

Per registrarti per ottenere un account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

   Come parte della procedura di registrazione riceverai una telefonata, durante la quale dovrai inserire un codice di verifica sulla tastiera del telefono.

Creazione di un utente IAM

Per accedere a unAWSServizio, fornisci le credenziali. Queste credenziali determinanoauthentication(chi sei) eAutorizzazione(quali autorizzazioni hai per eseguire azioni suAWSrisorse).

Quando crei per la prima volta un Amazon Web Services (AWS), si inizia con un'identità di accesso singolo. Tale identità ha accesso completo a tutti i servizi e le risorse AWS presenti nell'account.

Questa identità è denominataAWSaccountUtente root: . Quando accedi, immetti l'indirizzo e-mail e la password utilizzati per creare l'account.

> **Important**
>
> È vivamente consigliato di non utilizzare l'utente root per le attività quotidiane, anche quelle amministrative. Rispetta piuttosto la best practice di utilizzare l'utente root soltanto per creare il tuo primo utente &IAM;. Quindi conserva al sicuro le credenziali dell'utente root e utilizzale per eseguire solo alcune attività di gestione dell'account e del servizio. Per visualizzare le attività che richiedono l'accesso come utente root, consultaAttività che richiedono le credenziali dell'utente root: .

Per ulteriori informazioni sulle credenziali dell'utente root e dell'utente IAM, consultaAccount AWScredenziali utente root e credenziali utente IAMnellaAWSRiferimenti generali: .

Per creare un utente amministratore per se stessi e aggiungere l'utente a un gruppo di amministratori (console)

1. Accedi allaConsole IAMcome proprietario dell'account scegliendoUtente roote inserendo ilAWSL'indirizzo e-mail dell'account. Nella pagina successiva, inserisci la password.

   > **Note**
   >
   > È fortemente consigliato rispettare la best practice di utilizzare l'**`Administrator`**Utente IAM che segue e blocca in un luogo sicuro le credenziali dell'utente root. Accedere come utente root solo per eseguire alcune attività di gestione dell'account e del servizio.

2. Nel riquadro di navigazione selezionare Users (Utenti), quindi selezionare Add user (Aggiungi utente).
3. In User name (Nome utente), immettere **`Administrator`**.
4. Selezionare la casella di controllo accanto aAWS Management ConsoleAccesso: . Quindi, selezionare Custom password (Password personalizzata)e immettere la nuova password nella casella di testo.
5. (Facoltativo) Per impostazione predefinita,AWSÈ necessario che il nuovo utente crei una nuova password al primo accesso. Puoi deselezionare la casella di controllo accanto a User must create a new password at next sign-in (L'utente deve creare una nuova password al prossimo

accesso) per consentire al nuovo utente di reimpostare la propria password dopo aver effettuato l'accesso.

6. Seleziona Successivo: Autorizzazioni.
7. In Set permissions (Imposta autorizzazioni), selezionare Add user to group (Aggiungi l'utente al gruppo).
8. Seleziona Create group (Crea gruppo).
9. Nella finestra di dialogo Create group (Crea gruppo), per Group name (Nome gruppo) immettere **`Administrators`**.
10. ScegliereFiltrare policye quindi selezionareAWSgestita - funzione di lavoroper filtrare il contenuto della tabella.
11. Nell'elenco delle policy, selezionare la casella di controllo accanto ad AdministratorAccess. Seleziona quindi Create group (Crea gruppo).

> **Note**
>
> È necessario attivare l'accesso degli utenti e dei ruoli IAM alla fatturazione prima di poter utilizzare l'`AdministratorAccess`Autorizzazioni per accedere allaAWS Billing and Cost ManagementConsole A questo scopo, seguire le istruzioni nella fase 1 del tutorial sulla delega dell'accesso alla console di fatturazione.

12. Nell'elenco dei gruppi seleziona la casella di controllo per il tuo nuovo gruppo. Se necessario, selezionare Refresh (Aggiorna) per visualizzare il gruppo nell'elenco.
13. Seleziona Successivo: Tag.
14. (Facoltativo) Aggiungere metadati all'utente collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo di tag in IAM, consultare Tagging di utenti e ruoli IAM nella Guida per l'utente di IAM.
15. Seleziona Successivo: Review (Revisione)Per visualizzare l'elenco delle appartenenze ai gruppi da aggiungere al nuovo utente. Quando sei pronto per continuare, seleziona Create user (Crea utente).

È possibile utilizzare questa stessa procedura per creare altri gruppi e utenti e per concedere agli utenti l'accesso alle risorse del proprio account AWS. Per ulteriori informazioni sull'utilizzo di policy che limitano le autorizzazioni degli utenti a specificiAWSrisorse, consultaGestione degli accessieEsempio di policy: .

> **Important**
>
> Proteggi il tuoAccount AWS: . Non inviare o condividere mai le tue credenziali con nessuno al di fuori dell'organizzazione. Nessuno che rappresenta legittimamente Amazon richiederà mai le tue credenziali.

Dopo aver creato l'utente IAM, utilizza le credenziali per accedere allaAWS Management Console: . Per ulteriori informazioni, consultaIn che modo gli utenti IAM effettuano l'accesso allaAccount AWSnellaGuida per l'utente di IAM: .

Creare una chiave di accesso per l'utente IAM

Le chiavi di accesso sono composte da un ID chiave di accesso e una chiave di accesso segreta che vengono utilizzati per firmare le richieste a livello di programmazione che si effettuano adAWS: . Se non disponi di chiavi di accesso, puoi crearle dallaAWS Management Console: . Come best practice, non utilizzare l'AWSLe chiavi di accesso utente root dell'account per le attività in cui non sia necessario. Invece,Creare un nuovo utente IAM amministratorecon le chiavi di accesso per te.

L'unica volta in cui è possibile visualizzare o scaricare la chiave di accesso segreta è quando si creano le chiavi. Non puoi recuperarle successivamente. Tuttavia, è possibile creare nuove chiavi di accesso in qualsiasi momento. È inoltre necessario disporre delle autorizzazioni per effettuare le operazioni richieste dalle azioni IAM. Per ulteriori informazioni, consultaAutorizzazioni necessarie per accedere alle risorse IAMnellaGuida per l'utente di IAM: .

Per creare le chiavi di accesso per un utente IAM

1. Accedi alla AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Users (Utenti).
3. Selezionare il nome dell'utente sul quale si devono creare le chiavi di accesso e selezionare la casella di controlloCredenziali di sicurezzaScheda
4. NellaChiavi di accesso, scegliereCreare la chiave di accesso: .
5. Per visualizzare la nuova key pair di accesso, selezionaMostra: . Non sarà possibile accedere nuovamente alla chiave di accesso segreta dopo la chiusura di questa finestra di dialogo. Le credenziali saranno simili a quanto segue:

- ID chiave di accesso: AKIAIOSFODNN7EXAMPLE
- Chiave di accesso segreta: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

6. Per scaricare la coppia di chiavi, seleziona Download .csv file (Scarica file .csv). Conserva le chiavi in un posto sicuro. Non sarà possibile accedere nuovamente alla chiave di accesso segreta dopo la chiusura di questa finestra di dialogo.

   Mantieni la riservatezza delle chiavi per proteggere il tuoAWSaccount e non inviarle mai per e-mail. Non condividerle all'esterno della tua organizzazione, anche se ricevi una richiesta che sembra provenire da AWS o Amazon.com. Nessuno che rappresenta legittimamente Amazon richiederà mai la tua chiave segreta.
7. Dopo aver scaricato il`.csv`, scegliereChiudi: . Quando si crea una chiave di accesso, la coppia di chiavi è attiva per impostazione predefinita ed è possibile utilizzare la coppia immediatamente.

Argomenti correlati

- Che cos'è IAM?nellaGuida per l'utente di IAM
- AWSCredenziali di sicurezzainAWSRiferimenti generali

Fasi successive

Sono stati completati i passaggi dei prerequisiti. Per distribuire la prima applicazione in App Runner, vedereNozioni di base: .

Nzioni di base su App Runner

AWS App Runnerè unAWSche fornisce un modo rapido, semplice e conveniente per trasformare un'immagine contenitore o un codice sorgente esistente direttamente in un servizio Web in esecuzione nelCloud AWS: .

Questa esercitazione illustra come utilizzareAWS App Runnerper distribuire l'applicazione in un servizio App Runner. Viene descritta la configurazione del codice sorgente e della distribuzione, la compilazione del servizio e il runtime del servizio. Viene inoltre illustrato come distribuire una versione del codice, apportare una modifica alla configurazione e visualizzare i registri. Infine, nell'esercitazione viene illustrato come ripulire le risorse create mentre si seguono le procedure dell'esercitazione.

Argomenti

Prerequisites

Prima di iniziare l'esercitazione, assicurati di eseguire le seguenti azioni:

1. Completa le fasi di configurazione descritte inConfigurazione: .
2. Creazione di unGitHubSe non ne hai già uno. Se non hai mai usato GitHub, consultaGuida introduttiva a GitHubnellaDocumenti GitHub: .
3. Creare un archivio nel tuo account GitHub. Questa esercitazione utilizza il nome del repository`python-hello`: . Creare i file nella directory principale del repository, con i nomi e il contenuto specificati negli esempi seguenti.

File per il**python-hello**Repository

Example requirements.txt

```
Click==7.0
Flask==1.0.2
itsdangerous==1.1.0
Jinja2==2.10
MarkupSafe==1.1.1
Werkzeug==0.15.5
```

Example server.py

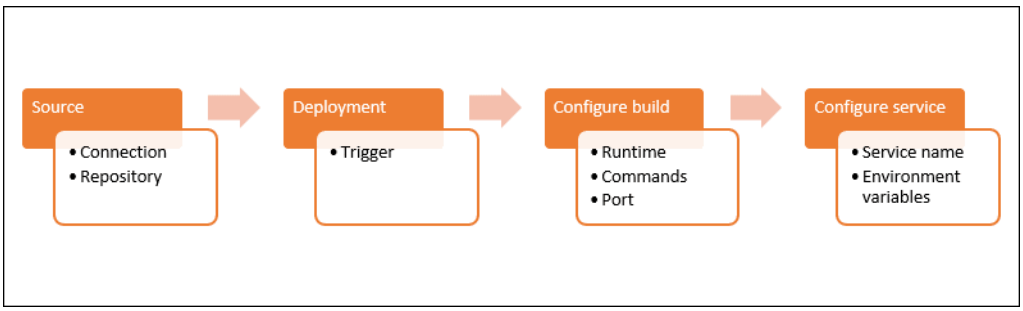
```
from flask import Flask
import os

PORT = 8080
name = os.environ['NAME']
if name == None or len(name) == 0:
  name = "world"
MESSAGE = "Hello, " + name + "!"
print("Message: '" + MESSAGE + "'")

app = Flask(__name__)


@app.route("/")
def root():
  print("Handling web request. Returning message.")
  result = MESSAGE.encode("utf-8")
  return result


if __name__ == "__main__":
  app.run(debug=True, host="0.0.0.0", port=PORT)
```

Fase 1: Creare un servizio App Runner

In questo passaggio si crea un servizio App Runner basato sul repository del codice sorgente di esempio creato su GitHub come parte dithe section called “Prerequisites”: . L'esempio contiene un semplice sito web Python. Questi sono i passaggi principali da seguire per creare un servizio:

1. Configurare il codice sorgente.
2. Configurare la distribuzione di origine.
3. Configurare la compilazione dell'applicazione.
4. Configurare il servizio.
5. Rivedere e confermare.

Il diagramma seguente illustra i passaggi per la creazione di un servizio App Runner:

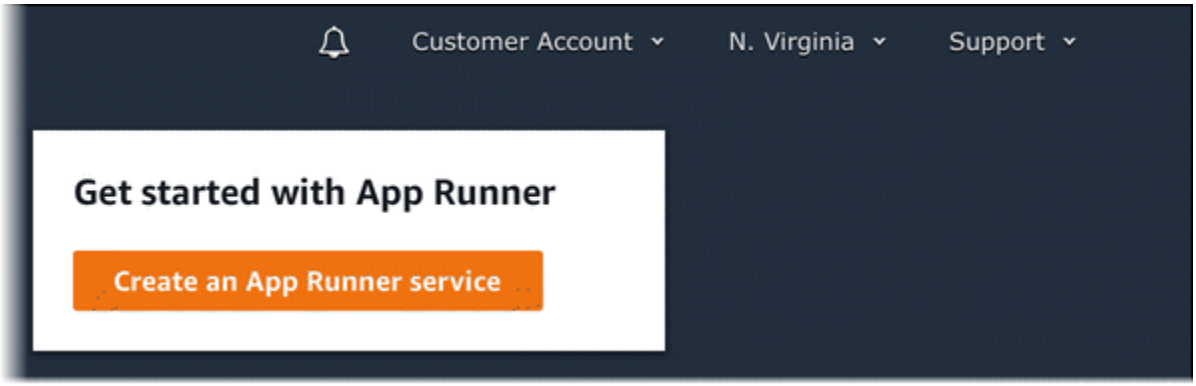


Per creare un servizio App Runner basato su un repository di codice sorgente

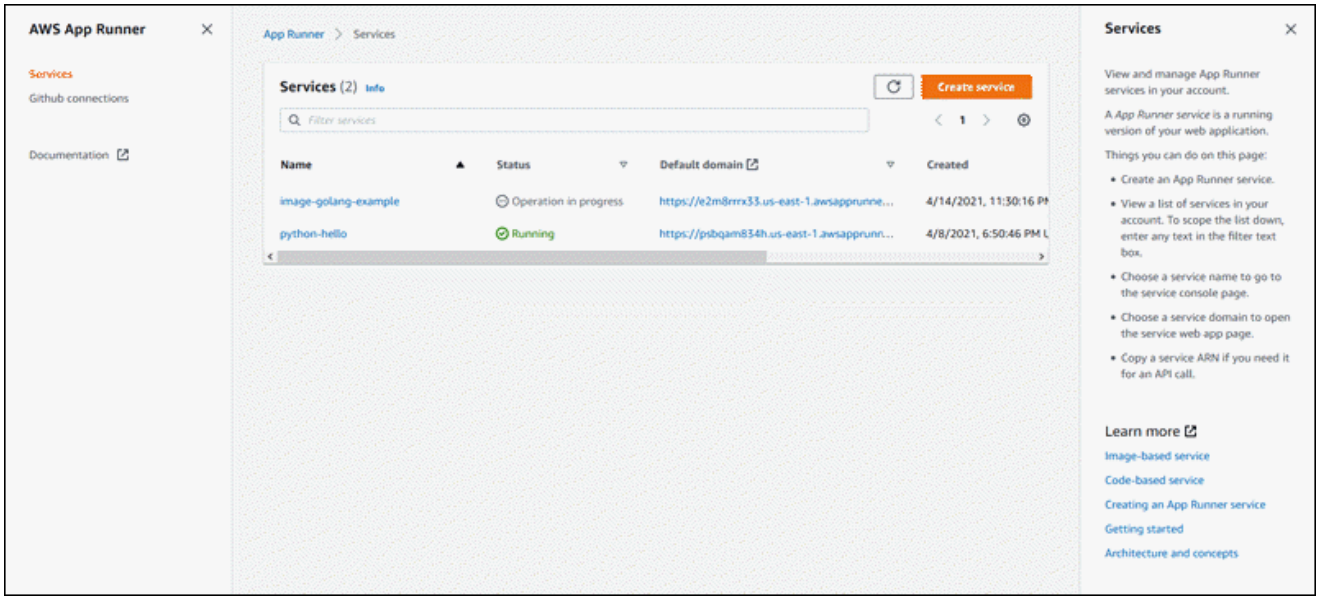
1. Configurare il codice sorgente.

   a. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .

   b. Se il fileAccount AWSnon dispone ancora di servizi App Runner, viene visualizzata la home page della console. ScegliereCreare un servizio App Runner: .



   Se il fileAccount AWSdispone di servizi esistenti, ilServiziViene visualizzata una lista dei servizi. Selezionare Create service (Crea servizio).



c. SulOrigine e distribuzione, nellaCreaSezione, perRepository type (Tipo di repository), scegliereRepository codice sorgente: .

d. UNDERConnect a GitHubscegliAggiungi nuovoE quindi, se richiesto, fornisci le credenziali GitHub.

> **Note**
>
> La procedura seguente per installare ilAWSIl connettore per GitHub al tuo account GitHub sono passaggi una tantum. È possibile riutilizzare la connessione per creare più servizi App Runner in base ai repository di questo account. Quando si dispone di una connessione esistente, sceglierla e passare alla selezione del repository.

e. NellaInstallaAWSConnettore per GitHub, se richiesto, scegli il nome dell'account GitHub.

f. Se viene richiesto di autorizzare ilAWSConnettore per GitHub, scegliereAutorizza connessioni AWS: .

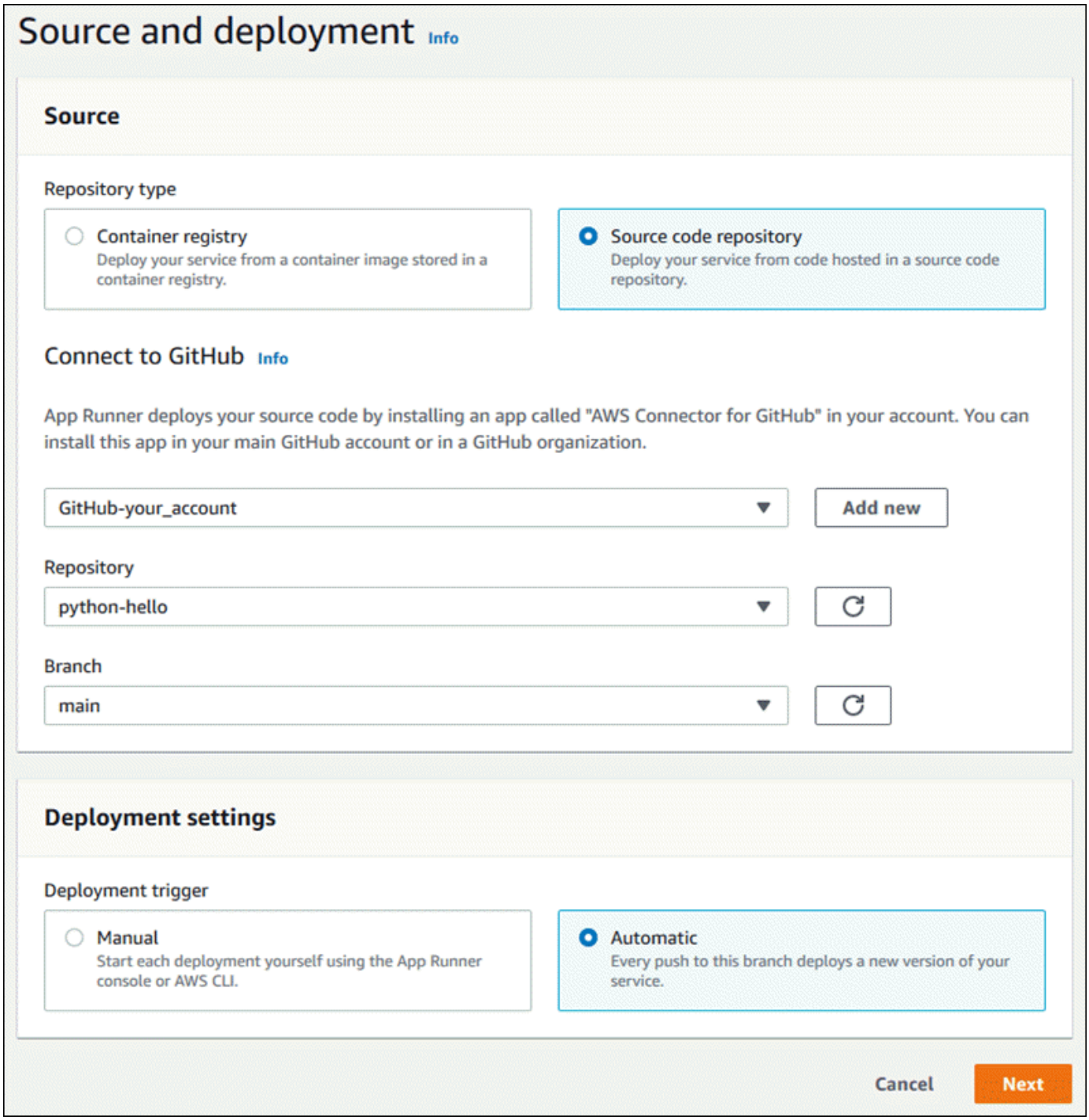
g. Scegli Install (Installa).

Il nome dell'account viene visualizzato comeConto/organizzazione GitHub: . Ora puoi scegliere un archivio nel tuo account.

h. PerRepository, scegliere il repository di esempio creato,`python-hello`: . PerRamoScegli il nome del ramo predefinito del repository (ad esempio,Principale).

2. Configurare le distribuzioni: NellaImpostazioni di distribuzione, sceglieriAutomatic, quindi sceglieriSuccessivo: .

> **Note**
>
> Con la distribuzione automatica, ogni nuovo commit nel repository distribuisce automaticamente una nuova versione del servizio.



3. Configurare la compilazione dell'applicazione.

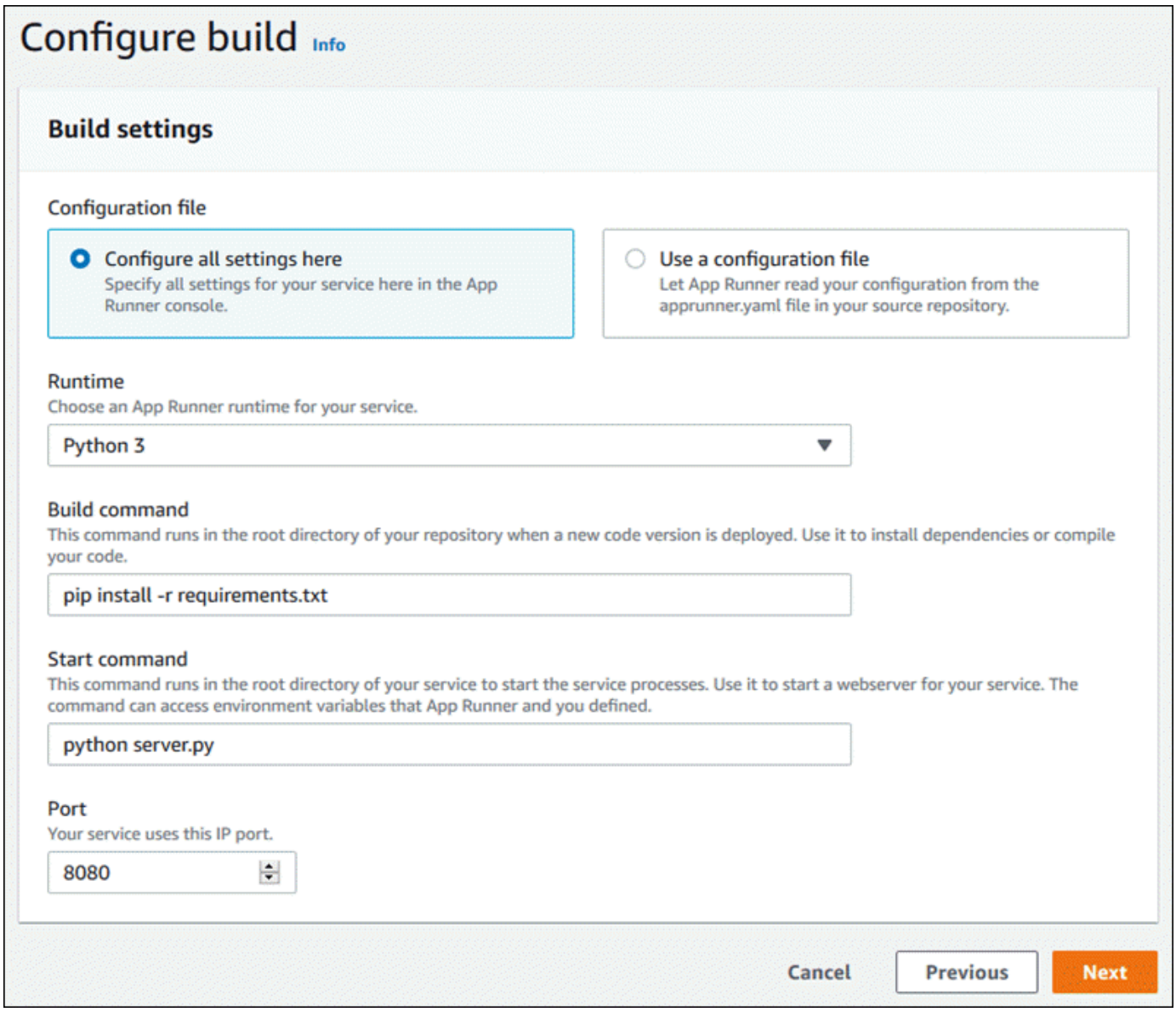
   a. SulConfigura compilazionePagina, perFile di configurazione, scegliereConfigura qui tutte le impostazioni: .

   b. Fornire le seguenti impostazioni di compilazione:

      - Runtime— ScegliPython 3: .
      - Comando Compila— Invio**`pip install -r requirements.txt`**: .
      - Comando Avvia— Invio**`python server.py`**: .
      - Porta— Invio**`8080`**: .

   c. Seleziona Successivo.

> **Note**
>
> Il runtime di Python 3 crea un'immagine Docker utilizzando un'immagine Python 3 di base e il codice Python di esempio. Viene quindi avviato un servizio che esegue un'istanza contenitore di questa immagine.



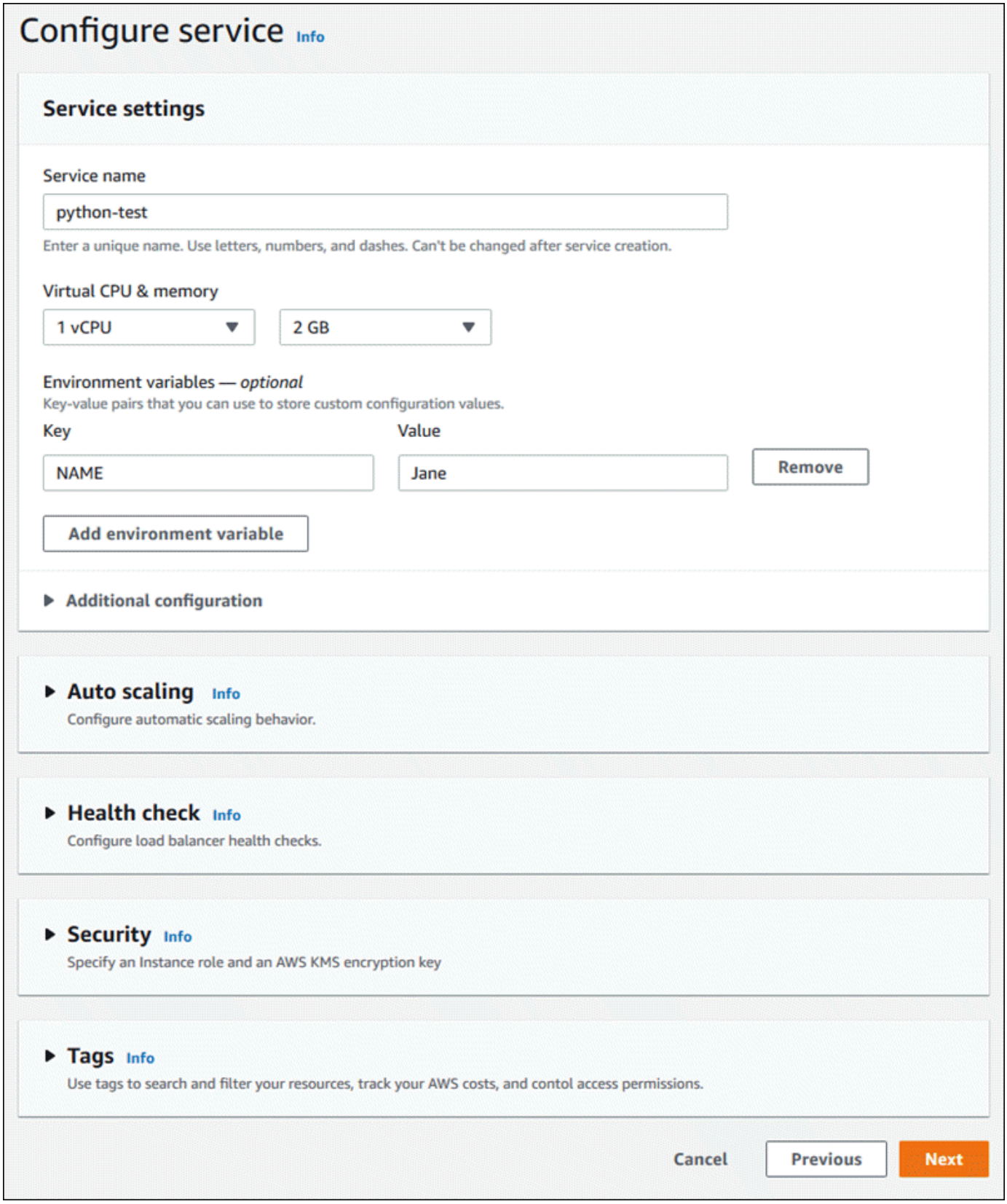
4. Configurare il servizio.

   a. SulConfigurare il servizio, nellaImpostazioni del servizio, immettere un nome di servizio.

   b. UNDEREnvironment variables (Variabili d'ambiente), aggiungere una singola variabile di ambiente. PerChiave, immettere**`NAME`**, e perValoreImmettere un nome qualsiasi (ad esempio, il nome).

> **Note**
>
> L'applicazione di esempio legge il nome impostato in questa variabile di ambiente e visualizza il nome nella relativa pagina Web.

c. Seleziona Successivo.


Configure service Info
Service settings
Service name
python-test
Enter a unique name. Use letters, numbers, and dashes. Can't be changed after service creation.
Virtual CPU & memory
1 vCPU
2 GB
Environment variables — optional
Key-value pairs that you can use to store custom configuration values.
Key
Value
NAME
Jane
Remove
Add environment variable
Additional configuration
Auto scaling Info
Configure automatic scaling behavior.
Health check Info
Configure load balancer health checks.
Security Info
Specify an Instance role and an AWS KMS encryption key
Tags Info
Use tags to search and filter your resources, track your AWS costs, and contol access permissions.
Cancel
Previous
Next

5. SulRivedi e creaverificare tutti i dettagli immessi, quindi scegliereCreazione e distribuzione: .

   Se il servizio viene creato correttamente, nella console viene visualizzato il dashboard del servizio, con unPanoramica del serviziodel nuovo servizio.

6. Verificare che il servizio sia in esecuzione.

   a. Nella pagina del dashboard del servizio, attendere che il servizioStatoèIn esecuzione: .

   b. SelezionaDominio predefinito: è l'URL del sito Web del tuo servizio.

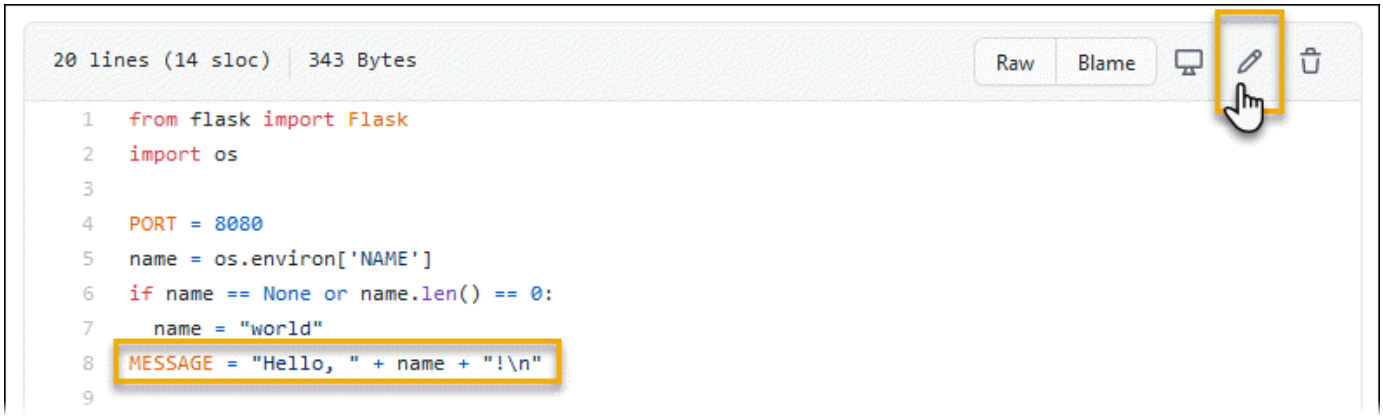
      Viene visualizzata una pagina Web: Ciao,***Nome***!

Fase 2: Modificare il codice di servizio

In questa fase, si apporta una modifica al repository del codice sorgente. La funzionalità CI/CD di App Runner crea e distribuisce automaticamente la modifica al servizio.

Per apportare una modifica al codice di servizio

1. Passa al repository GitHub di esempio.
2. Scegliere il nome file`server.py`per passare a quel file.
3. ScegliereModifica questo file(l'icona a forma di matita).
4. Nell'espressione assegnata alla variabile`MESSAGE`, modificare il testo`Hello`Da a`Good morning`: .



5. Scegliere Commit changes (Applica modifiche).

   Il nuovo commit inizia a distribuire. Nella pagina del dashboard del servizio, il servizioStatoModificazioni alleOperazione in corso: .

6. Attendere il termine della distribuzione. Nella pagina del dashboard del servizio, il servizioStatodovrebbe tornare aIn esecuzione: .

7. Verificare che la distribuzione abbia esito positivo: aggiornare la scheda del browser in cui viene visualizzata la pagina Web del servizio.

   La pagina visualizza ora il messaggio modificato: Buongiorno.***Nome***!

Fase 3: Applicazione di una modifica della configurazione

In questa fase, si apporta una modifica al**NAME**valore della variabile di ambiente, per dimostrare una modifica alla configurazione del servizio.

Per visualizzare i log per il servizio

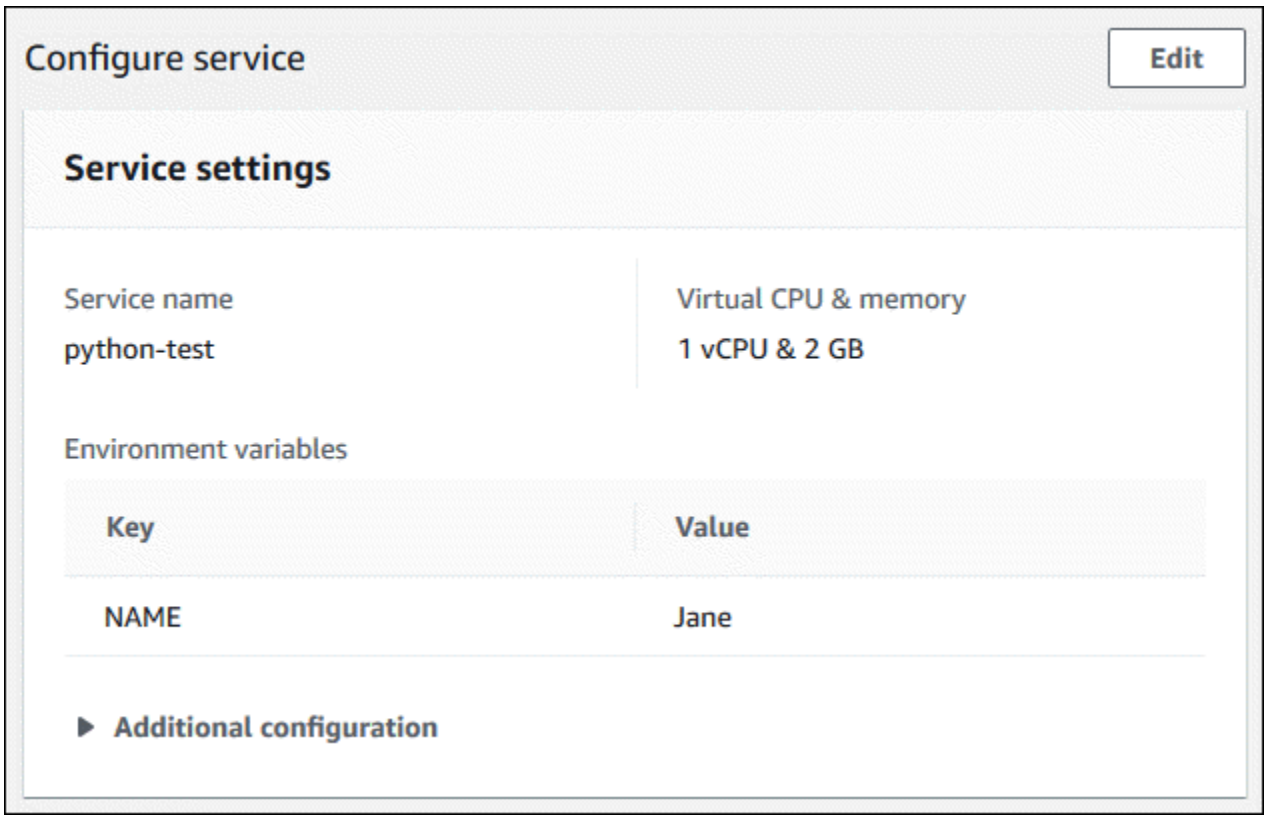
1. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionareServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica del servizio: .

3. Nella pagina di controllo del servizio, scegli la casella di controlloConfigurazioneScheda.

   La console visualizza le impostazioni di configurazione del servizio in diverse sezioni.

4. NellaConfigurare il servizio, scegliereModificare: .



5. Per la variabile di ambiente con la chiave**NAME**Modificare il valore in un nome diverso.
6. Scegli Apply changes (Applica modifiche).

   App Runner avvia il processo di aggiornamento. Nella pagina del dashboard del servizio, il servizioStatoModificazioni alleOperazione in corso: .
7. Attendi la fine dell'aggiornamento. Nella pagina del dashboard del servizio, il servizioStatodovrebbe tornare aIn esecuzione: .
8. Verificare che l'aggiornamento abbia esito positivo: aggiornare la scheda del browser in cui viene visualizzata la pagina Web del servizio.

   La pagina visualizza ora il nome modificato: Buongiorno.***Nuovo nome***!

Fase 4: Visualizzare i registri per il servizio

In questo passaggio, si utilizza la console App Runner per visualizzare i registri per il servizio App Runner. App Runner trasmette i registri su Amazon CloudWatch Logs (CloudWatch Logs) e li visualizza sul dashboard del servizio. Per informazioni sui registri di App Runner, vederethe section called “Registri (CloudWatch Logs)”: .

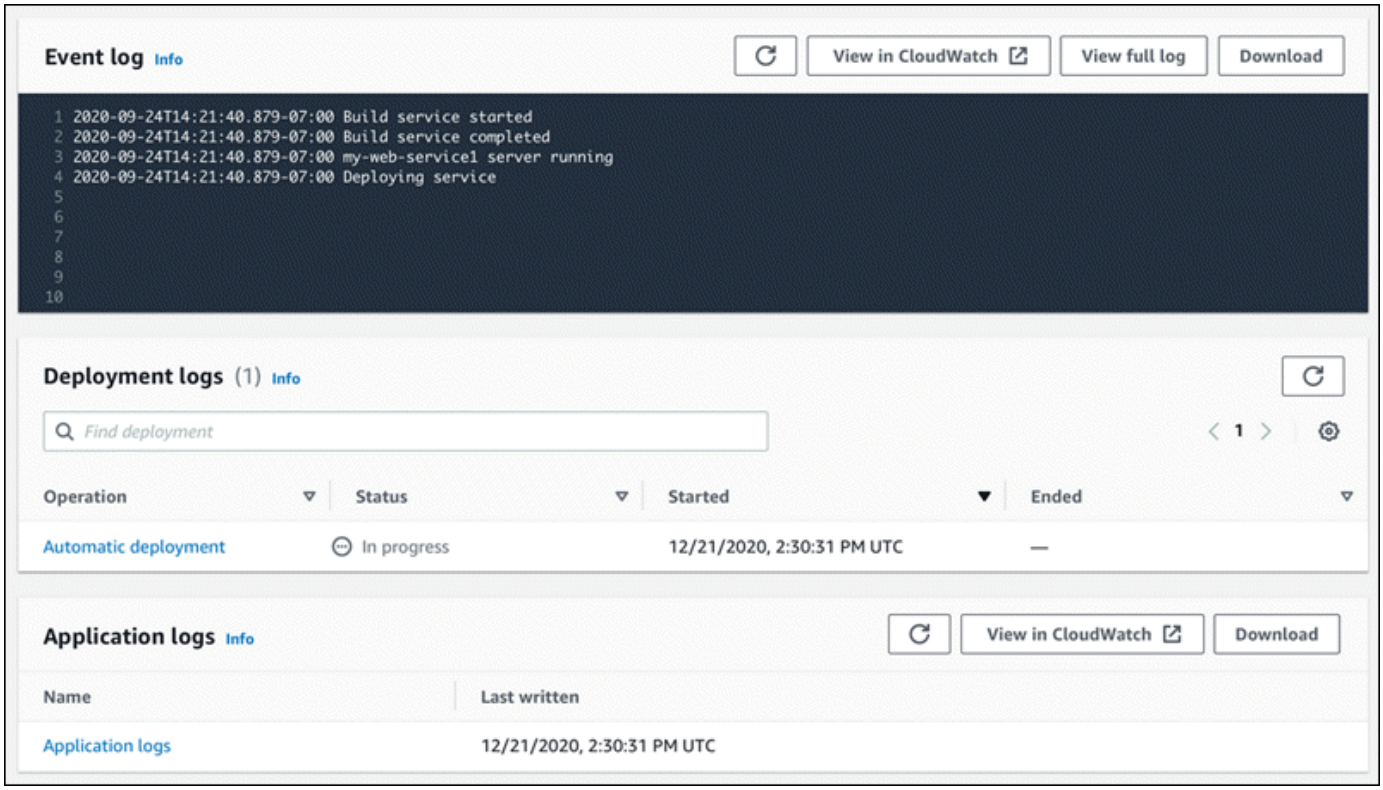
Per visualizzare i log per il servizio

1. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionareServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica del servizio: .
3. Nella pagina di controllo del servizio, scegli la casella di controlloLogScheda.

   La console visualizza alcuni tipi di log in diverse sezioni:

   - Log degli eventi— Attività nel ciclo di vita del servizio App Runner. La console mostra gli eventi più recenti.
   - Registri di distribuzione— Distribuzioni del repository di origine nel servizio App Runner. La console visualizza un flusso di log separato per ogni distribuzione.
   - Log di applicazioni— L'output dell'applicazione Web distribuita nel servizio App Runner. La console combina l'output di tutte le istanze in esecuzione in un unico flusso di log.



4. Per trovare distribuzioni specifiche, eseguire l'ambito nell'elenco dei log di distribuzione immettendo un termine di ricerca. È possibile cercare qualsiasi valore visualizzato nella tabella.

5. Per visualizzare il contenuto di un registro, scegliereVisualizzare il log completo(registro eventi) o il nome del flusso di registro (log di distribuzione e applicazioni).
6. SceglieScaricaper scaricare un registro. Per un flusso di log di distribuzione, selezionare prima un flusso di log.
7. SceglieVisualizzare in CloudWatchper aprire la console CloudWatch e utilizzare tutte le sue funzionalità per esplorare i log del servizio App Runner. Per un flusso di log di distribuzione, selezionare prima un flusso di log.

> **Note**
>
> La console CloudWatch è particolarmente utile se si desidera visualizzare i registri delle applicazioni di istanze specifiche anziché il registro applicazioni combinato.

Fase 5: Elimini

Ora hai imparato a creare un servizio App Runner, visualizzare i registri e apportare alcune modifiche. In questa fase, si elimina il servizio per rimuovere le risorse non più necessarie.

Per eliminare il servizio

1. Nella pagina di controllo del servizio, scegliOperazioni, quindi sceglieEliminazione del servizio: .
2. Nella finestra di dialogo di conferma immettere il testo richiesto, quindi sceglieElimina: .

   Risultato: La console passa alServizi(Certificato creato). Il servizio appena eliminato mostra lo stato diDELETING (ELIMINAZIONE IN CORSO): . Poco tempo dopo scompare dalla lista.

Considerare anche l'eliminazione della connessione GitHub creata come parte di questo tutorial. Per ulteriori informazioni, consulta the section called “Connessioni”.

Fasi successive

Dopo aver distribuito il primo servizio App Runner, scopri di più nei seguenti argomenti:

- Architettura e concetti— L'architettura, i concetti principali eAWSrisorse relative a App Runner.

- Servizio basato sull'immagineeServizio basato su codice in: i due tipi di origine dell'applicazione che App Runner può distribuire.
- Sviluppo per App Runner: cose da sapere quando si sviluppa o si esegue la migrazione del codice dell'applicazione per la distribuzione in App Runner.
- Console App Runner— Gestisci e monitora il tuo servizio utilizzando la console App Runner.
- Gestione dei serviziGestione del ciclo di vita del servizio App Runner.
- Logging e monitoraggio: monitora il tuo servizio App Runner visualizzando metriche, leggendo i registri e monitorando le chiamate di azione del servizio.
- File di configurazione di App Runner— Un modo basato sulla configurazione per specificare le opzioni per il comportamento di compilazione e runtime del servizio App Runner.
- Runner di app— Utilizzare l'API (Application Programming Interface) App Runner per creare, leggere, aggiornare ed eliminare le risorse di App Runner.
- Sicurezza— I diversi modi in cuiAWSe garantisci la sicurezza cloud mentre usi App Runner e altri servizi.

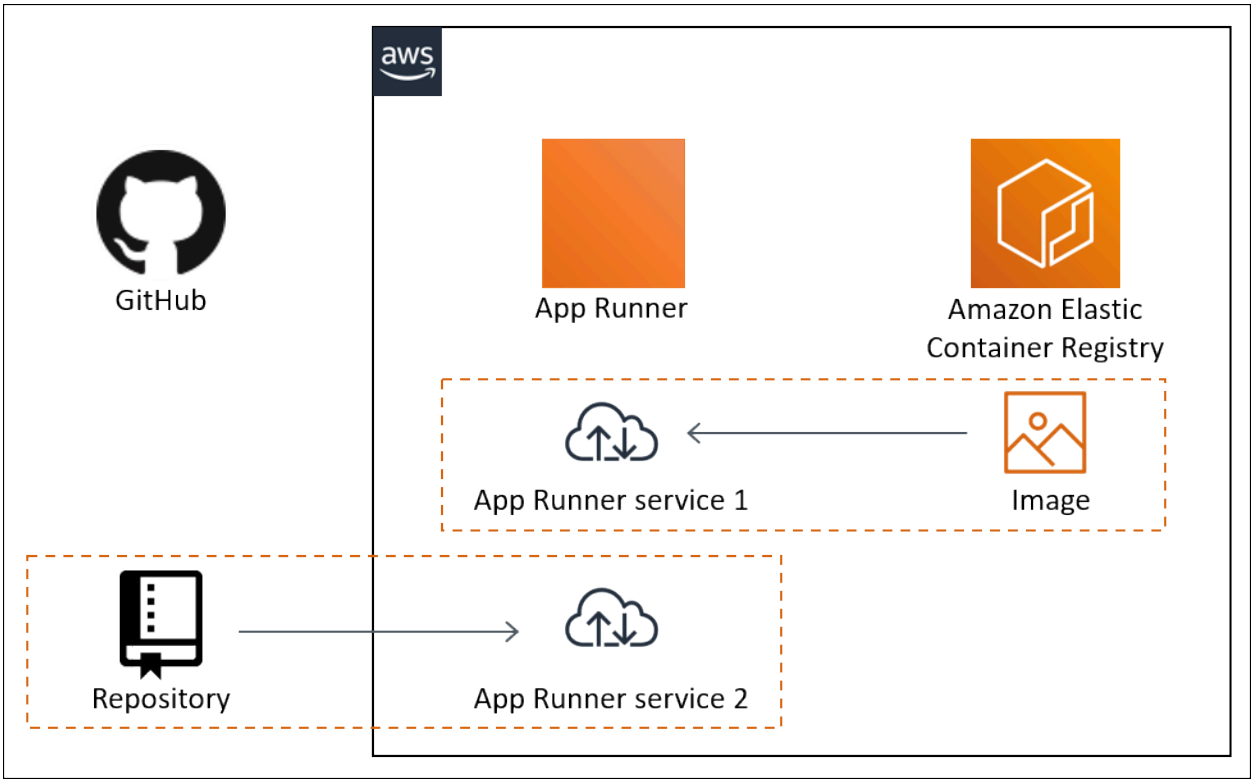
Architettura e concetti di App Runner

AWS App Runnerprende il codice sorgente o l'immagine sorgente da un repository e crea e mantiene un servizio Web in esecuzione per te nelCloud AWS: . In genere, è necessario chiamare solo un'azione App Runner,CreateService, per creare il servizio.

Con un repository di immagini di origine, è possibile fornire un'immagine contenitore pronta all'uso che App Runner può distribuire per eseguire il servizio Web. Con un repository di codice sorgente, è possibile fornire codice e istruzioni per la creazione e l'esecuzione di un servizio Web progettato per uno dei diversi ambienti runtime gestiti da App Runner.

In questo momento, App Runner può recuperare il codice sorgente da unGitHubo recuperare l'immagine sorgente daAmazon Elastic Container Registry (Amazon ECR)Nella paginaAccount AWS: .

Il diagramma seguente mostra una panoramica dell'architettura del servizio App Runner. Nel diagramma, ci sono due servizi di esempio: uno distribuisce il codice sorgente da GitHub e l'altro distribuisce un'immagine sorgente da Amazon ECR.



Concetti di App

Di seguito sono riportati i concetti chiave relativi al servizio Web in esecuzione in App Runner:

- App Runner di servizi— UnAWSutilizzata da App Runner per distribuire e gestire l'applicazione in base al repository del codice sorgente o all'immagine del contenitore. Un servizio App Runner è una versione in esecuzione dell'applicazione. Per ulteriori informazioni sulla creazione di un servizio, consultathe section called “Creazione”: .
- Tipo di origine— Il tipo di repository di origine fornito per la distribuzione del servizio App Runner:Codice sorgenteoimmagine sorgente: .
- Fornitori di repository— Il servizio repository che contiene l'origine dell'applicazione (ad esempio,GitHuboAmazon ECR).
- App Runner di connessione— UnAWSche consente ad App Runner di accedere a un account del provider del repository (ad esempio, un account o un'organizzazione GitHub). Per ulteriori informazioni sulle connessioni, consulta the section called “Connessioni”.
- Runtime— Un'immagine di base per la distribuzione di un repository di codice sorgente. App Runner offre una varietà diRuntime gestitiper diversi ambienti di programmazione. Per ulteriori informazioni, consulta Servizio basato su codice in.
- Distribuzione— Un'azione che applica una versione del repository di origine (codice o immagine) a un servizio App Runner. La prima distribuzione al servizio si verifica come parte della creazione del servizio. Le distribuzioni successive possono avvenire in uno dei due modi seguenti:
  - Distribuzioni automatiche— Una capacità CI/CD. È possibile configurare un servizio App Runner per creare automaticamente (per il codice sorgente) e distribuire ogni versione dell'applicazione così come appare nel repository. Può trattarsi di un nuovo commit in un repository di codice sorgente o di una nuova versione di immagine in un repository di immagini di origine.
  - Distribuzioni manuali— Una distribuzione al servizio App Runner che si avvia esplicitamente.
- Dominio personalizzato— Un dominio associato al servizio App Runner. Gli utenti dell'applicazione Web possono utilizzare questo dominio per accedere al servizio Web anziché il sottodominio predefinito App Runner. Per ulteriori informazioni, consulta the section called “Nomi di dominio personalizzati”.
- Manutenzione: un'attività che App Runner esegue occasionalmente sull'infrastruttura che esegue il servizio App Runner. Quando la manutenzione è in corso, lo stato del servizio cambia temporaneamente in`OPERATION_IN_PROGRESS`(Operazioni in corsoNella console) per alcuni minuti. Le azioni sul servizio (ad esempio, distribuzione, aggiornamento della configurazione,

pausa/riprendi o eliminazione) vengono bloccate durante questo periodo. Riprovare l'azione qualche minuto dopo, quando lo stato del servizio torna a`RUNNING`: .

> **Note**
>
> Se l'azione fallisce, non significa che il servizio App Runner sia inattivo. L'applicazione è attiva e continua a gestire le richieste. È improbabile che il servizio subisca tempi di inattività.

In particolare, App Runner esegue la migrazione del servizio se rileva problemi nell'hardware sottostante che ospita il servizio. Per evitare tempi di inattività del servizio, App Runner distribuisce il servizio in un nuovo set di istanze e sposta il traffico verso di esse (una distribuzione blu-verde). Occasionalmente potresti vedere un leggero aumento temporaneo delle accuse.

Risorse App Runner

Quando usi App Runner, crei e gestisci alcuni tipi di risorse nel tuoAccount AWS: . Queste risorse vengono utilizzate per accedere al codice e gestire i servizi.

La tabella seguente fornisce una panoramica delle risorse:

Nome risorsa	Descrizione
Service	Rappresenta una versione in esecuzione dell'applicazione. Gran parte del resto di questa guida descrive i tipi di servizio, la gestione, la configurazione e il monitoraggio. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:service/` *`service-name`* `[/`*`service-id`* `]`
Connection	Fornisce ai servizi App Runner l'accesso ai repository privati archiviati con provider di terze parti. Esiste come risorsa separata per la condivisi one tra più servizi. Per ulteriori informazioni sulle connessioni, consulta the section called “Connessioni”. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:connecti on/` *`connection-name`* `[/`*`connection-id`* `]`

Nome risorsa	Descrizione
AutoScalingConfiguration	Fornisce ai servizi App Runner impostazioni che controllano il ridimensionamento automatico dell'applicazione. Esiste come risorsa separata per la condivisione tra più servizi. Per ulteriori informazioni sulla scalabilità automatica, consultathe section called “Auto Scaling”: . ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`*` :autoscalingconfiguration/` *`config-name`* `[/`*`config-revision`* `[/`*`config-id`* `]]`

Quote di risorse App Runner

AWSimpone alcune quote (note anche come limiti) sul tuo account perAWSUtilizzo delle risorse in ogniRegione AWS: . La tabella seguente elenca le quote correlate alle risorse App Runner. Le quote sono anche elencate inAWS App RunnerEndpoint e quotenellaAWSRiferimenti generali: .

Quota di risorse	Descrizione	Valore predefinito	Modificabile?
Services	Numero massimo di servizi che puoi creare nel tuo account per ogniRegione AWS: .	10	✓ Sì
Connections	Numero massimo di connessioni che puoi creare nel tuo account per ogniRegione AWS: . È possibile condividere una singola connessione tra più servizi.	10	✓ Sì
Auto scaling configurations—nomi	Il numero massimo di nomi univoci che è possibile avere nelle configurazioni di ridimensionamento automatico create nell'account per ogniRegione AWS: . È possibile utilizzare una configurazione di ridimensionamento automatico in più servizi.	10	✓ Sì
Auto scaling configurations—	Numero massimo di revisioni della configurazione in scala automatica che puoi creare nell'account per ogniRegione AWSPer ogni nome univoco. È	10	✗ No

Quota di risorse	Descrizione	Valore predefinito	Modificabile?
revisioni per ogni nome	possibile utilizzare una revisione della configurazione di ridimensionamento automatico in più servizi.		

La maggior parte delle quote è regolabile e puoi richiedere un aumento delle quote. Per ulteriori informazioni, consultaRichiesta di aumento delle quoteNella Guida per l'utente Service Quotas.

Servizio App Runner basato su un'immagine sorgente

È possibile utilizzareAWS App Runnerper creare e gestire servizi basati su due tipi fondamentalmente diversi di origine del servizio:Codice sorgenteeimmagine sorgente: . Indipendentemente dal tipo di origine, App Runner si occupa di avviare, eseguire, ridimensionare e bilanciare il carico del servizio. È possibile utilizzare la funzionalità CI/CD di App Runner per tenere traccia delle modifiche all'immagine o al codice sorgente. Quando App Runner rileva una modifica, crea automaticamente (per il codice sorgente) e distribuisce la nuova versione nel servizio App Runner.

In questo capitolo vengono descritti i servizi basati su un'immagine sorgente. Per informazioni sui servizi basati sul codice sorgente, vedereServizio basato su codice in: .

Aimmagine sorgenteè un'immagine contenitore pubblica o privata memorizzata in un repository di immagini. Si punta App Runner a un'immagine e avvia un servizio che esegue un contenitore basato su questa immagine. Non è necessaria alcuna fase di costruzione. Piuttosto, si fornisce un'immagine pronta per la distribuzione.

Provider repository immagine

App Runner supporta i seguenti provider di repository immagine:

- Amazon Elastic Container Registry (Amazon ECR)— Memorizza le immagini private nel tuoAccount AWS: .
- Amazon Elastic Container Registry Public (Amazon ECR Public)— Memorizza immagini leggibili pubblicamente.

Distribuzione da Amazon ECR

Amazon ECRmemorizza le immagini nei repository. Ci sono repository privati e pubblici. Per distribuire l'immagine a un servizio App Runner da un repository privato, App Runner ha bisogno dell'autorizzazione per leggere l'immagine da Amazon ECR. Per concedere tale autorizzazione ad App Runner, devi fornire a App Runner unruolo di accesso: . Questo è unAWS Identity and Access ManagementRuolo (IAM) dotato dell'autorizzazione dell'operazione ECR necessaria. Quando si utilizza la console App Runner per creare il servizio, è possibile scegliere un ruolo esistente nel proprio account. In alternativa, è possibile utilizzare la console IAM per creare un nuovo ruolo

personalizzato oppure scegliere per la console App Runner per creare un ruolo in base ai criteri gestiti.

Quando si utilizza l'API App Runner o ilAWS CLI, è possibile completare un processo in due fasi. Innanzitutto, utilizzare la console IAM per creare un ruolo di accesso. È possibile utilizzare un criterio gestito fornito da App Runner o immettere autorizzazioni personalizzate. Quindi, si fornisce il ruolo di accesso durante la creazione del servizio utilizzando ilCreateServiceOperazione API.

Per informazioni sulla creazione del servizio App Runner, vederethe section called “Creazione”: .

Distribuzione da Amazon ECR Public

Amazon ECRmemorizza immagini leggibili pubblicamente. Queste sono le principali differenze tra Amazon ECR e Amazon ECR Public di cui dovresti essere a conoscenza nel contesto dei servizi App Runner:

- Le immagini pubbliche di Amazon ECR sono pubblicamente leggibili. Non è necessario fornire un ruolo di accesso quando crei un servizio basato su un'immagine pubblica Amazon ECR.
- App Runner non supporta la distribuzione automatica per le immagini pubbliche Amazon ECR.

Servizio App Runner basato sul codice sorgente

È possibile utilizzareAWS App Runnerper creare e gestire servizi basati su due tipi fondamentalmente diversi di origine del servizio:Codice sorgenteeimmagine sorgente: . Indipendentemente dal tipo di origine, App Runner si occupa di avviare, eseguire, ridimensionare e bilanciare il carico del servizio. È possibile utilizzare la funzionalità CI/CD di App Runner per tenere traccia delle modifiche all'immagine o al codice sorgente. Quando App Runner rileva una modifica, crea automaticamente (per il codice sorgente) e distribuisce la nuova versione nel servizio App Runner.

Questo capitolo discute i servizi basati sul codice sorgente. Per informazioni sui servizi basati su un'immagine di origine, vedereServizio basato sull'immagine: .

Il codice sorgente è il codice dell'applicazione che App Runner crea e distribuisce per te. Si punta App Runner a un repository di codice sorgente e si sceglie unruntime: . App Runner crea un'immagine basata sull'immagine di base del runtime e sul codice dell'applicazione. Viene quindi avviato un servizio che esegue un contenitore basato su questa immagine.

App Runner fornisce una comoda lingua specificaRuntime gestiti: . Ognuno di questi runtime crea un'immagine contenitore dal codice sorgente e aggiunge dipendenze di runtime del linguaggio all'immagine. Non è necessario fornire istruzioni di configurazione del container e compilazione come un Dockerfile.

I sottoargomenti di questo capitolo discutono i variRuntimeche App Runner supporta: il runtime Dockerfile generico e ilRuntime gestitiper diversi ambienti di programmazione.

Argomenti

- Fornitori del repository del codice sorgente
- Runtime gestiti da App Runner
- Utilizzo del runtime gestito da Python
- Utilizzo del runtime gestito Node.js

Fornitori del repository del codice sorgente

App Runner distribuisce il codice sorgente leggerlo da un repository di codice sorgente. App Runner supporta un provider di repository del codice sorgente: GitHub.

Distribuzione da GitHub

Per distribuire il codice sorgente in un servizio App Runner da unGitHub, App Runner stabilisce una connessione a GitHub. Se il tuo repository è privato (ovvero non è accessibile pubblicamente su GitHub), devi fornire a App Runner i dettagli della connessione. Quando si utilizza la console App Runner perCrea un servizio.I dettagli della connessione sono forniti come parte della procedura di creazione del servizio.

Quando si utilizza l'API App Runner oAWS CLI, una connessione è una risorsa separata. Innanzitutto, creare la connessione utilizzando il comandoCreateConnectionOperazione API. Quindi, si fornisce l'ARN della connessione durante la creazione del servizio utilizzando ilCreateServiceOperazione API.

Per ulteriori informazioni sulla creazione del servizio App Runner, consultathe section called “Creazione”: . Per ulteriori informazioni sulle connessioni di App Runner, consultathe section called “Connessioni”: .

Runtime gestiti da App Runner

App Runner fornisce runtime gestiti per vari ambienti di programmazione. Ogni runtime gestito semplifica la creazione ed esecuzione di contenitori basati su un particolare linguaggio di programmazione o ambiente di runtime. Quando si utilizza un runtime gestito, App Runner viene avviato con un'immagine runtime gestita. Questa immagine è basata sulImmagine Docker Amazon Linuxe contiene un pacchetto di runtime del linguaggio, nonché alcuni strumenti e pacchetti di dipendenza popolari. App Runner utilizza questa immagine runtime gestita come immagine di base e aggiunge il codice dell'applicazione per creare un'immagine Docker. Quindi distribuisce questa immagine per eseguire il servizio Web in un contenitore.

Si specifica un runtime per il servizio App Runner quandoCrea un servizio.utilizzando la console App Runner o ilCreateServiceAPI. È inoltre possibile specificare un runtime come parte del codice sorgente. Utilizzo dell'`runtime`Parola chiave in unFile di configurazione di App Runnerche includi nel tuo repository di codice. La convenzione di denominazione di un runtime gestito è*`<language-name> <major-version>`*: .

App Runner aggiorna il runtime del servizio alla versione più recente per ogni distribuzione o aggiornamento del servizio. Se l'applicazione richiede una versione specifica di un runtime gestito, è possibile specificarla utilizzando l'opzione`runtime-version`Parola chiave nellaFile di configurazione di App Runner: . Specificare una versione secondaria come*`<major>`*: .*`<minor>`*per

bloccare le versioni principali e secondarie (App Runner aggiorna solo le versioni delle patch). Specificare un particolare livello di patch come*<major>*: .*<minor>*: .*<patch>*per bloccare il servizio su una versione runtime specifica (App Runner non aggiorna mai il runtime).

Utilizzo del runtime gestito da Python

AWS App Runnerfornisce un runtime gestito da Python. Il runtime consente di creare ed eseguire facilmente contenitori con applicazioni Web basate su Python. Quando si utilizza il runtime Python, App Runner viene avviato con un'immagine runtime Python gestita. Questa immagine è basata sullImmagine della Docker Amazon Linuxe contiene il pacchetto runtime Python e alcuni strumenti e pacchetti di dipendenza popolari. App Runner utilizza questa immagine runtime gestita come immagine di base e aggiunge il codice dell'applicazione per creare un'immagine Docker. Quindi distribuisce questa immagine per eseguire il servizio Web in un contenitore.

Si specifica un runtime per il servizio App Runner quandoCreare un servizioutilizzando la console App Runner o ilCreateServiceAPI. È inoltre possibile specificare un runtime come parte del codice sorgente. Utilizzo dell'`runtime`Parola chiave in unFile di configurazione di App Runnerche includi nel tuo repository di codice. La convenzione di denominazione di un runtime gestito è*<language-name> <major-version>*: .

Per nomi di runtime Python validi, vederethe section called “Informazioni sulla versione di”: .

App Runner aggiorna il runtime del servizio alla versione più recente per ogni distribuzione o aggiornamento del servizio. Se l'applicazione richiede una versione specifica di un runtime gestito, è possibile specificarla utilizzando l'opzione`runtime-version`Parola chiave nellaFile di configurazione di App Runner: . Specificare una versione secondaria come*<major>*: .*<minor>*per bloccare le versioni principali e secondarie (App Runner aggiorna solo le versioni delle patch). Specificare un particolare livello di patch come*<major>*: .*<minor>*: .*<patch>*per bloccare il servizio su una versione runtime specifica (App Runner non aggiorna mai il runtime).

Argomenti

- Configurazione di runtime Python
- Esempio di runtime Python
- Informazioni sulla versione di Python

Configurazione di runtime Python

Quando si sceglie un runtime gestito, è inoltre necessario configurare, come minimo, i comandi di compilazione ed esecuzione. Li configuri mentrecreating (creazione in corso)oaggiornamentoil servizio App Runner. Vi sono vari modi per farlo:

- Utilizzo della console App Runner— Specifica i comandi nella finestraConfigurare builddel processo di creazione o della scheda di configurazione.
- Utilizzo dell'API App Runner— ChiamaCreateServiceoUpdateService: . Specificare i comandi utilizzando il comando`BuildCommand`e`StartCommand`Membri delCodeConfigurationValuesTipo di dati.
- Utilizzo di unaFile di configurazione: specifica uno o più comandi di compilazione in un massimo di tre fasi di compilazione e un singolo comando di esecuzione che serve per avviare l'applicazione. Sono disponibili impostazioni di configurazione opzionali aggiuntive.

Fornire un file di configurazione è facoltativo. Quando si crea un servizio App Runner utilizzando la console o l'API, è necessario specificare se App Runner ottiene le impostazioni di configurazione direttamente durante la creazione o da un file di configurazione.

Esempio di runtime Python

Gli esempi seguenti mostrano i file di configurazione di App Runner per la creazione e l'esecuzione di un servizio Python. L'ultimo esempio è il codice sorgente di un'applicazione Python completa che è possibile distribuire a un servizio runtime Python.

File di configurazione di Python

Questo esempio mostra un file di configurazione minimo che è possibile utilizzare con il runtime gestito da Python. Per le ipotesi che App Runner fa con un file di configurazione minimo, vederethe section called “Esempio di file di configurazione”: .

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    build:
```

```
      - pip install pipenv
      - pipenv install
run:
  command: python app.py
```

File di configurazione di Python

Questo esempio mostra l'uso di tutte le chiavi di configurazione con il runtime gestito da Python.

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    pre-build:
      - wget -c https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/test-lib.tar.gz -O - | tar
 -xz
    build:
      - pip install pipenv
      - pipenv install
    post-build:
      - python manage.py test
  env:
    - name: DJANGO_SETTINGS_MODULE
      value: "django_apprunner.settings"
    - name: MY_VAR_EXAMPLE
      value: "example"
run:
  runtime-version: 3.7.7
  command: pipenv run gunicorn django_apprunner.wsgi --log-file -
  network:
    port: 8000
    env: MY_APP_PORT
  env:
    - name: MY_VAR_EXAMPLE
      value: "example"
```

Origine dell'applicazione Python end-to-end

Questo esempio mostra il codice sorgente di un'applicazione Python completa che è possibile distribuire in un servizio runtime Python.

Example requirements.txt

```
Click==7.0
Flask==1.0.2
itsdangerous==1.1.0
Jinja2==2.10
MarkupSafe==1.1.1
Werkzeug==0.15.5
```

Example server.py

```
from flask import Flask
import os

PORT = 8080
name = os.environ['NAME']
if name == None or len(name) == 0:
  name = "world"
MESSAGE = "Hello, " + name + "!"
print("Message: '" + MESSAGE + "'")

app = Flask(__name__)


@app.route("/")
def root():
  print("Handling web request. Returning message.")
  result = MESSAGE.encode("utf-8")
  return result


if __name__ == "__main__":
  app.run(debug=True, host="0.0.0.0", port=PORT)
```

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    build:
      - pip install -r requirements.txt
```

```
run:
  command: python server.py
```

Informazioni sulla versione di Python

In questo argomento sono elencati i dettagli completi per le versioni runtime Python supportate da App Runner.

Python 3

Dettaglio	Descrizione
Nome runtime	Python3
Versioni secondarie	3.7, 3.8
Pacchetti inclusi	python, pip, setuptools, ruota, virtualenv

Utilizzo del runtime gestito Node.js

AWS App Runnerfornisce un runtime gestito Node.js. Il runtime consente di creare ed eseguire facilmente contenitori con applicazioni Web basate su Nodejs-based. Quando si utilizza il runtime Node.js, App Runner viene avviato con un'immagine di runtime Node.js gestita. Questa immagine è basata sullImmagine Docker Amazon Linuxe contiene il pacchetto runtime Node.js e alcuni strumenti. App Runner utilizza questa immagine runtime gestita come immagine di base e aggiunge il codice dell'applicazione per creare un'immagine Docker. Quindi distribuisce questa immagine per eseguire il servizio Web in un contenitore.

Si specifica un runtime per il servizio App Runner quandoCreare un servizioutilizzando la console App Runner o ilCreateServiceAPI. È inoltre possibile specificare un runtime come parte del codice sorgente. Utilizzo dell'`runtime`Parola chiave in unFile di configurazione di App Runnerche includi nel tuo repository di codice. La convenzione di denominazione di un runtime gestito è*`<language-name> <major-version>`*: .

Per nomi di runtime Node.js validi, vederethe section called “Informazioni sulla versione di”: .

App Runner aggiorna il runtime del servizio alla versione più recente per ogni distribuzione o aggiornamento del servizio. Se l'applicazione richiede una versione specifica di un runtime gestito, è possibile specificarla utilizzando l'opzione`runtime-version`Parole chiave nellaFile di

configurazione di App Runner: . Specificare una versione secondaria come*<major>*: .*<minor>*per bloccare le versioni principali e secondarie (App Runner aggiorna solo le versioni delle patch). Specificare un particolare livello di patch come*<major>*: .*<minor>*: .*<patch>*per bloccare il servizio su una versione runtime specifica (App Runner non aggiorna mai il runtime).

Argomenti

- Configurazione runtime per Node.js
- Esempi runtime per Node.js
- Informazioni sulla versione runtime per Node.js

Configurazione runtime per Node.js

Quando si sceglie un runtime gestito, è inoltre necessario configurare, come minimo, i comandi di compilazione ed esecuzione. Li configuri mentrecreating (creazione in corso)oaggiornamentoil servizio App Runner. Vi sono vari modi per farlo:

- Utilizzo della console App Runner— Specifica i comandi nella finestraConfigura builddel processo di creazione o della scheda di configurazione.
- Utilizzo dell'API App Runner— ChiamaCreateServiceoUpdateService: . Specificare i comandi utilizzando il comando`BuildCommand`e`StartCommand`Membri delCodeConfigurationValuesTipo di dati.
- Utilizzo di unaFile di configurazione: specifica uno o più comandi di compilazione in un massimo di tre fasi di compilazione e un singolo comando di esecuzione che serve per avviare l'applicazione. Sono disponibili impostazioni di configurazione opzionali aggiuntive.

Fornire un file di configurazione è facoltativo. Quando si crea un servizio App Runner utilizzando la console o l'API, è necessario specificare se App Runner ottiene le impostazioni di configurazione direttamente durante la creazione o da un file di configurazione.

Con il runtime Node.js in particolare, è anche possibile configurare la build e il runtime utilizzando un file JSON denominato`package.json`nella directory principale del repository di origine. Utilizzando questo file, è possibile configurare la versione del motore Node.js, i pacchetti delle dipendenze e vari comandi (applicazioni da riga di comando). I gestori di pacchetti come npm o yarn interpretano questo file come input per i loro comandi.

Ad esempio:

- npm installinstalla i pacchetti definiti daldependenciesedevDependenciesNodo inpackage.json: .
- npm startonpm run startesegue il comando definito dal comandoscripts/startNodo inpackage.json: .

Di seguito è riportato un esempio del file package.json.

package.json

```
{
  "name": "node-js-getting-started",
  "version": "0.3.0",
  "description": "A sample Node.js app using Express 4",
  "engines": {
    "node": "12.18.4"
  },
  "scripts": {
    "start": "node index.js",
    "test": "node test.js"
  },
  "dependencies": {
    "cool-ascii-faces": "^1.3.4",
    "ejs": "^2.5.6",
    "express": "^4.15.2"
  },
  "devDependencies": {
    "got": "^11.3.0",
    "tape": "^4.7.0"
  }
}
```

Per ulteriori informazioni supackage.json, consultaGuida package.jsonsul sito Web di Node.js.

Tips

- Se le ricette dipackage.jsondefinisce un filestart, puoi utilizzarlo comerunnel file di configurazione di App Runner, come illustrato nell'esempio seguente.

  Example

  package.json

```
{
  "scripts": {
    "start": "node index.js"
  }
}
```

apprunner.yaml

```
run:
  command: npm start
```

- Quando si eseguenpm installnel tuo ambiente di sviluppo, npm crea il file`package-lock.json`: . Questo file contiene un'istantanea delle versioni del pacchetto npm appena installate. Successivamente, quando npm installa le dipendenze, utilizza queste versioni esatte. Allo stesso modo, filato crea`yarn.json`: . Eseguire il commit di questi file nel repository del codice sorgente per assicurarsi che l'applicazione sia installata con le versioni delle dipendenze sviluppate e testate con.
- È inoltre possibile utilizzare un file di configurazione di App Runner per configurare la versione Node.js e il comando start. Quando si esegue questa operazione, queste definizioni sovrascrivono quelle in`package.json`: . Un conflitto tranodeversione in`package.json`e la`runtime-version`nel file di configurazione di App Runner causa il fallimento della fase di compilazione di App Runner.

Esempi runtime per Node.js

Gli esempi seguenti mostrano i file di configurazione di App Runner per la creazione e l'esecuzione di un servizio Node.js.

File di configurazione di Node.js

In questo esempio viene illustrato un file di configurazione minimo che è possibile utilizzare con il runtime gestito Node.js. Per le ipotesi che App Runner fa con un file di configurazione minimo, vederethe section called “Esempio di file di configurazione”: .

Example apprunner.yaml

```
version: 1.0
```

```
runtime: nodejs12
build:
  commands:
    build:
      - npm install --production
run:
  command: node app.js
```

File di configurazione di Node.js

In questo esempio viene illustrato l'utilizzo di tutte le chiavi di configurazione con il runtime gestito Node.js.

Example apprunner.yaml

```
version: 1.0
runtime: nodejs12
build:
  commands:
    pre-build:
      - npm install --only=dev
      - node test.js
    build:
      - npm install --production
    post-build:
      - node node_modules/ejs/postinstall.js
  env:
    - name: MY_VAR_EXAMPLE
      value: "example"
run:
  runtime-version: 12.18.4
  command: node app.js
  network:
    port: 8000
    env: APP_PORT
  env:
    - name: MY_VAR_EXAMPLE
      value: "example"
```

App Node.js con Grunt

In questo esempio viene illustrato come configurare un'applicazione Node.js sviluppata con Grunt. gruntè un task runner JavaScript da riga di comando. Esegue attività ripetitive e gestisce

l'automazione dei processi per ridurre gli errori umani. I plugin Grunt e Grunt sono installati e gestiti usando npm. È possibile configurare Grunt includendo il`Gruntfile.js`File nella directory principale del repository di origine.

Example package.json

```
{
  "scripts": {
    "build": "grunt uglify",
    "start": "node app.js"
  },
  "devDependencies": {
    "grunt": "~0.4.5",
    "grunt-contrib-jshint": "~0.10.0",
    "grunt-contrib-nodeunit": "~0.4.1",
    "grunt-contrib-uglify": "~0.5.0"
  },
  "dependencies": {
    "express": "^4.15.2"
  },
}
```

Example Gruntfile.js

```
module.exports = function(grunt) {

  // Project configuration.
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    uglify: {
      options: {
        banner: '/*! <%= pkg.name %> <%= grunt.template.today("yyyy-mm-dd") %> */\n'
      },
      build: {
        src: 'src/<%= pkg.name %>.js',
        dest: 'build/<%= pkg.name %>.min.js'
      }
    }
  });

  // Load the plugin that provides the "uglify" task.
  grunt.loadNpmTasks('grunt-contrib-uglify');
```

```
  // Default task(s).
  grunt.registerTask('default', ['uglify']);

};
```

Example apprunner.yaml

```
version: 1.0
runtime: nodejs12
build:
  commands:
    pre-build:
      - npm install grunt grunt-cli
      - npm install --only=dev
      - npm run build
    build:
      - npm install --production
run:
  runtime-version: 12.18.4
  command: node app.js
  network:
    port: 8000
    env: APP_PORT
```

Informazioni sulla versione runtime per Node.js

In questo argomento sono elencati i dettagli completi per le versioni runtime Node.js supportate da App Runner.

Node.js 12

Dettaglio	Descrizione
Nome runtime	nodejs12
Versioni secondarie	(più recente)
Pacchetti inclusi	nodejs (compreso npm), filati

Sviluppo del codice dell'applicazione per App Runner

In questo capitolo vengono illustrate le informazioni di runtime e le linee guida di sviluppo da considerare quando si sviluppa o si esegue la migrazione del codice dell'applicazione per la distribuzione inAWS App Runner: .

Runtime delle informazioni

Sia che tu fornisca un'immagine contenitore o che App Runner ne crei una per te, App Runner esegue il codice dell'applicazione in un'istanza del contenitore. Di seguito sono riportati alcuni aspetti chiave dell'ambiente runtime dell'istanza contenitore.

- Framework— App Runner supporta qualsiasi immagine che implementa un'applicazione web. È indipendente dal linguaggio di programmazione scelto e dal server applicazioni Web o framework che si utilizza, se si utilizza uno qualsiasi. Per comodità dell'utente, forniamo runtime gestiti specifici per la lingua per semplificare il processo di creazione dell'applicazione e la creazione di immagini astratte.
- Richieste Web— L'istanza del contenitore deve ascoltare le richieste HTTP, sulla porta 8080 per impostazione predefinita. Per ulteriori informazioni sulla configurazione del servizio, vedithe section called “Configurazione”: . Non è necessario implementare la gestione del traffico sicuro HTTPS. App Runner richiede il traffico HTTPS in ingresso e termina HTTPS prima di passare le richieste all'istanza del contenitore.
- App stateless— App Runner non garantisce la persistenza dello stato oltre la durata dell'elaborazione di una singola richiesta web in entrata.
- Storage— App Runner implementa il file system nell'istanza del contenitore comeStoccaggio Effimere: . I file sono transitori. Ad esempio, non persistono quando sospendi e riprendi il servizio App Runner. Più in generale, i file non sono garantiti per persistere oltre l'elaborazione di una singola richiesta, come parte della natura stateless dell'applicazione. I file archiviati, tuttavia, occupano parte dell'allocazione di archiviazione del servizio App Runner per tutta la durata della loro vita.

> **Note**
>
> Anche se i file di archiviazione effimeri potrebbero non persistere tra le richieste,A voltePersistere. Questo può essere utile in alcune situazioni. Ad esempio, quando si gestisce una richiesta, è possibile memorizzare nella cache i file scaricati dall'applicazione

se le richieste future potrebbero averne bisogno. Ciò potrebbe accelerare la gestione delle richieste future, ma non può garantire i guadagni di velocità. Il tuo codice non dovrebbe presumere che un file che è stato scaricato in una richiesta precedente esista ancora. Per garantire la memorizzazione nella cache utilizzando un Data Store in memoria con velocità effettiva elevata e bassa latenza, utilizzare un servizio comeAmazon ElastiCache: .

- Environment variables (Variabili d'ambiente)— Per impostazione predefinita, App Runner rende il`PORT`disponibile nell'istanza del contenitore. È possibile configurare il valore della variabile con le informazioni sulla porta e aggiungere variabili e valori di ambiente personalizzati. Per ulteriori informazioni sulla configurazione del servizio, vedithe section called “Configurazione”: .
- Ruolo dell'istanza— Se il codice dell'applicazione effettua chiamate a qualsiasiAWS, utilizzando le API del servizio o una delleAWSPer creare un ruolo dell'istanza tramiteAWS Identity and Access Management(IAM). Quindi, collegalo al tuo servizio App Runner quando lo crei. Tutti inclusiAWSche il codice richiede nel ruolo di istanza. Per ulteriori informazioni, consulta the section called “Ruolo dell'istanza”.

Linee guida per lo

Considerare queste linee guida quando si sviluppa codice per un'applicazione Web App Runner.

- Progettazione del codice stateless— Progetta l'applicazione Web che distribuisci nel servizio App Runner in modo che sia priva di stato. Il codice dovrebbe presumere che nessuno stato persista oltre la durata dell'elaborazione di una singola richiesta Web in arrivo.
- Eliminazione dei file temporanei: quando si creano file, questi vengono archiviati in un file system e occupano parte dell'allocazione di archiviazione del servizio. Per evitare errori di archiviazione insufficiente, non conservare i file temporanei per periodi prolungati. Bilanciare le dimensioni dello storage con la velocità di gestione delle richieste quando si prendono decisioni di memorizzazione nella cache

Utilizzo della console App Runner

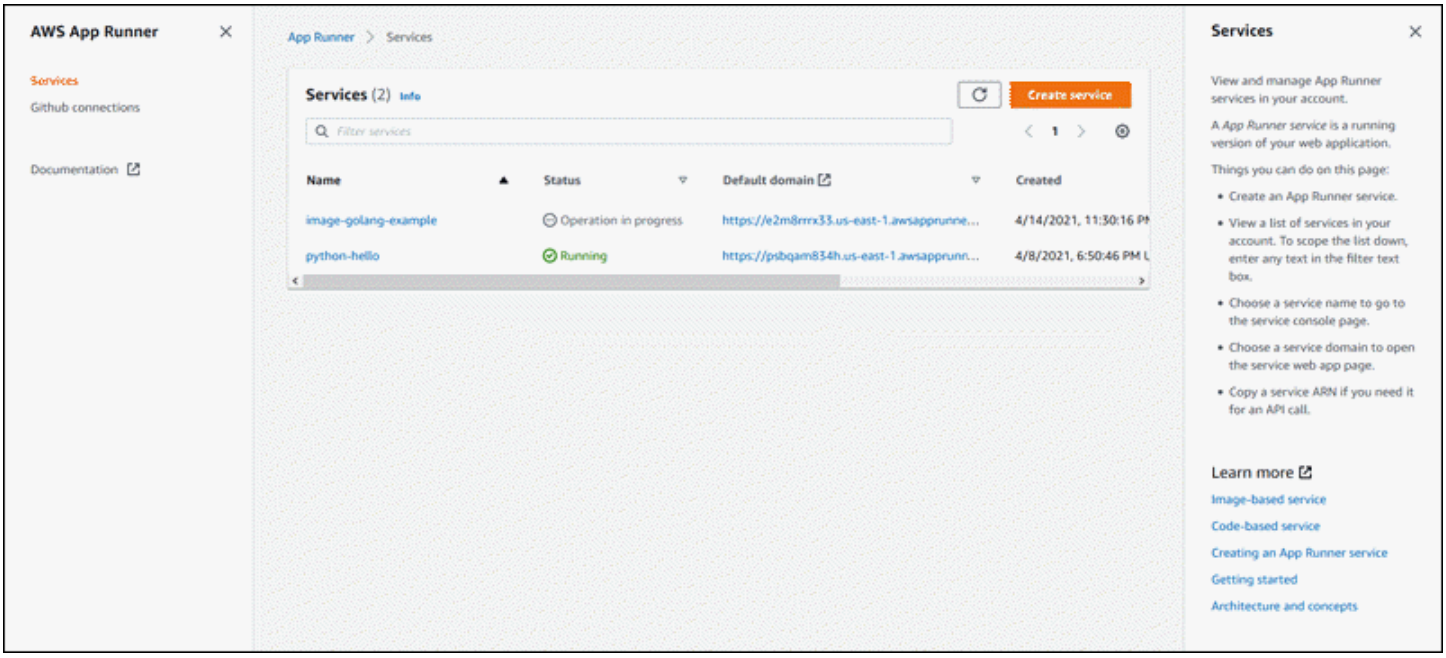
Utilizzo dell'AWS App Runnerper creare, gestire e monitorare i servizi App Runner e le risorse correlate, ad esempio le connessioni. È possibile visualizzare i servizi esistenti, crearne di nuovi e configurare un servizio. È possibile visualizzare lo stato di un servizio App Runner, nonché visualizzare i registri, monitorare l'attività e tenere traccia delle metriche. È inoltre possibile accedere al sito Web del servizio o al repository di origine.

Nelle sezioni seguenti vengono descritti il layout e le funzionalità della console e vengono indicate le informazioni correlate.

Layout console generale

La console App Runner ha tre aree. Da sinistra a destra:

- Riquadro di navigazione— Un riquadro laterale che può essere compresso o espanso. Usalo per scegliere la pagina della console di livello superiore che si desidera utilizzare.
- Riquadro del contenuto— La parte principale della pagina della console. Usalo per visualizzare le informazioni ed eseguire le attività.
- Riquadro della Guida— Un riquadro laterale per ulteriori informazioni. Espandilo per ricevere assistenza sulla pagina in cui ti trovi. Oppure scegli qualsiasiInfoin una pagina della console per ottenere una guida contestuale.



Pagina dei servizi

LaServizielenca i servizi App Runner nel tuo account. È possibile definire l'ambito dell'elenco in basso utilizzando la casella di testo del filtro.

Per accedere alla paginaServicespage

1. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, scegliereServizi: .

Cose che puoi fare qui:

- Creare un servizio App Runner. Per ulteriori informazioni, consulta the section called “Creazione”.
- Scegliere un nome di servizio per accedere alla pagina della console del dashboard del servizio.
- Scegliere un dominio di servizio per aprire la pagina dell'app Web del servizio.

Pagina Dashboard del servizio

È possibile visualizzare le informazioni su un servizio App Runner e gestirlo dalla pagina dashboard del servizio. Nella parte superiore della pagina, puoi vedere il nome del servizio.

Per accedere al dashboard del servizio, passare alla schedaServizi(vedere la sezione precedente), quindi scegliere il servizio App Runner.

LaPanoramica del serviziofornisce dettagli di base sul servizio App Runner e sull'applicazione. Cose che puoi fare qui:

- Visualizzare i dettagli del servizio, ad esempio stato, integrità e ARN.
- Passare alla .Dominio predefinito, il dominio fornito da App Runner per l'applicazione Web in esecuzione nel servizio. Questo è un sottodominio nella`awsapprunner.com`di proprietà di App Runner.
- Passare al repository di origine distribuito al servizio.
- Avviare una distribuzione del repository di origine nel servizio.
- Metti in pausa, riprendi ed elimina il tuo servizio.

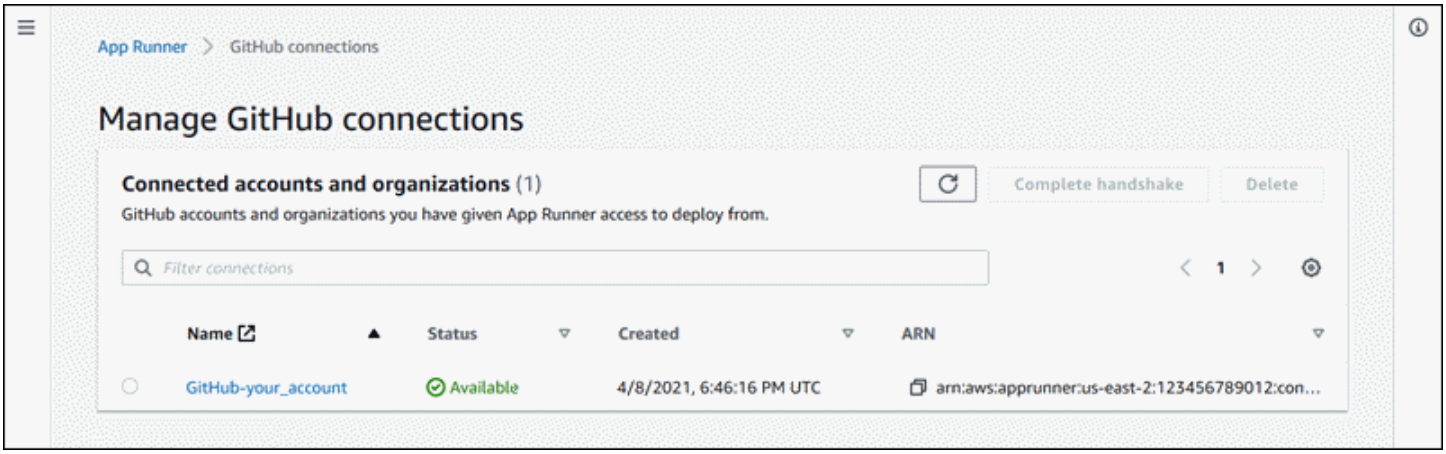
Le schede sotto la panoramica del servizio sono per il serviziogestioneeMonitoraggio di: .

La pagina delle connessioni di GitHub

LaConnessioni GitHubelenca le connessioni di App Runner a GitHub nel tuo account. È possibile definire l'ambito dell'elenco in basso utilizzando la casella di testo del filtro. Per ulteriori informazioni sulle connessioni, consulta the section called “Connessioni”.

Per accedere alla paginaConnessioni GitHubpage

1. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, scegliereConnessioni GitHub: .



Cose che puoi fare qui:

- Visualizza un elenco di connessioni GitHub nel tuo account. Per definire l'ambito dell'elenco verso il basso, immettere qualsiasi testo nella casella di testo del filtro.
- Scegliere un nome di connessione per accedere all'account o all'organizzazione GitHub correlata.
- Selezionare una connessione per completare l'handshake per una connessione appena stabilita (come parte della creazione di un servizio) o per eliminare la connessione.

Gestione del ciclo di vita del servizio App Runner

Questo capitolo descrive come gestire il ciclo di vita deiAWS App RunnerIl servizio. In questo capitolo viene illustrato come creare, configurare ed eliminare un servizio, come distribuire nuove versioni dell'applicazione al servizio e come gestire le connessioni. Imparerai inoltre a controllare la disponibilità del tuo servizio Web mettendo in pausa e riprendendo il servizio.

Argomenti

Creazione di un servizio App Runner

AWS App Runnerautomatizza il processo di passaggio da un'immagine contenitore o da un repository di codice sorgente a un servizio Web in esecuzione scalabile automaticamente. Si punta App Runner all'immagine o al codice sorgente, specificando solo un piccolo numero di impostazioni richieste. App Runner crea l'applicazione, se necessario, esegue il provisioning delle risorse di calcolo e distribuisce l'applicazione per l'esecuzione su di esse.

Quando crei un servizio, App Runner crea unservizioRisorsa. In alcuni casi, potrebbe essere necessario fornire unConnessioneRisorsa. Se si utilizza la console App Runner, la console crea implicitamente la risorsa di connessione. Per informazioni dettagliate sui tipi di risorse di App Runner, consultathe section called “Risorse App Runner”: . Questi tipi di risorse hanno quote associate all'account in ogniRegione AWS: . Per ulteriori informazioni, consulta the section called “Quote di risorse App Runner”.

Esistono sottili differenze nella procedura per la creazione di un servizio a seconda del tipo di origine e del provider. In questo argomento vengono illustrate procedure completamente separate per la

creazione di questi diversi tipi di origine, in modo da poter seguire la situazione più adatta alla propria situazione. Per una procedura di base di avvio con un esempio di codice, consultaNozioni di base: .

Prerequisites

Prima di creare il servizio App Runner, assicurati di completare le seguenti azioni:

- Completa le fasi di configurazione descritte inConfigurazione: .
- Avere la sorgente dell'applicazione pronta. È possibile utilizzare un repository di codice inGitHubo un'immagine contenitore inAmazon Elastic Container Registry (Amazon ECR)per creare un servizio App Runner.

Crea un servizio.

Questa sezione illustra il processo di creazione per i due tipi di servizio App Runner: basato sul codice sorgente e basato su un'immagine contenitore.

Creare un servizio da un repository di codice GitHub

Le sezioni seguenti illustrano come creare un servizio App Runner quando l'origine è un repository di codice inGitHub: . Quando si utilizza GitHub, App Runner deve connettersi all'organizzazione o all'account GitHub. Pertanto, è necessario aiutare a stabilire questa connessione. Per ulteriori informazioni sulle connessioni di App Runner, vedithe section called “Connessioni”: .

Quando crei il servizio, App Runner crea un'immagine Docker contenente il codice dell'applicazione e le dipendenze. Viene quindi avviato un servizio che esegue un'istanza contenitore di questa immagine.

Argomenti

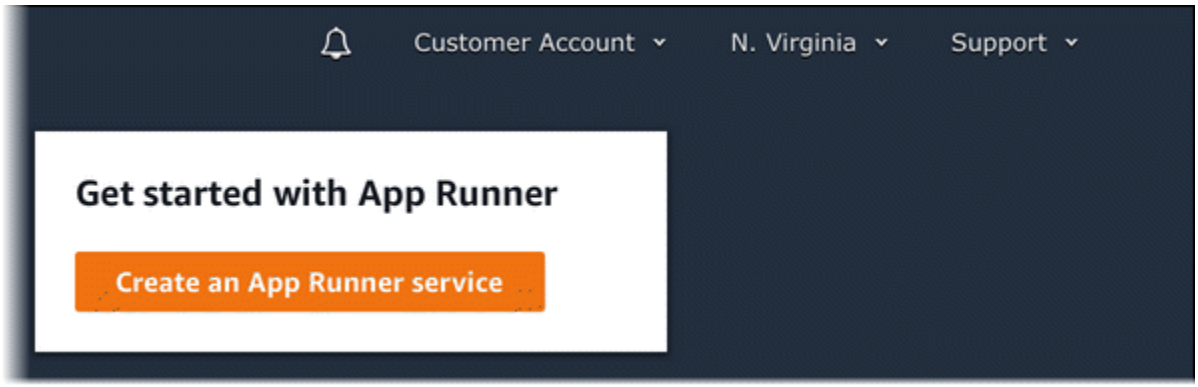
- Creazione di un servizio dal codice utilizzando la console App Runner
- Creazione di un servizio dal codice utilizzando l'API App Runner oAWS CLI

Creazione di un servizio dal codice utilizzando la console App Runner

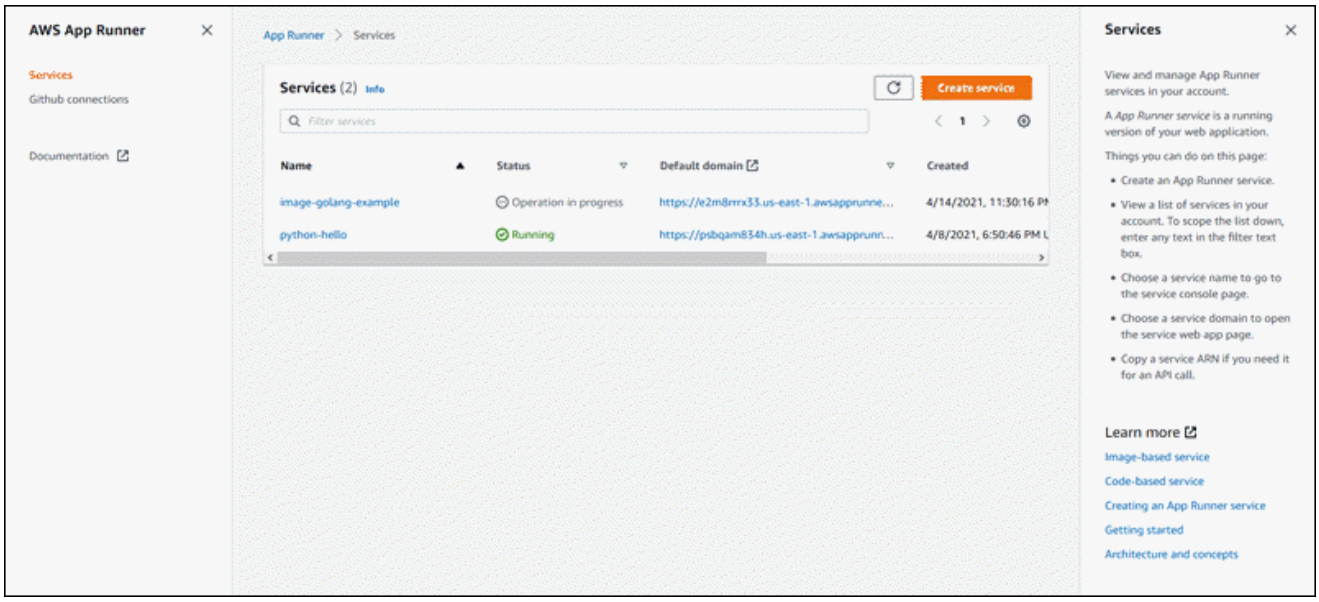
Per creare un servizio App Runner tramite la console

1. Configura il codice sorgente.
    a. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .

b. Se il fileAccount AWSnon dispone ancora di servizi App Runner, viene visualizzata la home page della console. ScegliereCreare un servizio App Runner: .



Se il fileAccount AWSdispone di servizi esistenti, ilServiziViene visualizzata una lista dei servizi. Selezionare Create service (Crea servizio).



c. SulOrigine e distribuzione(Italiano), nellaCreaSezione, perRepository type (Tipo di repository), sceglieReRepository codice sorgente: .

d. PerConnect a GitHub, seleziona un account o un'organizzazione GitHub che hai usato in precedenza oppure scegliAggiungi nuovo: . Quindi, passare attraverso il processo di fornitura delle credenziali GitHub e scegliere un account o un'organizzazione GitHub a cui connettersi.

e. PerRepositorySelezionare il repository che contiene il codice dell'applicazione.

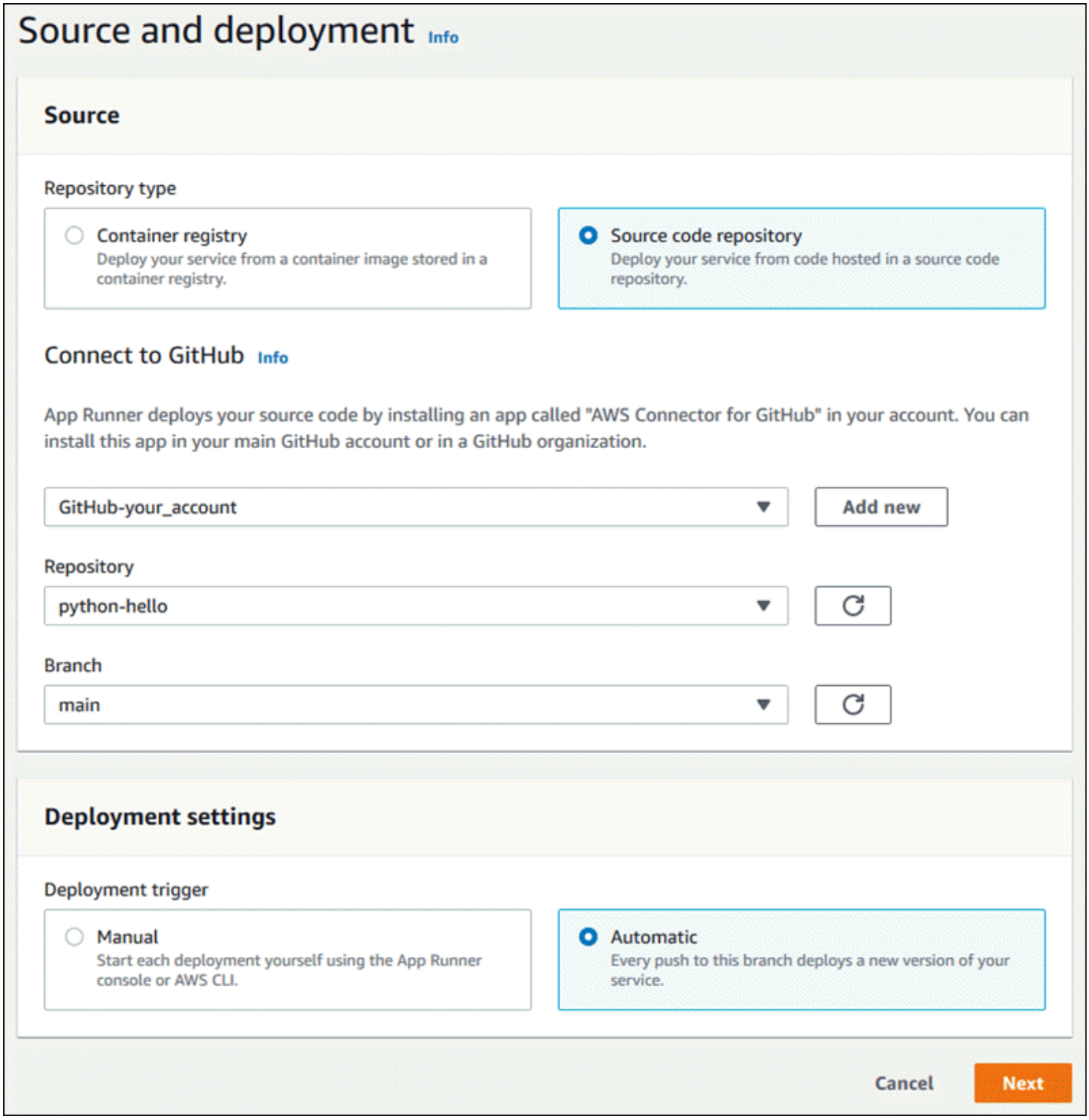
f. PerRamo, selezionare il ramo che si desidera distribuire.

2. Configurare le distribuzioni.

a. NellaImpostazioni di distribuzione, sceglieManualeoAutomatic: .

   Per ulteriori informazioni sui metodi di distribuzione, vedithe section called “Metodi di distribuzione”: .

b. Seleziona Successivo.



3. Configurare la build dell'applicazione.

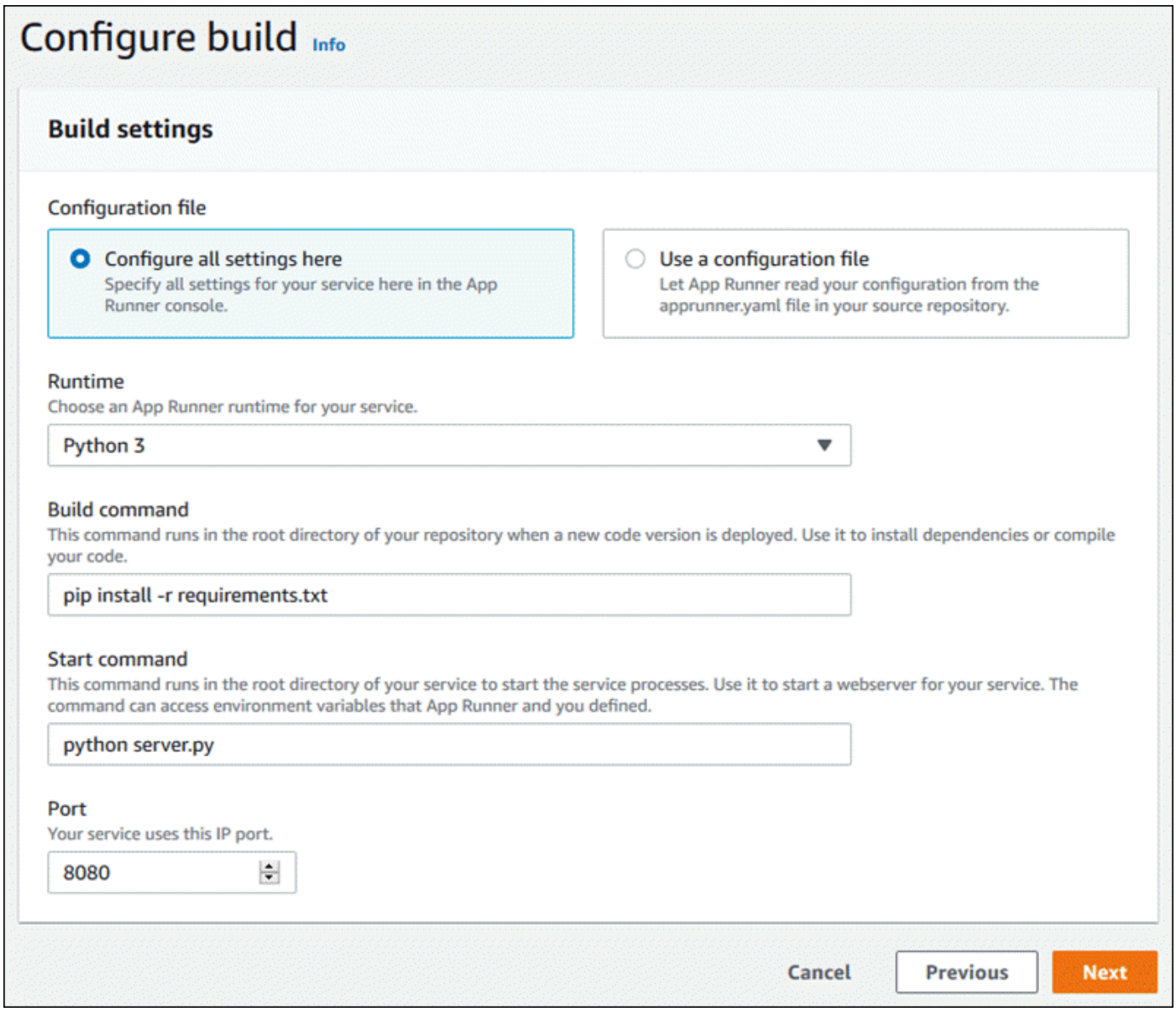
   a. SulConfigura compilazione(Italiano), perFile di configurazione, scegliereConfigura qui tutte le impostazionise il repository non contiene un file di configurazione di App Runner, oppureUtilizzo di un file di configurazioneSe funziona correttamente.

      > **Note**
      >
      > Un file di configurazione di App Runner è un modo per mantenere la configurazione di build come parte dell'origine dell'applicazione. Quando ne fornisci uno, App Runner legge alcuni valori dal file e non consente di impostarli nella console.

   b. Fornire le seguenti impostazioni di compilazione:

      - Runtime: scegliere un runtime gestito specifico per l'applicazione.
      - Comando Compila— Immettere un comando che crei l'applicazione dal codice sorgente. Potrebbe trattarsi di uno strumento specifico della lingua o di uno script fornito con il codice.
      - Comando Avvia— Immettere il comando che avvia il servizio Web.
      - Porta: immettere la porta IP che il servizio Web è in ascolto.

   c. Seleziona Successivo.



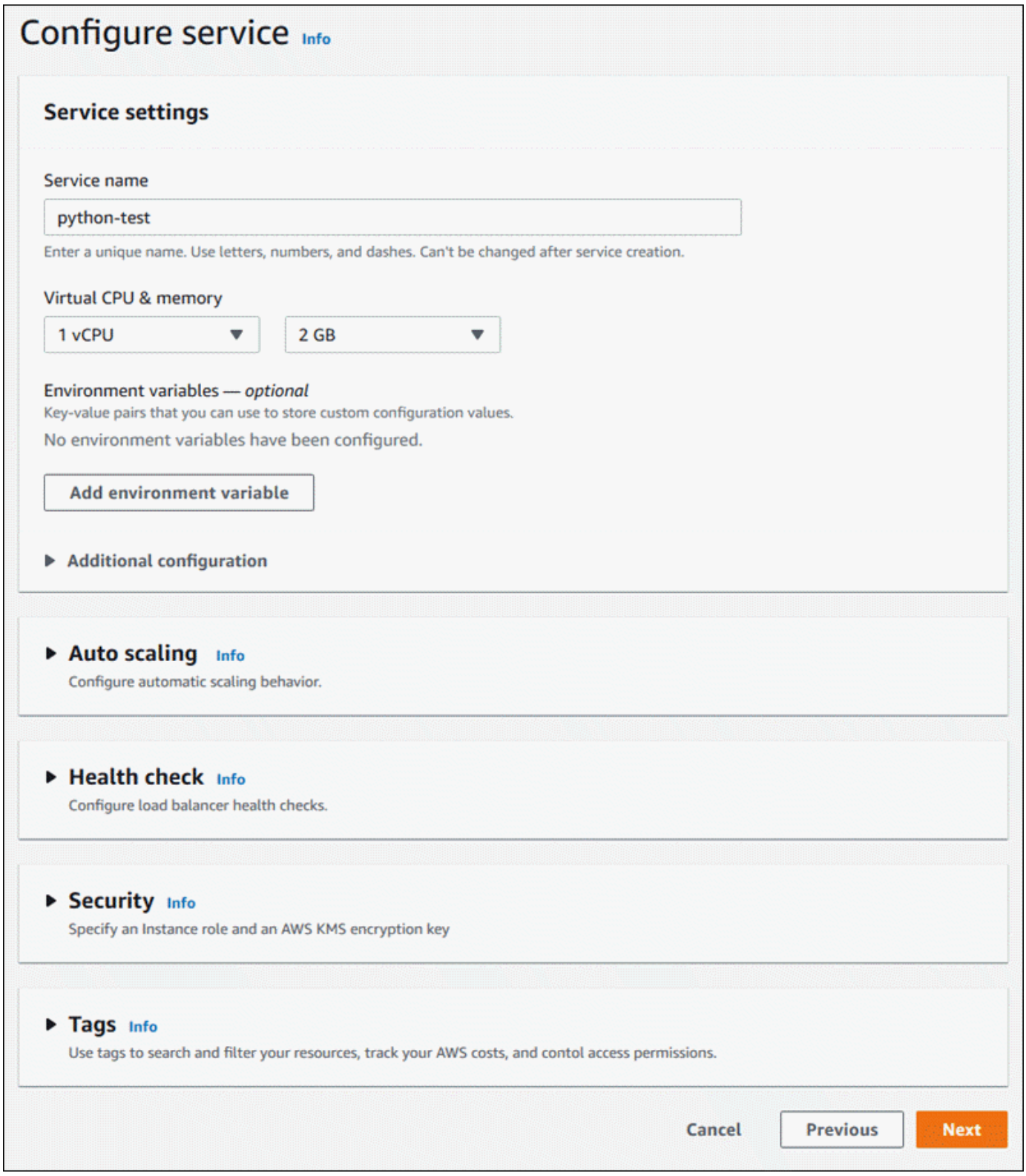
4. Configurare il servizio.

   a. SulConfigurare il servizio(Italiano), nellaImpostazioni del servizio, immettere un nome di servizio.

   > **Note**
   >
   > Tutte le altre impostazioni del servizio sono opzionali o dispongono di impostazioni predefinite fornite dalla console.

b. Se lo si desidera, modificare o aggiungere altre impostazioni per soddisfare i propri requisiti in termini di applicazioni.

c. Seleziona Successivo.



5. SulRivedi e creaverificare tutti i dettagli immessi, quindi sceglierCreazione e distribuzione: .

Risultato: Se la creazione del servizio ha esito positivo, la console dovrebbe mostrare il dashboard del servizio, con unPanoramica del serviziodel nuovo servizio.

6. Verificare che il servizio sia in esecuzione.

   a. Nella pagina del dashboard del servizio, attendere che il servizioStatoèIn esecuzione: .

   b. SelezionaDominio predefinito: è l'URL del sito Web del tuo servizio.

   c. Utilizza il sito Web e verifica che sia in esecuzione correttamente.

Creazione di un servizio dal codice utilizzando l'API App Runner oAWS CLI

Per creare un servizio utilizzando l'API App Runner oAWS CLI, richiama`CreateService`Operazione API. Per ulteriori informazioni e un esempio, consultaCreateService: . Se questa è la prima volta che crei un servizio utilizzando un'organizzazione o un account GitHub specifico, inizia chiamandoCreateConnection: . Questo stabilisce una connessione tra App Runner e l'organizzazione o l'account GitHub. Per ulteriori informazioni sulle connessioni di App Runner, vedithe section called “Connessioni”: .

La creazione del servizio inizia se la chiamata restituisce una risposta corretta con unService (Servizio)oggetto visualizzazione`"Status": "CREATING"`: .

Per una chiamata di esempio, consultaCreazione di un servizio repository del codice sorgentenellaAWS App RunnerDocumentazione di riferimento API

Creazione di un servizio da un'immagine Amazon ECR

Le sezioni seguenti illustrano come creare un servizio App Runner quando l'origine è un'immagine contenitore archiviata inAmazon ECR: . Amazon ECR è unAWSServizio. Pertanto, per creare un servizio basato su un'immagine Amazon ECR, fornisci ad App Runner un ruolo di accesso contenente le necessarie autorizzazioni di azione ECR di Amazon.

> **Note**
>
> Non è necessario un ruolo di accesso se l'immagine è archiviata in Amazon ECR Public, dove le immagini sono pubblicamente disponibili.

Durante la creazione del servizio, App Runner avvia un servizio che esegue un'istanza contenitore dell'immagine fornita. Non esiste una fase di costruzione in questo caso.

Argomenti

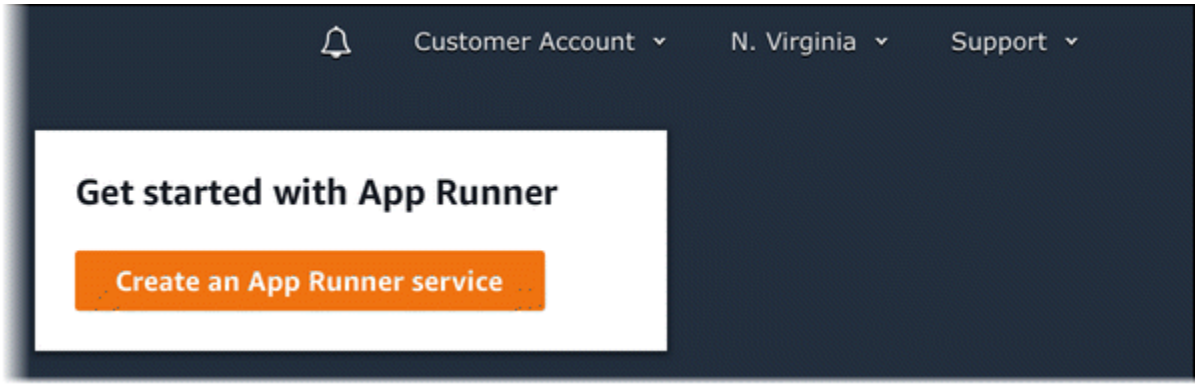
Creazione di un servizio da un'immagine utilizzando la console App Runner

Per creare un servizio App Runner tramite la console

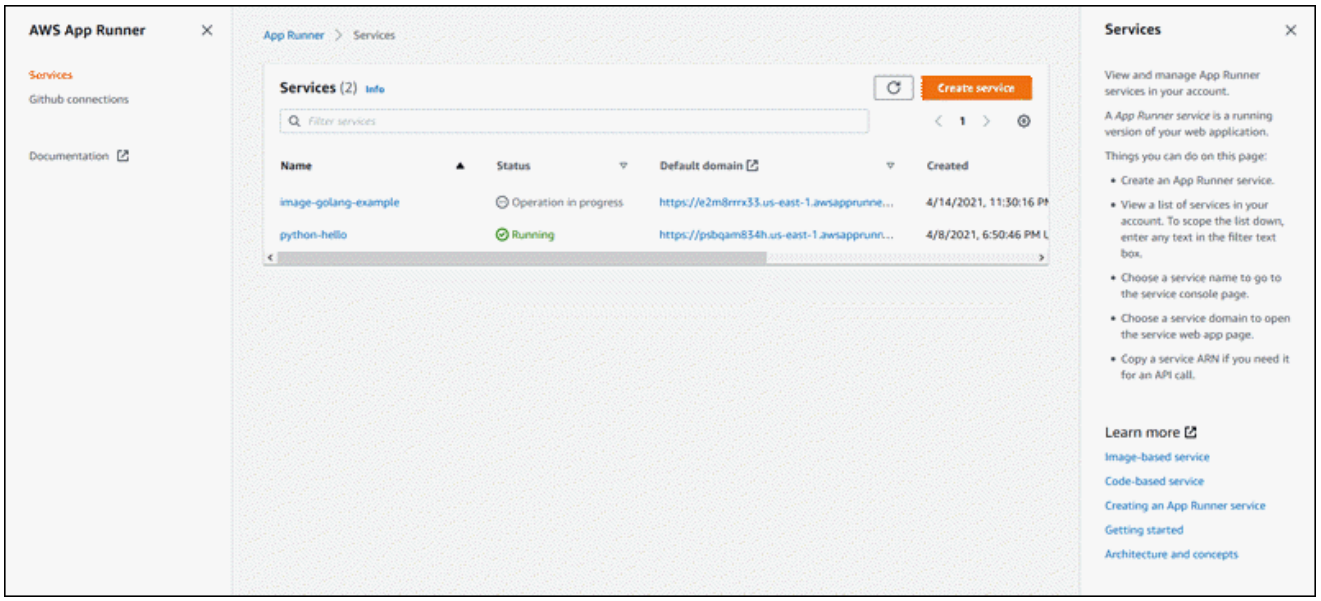
1. Configura il codice sorgente.

   a. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .

   b. Se il fileAccount AWSnon dispone ancora di servizi App Runner, viene visualizzata la home page della console. ScegliereCreare un servizio App Runner: .



   Se il fileAccount AWSdispone di servizi esistenti, ilServiziViene visualizzata una lista dei servizi. Selezionare Create service (Crea servizio).



c. SulOrigine e distribuzione(Italiano), nellaCreaSezione, perRepository type (Tipo di repository), scegliereRegistro di sistema container: .

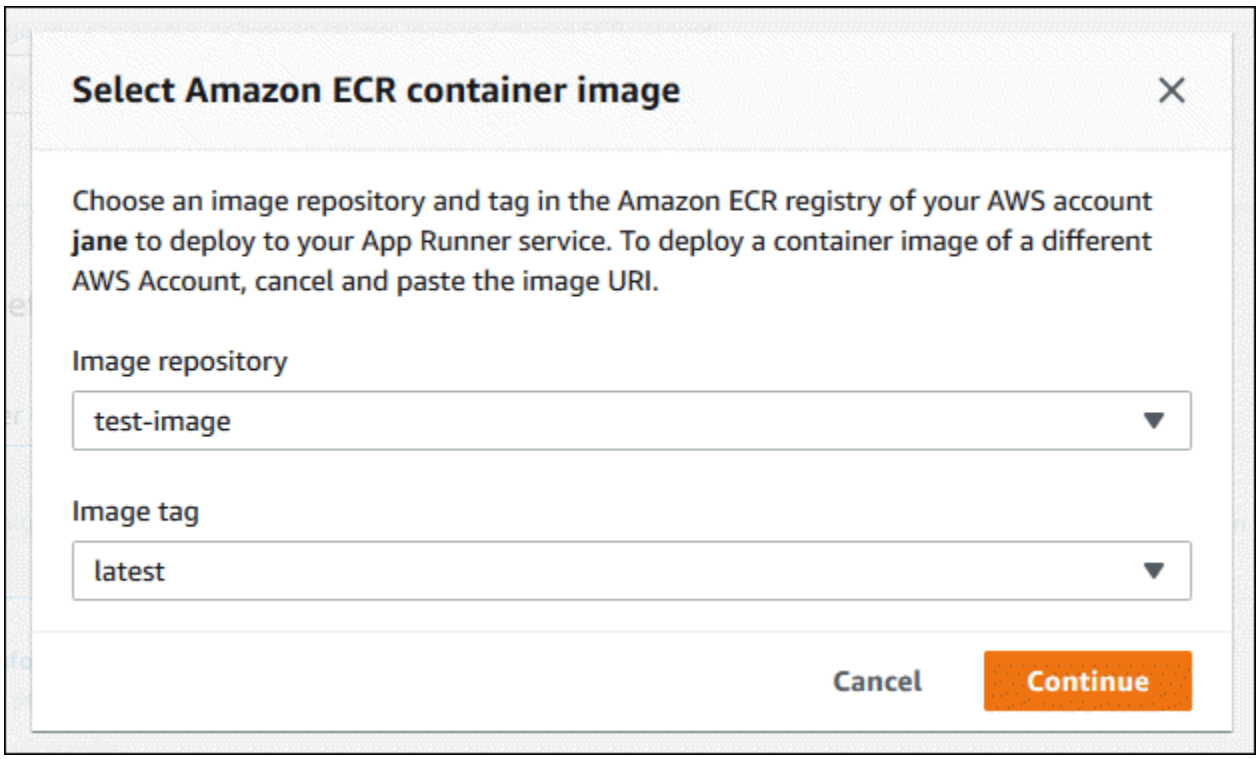
d. PerProvider, scegli il provider in cui è memorizzata l'immagine:

- Amazon ECR— Un'immagine privata memorizzata in Amazon ECR nel tuoAccount AWS: .
- Amazon ECR Public— Un'immagine pubblicamente leggibile memorizzata in Amazon ECR Public.

e. PerURI immagine contenitore, scegliereSfoglia: .

f. NellaSeleziona l'immagine del contenitore Amazon ECRFinestra di dialogo, perRepository immagine, selezionare il repository che contiene l'immagine.

g. PerTag immagineSelezionare il tag immagine specifico che si desidera distribuire, ad esempio(più recente), quindi sceglieréContinua: .



2. Configurare le distribuzioni.

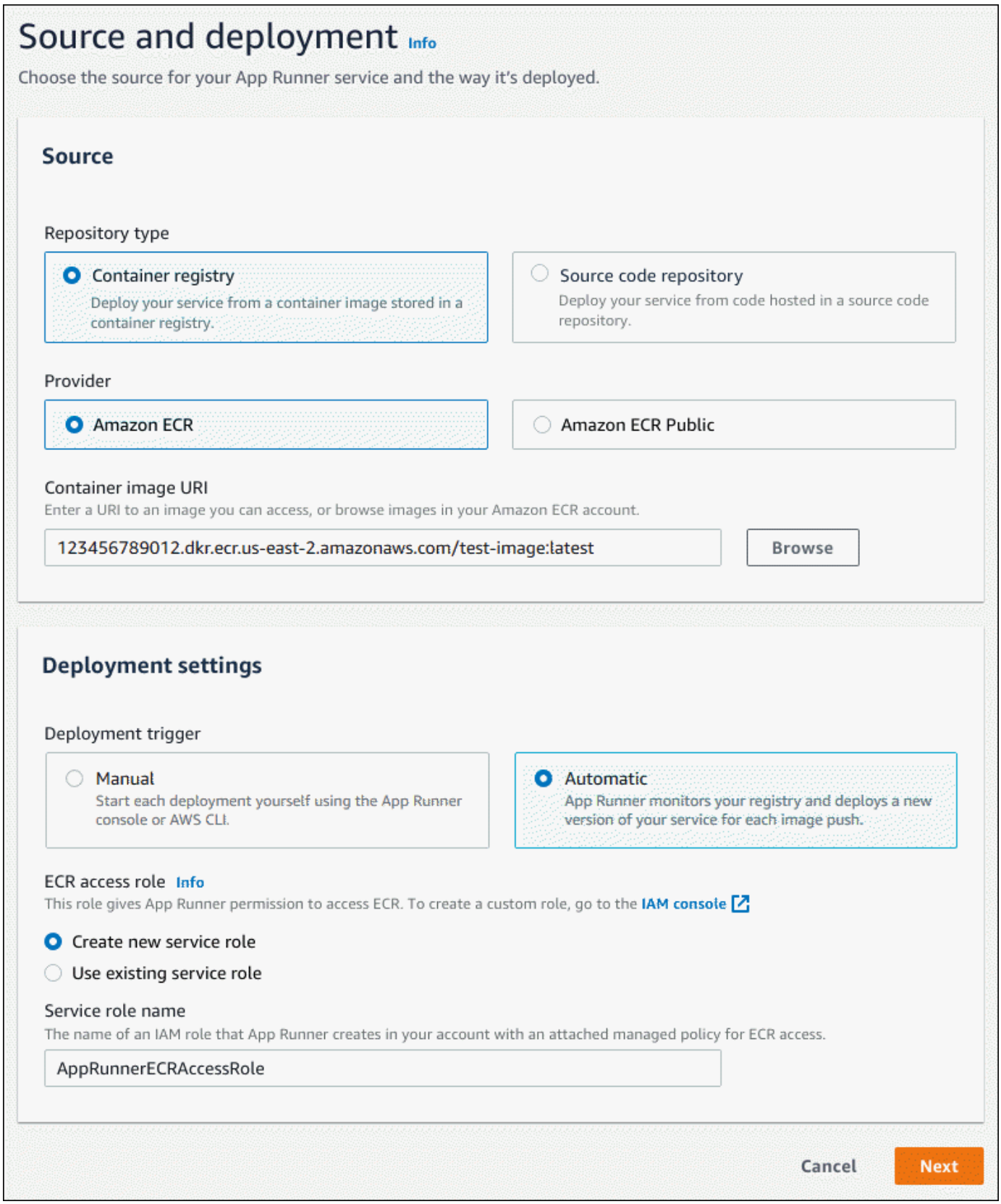
a. NellaImpostazioni di distribuzione, sceglieréManualeoAutomatic: .

Per ulteriori informazioni sui metodi di distribuzione, vedithe section called “Metodi di distribuzione”: .

> **Note**
>
> App Runner non supporta la distribuzione automatica per le immagini pubbliche Amazon ECR.

b. [Amazon ECRprovider] PerRuolo di accesso ECRPuoi scegliere un ruolo di servizio esistente nel tuo account o creane uno nuovo. Se si utilizza la distribuzione manuale, è inoltre possibile scegliere di utilizzare il ruolo utente IAM al momento della distribuzione.

c. Seleziona Successivo.


Source and deployment Info
Choose the source for your App Runner service and the way it's deployed.
Source
Repository type
Container registry
Deploy your service from a container image stored in a container registry.
Source code repository
Deploy your service from code hosted in a source code repository.
Provider
Amazon ECR
Amazon ECR Public
Container image URI
Enter a URI to an image you can access, or browse images in your Amazon ECR account.
123456789012.dkr.ecr.us-east-2.amazonaws.com/test-image:latest
Browse
Deployment settings
Deployment trigger
Manual
Start each deployment yourself using the App Runner console or AWS CLI.
Automatic
App Runner monitors your registry and deploys a new version of your service for each image push.
ECR access role Info
This role gives App Runner permission to access ECR. To create a custom role, go to the IAM console
Create new service role
Use existing service role
Service role name
The name of an IAM role that App Runner creates in your account with an attached managed policy for ECR access.
AppRunnerECRAccessRole
Cancel
Next

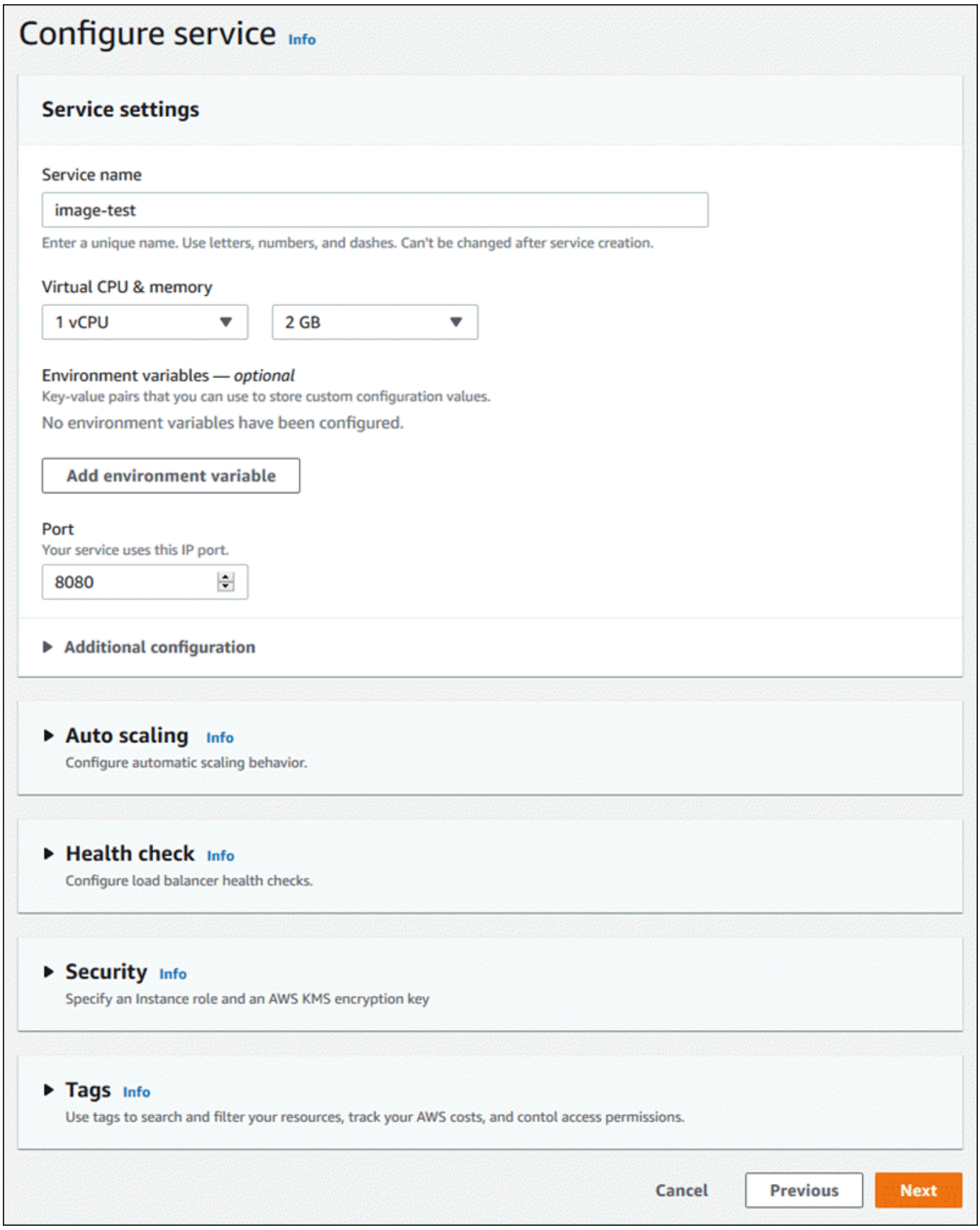
3. Configurare il servizio.

   a. SulConfigurare il servizio(Italiano), nellaImpostazioni del servizio, immettere un nome di servizio e la porta IP che il sito Web del servizio è in ascolto.

      > **Note**
      >
      > Tutte le altre impostazioni del servizio sono opzionali o dispongono di impostazioni predefinite fornite dalla console.

   b. (Facoltativo) Modificare o aggiungere altre impostazioni in base alle esigenze dell'applicazione.

   c. Seleziona Successivo.


Configure service Info
Service settings
Service name
image-test
Enter a unique name. Use letters, numbers, and dashes. Can't be changed after service creation.
Virtual CPU & memory
1 vCPU
2 GB
Environment variables — optional
Key-value pairs that you can use to store custom configuration values.
No environment variables have been configured.
Add environment variable
Port
Your service uses this IP port.
8080
Additional configuration
Auto scaling Info
Configure automatic scaling behavior.
Health check Info
Configure load balancer health checks.
Security Info
Specify an Instance role and an AWS KMS encryption key
Tags Info
Use tags to search and filter your resources, track your AWS costs, and contol access permissions.
Cancel
Previous
Next

4. SulRivedi e creaVerificare tutti i dettagli che sono stati inseriti e selezionareCreazione e distribuzione: .

   Risultato: Se la creazione del servizio ha esito positivo, la console dovrebbe mostrare il dashboard del servizio, con unPanoramica del serviziodel nuovo servizio.

5. Verificare che il servizio sia in esecuzione.

   a. Nella pagina del dashboard del servizio, attendere che il servizioStatoèIn esecuzione: .

   b. SelezionaDominio predefinito: è l'URL del sito Web del tuo servizio.

   c. Utilizza il sito Web e verifica che sia in esecuzione correttamente.

Creazione di un servizio da un'immagine utilizzando l'API App Runner oAWS CLI

Per creare un servizio utilizzando l'API App Runner oAWS CLI, richiamaCreateServiceOperazione API.

La creazione del servizio inizia se la chiamata restituisce una risposta corretta con unService (Servizio)oggetto visualizzazione`"Status": "CREATING"`: .

Per una chiamata di esempio, consultaCreare un servizio repository di immagini di originenellaAWS App RunnerDocumentazione di riferimento API

Quando la creazione del servizio non riesce

Se il tentativo di creare un servizio App Runner non riesce, il servizio mostra lo stato`CREATE_FAILED`(Creazione non riuscitasulla console).

Il tentativo di creare un servizio potrebbe non riuscire a causa di problemi nel codice dell'applicazione, nel processo di compilazione o nella configurazione, perché sono state raggiunte le quote di risorse o a causa di problemi temporanei con ilAWSservizi che il servizio deve utilizzare. Per risolvere un errore, consigliamo di procedere come segue. Per prima cosa, leggere gli eventi e i registri del servizio per scoprire cosa ha causato l'errore. Successivamente, apporta tutte le modifiche necessarie al codice o alla configurazione. Infine, eliminare uno o più servizi se hai raggiunto la quota di servizio. Quindi, dopo aver completato tutti questi passaggi, provare a creare nuovamente il servizio.

> **Important**
>
> Il servizio non riuscito non è utilizzabile. Non ti verrà addebitato alcun costo aggiuntivo oltre il tentativo di creazione iniziale. Tuttavia, App Runner non elimina automaticamente il servizio non riuscito e viene comunque conteggiato per la quota del servizio. Al termine dell'analisi dell'errore, assicurarsi di eliminare il servizio non riuscito.

Distribuzione di una nuova versione dell'applicazione in App Runner

QuandoCrea un servizio.inAWS App Runner, è possibile configurare un'origine applicazione, ovvero un'immagine contenitore o un repository di origine. App Runner fornisce risorse per eseguire il servizio e distribuisce l'applicazione su di essi.

In questo argomento vengono descritti i modi per ridistribuire l'origine dell'applicazione nel servizio App Runner quando diventa disponibile una nuova versione. Può trattarsi di una nuova versione dell'immagine nel repository delle immagini o di un nuovo commit nel repository del codice. App Runner fornisce due metodi per la distribuzione a un servizio:AutomaticeManuale: .

Metodi di distribuzione

App Runner fornisce i seguenti metodi per controllare il modo in cui vengono avviate le distribuzioni delle applicazioni.

Distribuzioni automatiche

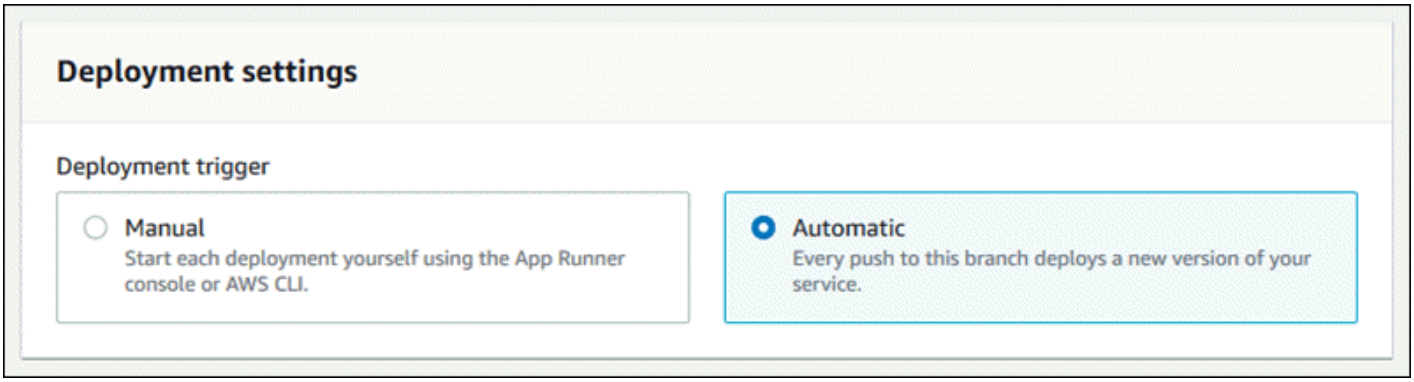
Utilizzare la distribuzione automatica quando si desidera un comportamento CICD (Continuous Integration and Deployment) per il servizio. App Runner monitora il tuo repository di immagini o codice. Ogni volta che invii una nuova versione di immagini nel tuo repository di immagini o un nuovo commit nel tuo repository di codice, App Runner la distribuisce automaticamente al tuo servizio senza ulteriori azioni da parte tua.

Distribuzioni manuali

Utilizzare la distribuzione manuale quando si desidera avviare esplicitamente ogni distribuzione nel servizio. Si avvia una distribuzione se il repository configurato per il servizio dispone di una nuova versione che si desidera distribuire. Per ulteriori informazioni, consulta the section called “Distribuzioni manuali”.

Puoi configurare il metodo di distribuzione per il servizio nei seguenti modi:

- Console: per un nuovo servizio che si sta creando o per un servizio esistente, nella finestraImpostazioni di distribuzioneSezione dellaOrigine e distribuzionePagina di configurazione, scegliereManualeoAutomatic: .



- API oAWS CLI— In una chiamata aCreateServiceoUpdateService, impostare la proprietà`AutoDeploymentsEnabled`membro delSourceConfigurationParametro per`False`per la distribuzione manuale o`True`per la distribuzione automatica.

Distribuzioni manuali

Con la distribuzione manuale, è necessario avviare esplicitamente ogni distribuzione nel servizio. Quando si dispone di una nuova versione dell'immagine o del codice dell'applicazione pronta per la distribuzione, è possibile fare riferimento alle sezioni seguenti per informazioni su come eseguire una distribuzione utilizzando la console e l'API.

Distribuire una versione dell'applicazione utilizzando la console App Runner

Per eseguire la distribuzione tramite la console App Runner

1. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionaServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica del servizio: .
3. Seleziona Deploy (Distribuisci).

   Risultato: Viene avviata la distribuzione della nuova versione. Nella pagina del dashboard del servizio, il servizioStatoModificazioni alleOperazione in corso: .

4. Attendere il termine della distribuzione. Nella pagina del dashboard del servizio, il servizioStatodovrebbe tornare aIn esecuzione: .
5. Per verificare che la distribuzione abbia esito positivo, nella pagina del dashboard del servizio selezionare la casella di controlloDominio predefinita: è l'URL del sito Web del tuo servizio. Ispeziona o interagisci con la tua applicazione web e verifica la modifica della versione.

Distribuire una versione dell'applicazione utilizzando l'API App Runner oAWS CLI

Per eseguire la distribuzione utilizzando l'API App Runner oAWS CLI, richiamaStartDeploymentOperazione API. L'unico parametro da passare è il tuo servizio ARN. È già stato configurato il percorso di origine dell'applicazione al momento della creazione del servizio e App Runner può trovare la nuova versione. La distribuzione viene avviata se la chiamata restituisce una risposta corretta.

Configurazione di un servizio di App Runner

QuandoCreazione di unAWS App Runnerservizio, si impostano vari valori di configurazione. Alcune di queste impostazioni di configurazione possono essere modificate dopo la creazione del servizio. Altre impostazioni possono essere applicate solo durante la creazione del servizio e non possono essere modificate in seguito. In questo argomento viene illustrata la configurazione del servizio utilizzando l'API App Runner, la console App Runner e un file di configurazione di App Runner.

Configurare il servizio utilizzando l'API App Runner oAWS CLI

L'API definisce quali impostazioni possono essere modificate dopo la creazione del servizio. Nell'elenco seguente vengono illustrate le azioni, i tipi e le limitazioni pertinenti.

- UpdateServiceaction — Può essere richiamato dopo la creazione per aggiornare alcune impostazioni di configurazione.
  - Può essere aggiornato— È possibile aggiornare le impostazioni nella finestra di dialogo`SourceConfiguration`,`InstanceConfiguration`, e`HealthCheckConfiguration`Parametri. Tuttavia, in`SourceConfiguration`, non puoi cambiare il tuo tipo di origine da codice a immagine o viceversa. È necessario fornire lo stesso parametro repository che è stato fornito a quando è stato creato il servizio. E' o...`CodeRepository`o`ImageRepository`: .

È inoltre possibile aggiornare`AutoScalingConfigurationArn`, l'ARN della risorsa di configurazione di ridimensionamento automatico associata al servizio.

- Non può essere aggiornato— Non è possibile modificare il`ServiceName`e`EncryptionConfiguration`che sono disponibili nella finestra di dialogoCreateServiceOperazione . Non possono essere modificati dopo la creazione. LaUpdateServicenon include questi parametri.
- File di API rispetto a— È possibile impostare la proprietà`ConfigurationSource`Parametro del parametroCodeConfiguration(utilizzato per i repository di codice sorgente come parte di`SourceConfiguration`) a`Repository`: . In questo caso, App Runner ignora le impostazioni di configurazione in`CodeConfigurationValues`e legge queste impostazioni da unFile di configurazioneNel repository. Se si imposta`ConfigurationSource`Da a`API`, App Runner ottiene tutte le impostazioni di configurazione dalla chiamata API e ignora il file di configurazione, anche se esiste.

- TagResourceaction — Può essere richiamata dopo la creazione del servizio per aggiungere tag al servizio o aggiornare i valori dei tag esistenti.
- UntagResourceaction — Può essere chiamato dopo la creazione del servizio per rimuovere i tag dal servizio.

Configurare il servizio utilizzando la console App Runner

La console utilizza l'API App Runner per applicare gli aggiornamenti di configurazione. Le regole di aggiornamento che l'API impone, come definito nella sezione precedente, determinano cosa è possibile configurare utilizzando la console. Alcune impostazioni che erano disponibili durante la creazione del servizio non sono disponibili per la modifica in un secondo momento. Inoltre, se si decide di utilizzare unFile di configurazione, le impostazioni aggiuntive vengono nascoste nella console e App Runner le legge dal file.

Per configurare il servizio

1. Apertura dellaConsole dell'app Runner, e nelRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionaServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica del servizio: .
3. Nella pagina di controllo del servizio, scegli la casella di controlloConfigurazioneScheda.

Risultato: La console visualizza le impostazioni di configurazione correnti del servizio in diverse sezioni: Origine e distribuzione,Configura build, eConfigurare il servizio: .

4. Per aggiornare le impostazioni in qualsiasi categoria, scegliereModificare: .
5. Nella pagina di modifica della configurazione apportare le modifiche desiderate, quindi selezionareSalva le modifiche: .

Configurare il servizio utilizzando un file di configurazione di App Runner

Quando si crea o si aggiorna un servizio App Runner, è possibile indicare ad App Runner di leggere alcune impostazioni di configurazione da un file di configurazione fornito come parte del repository di origine. In questo modo, è possibile gestire le impostazioni relative al codice sorgente sotto il controllo del codice sorgente, insieme al codice stesso. Il file di configurazione fornisce inoltre alcune impostazioni avanzate che non è possibile impostare utilizzando la console o l'API. Per ulteriori informazioni, consulta File di configurazione di App Runner.

Gestione delle connessioni di App Runner

QuandoCrea un servizio.inAWS App Runner, è possibile configurare un'origine applicazione, ovvero un'immagine contenitore o un repository di origine archiviato con un provider. Se un repository archiviato con un provider di terze parti è privato (non leggibile pubblicamente), App Runner deve stabilire una connessione autenticata e autorizzata con il provider. Quindi, App Runner può leggere il tuo repository e distribuirlo al tuo servizio. App Runner non richiede la creazione di una connessione quando si crea un servizio che accede al codice memorizzato nelAccount AWSo in una posizione con codice pubblico.

App Runner mantiene le informazioni di connessione in una risorsa chiamataConnessione: . App Runner richiede una risorsa di connessione quando si crea un servizio che richiede informazioni di connessione di terze parti. Di seguito sono riportate alcune informazioni importanti sulle connessioni:

- provider— App Runner richiede attualmente risorse di connessione conGitHub: .
- Condiviso: è possibile utilizzare una risorsa di connessione per creare più servizi App Runner che utilizzano lo stesso account del provider del repository.
- Gestione delle risorse— In App Runner, è possibile creare ed eliminare connessioni. Tuttavia, non è possibile modificare una connessione esistente.
- Quote di risorse: le risorse di connessione hanno una quota impostata associata alAccount AWSin ogniRegione AWS: . Se raggiungi questa quota, potrebbe essere necessario eliminare una

connessione prima di poterti connettere a un nuovo account del provider. È possibile eliminare una connessione tramite App Runner (App Runner)consoleoAPI: . Per ulteriori informazioni, consulta the section called “Quote di risorse App Runner”.

Gestire le connessioni utilizzando la console App Runner

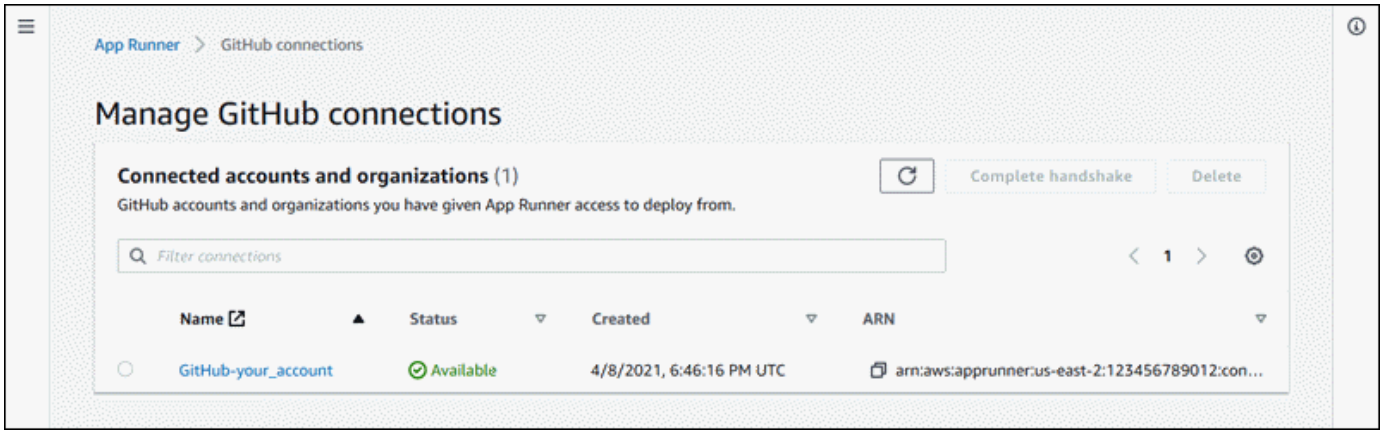
Quando si utilizza la console App Runner perCrea un servizio., fornisci i dettagli della connessione. Non è necessario creare in modo esplicito una risorsa di connessione. Nella console, puoi scegliere di connetterti a un account GitHub a cui ti sei connesso in precedenza o connetterti a un nuovo account. Se necessario, App Runner crea una risorsa di connessione per te. Per una nuova connessione, alcuni provider (ad esempio, GitHub) richiedono di completare un handshake di autenticazione prima di poter utilizzare la connessione. La console ti guierà attraverso questo processo.

La console dispone anche di una pagina per la gestione delle connessioni esistenti. È possibile completare l'handshake di autenticazione per una connessione se non è stato eseguito durante la creazione del servizio. È inoltre possibile eliminare le connessioni che non vengono più utilizzate. La procedura seguente descrive come gestire le connessioni GitHub.

Per gestire le connessioni GitHub nel tuo account

1. Apertura dellaConsole Runner di app, e nelRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionaConnessioni GitHub: .

   La console visualizza quindi un elenco di connessioni GitHub nel tuo account.



3. Ora puoi eseguire una delle seguenti azioni con qualsiasi connessione nell'elenco:

   - Apri l'account o l'organizzazione GitHub: Scegli il nome della connessione.

- Handshake di autenticazione completa— Selezionare la connessione, quindi scegliereHandshake completo: . La console consente di eseguire il processo di handshake di autenticazione.
- Elimina connessione— Selezionare la connessione, quindi scegliereElimina: . Seguire le istruzioni sul prompt di eliminazione.

Gestire le connessioni utilizzando l'API App Runner oAWS CLI

Puoi utilizzare le seguenti azioni dell'API di App Runner di per gestire le connessioni.

- CreateConnection— Crea una connessione a un account del provider del repository. Dopo aver creato la connessione, è necessario completare manualmente l'handshake di autenticazione utilizzando la console App Runner. Questo processo è spiegato nella sezione precedente.
- ListConnections— Restituisce un elenco di connessioni di App Runner associate alAccount AWS: .
- DeleteConnection— Elimina una connessione. Potrebbe essere necessario eliminare le connessioni non necessarie se si raggiunge la quota di connessione perAccount AWS: .

Gestione del ridimensionamento automatico di App Runner

AWS App Runnerridimensiona automaticamente le risorse di calcolo (istanze) verso l'alto o verso il basso per l'applicazione App Runner. Il ridimensionamento automatico fornisce una gestione adeguata delle richieste quando il traffico in entrata è elevato e riduce i costi quando il traffico rallenta. È possibile configurare alcuni parametri per regolare il comportamento di ridimensionamento automatico per il servizio.

App Runner mantiene le impostazioni di ridimensionamento automatico in una risorsa chiamataAutoscalingConfiguration: . Quando si crea o si aggiorna un servizio, puoi fornire una risorsa di configurazione di ridimensionamento automatico. La console App Runner ne crea una per te quando crei un nuovo servizio App Runner. Fornire una configurazione di ridimensionamento automatico è facoltativa. Se non ne fornisci una, App Runner fornisce una configurazione di ridimensionamento automatico predefinita con valori consigliati.

Una configurazione di ridimensionamento automatico ha unNomee un numericrevisione: . Più revisioni di una configurazione hanno lo stesso nome e numeri di revisione diversi. È possibile utilizzare nomi di configurazione diversi per scenari di ridimensionamento automatico diversi, ad

esempio ad alta disponibilità o basso costo. Per ogni nome, è possibile aggiungere più revisioni per ottimizzare le impostazioni per uno scenario specifico.

Di seguito sono riportate alcune informazioni importanti sulle configurazioni di ridimensionamento automatico:

- Impostazioni: Ecco come:
  - Simultaneità massima— Il numero massimo di richieste simultanee elaborate da un'istanza. Quando il numero di richieste simultanee supera questa quota, App Runner aumenta il servizio.
  - Dimensioni massime: il numero massimo di istanze del servizio scalabili fino a. Al massimo questo numero di istanze serve attivamente il traffico per il servizio.
  - Dimensione Min: il numero minimo di istanze di App Runner predisposte per il servizio. Il servizio dispone sempre di almeno questo numero di istanze di cui è stato eseguito il provisioning. Alcuni di loro servono attivamente il traffico. Il resto di essi (istanze con provisioning e inattive) è disponibile come riserva di capacità di calcolo conveniente, pronta per essere attivata rapidamente. Si paga per l'utilizzo della memoria di tutte le istanze di cui è stato eseguito il provisioning. Si paga per l'utilizzo della CPU solo del sottoinsieme attivo.

    App Runner raddoppia temporaneamente il numero di istanze di cui è stato eseguito il provisioning durante le distribuzioni, per mantenere la stessa capacità sia per il codice vecchio che per quello nuovo.
- Revisioni— La prima configurazione creata con un nome ottiene il numero di revisione 1. Le configurazioni successive con lo stesso nome ottengono numeri di revisione consecutivi (a partire da 2). È possibile associare il servizio App Runner a una specifica revisione della configurazione di ridimensionamento automatico o all'ultima revisione della configurazione.
- Condiviso: è possibile condividere una singola risorsa di configurazione di ridimensionamento automatico tra più servizi App Runner. Ciò è utile se hanno requisiti di ridimensionamento simili. In particolare, è possibile configurare più servizi in modo che tutti utilizzino la versione più recente di una configurazione specificando il nome della configurazione ma non specificando una revisione. In questo modo, tutti i servizi configurati in questo modo ricevono aggiornamenti di configurazione di ridimensionamento automatico quando si aggiorna il servizio. Per ulteriori informazioni sulle modifiche di configurazione di, consultathe section called “Configurazione”: .
- Gestione delle risorse— È possibile utilizzare App Runner per creare ed eliminare configurazioni di ridimensionamento automatico. Non è possibile aggiornare direttamente una configurazione. È invece possibile creare una nuova revisione per un nome di configurazione esistente per aggiornare efficacemente la configurazione.

> **Note**
>
> In questo momento, è possibile creare una configurazione solo con una singola revisione nella console App Runner. Per creare più revisioni ed eliminare configurazioni, utilizzare App RunnerAPI: .

- Quote di risorse— Esistono quote impostate per il numero di nomi di configurazione univoci e revisioni che è possibile avere per le risorse di configurazione di ridimensionamento automatico in ogniRegione AWS: . Se si raggiungono queste quote, è necessario eliminare un nome di configurazione o almeno alcune delle relative revisioni prima di poterne creare altre. Utilizzo dell'app RunnerAPIper eliminarli. Per ulteriori informazioni, consulta the section called “Quote di risorse App Runner”.

Gestire il ridimensionamento automatico utilizzando la console App Runner

QuandoCreare un servizionella console App Runner, è possibile utilizzare la configurazione di ridimensionamento automatico predefinita o una configurazione personalizzata. Per utilizzare una configurazione personalizzata, scegliere una configurazione esistente o fornire un nuovo nome e impostazioni. Se si tratta di una nuova configurazione, App Runner crea una nuova risorsa di configurazione di ridimensionamento automatico e quindi la associa al nuovo servizio.

Gestire il ridimensionamento automatico utilizzando l'API App Runner oAWS CLI

Per gestire le configurazioni di ridimensionamento automatico, puoi usare le operazioni dell'API di App Runner di elencate di seguito.

- CreateAutoScalingConfiguration— Crea una nuova configurazione di ridimensionamento automatico o una revisione rispetto a una configurazione esistente.
- ListAutoScalingConfigurations— Restituisce un elenco delle configurazioni di ridimensionamento automatico associate alAccount AWS, con informazioni di riepilogo.
- DescribeAutoScalingConfiguration— Restituisce una descrizione completa di una configurazione di ridimensionamento automatico.
- DeleteAutoScalingConfiguration— Elimina una configurazione di ridimensionamento automatico. È possibile eliminare una revisione specifica o l'ultima revisione attiva. Potrebbe essere necessario

eliminare le configurazioni di ridimensionamento automatico non necessarie se si raggiunge la quota di configurazione della scalabilità automatica per ilAccount AWS: .

Gestione di nomi di dominio personalizzati per un servizio App Runner

Quando si crea un fileAWS App Runner, App Runner assegna un nome di dominio per esso. Questo è un sottodominio nella`awsapprunner.com`di proprietà di App Runner. Può essere utilizzato per accedere all'applicazione Web in esecuzione nel servizio.

Se disponi di un nome di dominio, puoi associarlo al servizio App Runner. Dopo che App Runner ha convalidato il nuovo dominio, può essere utilizzato per accedere all'applicazione oltre al dominio App Runner. Puoi associare fino a cinque domini personalizzati.

> **Note**
>
> Puoi anche includere il file`www`sottodominio del tuo dominio. Tuttavia, questo è attualmente supportato solo nell'API. La console App Runner non lo supporta.

Quando si associa un dominio personalizzato al servizio, App Runner fornisce un set di record di convalida dei certificati. Aggiungili al tuo Domain Name System (DNS) in modo che App Runner possa convalidare che tu possiedi o controlli il dominio. Inoltre, aggiungere i record CNAME o ALIAS al DNS per indirizzare il dominio App Runner. È necessario aggiungere un record per il dominio personalizzato e un altro per il`www`, se hai scelto questa opzione. Quindi attendi che lo stato del dominio personalizzato diventiActive (Attivo)nella console App Runner. Questo richiede in genere diversi minuti (ma potrebbe richiedere 24-48 ore). A questo punto, il dominio personalizzato viene convalidato e App Runner avvia il routing del traffico da questo dominio all'applicazione Web.

È possibile specificare un dominio da associare al servizio App Runner nei seguenti modi:

- Un dominio radice— Ad esempio,`example.com`: . È possibile associare facoltativamente`www.example.com`Anche come parte della stessa operazione.
- Un sottodominio— Ad esempio,`login.example.com`o`admin.login.example.com`: . Facoltativamente, è possibile associare il`www`Sottodominio anche come parte della stessa operazione.

- Carattere jolly— Ad esempio,`*.example.com`: . Non è possibile utilizzarewwwIn questo caso. È possibile specificare un carattere jolly solo come sottodominio immediato di un dominio radice e solo da solo (queste non sono specifiche valide:`login*.example.com`,`*.login.example.com`). Questa specifica jolly associa tutti i sottodomini immediati e non associa il dominio radice stesso (il dominio principale dovrebbe essere associato in un'operazione separata).

Un'associazione di dominio più specifica sostituisce quella meno specifica. Ad esempio:`login.example.com`Overrides`*.example.com`: . Vengono utilizzati il certificato e il CNAME dell'associazione più specifica.

Nell'esempio seguente viene illustrato come utilizzare più associazioni di dominio personalizzate:

1. Associate`example.com`con la home page del tuo servizio. Abilitazione di`www`per associare anche`www.example.com`: .
2. Associate`login.example.com`con la pagina di accesso del tuo servizio.
3. Associate`*.example.com`con una pagina personalizzata «non trovata».

È possibile disassociare (scollegare) un dominio personalizzato dal servizio App Runner. Quando si scollega un dominio, App Runner interrompe il routing del traffico da questo dominio all'applicazione Web. È necessario eliminare i record per questo dominio dal DNS.

App Runner crea internamente certificati che tengono traccia della validità del dominio. Sono memorizzati inAWS Certificate Manager(ACM). App Runner non elimina questi certificati per sette giorni dopo la disassociazione di un dominio dal servizio o dopo l'eliminazione del servizio.

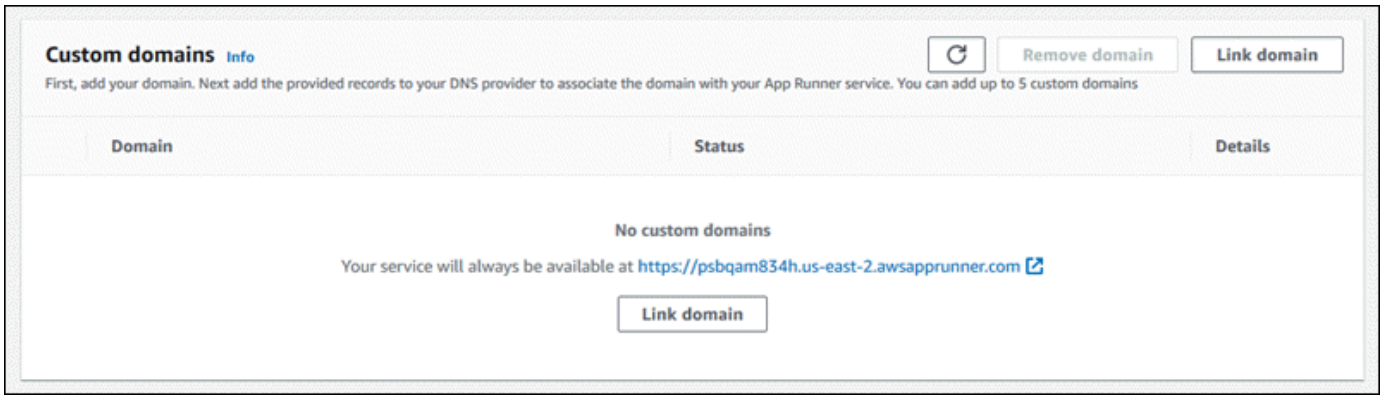
Gestire domini personalizzati utilizzando la console App Runner

Per associare (collegare) un dominio personalizzato utilizzando la console App Runner

1. Apertura dellaConsole App Runner, e nelRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionareServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica del servizio: .
3. Nella pagina di controllo del servizio, selezionare la casella di controlloDomains personalizzatiScheda.

Nella console vengono visualizzati i domini personalizzati associati al servizio oppureNessun dominio personalizzato: .



4. SulDomains personalizzati, scegliereCollegamento del dominio: .
5. NellaCollegamento del dominio personalizzato, immettere un nome di dominio e scegliereVisualizzazione della configurazione DNS: .
6. Segui le istruzioni visualizzateConfigurazione di DNSper avviare il processo di convalida del dominio.
7. Quando lo stato del dominio cambia inActive (Attivo), verificare che il dominio funzioni per il routing del traffico esplorando esso.

Per disassociare (scollegare) un dominio personalizzato utilizzando la console App Runner

1. SulDomains personalizzatiSelezionare il riquadro del dominio che si desidera disassociare e sceglieresCollegamento del dominio: .
2. NellaScollegamento del dominio, verificare l'azione scegliendoScollegamento del dominio: .

Gestire domini personalizzati utilizzando l'API App Runner oAWS CLI

Per associare un dominio personalizzato al servizio utilizzando l'API App Runner oAWS CLI, richiamaDominio AssociateCustomDomainOperazione API Quando la chiamata ha esito positivo, restituisce unCustomDomainche descrive il dominio personalizzato associato al servizio. L'oggetto dovrebbe mostrare uno stato di`CREATING`e contiene un elenco diCertificateValidationRecordOggetti. Questi sono record che puoi aggiungere al tuo DNS.

Per dissociare un dominio personalizzato dal servizio utilizzando l'API App Runner oAWS CLI, richiamaDisassociateCustomDomainOperazione API Quando la chiamata ha esito positivo,

restituisce unCustomDomainche descrive il dominio personalizzato che viene disassociato dal servizio. L'oggetto dovrebbe mostrare uno stato di`DELETING`: .

Sospensione e ripresa di un servizio App Runner

Se è necessario disabilitare temporaneamente l'applicazione Web e interrompere l'esecuzione del codice, è possibile sospendere ilAWS App RunnerServizio. App Runner riduce la capacità di calcolo del servizio a zero.

Quando si è pronti a eseguire nuovamente l'applicazione, è possibile riprendere il servizio App Runner. App Runner fornisce nuova capacità di elaborazione, distribuisce l'applicazione su di essa ed esegue l'applicazione. L'origine dell'applicazione non viene ridistribuita e non è necessaria alcuna compilazione. Piuttosto, App Runner riprende con la versione attualmente distribuita. L'applicazione mantiene il suo dominio App Runner.

⚠ Important

- Quando si sospende il servizio, l'applicazione perde il suo stato. Ad esempio, qualsiasi archiviazione effimera utilizzata dal codice viene persa. Per il codice, sospendere e riprendere il servizio equivale alla distribuzione in un nuovo servizio.
- Se si sospende un servizio a causa di un difetto nel codice (ad esempio, un bug rilevato o un problema di sicurezza), non è possibile distribuire una nuova versione prima di riprendere il servizio.

  Pertanto, si consiglia di mantenere il servizio in esecuzione e ripristinare l'ultima versione dell'applicazione stabile.
- Quando riprendi il servizio, App Runner distribuisce l'ultima versione dell'applicazione utilizzata prima di mettere in pausa il servizio. Se hai aggiunto nuove versioni di origine dopo la sospensione del servizio, App Runner non le distribuisce automaticamente anche se è selezionata la distribuzione automatica. Ad esempio, si supponga di avere nuove versioni di immagini nel repository di immagini o nuovi commit nel repository di codice. Queste versioni non vengono distribuite automaticamente.

  Per distribuire una versione più recente, eseguire una distribuzione manuale o aggiungere un'altra versione al repository di origine dopo aver ripreso il servizio App Runner.

Sospensione ed eliminazione confrontata

Pause (Metti in pausa)il servizio App Runner pertemporaneamentedisabilitate. Solo le risorse di calcolo vengono terminate e i dati archiviati (ad esempio, l'immagine contenitore con la versione dell'applicazione) rimangono intatti. La ripresa del servizio è rapida: l'applicazione è pronta per essere distribuita su nuove risorse di elaborazione. Il dominio App Runner resta lo stesso.

Eliminail servizio App Runner perpermanentementeRimuoverlo. I dati memorizzati vengono cancellati. Se è necessario ricreare il servizio, App Runner deve recuperare nuovamente la fonte e anche crearla se si tratta di un repository di codice. L'applicazione Web ottiene un nuovo dominio App Runner.

Quando il servizio è in pausa

Quando sospendi il tuo servizio ed è nellaIn pausa, risponde in modo diverso alle richieste di azione, incluse le chiamate API o le operazioni della console. Quando un servizio viene sospeso, è comunque possibile eseguire azioni di App Runner che non modificano la definizione o la configurazione del servizio in modo tale da influire sul runtime. In altre parole, se un'azione modifica il comportamento, la scala o altre caratteristiche di un servizio in esecuzione, non sarà possibile eseguire tale azione su un servizio in pausa.

Negli elenchi seguenti vengono fornite informazioni sulle azioni API che è possibile e non è possibile eseguire su un servizio in pausa. Le operazioni della console equivalenti sono analogamente consentite o negate.

Azionicaneseguire su un servizio in pausa

- *`List*`*e*`Describe*`*Operazioni— Azioni che leggono solo le informazioni.
- *`DeleteService`*— È sempre possibile eliminare un servizio.
- *`TagResource`*,*`UntagResource`*— I tag sono associati a un servizio, ma non fanno parte della sua definizione e non influenzano il suo comportamento di runtime.

Azionicannoteseguire su un servizio in pausa

- *`StartDeployment`*Operazioni(o unDistribuzioni manualiUtilizzo della console)
- *`UpdateService`*(o una modifica alla configurazione utilizzando la console, ad eccezione delle modifiche ai tag)
- *`CreateCustomDomainAssociations`*, *`DeleteCustomDomainAssociations`*

- *`CreateConnection`*, *`DeleteConnection`*

Sospendere e riprendere il servizio utilizzando la console App Runner

Per mettere in pausa il servizio utilizzando la console App Runner

1. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionaServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica sul servizio: .
3. ScegliereOperazioni, quindi sceglierePause (Metti in pausa): .

   Nella pagina del dashboard del servizio, il servizioStatoModifiches inOperazione in corsoe quindi cambia inIn pausa: . Il tuo servizio è ora in pausa.

Per riprendere il servizio utilizzando la console App Runner

1. ScegliereOperazioni, quindi scegliereRiprendi: .

   Nella pagina del dashboard del servizio, il servizioStatoModifiches inOperazione in corso: .
2. Attendere che il servizio riprenda. Nella pagina del dashboard del servizio, il servizioStatotorna aIn esecuzione: .
3. Per verificare che la ripresa del servizio abbia esito positivo, nella pagina del dashboard del servizio scegliere la casella di controlloDominio App Runnervalue. È l'URL del sito web del tuo servizio. Verificare che l'applicazione Web sia eseguita correttamente.

Metti in pausa e riprendi il tuo servizio utilizzando l'API App Runner oAWS CLI

Per mettere in pausa il servizio utilizzando l'API App Runner oAWS CLI, chiama ilPausServiceOperazione API. Se la chiamata restituisce una risposta riuscita con unService (Servizio)oggetto che mostra`"Status": "OPERATION_IN_PROGRESS"`, App Runner inizia a mettere in pausa il servizio.

Per riprendere il servizio utilizzando l'API App Runner oAWS CLI, chiama ilCurriculum ServiceOperazione API. Se la chiamata restituisce una risposta riuscita con unService

(Servizio)oggetto che mostra`"Status": "OPERATION_IN_PROGRESS"`, App Runner inizia a riprendere il servizio.

Eliminazione di un servizio App Runner

Quando si desidera terminare l'applicazione Web in esecuzione nelAWS App RunnerPuoi eliminare il servizio. L'eliminazione di un servizio interrompe il servizio Web in esecuzione, rimuove le risorse sottostanti ed elimina i dati associati.

Potresti voler eliminare un servizio App Runner per uno o più dei seguenti motivi:

- Non è più necessario l'applicazione web— Ad esempio, è ritirato o è una versione di sviluppo che hai finito di usare.
- Hai raggiunto la quota del servizio di App Runner— Desideri creare un nuovo servizio nello stessoRegione AWSe hai raggiunto la quota associata al tuo account. Per ulteriori informazioni, consulta the section called “Quote di risorse App Runner”.
- Considerazioni sulla sicurezza o sulla privacy: vuoi che App Runner elimini i dati archiviati per il tuo servizio.

Sospensione rispetto all'eliminazione

Pause (Metti in pausa)il servizio App Runner pertemporaneamentedisabilitarlo. Solo le risorse di calcolo vengono terminate e i dati archiviati (ad esempio, l'immagine contenitore con la versione dell'applicazione) rimangono intatti. La ripresa del servizio è rapida: l'applicazione è pronta per essere distribuita su nuove risorse di elaborazione. Il dominio App Runner rimane lo stesso.

Eliminail servizio App Runner perpermanentementeRimuoverlo. I dati memorizzati vengono cancellati. Se è necessario ricreare il servizio, App Runner deve recuperare nuovamente la fonte e anche crearla se si tratta di un repository di codice. L'applicazione Web ottiene un nuovo dominio App Runner.

Che cosa elimina App Runner?

Quando elimini il servizio, App Runner elimina alcuni elementi associati e non ne elimina altri. Gli elenchi seguenti forniscono i dettagli.

Elementi eliminati da App Runner:

- Immagine container— Una copia dell'immagine distribuita o dell'immagine creata da App Runner dal codice sorgente. È memorizzato in Amazon Elastic Container Registry (Amazon ECR) utilizzando l'Account AWSdi proprietà di App Runner.
- Configurazione del servizio— Le impostazioni di configurazione associate al servizio App Runner. Sono memorizzati in Amazon DynamoDB utilizzando l'Account AWSdi proprietà di App Runner.

Elementi che App Runner non elimina:

- Connessione— È possibile che si disponga di una connessione associata al servizio. Una connessione App Runner è una risorsa separata che potrebbe essere condivisa tra diversi servizi App Runner. Se la connessione non serve più, è possibile eliminarlo esplicitamente. Per ulteriori informazioni, consulta the section called “Connessioni”.
- Certificati di dominio personalizzati: se si collegano domini personalizzati a un servizio App Runner, App Runner crea internamente certificati che tengono traccia della validità del dominio. Sono memorizzati inAWS Certificate Manager(ACM). App Runner non elimina il certificato per sette giorni dopo che un dominio è stato scollegato dal servizio o dopo l'eliminazione del servizio. Per ulteriori informazioni, consulta the section called “Nomi di dominio personalizzati”.

Eliminare il servizio utilizzando la console App Runner

Per eliminare il servizio utilizzando la console App Runner

1. Apertura dellaConsole App Runner, e nelRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionaServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica del servizio: .
3. Scegliere Actions (Operazioni), quindi Delete (Elimina).

   La console passerà automaticamente allaServizi(Certificato creato). Il servizio eliminato visualizza ilOperazione in corsoe quindi il servizio scompare dall'elenco. Il tuo servizio è ora eliminato.

Eliminare il servizio utilizzando l'API App Runner oAWS CLI

Per eliminare il servizio utilizzando l'API App Runner oAWS CLIChiamata delDeleteServiceOperazione API. Se la chiamata restituisce una risposta riuscita con unService (Servizio)oggetto visualizzazione`"Status": "OPERATION_IN_PROGRESS"`, App Runner inizia a eliminare il servizio.

Registrazione e monitoraggio per il tuo servizio App Runner

AWS App Runnersi integra con diversiAWSper fornirti una vasta suite di strumenti di registrazione e monitoraggio per il tuo servizio App Runner. Argomenti di questo capitolo descrivono queste funzionalità.

Argomenti

Monitoraggio dell'attività del servizio App

AWS App Runnerutilizza un elenco di operazioni per tenere traccia dell'attività nel servizio App Runner. Un'operazione rappresenta una chiamata asincrona a un'azione API, ad esempio la creazione di un servizio, l'aggiornamento di una configurazione e la distribuzione di un servizio. Le sezioni che seguono illustrano come monitorare l'attività nella console App Runner e l'utilizzo dell'API di.

Monitoraggio dell'attività del servizio App Runner nella console

La console App Runner visualizza l'attività del servizio App Runner e fornisce più modi per esplorare le operazioni.

Per visualizzare l'attività del servizio

1. Apertura dellaConsole App Runner, e nellaRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, scegliereServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica del servizio: .
3. Nella pagina di controllo del servizio, scegliere l'opzioneAttivitàSe non è già stato scelto.

   La console di visualizza un elenco di operazioni.

4. Per trovare operazioni specifiche, eseguire l'ambito nell'elenco immettendo un termine di ricerca. È possibile cercare qualsiasi valore visualizzato nella tabella.
5. Scegliere qualsiasi operazione elencata per visualizzare o scaricare il registro correlato.

Recupero delle operazioni del servizio App Runner utilizzando l'API App Runner oAWS CLI

LaListOperationsL'azione, dato l'Amazon Resource Name (ARN) di un servizio App Runner, restituisce un elenco di operazioni che si sono verificate su questo servizio. Ogni voce dell'elenco contiene un ID operazione e alcuni dettagli di tracciabilità.

Visualizzazione dei log di App Runner trasmessi in streaming ai CloudWatch Logs

È possibile utilizzare Amazon CloudWatch Logs per monitorare, archiviare e accedere ai file di log che le risorse sono disponibili in variAWSi servizi generano. Per ulteriori informazioni, consultaAmazon CloudWatch Logs Guida all'utente: .

AWS App Runnerraccoglie l'output delle distribuzioni delle applicazioni e del servizio attivo e lo trasmette ai CloudWatch Logs. Nelle sezioni seguenti sono elencati i flussi di log di App Runner e viene illustrato come visualizzarli nella console di App Runner.

Flussi e gruppi di log di App Runner

CloudWatch Logs mantiene i dati di log nei flussi di log che organizza ulteriormente nei gruppi di log. AFlusso di logÈ una sequenza di eventi di log da una origine specifica. Un gruppo di log è un gruppo di flussi di log che condividono le stesse impostazioni di conservazione, monitoraggio e controllo degli accessi.

App Runner definisce due gruppi di log CloudWatch Logs, ciascuno con più flussi di log, per ciascuno dei servizi App Runner nelAccount AWS: .

Log dei servizi

Il gruppo di log del servizio contiene l'output di registrazione generato da App Runner mentre gestisce il servizio App Runner e agisce su di esso.

nome gruppo di log	Esempio
`/aws/apprunner/` *`service-name`* `/` *`service-id`* `/service`	`/aws/apprunner/python-test/ac7ec8b51ff34746bcb6654e0bcb23da/service`

All'interno del gruppo di log del servizio, App Runner crea un flusso di registro eventi per acquisire l'attività nel ciclo di vita del servizio App Runner. Ad esempio, questo potrebbe essere l'avvio dell'applicazione o la sospensione.

Inoltre, App Runner crea un flusso di log per ogni operazione asincrona a esecuzione prolungata correlata al servizio. Il nome del flusso di registro riflette il tipo di operazione e l'ID di operazione specifico.

Adistribuzioneè un tipo di operazione. I registri di distribuzione contengono l'output di registrazione dei passaggi di compilazione e distribuzione eseguiti da App Runner quando si crea un servizio o si distribuisce una nuova versione dell'applicazione. I nomi dei flussi di log di distribuzione iniziano con`deployment/`e terminare con l'ID dell'operazione che esegue la distribuzione. Questa operazione è unCreateServiceper la distribuzione iniziale dell'applicazione o unStartDeploymentper ogni ulteriore distribuzione.

All'interno di un registro di distribuzione, ogni messaggio di registro inizia con un prefisso:

- `[AppRunner]`: output generato da App Runner durante la distribuzione.
- `[Build]`— Output dei propri script di compilazione.

Nome del flusso di log	Esempio
`events`	N/D (nome fisso)
`operation-type` `/` *`operation-id`*	`deployment/c2c8eeedea164f459cf78f12a8953390`

Log di applicazioni

Il gruppo di log dell'applicazione contiene l'output del codice dell'applicazione in esecuzione.

nome gruppo di log	Esempio
`/aws/apprunner/ service-name /service-id /application`	`/aws/apprunner/python-test/ ac7ec8b51ff34746bcb6654e0bcb23da/ application`

All'interno del gruppo di log dell'applicazione, App Runner crea un flusso di log per ogni istanza (unità di ridimensionamento) che esegue l'applicazione.

Nome del flusso di log	Esempio
`instance/ instance-id`	`instance/1a80bc9134a84699b7 b3432ebeebb591`

Visualizzazione dei log di App Runner nella console

La console App Runner visualizza un riepilogo di tutti i registri del servizio e consente di visualizzarli, esplorarli e scaricarli.

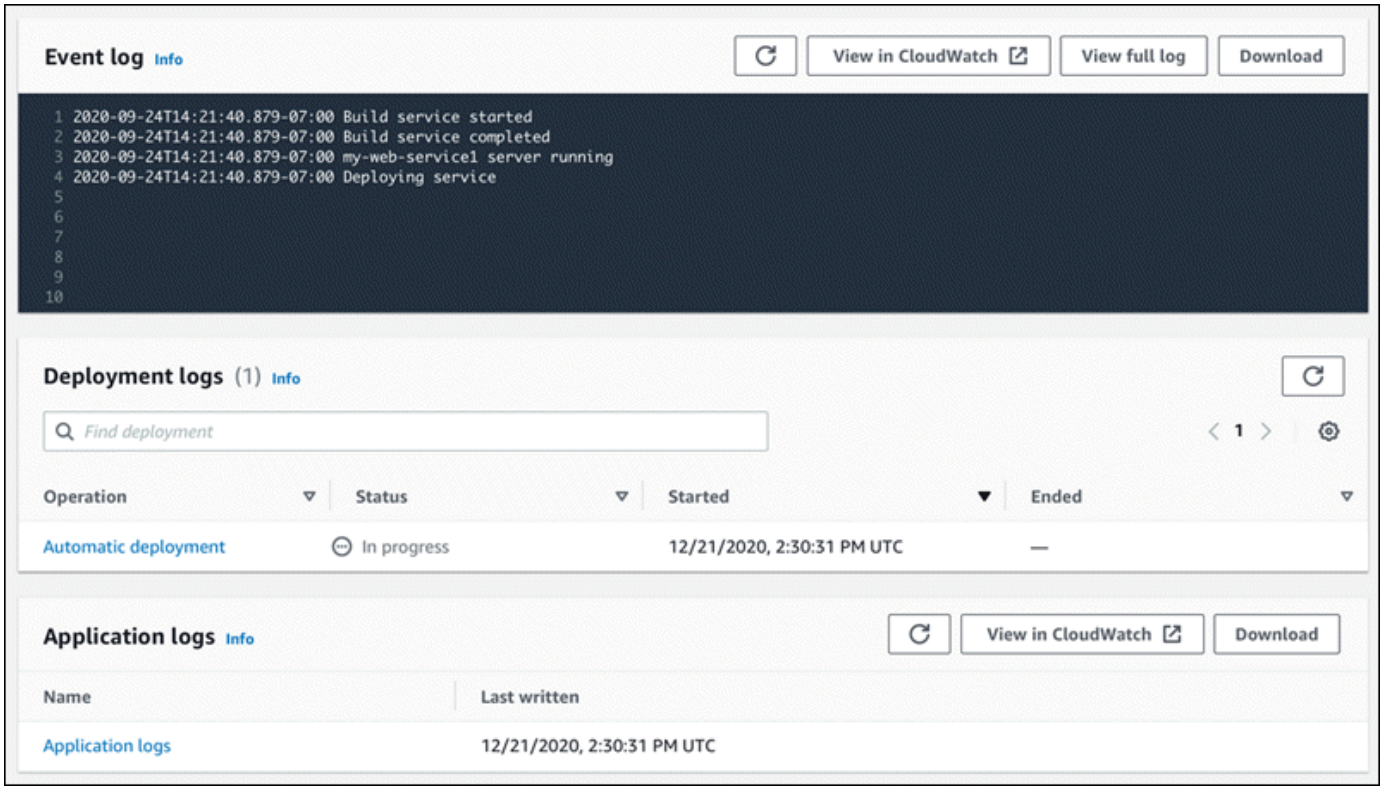
Per visualizzare i log per il servizio

1. Apertura dellaConsole di App Runner, e nelRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, scegliereServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica dei servizi: .
3. Nella pagina di controllo dei servizi, scegli la schedaLogScheda.

   La console visualizza alcuni tipi di log in diverse sezioni:

   - Log degli eventi— Attività nel ciclo di vita del servizio App Runner. La console mostra gli eventi più recenti.
   - Registri di distribuzione— Distribuzioni del repository di origine nel servizio App Runner. La console visualizza un flusso di log separato per ogni distribuzione.
   - Log di applicazioni— L'output dell'applicazione Web distribuita nel servizio App Runner. La console combina l'output di tutte le istanze in esecuzione in un unico flusso di log.



4. Per trovare distribuzioni specifiche, eseguire l'ambito nell'elenco dei log di distribuzione immettendo un termine di ricerca. È possibile cercare qualsiasi valore visualizzato nella tabella.
5. Per visualizzare il contenuto di un registro, scegliereVisualizzare il log completo(registro eventi) o il nome del flusso di registro (log di distribuzione e applicazioni).
6. ScegliereScaricaper scaricare un registro. Per un flusso di log di distribuzione, selezionare prima un flusso di log.
7. ScegliereVisualizza in CloudWatchper aprire la console CloudWatch e utilizzare tutte le sue funzionalità per esplorare i log del servizio App Runner. Per un flusso di log di distribuzione, selezionare prima un flusso di log.

> **Note**
>
> La console CloudWatch è particolarmente utile se si desidera visualizzare i registri delle applicazioni di istanze specifiche anziché il registro applicazioni combinato.

Visualizzazione delle metriche del servizio App Runner segnalate a CloudWatch

Amazon CloudWatch monitora i Amazon Web Services (AWS) e le applicazioni eseguite suAWSIn tempo reale. Puoi utilizzare CloudWatch per raccogliere e tenere traccia dei parametri, che sono delle variabili che si possono misurare per le risorse e le applicazioni. Puoi anche utilizzarlo per creare allarmi che guardano i parametri. Quando viene raggiunta una determinata soglia, CloudWatch invia notifiche o apporta automaticamente modifiche alle risorse monitorate. Per ulteriori informazioni, consultaGuida per l'utente di Amazon CloudWatch: .

AWS App Runnerraccoglie una varietà di metriche che forniscono una maggiore visibilità sull'utilizzo, le prestazioni e la disponibilità dei servizi App Runner. Alcune metriche tengono traccia delle singole istanze che eseguono il servizio Web, mentre altre si trovano al livello di servizio generale. Nelle sezioni seguenti vengono elencate le metriche di App Runner e viene illustrato come visualizzarle nella console App Runner.

Metriche di App Runner

App Runner raccoglie le seguenti metriche relative al servizio dell'utente e le pubblica su CloudWatch nella`AWS/AppRunner`Lo spazio dei nomi.

Metriche a livello di istanzavengono raccolti singolarmente per ogni istanza (unità di ridimensionamento).

Cosa viene misurato?	Parametro	Descrizione
CPU utilization	`CPUUtilization`	L'utilizzo medio della CPU durante periodi di un minuto.
Memory utilizati on	`MemoryUti lization`	L'utilizzo medio della memoria durante i periodi di un minuto.

Parametri sul livello di servizioVengono raccolti per l'intero servizio.

Cosa viene misurato?	Metrics	Descrizione
HTTP request count	`Requests`	Il numero di richieste HTTP ricevute dal servizio.

HTTP status counts	`2xxStatus Responses`
	
	`4xxStatus Responses`
	
	`5xxStatus Responses`
		Il numero di richieste HTTP che hanno restituito lo stato di risposta, raggruppato per categoria (2XX, 4XX, 5XX).

HTTP request latency	`RequestLatency`	Il tempo impiegato dal servizio Web per elaborare le richieste HTTP.
Instance counts	`ActiveInstances`	Il numero di istanze che elaborano le richieste HTTP per il servizio.

Visualizzazione dei parametri di App Runner nella console

La console App Runner visualizza graficamente le metriche raccolte da App Runner per il tuo servizio e fornisce altri modi per esplorarle.

> **Note**
>
> In questo momento, la console visualizza solo le metriche del servizio. Per visualizzare le metriche delle istanze, utilizzare la console CloudWatch.

Per visualizzare i log per il servizio

1. Apertura dellaConsole Runner di app, e nelRegioni, seleziona il tuoRegione AWS: .
2. Nel riquadro di navigazione, selezionaServizi, quindi scegli il tuo servizio App Runner.

   La console visualizza il dashboard del servizio con unPanoramica del servizio: .

3. Nella pagina di controllo del servizio, seleziona la casella di controlloParametri diScheda.

   La console visualizza una serie di grafici delle metriche.

4. Scegliere una durata (ad esempio12h) per applicare i grafici delle metriche al periodo recente di tale durata.

5. ScegliereAggiungi a pannello di controllonella parte superiore di una delle sezioni del grafico, o utilizzare il menu su qualsiasi grafico, per aggiungere le metriche pertinenti a un dashboard nella console CloudWatch per ulteriori indagini.

Gestione degli eventi App Runner in EventBridge

Utilizzando Amazon EventBridge, puoi impostare regole basate sugli eventi che monitorano un flusso di dati in tempo reale dal tuoAWS App Runnerservizio per determinati modelli. Quando un modello per una regola viene abbinato, EventBridge avvia un'azione in una destinazione comeAWS Lambda, Amazon ECS,AWS Batche Amazon SNS. Ad esempio, puoi impostare una regola per l'invio di notifiche e-mail segnalando un argomento Amazon SNS ogni volta che una distribuzione al servizio ha esito negativo. In alternativa, è possibile impostare una funzione Lambda per notificare un canale Slack ogni volta che un aggiornamento del servizio ha esito negativo. Per ulteriori informazioni su EventBridge, consultaGuida per l'utente di Amazon EventBridge: .

App Runner invia i seguenti tipi di evento a EventBridge

- Modifica dello stato del servizio— Una modifica dello stato di un servizio App Runner. Ad esempio, uno stato del servizio è stato modificato in`DELETE_FAILED`: .
- Modifica dello stato delle operazioni dei servizi— Una modifica dello stato di un'operazione lunga e asincrona su un servizio App Runner. Ad esempio, un servizio avviato per la creazione, un aggiornamento del servizio completato o una distribuzione del servizio completata con errori.

Creazione di una regola EventBridge per agire sugli eventi App Runner

EventBridgeEvento .è un oggetto che definisce alcuni campi EventBridge standard, come l'origineAWSe il tipo di dettaglio (evento) e un insieme di campi specifici dell'evento con i dettagli dell'evento. Per creare una regola EventBridge, utilizzare la console EventBridge per definire unModello di eventi(quali eventi dovrebbero puntare tracciati) e specificare unazione target(cosa dovrebbe essere fatto su una partita). Un modello di evento è simile agli eventi che corrisponde. È possibile specificare un sottoinsieme di campi da abbinare e, per ogni campo, specificare un elenco

di valori possibili. In questo argomento vengono forniti esempi di eventi e modelli di eventi di App Runner.

Per ulteriori informazioni sulla creazione di regole EventBridge, consultaCreazione di una regola per un oggettoAWSservizionellaGuida per l'utente di Amazon EventBridge: .

> **Note**
>
> Alcuni servizi supportanoModelli predefinitiin EventBridge. Ciò semplifica il modo in cui viene creato un modello di evento. È possibile selezionare i valori di campo in un modulo e EventBridge genera automaticamente il motivo. Al momento, App Runner non supporta modelli predefiniti. Devi inserire il modello come oggetto JSON. È possibile utilizzare gli esempi riportati in questo argomento come punto di partenza.

Esempi di eventi App Runner

Questi sono alcuni esempi di eventi che App Runner invia a EventBridge.

- Evento di modifica dello stato del servizio. In particolare, un servizio che è cambiato dal`OPERATION_IN_PROGRESS`al`RUNNING`Stato.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "Service Status Change",
  "source": "aws.apprunner",
  "account": "111122223333",
  "time": "2021-04-29T11:54:23Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:apprunner:us-east-2:123456789012:service/my-
app/8fe1e10304f84fd2b0df550fe98a71fa"
  ],
  "detail": {
    "PreviousStatus": "OPERATION_IN_PROGRESS",
    "CurrentStatus": "RUNNING",
    "ServiceName": "my-app",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "Message": "Service status is set to RUNNING.",
    "Severity": "INFO"
```

```
  }
}
```

- Evento di modifica dello stato dell'operazione. Nello specifico, un`UpdateService`operazione completata correttamente.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "Service Operation Status Change",
  "source": "aws.apprunner",
  "account": "111122223333",
  "time": "2021-04-29T18:43:48Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:apprunner:us-east-2:123456789012:service/my-
app/8fe1e10304f84fd2b0df550fe98a71fa"
  ],
  "detail": {
    "Status": "UpdateServiceCompletedSuccessfully",
    "ServiceName": "my-app",
    "ServiceId": "8fe1e10304f84fd2b0df550fe98a71fa",
    "Message": "Service update completed successfully. New application and
 configuration is deployed.",
    "Severity": "INFO"
  }
}
```

Esempi di pattern di eventi di App

Negli esempi seguenti vengono illustrati i modelli di eventi che è possibile utilizzare nelle regole EventBridge per abbinare uno o più eventi App Runner. Un modello di evento è simile a un evento. Includere solo i campi che si desidera abbinare e fornire un elenco anziché uno scalare a ciascuno di essi.

- Corrispondenza di tutti gli eventi di modifica dello stato del servizio per i servizi di un account specifico, in cui il servizio non è più in`RUNNING`Stato.

```
{
  "detail-type": [ "Service Status Change" ],
  "source": [ "aws.apprunner" ],
```

```
  "account": [ "111122223333" ],
  "detail": {
    "PreviousStatus": [ "RUNNING" ]
  }
}
```

- Corrisponde a tutti gli eventi di modifica dello stato dell'operazione per i servizi di un account specifico, in cui l'operazione non è riuscita.

```
{
  "detail-type": [ "Service Operation Status Change" ],
  "source": [ "aws.apprunner" ],
  "account": [ "111122223333" ],
  "detail": {
    "Status": [
      "CreateServiceFailed",
      "DeleteServiceFailed",
      "UpdateServiceFailed",
      "DeploymentFailed",
      "PauseServiceFailed",
      "ResumeServiceFailed"
    ]
  }
}
```

Riferimento all'evento App Runner

Modifica dello stato del servizio

Un evento di modifica dello stato del servizio ha`detail-type`impostato su.`Service Status Change`: . Contiene i seguenti campi e valori dettagliati:

```
"PreviousStatus": "any valid service status",
"CurrentStatus": "any valid service status",
"ServiceName": "your service name",
"ServiceId": "your service ID",
"Message": "Service status is set to CurrentStatus.",
"Severity": "varies"
```

Modifica dello stato delle operazioni

Un evento di modifica dello stato dell'operazione ha`detail-type`impostato su.`Service Operation Status Change`: . Contiene i seguenti campi e valori dettagliati:

```
"Status": "see following table",
"ServiceName": "your service name",
"ServiceId": "your service ID",
"Message": "see following table",
"Severity": "varies"
```

Nella tabella seguente sono elencati tutti i possibili codici di stato e i messaggi correlati.

Stato	Message
`CreateServiceStarted`	Creazione del servizio avviata.
`CreateServiceCompletedSuccessfully`	Creazione del servizio completata correttamente.
`CreateServiceFailed`	Creazione del servizio non riuscita. Per informazioni dettagliate, consulta i registri dei servizi.
`DeleteServiceStarted`	Eliminazione del servizio avviata.
`DeleteServiceCompletedSuccessfully`	Eliminazione del servizio completata correttamente.
`DeleteServiceFailed`	Eliminazione del servizio non riuscita.
`UpdateServiceStarted`	
`UpdateServiceCompletedSuccessfully`	Aggiornamento del servizio completato correttamente. Viene distribuita una nuova applicazione e configurazione.
	Aggiornamento del servizio completato correttamente. Viene distribuita una nuova configurazione.
`UpdateServiceFailed`	Aggiornamento del servizio non riuscito. Per informazioni dettagliate, consulta i registri dei servizi.

Stato	Message
`DeploymentStarted`	Distribuzione avviata.
`DeploymentCompletedSuccessfully`	La tua distribuzione ha esito positivo.
`DeploymentFailed`	La distribuzione non è riuscita Per informazioni dettagliate, consulta i registri dei servizi.
`PauseServiceStarted`	Pausa del servizio avviata.
`PauseServiceCompletedSuccessfully`	Pausa del servizio completata correttamente.
`PauseServiceFailed`	Pausa del servizio non riuscita.
`ResumeServiceStarted`	Curriculum del servizio avviato.
`ResumeServiceCompletedSuccessfully`	Ripristino del servizio completato correttamente.
`ResumeServiceFailed`	Ripresa del servizio non riuscita.

Registrazione delle chiamate API di App Runner conAWS CloudTrail

App Runner è integrato conAWS CloudTrail, un servizio che fornisce un record di operazioni eseguite da un utente, un ruolo o unAWSin App Runner. CloudTrail acquisisce tutte le chiamate API per App Runner come eventi. Le chiamate acquisite includono le chiamate dalla console App Runner e le chiamate di codice alle operazioni API di App Runner. Se crei un trail, puoi abilitare la distribuzione continua di eventi CloudTrail in un bucket Amazon S3, inclusi gli eventi per App Runner. Se invece non configuri un trail, puoi comunque visualizzare gli eventi più recenti nella console CloudTrail nella cronologia degli eventi. Le informazioni raccolte da CloudTrail consentono di determinare la richiesta effettuata ad App Runner, l'indirizzo IP da cui è partita la richiesta, l'autore della richiesta, il momento in cui è stata eseguita e altri dettagli.

Per ulteriori informazioni su CloudTrail, consulta laAWS CloudTrailGuida per l'utente: .

Informazioni su App Runner in CloudTrail

CloudTrail è abilitato sulAccount AWSQuando crei l'account. Quando si verifica un'attività in App Runner, questa viene registrata in un evento CloudTrail insieme ad altriAWSEventi di servizio inCronologia degli eventi: . Puoi visualizzare, cercare e scaricare gli eventi recenti nell'Account AWS: . Per ulteriori informazioni, consulta Visualizzazione di eventi nella cronologia degli eventi di CloudTrail.

Per una registrazione continuativa di attività ed eventi nell'Account AWS, inclusi gli eventi per App Runner, crea un trail. Un trail consente a CloudTrail di distribuire i file di log in un bucket Amazon S3. Per impostazione di default, quando crei un trail nella console, il trail sarà valido in tutti iRegioni AWS: . Il trail registra gli eventi provenienti da tutte le regioni nellaAWSDistribuisce i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altriAWSPer analizzare con maggiore dettaglio e usare i dati raccolti nei log CloudTrail. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un trail
- Servizi e integrazioni CloudTrail supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di log CloudTrail da più regioni e Ricezione di file di log CloudTrail da più account

Tutte le azioni di App Runner vengono registrate da CloudTrail e sono documentate inAWS App RunnerRiferimento alle API. Ad esempio, le chiamate alle operazioni `CreateService`, `DeleteConnection` e `StartDeployment` generano voci nei file di log di 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:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro servizio AWS.

Per ulteriori informazioni, consulta Elemento userIdentity di CloudTrail.

Informazioni sulle voci dei file di log di di App

Un trail è una configurazione che consente la distribuzione di eventi come i file di log in un bucket Amazon S3 che specifichi. I file di log di CloudTrail possono contenere una o più voci di log. Un evento rappresenta una singola richiesta inviata da un'origine e include informazioni sull'operazione richiesta, data e ora dell'operazione e parametri richiesti. I file di log CloudTrail non sono una traccia di stack ordinata delle chiamate API pubbliche e di conseguenza non devono apparire in base a un ordine specifico.

L'esempio seguente mostra una voce di log di CloudTrail che illustra l'operazione `CreateService`.

> **Note**
>
> Per motivi di sicurezza, alcuni valori di proprietà vengono redatti nei registri e sostituiti con il testo`HIDDEN_DUE_TO_SECURITY_REASONS`: . Ciò impedisce l'esposizione involontaria di informazioni segrete. Tuttavia, è ancora possibile vedere che queste proprietà sono state passate nella richiesta o restituite nella risposta.

Esempio di voce di log CloudTrail per il**`CreateService`**Runner app, azione

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/aws-user",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "aws-user",
  },
  "eventTime": "2020-10-02T23:25:33Z",
  "eventSource": "apprunner.amazonaws.com",
  "eventName": "CreateService",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/77.0.3865.75 Safari/537.36",
  "requestParameters": {
    "serviceName": "python-test",
    "sourceConfiguration": {
```

```
      "codeRepository": {
        "repositoryUrl": "https://github.com/github-user/python-hello",
        "sourceCodeVersion": {
          "type": "BRANCH",
          "value": "main"
        },
        "codeConfiguration": {
          "configurationSource": "API",
          "codeConfigurationValues": {
            "runtime": "python3",
            "buildCommand": "HIDDEN_DUE_TO_SECURITY_REASONS",
            "startCommand": "HIDDEN_DUE_TO_SECURITY_REASONS",
            "port": "8080",
            "runtimeEnvironmentVariables": "HIDDEN_DUE_TO_SECURITY_REASONS"
          }
        }
      },
      "autoDeploymentsEnabled": true,
      "authenticationConfiguration": {
        "connectionArn": "arn:aws:apprunner:us-east-2:123456789012:connection/your-
connection/e7656250f67242d7819feade6800f59e"
      }
    },
    "healthCheckConfiguration": {
      "protocol": "HTTP"
    },
    "instanceConfiguration": {
      "cpu": "256",
      "memory": "1024"
    }
  },
  "responseElements": {
    "service": {
      "serviceName": "python-test",
      "serviceId": "dfa2b7cc7bcb4b6fa6c1f0f4efff988a",
      "serviceArn": "arn:aws:apprunner:us-east-2:123456789012:service/python-test/
dfa2b7cc7bcb4b6fa6c1f0f4efff988a",
      "serviceUrl": "generated domain",
      "createdAt": "2020-10-02T23:25:32.650Z",
      "updatedAt": "2020-10-02T23:25:32.650Z",
      "status": "OPERATION_IN_PROGRESS",
      "sourceConfiguration": {
        "codeRepository": {
          "repositoryUrl": "https://github.com/github-user/python-hello",
```

```
          "sourceCodeVersion": {
            "type": "Branch",
            "value": "main"
          },
          "codeConfiguration": {
            "codeConfigurationValues": {
            "configurationSource": "API",
              "runtime": "python3",
              "buildCommand": "HIDDEN_DUE_TO_SECURITY_REASONS",
              "startCommand": "HIDDEN_DUE_TO_SECURITY_REASONS",
              "port": "8080",
              "runtimeEnvironmentVariables": "HIDDEN_DUE_TO_SECURITY_REASONS"
            }
          }
        },
        "autoDeploymentsEnabled": true,
        "authenticationConfiguration": {
          "connectionArn": "arn:aws:apprunner:us-east-2:123456789012:connection/your-
connection/e7656250f67242d7819feade6800f59e"
        }
      },
      "healthCheckConfiguration": {
        "protocol": "HTTP",
        "path": "/",
        "interval": 5,
        "timeout": 2,
        "healthyThreshold": 3,
        "unhealthyThreshold": 5
      },
      "instanceConfiguration": {
        "cpu": "256",
        "memory": "1024"
      },
      "autoScalingConfigurationSummary": {
        "autoScalingConfigurationArn": "arn:aws:apprunner:us-
east-2:123456789012:autoscalingconfiguration/
DefaultConfiguration/1/00000000000000000000000000000001",
        "autoScalingConfigurationName": "DefaultConfiguration",
        "autoScalingConfigurationRevision": 1
      }
    }
  },
  "requestID": "1a60af60-ecf5-4280-aa8f-64538319ba0a",
  "eventID": "e1a3f623-4d24-4390-a70b-bf08a0e24669",
```

```
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

Impostazione delle opzioni del servizio App Runner utilizzando un file di configurazione

> **Note**
>
> I file di configurazione sono applicabili solo aservizi basati sul codice sorgente: . Non è possibile utilizzare i file di configurazione conservizi basati su immagini: .

Quando si crea un fileAWS App Runnerservizio utilizzando un repository del codice sorgente,AWS App Runnerrichiede informazioni sulla creazione e l'avvio del servizio. È possibile fornire queste informazioni ogni volta che si crea un servizio utilizzando la console App Runner o l'API. In alternativa, puoi impostare le opzioni di servizio utilizzando un fileFile di configurazione: . Le opzioni specificate in un file diventano parte del repository di origine e le eventuali modifiche apportate a queste opzioni vengono monitorate in modo simile a come vengono registrate le modifiche apportate al codice sorgente. È possibile utilizzare il file di configurazione di App Runner per specificare più opzioni rispetto a quelle supportate dall'API. Non è necessario fornire un file di configurazione se sono necessarie solo le opzioni di base supportate dall'API.

Il file di configurazione di App Runner è un file YAML denominato`apprunner.yaml`Nella directory radice del repository dell'applicazione. Fornisce opzioni di compilazione e runtime per il servizio. I valori in questo file istruiscono App Runner come creare e avviare il servizio e forniscono contesto di runtime, ad esempio le impostazioni di rete e le variabili di ambiente.

Il file di configurazione di App Runner non include impostazioni operative, come CPU e memoria.

Per gli esempi di file di configurazione di App Runner, consultarethe section called “Esempi”: . Per una guida di riferimento completa, vederethe section called “Riferimento”: .

Argomenti

- Esempi di file di configurazione di App
- Informazioni di riferimento sul file di configurazione App

Esempi di file di configurazione di App

> **Note**
>
> I file di configurazione sono applicabili solo aservizi basati sul codice sorgente: . Non è possibile utilizzare i file di configurazione conservizi basati su immagini: .

I seguenti esempi mostranoAWS App RunnerFile di configurazione. Alcuni sono minimi e contengono solo impostazioni richieste. Altri sono completi, incluse tutte le sezioni dei file di configurazione.
Per una panoramica dei file di configurazione di App Runner, consultaFile di configurazione di App Runner: .

Esempio di file di configurazione

File di configurazione minimo

Con un file di configurazione minimo, App Runner fa le seguenti ipotesi:

- Durante la compilazione o l'esecuzione non sono necessarie variabili di ambiente personalizzate.
- Viene utilizzata la versione più recente di runtime.
- Vengono utilizzati il numero di porta predefinito e la variabile di ambiente della porta.

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    build:
      - pip install pipenv
      - pipenv install
run:
  command: python app.py
```

File completi di configurazione

In questo esempio viene illustrato l'utilizzo di tutte le chiavi di configurazione con un runtime gestito.

Example apprunner.yaml

```
version: 1.0
runtime: python3
build:
  commands:
    pre-build:
      - wget -c https://s3.amazonaws.com/DOC-EXAMPLE-BUCKET/test-lib.tar.gz -O - | tar
 -xz
    build:
      - pip install pipenv
      - pipenv install
    post-build:
      - python manage.py test
  env:
    - name: DJANGO_SETTINGS_MODULE
      value: "django_apprunner.settings"
    - name: MY_VAR_EXAMPLE
      value: "example"
run:
  runtime-version: 3.7.7
  command: pipenv run gunicorn django_apprunner.wsgi --log-file -
  network:
    port: 8000
    env: MY_APP_PORT
  env:
    - name: MY_VAR_EXAMPLE
      value: "example"
```

Per esempi di file di configurazione di runtime gestiti specifici, vedere l'argomento secondario relativo al runtime specifico inServizio basato su codice in: .

Informazioni di riferimento sul file di configurazione App

> **Note**
>
> I file di configurazione sono applicabili solo aservizi basati sul codice sorgente: . Non è possibile utilizzare i file di configurazione conservizi basati su immagini: .

Questo argomento è una guida di riferimento completa alla sintassi e alla semantica di unAWS App RunnerFile di configurazione. Per una panoramica dei file di configurazione di App Runner, consultaFile di configurazione di App Runner: .

Il file di configurazione di App Runner è un file YAML. Denominarlo`apprunner.yaml`e posizionalo nella directory radice del repository dell'applicazione.

Panoramica sulla struttura

Il file di configurazione di App Runner è un file YAML. Denominarlo`apprunner.yaml`e posizionalo nella directory radice del repository dell'applicazione.

Il file di configurazione di App Runner contiene le seguenti parti principali:

- Sezione superiore— Contiene tasti di primo livello
- Sezione Build— Configura la fase di compilazione
- Sezione Esegui— Configura la fase di runtime

Sezione superiore

Le chiavi nella parte superiore del file forniscono informazioni generali sul file e sul runtime del servizio. Sono disponibili le seguenti chiavi:

- `version`–Campo obbligatorio. Versione del file di configurazione di App Runner. Idealmente, utilizzare la versione più recente.

  Sintassi

  ```
  version: version
  ```

  Example

  ```
  version: 1.0
  ```

- `runtime`–Campo obbligatorio. Il nome del runtime utilizzato dall'applicazione. Sono attualmente disponibili i seguenti runtime:

  `python3`, `nodejs12`

> **Note**
>
> La convenzione di denominazione di un runtime gestito è*<language-name> <major-version>*: .

Sintassi

```
runtime: runtime-name
```

Example

```
runtime: python3
```

Sezione Build

La sezione di compilazione configura la fase di compilazione della distribuzione del servizio App Runner. È possibile specificare comandi di compilazione e variabili di ambiente. I comandi di compilazione sono obbligatori.

La sezione inizia con il`build:`e ha le seguenti sottochiavi:

- `commands`–Campo obbligatorio. Specifica i comandi eseguiti da App Runner durante varie fasi di compilazione. Include le seguenti sottochiavi:
  - `pre-build`–Facoltativo. Comandi eseguiti da App Runner prima della build. Ad esempio, installarenpmdipendenze o librerie di test.
  - `build`–Campo obbligatorio. I comandi eseguiti da App Runner per creare l'applicazione. Ad esempio, utilizzarepipenv: .
  - `post-build`–Facoltativo. Comandi eseguiti da App Runner dopo la build. Ad esempio, puoi utilizzare Maven per creare pacchetti degli artefatti di build in un file JAR o WAR; oppure puoi eseguire un test.

Sintassi

```
build:
  commands:
```

```
    pre-build:
      - command
      - …
    build:
      - command
      - …
    post-build:
      - command
      - …
```

Example

```
build:
  commands:
    pre-build:
      - yum install openssl
    build:
      - pip install -r requirements.txt
    post-build:
      - python manage.py test
```

- env–Facoltativo. Specifica le variabili di ambiente personalizzate per la fase di compilazione. Definito come mapping scalari nome-valore. Puoi fare riferimento a queste variabili per nome nei comandi di compilazione.

  Sintassi

```
build:
  env:
    - name: name1
      value: value1
    - name: name2
      value: value2
    - …
```

  Example

```
build:
  env:
    - name: DJANGO_SETTINGS_MODULE
      value: "django_apprunner.settings"
    - name: MY_VAR_EXAMPLE
```

```
      value: "example"
```

Sezione Esegui

La sezione Esegui configura la fase in esecuzione contenitore della distribuzione dell'applicazione App Runner. È possibile specificare la versione runtime, il comando start, la porta di rete e le variabili di ambiente.

La sezione inizia con il`run:`e ha le seguenti sottochiavi:

- `runtime-version`–Facoltativo. Specifica una versione runtime che si desidera bloccare per il servizio App Runner.

  Per impostazione predefinita, solo la versione principale è bloccata. App Runner utilizza le versioni secondarie e patch più recenti disponibili per il runtime in ogni distribuzione o aggiornamento del servizio. Se si specificano versioni principali e secondarie, entrambe vengono bloccate e App Runner aggiorna solo le versioni delle patch. Se si specificano versioni principali, secondarie e patch, il servizio è bloccato su una versione runtime specifica e App Runner non lo aggiorna mai.

  Sintassi

  ```
  run:
    runtime-version: major[.minor[.patch]]
  ```

  Example

  ```
  runtime: python3
  run:
    runtime-version: 3.7
  ```

- `command`–Campo obbligatorio. Il comando utilizzato da App Runner per eseguire l'applicazione al termine della compilazione dell'applicazione.

  Sintassi

  ```
  run:
    command: command
  ```

- `network`–Facoltativo. Specifica la porta che l'applicazione ascolta. tra cui:

- `port`–Facoltativo. Se specificato, è il numero di porta che l'applicazione è in ascolto. Il valore di default è `8080`.
- `env`–Facoltativo. Se specificato, App Runner passa il numero di porta al contenitore in questa variabile di ambiente, oltre a passare lo stesso numero di porta nella variabile di ambiente predefinita,`PORT`: . In altre parole, se si specifica`env`, App Runner passa il numero di porta in due variabili di ambiente.

  Sintassi

```
run:
  network:
    port: port-number
    env: env-variable-name
```

  Example

```
run:
  network:
    port: 8000
    env: MY_APP_PORT
```

- `env`–Facoltativo. Definizione di variabili di ambiente personalizzate per la fase di esecuzione. Definito come mapping scalari nome-valore. È possibile fare riferimento a queste variabili per nome nell'ambiente di runtime.

  Sintassi

```
run:
  env:
    - name: name1
      value: value1
    - name: name2
      value: value2
    - …
```

  Example

```
run:
  env:
    - name: MY_VAR_EXAMPLE
```

```
      value: "example"
```

L'API App Runner

LaAWS App RunnerAPI (Application Programming Interface) è un'API RESTful per effettuare richieste al servizio App Runner. È possibile utilizzare l'API per creare, elencare, descrivere, aggiornare ed eliminare le risorse di Runner di App nellaAccount AWS: .

È possibile chiamare l'API direttamente nel codice dell'applicazione oppure utilizzare uno deiAWSSDK. Per gli script della riga di comando, utilizzare il comandoAWS CLIper effettuare chiamate al servizio App Runner. Per ulteriori informazioni suAWSstrumenti di sviluppo, vedereStrumenti su cui costruireAWS: .

Per informazioni di riferimento sulle API complete, consulta laAWS App RunnerDocumentazione di riferimento API: .

Sicurezza in App Runner

Per AWS, la sicurezza della cloud ha la massima priorità. In quanto cliente AWS, puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e l'utente. Il modello di responsabilità condivisa descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud–AWSè responsabile della protezione dell'infrastruttura che esegueAWSServizi diCloud AWS: .AWSFornisce, inoltre, servizi che puoi utilizzare in modo sicuro. I revisori di terze parti testano regolarmente e verificano l'efficacia della nostra sicurezza nell'ambito dei Programmi di conformità AWS. Per ulteriori informazioni sui programmi di conformità che si applicano adAWS App RunnerConsultare.AWSServizi contemplati dal programma di compliance: .
- Sicurezza nel cloud— La tua responsabilità è determinata dalAWSche si utilizza. L'utente è anche responsabile per altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e leggi e normative applicabili.

Questa documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si usa App Runner. I seguenti argomenti illustrano come configurare App Runner per soddisfare gli obiettivi di sicurezza e conformità. Viene anche spiegato come utilizzare altriAWSche consentono di monitorare e proteggere le risorse di App Runner.

Argomenti

Protezione dei dati in App Runner

LaAWS Modello di responsabilità condivisaSi applica alla protezione dei dati inAWS App Runner: . Come descritto in questo modello,AWSè responsabile della protezione dell'infrastruttura globale che esegue tutti iAWSNuvola. L'utente è responsabile di mantenere il controllo sui contenuti ospitati su questa infrastruttura. Questo contenuto include la configurazione della protezione e le attività di gestione per l'AWSServizi utilizzati dall'utente. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta laAWSModello di responsabilità condivisa e GDPRPost del blog sullaAWSBlog sulla sicurezza: .

Per garantire la protezione dei dati, si consiglia di proteggereAWSe configurare singoli account utente conAWS Identity and Access Management(IAM). In questo modo, a ogni utente vengono solo assegnate le autorizzazioni necessarie per svolgere il suo lavoro. Ti consigliamo inoltre di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Utilizza SSL/TLS per comunicare con risorse AWS. È consigliabile TLS 1.2 o versioni successive.
- Configura la registrazione dell'API e delle attività degli utenti con AWS CloudTrail.
- Utilizza le soluzioni di crittografia AWS, insieme a tutti i controlli di sicurezza predefiniti all'interno dei servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, ad esempio Amazon Macie, che aiutano a individuare e proteggere i dati personali archiviati in Amazon S3.
- Se si richiedono moduli crittografici convalidati FIPS 140-2 quando si accede ad AWS tramite un'interfaccia a riga di comando o un'API, utilizzare un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

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). Questo include il lavoro con App Runner o altriAWSservizi che utilizzano la console, l'API,AWS CLI, oppureAWSSDK. Gli eventuali dati immessi in App Runner o altri servizi potrebbero essere prelevati per l'inserimento nei log 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 altri argomenti sulla sicurezza di App Runner, consulta.Sicurezza: .

Argomenti

- Protezione dei dati tramite crittografia
- Riservatezza del traffico Internet
- Utilizzo di App Runner con endpoint VPC

Protezione dei dati tramite crittografia

AWS App Runnerlegge l'origine dell'applicazione (immagine di origine o codice sorgente) da un repository specificato e la memorizza per la distribuzione nel servizio. Per ulteriori informazioni, consulta Architettura e concetti.

La protezione dei dati si riferisce alla protezione dei dati mentreIn transito(mentre viaggia da e verso App Runner) eA riposo(mentre è memorizzato inAWScentri dati).

Per ulteriori informazioni sulla protezione dei dati, consulta.the section called “Protezione dei dati”: .

Per altri argomenti sulla sicurezza di App Runner, consulta.Sicurezza: .

Crittografia in transito

È possibile ottenere la protezione dei dati in transito in due modi: crittografare la connessione utilizzando Transport Layer Security (TLS) o utilizzare la crittografia lato client (dove l'oggetto viene crittografato prima dell'invio). Entrambi i metodi sono validi per proteggere i dati dell'applicazione. Per proteggere la connessione, crittografala utilizzando TLS ogni volta che la tua applicazione, i suoi sviluppatori e amministratori e i suoi utenti finali inviano o ricevono oggetti. App Runner configura l'applicazione per ricevere il traffico su TLS.

La crittografia lato client non è un metodo valido per proteggere l'immagine o il codice di origine fornito a App Runner per la distribuzione. App Runner necessita dell'accesso alla sorgente dell'applicazione, pertanto non può essere crittografato. Di conseguenza, assicurati di proteggere la connessione tra l'ambiente di sviluppo o distribuzione e App Runner.

Crittografia degli inattivi e gestione delle chiavi

Per proteggere i dati inattivi della tua applicazione, App Runner crittografa tutte le copie memorizzate dell'immagine di origine dell'applicazione o bundle di origine. Quando crei un servizio App Runner, puoi fornire una chiave master del cliente (CMK). Se ne fornisci uno, App Runner utilizza la chiave fornita per crittografare la fonte. Se non ne fornisci una, App Runner utilizza unaAWSCMK gestita invece.

Per informazioni dettagliate sui parametri di creazione del servizio App Runner, vedereCreateService: . Per informazioni suAWS Key Management Service(AWS KMS), consulta la sezioneAWS Key Management ServiceGuida per gli sviluppatori: .

Riservatezza del traffico Internet

App Runner utilizza Amazon Virtual Private Cloud (Amazon VPC) per creare delimitatori tra le risorse nella tua applicazione App Runner e controllare il traffico tra di loro, la rete locale e Internet. Per ulteriori informazioni sulla sicurezza di Amazon VPC, consulta la sezioneRiservatezza del traffico Internet in Amazon VPCnellaGuida per l'utente di Amazon VPC: .

Per informazioni su come proteggere le richieste ad App Runner utilizzando un endpoint VPC, consulta.the section called “Endpoint VPC”: .

Per ulteriori informazioni sulla protezione dei dati, consulta.the section called “Protezione dei dati”: .

Per altri argomenti sulla sicurezza di App Runner, consulta.Sicurezza: .

Utilizzo di App Runner con endpoint VPC

Il tuoAWSl'applicazione potrebbe integrareAWS App RunnerServizi con altriAWSin esecuzione in unAmazon Virtual Private Cloud (Amazon VPC): . Alcune parti dell'applicazione potrebbero inoltrare richieste a App Runner dall'interno del VPC. È ad esempio possibile utilizzareAWS CodePipelineper la distribuzione continua nel servizio App Runner. Un modo per migliorare la sicurezza della tua applicazione consiste nell'inviare queste richieste di App Runner (e richieste ad altriAWS) su un endpoint VPC.

Un VPC endpoint (Endpoint VPC) consente di connettere privatamente il VPC a servizi AWS supportati e servizi endpoint VPC con tecnologia AWS PrivateLink senza richiedere un Internet gateway, un dispositivo NAT, una connessione VPN o una connessione AWS Direct Connect.

Le risorse presenti nel VPC non utilizzano indirizzi IP pubblici per interagire con le risorse di App Runner. Il traffico tra il VPC e App Runner non lascia la rete Amazon. Per informazioni complete sugli endpoint VPC, consulta.Endpoint VPCnellaAWS PrivateLinkGuida: .

App Runner supportaAWS PrivateLink, che fornisce connettività privata a App Runner ed elimina l'esposizione del traffico a Internet pubblico. Per abilitare l'applicazione a inviare richieste ad App Runner utilizzandoAWS PrivateLink, è possibile configurare un tipo di endpoint VPC noto comeendpoint VPC dell'interfaccia(endpoint dell'interfaccia). Per ulteriori informazioni, consultaEndpoint VPC dell'interfaccia (AWS PrivateLink)nellaAWS PrivateLinkGuida: .

> **Note**
>
> L'applicazione Web nel servizio App Runner viene eseguita in un VPC fornito da App Runner e configurato. Questo VPC è pubblico, è connesso a Internet pubblico. App Runner non supporta l'esecuzione dell'applicazione in un VPC privato o la creazione di un endpoint VPC per esso.

Impostazione di un endpoint VPC per App Runner

Per creare l'endpoint VPC dell'interfaccia per il servizio App Runner nel VPC, segui laCreare un endpoint dell'interfacciaprocedura nella proceduraAWS PrivateLinkGuida: . Per Service Name (Nome del servizio), selezionare `com.amazonaws.region.apprunner`.

Considerazioni sulla privacy della rete VPC

> **Important**
>
> L'utilizzo di un endpoint VPC per App Runner non garantisce che tutto il traffico proveniente dal VPC rimanga fuori da Internet. Il VPC potrebbe essere pubblico (connesso a Internet) e alcune parti della soluzione potrebbero non utilizzare endpoint VPC per rendereAWSChiamate alle API. Ad esempio:AWSServizi potrebbero chiamare altri servizi utilizzando i loro endpoint pubblici. Se è necessaria la privacy del traffico per la soluzione nel VPC, leggere questa sezione.

Per garantire la privacy del traffico di rete nel VPC, considera quanto segue:

- Abilitazione del nome DNS— Alcune parti dell'applicazione potrebbero comunque inviare richieste ad App Runner tramite Internet utilizzando il`apprunner.region.amazonaws.com`Endpoint pubblico. Se il VPC è configurato con accesso a Internet pubblico, queste richieste hanno esito positivo senza alcuna indicazione per l'utente. È possibile evitare questo problema assicurando che Enable DNS name (Abilita nome DNS) sia abilitato durante la creazione dell'endpoint (true per impostazione predefinita). In questo modo viene aggiunta una voce DNS nel VPC che associa l'endpoint del servizio pubblico all'endpoint VPC dell'interfaccia.
- Configura gli endpoint VPC per servizi aggiuntivi— La soluzione potrebbe inviare richieste ad altriAWSServizi . Ad esempio:AWS CodePipelinepotrebbe inviare richieste aAWS CodeBuild: . Configura gli endpoint VPC anche per questi servizi e abilita i nomi DNS su questi endpoint.

Utilizzo dei criteri endpoint per controllare l'accesso con gli endpoint VPC

Per impostazione predefinita, un endpoint VPC consente l'accesso completo al servizio a cui è associato. Quando crei o modifichi un endpoint VPC per App Runner, puoi allegare unaCrittografia endpoint.

Un criterio endpoint è unAWS Identity and Access ManagementPolicy di risorse (IAM) che controlla l'accesso dall'endpoint al servizio specificato. La policy endpoint è specifica per l'endpoint. Inoltre, è separata da tutte le policy IAM dell'utente o dell'istanza disponibili per l'ambiente e non le sovrascrive né le sostituisce. Per informazioni dettagliate sulla creazione e sull'utilizzo delle policy degli endpoint VPC, consulta.Controllo dell'accesso ai servizi con endpoint VPCnellaAWS PrivateLinkGuida: .

L'esempio seguente consente l'accesso in sola lettura dall'endpoint VPC a App Runner. Si tratta di diritti di accesso minimi quando si utilizza l'endpoint VPC. Le entità potrebbero disporre di autorizzazioni aggiuntive tramite altri criteri.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apprunner:List*",
        "apprunner:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

Identity and Access Management per App Runner

AWS Identity and Access Management(IAM) è unAWSche aiuta gli amministratori a controllare in modo sicuro l'accesso aAWSrisorse AWS. Gli amministratori IAM controllano chi può essereautenticato(effettuato l'accesso) eautorizzato(disporre delle autorizzazioni) per utilizzare le risorse di App Runner. IAM è unAWSServizio che è possibile utilizzato senza alcun costo aggiuntivo.

Per altri argomenti sulla sicurezza di App Runner, consultaSicurezza: .

Argomenti

Audience

Modalità di utilizzoAWS Identity and Access Management(IAM) varia in base al lavoro svolto in App Runner.

Utente del servizio: se utilizzi il servizio App Runner per eseguire il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità di App Runner utilizzate per il lavoro, potrebbero essere necessarie ulteriori autorizzazioni. La comprensione della gestione dell'accesso consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una caratteristica in App Runner, consultaRisoluzione dei problemi di identità e accesso di App Runner: .

Amministratore del servizioSe sei il responsabile delle risorse di App Runner nella tua azienda, probabilmente disponi dell'accesso completo ad App Runner. Il tuo compito è determinare le caratteristiche e le risorse di App Runner a cui i dipendenti devono accedere. Devi quindi inviare inviare richieste all'amministratore IAM per la modifica delle autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con App Runner, consultaFunzionamento di App Runner con IAM: .

Amministratore IAM— Se sei un amministratore IAM, potresti essere interessato a ottenere informazioni su come scrivere policy per gestire l'accesso ad App Runner. Per visualizzare policy basate su identità di esempio di App Runner che puoi utilizzare in IAM, consultaEsempi di policy basate su identità: .

Autenticazione con identità

L'autenticazione è la procedura di accesso ad AWS utilizzando le credenziali di identità. Per ulteriori informazioni sull'accesso utilizzando ilAWS Management Console, consultaAccedi allaAWS Management Consolecome utente IAM o utente rootnellaGuida per l'utente di IAM: .

Devi essereautenticato(effettuato l'accesso adAWS) comeAWSUtente root dell'account, utente IAM o assumendo un ruolo IAM. È anche possibile utilizzare l'autenticazione Single Sign-On (SSO) della propria azienda oppure collegarsi utilizzando Google o Facebook. In questi casi, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando si accede ad AWS utilizzando le credenziali di un'altra azienda, si assume indirettamente un ruolo.

Per effettuare l'accesso direttamente allaAWS Management ConsoleUtilizza la password con l'e-mail dell'utente root o il nome utente IAM. Puoi accedere aAWSUtilizzando le chiavi di accesso dell'utente root o utente IAM.AWSfornisce SDK e strumenti a riga di comando per firmare la richiesta in maniera crittografica utilizzando le tue credenziali. Se non utilizzi gli strumenti AWS, devi firmare la richiesta personalmente. A questo scopo, utilizza Signature Version 4, un protocollo per l'autenticazione di richieste API in entrata. Per ulteriori informazioni sull'autenticazione delle richieste, consultaProcesso di firma Versione 4nellaAWSRiferimenti generali: .

Indipendentemente dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. AWS consiglia ad esempio di utilizzare la multi-factor authentication (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consultaUtilizzo dell'autenticazione a più fattori (MFA) inAWSnellaGuida per l'utente di IAM: .

Utente root dell'account AWS

Quando crei un account AWS per la prima volta, inizi con una singola identità di accesso che ha accesso completo a tutti i servizi e le risorse AWS nell'account. Questa identità è denominataAWSaccountUtente roote puoi accedervi con l'indirizzo e-mail e la password utilizzati per creare l'account. È vivamente consigliato di non utilizzare l'utente root per le attività quotidiane, anche quelle amministrative. Rispetta piuttosto la best practice di utilizzare l'utente root soltanto per creare il tuo primo utente &IAM;. Quindi conserva al sicuro le credenziali dell'utente root e utilizzale per eseguire solo alcune attività di gestione dell'account e del servizio.

Utenti e gruppi IAM

UnUtente IAMè un'identità all'interno del tuoAWSche dispone di autorizzazioni specifiche per una singola persona o applicazione. Un utente IAM può disporre di credenziali a lungo termine, ad

esempio un nome utente e una password oppure un set di chiavi di accesso. Per informazioni su come generare le chiavi di accesso, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM. Quando generi le chiavi di accesso per un utente IAM, assicurarti di visualizzare e salvare la coppia di chiavi in modo sicuro. Non puoi recuperare la chiave di accesso segreta in futuro. Al contrario, sarà necessario generare una nuova coppia di chiavi di accesso.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. Puoi 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, puoi avere un gruppo denominato Amministratori IAM e concedere a tale 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 Quando creare un utente IAM invece di un ruolo) nella Guida per l'utente di IAM.

Ruoli IAM

UnRuolo IAMè un'identità all'interno del tuoAWSAccount che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. È possibile assumere temporaneamente un ruolo IAM nellaAWS Management Consoledacambio di ruoli: . Puoi assumere un ruolo chiamando un'operazione AWS CLI o API AWS oppure utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Autorizzazioni utente IAM temporanee: un utente IAM può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso utente federato: invece di creare un utente IAM, puoi utilizzare identità preesistenti daAWS Directory Service, la directory di utente aziendale o un provider di identità Web. Sono noti come utenti federati. AWS assegna un ruolo a un utente federato quando è richiesto l'accesso tramite un provider di identità. Per ulteriori informazioni sugli utenti federati, consulta la sezione relativa a utenti federati e ruoli nella Guida per l'utente di IAM.
- Accesso multi-account: puoi utilizzare un ruolo IAM per permettere a un utente (principale attendibile) con un account diverso di accedere alle risorse nel tuo account. I ruoli sono lo

strumento principale per concedere l'accesso tra account. Tuttavia, con alcuni dei servizi AWS, puoi collegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come un proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

- Accesso tra servizi— AlcuniAWSi servizi utilizzano funzionalità in altriAWSServizi . Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o memorizzi 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.
  - Autorizzazioni principali: quando si utilizza un utente o un ruolo IAM per eseguire azioni inAWS, sei considerato un preside. I criteri concedono autorizzazioni a un'entità. Quando si utilizzano alcuni servizi, è possibile eseguire un'azione che attiva un'altra azione in un servizio diverso. In questo caso, è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per verificare se un'azione richiede azioni dipendenti aggiuntive in un criterio, consultaOperazioni, risorse e chiavi di condizione perAWS App RunnernellaService Authorization Reference: .
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni nel tuo account a tuo nome. I ruoli del servizio forniscono l'accesso all'interno del tuo account e non possono essere utilizzati per concedere l'accesso ai servizi in altri account. Un amministratore di IAM può creare, modificare ed eliminare un ruolo di servizio da IAM. Per ulteriori informazioni, consultaCreazione di un ruolo per delegare le autorizzazioni a unAWSservizionellaGuida per l'utente di IAM: .
  - Ruolo collegato ai servizi— Un ruolo collegato al servizio è un tipo di ruolo del servizio collegato a unAWSServizio. Il servizio può assumere il ruolo di eseguire un'azione per conto dell'utente. I ruoli collegati ai servizi sono visualizzati nell'account IAM e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non può modificarle.
- Applicazioni in esecuzione su Amazon EC2— È possibile utilizzare un ruolo IAM per gestire credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che inviaAWS CLIoAWSRichieste API. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un ruolo AWS a un'istanza EC2, affinché sia disponibile per tutte le relative applicazioni, puoi creare un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta Quando creare un ruolo IAM (invece di un utente) nella Guida per l'utente di IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso inAWSCreando delle policy e collegandole a identità IAM oAWSrisorse AWS. Una policy è un oggetto in AWS che, se associato a un'identità o risorsa, ne definisce le relative autorizzazioni. È possibile accedere come utente root o utente IAM oppure assumere un ruolo IAM. Quando poi fai una richiesta,AWSvaluta le policy basate su identità o sulle risorse correlate. Le autorizzazioni nella policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy viene memorizzata in AWS sotto forma di 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 di IAM.

Gli amministratori possono utilizzareAWSPolicy JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

Ogni entità IAM (utente o ruolo) inizialmente non dispone di autorizzazioni. Ovvero, per impostazione predefinita, gli utenti non possono eseguire alcuna operazione, neppure modificare la propria password. Per autorizzare un utente a eseguire operazioni, un amministratore deve collegare una policy di autorizzazioni a tale utente. In alternativa, l'amministratore può aggiungere l'utente a un gruppo che dispone delle autorizzazioni desiderate. Quando un amministratore fornisce le autorizzazioni a un gruppo, le autorizzazioni vengono concesse a tutti gli utenti in tale gruppo.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, indipendentemente 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 dalla AWS Management Console, la AWS CLI o l'API AWS.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali le risorse e in quali condizioni . Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono incorporate direttamente in un singolo utente, gruppo o ruolo. Le policy gestite sono policy standalone che possono essere collegate a più utenti, gruppi e ruoli

nell'account AWS. Le policy gestite includono le policy gestite da AWS e le policy gestite dal cliente. 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 di IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy dei bucket Amazon S3 e le policy di affidabilità dei ruoli IAM. Nei servizi che supportano criteri basati sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Per la risorsa a cui è associato il criterio, il criterio definisce le azioni che un'entità specificata può eseguire su tale risorsa e a quali condizioni. È necessario specificare un'entità in un criterio basato sulle risorse. Le entità possono includere account, utenti, ruoli, utenti federati oAWSServizi .

I criteri basati sulle risorse sono criteri inline che si trovano in tale servizio. Non è possibile utilizzareAWSle policy gestite da IAM in una policy basata su risorse.

Liste di controllo accessi di rete (ACL)

Le policy di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Gli ACL sono simili ai criteri basati sulle risorse, sebbene non utilizzino il formato del documento dei criteri JSON.

Amazon S3,AWS WAFe Amazon VPC sono esempi di servizi che supportano gli ACL. Per maggiori informazioni sugli ACL, consulta Panoramica dell'elenco di controllo degli accessi nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta tipi di policy meno comuni aggiuntivi. 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 funzione 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 suoi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono limitate 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 di IAM.

- Policy di controllo dei servizi (Service Control Policies, SCP)— Le SCP sono policy JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (UO) inAWS Organizations: .AWS Organizationsè un servizio per il raggruppamento e la gestione centralizzata di piùAWSAccount di proprietà della tua azienda. Se si abilitano tutte le caratteristiche in un'organizzazione, è possibile applicare le policy di controllo dei servizi (SCP) a uno o tutti i propri account. L'SCP limita le autorizzazioni per le entità negli account membri, compreso ogniAWSUtente root dell'account. Per ulteriori informazioni su Organizations e le policy SCP, consultaUtilizzo delle SCPnellaAWS OrganizationsGuida per l'utente: .
- Policy di sessione - Le policy di sessione sono policy avanzate che si passano 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 sull'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 gestite AWS nella Guida per l'utente di 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 comeAWSdetermina se consentire una richiesta quando sono coinvolti più tipi di policy, consultaLogica di valutazione deinellaGuida per l'utente di IAM: .

Funzionamento di App Runner con IAM

Prima di utilizzare IAM per gestire l'accesso aAWS App RunnerÈ necessario comprendere quali caratteristiche IAM sono disponibili per l'uso con App Runner. Per ottenere una presentazione generale di come App Runner e altriAWSfunzionano con IAM, vedereAWSServizi supportati da IAMnellaGuida per l'utente di IAM: .

Per altri argomenti sulla sicurezza di App Runner, consultaSicurezza: .

Argomenti

- App Runner Policy basate su identità
- App Runner Policy basate su risorse
- Autorizzazione basata sui tag di App Runner
- Autorizzazioni utente App Runner
- Ruoli IAM di App Runner

App Runner Policy basate su identità

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o rifiutate, nonché le condizioni in base alle quali le operazioni sono consentite o rifiutate. App Runner supporta specifiche operazioni, risorse e chiavi di condizione. Per informazioni su tutti gli elementi utilizzati in una policy JSON, consulta Documentazione di riferimento degli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Actions

Gli amministratori possono utilizzareAWSPolicy JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L' `Action` elemento di un criterio JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le operazioni della policy hanno spesso lo stesso nome dell'operazione API AWS. Ci sono alcune eccezioni, ad esempio azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più azioni in un criterio. Queste azioni aggiuntive sono chiamate azioni dipendenti.

L'operazione viene utilizzata in una policy per concedere le autorizzazioni di eseguire l'operazione associata.

Le operazioni delle policy in App Runner utilizzano il seguente prefisso prima dell'operazione: `apprunner:` . Ad esempio, per concedere a qualcuno l'autorizzazione per eseguire un'istanza Amazon EC2 con Amazon EC2.`RunInstances`, si include l'operazione API`ec2:RunInstances`azione nella loro politica. Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. App Runner definisce un proprio set di operazioni che descrivono le attività che puoi eseguire con quel servizio.

Per specificare più operazioni in una sola dichiarazione, separa ciascuna di esse con una virgola come mostrato di seguito:

```
"Action": [
   "apprunner:CreateService",
   "apprunner:CreateConnection"
]
```

Puoi specificare più operazioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le operazioni che iniziano con la parola `Describe`, includere la seguente operazione:

```
"Action": "apprunner:Describe*"
```

Per visualizzare un elenco delle operazioni di App Runner, consultaOperazioni definite daAWS App RunnernellaService Authorization Reference: .

Resources

Gli amministratori possono utilizzareAWSPolicy JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Resource` JSON 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 risorsa Amazon (ARN). È possibile eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, noto come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorse, ad esempio le operazioni di elenco, utilizzare un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Le risorse di App Runner presentano la seguente struttura ARN:

```
arn:aws:apprunner:region:account-id:resource-type/resource-name[/resource-id]
```

Per ulteriori informazioni sul formato di ARN, consultaAmazon Resource Name (ARN) eAWSSpazi dei nomi del servizionellaAWSRiferimenti generali: .

Ad esempio, per specificare la proprietà`my-service`Nella dichiarazione, utilizza il seguente ARN:

```
"Resource": "arn:aws:apprunner:us-east-1:123456789012:service/my-service"
```

Per specificare tutti i servizi che appartengono a un account specifico, utilizza il carattere jolly (*):

```
"Resource": "arn:aws:apprunner:us-east-1:123456789012:service/*"
```

Alcune operazioni di App Runner, ad esempio quelle per la creazione di risorse, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (*).

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di risorse di App Runner e dei relativi ARN, consultaRisorse definite daAWS App RunnernellaService Authorization Reference: . Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consultaOperazioni definite daAWS App Runner: .

Chiavi di condizione

Gli amministratori possono utilizzareAWSPolicy JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui una dichiarazione è attiva. L'elemento `Condition` è facoltativo. Puoi 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 specifichi più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione `OR` logica. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi 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.

AWSsupporta chiavi di condizione globali e chiavi di condizione specifiche per il servizio. Per visualizzare tuttiAWSChiavi di condizione globali, consultaAWSChiavi di contesto delle condizioni globalinellaGuida per l'utente di IAM: .

App Runner supporta l'utilizzo di alcune chiavi di condizione globali. Per visualizzare tuttiAWSChiavi di condizione globali, consultaAWSChiavi di contesto delle condizioni globalinellaGuida per l'utente di IAM: .

App Runner definisce un insieme di chiavi di condizione specifiche per il servizio. Inoltre, App Runner supporta il controllo degli accessi basato su tag, che viene implementato utilizzando le chiavi di

condizione. Per informazioni dettagliate, consulta the section called “Autorizzazione basata sui tag di App Runner”.

Per visualizzare l'elenco delle chiavi di condizione di App Runner, consultaChiavi di condizione perAWS App RunnernellaService Authorization Reference: . Per informazioni su operazioni e risorse con cui è possibile utilizzare una chiave di condizione, consultaOperazioni definite daAWS App Runner: .

Examples

Per visualizzare esempi di policy basate su identità di App Runner, consultaEsempi di policy basate su identità: .

App Runner Policy basate su risorse

App Runner non supporta policy basate su risorse.

Autorizzazione basata sui tag di App Runner

Puoi collegare i tag alle risorse di App Runner o passare i tag in una richiesta ad App Runner. Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `apprunner:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`. Per ulteriori informazioni sul tagging delle risorse di App Runner, consultathe section called “Configurazione”: .

Per visualizzare un esempio di policy basata su identità per limitare l'accesso a una risorsa in base ai tag di tale risorsa, consulta Controllo dell'accesso ai servizi App Runner in base ai tag.

Autorizzazioni utente App Runner

Per utilizzare App Runner, gli utenti IAM necessitano delle autorizzazioni per le azioni App Runner. Un modo comune per concedere autorizzazioni agli utenti consiste nell'allegare un criterio a utenti o gruppi IAM. Per ulteriori informazioni sulla gestione delle autorizzazioni utente, consultaModifica delle autorizzazioni per un utente IAMnellaGuida per l'utente di IAM: .

App Runner fornisce due policy gestite da disponibili per il collegamento agli utenti.

- `AWSAppRunnerReadOnlyAccess`: concede le autorizzazioni per elencare e visualizzare i dettagli relativi alle risorse di App Runner.
- `AWSAppRunnerFullAccess`: concede le autorizzazioni a tutte le azioni di App Runner.

Per un controllo più granulare delle autorizzazioni utente, è possibile creare un criterio personalizzato e collegarlo agli utenti. Per informazioni dettagliate, consultaCreazione di criteri IAMnellaGuida per l'utente di IAM: .

Per esempi di policy utente, consultathe section called “Policy utente”: .

AWSAppRunnerReadOnlyAccess

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apprunner:List*",
        "apprunner:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWSAppRunnerFullAccess

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/apprunner.amazonaws.com/
AWSServiceRoleForAppRunner",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "apprunner.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
```

```
        "StringLike": {
          "iam:PassedToService": "apprunner.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:CreateGrant"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Administrative permission over AppRunner applications",
      "Effect": "Allow",
      "Action": "apprunner:*",
      "Resource": "*"
    }
  ]
}
```

Ruoli IAM di App Runner

UnRuolo IAMè un'entità all'interno del tuoAccount AWSche dispone di autorizzazioni specifiche.

Ruoli collegati ai servizi

Ruoli collegati al servizio consentono ai servizi AWS di accedere a risorse in altri servizi per completare un'operazione a tuo nome. I ruoli collegati ai servizi sono visualizzati nell'account IAM e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non può modificarle.

App Runner supporta i ruoli collegati ai servizi. Per informazioni su come creare e gestire i ruoli collegati ai servizi di App Runner, consultathe section called “Utilizzo di ruoli collegati ai servizi”: .

Ruoli dei servizi

Questa caratteristica consente a un servizio di assumere un ruolo di servizio per conto dell'utente. Questo ruolo consente al servizio di accedere alle risorse in altri servizi per completare un'operazione per conto dell'utente. I ruoli dei servizi sono visualizzati nell'account IAM e sono di proprietà dell'account. Ciò significa che un amministratore IAM può modificare le autorizzazioni per questo ruolo. Tuttavia, questo potrebbe pregiudicare la funzionalità del servizio.

App Runner supporta alcuni ruoli di servizio.

Ruolo di accesso

Il ruolo di accesso è un ruolo utilizzato da App Runner per accedere alle immagini in Amazon Elastic Container Registry (Amazon ECR) nel tuo account. È necessario accedere a un'immagine in Amazon ECR e non è obbligatorio con Amazon ECR Public. Prima di creare un servizio basato su un'immagine in Amazon ECR, utilizzare IAM per creare un ruolo di servizio e utilizzare il`AWSAppRunnerServicePolicyForECRAccess`politica gestita in esso. È quindi possibile passare questo ruolo ad App Runner quando si chiama il metodoCreateServiceAPI nellaAuthenticationConfiguration(un membro dellaSourceConfiguration) o quando si utilizza la console App Runner per creare un servizio.

AWSAppRunnerServicePolicyForECRAccess

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:DescribeImages",
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> Se si creano criteri personalizzati per il ruolo di accesso, assicurarsi di specificare`"Resource": "*"`per`ecr:GetAuthorizationToken`Operazione . I token possono essere utilizzati per accedere a qualsiasi registro Amazon ECR a cui si ha accesso.

Quando si crea il ruolo di accesso, assicurarsi di aggiungere un criterio di attendibilità che dichiara l'entità servizio App Runner`build.apprunner.amazonaws.com`Come entità attendibile.

Criteri di attendibilità per un ruolo di accesso

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "build.apprunner.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Se si utilizza la console App Runner per creare un servizio, la console può creare automaticamente un ruolo di accesso e sceglierlo per il nuovo servizio. Nella console sono elencati anche altri ruoli nell'account e, se lo si desidera, è possibile selezionare un ruolo diverso.

Ruolo dell'istanza

Il ruolo di istanza è un ruolo facoltativo utilizzato da App Runner per fornire le autorizzazioni perAWSche il codice dell'applicazione chiama. Prima di creare un servizio App Runner, utilizzare IAM per creare un ruolo di servizio con le autorizzazioni necessarie al codice dell'applicazione. È quindi possibile passare questo ruolo ad App Runner nelCreateServiceo quando si utilizza la console App Runner per creare un servizio.

Quando si crea il ruolo di istanza, assicurarsi di aggiungere un criterio di attendibilità che dichiara l'entità servizio App Runner`tasks.apprunner.amazonaws.com`Come entità attendibile.

Criteri di attendibilità per un ruolo di istanza

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "tasks.apprunner.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
```

```
    }
  ]
}
```

Se si utilizza la console App Runner per creare un servizio, la console elenca i ruoli nell'account ed è possibile selezionare il ruolo creato per questo scopo.

Per informazioni su come creare un servizio, consulta.the section called “Creazione”: .

> **Note**
>
> Le autorizzazioni che il ruolo di istanza deve fornire dipendono interamente dall'applicazione. Il tuo codice potrebbe non chiamare alcunAWSAPI e in questo caso non è necessario fornire un ruolo di istanza a App Runner.
> Nel caso in cui il tuo codice chiamiAWSAPI, non abbiamo modo di anticipare quali chiamate sono. Di conseguenza, non viene fornito un criterio gestito per i ruoli di istanza. È necessario includere esplicitamente le autorizzazioni richieste nel ruolo di istanza oppure creare criteri personalizzati e utilizzarli nel ruolo di istanza.

Esempi di policy basate su identità

Per impostazione predefinita, gli utenti e i ruoli IAM non dispongono dell'autorizzazione per creare o modificareAWS App Runnerrisorse AWS. Inoltre, non sono in grado di eseguire attività utilizzando la AWS Management Console, AWS CLI o un'API AWS. Un amministratore IAM deve creare policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse specificate di cui hanno bisogno. L'amministratore devi quindi collegare queste policy a utenti o IAM che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy nella scheda JSON nella Guida per l'utente di IAM.

Per altri argomenti sulla sicurezza di App Runner, consultaSicurezza: .

Argomenti

- Best practice delle policy
- Policy utente

- Controllo dell'accesso ai servizi App Runner in base ai tag

Best practice delle policy

Le policy basate su identità sono molto potenti. Determinano se qualcuno può creare, accedere o eliminare risorse di App Runner nell'account. Queste operazioni possono comportare costi aggiuntivi per l'account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia subito a utilizzareAWSPolicy gestite da— Per iniziare a utilizzare rapidamente App Runner, usaAWSPolicy gestite da per fornire ai dipendenti le autorizzazioni richieste. Queste policy sono già disponibili nel tuo account e sono gestite e aggiornate da AWS. Per ulteriori informazioni, consultaInizia subito a utilizzare le autorizzazioni conAWSPolicy gestite danellaGuida per l'utente di IAM: .
- Assegna il privilegio minimo - Quando crei policy personalizzate, concedi solo le autorizzazioni richieste per eseguire un'attività. Iniziare con un set di autorizzazioni minimo e concedere autorizzazioni aggiuntive quando necessario. In questo modo, è più sicuro che iniziare con autorizzazioni che sono troppo permissive e cercare di limitarle in un secondo momento. Per ulteriori informazioni, consulta Assegnare il privilegio minimo nella Guida per l'utente di IAM.
- Abilitare MFA per operazioni sensibili: per una maggiore sicurezza, richiedi agli utenti IAM di utilizzare l'autenticazione a più fattori (MFA) per accedere a risorse sensibili o operazioni API. Per ulteriori informazioni, consultaUtilizzo dell'autenticazione a più fattori (MFA) inAWSnellaGuida per l'utente di IAM: .
- Utilizza le condizioni della policy per ulteriore sicurezza - Per quanto possibile, definisci le condizioni per cui le policy basate su identità consentono l'accesso a una risorsa. Ad esempio, è possibile scrivere condizioni per specificare un intervallo di indirizzi IP consentiti dai quali deve provenire una richiesta. È anche possibile scrivere condizioni per consentire solo le richieste all'interno di un intervallo di date o ore specificato oppure per richiedere l'utilizzo di SSL o MFA. Per ulteriori informazioni, consultaElementi delle policy JSON IAM: ConditionnellaGuida per l'utente di IAM: .

Policy utente

Per accedere alla console App Runner, gli utenti IAM devono disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentire di elencare e visualizzare i dettagli relativi alle risorse di App Runner nell'Account AWS: . Se crei una policy basata su identità più restrittiva delle

autorizzazioni minime richieste, la console non funzionerà nel modo previsto per gli utenti con tale policy.

App Runner fornisce due policy gestite da disponibili per il collegamento agli utenti.

- `AWSAppRunnerReadOnlyAccess`: concede le autorizzazioni per elencare e visualizzare i dettagli relativi alle risorse di App Runner.
- `AWSAppRunnerFullAccess`: concede le autorizzazioni a tutte le azioni di App Runner.

Per garantire che gli utenti possano utilizzare la console App Runner, allegare, come minimo, il`AWSAppRunnerReadOnlyAccess`la policy gestita agli utenti. Puoi allegare alla`AWSAppRunnerFullAccess`oppure aggiungere autorizzazioni aggiuntive specifiche per consentire agli utenti di creare, modificare ed eliminare risorse. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente di IAM.

Non sono necessarie le autorizzazioni minime della console per gli utenti che effettuano chiamate solo all'API AWS CLI o AWS. Al contrario, concedere l'accesso solo alle operazioni che soddisfano l'operazione API che si desidera consentire agli utenti di eseguire.

Negli esempi seguenti vengono illustrate le policy utente personalizzate. È possibile utilizzarli come punti di partenza per definire criteri utente personalizzati. Copiare l'esempio o rimuovere azioni, definire risorse e aggiungere condizioni.

Esempio: criteri utente per la gestione della console e delle connessioni

Questo criterio di esempio consente l'accesso alla console e consente la creazione e la gestione della connessione. Non consente la creazione e la gestione del servizio App Runner. Può essere collegato a un utente il cui ruolo è quello di gestire l'accesso al servizio App Runner alle risorse del codice sorgente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "apprunner:List*",
        "apprunner:Describe*",
        "apprunner:CreateConnection",
        "apprunner:DeleteConnection"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

Esempio: criteri utente che utilizzano chiavi di condizione

Negli esempi riportati in questa sezione vengono illustrate le autorizzazioni condizionali che dipendono da alcune proprietà delle risorse o parametri di azione.

Questo criterio di esempio abilita la creazione di un servizio App Runner ma nega l'utilizzo di una connessione denominata`prod`: .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateAppRunnerServiceWithNonProdConnections",
      "Effect": "Allow",
      "Action": "apprunner:CreateService",
      "Resource": "*",
      "Condition": {
        "ArnNotLike": {
          "apprunner:ConnectionArn": "arn:aws:apprunner:*:*:connection/prod/*"
        }
      }
    }
  ]
}
```

Questo criterio di esempio abilita l'aggiornamento di un servizio App Runner denominato`preprod`solo con una configurazione di ridimensionamento automatica denominata`preprod`: .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUpdatePreProdAppRunnerServiceWithPreProdASConfig",
      "Effect": "Allow",
      "Action": "apprunner:UpdateService",
```

```
      "Resource": "arn:aws:apprunner:*:*:service/preprod/*",
      "Condition": {
        "ArnLike": {
          "apprunner:AutoScalingConfigurationArn":
 "arn:aws:apprunner:*:*:autoscalingconfiguration/preprod/*"
        }
      }
    }
  ]
}
```

Controllo dell'accesso ai servizi App Runner in base ai tag

Puoi utilizzare le condizioni nella policy basata su identità per controllare l'accesso alle risorse di App Runner in base ai tag. Questo esempio mostra come creare una policy che consente di eliminare un servizio App Runner. Tuttavia, l'autorizzazione viene concessa solo se il valore del tag del servizio `Owner` è quello del nome utente dell'utente. Questa policy concede anche le autorizzazioni necessarie per completare questa azione nella console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListServicesInConsole",
      "Effect": "Allow",
      "Action": "apprunner:ListServices",
      "Resource": "*"
    },
    {
      "Sid": "DeleteServiceIfOwner",
      "Effect": "Allow",
      "Action": "apprunner:DeleteService",
      "Resource": "arn:aws:apprunner:*:*:service/*",
      "Condition": {
        "StringEquals": {"apprunner:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}
```

Puoi collegare questa policy agli utenti IAM nel tuo account. Se un utente denominato`richard-roe`cerca di eliminare un servizio App Runner, il servizio deve avere il tag`Owner=richard-`

`roeoowner=richard-roe`: . In caso contrario l'accesso è negato. La chiave di tag di condizione `0wner` corrisponde a `0wner` e `owner` perché i nomi delle chiavi di condizione non effettuano la distinzione tra maiuscole e minuscole. Per ulteriori informazioni, consultaElementi delle policy JSON IAM: ConditionnellaGuida per l'utente di IAM: .

Utilizzo di ruoli collegati ai servizi per App Runner

AWS App RunnerUtilizzi diAWS Identity and Access Management(IAM)Ruoli collegati ai servizi: . Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente ad App Runner. I ruoli collegati ai servizi sono definiti automaticamente da App Runner e includono tutte le autorizzazioni richieste dal servizio per eseguire chiamate agli altriAWSPer tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di App Runner perché permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. App Runner definisce le autorizzazioni dei propri ruoli associati ai servizi e, salvo diversamente definito, solo App Runner potrà assumere i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di App Runner perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta la sezione relativa ai servizi AWS che funzionano con IAM e cerca i servizi che nella colonna Service-Linked Role (Ruolo associato ai servizi) riportano Yes (Sì). Scegli un Sì con un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Per altri argomenti sulla sicurezza di App Runner, consultaSicurezza: .

Autorizzazioni del ruolo collegato ai servizi per App Runner

App Runner utilizza il ruolo collegato ai servizi denominato ai servizi.AwsserviceroleForAPPRUNNER: .

Il ruolo consente ad App Runner di eseguire le attività sotto elencate:

- Invia i log ai gruppi di log Amazon CloudWatch Logs Logs
- Crea le regole degli Amazon CloudWatch Events per iscriverti ai push delle immagini Amazon Elastic Container Registry (Amazon ECR).

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi AWSServiceRoleForAPPRUNNER considera attendibili i seguenti servizi:

- `apprunner.amazonaws.com`

La policy delle autorizzazioni del ruolo collegato ai servizi AWSServiceRoleForAPPRUNNER contiene tutte le autorizzazioni di cui App Runner ha bisogno per completare le operazioni a tuo nome.

AppRunnerServiceRolePolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:PutRetentionPolicy"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:*:*:log-group:/aws/apprunner/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/apprunner/*:log-stream:*"
      ]
    },
    {
      "Sid": "AllowPutRuleForManagedRules",
      "Effect": "Allow",
      "Action": [
        "events: PutRule",
        "events: PutTargets",
        "events: DeleteRule",
        "events: RemoveTargets",
        "events: DisableRule",
```

```
        "events: EnableRule"
      ],
      "Resource": "arn:aws:events:*:*:rule/apprunner-*",
      "Condition": {
        "StringEquals": {
          "events:ManagedBy": "apprunner.amazonaws.com",
          "events:source": "aws.ecr",
          "events:detail-type": "ECR Image Action"
        }
      }
    }
  ]
}
```

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per App Runner

Non è necessario creare manualmente un ruolo collegato ai servizi. Quando si crea un servizio App Runner nelAWS Management Console, ilAWS CLI, o ilAWSAPI, App Runner crea il ruolo collegato ai servizi per tuo conto.

Se elimini questo ruolo collegato ai servizi e quindi devi ricrearlo di nuovo, puoi utilizzare lo stesso processo per ricreare il ruolo nel tuo account. Quando crei un servizio App Runner, App Runner crea nuovamente il ruolo collegato al servizio.

Modifica di un ruolo collegato ai servizi per App Runner

App Runner non consente di modificare il ruolo collegato ai servizi AWSServiceRoleForAPPRUNNER. Dopo aver creato un ruolo collegato ai servizi, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. Puoi tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per App Runner

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non

utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare manualmente.

In App Runner, ciò significa eliminare tutti i servizi App Runner nel tuo account. Per informazioni sull'eliminazione dei servizi App Runner, vederethe section called “Eliminazione”: .

> **Note**
>
> Se il servizio App Runner utilizza tale ruolo quando tenti di eliminare le risorse, è possibile che l'eliminazione non abbia esito positivo. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare manualmente il ruolo collegato ai servizi utilizzando IAM

Utilizzare la console IAM,AWS CLI, o ilAWSPer eliminare il ruolo collegato ai servizi AWSServiceroleForAppRunner. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di App Runner

App Runner supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consultaAWS App RunnerEndpoint e quotenellaAWSRiferimenti generali: .

Policy gestite da AWS per AWS App Runner

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più semplice utilizzareAWSpolitiche gestite piuttosto che scrivere le politiche da soli. Ci vuole tempo e competenza perEsempi di policy gestite dal clienteche forniscono al team solo le autorizzazioni richieste. Per un avvio rapido, puoi utilizzare il nostroAWSPolicy gestite da Questi criteri coprono i casi d'uso comuni e sono disponibili nelAWSaccount. Per ulteriori informazioni suAWSPolicy gestite da, consultaAWSPolicy gestite danellaGuida per l'utente di IAM: .

AWSservizi di manutenzione e aggiornamentoAWSPolicy gestite da Non è possibile modificare le autorizzazioni inAWSPolicy gestite da I servizi occasionalmente aggiungono autorizzazioni aggiuntive a unAWSper supportare nuove funzionalità. Questo tipo di aggiornamento interessa

tutte le identità (utenti, gruppi e ruoli) in cui è collegata la policy. È più probabile che i servizi aggiornino unAWSquando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da unAWS, in modo che gli aggiornamenti dei criteri non violino le autorizzazioni esistenti.

Inoltre,AWS; supporta policy gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, le ricetteReadOnlyAccess AWSLa policy gestita fornisce accesso in sola lettura a tuttiAWSServizi e risorse. Quando un servizio avvia una nuova funzionalità,AWSaggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per un elenco e descrizioni delle mansioni lavorative, consultaAWSPolicy gestite per le funzioni lavorativenellaGuida per l'utente di IAM: .

App Runner si aggiorna aAWSPolicy gestite da

Visualizzare i dettagli sugli aggiornamenti diAWScriteri gestiti per App Runner da quando questo servizio ha iniziato a monitorare queste modifiche. Per avvisi automatici sulle modifiche apportate a questa pagina, iscriviti al feed RSS nella pagina Cronologia documenti App Runner.

Modifica	Descrizione	Data
ApprunnerServiceRuolePolicy— Nuova policy	App Runner ha aggiunto una nuova politica per consentire a App Runner di effettuar e chiamate a Amazon CloudWatch Logs e Amazon CloudWatch Events per conto dei servizi App Runner.	1 marzo 2021
AWSAPPRunnerReadOnlyAccess— Nuova policy	App Runner ha aggiunto un nuovo criterio per consentire agli utenti di elencare e visualizzare i dettagli sulle risorse di App Runner.	1 marzo 2021
AWSAPprunnerFullAccess— Nuova policy	App Runner ha aggiunto un nuovo criterio per consentire agli utenti di eseguire tutte le azioni di App Runner.	1 marzo 2021

Modifica	Descrizione	Data
AWSAPprunnerServicePolicyForecrAccess— Nuova policy	App Runner ha aggiunto una nuova policy per consentire ad App Runner di accedere alle immagini Amazon Elastic Container Registry (Amazon ECR) nell'account.	1 marzo 2021
App Runner ha iniziato a tenere traccia	App Runner ha iniziato a monitorare le modifiche per ilAWSPolicy gestite da	1 marzo 2021

Risoluzione dei problemi di identità e accesso di App Runner

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo diAWS App Runnere IAM.

Per altri argomenti sulla sicurezza di App Runner, consultaSicurezza: .

Argomenti

- Non sono autorizzato a eseguire un'operazione in App Runner
- Sono un amministratore e desidero consentire ad altri utenti di accedere ad App Runner
- Voglio consentire alle persone esterne al mioAccount AWSper accedere alle risorse dell'App Runner

Non sono autorizzato a eseguire un'operazione in App Runner

Se AWS Management Console indica che non sei autorizzato a eseguire un'operazione, contatta l'amministratore per assistenza. L'amministratore è la persona che ti ha fornito ilAWSNome utente e password.

L'errore di esempio seguente si verifica quando un utente IAM denominato`marymajor`cerca di utilizzare la console per visualizzare i dettagli relativi a un servizio App Runner ma non dispone di`apprunner:DescribeService`Autorizzazioni.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 apprunner:DescribeService on resource: my-example-service
```

In questo caso, Mary chiede all'amministratore di aggiornare le sue policy per poter accedere alla*my-example-service*Utilizzo della risorsa`apprunner:DescribeService`Operazione .

Sono un amministratore e desidero consentire ad altri utenti di accedere ad App Runner

Per consentire ad altri utenti di accedere ad App Runner devi creare un'entità IAM (utente o ruolo) per la persona o l'applicazione che richiede l'accesso. Tale utente o applicazione utilizzerà le credenziali dell'entità per accedere ad AWS. Dovrai quindi collegare all'entità una policy che conceda le autorizzazioni corrette in App Runner.

Per iniziare immediatamente, consulta Creazione dei primi utenti e gruppi delegati IAM nella Guida per l'utente di IAM.

Voglio consentire alle persone esterne al mioAccount AWSper accedere alle risorse dell'App Runner

Puoi creare un ruolo che può essere utilizzato dagli utenti in altri account o da persone esterne all'organizzazione per accedere alle tue risorse. Puoi specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), puoi utilizzare tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per capire se App Runner supporta queste funzionalità, consultaFunzionamento di App Runner con IAM: .
- Per informazioni su come fornire l'accesso alle risorse inAWSaccount di cui si è proprietari, vedereFornire l'accesso a un utente IAM in un altroAWSaccount di tua proprietànellaGuida per l'utente di IAM: .
- Per informazioni su come fornire l'accesso alle risorse a terze partiAWSaccount, vedereFornire l'accesso aAWSAccount di proprietà di terze partinellaGuida per l'utente di IAM: .
- Per capire come fornire l'accesso tramite la federazione delle identità, consulta Fornire accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Registrazione e monitoraggio in App Runner

Il monitoraggio è importante per mantenere l'affidabilità, la disponibilità e le prestazioni delAWS App RunnerServizio. Raccolta di dati di monitoraggio da tutte le parti delAWSconsente di eseguire più facilmente il debug di un errore se ne verifica uno. App Runner si integra con diversiAWSper monitorare i servizi App Runner e rispondere a potenziali incidenti.

Allarmi di Amazon CloudWatch

Con gli allarmi di Amazon CloudWatch, puoi osservare un parametro di servizio per il periodo di tempo specificato. Se il parametro supera una determinata soglia per un determinato numero di periodi, viene inviata una notifica.

App Runner raccoglie una serie di metriche relative al servizio nel suo complesso e alle istanze (unità di ridimensionamento) che eseguono il servizio Web. Per ulteriori informazioni, consulta Metriche (CloudWatch).

Log di applicazioni

App Runner raccoglie l'output del codice dell'applicazione e lo trasmette su Amazon CloudWatch Logs. Ciò che c'è in questo output dipende da te. Ad esempio, è possibile includere record dettagliati delle richieste effettuate al servizio Web. Questi record di log potrebbero rivelarsi utili nei controlli di accesso e di sicurezza. Per ulteriori informazioni, consulta Registri (CloudWatch Logs).

AWS CloudTrailLog azioni

App Runner è integrato conAWS CloudTrail, un servizio che fornisce un record delle operazioni eseguite da un utente, un ruolo o unAWSin App Runner. CloudTrail acquisisce tutte le chiamate API per App Runner come eventi. Puoi visualizzare gli eventi più recenti nella console CloudTrail e creare un trail per abilitare la consegna continua degli eventi CloudTrail a un bucket Amazon Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta Azioni API (CloudTrail).

Convalida della conformità per App Runner

Revisori di terze parti valutano la sicurezza e la conformità di AWS App Runner come parte di più programmi di conformità di AWS. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per sapere se App Runner o altriAWSI servizi rientrano nell'ambito di un programma di conformità specifico, consultaAWSServizi inclusi nell'ambito di un programma di conformità: . Per informazioni generali, consulta Programmi per la conformità di AWS.

Puoi scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Download di report in AWS Artifact.

La responsabilità per la compliance quando utilizzi i servizi AWS è determinata dalla riservatezza dei dati, dagli obiettivi di compliance dell'azienda e dalle normative vigenti. AWS fornisce le risorse seguenti per semplificare la compliance:

- Guide rapide su sicurezza e compliance: queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base suAWSche sono incentrati sulla sicurezza e sulla conformità.
- Whitepaper Architecting for HIPAA Security and Compliance— Questo white paper descrive come le aziende possono utilizzareAWSPer creare applicazioni conformi a HIPAA.

  > **Note**
  >
  > Non tutti i servizi sono conformi a HIPAA.

- AWSRisorse per la conformità: questa raccolta di cartelle di lavoro e guide può essere utile al tuo settore e alla tua posizione.
- Valutazione delle risorse con le regolenellaAWS ConfigGuida per gli sviluppatori— IlAWS Config: questo servizio valuta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti industriali.
- AWS Security Hub— QuestoAWSQuesto servizio fornisce una visione completa dello stato di sicurezza all'interno diAWSQuesto servizio consente di verificare la conformità con standard industriali di sicurezza e best practice
- AWS Audit Manager— QuestoAWSti aiuta a controllare continuamente il tuoAWSper semplificare la gestione dei rischi e della conformità alle normative e agli standard di settore.

Per altri argomenti di sicurezza di App Runner, consultaSicurezza: .

Resilienza in App Runner

LaAWSL'infrastruttura globale di è basata suRegioni AWSLe zone di disponibilità.Regioni AWSLe forniscono più zone di disponibilità fisicamente separate e isolate che sono connesse tramite reti altamente ridondanti, a bassa latenza e throughput elevato. Con le zone di disponibilità, è possibile progettare e gestire le applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo.

Per ulteriori informazioni suRegioni AWSLe zone di disponibilità, consultaAWSL'infrastruttura globale: .

AWS App Runner gestisce e automatizza l'utilizzo dell'infrastruttura AWS globale per tuo conto. Quando utilizzi App Runner, puoi trarre vantaggio dai meccanismi di disponibilità e tolleranza ai guasti offerti daAWSOffer.

Per altri argomenti sulla sicurezza di App Runner, consultaSicurezza: .

Sicurezza dell'infrastruttura in AWS App Runner

Come servizio gestito,AWS App Runnerè protetto dallaAWSProcedure di sicurezza di rete globali di descritte nelAmazon Web Services: Panoramica sui processi di sicurezza diWhitepaper.

UtilizzaAWSPubblicate le chiamate all'API per accedere ad App Runner tramite la rete. I client devono supportare Transport Layer Security (TLS) 1.0 o versioni successive. È consigliabile TLS 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), 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. In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per firmare le richieste.

Per altri argomenti sulla sicurezza di App Runner, consultaSicurezza: .

Analisi della configurazione e delle vulnerabilità in App Runner

AWS e i nostri clienti condividono la responsabilità del raggiungimento di un elevato livello di sicurezza e conformità dei componenti del software. Per ulteriori informazioni, consulta la .AWS Modello di responsabilità condivisa: .

Per altri argomenti sulla sicurezza di App Runner, consultaSicurezza: .

Best practice relative alla sicurezza di App Runner

AWS App Runner fornisce una serie di caratteristiche di sicurezza che occorre valutare durante lo sviluppo e l'implementazione delle policy 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, gestiscile come considerazioni utili anziché prescrizioni.

Per altri argomenti di sicurezza di App Runner, consultaSicurezza: .

Best practice relative alla sicurezza preventiva

I controlli di sicurezza preventivi tentano di prevenire gli incidenti prima che si verifichino.

Implementazione dell'accesso con privilegi minimi

Runner di app forniscAWS Identity and Access ManagementPolicy IAM gestite perUtenti IAMe laruolo di accesso: . Queste policy gestite specificano tutte le autorizzazioni che potrebbero essere necessarie per il corretto funzionamento del servizio App Runner.

La tua applicazione potrebbe non richiedere tutte le autorizzazioni nelle nostre policy gestite. Puoi personalizzarle e concedere solo le autorizzazioni necessarie per gli utenti e il servizio App Runner per eseguire le loro attività. Ciò è particolarmente rilevante per le policy utente, dove ruoli utente diversi potrebbero avere esigenze di autorizzazione diverse. L'implementazione dell'accesso con privilegi minimi è fondamentale per ridurre i rischi di sicurezza e l'impatto risultante da errori o intenzioni dannose.

Best practice relative alla sicurezza di rilevamento

I controlli di sicurezza di rilevamento identificano le violazioni della sicurezza dopo che si sono verificate. Possono aiutarti a rilevare potenziali minacce o incidenti alla sicurezza.

Implementazione del monitoraggio

Il monitoraggio è importante per garantire l'affidabilità, la disponibilità e le prestazioni delle soluzioni App Runner.AWSFornisce diversi strumenti e servizi per aiutarti a monitorare il tuoAWSServizi .

Di seguito sono elencati alcuni esempi di elementi da monitorare:

- Metriche Amazon CloudWatch per Runner di appImposta allarmi per i parametri chiave di App Runner e per i parametri personalizzati dell'applicazione. Per informazioni dettagliate, consulta Metriche (CloudWatch).
- AWS CloudTrailvoci: monitora le operazioni che potrebbero influire sulla disponibilità, ad esempio`PauseService`o`DeleteConnection`: . Per informazioni dettagliate, consulta Azioni API (CloudTrail).

AWSGlossario

Per gli utilizziAWSterminologia, vedere laAWSGlossarionellaAWSRiferimenti generali: .

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.