aws

Entwicklerhandbuch

AWS App Runner

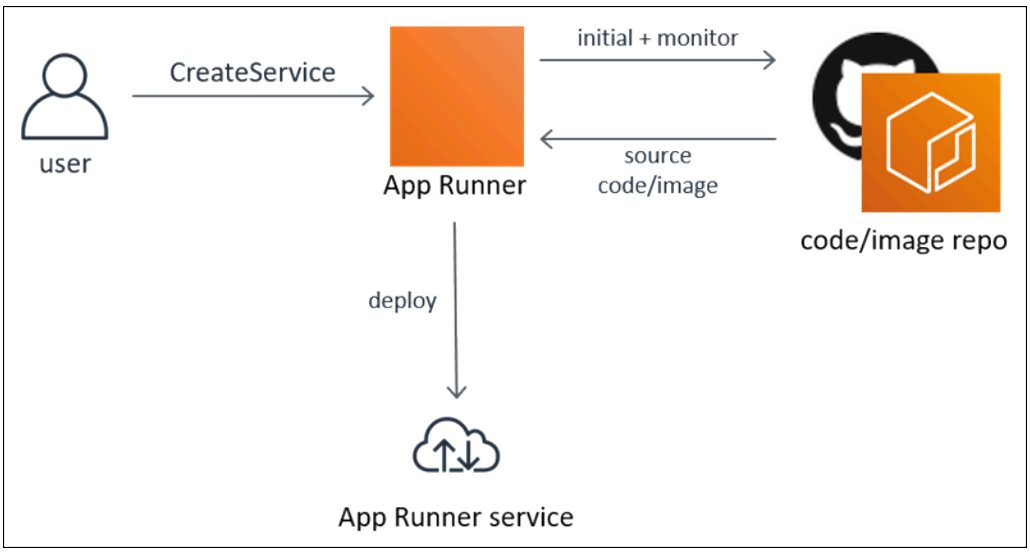
AWS App Runner: Entwicklerhandbuch

Table of Contents

Was ist AWS App Runner?

AWS App Runnerist einAWS-Dienst, der eine schnelle, einfache und kostengünstige Möglichkeit bietet, vom Quellcode oder einem Container-Image direkt in eine skalierbare und sichere Webanwendung imAWSCloud Sie müssen keine neuen Technologien erlernen, entscheiden, welchen Compute-Service Sie verwenden möchten, oder wissen, wie Sie bereitstellen und konfigurierenAWSRessourcen schätzen.

App Runner verbindet sich direkt mit Ihrem Code oder Bild-Repository. Es bietet eine automatische Integrations- und Bereitstellungspipeline mit vollständig verwaltetem Betrieb, hoher Leistung, Skalierbarkeit und Sicherheit.



Für wen ist App Runner geeignet?

Wenn Sie einDeveloperkönnen Sie App Runner verwenden, um die Bereitstellung einer neuen Version Ihres Code- oder Bild-Repositorys zu vereinfachen.

FürOperationsteamsaktiviert App Runner jedes Mal automatische Bereitstellungen, wenn ein Commit in das Code-Repository übertragen oder eine neue Container-Image-Version in das Bild-Repository übertragen wird.

Zugriff auf App Runner

Sie können Ihre App Runner-Dienstbereitstellungen mit einer der folgenden Schnittstellen definieren und konfigurieren:

- App Runner Konsole— Bietet eine Weboberfläche für die Verwaltung Ihrer App Runner-Dienste.
- API für App Runner— Bietet eine RESTful-API zum Ausführen von App Runner-Aktionen. Weitere Informationen finden Sie unter AWS App Runner-API-Referenz.
- AWSBefehlszeilenschnittstelleAWS CLI)— Bietet Befehle für eine breite Palette vonAWS-Dienste, einschließlich Amazon VPC und wird in Windows-, macOS- und Linux-Betriebssystemen unterstützt. Weitere Informationen finden Sie unter AWS Command Line Interface.
- AWS-SDKs— Bietet sprachspezifische APIs und übernimmt viele der Verbindungsdetails, wie zum Beispiel die Berechnung der Signaturen, die Verarbeitung des erneuten Absendens von Anforderungen und die Fehlerbehandlung. Weitere Informationen finden Sie unter AWS-SDKs.

Preise für App Runner

Kostenberechnung bietet eine kostengünstige Möglichkeit, Ihre Anwendung auszuführen. Sie zahlen nur für Ressourcen, die Ihr App Runner-Dienst verbraucht. Ihr Service skaliert auf weniger Compute-Instances, wenn der Anforderungsdatenverkehr langsamer ist. Sie haben die Kontrolle über die Skalierbarkeitseinstellungen: die niedrigste und höchste Anzahl bereitgestellter Instances und die höchste Last, die eine Instanz verarbeitet.

Weitere Informationen zur automatischen Skalierung von App Runner finden Sie unterthe section called “Auto Scaling”.

Preisinformationen finden Sie unter AWS App Runner– Preise.

Was kommt als nächstes

Erfahren Sie mehr über die ersten Schritte mit App Runner in den folgenden Themen:

- Einrichten— Führen Sie die erforderlichen Schritte für die Verwendung von App Runner aus.
- Erste Schritte— Stellen Sie Ihre erste Anwendung auf App Runner bereit.

Einrichten für App Runner

Wenn Sie ein neuerAWS-Kunden die Setupvoraussetzungen aus, die auf dieser Seite aufgeführt sind, bevor SieAWS App Runner.

Für diese Setup-Prozeduren verwenden Sie dieAWS Identity and Access Management(IAM) -Dienst. Vollständige Informationen zu IAM finden Sie unter den folgenden Referenzmaterialien:

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

Erstellen eines AWS-Konto

Wenn Sie sich beiAWSerhalten Sie eine Kontonummer mit Zugriff auf alle Dienste, dieAWSAngebote, einschließlichAWS App Runner.

Wenn Sie bereits über einenAWS-KontoGehen Sie direkt zur nächsten Voraussetzung über.

Wenn Sie noch kein Konto haben, einAWSSo erstellen Sie ein Konto.

Für ein AWS-Konto registrieren Sie sich wie folgt:

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Der Anmeldeprozess beinhaltet auch einen Telefonanruf und die Eingabe eines Verifizierungscodes über die Telefontastatur.

Erstellen eines IAM-Benutzers

So greifen Sie auf einenAWS-Dienst verwenden Sie Anmeldeinformationen. Diese Anmeldeinformationen bestimmenAuthentifizierung(wer Sie sind) undauthorization(welche Berechtigungen Sie zum Ausführen von Aktionen aufAWS-Ressourcen).

Wenn Sie zum ersten Mal einen Amazon Web Services (AWS-Konto verwenden, beginnen Sie mit einer Single-Sign-In-Identität. Diese Identität hat vollständigen Zugriff auf alle AWS-Services und -Ressourcen im Konto. Diese Identität wird als die MethodeAWSaccountStammbenutzer. Geben Sie

bei der Anmeldung die E-Mail-Adresse und das Passwort ein, die bzw. das Sie beim Erstellen des Kontos verwendet haben.

> **Important**
>
> Wir raten ausdrücklich davon ab, den Stammbenutzer für Alltagsaufgaben einschließlich administrativen Aufgaben zu verwenden. Bleiben Sie stattdessen bei dem bewährten -Verfahren, den Stammbenutzer nur zu verwenden, um Ihren ersten IAM-Benutzer zu erstellen. Anschließend legen Sie die Anmeldedaten für den Stammbenutzer an einem sicheren Ort ab und verwenden sie nur, um einige Konto- und Service-Verwaltungsaufgaben durchzuführen. Weitere Informationen zum Anzeigen der Aufgaben, für die Sie sich als Stammbenutzer anmelden müssen, finden Sie unter.Aufgaben, die Anmeldeinformationen des Stammbenutzers.

Weitere Informationen zu den Anmeldeinformationen des Stammbenutzers und IAM-Benutzers finden Sie unter.AWS-KontoStammbenutzer-Anmeldeinformationen und IAM-BenutzeranmeldeimAWS–Allgemeine Referenz.

So erstellen Sie einen Administratorbenutzer für sich selbst und fügen ihn einer Administratorengruppe hinzu (Konsole)

1. Melden Sie sich beimIAM-Konsoleals Kontoinhaber, indem SieStammbenutzerund geben Sie IhreAWS-E-Mail-Adresse des Kontos. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   > **Note**
   >
   > Wir empfehlen nachdrücklich, die bewährten Verfahren bei der Verwendung des -Programms**`Administrator`**IAM-Benutzer, der die Anmeldeinformationen des Stammbenutzers an einem sicheren Ort ablegt. Melden Sie sich als Stammbenutzer an, um einige Konto- und Service-Verwaltungsaufgaben durchzuführen.

2. Wählen Sie im Navigationsbereich Users und dann Add User aus.
3. Geben Sie unter Benutzername **`Administrator`** als Benutzernamen ein.
4. Aktivieren Sie das Kontrollkästchen nebenAWS Management Console-Zugriff. Wählen Sie dann Custom password (Benutzerdefiniertes Passwort) aus und geben Sie danach Ihr neues Passwort in das Textfeld ein.

5. (Optional) Standardmäßig wählt die MethodeAWSBeim ersten Anmelden muss der neue Benutzer ein neues Passwort erstellen. Sie können das Kontrollkästchen neben User must create a new password at next sign-in (Benutzer muss bei der nächsten Anmeldung ein neues Passwort erstellen) deaktivieren, um dem neuen Benutzer zu ermöglichen, sein Kennwort nach der Anmeldung zurückzusetzen.
6. Wählen Sie Weiter. Berechtigungen.
7. Wählen Sie unter Set permissions (Berechtigungen festlegen) die Option Add user to group (Benutzer der Gruppe hinzufügen) aus.
8. Wählen Sie Create group (Gruppe erstellen) aus.
9. Geben Sie im Dialogfeld Create group (Gruppe erstellen) unter Group name (Gruppenname) **`Administrators`** ein.
10. Klicken Sie aufFilterrichtlinienWählen Sie und dann aus.AWSverwaltet - Job-Funktion, um den Tabelleninhalt zu filtern.
11. Aktivieren Sie in der Richtlinienliste das Kontrollkästchen AdministratorAccess. Wählen Sie dann Create group aus.

    > **Note**
    >
    > Sie müssen IAM-Benutzer- und Rollenzugriff auf die Fakturierung aktivieren, bevor Sie die Funktion`AdministratorAccess`Berechtigungen für den Zugriff auf die RegisterkarteAWS Billing and Cost Managementconsole. Befolgen Sie hierzu die Anweisungen in Schritt 1 des Tutorials zum Delegieren des Zugriffs auf die Abrechnungskonsole.

12. Kehren Sie zur Gruppenliste zurück und aktivieren Sie das Kontrollkästchen der neuen Gruppe. Möglicherweise müssen Sie Refresh auswählen, damit die Gruppe in der Liste angezeigt wird.
13. Wählen Sie Weiter. Tags.
14. (Optional) Fügen Sie dem Benutzer Metadaten hinzu, indem Sie Tags als Schlüssel-Wert-Paare anfügen. Weitere Informationen zur Verwendung von Tags in IAM finden Sie unter Tagging von IAM-Entitäten im IAM-Benutzerhandbuch.
15. Wählen Sie Weiter. PrüfenUm die Liste der Gruppenmitgliedschaften anzuzeigen, die dem neuen Benutzer hinzugefügt werden sollen. Wenn Sie bereit sind, fortzufahren, wählen Sie Create user (Benutzer erstellen) aus.

Mit diesen Schritten können Sie weitere Gruppen und Benutzer erstellen und Ihren Benutzern Zugriff auf Ihre AWS-Kontoressourcen gewähren. So erfahren Sie mehr über die Verwendung von Richtlinien, mit denen Benutzerberechtigungen auf bestimmte bestimmteAWS-Ressourcen finden Sie unter.ZugriffsverwaltungundBeispielrichtlinien.

> **Important**
>
> Schützen Sie IhreAWS-Konto. Senden oder teilen Sie Ihre Anmeldeinformationen niemals mit Personen außerhalb Ihrer Organisation. Niemand, der Amazon legitim vertritt, wird Sie nach Ihren Anmeldeinformationen fragen.

Nachdem Sie Ihren IAM-Benutzer erstellt haben, verwenden Sie seine Anmeldeinformationen, um sich beim Anmelden bei der RegisterkarteAWS Management Console. Weitere Informationen finden Sie unterWie sich IAM-Benutzer bei IhremAWS-KontoimIAM-Benutzerhandbuch.

Erstellen eines Zugriffsschlüssels für Ihren IAM-Benutzer

Zugriffsschlüssel bestehen aus einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel. Diese werden zur Signatur der von Ihnen durchgeführten programmgesteuerten Anforderungen an die AdresseAWS. Wenn Sie noch keine Zugriffsschlüssel besitzen, können Sie diese über die RegisterkarteAWS Management Console. Als bewährte Methode verwenden Sie nicht die MethodeAWSStammbenutzer-Zugriffsschlüssel für die Aufgaben, für die dies nicht erforderlich ist. Stattdessen wähltErstellen eines neuen Administrator-IAM-Benutzersmit Zugangstasten für sich selbst.

Sie können den geheimen Zugriffsschlüssel nur beim Erstellen der -Schlüssel anzeigen oder herunterladen. Später kann er nicht mehr wiederhergestellt werden. Sie können jedoch jederzeit neue Zugriffsschlüssel erstellen. Sie müssen außerdem über Berechtigungen zum Ausführen der erforderlichen IAM-Aktionen verfügen. Weitere Informationen finden Sie unterErforderliche Berechtigungen für den Zugriff auf IAM-RessourcenimIAM-Benutzerhandbuch.

So erstellen Sie Zugriffsschlüssel für einen IAM-Benutzer

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich auf Users.

3. Wählen Sie den Namen des Benutzers aus, dessen Zugriffsschlüssel Sie erstellen möchten, und dann das Kontrollkästchen-SicherheitsanmeldeinRegisterkarte auswählen.
4. In derZugriffsschlüsselWählen Sie im BereichErstellen eines Zugriffsschlüssels.
5. Um das neue Zugriffskey pair anzuzeigen, wählen Sie im MenüAnzeigen. Sie haben keinen Zugriff auf den geheimen Zugriffsschlüssel mehr, nachdem das Dialogfeld geschlossen wird. Ihre Anmeldeinformationen sehen etwa folgendermaßen aus:

   - Zugriffsschlüssel-ID: AKIAIOSFODNN7BEISPIEL
   - Geheimer Zugriffsschlüssel: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
6. Wählen Sie zum Herunterladen des Schlüsselpaares Download .csv file aus. Speichern Sie die Schlüssel an einem sicheren Ort. Sie haben keinen Zugriff auf den geheimen Zugriffsschlüssel mehr, nachdem das Dialogfeld geschlossen wird.

   Halten Sie die Schlüssel vertraulich, um IhrAWS-Konto und senden Sie sie niemals per E-Mail. Geben Sie die Schlüssel nicht außerhalb Ihrer Organisation weiter, auch nicht im Falle von Anfragen, die von AWS oder Amazon.com zu kommen scheinen. Niemand, der Amazon legitim vertritt, wird Sie nach dem geheimen Schlüssel fragen.
7. Nachdem Sie die`.csv`-Datei, wählen SieSchließen. Wenn Sie einen Zugriffsschlüssel erstellen, ist das Schlüsselpaar standardmäßig aktiv, und Sie können es sofort verwenden.

Verwandte Themen

- Was ist IAM?imIAM-Benutzerhandbuch
- AWS-SicherheitsanmeldeinformationeninAWS– Allgemeine Referenz

Wie geht es weiter?

Sie haben die erforderlichen Schritte abgeschlossen. Informationen zum Bereitstellen der ersten Anwendung in App Runner finden Sie unterErste Schritte.

Erste Schritte mit App Runner

AWS App Runnerist einAWS-Dienst, der eine schnelle, einfache und kostengünstige Möglichkeit bietet, ein vorhandenes Container-Image oder Quellcode direkt in einen ausgeführten Webdienst imAWS Cloud.

Dieses Tutorial beschreibt, wie SieAWS App Runner, um Ihre Anwendung in einem App Runner-Dienst bereitzustellen. Sie führt durch die Konfiguration des Quellcodes und der Bereitstellung, des Dienstbuilds und der Dienstlaufzeit. Außerdem wird gezeigt, wie Sie eine Codeversion bereitstellen, eine Konfigurationsänderung vornehmen und Protokolle anzeigen. Zuletzt wird im Lernprogramm veranschaulicht, wie Sie die Ressourcen bereinigen, die Sie erstellt haben, während Sie die Vorgehensweisen des Lernprogramms befolgen.

Themen

Prerequisites

Führen Sie vor dem Starten des Tutorials die folgenden Schritte aus:

1. Führen Sie die Setup-Schritte unter aus.Einrichten.
2. Erstellen einesGitHubWenn Sie noch kein Konto besitzen, müssen Sie eines eröffnen. Wenn Sie noch nicht in GitHub sind, lesen Sie unterErste Schritte mit GitHubimGitHub Dokumente.
3. Erstellen Sie ein Repository in Ihrem GitHub Konto. In diesem Tutorial wird der Repository-Name verwendet`python-hello`. Erstellen Sie Dateien im Stammverzeichnis des Repositorys mit den in den folgenden Beispielen angegebenen Namen und Inhalten.

Dateien für die**`python-hello`**Beispiel-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)
```

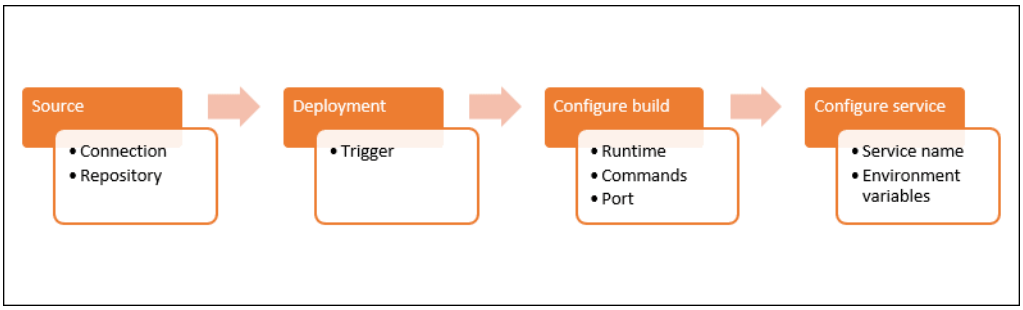
Schritt 1: Erstellen eines App Runner-Service

In diesem Schritt erstellen Sie einen App Runner-Dienst basierend auf dem Beispiel-Quellcode-Repository, das Sie auf GitHub als Teil vonthe section called “Prerequisites”. Das Beispiel enthält

eine einfache Python-Website. Dies sind die wichtigsten Schritte, die Sie unternehmen, um einen Dienst zu erstellen:

1. Konfigurieren Sie den Quellcode.
2. Konfigurieren Sie die Quellbereitstellung.
3. Konfigurieren Sie den Anwendungsaufbau.
4. Konfigurieren Sie Ihren Dienst.
5. Überprüfen und bestätigen.

Im folgenden Diagramm werden die Schritte zum Erstellen eines App Runner-Dienstes beschrieben:

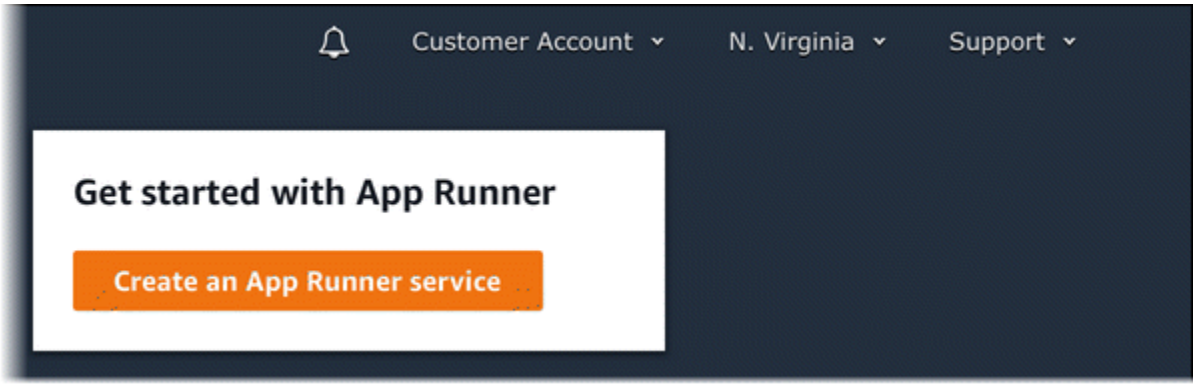


So erstellen Sie einen App Runner-Dienst basierend auf einem Quellcode-Repository

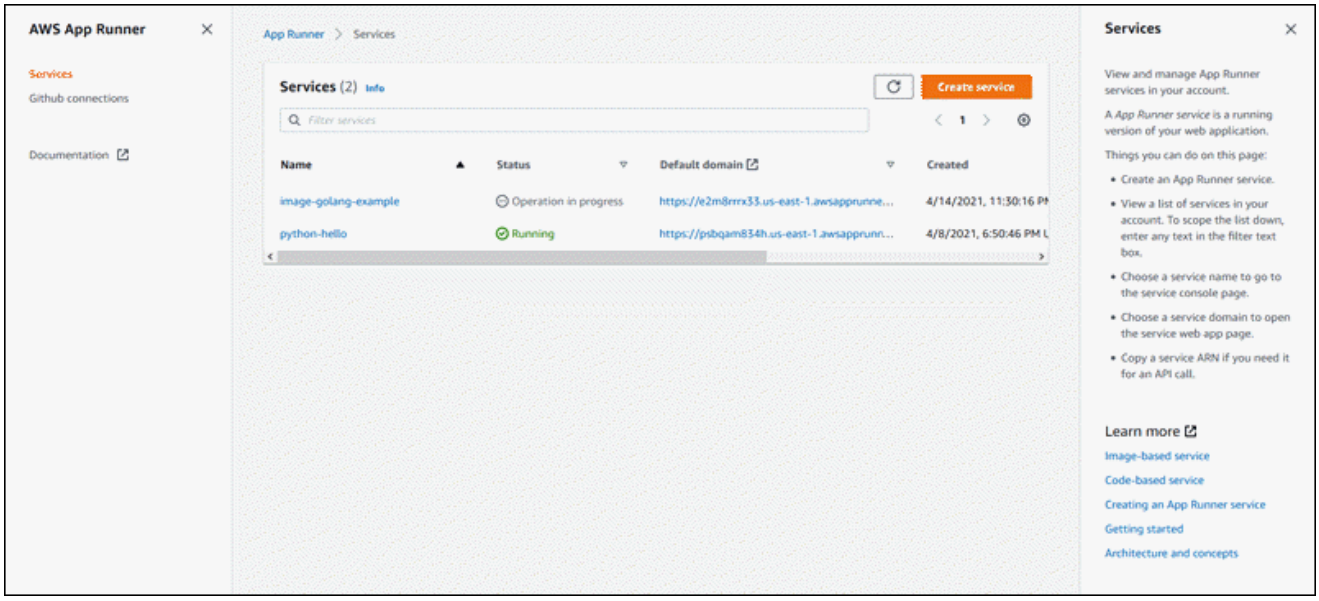
1. Konfigurieren Sie den Quellcode.

   a. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.

   b. Wenn das SymbolAWS-Kontonoch keine App Runner-Dienste hat, wird die Konsolen-Startseite angezeigt. Klicken Sie aufErstellen eines App Runner-Service.



   Wenn das SymbolAWS-Kontoverfügt über bestehende Dienste, dieServicesEine Liste Ihrer Services wird angezeigt. Wählen Sie Create service.



c. Klicken Sie auf derQuelle und BereitstellungKlicken Sie auf und danach auf.Source-Abschnitt fürRepository type (Repository-Typ), wählen SieQuellcode-Repository.

d. UNDERConnect mit GitHubWählen SieAdd NewGeben Sie bei Aufforderung Ihre GitHub Anmeldeinformationen an.

> **Note**
>
> Führen Sie die folgenden Schritte aus, um dasAWSConnector für GitHub zu Ihrem GitHub Konto sind einmalige Schritte. Sie können die Verbindung wiederverwenden, um mehrere App Runner-Dienste basierend auf Repositories in diesem Konto zu erstellen. Wenn Sie eine bestehende Verbindung haben, wählen Sie diese aus, und fahren Sie mit der Repository-Auswahl fort.

e. In derInstallAWSConnector für GitHub, wenn Sie dazu aufgefordert werden, wählen Sie Ihren GitHub Kontonamen aus.

f. Wenn Sie dazu aufgefordert werden, dieAWSConnector für GitHub, wählen SieAWS Verbindungen autorisieren.

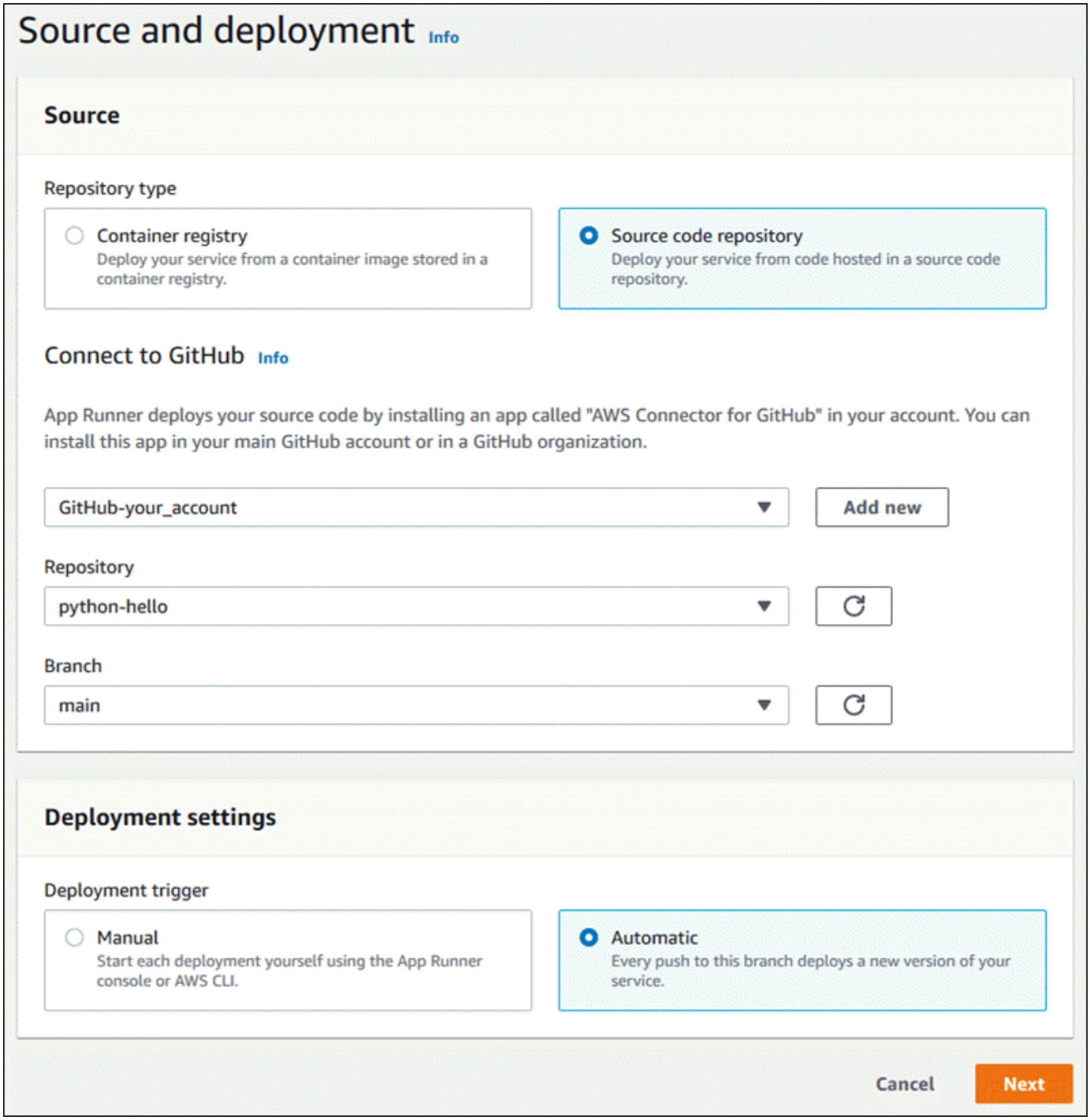
g. Wählen Sie Installieren aus.

Ihr Kontoname wird als ausgewählterGitHub Konto/Organisation. Sie können jetzt ein Repository in Ihrem Konto auswählen.

h. FürAblagedas Beispiel-Repository aus, das Sie erstellt haben,`python-hello`. FürVerzweigenWählen Sie den Standard-Zweignamen Ihres Repositorys aus (z. B.Haupt) enthalten.

2. Konfigurieren Sie Ihre Bereitstellungen: In derBereitstellungseinstellungenKlicken Sie auf und danach aufAutomatischKlicken Sie auf und danach auf.Weiter.

> **Note**
>
> Bei automatischer Bereitstellung stellt jedes neue Commit in Ihr Repository automatisch eine neue Version Ihres Dienstes bereit.



3. Konfigurieren Sie den Anwendungsaufbau.

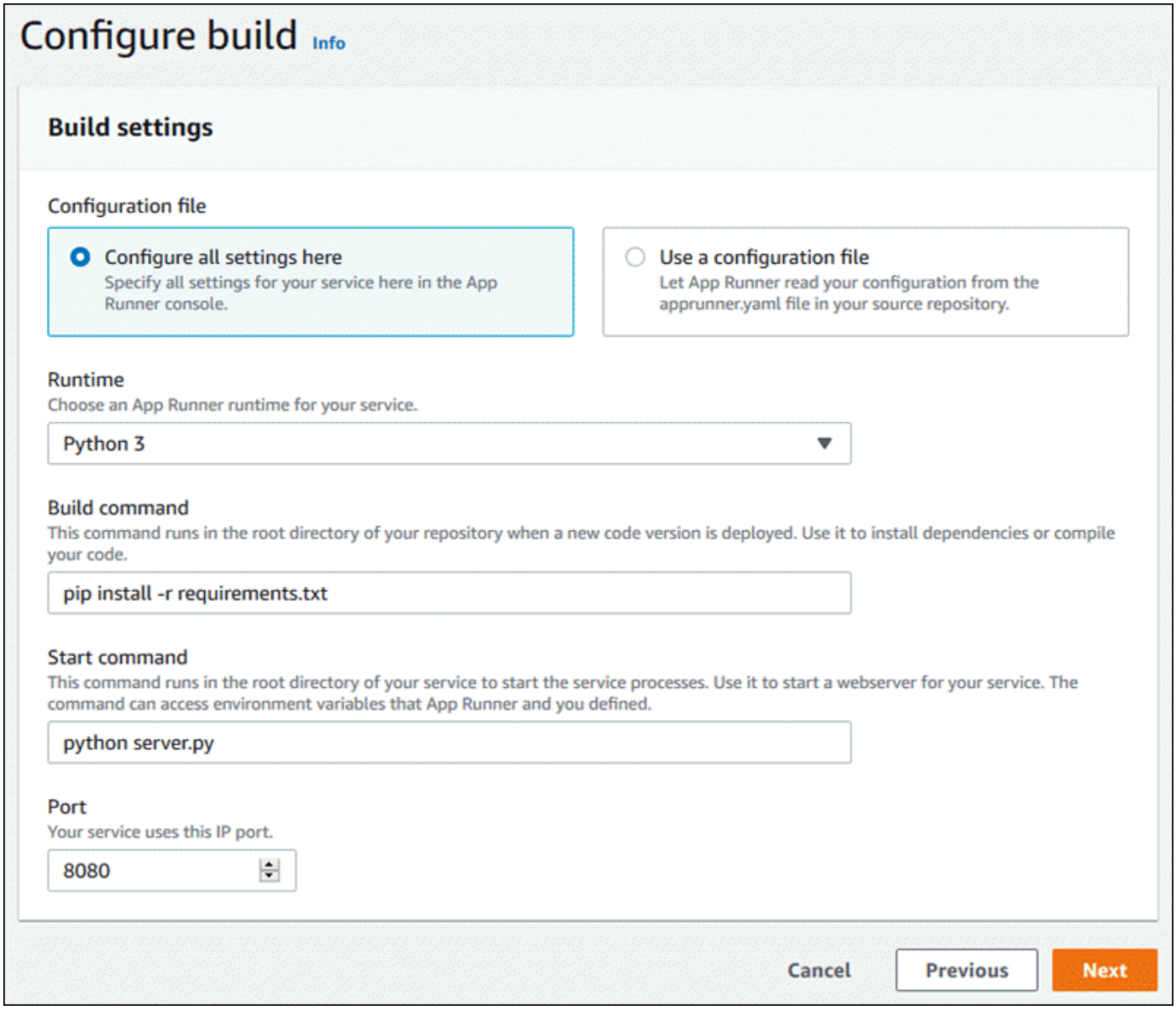
   a. Klicken Sie auf derBuild-KonfigurationSeite, fürKonfigurationsdatei, wählen SieHier alle Einstellungen konfigurieren.

   b. Geben Sie die folgenden Build-Einstellungen an:

- Runtime (Laufzeit)Wählen Sie aus.Python 3.
- Build-BefehlGeben Sie ein.**`pip install -r requirements.txt`**.
- StartbefehlGeben Sie ein.**`python server.py`**.
- PortGeben Sie ein.**`8080`**.

c. Wählen Sie Next.

> **Note**
>
> Die Python 3-Laufzeitumgebung erstellt ein Docker-Image mit einem Python 3-Basisbild und Ihrem Python-Beispielcode. Anschließend wird ein Dienst gestartet, der eine Containerinstanz dieses Abbilds ausführt.



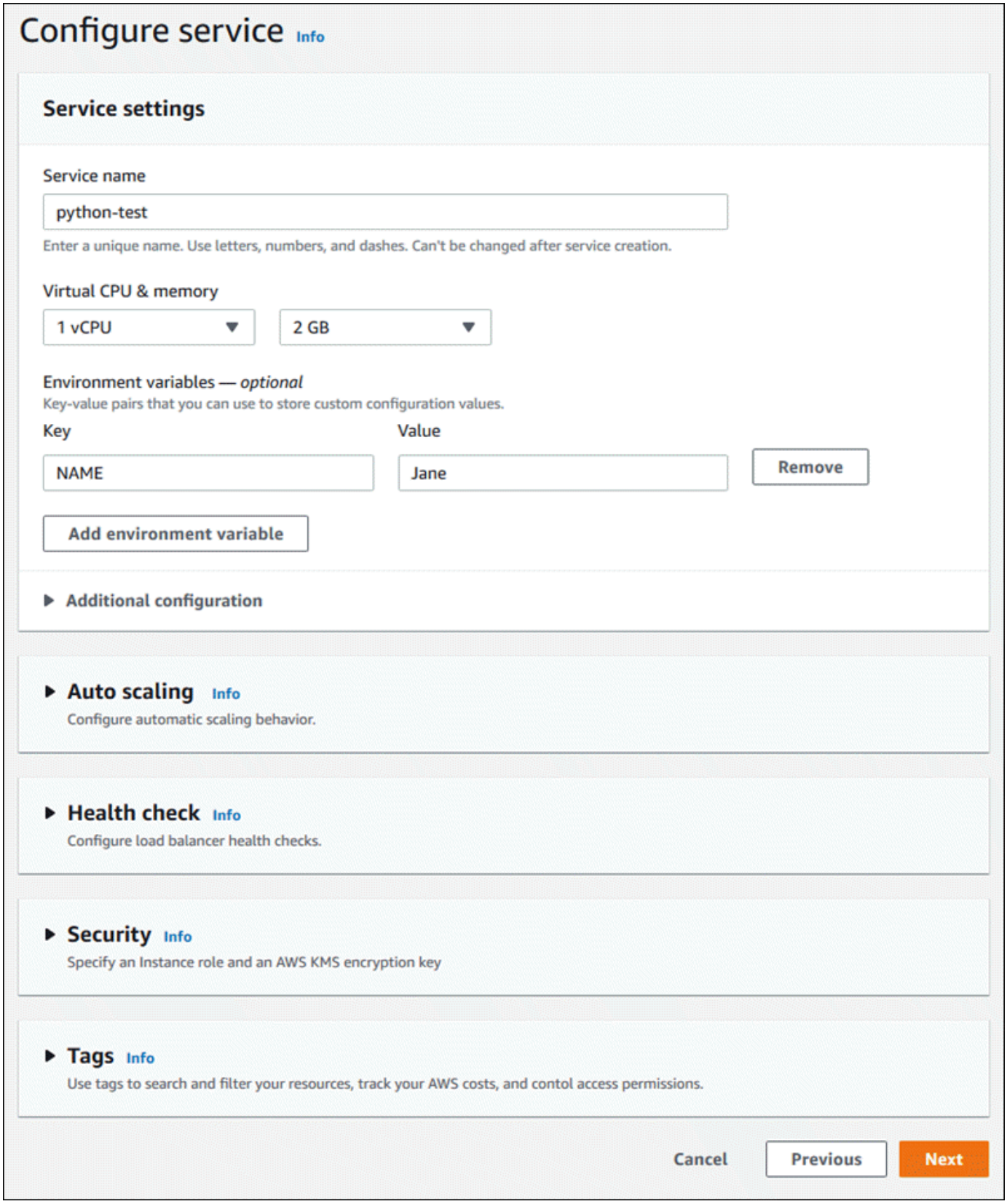
4. Konfigurieren Sie Ihren Dienst.

   a. Klicken Sie auf derKonfigurieren des ServiceKlicken Sie auf und danach auf.ServiceeinstellungenGeben Sie einen Service-Namen ein.

   b. UNDERUmgebungsvariablen, fügen Sie eine einzelne Umgebungsvariable hinzu. FürSchlüsselden Wert ein.**`NAME`**und fürValueGeben Sie einen beliebigen Namen ein (z. B. Ihren Vornamen).

> **Note**
>
> Die Beispielanwendung liest den Namen, den Sie in dieser Umgebungsvariablen festgelegt haben, und zeigt den Namen auf der Webseite an.

c. Wählen Sie Next.


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. Klicken Sie auf derÜberprüfen und erstellen, überprüfen Sie alle eingegebenen Details, und wählen SieErstellen und Bereitstellen.

   Wenn der Dienst erfolgreich erstellt wurde, zeigt die Konsole das Dienst-Dashboard mit einemServiceübersichtdes neuen Dienstes.

6. Stellen Sie sicher, dass Ihr Service ausgeführt wird.

   a. Warten Sie auf der Seite „Service-Dashboard“, bis der DienstStatusistAusführen von.

   b. Wählen Sie das SymbolStandarddomäne-Wert — es ist die URL zur Website Ihres Dienstes.

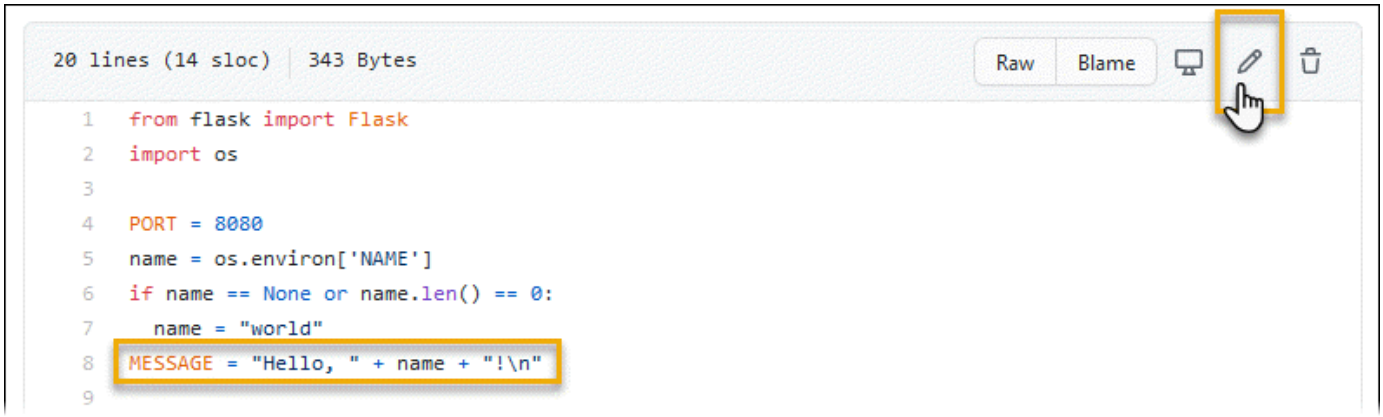
      Eine Webseite wird angezeigt: Hallo,***Ihr Name***!

Schritt 2: Ändern Sie Ihren Service-Code

In diesem Schritt ändern Sie das Quellcode-Repository. Die App Runner CI/CD Funktion erstellt automatisch und stellt die Änderung an Ihrem Service bereit.

So nehmen Sie eine Änderung an Ihrem Service-Code vor

1. Navigieren Sie zu Ihrem Beispiel GitHub Repository.
2. Wählen Sie den Dateinamen aus`server.py`, um zu dieser Datei zu navigieren.
3. Klicken Sie aufBearbeiten dieser DateiKlicken Sie auf (das Stiftsymbol).
4. In dem Ausdruck, der der Variablen zugewiesen ist`MESSAGE`, ändern Sie den Text`Hello`auf`Good morning`.



5. Wählen Sie Commit changes (Änderungen übernehmen) aus.

   Das neue Commit beginnt mit der Bereitstellung. Auf der Seite „Service-Dashboard“ wird der ServiceStatusÄnderungen anWird ausgeführt.

6. Warten Sie, bis die Bereitstellung beendet ist. Auf der Seite „Service-Dashboard“ wird der ServiceStatussollte zurück zuAusführen von.

7. Stellen Sie sicher, dass die Bereitstellung erfolgreich ist: Aktualisieren Sie die Browserregisterkarte, auf der die Webseite Ihres Dienstes angezeigt wird.

   Auf der Seite wird nun die geänderte Meldung angezeigt: Guten Morgen.***Ihr Name***!

Schritt 3: Durchführen einer Konfigurationsänderung

In diesem Schritt ändern Sie die**NAME**-Umgebungsvariablenwert, um eine Änderung der Dienstkonfiguration zu demonstrieren.

So zeigen Sie Protokolle für Ihren Service an:

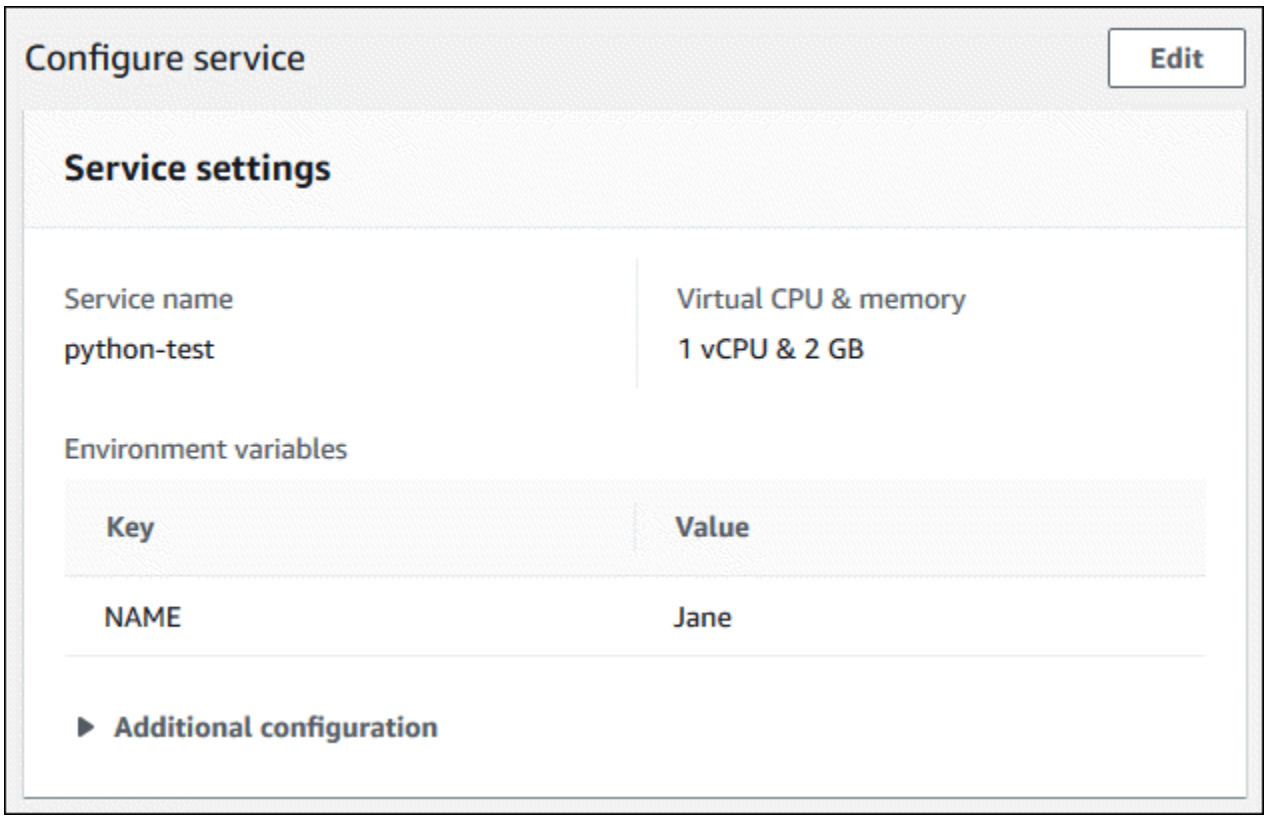
1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im Navigationsbereich und dann aus.Services, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Service-Dashboard mit einemServiceübersicht.

3. Wählen Sie auf der Service-Dashboard-Seite die OptionKonfiguration-Registerkarte.

   Die Konsole zeigt die Einstellungen für die Dienstkonfiguration in mehreren Abschnitten an.

4. In derKonfigurieren des ServiceKlicken Sie auf und danach aufBearbeiten.



5. Für die Umgebungsvariable mit dem Schlüssel**`NAME`**Ändern Sie den Wert in einen anderen Namen.
6. Wählen Sie Apply changes.

   App Runner startet den Update-Prozess. Auf der Seite „Service-Dashboard“ wird der ServiceStatusÄnderungen anWird ausgeführt.
7. Warten Sie bis die Aktualisierung beendet ist. Auf der Seite „Service-Dashboard“ wird der ServiceStatussollte zurück zuAusführen von.
8. Überprüfen Sie, ob die Aktualisierung erfolgreich ist: Aktualisieren Sie die Browserregisterkarte, auf der die Webseite Ihres Dienstes angezeigt wird.

   Auf der Seite wird nun der geänderte Name angezeigt: Guten Morgen.***`Neuer Name`***!

Schritt 4: Protokolle für Ihren Service anzeigen

In diesem Schritt verwenden Sie die App Runner-Konsole, um Protokolle für Ihren App Runner-Dienst anzuzeigen. App Runner streamt Protokolle zu Amazon CloudWatch Logs (CloudWatch Logs) und zeigt sie auf dem Dashboard Ihres Dienstes an. Informationen zu App Runner-Protokollen finden Sie unterthe section called “Protokolle (CloudWatch Logs)”.

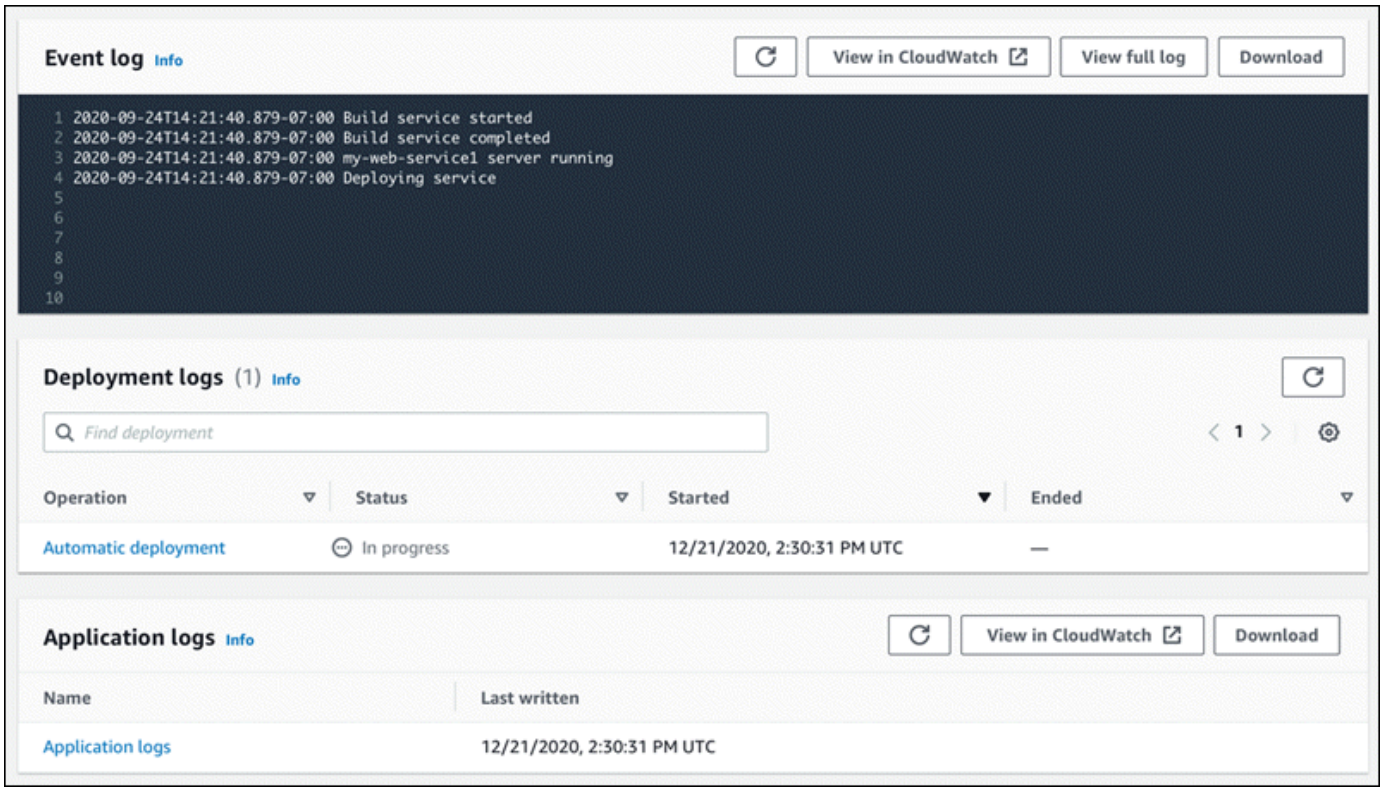
So zeigen Sie Protokolle für Ihren Service an:

1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im Navigationsbereich und dann aus.Services, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Service-Dashboard mit einemServiceübersicht.
3. Wählen Sie auf der Service-Dashboard-Seite die OptionProtokolle-Registerkarte.

   Die Konsole zeigt einige Arten von Protokollen in mehreren Abschnitten an:

   - Ereignisprotokoll— Aktivität im Lebenszyklus Ihres App Runner-Dienstes. Die Konsole zeigt die neuesten Ereignisse an.
   - Bereitstellungsprotokolle— Quell-Repository-Bereitstellungen für Ihren App Runner-Dienst. Die Konsole zeigt für jede Bereitstellung einen separaten Protokolldatenstrom an.
   - Anwendungsprotokolle— Die Ausgabe der Webanwendung, die für Ihren App Runner-Dienst bereitgestellt wird. Die Konsole kombiniert die Ausgabe aller ausgeführten Instanzen in einem einzigen Log-Stream.



4. Um bestimmte Bereitstellungen zu suchen, erweitern Sie die Bereitstellungsprotokollliste, indem Sie einen Suchbegriff eingeben. Sie können nach jedem Wert suchen, der in der Tabelle angezeigt wird.
5. Um den Inhalt eines Protokolls anzuzeigen, wählen SieAnzeigen des vollständigen Protokolls(Ereignisprotokoll) oder den Namen des Protokolldatenstroms (Bereitstellungs- und Anwendungsprotokolle).
6. Klicken Sie aufHerunterladen, um ein Protokoll herunterzuladen. Wählen Sie für einen Bereitstellungsprotokolldatenstrom zuerst einen Protokolldatenstrom aus.
7. Klicken Sie aufAnzeigen in CloudWatch, um die CloudWatch Konsole zu öffnen und die vollständigen Funktionen zu nutzen, um Ihre App Runner-Service-Protokolle zu erkunden. Wählen Sie für einen Bereitstellungsprotokolldatenstrom zuerst einen Protokolldatenstrom aus.

> **Note**
>
> Die CloudWatch Konsole ist besonders nützlich, wenn Sie Anwendungsprotokolle bestimmter Instanzen anstelle des kombinierten Anwendungsprotokolls anzeigen möchten.

Schritt 5: Bereinigen

Sie haben jetzt gelernt, wie Sie einen App Runner-Dienst erstellen, Protokolle anzeigen und einige Änderungen vornehmen. In diesem Schritt löschen Sie den Service, um -Ressourcen zu entfernen, die Sie nicht mehr benötigen.

So löschen Sie den -Service

1. Wählen Sie auf der Service-Dashboard-Seite die OptionAktionenKlicken Sie auf und danach auf.So löschen Sie -Service.
2. Geben Sie im Bestätigungsdialogfeld den gewünschten Text ein und wählen Sie dann aus.Löschen.

   Ergebnis: Die Konsole navigiert zumServicesangezeigten. Der soeben gelöschte Dienst zeigt den StatusDELETING. Kurze Zeit später verschwindet es aus der Liste.

Sie sollten auch die GitHub Verbindung löschen, die Sie im Rahmen dieses Tutorials erstellt haben. Weitere Informationen finden Sie unter the section called “Verbindungen”.

Wie geht es weiter?

Nachdem Sie Ihren ersten App Runner-Dienst bereitgestellt haben, erfahren Sie in den folgenden Themen mehr:

- Architektur und Konzepte— Die Architektur, die wichtigsten Konzepte undAWSRessourcen im Zusammenhang mit App Runner.
- Image-basierter ServiceundCode-basierter Service— Die beiden Arten von Anwendungsquellen, die App Runner bereitstellen kann.
- Entwickeln für App Runner— Dinge, die Sie beim Entwickeln oder Migrieren von Anwendungscode für die Bereitstellung in App Runner wissen sollten.
- App Runner— Verwalten und überwachen Sie Ihren Dienst über die App Runner-Konsole.
- Verwaltung von Service-ServiceVerwalten Sie den Lebenszyklus Ihres App Runner-Service.
- Protokollierung und Überwachung— Überwachen Sie Ihren App Runner-Dienst, indem Sie Metriken anzeigen, Protokolle lesen und Service-Aktionsaufrufe verfolgen.
- App Runner-Konfigurationsdatei— Eine konfigurationsbasierte Methode zum Angeben von Optionen für das Build- und Laufzeitverhalten Ihres App Runner-Dienstes.
- App RunnerVerwenden Sie die App Runner-Application Programming Interface (API), um App Runner-Ressourcen zu erstellen, zu lesen, zu aktualisieren und zu löschen.
- Sicherheit— Die verschiedenen Möglichkeiten, dieAWSund Sie gewährleisten Cloud-Sicherheit, während Sie App Runner und andere Dienste nutzen.

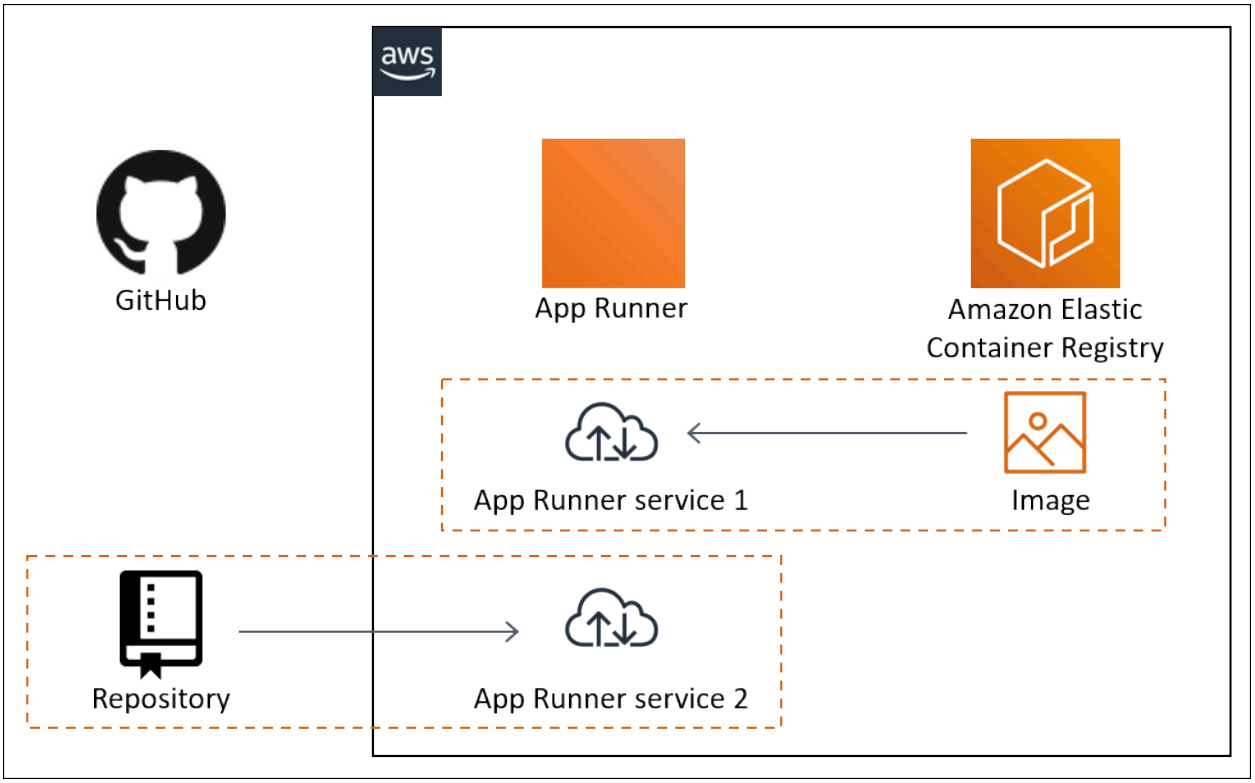
App Runner Architektur und Konzepte

AWS App Runnernimmt den Quellcode oder das Quellbild aus einem Repository und erstellt und verwaltet einen laufenden Webdienst für Sie imAWS Cloud. In der Regel müssen Sie nur eine App Runner-Aktion aufrufen,CreateService, um Ihren Service zu erstellen.

Mit einem Quell-Image-Repository stellen Sie ein gebrauchsfertiges Container-Image bereit, das App Runner zur Ausführung Ihres Webdienstes bereitstellen kann. Mit einem Quellcode-Repository können Sie Code und Anweisungen zum Erstellen und Ausführen eines Webservices bereitstellen, der für eine von mehreren Laufzeitumgebungen entwickelt wurde, die von App Runner verwaltet werden.

Derzeit kann App Runner Ihren Quellcode von einemGitHub-Repository oder rufen Sie Ihr Quellbild vonAmazon Elastic Container Registry (Amazon ECR)in IhremAWS-Konto.

Das folgende Diagramm zeigt eine Übersicht über die App Runner Service Architektur. Im Diagramm gibt es zwei Beispieldienste: Einer stellt Quellcode von GitHub bereit und der andere stellt ein Quellbild von Amazon ECR bereit.



App Runner

Im Folgenden finden Sie wichtige Konzepte für Ihren Webdienst, der in App Runner ausgeführt wird:

- App Runner— EinAWS-Ressource, die App Runner verwendet, um Ihre Anwendung basierend auf ihrem Quellcode-Repository oder Container-Image bereitzustellen und zu verwalten. Ein App Runner-Dienst ist eine laufende Version Ihrer Anwendung. Weitere Informationen zum Erstellen eines Service finden Sie unterthe section called “Erstellung”.
- Quelldatentyp— Der Typ des Quell-Repositorys, den Sie für die Bereitstellung Ihres App Runner-Dienstes bereitstellen:-Quellcodeoder .Quellbild.
- Repository-Anbieter— Der Repository-Dienst, der Ihre Anwendungsquelle enthält (z. B.GitHuboder .Amazon ECR) enthalten.
- App Runner— EinAWS-Ressource, mit der App Runner auf ein Repository-Anbieterkonto zugreifen kann (z. B. ein GitHub Konto oder eine Organisation). Weitere Informationen zu Verbindungen finden Sie unter the section called “Verbindungen”.
- Runtime (Laufzeit)— Ein Basis-Image für die Bereitstellung eines Quellcode-Repositorys. App Runner bietet eine Vielzahl vonVerwaltete Laufzeitenfür verschiedene Programmierumgebungen. Weitere Informationen finden Sie unter Code-basierter Service.
- Bereitstellung— Eine Aktion, die eine Version Ihres Quell-Repositorys (Code oder Image) auf einen App Runner-Dienst anwendet. Die erste Bereitstellung für den Dienst erfolgt im Rahmen der Diensterstellung. Spätere Bereitstellungen können auf zwei Arten erfolgen:
  - Automatische Bereitstellung— Eine CI/CD-Funktion. Sie können einen App Runner-Dienst so konfigurieren, dass jede Version Ihrer Anwendung automatisch erstellt und bereitgestellt wird, wie sie im Repository angezeigt wird. Dies kann ein neues Commit in einem Quellcode-Repository oder eine neue Bildversion in einem Quellbild-Repository sein.
  - Manuelle Bereitstellung— Eine Bereitstellung für Ihren App Runner-Dienst, die Sie explizit starten.
- Benutzerdefinierte Domäne— Eine Domäne, die Sie Ihrem App Runner-Dienst zuordnen. Benutzer Ihrer Webanwendung können diese Domain für den Zugriff auf Ihren Webdienst anstelle der Standard-Subdomain von App Runner verwenden. Weitere Informationen finden Sie unter the section called “Benutzerdefinierte Domänennamen”.
- Wartung— Eine Aktivität, die App Runner gelegentlich für die Infrastruktur ausführt, auf der Ihr App Runner-Dienst ausgeführt wird. Wenn die Wartung ausgeführt wird, ändert sich der Servicestatus vorübergehend in`OPERATION_IN_PROGRESS`(Wird ausgeführtin der Konsole) für ein paar Minuten. Aktionen in Ihrem Dienst (z. B. Bereitstellung, Konfigurationsupdate, Anhalten/Fortsetzen oder

Löschen) werden während dieser Zeit blockiert. Versuchen Sie die Aktion einige Minuten später erneut, wenn der Dienststatus auf`RUNNING`.

> **Note**
>
> Wenn Ihre Aktion fehlschlägt, bedeutet dies nicht, dass Ihr App Runner-Dienst heruntergefahren ist. Ihre Anwendung ist aktiv und bearbeitet weiterhin Anfragen. Es ist unwahrscheinlich, dass Ihr Service Ausfallzeiten hat.

Insbesondere migriert App Runner Ihren Service, wenn er Probleme in der zugrunde liegenden Hardware erkennt, die den Dienst hostet. Um Dienstausfallzeiten zu vermeiden, stellt App Runner Ihren Service auf einer neuen Gruppe von Instances bereit und verschiebt den Datenverkehr auf diese (eine blaugrüne Bereitstellung). Sie können gelegentlich einen leichten vorübergehenden Anstieg der Gebühren feststellen.

App Runner

Wenn Sie App Runner verwenden, erstellen und verwalten Sie einige Arten von Ressourcen in IhremAWS-Konto. Diese Ressourcen werden verwendet, um auf Ihren Code zuzugreifen und Ihre Dienste zu verwalten.

Die folgende Tabelle bietet eine Übersicht über diese Ressourcen:

Ressourcenname	Beschreibung
Service	Stellt eine laufende Version Ihrer Anwendung dar. Ein Großteil des restlichen Handbuchs beschreibt Servicetypen, Verwaltung, Konfiguration und Überwachung. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`*` :service/`*`service-name`*` [/`*`service-id`*` ]`
Connection	Bietet Ihren App Runner-Diensten Zugriff auf private Repositories, die bei Drittanbietern gespeichert sind. Existiert als separate Ressource für die gemeinsame Nutzung über mehrere Dienste hinweg. Weitere Informationen zu Verbindungen finden Sie unter the section called “Verbindungen”.

Ressourcenname	Beschreibung
	ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:connection/` *`connection-name`* `[/`*`connection-id`* `]`
AutoScalingConfiguration	Stellt Ihren App Runner-Diensten Einstellungen zur Verfügung, die die automatische Skalierung Ihrer Anwendung steuern. Existiert als separate Ressource für die gemeinsame Nutzung über mehrere Dienste hinweg. Weitere Informationen zur automatischen Skalierung finden Sie unterthe section called “Auto Scaling”. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:autoscalingconfiguration/` *`config-name`* `[/`*`config-revision`* `[/`*`config-id`* `]]`

App Runner Ressourcenkontingente

AWSlegt einige Kontingente (auch als Limits bezeichnet) auf Ihrem Konto fürAWS-Ressourcennutzung in jedemAWS-Region. In der folgenden Tabelle werden die Kontingente im Zusammenhang mit App Runner -Ressourcen aufgeführt. Kontingente werden auch inAWS App Runner-Endpunkte und -KontingenteimAWS– Allgemeine Referenz.

Ressourcenkontingente	Beschreibung	Standardwert	Anpassbar?
Services	Die maximale Anzahl von Services, die Sie in Ihrem Konto für jedeAWS-Region.	10	✓ Ja
Connections	Die maximale Anzahl von Verbindungen, die Sie in Ihrem Konto erstellen könnenAWS-Region. Sie können eine einzelne Verbindung über mehrere Dienste hinweg freigeben.	10	✓ Ja
Auto scaling configurations—Namen	Die maximale Anzahl eindeutiger Namen, die Sie in automatischen Skalierungskonfigurationen haben können, die Sie in Ihrem Konto für jedeAWS-Region.	10	✓ Ja

Ressourcenkontingente	Beschreibung	Standardwert	Anpassbar?
	Sie können eine automatische Skalierungskonfiguration in mehreren Diensten verwenden.		
Auto scaling configurations—Revisionen für jeden Namen	Die maximale Anzahl der automatischen Skalierungs-Konfigurationsversionen, die Sie in Ihrem Konto für jedeAWS-RegionFür jeden eindeutigen Namen. Sie können eine automatische Skalierungskonfigurationsrevision in mehreren Diensten verwenden.	10	✗ Nein

Die meisten Kontingente sind einstellbar, und Sie können eine Kontingenterhöhung für sie beantragen. Weitere Informationen finden Sie unterBeantragen einer Kontingenterhöhungim Service Quotas User Guide.

App Runner-Dienst basierend auf einem Quellbild

Sie könnenAWS App Runnerzum Erstellen und Verwalten von Diensten, die auf zwei grundsätzlich unterschiedlichen Arten von Dienstquellen basieren:-QuellcodeundQuellbild. Unabhängig vom Quelltyp kümmert sich App Runner um das Starten, Ausführen, Skalieren und Lastenausgleich Ihres Dienstes. Sie können die CI/CD-Funktion von App Runner verwenden, um Änderungen an Ihrem Quellbild oder Quellcode nachzuverfolgen. Wenn App Runner eine Änderung entdeckt, erstellt er automatisch (für Quellcode) und stellt die neue Version für Ihren App Runner-Dienst bereit.

In diesem Kapitel werden Dienste, die auf einem Quellbild basieren, erläutert. Informationen zu Diensten, die auf Quellcode basieren, finden Sie unterCode-basierter Service.

AQuellbildist ein öffentliches oder privates Container-Image, das in einem Bild-Repository gespeichert ist. Sie verweisen App Runner auf ein Image, und es startet einen Dienst, der einen Container auf der Grundlage dieses Abbilds ausführt. Es ist keine Build-Phase notwendig. Stattdessen stellen Sie ein bereitzustellendes Image bereit.

Image Repository-Anbieter

App Runner unterstützt die folgenden Image-Repository-Anbieter:

- Amazon Elastic Container Registry (Amazon ECR)— Speichert private Bilder in IhremAWS-Konto.
- Amazon Elastic Container Registry Public— Speichert öffentlich lesbare Bilder.

Von Amazon ECR aus bereitstellen

Amazon ECRspeichert Bilder in Repositories. Es gibt private und öffentliche Repositories. Um Ihr Image aus einem privaten Repository in einem App Runner-Dienst bereitzustellen, benötigt App Runner die Berechtigung, Ihr Bild von Amazon ECR zu lesen. Um App Runner diese Berechtigung zu erteilen, müssen Sie App Runner eineRolle für den Zugriff. Dies ist einAWS Identity and Access Management-Rolle (IAM), die über die erforderlichen Amazon ECR-Aktionsberechtigungen verfügt. Wenn Sie den Dienst mithilfe der App Runner-Konsole erstellen, können Sie eine vorhandene Rolle in Ihrem Konto auswählen. Alternativ können Sie die IAM-Konsole verwenden, um eine neue benutzerdefinierte Rolle zu erstellen oder für die App Runner-Konsole eine Rolle für Sie basierend auf verwalteten Richtlinien zu erstellen.

Wenn Sie die App Runner-API oder dieAWS CLISie schließen einen zweistufigen Prozess ab. Sie können die IAM-Konsole für das Erstellen einer Zugriffsrolle verwenden. Sie können eine von App Runner bereitgestellte verwaltete Richtlinie verwenden oder eigene benutzerdefinierte Berechtigungen eingeben. Anschließend stellen Sie die Zugriffsrolle während der Diensterstellung mithilfe derCreateService-API-Aktion

Weitere Informationen zur Erstellung von App Runner-Diensten finden Sie unterthe section called “Erstellung”.

Bereitstellen von Amazon ECR Public

Amazon ECR Publicspeichert öffentlich lesbare Bilder. Dies sind die Hauptunterschiede zwischen Amazon ECR und Amazon ECR Public, die Sie im Zusammenhang mit App Runner-Services beachten sollten:

- Öffentliche Bilder von Amazon ECR sind öffentlich lesbar. Sie müssen keine Zugriffsrolle bereitstellen, wenn Sie einen Service basierend auf einem Amazon ECR Public Image erstellen.
- App Runner unterstützt keine automatische Bereitstellung von Amazon ECR Public Images.

App Runner-Dienst basierend auf Quellcode

Sie können denAWS App Runnerzum Erstellen und Verwalten von Diensten, die auf zwei grundsätzlich unterschiedlichen Arten von Dienstquellen basieren:-QuellcodeundQuellbild. Unabhängig vom Quelltyp kümmert sich App Runner um das Starten, Ausführen, Skalieren und Lastenausgleich Ihres Dienstes. Sie können die CI/CD-Funktion von App Runner verwenden, um Änderungen an Ihrem Quellbild oder Quellcode nachzuverfolgen. Wenn App Runner eine Änderung entdeckt, erstellt er automatisch (für Quellcode) und stellt die neue Version für Ihren App Runner-Dienst bereit.

In diesem Kapitel werden Dienste auf Basis von Quellcode diskutiert. Informationen zu Diensten, die auf einem Quellbild basieren, finden Sie unterImage-basierter Service.

Quellcode ist Anwendungscode, den App Runner für Sie erstellt und bereitstellt. Sie verweisen App Runner auf ein Quellcode-Repository und wählen eine geeigneteruntime. App Runner erstellt ein Image, das auf dem Basisimage der Laufzeitumgebung und Ihrem Anwendungscode basiert. Es startet dann einen Dienst, der einen Container auf der Grundlage dieses Abbilds ausführt.

App Runner bietet praktische sprachspezifischeVerwaltete Laufzeiten. Jede dieser Laufzeitumgebungen erstellt ein Container-Image aus Ihrem Quellcode und fügt dem Image Language Runtime Abhängigkeiten hinzu. Sie müssen keine Container-Konfigurations- und -Build-Anweisungen wie z. B. eine Docker-Datei bereitstellen.

Unterthemen dieses Kapitels diskutieren die verschiedenen-Laufzeiten, die App Runner unterstützt — die generische Docker-File-Runtime und dieVerwaltete Laufzeitenfür verschiedene Programmierumgebungen.

Themen

Anbieter des Quellcode-Repository

App Runner stellt Ihren Quellcode bereit, indem er ihn aus einem Quellcode-Repository liest. App Runner unterstützt einen Quellcode-Repository-Anbieter: GitHub.

Bereitstellung von über GitHub

So stellen Sie den Quellcode über einen App Runner-Dienst aus einemGitHub-Repository erstellt App Runner eine Verbindung zu GitHub. Wenn Ihr Repository privat ist (d. h., es ist auf GitHub nicht öffentlich zugänglich), müssen Sie App Runner Verbindungsdetails zur Verfügung stellen. Wenn Sie die App Runner-Konsole zumErstellen eines ServicesGeben Sie im Rahmen des Service - Erstellungsvorgangs Verbindungsdetails an.

Wenn Sie die App Runner-API oder dieAWS CLIist eine Verbindung eine separate Ressource. Zuerst erstellen Sie die Verbindung mit demCreateConnectionAPI-Aktion. Anschließend stellen Sie den ARN der Verbindung während der Diensterstellung mithilfe desCreateServiceAPI-Aktion.

Weitere Informationen zur Erstellung des App Runner-Dienstes finden Sie unterthe section called “Erstellung”. Weitere Informationen zu App Runner-Verbindungen finden Sie unterthe section called “Verbindungen”.

Von App Runner verwaltete Laufzeiten

App Runner bietet verwaltete Laufzeiten für verschiedene Programmierumgebungen. Jede verwaltete Laufzeit macht es einfach, Container basierend auf einer bestimmten Programmiersprache oder Laufzeitumgebung zu erstellen und auszuführen. Wenn Sie eine verwaltete Laufzeit verwenden, beginnt App Runner mit einem verwalteten Laufzeitabbild. Dieses Bild basiert auf derAmazon Linux Docker-Imageund enthält ein Language Runtime-Paket sowie einige Tools und gängige Abhängigkeitspakete. App Runner verwendet dieses verwaltete Laufzeitabbild als Basis-Image und fügt Ihren Anwendungscode hinzu, um ein Docker-Image zu erstellen. Anschließend wird dieses Abbild bereitgestellt, um Ihren Webdienst in einem Container auszuführen.

Sie geben eine Laufzeit für Ihren App Runner-Dienst an, wenn SieErstellen eines Servicesüber die App Runner-Konsole oder dieCreateService-API. Sie können auch eine Laufzeit als Teil des Quellcodes angeben. Verwenden der`runtime`-Schlüsselwort in einemApp Runner-Konfigurationsdatei, die Sie in Ihr Code-Repository aufnehmen. Die Namenskonvention einer verwalteten Laufzeit ist*`<language-name> <major-version>`*.

App Runner aktualisiert die Laufzeitumgebung für Ihren Dienst bei jeder Bereitstellung oder Dienstaktualisierung auf die neueste Version. Wenn Ihre Anwendung eine bestimmte Version einer verwalteten Laufzeitumgebung erfordert, können Sie diese mithilfe des Befehls`runtime-version`-Schlüsselwort imApp Runner-Konfigurationsdatei. Geben Sie eine Nebenversion als*`<major>`*.*`<minor>`*, um die Haupt- und Nebenversionen zu sperren (App Runner aktualisiert nur

Patch-Versionen). Geben Sie eine bestimmte Patch-Ebene als*<major>*.*<minor>*.*<patch>*, um Ihren Dienst für eine bestimmte Laufzeitversion zu sperren (App Runner aktualisiert die Laufzeit nie).

Verwenden der verwalteten Python-Laufzeit

AWS App Runnerstellt eine verwaltete Python-Laufzeit bereit. Die Laufzeit erleichtert das Erstellen und Ausführen von Containern mit Python-basierten Webanwendungen. Wenn Sie die Python-Laufzeitumgebung verwenden, beginnt App Runner mit einem verwalteten Python-Laufzeitabbild. Dieses Bild basiert auf derAmazon Linux Docker-Imageund enthält das Python-Runtime-Paket sowie einige Tools und gängige Abhängigkeitspakete. App Runner verwendet dieses verwaltete Laufzeitabbild als Basis-Image und fügt Ihren Anwendungscode hinzu, um ein Docker-Image zu erstellen. Anschließend wird dieses Abbild bereitgestellt, um Ihren Webdienst in einem Container auszuführen.

Sie geben eine Laufzeit für Ihren App Runner-Dienst an, wenn SieErstellen eines Servicemithilfe der App Runner-Konsole oder derCreateService-API. Sie können auch eine Laufzeit als Teil des Quellcodes angeben. Verwenden der`runtime`Das Schlüsselwort in einemApp Runner-Konfigurationsdatei, die Sie in Ihr Code-Repository aufnehmen. Die Namenskonvention einer verwalteten Laufzeit ist*<language-name> <major-version>*.

Gültige Python-Laufzeitnamen finden Sie unterthe section called “Versionsinformationen”.

App Runner aktualisiert die Laufzeitumgebung für Ihren Dienst bei jeder Bereitstellung oder Dienstaktualisierung auf die neueste Version. Wenn Ihre Anwendung eine bestimmte Version einer verwalteten Laufzeitumgebung erfordert, können Sie diese mithilfe des Befehls`runtime-version`Das SchlüsselwortApp Runner-Konfigurationsdatei. Geben Sie eine Nebenversion als*<major>*.*<minor>*, um die Haupt- und Nebenversionen zu sperren (App Runner aktualisiert nur Patch-Versionen). Geben Sie eine bestimmte Patch-Ebene als*<major>*.*<minor>*.*<patch>*, um Ihren Dienst für eine bestimmte Laufzeitversion zu sperren (App Runner aktualisiert die Laufzeit nie).

Themen

- Python-Laufzeitkonfiguration
- Python-Laufzeitbeispiele
- Python-Laufzeit-Versionsinformationen

Python-Laufzeitkonfiguration

Wenn Sie eine verwaltete Laufzeitumgebung auswählen, müssen Sie mindestens auch Build- und Ausführen von Befehlen konfigurieren. Sie konfigurieren sie, währendcreatingoder .aktualisierenIhren App Runner-Dienst. Es gibt einige Möglichkeiten, dies zu tun:

- Verwenden der App Runner-Konsole— Geben Sie die Befehle in derBuild-Konfigurationauf der Registerkarte „Erstellungsprozess“ oder „Konfiguration“.
- Verwenden der App Runner-API— Rufen SieCreateServiceoder .UpdateService. Geben Sie die Befehle mit der`BuildCommand`und`StartCommand`Mitglieder desCodeConfigurationValuesDatentyp.
- Verwenden einesKonfigurationsdatei— Geben Sie einen oder mehrere Build-Befehle in bis zu drei Build-Phasen und einen einzigen Ausführungsbefehl an, der zum Starten Ihrer Anwendung dient. Es gibt zusätzliche optionale Konfigurationseinstellungen.

Die Bereitstellung einer Konfigurationsdatei ist optional. Wenn Sie einen App Runner-Dienst über die Konsole oder die API erstellen, geben Sie an, ob App Runner Ihre Konfigurationseinstellungen direkt während der Erstellung oder aus einer Konfigurationsdatei abruft.

Python-Laufzeitbeispiele

Die folgenden Beispiele zeigen App Runner-Konfigurationsdateien zum Erstellen und Ausführen eines Python-Dienstes. Das letzte Beispiel ist der Quellcode für eine vollständige Python-Anwendung, die Sie in einem Python-Laufzeitdienst bereitstellen können.

Minimale Python-Konfigurationsdatei

Dieses Beispiel zeigt eine minimale Konfigurationsdatei, die Sie mit der verwalteten Python-Laufzeit verwenden können. Informationen zu den Annahmen, die App Runner mit einer minimalen Konfigurationsdatei macht, finden Sie unterthe section called “Beispielkonfigurationsdatei”.

Example apprunner.yaml

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

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

Erweiterte Python-Konfigurationsdatei

Dieses Beispiel zeigt die Verwendung aller Konfigurationsschlüssel mit der verwalteten Python-Laufzeitumgebung.

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"
```

End-to-End-Python-Anwendungsquelle

Dieses Beispiel zeigt den Quellcode für eine vollständige Python-Anwendung, die Sie in einem Python-Laufzeitdienst bereitstellen können.

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
```

Python-Laufzeit-Versionsinformationen

In diesem Thema werden die vollständigen Details zu den Python-Laufzeitversionen aufgeführt, die App Runner unterstützt.

Python 3

Detail	Beschreibung
Laufzeitname	Python3
Nebenversionen	3.7, 3.8
Inklusive Pakete	Python, Pip, Setuptools, Rad, virtualenv

Verwenden der verwalteten Laufzeit von Node.js

AWS App Runnerstellt eine verwaltete Laufzeit von Node.js bereit. Die Laufzeit erleichtert das Erstellen und Ausführen von Containern mit Node.js-basierten Webanwendungen. Wenn Sie die Laufzeit Node.js verwenden, wird App Runner mit einem verwalteten Laufzeitabbild Node.js gestartet. Dieses Bild basiert auf derAmazon Linux Docker-Imageund enthält das Laufzeitpaket Node.js und einige Tools. App Runner verwendet dieses verwaltete Laufzeitabbild als Basis-Image und fügt Ihren Anwendungscode hinzu, um ein Docker-Image zu erstellen. Anschließend wird dieses Abbild bereitgestellt, um Ihren Webdienst in einem Container auszuführen.

Sie geben eine Laufzeit für Ihren App Runner-Dienst an, wenn SieErstellen eines Serviceüber die App Runner-Konsole oder dieCreateService-API. Sie können auch eine Laufzeit als Teil des Quellcodes angeben. Verwenden der`runtime`-Schlüsselwort in einemApp Runner-Konfigurationsdatei, die Sie in Ihr Code-Repository aufnehmen. Die Namenskonvention einer verwalteten Laufzeit ist*`<language-name> <major-version>`*.

Gültige Laufzeitnamen von Node.js finden Sie unterthe section called “Release-Informationen”.

App Runner aktualisiert die Laufzeitumgebung für Ihren Dienst bei jeder Bereitstellung oder Dienstaktualisierung auf die neueste Version. Wenn Ihre Anwendung eine bestimmte Version einer verwalteten Laufzeitumgebung erfordert, können Sie sie mit dem`runtime-version`-Schlüsselwort

imApp Runner-Konfigurationsdatei. Geben Sie eine Nebenversion als*<major>*.*<minor>*, um die Haupt- und Nebenversionen zu sperren (App Runner aktualisiert nur Patch-Versionen). Geben Sie eine bestimmte Patch-Ebene als*<major>*.*<minor>*.*<patch>*, um Ihren Dienst für eine bestimmte Laufzeitversion zu sperren (App Runner aktualisiert die Laufzeit nie).

Themen

- Node.js Laufzeitkonfiguration
- Node.js Laufzeitbeispiele
- Node.js Laufzeit-Versionsinformationen

Node.js Laufzeitkonfiguration

Wenn Sie eine verwaltete Laufzeitumgebung auswählen, müssen Sie mindestens auch Befehle erstellen und ausführen. Sie konfigurieren sie, währendcreatingoder .aktualisierenIhren App Runner-Dienst. Es gibt mehrere Möglichkeiten, dies zu tun:

- Verwenden der App Runner-Konsole— Geben Sie die Befehle in derKonfigurieren des Buildsauf der Registerkarte „Erstellungsprozess“ oder „Konfiguration“.
- Verwenden der App Runner-API— Rufen SieCreateServiceoder .UpdateService. Geben Sie die Befehle mit der`BuildCommand`und`StartCommand`-Mitglieder derCodeConfigurationValues-Datentyp.
- Verwenden einesKonfigurationsdatei— Geben Sie einen oder mehrere Build-Befehle in bis zu drei Build-Phasen und einen einzigen Ausführungsbefehl an, der zum Starten Ihrer Anwendung dient. Es gibt zusätzliche optionale Konfigurationseinstellungen.

Die Bereitstellung einer Konfigurationsdatei ist optional. Wenn Sie einen App Runner-Dienst über die Konsole oder die API erstellen, geben Sie an, ob App Runner Ihre Konfigurationseinstellungen direkt während der Erstellung oder aus einer Konfigurationsdatei abruft.

Mit der Runtime Node.js können Sie den Build und die Laufzeitumgebung auch mit einer JSON-Datei namens`package.json`im Stammverzeichnis Ihres -Quellrepositorys. Mit dieser Datei können Sie die Modulversion Node.js, Abhängigkeitspakete und verschiedene Befehle (Befehlszeilenanwendungen) konfigurieren. Paketmanager wie npm oder yarn interpretieren diese Datei als Eingabe für ihre Befehle.

Beispiel:

- npm installinstalliert Pakete, die von der`dependencies`und`devDependencies`-Knoten in`package.json`.
- npm startoder .npm run startführt den Befehl aus, der durch die`scripts/start`-Knoten in`package.json`.

Im Folgenden sehen Sie ein Beispiel für eine `package.json`-Datei.

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"
  }
}
```

Weitere Informationen zu`package.json`finden Sie unterPackage.json-Leitfadenauf der Node.js -Website.

Tips

- Wenn Ihre`package.json`-Datei definiert einestartSie können ihn alsrunin Ihrer App Runner-Konfigurationsdatei, wie das folgende Beispiel zeigt.

Example

package.json

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

apprunner.yaml

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

- Wenn Sienpm installin Ihrer Entwicklungsumgebung erstellt npm die Dateipackage-lock.json. Diese Datei enthält einen Snapshot der gerade installierten Paketversionen npm. Danach, wenn npm Abhängigkeiten installiert, verwendet es diese exakten Versionen. In ähnlicher Weise erzeugt Garnyarn.json. Übertragen Sie diese Dateien in Ihr Quellcode-Repository, um sicherzustellen, dass Ihre Anwendung mit den Versionen der Abhängigkeiten installiert wird, mit denen Sie sie entwickelt und getestet haben.
- Sie können auch eine App Runner-Konfigurationsdatei verwenden, um die Version Node.js und den Startbefehl zu konfigurieren. Wenn Sie dies tun, überschreiben diese Definitionen die unterpackage.json. Ein Konflikt zwischen dernode-Version inpackage.jsonund dieruntime-version-Wert in der App Runner-Konfigurationsdatei bewirkt, dass die Buildphase von App Runner fehlschlägt.

Node.js Laufzeitbeispiele

Die folgenden Beispiele zeigen App Runner-Konfigurationsdateien zum Erstellen und Ausführen eines Dienstes Node.js.

Minimale Konfigurationsdatei Node.js

Dieses Beispiel zeigt eine minimale Konfigurationsdatei, die Sie mit der verwalteten Laufzeit Node.js verwenden können. Informationen zu den Annahmen, die App Runner mit einer minimalen Konfigurationsdatei macht, finden Sie unterthe section called “Beispielkonfigurationsdatei”.

Example apprunner.yaml

```
version: 1.0
runtime: nodejs12
build:
  commands:
    build:
      - npm install --production
run:
  command: node app.js
```

Erweiterte Konfigurationsdatei Node.js

Dieses Beispiel zeigt die Verwendung aller Konfigurationsschlüssel mit der verwalteten Laufzeit 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"
```

Node.js App mit Grunt

Dieses Beispiel zeigt, wie eine Node.js -Anwendung konfiguriert wird, die mit Grunt entwickelt wurde. Gruntist ein JavaScript Aufgabenläufer in der Befehlszeile. Es führt sich wiederholende Aufgaben aus und verwaltet die Prozessautomatisierung, um menschliche Fehler zu reduzieren. Die Plugins Grunt und Grunt werden mit npm installiert und verwaltet. Sie konfigurieren Grunt, indem Sie die`Gruntfile.js`-Datei im Stammverzeichnis Ihres -Quellrepositorys.

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
```

Node.js Laufzeit-Versionsinformationen

In diesem Thema werden die vollständigen Details zu den von App Runner unterstützten Laufzeitversionen von Node.js aufgeführt.

Node.js 12

Detail	Beschreibung
Laufzeitname	nodejs12
Nebenversionen	latest
Inklusive Pakete	nodejs (einschließlich npm), Garn

Entwickeln von Anwendungscode für App Runner

In diesem Kapitel werden Laufzeitinformationen und Entwicklungsrichtlinien erläutert, die Sie beim Entwickeln oder Migrieren von Anwendungscode für die Bereitstellung inAWS App Runner.

Informationen zur Laufzeit

Unabhängig davon, ob Sie ein Container-Image bereitstellen oder App Runner eines für Sie erstellt, App Runner führt Ihren Anwendungscode in einer Container-Instanz aus. Hier sind einige wichtige Aspekte der Laufzeitumgebung für Containerinstanzen.

- Unterstützung des Frameworks— App Runner unterstützt jedes Image, das eine Webanwendung implementiert. Es ist unabhängig von der von Ihnen gewählten Programmiersprache und dem Webanwendungsserver oder Framework, das Sie verwenden, falls Sie eine verwenden. Für Ihre Bequemlichkeit stellen wir sprachspezifische verwaltete Laufzeiten bereit, um den Anwendungserstellungsprozess und die Erstellung abstrakter Bilder zu optimieren.
- Web-Anfragen— Ihre Containerinstanz muss HTTP-Anforderungen standardmäßig auf Port 8080 abhören. Weitere Informationen zur Konfiguration des Dienstes finden Sie unterthe section called “Konfiguration”. Sie müssen die Verarbeitung des sicheren HTTPS Traffics nicht implementieren. App Runner erfordert eingehenden HTTPS-Datenverkehr und beendet HTTPS, bevor Anforderungen an Ihre Container-Instanz übergeben werden.
- Zustandslose Apps— App Runner garantiert keine Statuspersistenz über die Dauer der Verarbeitung einer einzelnen eingehenden Webanfrage hinaus.
- Speicher— App Runner implementiert das Dateisystem in Ihrer Container-Instanz alsFlüchtige Speicherung. Dateien sind transient. Sie bleiben beispielsweise nicht bestehen, wenn Sie Ihren App Runner-Service pausieren und fortsetzen. Generell sind Dateien nicht garantiert, dass sie über die Verarbeitung einer einzelnen Anfrage hinaus bestehen, als Teil der zustandslosen Natur Ihrer Anwendung. Gespeicherte Dateien nehmen jedoch einen Teil der Speicherzuweisung Ihres App Runner-Dienstes für die Dauer ihrer Lebensdauer ein.

> **Note**
>
> Obwohl kurzlebige Speicherdateien möglicherweise nicht über Anforderungen hinweg bestehen bleiben, werden sieManchmalbestehen bleiben. Dies kann in bestimmten Situationen nützlich sein. Wenn Sie beispielsweise eine Anforderung verarbeiten, können Sie Dateien zwischenspeichern, die Ihre Anwendung herunterlädt, wenn sie für

zukünftige Anforderungen erforderlich sind. Dies kann die zukünftige Bearbeitung von Anfragen beschleunigen, kann aber die Geschwindigkeitsgewinne nicht garantieren. Ihr Code sollte nicht davon ausgehen, dass eine Datei, die in einer vorherigen Anforderung heruntergeladen wurde, noch existiert.
Für garantiertes Caching mit einem hohen Durchsatz und niedriger Latenz In-Memory-Datenspeicher verwenden Sie einen Service wieAmazon ElastiCache.

- Umgebungsvariablen— Standardmäßig erstellt App Runner die`PORT`, die in Ihrer Container-Instanz verfügbar ist. Sie können den Variablenwert mit Portinformationen konfigurieren und benutzerdefinierte Umgebungsvariablen und -werte hinzufügen. Weitere Informationen zur Konfiguration des Dienstes finden Sie unterthe section called “Konfiguration”.
- Instance-Rolle— Wenn Ihr Anwendungscode Aufrufe zu einemAWS-Dienste mithilfe der Service-APIs oder einer derAWS-SDKs erstellen Sie eine Instance-Rolle mitAWS Identity and Access Management(IAM). Hängen Sie ihn dann beim Erstellen an Ihren App Runner-Dienst an. Alle einschließenAWS-Dienstaktionsberechtigungen, die Ihr Code in Ihrer Instanzrolle benötigt. Weitere Informationen finden Sie unter the section called “Instance-Rolle”.

Richtlinien für die Code-Entwicklung

Beachten Sie diese Richtlinien, wenn Sie Code für eine App Runner-Webanwendung entwickeln.

- Zustandslose Codes entwerfen— Entwerfen Sie die Webanwendung, die Sie für Ihren App Runner-Dienst bereitstellen, zustandslos. Ihr Code sollte davon ausgehen, dass kein Status über die Dauer der Verarbeitung einer einzelnen eingehenden Webanforderung hinaus besteht.
- Temporäre Dateien löschen— Wenn Sie Dateien erstellen, werden sie auf einem Dateisystem gespeichert und nehmen einen Teil der Speicherzuweisung Ihres Dienstes in Anspruch. Um Fehler außerhalb des Speichers zu vermeiden, sollten Sie temporäre Dateien nicht über längere Zeiträume aufbewahren. Balance der Speichergröße mit der Geschwindigkeit der Anforderungsverarbeitung bei Entscheidungen im Dateizwischenspeichern

Verwenden der App Runner-Konsole

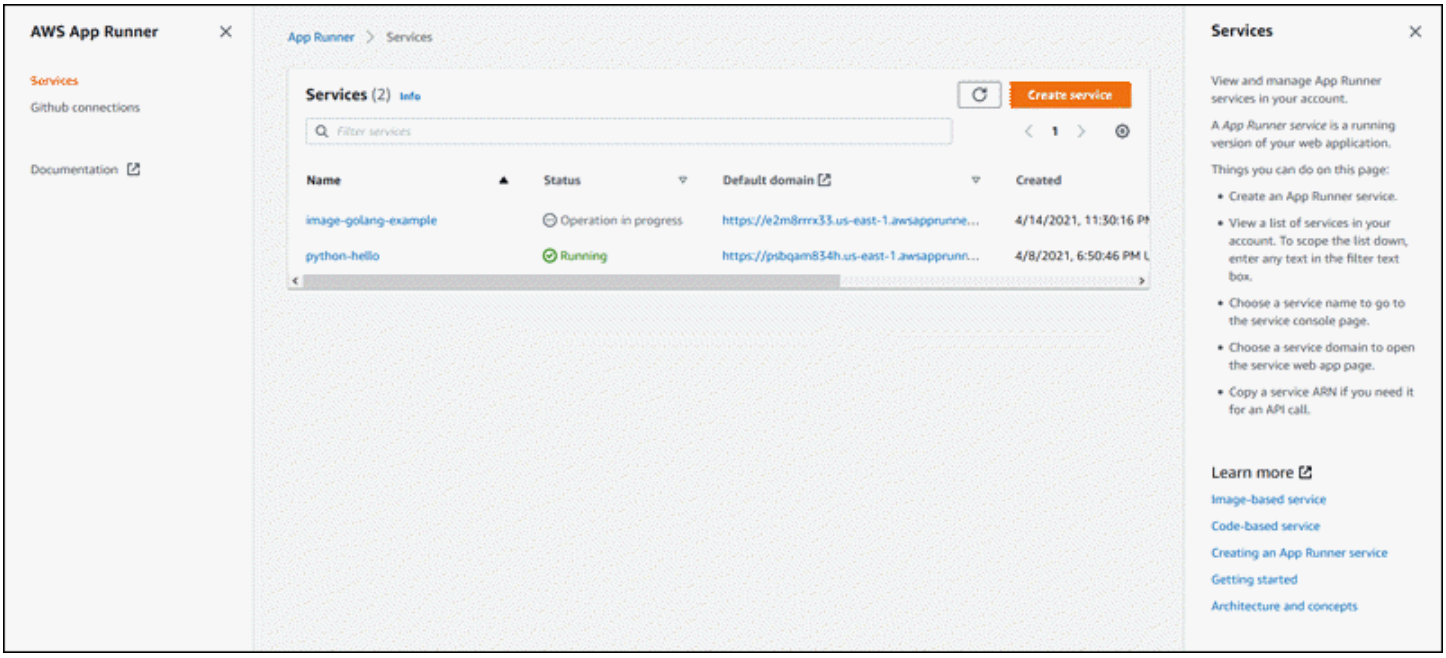
Verwenden derAWS App Runner-Konsole können Sie Ihre App Runner-Dienste und zugehörigen Ressourcen wie Verbindungen erstellen, verwalten und überwachen. Sie können vorhandene Dienste anzeigen, neue erstellen und einen Dienst konfigurieren. Sie können den Status eines App Runner-Dienstes anzeigen sowie Protokolle anzeigen, Aktivitäten überwachen und Metriken verfolgen. Sie können auch zur Website Ihres Dienstes oder zu Ihrem Quell-Repository navigieren.

In den folgenden Abschnitten wird das Layout und die Funktionalität der Konsole beschrieben und Sie verweisen auf zugehörige Informationen.

Gesamt-Konsolen-Layout

Die App Runner-Konsole verfügt über drei Bereiche. Von links nach rechts:

- Navigationsbereich— Ein Seitenbereich, der ausgeblendet oder erweitert werden kann. Verwenden Sie es, um die Konsolenseite der obersten Ebene auszuwählen, die Sie verwenden möchten.
- Inhaltsbereich— Der Hauptteil der Konsolenseite. Verwenden Sie es, um Informationen anzuzeigen und Ihre Aufgaben auszuführen.
- Hilfebereich— Ein Seitenbereich für weitere Informationen. Erweitern Sie es, um Hilfe zu der Seite zu erhalten, auf der Sie sich befinden. Oder wählen Sie eineInformationenauf einer Konsolenseite, um kontextabhängige Hilfe zu erhalten.



Seite Services

DieServiceslistet App Runner-Dienste in Ihrem Konto auf. Sie können die Liste mithilfe des Filter-Textfelds nach unten erweitern.

Um zurServicesangezeigten

1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im Navigationsbereich aus.Services.

Dinge, die Sie hier tun können:

- Erstellen eines App Runner Weitere Informationen finden Sie unter the section called “Erstellung”.
- Wählen Sie einen Dienstnamen aus, um zur Seite der Service-Dashboard-Konsole zu wechseln.
- Wählen Sie eine Dienstdomäne aus, um die Service-Web-App-Seite zu öffnen.

Die Seite „Service-Dashboard“

Sie können Informationen zu einem App Runner-Dienst anzeigen und über die Service-Dashbaord-Seite verwalten. Am oberen Rand der Seite sehen Sie den Service-Namen.

Um zum Service-Dashboard zu gelangen, navigieren Sie zumServices(siehe vorheriger Abschnitt), und wählen Sie dann Ihren App Runner-Dienst aus.

DieÜbersicht über den Servicefinden Sie grundlegende Details zum App Runner-Dienst und Ihrer Anwendung. Dinge, die Sie hier tun können:

- Zeigen Sie Service-Details wie Status, Status und ARN an.
- Navigieren Sie zurStandarddomäne— die Domäne, die App Runner für die Webanwendung bereitstellt, die in Ihrem Dienst ausgeführt wird. Dies ist eine Subdomain in der`awsapprunner.com`Domain im Besitz von App Runner.
- Navigieren Sie zum Quell-Repository, das für den Service bereitgestellt wurde.
- Starten Sie eine Bereitstellung des Quell-Repositorys für Ihren Service.
- Halten Sie Ihren Dienst an, setzen Sie ihn fort und löschen Sie ihn.

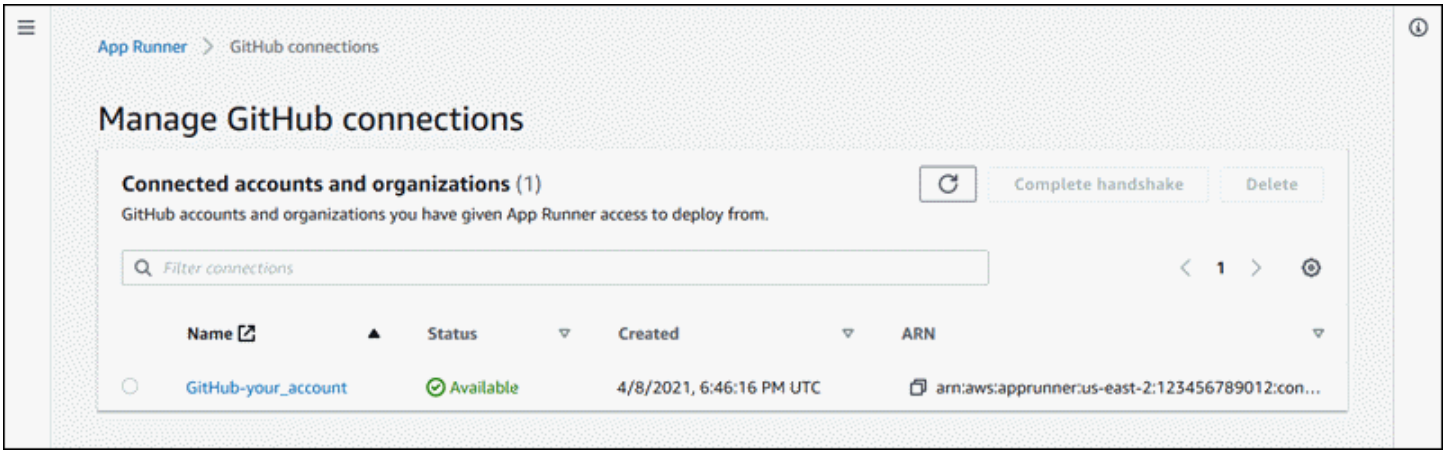
Die Registerkarten unterhalb der Serviceübersicht sind für den ServiceVerwaltungund-Überwachung.

Die GitHub Verbindungsseite

DieGitHublistet App Runner-Verbindungen zu GitHub in Ihrem Konto auf. Sie können die Liste mithilfe des Filter-Textfelds nach unten erweitern. Weitere Informationen zu Verbindungen finden Sie unter the section called “Verbindungen”.

Um zurGitHubangezeigten

1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im Navigationsbereich aus.GitHub.



Dinge, die Sie hier tun können:

- Anzeigen einer Liste von GitHub Verbindungen in Ihrem Konto. Um die Liste auszuweiten, geben Sie einen beliebigen Text in das Filtertextfeld ein.
- Wählen Sie einen Verbindungsnamen aus, um zum zugehörigen GitHub Konto oder Organisation zu wechseln.
- Wählen Sie eine Verbindung aus, um den Handshake für eine Verbindung abzuschließen, die Sie soeben hergestellt haben (als Teil der Erstellung eines Dienstes), oder um die Verbindung zu löschen.

Verwalten Ihres App Runner-Service-Lebenszyklus

In diesem Kapitel wird beschrieben, wie Sie den Lebenszyklus vonAWS App Runner-Service. In diesem Kapitel erfahren Sie, wie Sie einen Dienst erstellen, konfigurieren und löschen, wie Sie neue Anwendungsversionen für Ihren Dienst bereitstellen und wie Sie Verbindungen verwalten. Außerdem erfahren Sie, wie Sie die Verfügbarkeit Ihres Webdienstes steuern können, indem Sie Ihren Dienst pausieren und fortsetzen.

Themen

Erstellen eines App Runner-Dienstes

AWS App Runnerautomatisiert den Prozess des Übergangs von einem Containerabbild oder einem Quellcode-Repository zu einem ausgeführten Webdienst, der automatisch skaliert wird. Sie verweisen App Runner auf Ihr Quellbild oder Ihren Quellcode und geben nur eine kleine Anzahl von erforderlichen Einstellungen an. App Runner erstellt Ihre Anwendung bei Bedarf, stellt Rechenressourcen bereit und stellt Ihre Anwendung bereit, damit sie ausgeführt werden kann.

Wenn Sie einen Service erstellen, erstellt App Runner eine-Service-Ressource erstellen. In einigen Fällen müssen Sie möglicherweise eineVerbindung-Ressource erstellen. Wenn Sie die App Runner-Konsole verwenden, erstellt die Konsole implizit die Verbindungsressource. Weitere Informationen zu App Runner Ressourcentypen finden Sie unterthe section called “App Runner”. Diese Ressourcentypen verfügen über Kontingente, die Ihrem Konto in jedemAWS-Region. Weitere Informationen finden Sie unter the section called “App Runner Ressourcenkontingente”.

Je nach Quelltyp und Provider gibt es subtile Unterschiede in der Vorgehensweise zum Erstellen eines Dienstes. In diesem Thema werden völlig separate Verfahren zum Erstellen dieser

verschiedenen Quelltypen angezeigt, sodass Sie den am besten zu Ihrer Situation passenden Quelltypen folgen können. Eine grundlegende Startprozedur mit einem Codebeispiel finden Sie unterErste Schritte.

Prerequisites

Bevor Sie Ihren App Runner-Dienst erstellen, müssen Sie die folgenden Aktionen ausführen:

- Führen Sie die Einrichtungsschritte unterEinrichten.
- Halten Sie Ihre Anwendungsquelle bereit. Sie können entweder ein Code-Repository inGitHuboder ein Containerbild inAmazon Elastic Container Registry (Amazon ECR), um einen App Runner-Dienst zu erstellen.

Erstellen eines Services

In diesem Abschnitt wird der Erstellungsprozess für die beiden App Runner-Diensttypen erläutert: basierend auf Quellcode und basierend auf einem Containerabbild.

Erstellen eines Dienstes aus einem GitHub Code-Repository

In den folgenden Abschnitten wird gezeigt, wie man einen App Runner-Service erstellt, wenn es sich bei der Quelle um ein Code-Repository inGitHub. Wenn Sie GitHub verwenden, muss App Runner eine Verbindung mit der GitHub-Organisation oder dem Konto herstellen. Daher müssen Sie helfen, diese Verbindung herzustellen. Weitere Informationen zu App Runner-Verbindungen finden Sie unterthe section called “Verbindungen”.

Wenn Sie den Service erstellen, erstellt App Runner ein Docker-Image, das Ihren Anwendungscode und Abhängigkeiten enthält. Anschließend wird ein Dienst gestartet, der eine Containerinstanz dieses Abbilds ausführt.

Themen

- Erstellen eines Dienstes aus Code mit der App Runner-Konsole
- Erstellen eines Dienstes aus Code mit der App Runner-API oderAWS CLI

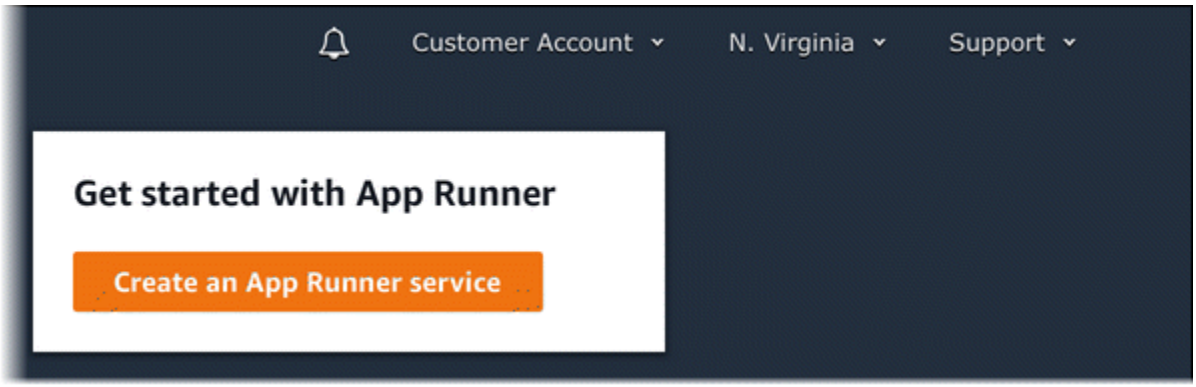
Erstellen eines Dienstes aus Code mit der App Runner-Konsole

So erstellen Sie mit der Konsole einen App Runner-Dienst

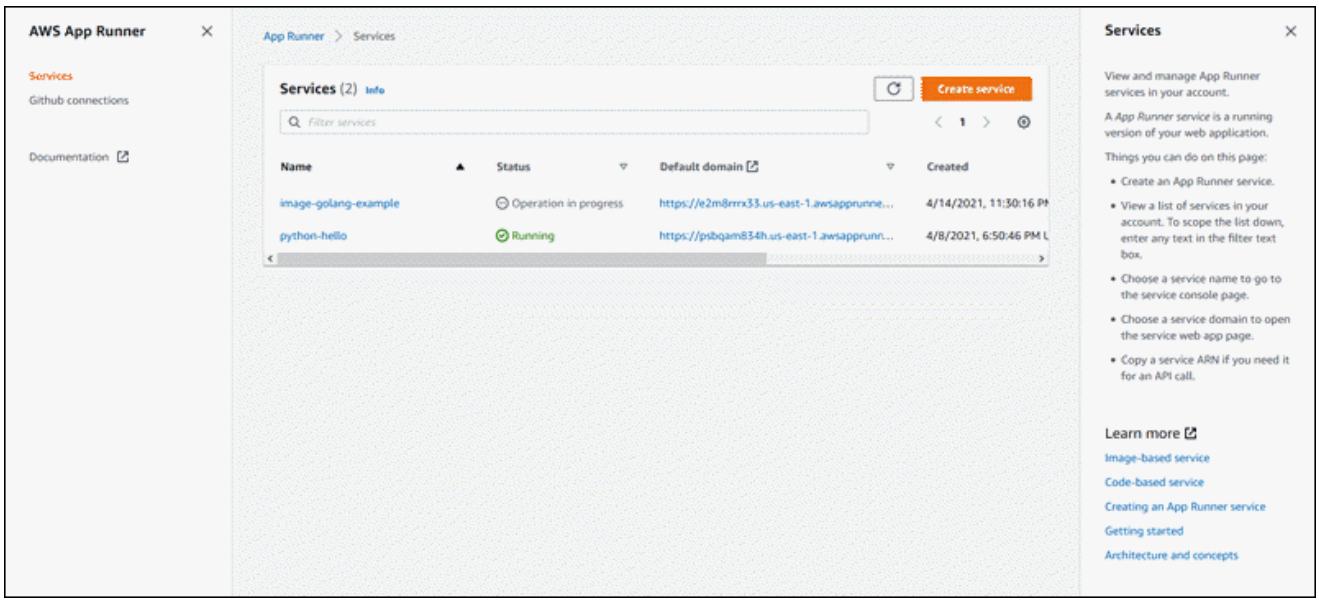
1. Konfigurieren Sie den Quellcode.

a. Öffnen SieApp-Runner-Konsoleund in derRegionenaus, wählen Sie IhreAWS-Region.

b. Wenn das SymbolAWS-Kontonoch keine App Runner-Dienste hat, wird die Konsolen-Startseite angezeigt. Klicken Sie aufErstellen eines App Runner-Service.



Wenn das SymbolAWS-Kontoverfügt über bereits vorhandene Dienste, dieServicesEine Seite mit einer Liste Ihrer Services wird angezeigt. Wählen Sie Create service.



c. Klicken Sie auf derQuelle und BereitstellungKlicken Sie auf derSourceFürRepository type (Repository-Typ), wählen SieQuellcode-Repository.

d. FürStellen Sie sich mit GitHub, wählen Sie ein GitHub Konto oder eine Organisation aus, die Sie zuvor verwendet haben, oder wählen SieAdd New. Gehen Sie dann durch den Prozess der Bereitstellung Ihrer GitHub Anmeldeinformationen und wählen Sie ein GitHub-Konto oder eine Organisation, mit der eine Verbindung hergestellt werden soll.

e. FürAblageWählen Sie das Repository aus, das Ihren Anwendungscode enthält.

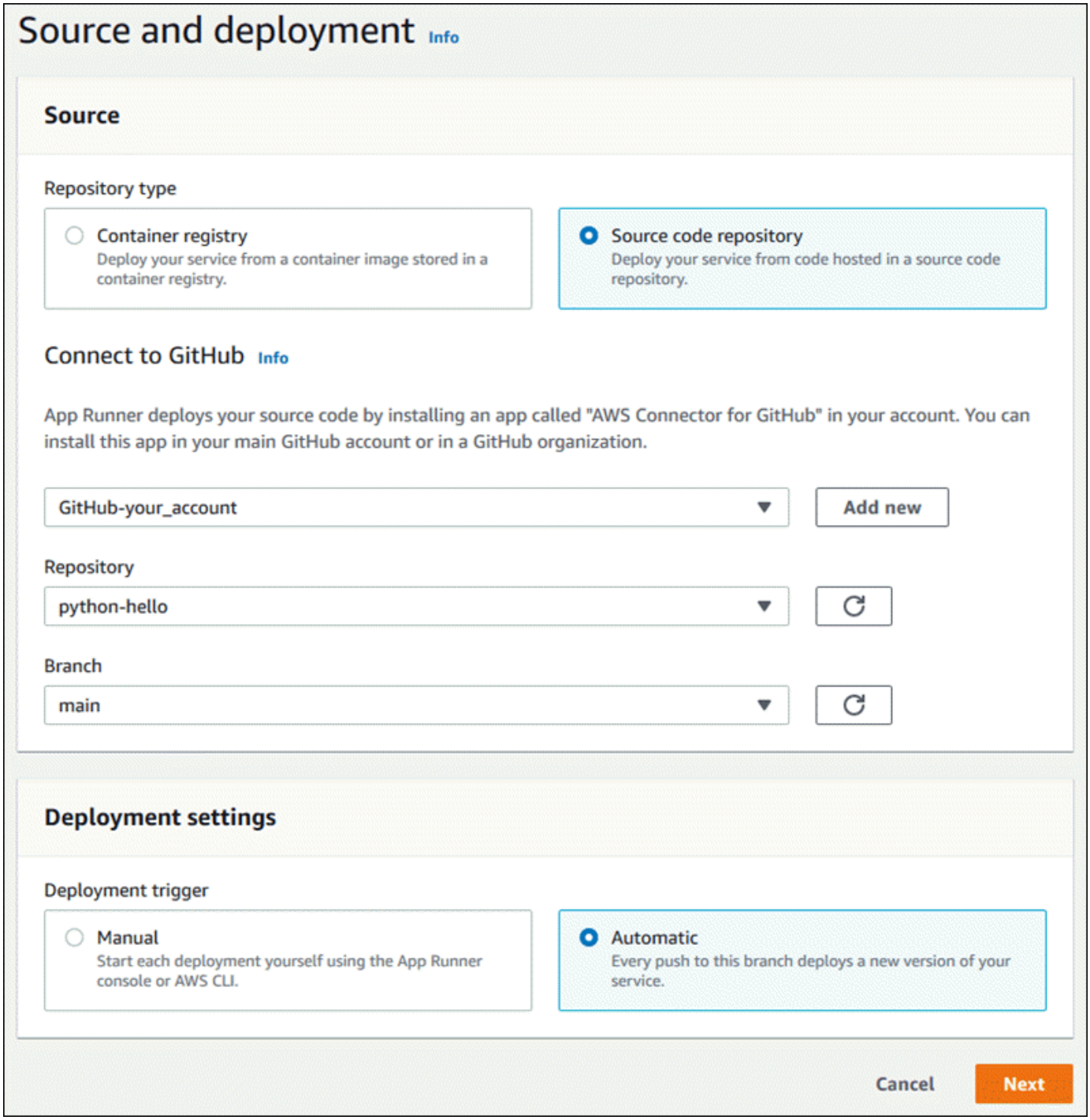
f. FürVerzweigenWählen Sie die Verzweigung aus, Sie bereitstellen möchten.

2. Konfigurieren Sie Ihre Bereitstellungen.

   a. In derBereitstellungseinstellungenWählen Sie unterManuelloder .Automatisch.

      Weitere Informationen zu Bereitstellungsmethoden finden Sie unterthe section called “Bereitstellungsmethoden”.

   b. Wählen Sie Next.



3. Konfigurieren Sie den Anwendungsbuild.

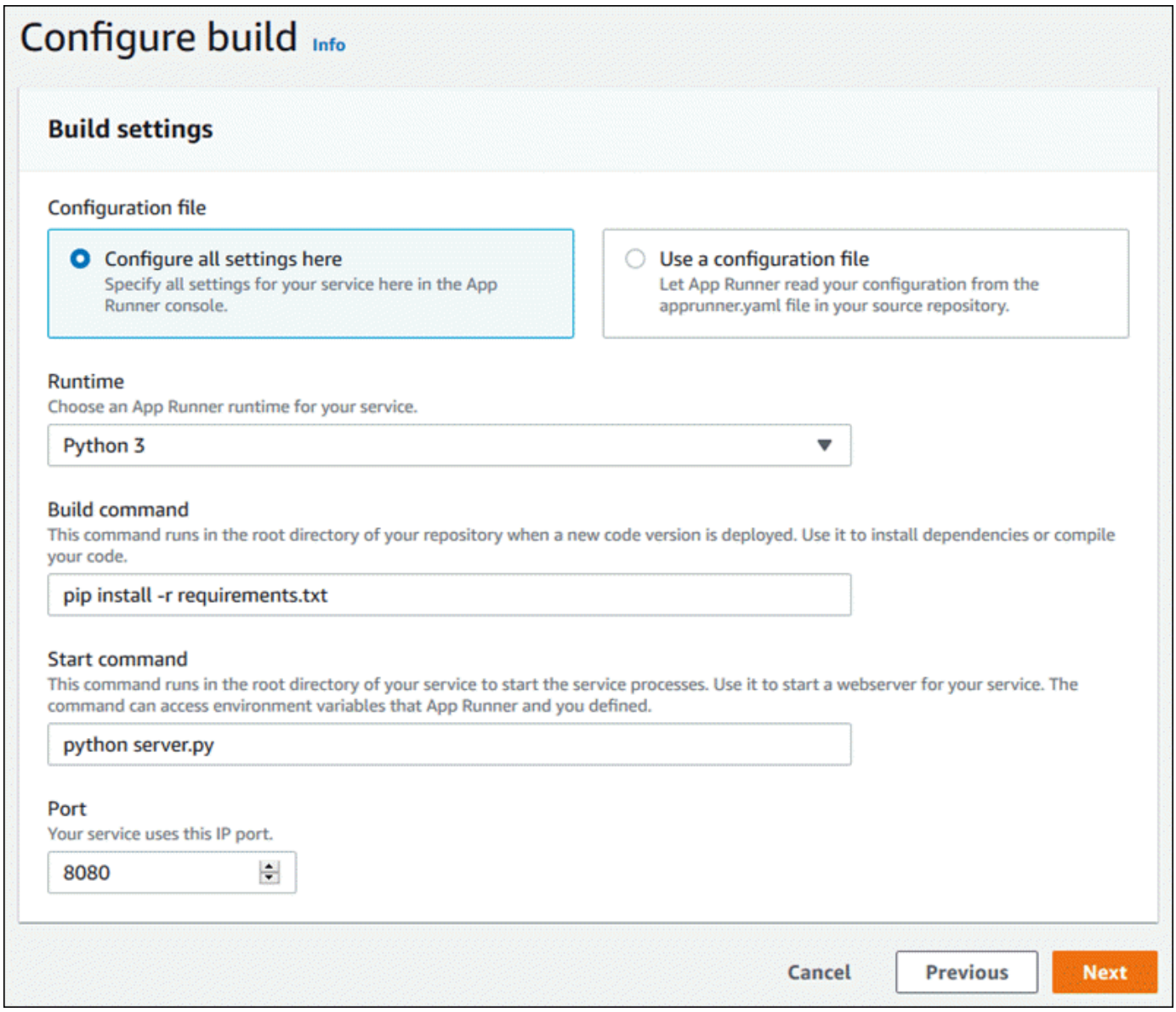
   a. Klicken Sie auf derBuild-KonfigurationFürKonfigurationsdatei, wählen SieHier können Sie alle Einstellungen konfigurieren, wenn Ihr Repository keine App Runner-Konfigurationsdatei enthält, oderVerwendung einer KonfigurationsdateiWenn dies der Fall ist.

> **Note**
>
> Eine App Runner-Konfigurationsdatei ist eine Möglichkeit, Ihre Build-Konfiguration als Teil Ihrer Anwendungsquelle zu verwalten. Wenn Sie einen angeben, liest App Runner einige Werte aus der Datei und lässt Sie sie nicht in der Konsole festlegen.

b. Geben Sie die folgenden Build-Einstellungen an:

- Runtime (Laufzeit)— Wählen Sie eine bestimmte verwaltete Laufzeit für Ihre Anwendung aus.
- Build-Befehl— Geben Sie einen Befehl ein, der Ihre Anwendung aus ihrem Quellcode erstellt. Dies kann ein sprachspezifisches Tool oder ein Skript sein, das mit Ihrem Code bereitgestellt wird.
- Startbefehl— Geben Sie den Befehl ein, mit dem Ihr Webdienst gestartet wird.
- Port— Geben Sie den IP-Port ein, den Ihr Webdienst abhört.

c. Wählen Sie Next.



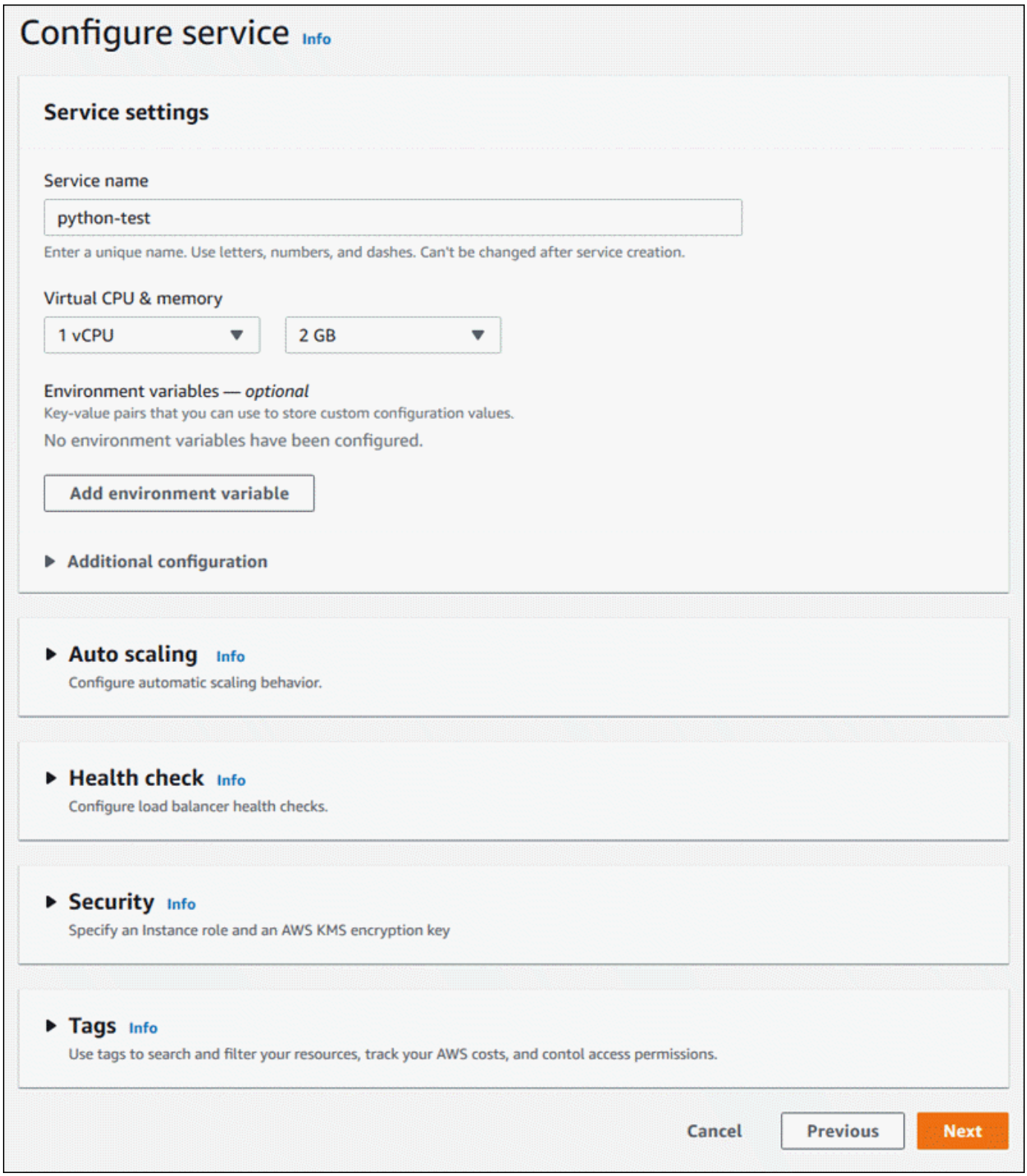
4. Konfigurieren Sie Ihren Dienst.

   a. Klicken Sie auf derKonfigurieren des ServiceKlicken Sie auf derService-EinstellungenGeben Sie einen Service-Namen ein.

> **Note**
>
> Alle anderen Diensteinstellungen sind entweder optional oder verfügen über konsolenbereitgestellte Standardeinstellungen.

b. Optional können Sie weitere Einstellungen ändern oder hinzufügen, um Ihre Anwendungsanforderungen zu erfüllen.

c. Wählen Sie Next.



5. Klicken Sie auf derÜberprüfen und erstellen, überprüfen Sie alle eingegebenen Details, und wählen SieErstellen und Bereitstellen.

Ergebnis: Wenn die Diensterstellung erfolgreich ist, sollte die Konsole das Dienst-Dashboard mit einemServiceübersichtDer neuen Service erstellt.

6. Stellen Sie sicher, dass Ihr Service ausgeführt wird.

   a. Warten Sie auf der Seite „Service-Dashboard“, bis der DienstStatusistAusführen von.

   b. Wählen Sie das SymbolStandarddomäne-Wert — es ist die URL zur Website Ihres Dienstes.

   c. Verwenden Sie Ihre Website und überprüfen Sie, ob sie richtig ausgeführt wird.

Erstellen eines Dienstes aus Code mit der App Runner-API oderAWS CLI

So erstellen Sie einen Dienst mithilfe der App Runner-API oderAWS CLIRufen Sie die`CreateService`API-Aktion. Weitere Informationen und ein Beispiel finden Sie unterCreateService. Wenn Sie zum ersten Mal einen Dienst mit einer bestimmten GitHub Organisation oder einem bestimmten Konto erstellen, rufen SieCreateConnection. Dadurch wird eine Verbindung zwischen App Runner und der GitHub Organisation oder dem Konto hergestellt. Weitere Informationen zu App Runner-Verbindungen finden Sie unterthe section called “Verbindungen”.

Ihre Diensterstellung beginnt, wenn der Anruf eine erfolgreiche Antwort mit einem-ServiceObjekt anzeigen`"Status": "CREATING"`.

Ein Beispiel finden Sie unterErstellen eines Quellcode-Repository-DienstesimAWS App Runner-API-Referenz

Erstellen eines Services über ein Amazon ECR-Image

In den folgenden Abschnitten wird gezeigt, wie man einen App Runner-Service erstellt, wenn es sich bei der Quelle um ein Container-Image handelt, das inAmazon ECR. Amazon ECR ist einAWSService-Service. Um einen Service basierend auf einem Amazon ECR-Image zu erstellen, stellen Sie App Runner daher eine Zugriffsrolle bereit, die die erforderlichen Amazon ECR-Aktionsberechtigungen enthält.

> **Note**
>
> Eine Zugriffsrolle ist nicht erforderlich, wenn Ihr Bild in Amazon ECR Public gespeichert ist, wo Bilder öffentlich verfügbar sind.

Während der Diensterstellung startet App Runner einen Dienst, der eine Containerinstanz des bereitgestellten Images ausführt. In diesem Fall gibt es keine Build-Phase.

Themen

- Erstellen eines Dienstes über ein Image mit der App Runner-Konsole
- Erstellen eines Dienstes aus einem Image mit der App Runner-API oderAWS CLI

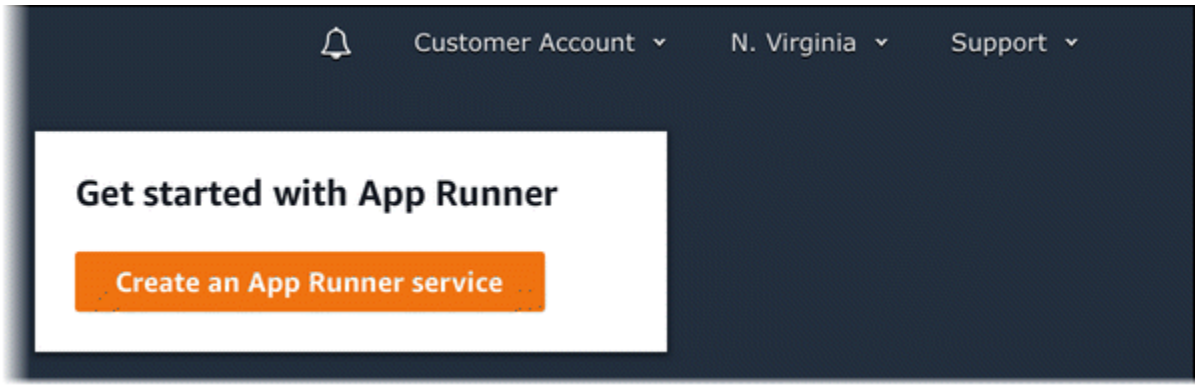
Erstellen eines Dienstes über ein Image mit der App Runner-Konsole

So erstellen Sie mit der Konsole einen App Runner-Dienst

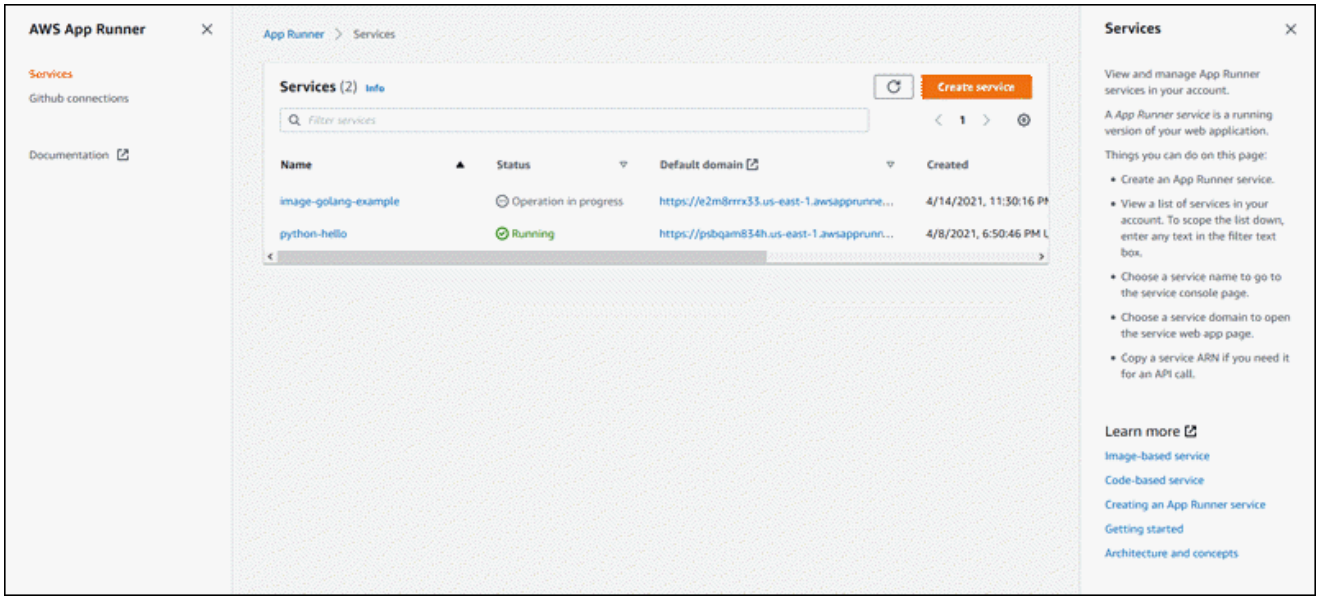
1. Konfigurieren Sie den Quellcode.

   a. Öffnen SieApp-Runner-Konsoleund in derRegionenaus, wählen Sie IhreAWS-Region.

   b. Wenn das SymbolAWS-Kontonoch keine App Runner-Dienste hat, wird die Konsolen-Startseite angezeigt. Klicken Sie aufErstellen eines App Runner-Service.



   Wenn das SymbolAWS-Kontoverfügt über bereits vorhandene Dienste, dieServicesEine Seite mit einer Liste Ihrer Services wird angezeigt. Wählen Sie Create service.



c. Klicken Sie auf derQuelle und BereitstellungKlicken Sie auf derSourceFürRepository type (Repository-Typ), wählen SieContainer-Registrierung.

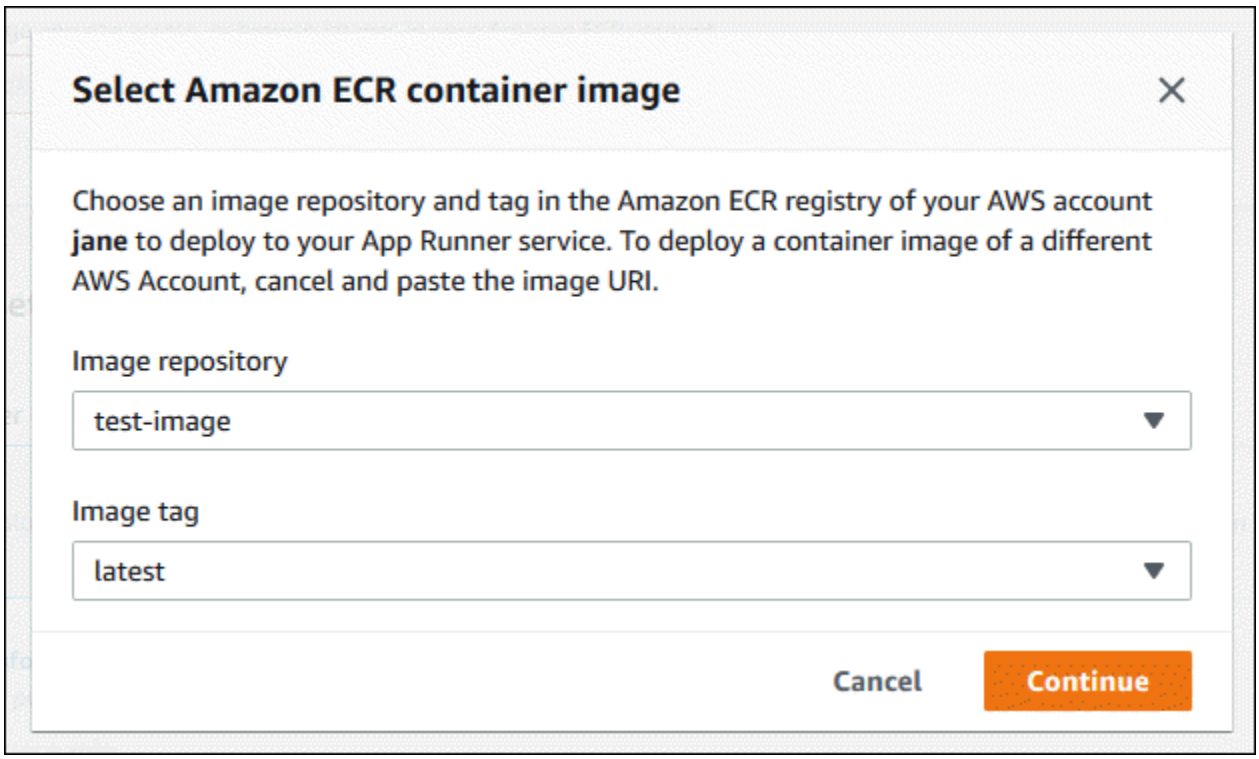
d. FürAnbieterden Anbieter aus, in dem Ihr Bild gespeichert ist:

- Amazon ECR— Ein privates Bild, das in Amazon ECR in IhremAWS-Konto.
- Public Amazon ECR— Ein öffentlich lesbares Bild, das in Amazon ECR Public gespeichert ist.

e. FürContainer-Image-URI, wählen SieDurchsuchen.

f. In derAmazon ECR-Containerbild auswählenFürImage-Repositorydas Repository aus, das Ihr Image enthält.

g. FürImage-MarkierungWählen Sie das spezifische Image-Tag aus, das Sie bereitstellen möchten, z. B.latestWählen Sie und anschließend die OptionContinue.



2. Konfigurieren Sie Ihre Bereitstellungen.

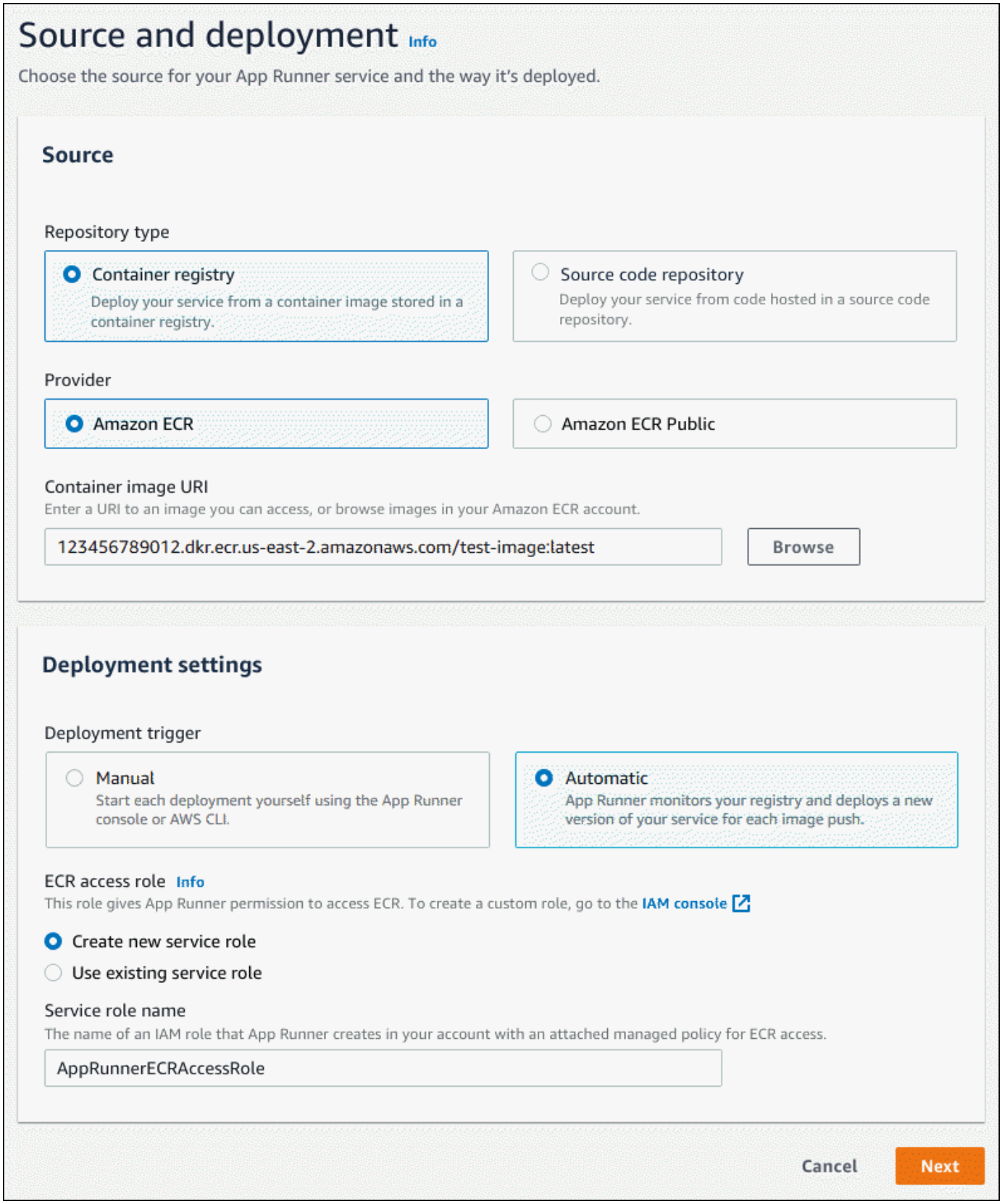
   a. In derBereitstellungseinstellungenWählen Sie unterManuelloder .Automatisch.

      Weitere Informationen zu Bereitstellungsmethoden finden Sie unterthe section called “Bereitstellungsmethoden”.

      > **Note**
      >
      > App Runner unterstützt keine automatische Bereitstellung von Amazon ECR Public Images.

   b. [Amazon ECRFür provider]ECR-ZugriffsrolleWählen Sie eine vorhandene Dienstrolle in Ihrem Konto aus oder wählen Sie aus, ob Sie eine neue Rolle erstellen möchten. Wenn Sie die manuelle Bereitstellung verwenden, können Sie auch die IAM-Benutzerrolle zum Zeitpunkt der Bereitstellung verwenden.

   c. Wählen Sie Next.


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

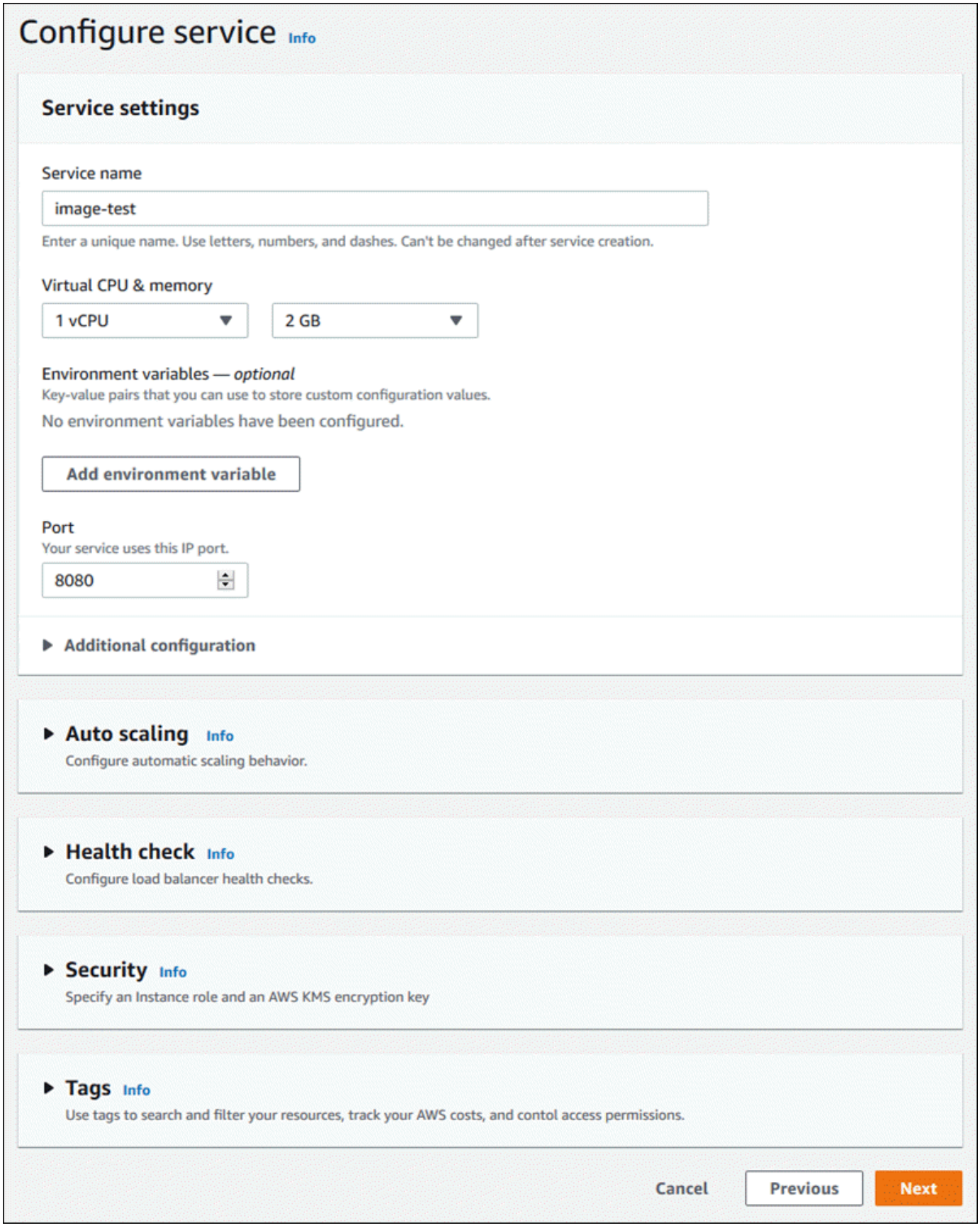
3. Konfigurieren Sie Ihren Dienst.

   a. Klicken Sie auf derKonfigurieren des ServiceKlicken Sie auf derService-Einstellungeneinen Dienstnamen und den IP-Port ein, den Ihre Dienstwebsite überwacht.

   > **Note**
   >
   > Alle anderen Diensteinstellungen sind entweder optional oder verfügen über konsolenbereitgestellte Standardeinstellungen.

   b. (Optional) Ändern oder fügen Sie andere Einstellungen hinzu, die den Anforderungen Ihrer Anwendung entsprechen.

   c. Wählen Sie Next.


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. Klicken Sie auf derÜberprüfen und erstellenÜberprüfen Sie alle eingegebenen Details, und wählen Sie dannErstellen und Bereitstellen.

   Ergebnis: Wenn die Diensterstellung erfolgreich ist, sollte die Konsole das Dienst-Dashboard mit einemServiceübersichtDer neuen Service erstellt.

5. Stellen Sie sicher, dass Ihr Service ausgeführt wird.

   a. Warten Sie auf der Seite „Service-Dashboard“, bis der DienstStatusistAusführen von.

   b. Wählen Sie das SymbolStandarddomäne-Wert — es ist die URL zur Website Ihres Dienstes.

   c. Verwenden Sie Ihre Website und überprüfen Sie, ob sie richtig ausgeführt wird.

Erstellen eines Dienstes aus einem Image mit der App Runner-API oderAWS CLI

So erstellen Sie einen Dienst mithilfe der App Runner-API oderAWS CLIRufen Sie dieCreateServiceAPI-Aktion.

Ihre Diensterstellung beginnt, wenn der Anruf eine erfolgreiche Antwort mit einem-ServiceObjekt anzeigen`"Status": "CREATING"`.

Ein Beispiel finden Sie unterErstellen eines Quellbild-Repository-DienstesimAWS App Runner-API-Referenz

Wenn Service-Erstellung fehlschlägt

Wenn Ihr Versuch, einen App Runner-Dienst zu erstellen, fehlschlägt, zeigt der Dienst den Status`CREATE_FAILED`(Erstellen fehlgeschlagenauf der Konsole).

Ihr Versuch, einen Dienst zu erstellen, schlägt möglicherweise aufgrund von Problemen im Anwendungscode, im Buildprozess oder in der Konfiguration fehl, da Sie Ressourcenkontingente erreicht haben, oder aufgrund temporärer Probleme mit dem zugrunde liegendenAWS-Dienste, die Ihr Dienst verwenden muss. Um einen Fehler zu beheben, empfehlen wir, dass Sie die folgenden Aktionen ausführen. Lesen Sie zuerst die Dienstereignisse und Protokolle, um herauszufinden, was den Fehler verursacht hat. Nehmen Sie als Nächstes alle erforderlichen Änderungen an Ihrem Code oder Ihrer Konfiguration vor. Löschen Sie zuletzt einen oder mehrere Dienste, wenn Sie Ihr Dienstkontingent erreicht haben. Versuchen Sie dann, nachdem Sie alle diese Schritte ausgeführt haben, erneut, den Dienst zu erstellen.

> **Important**
>
> Der ausgefallene Dienst ist nicht verwendbar. Sie werden keine zusätzlichen Gebühren für sie über den ursprünglichen Erstellungsversuch hinaus anfallen. App Runner löscht den ausgefallenen Dienst jedoch nicht automatisch und zählt weiterhin zu Ihrem Dienstkontingent. Wenn Sie mit der Analyse des Fehlers fertig sind, stellen Sie sicher, dass Sie den fehlgeschlagenen Dienst löschen.

So stellen Sie eine neue Anwendungsversion in App Runner bereit

Wenn SieErstellen eines ServiceinAWS App Runnerkonfigurieren Sie eine Anwendungsquelle — ein Container-Image oder ein Quell-Repository. App Runner stellt Ressourcen bereit, um Ihren Dienst auszuführen, und stellt Ihre Anwendung für sie bereit.

In diesem Thema wird beschrieben, wie Sie Ihre Anwendungsquelle erneut auf Ihrem App Runner-Dienst bereitstellen können, wenn eine neue Version verfügbar ist. Dies kann eine neue Bildversion im Bild-Repository oder ein neues Commit im Code-Repository sein. App Runner bietet zwei Methoden für die Bereitstellung in einem Dienst:AutomatischundManuell.

Bereitstellungsmethoden

App Runner stellt Ihnen die folgenden Methoden zur Verfügung, mit denen Sie steuern können, wie Anwendungsbereitstellungen initiiert werden.

Automatische Bereitstellung

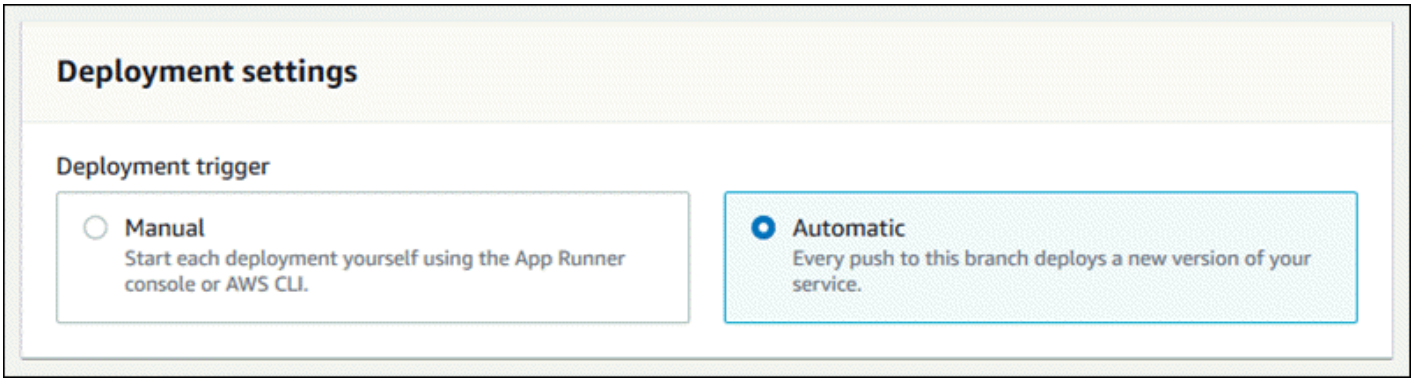
Verwenden Sie die automatische Bereitstellung, wenn Sie ein CICD-Verhalten (Continuous Integration and Deployment) für Ihren Service wünschen. App Runner überwacht Ihr Bild- oder Code-Repository. Immer wenn Sie eine neue Image-Version in Ihr Bild-Repository oder ein neues Commit in Ihr Code-Repository übertragen, stellt App Runner diese automatisch für Ihren Dienst bereit, ohne dass Sie weitere Aktionen auf Ihrer Seite ausführen müssen.

Manuelle Bereitstellung

Verwenden Sie die manuelle Bereitstellung, wenn Sie jede Bereitstellung explizit für Ihren Dienst initiieren möchten. Sie initiieren eine Bereitstellung, wenn das Repository, das Sie für Ihren Dienst konfiguriert haben, über eine neue Version verfügt, die Sie bereitstellen möchten. Weitere Informationen finden Sie unter the section called “Manuelle Bereitstellung”.

Sie können die Bereitstellungsmethode für Ihren Service wie folgt konfigurieren:

- Konsole— Für einen neuen Dienst, den Sie erstellen, oder für einen vorhandenen Service, finden Sie imBereitstellungseinstellungen-Abschnitt imQuelle und Bereitstellung-KonfigurationsseiteManuelloder .Automatisch.



- -API oderAWS CLI— Bei einem Aufruf derCreateServiceoder .UpdateService-Aktion, legen Sie die`AutoDeploymentsEnabled`-Mitglied derSourceConfiguration-Parameter`False`für die manuelle Bereitstellung oder`True`für die automatische Bereitstellung.

Manuelle Bereitstellung

Bei der manuellen Bereitstellung müssen Sie jede Bereitstellung explizit für Ihren Dienst initiieren. Wenn Sie eine neue Version Ihres Anwendungsabbilds oder Codes bereit für die Bereitstellung haben, können Sie in den folgenden Abschnitten erfahren, wie Sie eine Bereitstellung mit der Konsole und der API durchführen.

Bereitstellen einer Anwendungsversion mit der App Runner-Konsole

So stellen Sie die Bereitstellung mithilfe der App Runner -Konsole bereit

1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im NavigationsbereichServices, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Dienst-Dashboard mit einemÜbersicht über den Service.
3. Wählen Sie Deploy (Bereitstellen) aus.

   Ergebnis: Die Bereitstellung der neuen Version wird gestartet. Auf der Seite „Service-Dashboard“ wird der ServiceStatusÄnderungen anWird ausgeführt.

4. Warten Sie, bis die Bereitstellung beendet ist. Auf der Seite „Service-Dashboard“ wird der ServiceStatussollte zurück zuAusführen von.
5. Um zu überprüfen, ob die Bereitstellung erfolgreich ist, wählen Sie auf der Seite „Service-Dashboard“ die OptionStandarddomäne-Wert — es ist die URL zur Website Ihres Dienstes. Überprüfen Sie Ihre Webanwendung oder interagieren Sie mit dieser und überprüfen Sie Ihre Versionsänderung.

Bereitstellen einer Anwendungsversion mithilfe der App Runner-API oderAWS CLI

So stellen Sie die Bereitstellung mithilfe der App Runner-API oderAWS CLISie haben dieStartDeployment-API-Aktion. Der einzige zu übergebende Parameter ist Ihr Service ARN. Sie haben Ihren Anwendungsquellspeicherort bereits konfiguriert, als Sie den Dienst erstellt haben, und App Runner kann die neue Version finden. Ihre Bereitstellung wird gestartet, wenn der Anruf eine erfolgreiche Antwort zurückgibt.

Konfigurieren eines App Runner-Dienstes

Wenn SieErstellen einesAWS App Runner-Servicefestlegen, legen Sie verschiedene Konfigurationswerte fest. Sie können einige dieser Konfigurationseinstellungen ändern, nachdem Sie den Service erstellt haben. Andere Einstellungen können nur beim Erstellen des Dienstes angewendet werden und können danach nicht geändert werden. In diesem Thema wird die Konfiguration Ihres Dienstes mithilfe der App Runner-API, der App Runner-Konsole und einer App Runner-Konfigurationsdatei erläutert.

Konfigurieren Sie Ihren Dienst mit der App Runner-API oderAWS CLI

Die API definiert, welche Einstellungen nach der Service-Erstellung geändert werden können. In der folgenden Liste werden die relevanten Aktionen, Typen und Einschränkungen erläutert.

- UpdateServiceaction — Kann nach der Erstellung aufgerufen werden, um einige Konfigurationseinstellungen zu aktualisieren.
  - Kann aktualisiert werden.— Sie können die Einstellungen im Dialogfeld`SourceConfiguration`,`InstanceConfiguration`, und`HealthCheckConfiguration`-Parameter. Jedoch in`SourceConfiguration`, können Sie Ihren Quelltyp nicht von Code zu Bild oder umgekehrt umschalten. Sie müssen denselben Repository-Parameter angeben, den Sie angegeben haben, als Sie den Service erstellt haben. Es ist entweder`CodeRepository`oder .`ImageRepository`.

Sie können auch aktualisieren.`AutoScalingConfigurationArn`, der ARN der Konfigurationsressource für die automatische Skalierung, die dem Dienst zugeordnet ist.

- Kann nicht aktualisiert werden— Sie können die`ServiceName`und`EncryptionConfiguration`-Parameter, die in derCreateServiceAktion Sie können nicht mehr geändert werden, nachdem sie erstellt wurden. DieUpdateServiceenthält diese Parameter nicht.
- API vs.— Sie können die`ConfigurationSource`-Parameter desCodeConfigurationTyp (wird für Quellcode-Repositories als Teil von`SourceConfiguration`) zu`Repository`. In diesem Fall ignoriert App Runner die Konfigurationseinstellungen in`CodeConfigurationValues`und liest diese Einstellungen aus einemKonfigurationsdatei-Repository. Wenn Sie`ConfigurationSource`auf`API`, ruft App Runner alle Konfigurationseinstellungen aus dem API-Aufruf ab und ignoriert die Konfigurationsdatei, selbst wenn eine vorhanden ist.

- TagResourceaction — Kann aufgerufen werden, nachdem der Dienst erstellt wurde, um dem Dienst Tags hinzuzufügen oder Werte vorhandener Tags zu aktualisieren.
- UntagResourceaction — Kann aufgerufen werden, nachdem Ihr Dienst erstellt wurde, um Tags aus dem Dienst zu entfernen.

Konfigurieren Sie Ihren Dienst mit der App Runner-Konsole

Die Konsole verwendet die App Runner-API, um Konfigurationsupdates anzuwenden. Die Aktualisierungsregeln, die die API erlegt, wie im vorherigen Abschnitt definiert, bestimmen, was Sie mit der Konsole konfigurieren können. Einige Einstellungen, die während der Diensterstellung verfügbar waren, können später nicht geändert werden. Wenn Sie sich darüber hinaus für die Verwendung einesKonfigurationsdatei, werden zusätzliche Einstellungen in der Konsole ausgeblendet und App Runner liest sie aus der Datei.

So konfigurieren Sie Ihren Dienst

1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im Navigationsbereich und dann aus.Services, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Service-Dashboard mit einemService-Übersicht.
3. Wählen Sie auf der Service-Dashboard-Seite die OptionKonfiguration-Registerkarte.

Ergebnis: Die Konsole zeigt die aktuellen Konfigurationseinstellungen Ihres Dienstes in mehreren Abschnitten an: Quelle und Bereitstellung,Build-Konfiguration, undKonfigurieren des Service.

4. Um Einstellungen in einer beliebigen Kategorie zu aktualisieren, wählen SieBearbeiten.
5. Nehmen Sie auf der Konfigurationsbearbeitungsseite die gewünschten Änderungen vor und wählen Sie dannSpeichern Sie die Änderungen.

Konfigurieren Sie Ihren Dienst mit einer App Runner-Konfigurationsdatei

Wenn Sie einen App Runner-Dienst erstellen oder aktualisieren, können Sie App Runner anweisen, einige Konfigurationseinstellungen aus einer Konfigurationsdatei zu lesen, die Sie als Teil Ihres Quell-Repositorys bereitstellen. Auf diese Weise können Sie die Einstellungen, die sich auf Ihren Quellcode beziehen, unter Quellcodeverwaltung, zusammen mit dem Code selbst verwalten. Die Konfigurationsdatei enthält auch bestimmte erweiterte Einstellungen, die Sie nicht über die Konsole oder die API festlegen können. Weitere Informationen finden Sie unter App Runner-Konfigurationsdatei.

Verwalten von App Runner-Verbindungen

Wenn SieSo erstellen Sie einen ServiceinAWS App Runnerkonfigurieren Sie eine Anwendungsquelle — ein Container-Image oder ein Quell-Repository, das bei einem Provider gespeichert ist. Wenn ein Repository, das bei einem Drittanbieter gespeichert ist, privat ist (nicht öffentlich lesbar), muss App Runner eine authentifizierte und autorisierte Verbindung mit dem Anbieter herstellen. Dann kann App Runner Ihr Repository lesen und für Ihren Service bereitstellen. App Runner erfordert keinen Verbindungsaufbau, wenn Sie einen Dienst erstellen, der auf Code zugreift, der in IhremAWS-Kontooder an einem öffentlichen Code-Speicherort.

App Runner verwaltet Verbindungsinformationen in einer Ressource, dieVerbindung. App Runner erfordert eine Verbindungsressource, wenn Sie einen Dienst erstellen, der Verbindungsinformationen von Drittanbietern benötigt. Im Folgenden finden Sie einige wichtige Informationen über Verbindungen:

- Anbieter— App Runner benötigt derzeit Verbindungsressourcen mitGitHub.
- Freigegeben— Sie können eine Verbindungsressource verwenden, um mehrere App Runner-Dienste zu erstellen, die dasselbe Repository-Anbieterkonto verwenden.
- Ressourcenmanagement— In App Runner können Sie Verbindungen erstellen und löschen. Sie können jedoch keine bestehende Verbindung ändern.

- Ressourcenkontingente— Verbindungsressourcen verfügen über ein festgelegtes Kontingent, das IhremAWS-KontoIn jedemAWS-Region. Wenn Sie dieses Kontingent erreichen, müssen Sie möglicherweise eine Verbindung löschen, bevor Sie eine Verbindung zu einem neuen Anbieterkonto herstellen können. Sie können eine -Verbindung mit App Runnerconsoleoder .API. Weitere Informationen finden Sie unter the section called “App Runner Ressourcenkontingente”.

Verwalten von Verbindungen über die App Runner-Konsole

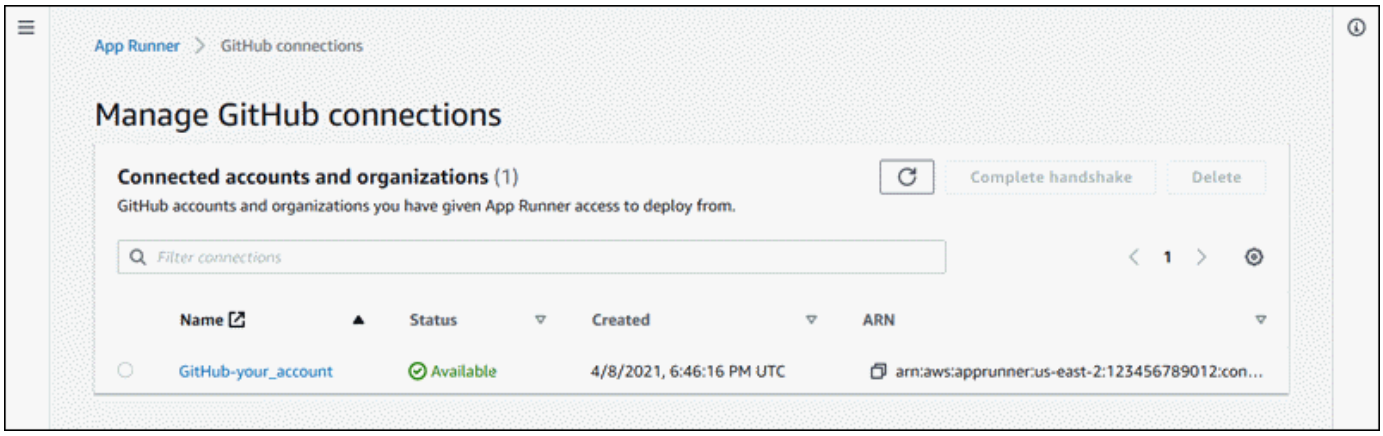
Wenn Sie die App Runner-Konsole zumSo erstellen Sie einen ServiceSie geben Verbindungsdetails an. Sie müssen keine Verbindungsressource explizit erstellen. In der Konsole können Sie eine Verbindung zu einem GitHub Konto herstellen, mit dem Sie zuvor eine Verbindung hergestellt haben, oder eine Verbindung zu einem neuen Konto herstellen. Bei Bedarf erstellt App Runner eine Verbindungsressource für Sie. Für eine neue Verbindung verlangen einige Anbieter (z. B. GitHub), dass Sie einen Authentifizierungs-Handshake abschließen, bevor Sie die Verbindung verwenden können. Die Konsole führt Sie durch diesen Prozess.

Die Konsole verfügt auch über eine Seite zum Verwalten Ihrer vorhandenen Verbindungen. Sie können den Authentifizierungs-Handshake für eine Verbindung abschließen, wenn Sie dies bei der Erstellung des Dienstes nicht getan haben. Sie können Verbindungen, die Sie nicht mehr verwenden, auch löschen. Das folgende Verfahren zeigt, wie Sie GitHub Verbindungen verwalten können.

So verwalten Sie GitHub Verbindungen in Ihrem Konto

1. Öffnen SieApp Runner Konsoleund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im NavigationsbereichGitHub Verbindungen.

   In der Konsole wird eine Liste der GitHub Verbindungen in Ihrem Konto angezeigt.



3. Sie können jetzt eine der folgenden Aktionen für eine beliebige Verbindung in der Liste ausführen:

   - GitHub Konto oder Organisation öffnen— Wählen Sie den Namen der Verbindung aus.
   - Handshake für die Authentifizierung— Wählen Sie die -Verbindung und die OptionenHandshake. Die Konsole führt Sie durch den Authentifizierungs-Handshake-Prozess.
   - Verbindung löschen— Wählen Sie die -Verbindung und die OptionenLöschen. Folgen Sie den Anweisungen auf der Löschaufforderung.

Verwalten von Verbindungen über die App Runner-API oderAWS CLI

Sie können zur Verwaltung Ihrer Verbindungen die folgenden -API-Aktionen verwenden.

- CreateConnection— Erstellt eine Verbindung zu einem Repository-Provider-Konto. Nachdem die Verbindung erstellt wurde, müssen Sie den Authentifizierungshandshake manuell über die App Runner-Konsole abschließen. Dieser Prozess wird im vorherigen Abschnitt erläutert.
- ListenVerbindungen— Gibt eine Liste der App Runner Verbindungen zurück, die IhremAWS-Konto.
- DeleteConnection— Löscht eine Verbindung. Möglicherweise müssen Sie unnötige Verbindungen löschen, wenn Sie das Verbindungskontingent für IhreAWS-Konto.

Automatische Skalierung von App Runner verwalten

AWS App Runnerskaliert Compute-Ressourcen (Instanzen) für Ihre App Runner-Anwendung automatisch nach oben oder unten. Die automatische Skalierung bietet eine angemessene Anforderungsbehandlung bei hohem eingehenden Datenverkehr und senkt Ihre Kosten, wenn der Datenverkehr verlangsamt wird. Sie können einige Parameter konfigurieren, um das automatische Skalierungsverhalten für Ihren Dienst anzupassen.

App Runner verwaltet die Einstellungen für die automatische Skalierung in einer Ressource namensAutoScalingKonfiguration. Sie können eine automatische Skalierungskontingente bereitstellen, wenn Sie einen Service erstellen oder aktualisieren. Die App Runner-Konsole erstellt eine für Sie, wenn Sie einen neuen App Runner-Dienst erstellen. Die Bereitstellung einer automatischen Skalierungskonfiguration ist optional. Wenn Sie dies nicht tun, stellt App Runner eine standardmäßige automatische Skalierungskonfiguration mit empfohlenen Werten zur Verfügung.

Eine automatische Skalierungskonfiguration verfügt über eine-Nameund eine numericÄnderung. Mehrere Revisionen einer Konfiguration haben denselben Namen und unterschiedliche

Revisionsnummern. Sie können verschiedene Konfigurationsnamen für verschiedene automatische Skalierungsszenarien verwenden, z. B. Hochverfügbarkeit oder niedrige Kosten. Für jeden Namen können Sie mehrere Revisionen hinzufügen, um die Einstellungen für ein bestimmtes Szenario zu optimieren.

Im Folgenden finden Sie einige wichtige Informationen zu Konfigurationen für die automatische Skalierung:

- Einstellungen— Das können Sie konfigurieren:
  - Max. gleichzeitig— Die maximale Anzahl gleichzeitiger Anforderungen, die eine Instanz verarbeitet. Wenn die Anzahl der gleichzeitigen Anforderungen dieses Kontingent überschreitet, skaliert App Runner den Dienst.
  - Max Größe— Die maximale Anzahl von Instances, auf die Ihr Service skaliert. In den meisten Fällen dient diese Anzahl von Instanzen aktiv Datenverkehr für Ihren Dienst.
  - Min Größe— Die minimale Anzahl von Instances, die App Runner für Ihren Service bereitstellt. Der Dienst verfügt immer über mindestens diese Anzahl bereitgestellter Instanzen. Einige von ihnen dienen aktiv dem Verkehr. Der Rest von ihnen (bereitgestellte und inaktive Instances) steht als kostengünstige Rechenkapazitätsreserve zur Verfügung, die schnell aktiviert werden kann. Sie zahlen für die Speicherauslastung aller bereitgestellten Instances. Sie zahlen nur für die CPU-Auslastung der aktiven Teilmenge.

    App Runner verdoppelt vorübergehend die Anzahl der bereitgestellten Instances während der Bereitstellungen, um die gleiche Kapazität für alten und neuen Code beizubehalten.
- Änderungen— Die erste Konfiguration, die Sie mit einem Namen erstellen, erhält die Revisionsnummer 1. Nachfolgende Konfigurationen mit dem gleichen Namen erhalten fortlaufende Revisionsnummern (beginnend mit 2). Sie können Ihren App Runner-Dienst mit einer bestimmten Version für die automatische Skalierung oder mit der neuesten Version der Konfiguration verknüpfen.
- Freigegeben— Sie können eine einzelne automatische Skalierungs-Konfigurationsressource für mehrere App Runner-Dienste freigeben. Dies ist nützlich, wenn sie ähnliche Skalierungsanforderungen haben. Insbesondere können Sie mehrere Dienste so konfigurieren, dass alle die neueste Version einer Konfiguration verwenden, indem Sie den Konfigurationsnamen angeben, jedoch keine Revision angeben. Auf diese Weise erhält jeder der Dienste, die Sie auf diese Weise konfiguriert haben, automatische Skalierungskonfigurationsupdates, wenn Sie den Dienst aktualisieren. Weitere Informationen zu Konfigurationsänderungen finden Sie unterthe section called “Konfiguration”.

- Ressourcenmanagement— Sie können App Runner verwenden, um automatische Skalierungskonfigurationen zu erstellen und zu löschen. Sie können eine Konfiguration nicht direkt aktualisieren. Stattdessen können Sie eine neue Version eines vorhandenen Konfigurationsnamens erstellen, um die Konfiguration effektiv zu aktualisieren.

  > **Note**
  >
  > Derzeit können Sie nur eine Konfiguration mit einer einzigen Revision in der App Runner-Konsole erstellen. Um weitere Revisionen zu erstellen und Konfigurationen zu löschen, verwenden Sie die App RunnerAPI.

- Ressourcenkontingente— Es gibt festgelegte Kontingente für die Anzahl der eindeutigen Konfigurationsnamen und Revisionen, die Sie für die automatische Skalierung Konfigurationsressourcen in jedemAWS-Region. Wenn Sie diese Kontingente erreichen, müssen Sie entweder einen Konfigurationsnamen oder zumindest einige seiner Revisionen löschen, bevor Sie weitere erstellen können. App Runner verwendenAPI, um sie zu löschen. Weitere Informationen finden Sie unter the section called “App Runner Ressourcenkontingente”.

Verwalten der automatischen Skalierung mit der App Runner-Konsole

Wenn SieErstellen eines Servicesin der App Runner-Konsole können Sie die Standardkonfiguration für die automatische Skalierung oder eine benutzerdefinierte Konfiguration verwenden. Um eine benutzerdefinierte Konfiguration zu verwenden, wählen Sie entweder eine vorhandene Konfiguration aus oder geben Sie einen neuen Namen und Einstellungen ein. Wenn es sich um eine neue Konfiguration handelt, erstellt App Runner eine neue Konfigurationsressource für die automatische Skalierung und ordnet sie dann Ihrem neuen Dienst zu.

Verwalten Sie die automatische Skalierung mithilfe der App Runner-API oderAWS CLI

Sie können zur Verwaltung Ihrer Auto Scaling-Konfigurationen die folgenden -API-Aktionen verwenden.

- ErstellenAutoScalingKonfiguration— Erstellt eine neue automatische Skalierungskonfiguration oder eine Revision für eine vorhandene Konfiguration.
- ListAutoScalingKonfigurationen— Gibt eine Liste der automatischen Skalierungskonfigurationen zurück, die mit IhremAWS-Konto, mit zusammenfassenden Informationen.

- DescribeBeautoScalingConfiguration— Gibt eine vollständige Beschreibung einer automatischen Skalierungskonfiguration zurück.
- DeleteAutoScalingConfiguration— Löscht eine automatische Skalierungskonfiguration. Sie können eine bestimmte Revision oder die letzte aktive Revision löschen. Möglicherweise müssen Sie unnötige automatische Skalierungskonfigurationen löschen, wenn Sie das Konfigurationskontingent für die automatische Skalierung IhrerAWS-Konto.

Verwalten benutzerdefinierter Domänennamen für einen App Runner-Dienst

Wenn Sie einenAWS App Runner-Dienst zuweist App Runner ihm einen Domänennamen zu. Dies ist eine Subdomain in der`awsapprunner.com`-Domain, die im Besitz von App Runner ist. Es kann verwendet werden, um auf die Webanwendung zuzugreifen, die in Ihrem Dienst ausgeführt wird.

Wenn Sie einen Domänennamen besitzen, können Sie diesen mit Ihrem App Runner-Dienst verknüpfen. Nachdem App Runner Ihre neue Domain überprüft hat, kann sie zusätzlich zur App Runner-Domain für den Zugriff auf Ihre Anwendung verwendet werden. Sie können bis zu fünf benutzerdefinierte Domänen zuordnen.

> **Note**
>
> Sie können optional die`www`Subdomain Ihrer Domain. Dies wird derzeit jedoch nur in der API unterstützt. Die App Runner-Konsole unterstützt diese nicht.

Wenn Sie eine benutzerdefinierte Domäne mit Ihrem Dienst verknüpfen, stellt App Runner Ihnen einen Satz von Zertifikatüberprüfungsdatensätzen zur Verfügung. Fügen Sie sie Ihrem Domain Name System (DNS) hinzu, damit App Runner überprüfen kann, ob Sie die Domäne besitzen oder steuern. Fügen Sie außerdem die CNAME- oder ALIAS-Einträge zu Ihrem DNS hinzu, um auf die App Runner-Domäne zuzugreifen. Sie müssen einen Datensatz für die benutzerdefinierte Domäne und einen anderen für die`www`Subdomain, wenn Sie diese Option gewählt haben. Warten Sie dann, bis der benutzerdefinierte DomänenstatusAktivin der App Runner-Konsole. Dies dauert in der Regel einige Minuten (kann aber 24-48 Stunden dauern). Zu diesem Zeitpunkt wird Ihre benutzerdefinierte Domäne validiert, und App Runner leitet den Datenverkehr von dieser Domäne an Ihre Webanwendung weiter.

Sie können eine Domäne angeben, die mit Ihrem App Runner-Dienst verknüpft werden soll:

- Eine Stammdomäne— Zum Beispiel,`example.com`. Sie können optional`www.example.com`auch als Teil derselben Operation.
- Eine Unterdomäne— Zum Beispiel,`login.example.com`oder .`admin.login.example.com`. Optional können Sie die`www`-Domäne als Teil desselben Vorgangs.
- Platzhalter— Zum Beispiel,`*.example.com`. Sie können die`www`Option in diesem Fall. Sie können einen Platzhalter nur als unmittelbare Unterdomäne einer Stammdomäne und nur für sich selbst angeben (dies sind keine gültigen Spezifikationen:`login*.example.com`,`*.login.example.com`) enthalten. Diese Platzhalterspezifikation verknüpft alle unmittelbaren Unterdomänen und verknüpft nicht die Stammdomäne selbst (die Stammdomäne müsste in einem separaten Vorgang zugeordnet werden).

Eine spezifischere Domänenzuordnung überschreibt eine weniger spezifische Domänenzuordnung. Beispiel,`login.example.com`Überschreibungen`*.example.com`. Das Zertifikat und der CNAME der spezifischeren Zuordnung werden verwendet.

Das folgende Beispiel zeigt, wie Sie mehrere benutzerdefinierte Domänennamen verwenden können:

1. Zuordnung`example.com`mit der Startseite Ihres Dienstes. Aktivieren des`www`, um auch`www.example.com`.
2. Zuordnung`login.example.com`mit der Login-Seite Ihres Dienstes.
3. Zuordnung`*.example.com`mit einer benutzerdefinierten Seite „nicht gefunden“.

Sie können die Zuordnung einer benutzerdefinierten Domäne zu Ihrem App Runner-Dienst aufheben (die Verknüpfung aufheben). Wenn Sie die Verknüpfung einer Domäne aufheben, stoppt App Runner das Weiterleiten von Datenverkehr von dieser Domäne an Ihre Webanwendung. Sie müssen die Einträge für diese Domäne aus Ihrem DNS löschen.

App Runner erstellt intern Zertifikate, die die Gültigkeit der Domäne verfolgen. Sie sind in AWS Certificate Manager (ACM). App Runner löscht diese Zertifikate sieben Tage lang nicht, nachdem eine Domain von Ihrem Dienst getrennt wurde oder nachdem der Dienst gelöscht wurde.

Verwalten benutzerdefinierter Domänen mit der App Runner-Konsole

So ordnen Sie eine benutzerdefinierte Domäne mithilfe der App Runner-Konsole zu (verknüpfen)

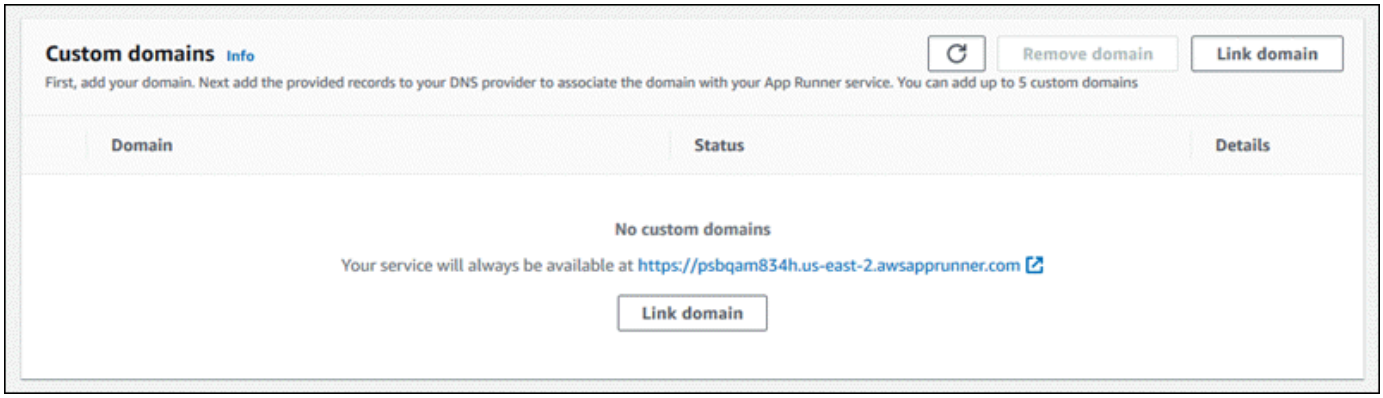
1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.

2. Wählen Sie im NavigationsbereichServices, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Dienst-Dashboard mit einemÜbersicht über den Service.

3. Wählen Sie auf der Service-Dashboard-Seite die OptionBenutzerdefinierte DomänenRegisterkarte.

   In der Konsole werden die benutzerdefinierten Domänen angezeigt, die Ihrem Dienst zugeordnet sind, oderKeine benutzerdefinierten Domänen.



4. Klicken Sie auf derBenutzerdefinierte DomänenWählen Sie auf der RegisterkarteVerknüpfen der Domäne.
5. In derVerknüpfen von benutzerdefinierten DomänennamenGeben Sie einen Domänennamen ein und wählen Sie dannAnzeigen der DNS-Konfiguration.
6. Folgen Sie den Anweisungen auf derKonfigurieren von DNS, um den Domain-Validierungsprozess zu starten.
7. Wenn sich der Domänenstatus inAktiv, überprüfen Sie, ob die Domäne für das Routing von Datenverkehr funktioniert, indem Sie dazu navigieren.

So trennen Sie die Zuordnung einer benutzerdefinierten Domäne mithilfe der App Runner-Konsole

1. Klicken Sie auf derBenutzerdefinierte DomänenWählen Sie die Kachel für die Domäne aus, die Sie aufheben möchten, und wählen Sie dannAufheben der Verknüpfungen.
2. In derAufheben der Verknüpfungen, überprüfen Sie die Aktion, indem SieAufheben der Verknüpfungen.

Verwalten Sie benutzerdefinierte Domänen mit der App Runner-API oderAWS CLI

So verknüpfen Sie eine benutzerdefinierte Domäne mit Ihrem Dienst mithilfe der App Runner-API oderAWS CLIrufen Sie auf, indem SieAssociateCustomDomainAPI-Aktion Wenn der Aufruf erfolgreich ist, gibt es einCustomDomain-Objekt, das die benutzerdefinierte Domäne beschreibt, die Ihrem Dienst zugeordnet ist. Das Objekt sollte den Status`CREATING`und enthält eine Liste vonCertificateValidationRecord-Objekten Dies sind Datensätze, die Sie Ihrem DNS hinzufügen können.

So trennen Sie die Zuordnung einer benutzerdefinierten Domäne zu Ihrem Dienst mithilfe der App Runner-API oderAWS CLIrufen Sie auf, indem SieDisassociateCustomDomainAPI-Aktion Wenn der Aufruf erfolgreich ist, gibt es einCustomDomain-Objekt, das die benutzerdefinierte Domäne beschreibt, die von Ihrem Dienst getrennt wird. Das Objekt sollte den Status`DELETING`.

Anhalten und Fortsetzen eines App Runner-Dienstes

Wenn Sie Ihre Webanwendung vorübergehend deaktivieren und die Ausführung des Codes beenden müssen, können Sie IhreAWS App Runner-Service. App Runner reduziert die Rechenkapazität für den Service auf Null.

Wenn Sie bereit sind, Ihre Anwendung erneut auszuführen, können Sie Ihren App Runner-Dienst fortsetzen. App Runner stellt neue Rechenkapazität bereit, stellt Ihre Anwendung bereit und führt die Anwendung aus. Ihre Anwendungsquelle wird nicht erneut bereitgestellt, und es ist kein Build erforderlich. Stattdessen wird App Runner mit Ihrer aktuell bereitgestellten Version fortgesetzt. Ihre Anwendung behält ihre App Runner-Domain bei.

Important

- Wenn Sie den Dienst anhalten, verliert die Anwendung ihren Status. Beispielsweise geht jeder ephemere Speicher, den Ihr Code verwendet, verloren. Für Ihren Code entspricht das Anhalten und Fortsetzen des Dienstes der Bereitstellung für einen neuen Dienst.
- Wenn Sie einen Dienst aufgrund eines Fehlers in Ihrem Code anhalten (z. B. eines erkannten Fehlers oder eines Sicherheitsproblems), können Sie keine neue Version bereitstellen, bevor Sie den Dienst fortsetzen.

  Daher empfehlen wir, dass Sie den Dienst laufen lassen und stattdessen auf die letzte stabile Anwendungsversion zurücksetzen.

- Wenn Sie Ihren Dienst fortsetzen, stellt App Runner die letzte Anwendungsversion bereit, die verwendet wurde, bevor Sie den Dienst angehalten haben. Wenn Sie seit dem Anhalten des Dienstes neue Quellversionen hinzugefügt haben, werden sie von App Runner nicht automatisch bereitgestellt, selbst wenn die automatische Bereitstellung ausgewählt ist. Angenommen, Sie haben neue Bildversionen im Bild-Repository oder neue Commits im Code-Repository. Diese Versionen werden nicht automatisch bereitgestellt.

  Um eine neuere Version bereitzustellen, führen Sie eine manuelle Bereitstellung durch oder fügen Sie eine weitere Version zu Ihrem Quell-Repository hinzu, nachdem Sie Ihren App Runner-Dienst fortgesetzt haben.

Anhalten und Löschen von verglichen

PauseIhren App Runner-Dienst aufvorübergehendDeaktivieren Sie es. Nur Rechenressourcen werden beendet, und Ihre gespeicherten Daten (z. B. das Container-Image mit Ihrer Anwendungsversion) bleiben intakt. Der Service wird schnell fortgesetzt. Ihre Anwendung kann auf neue Rechenressourcen bereitgestellt werden. Die App Runner Domain bleibt gleich.

LöschenIhren App Runner-Dienst aufPermanentEntfernen Sie es. Ihre gespeicherten Daten werden gelöscht. Wenn Sie den Dienst neu erstellen müssen, muss App Runner Ihre Quelle erneut abrufen und sie auch erstellen, wenn es sich um ein Code-Repository handelt. Ihre Webanwendung erhält eine neue App Runner-Domain.

Wenn Ihr Dienst angehalten ist

Wenn Sie Ihren Dienst pausieren und sich in derPaused, reagiert er anders auf Aktionsanforderungen, einschließlich API-Aufrufe oder Konsolenvorgänge. Wenn ein Dienst angehalten wird, können Sie weiterhin App Runner-Aktionen ausführen, die die Definition oder Konfiguration des Dienstes nicht so ändern, dass es sich auf seine Laufzeit auswirkt. Mit anderen Worten: Wenn eine Aktion das Verhalten, die Skalierung oder andere Merkmale eines ausgeführten Dienstes ändert, können Sie diese Aktion nicht für einen angehaltenen Dienst ausführen.

Die folgenden Listen enthalten Informationen zu API-Aktionen, die Sie für einen angehaltenen Dienst ausführen können und nicht. Die entsprechenden Konsolenoperationen werden ähnlich erlaubt oder verweigert.

Aktionen, die Siecanauf einem angehaltenen Dienst ausführen

- *List**und*Describe**-Aktionen— Aktionen, die nur Informationen lesen.
- *DeleteService*— Sie können einen Dienst jederzeit löschen.
- *TagResource*,*UntagResource*— Tags sind einem Service zugeordnet, sind aber nicht Teil seiner Definition und haben keinen Einfluss auf sein Laufzeitverhalten.

Aktionen, die Siecannotauf einem angehaltenen Dienst ausführen

- *StartDeployment*-Aktionen(oder einManuelle BereitstellungÜber die Konsole)
- *UpdateService*(oder eine Konfigurationsänderung über die Konsole, mit Ausnahme von Tagging-Änderungen)
- *CreateCustomDomainAssociations*, *DeleteCustomDomainAssociations*
- *CreateConnection*, *DeleteConnection*

Anhalten und Fortsetzen des Dienstes über die App Runner-Konsole

So halten Sie Ihren Dienst über die App Runner-Konsole an

1. Öffnen SieApp Runner Konsoleund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im NavigationsbereichServices, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Service-Dashboard mit einemService-Übersicht.
3. Klicken Sie aufAktionenKlicken Sie auf und danach aufPause.

   Auf der Seite „Service-Dashboard“ wird der ServiceStatusÄnderungen anWird durchgeführt, und ändert sich dann zuPaused. Ihr Service ist jetzt angehalten.

So setzen Sie Ihren Dienst über die App Runner-Konsole fort

1. Klicken Sie aufAktionenKlicken Sie auf und danach aufResume.

   Auf der Seite „Service-Dashboard“ wird der ServiceStatusÄnderungen anWird durchgeführt.
2. Warten Sie, bis der Dienst fortgesetzt wird. Auf der Seite „Service-Dashboard“ wird der ServiceStatusÄnderungen zurück zuAusführen von.

3. Um sicherzustellen, dass die Fortsetzung des Dienstes erfolgreich ist, wählen Sie auf der Seite des Service-Dashboards die OptionApp RunnerWert. Es ist die URL für die Website Ihres Dienstes. Überprüfen Sie, ob Ihre Webanwendung korrekt läuft.

Anhalten und Fortsetzen des Dienstes mithilfe der App Runner-API oderAWS CLI

So pausieren Sie Ihren Dienst mithilfe der App Runner-API oderAWS CLIRufen Sie diePauseService-API-Aktion. Wenn der Aufruf eine erfolgreiche Antwort mit einem-ServiceObjekt anzeigen`"Status": "OPERATION_IN_PROGRESS"`, beginnt App Runner, Ihren Dienst zu pausieren.

So setzen Sie Ihren Dienst mithilfe der App Runner-API oderAWS CLIRufen Sie dieLebenslauf Service-API-Aktion. Wenn der Aufruf eine erfolgreiche Antwort mit einem-ServiceObjekt anzeigen`"Status": "OPERATION_IN_PROGRESS"`, beginnt App Runner mit der Fortsetzung Ihres Dienstes.

Löschen eines App Runner-Dienstes

Wenn Sie die Webanwendung beenden möchten, die in IhremAWS App Runner-Dienst löschen, können Sie den Service löschen. Das Löschen eines Dienstes stoppt den ausgeführten Webdienst, entfernt die zugrunde liegenden Ressourcen und löscht die zugeordneten Daten.

Wenn Sie einen App Runner-Dienst aus den folgenden Gründe löschen möchten:

- Die Webanwendung wird nicht mehr benötigt— Zum Beispiel ist es eingestellt, oder es ist eine Entwicklungsversion, die Sie bereits verwenden.
- Sie haben das App-Runner-Servicekonto erreicht— Sie möchten einen neuen Service in der gleichenAWS-Regionund Sie haben das mit Ihrem Konto verknüpfte Kontingent erreicht. Weitere Informationen finden Sie unter the section called “App Runner Ressourcenkontingente”.
- Sicherheits- oder Datenschutzüberlegungen— Sie möchten, dass App Runner die Daten löscht, die er für Ihren Dienst speichert.

Pausieren vs.

PauseIhren App Runner-Dienst aufvorübergehendDeaktivieren Sie es. Nur Rechenressourcen werden beendet, und Ihre gespeicherten Daten (z. B. das Container-Image mit Ihrer

Anwendungsversion) bleiben intakt. Der Service wird schnell fortgesetzt. Ihre Anwendung kann auf neue Rechenressourcen bereitgestellt werden. Die App Runner Domain bleibt gleich.

LöschenIhren App Runner-Dienst aufPermanentRemove es. Ihre gespeicherten Daten werden gelöscht. Wenn Sie den Dienst neu erstellen müssen, muss App Runner Ihre Quelle erneut abrufen und sie auch erstellen, wenn es sich um ein Code-Repository handelt. Ihre Webanwendung erhält eine neue App Runner-Domain.

Was löscht App Runner?

Wenn Sie Ihren Dienst löschen, löscht App Runner einige zugeordnete Elemente und löscht keine anderen. Die folgenden Listen bieten die Details.

Elemente, die App Runner löscht:

- Container-Images— Eine Kopie des bereitgestellten Images oder des Images, das App Runner aus Ihrem Quellcode erstellt hat. Es wird in der Amazon Elastic Container Registry (Amazon ECR) mit internerAWS-Konten, die im Besitz von App Runner sind.
- Service-Konfiguration— Die Konfigurationseinstellungen, die Ihrem App Runner-Dienst zugeordnet sind. Sie werden in Amazon DynamoDB mit internerAWS-Konten, die im Besitz von App Runner sind.

Elemente, die App Runner nicht löscht:

- Connection (Verbindung)— Möglicherweise haben Sie eine Verbindung, die Ihrem Dienst zugeordnet ist. Eine App Runner-Verbindung ist eine separate Ressource, die von mehreren App Runner-Diensten gemeinsam genutzt werden kann. Wenn Sie die Verbindung nicht mehr benötigen, können Sie sie explizit löschen. Weitere Informationen finden Sie unter the section called “Verbindungen”.
- Benutzerdefinierte Domänenzertifikate— Wenn Sie benutzerdefinierte Domänen mit einem App Runner-Dienst verknüpfen, erstellt App Runner intern Zertifikate, die die Domänengültigkeit verfolgen. Sie sind inAWS Certificate Manager(ACM). App Runner löscht das Zertifikat sieben Tage lang nicht, nachdem eine Domain von Ihrem Dienst aufgehoben wurde oder nachdem der Dienst gelöscht wurde. Weitere Informationen finden Sie unter the section called “Benutzerdefinierte Domänennamen”.

Löschen Sie Ihren Dienst mit der App Runner-Konsole

So löschen Sie Ihren Dienst mit der App Runner-Konsole

1. Öffnen SieApp Runner Konsoleund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im NavigationsbereichServices, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Service-Dashboard mit einemServicekonto.
3. Wählen Sie Actions (Instance-Aktionen) und anschließend Delete (Löschen) aus.

   Die -Konsole führt Sie zurServicesangezeigten. Der gelöschte Dienst zeigt dieWird ausgeführt, und dann verschwindet der Dienst aus der Liste. Ihr Dienst wird jetzt gelöscht.

Löschen Sie Ihren Dienst mit der App Runner-API oderAWS CLI

So löschen Sie Ihren Dienst mithilfe der App Runner-API oderAWS CLIRufen Sie dieDeleteServiceAPI-Aktion. Wenn der Aufruf eine erfolgreiche Antwort mit einem-ServiceObjekt anzeigen`"Status": "OPERATION_IN_PROGRESS"`, beginnt App Runner mit dem Löschen Ihres Dienstes.

Protokollierung und Überwachung für Ihren App Runner

AWS App Runnerintegriert sich mit mehrerenAWS, um Ihnen eine umfangreiche Suite an Protokollierungs- und Überwachungstools für Ihren App Runner-Service zur Verfügung zu stellen. Themen in diesem Kapitel beschreiben diese Funktionen.

Themen

Tracking App Runner-Dienstaktivität

AWS App Runnerverwendet eine Liste von Vorgängen, um die Aktivitäten in Ihrem App Runner-Dienst zu verfolgen. Ein Vorgang stellt einen asynchronen Aufruf einer API-Aktion dar, z. B. das Erstellen eines Dienstes, das Aktualisieren einer Konfiguration und das Bereitstellen eines Dienstes. Im folgenden Abschnitt wird beschrieben, wie Sie Aktivitäten in der App Runner-Konsole und mit der -API verfolgen.

Tracking App Runner-Dienstaktivität in der Konsole

Die App Runner-Konsole zeigt Ihre App Runner-Dienstaktivität an und bietet weitere Möglichkeiten, Vorgänge zu untersuchen.

Anzeigen der Aktivität Ihres Dienstes

1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im NavigationsbereichServices, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Service-Dashboard mit einemÜbersicht über den Service.
3. Wählen Sie auf der Service-Dashboard-Seite die OptionAktivität-Tab, sofern noch nicht geschehen.

   In der Konsole wird eine Liste mit Vorgängen angezeigt.

4. Um bestimmte Vorgänge zu suchen, erweitern Sie die Liste, indem Sie einen Suchbegriff eingeben. Sie können nach jedem Wert suchen, der in der Tabelle angezeigt wird.
5. Wählen Sie einen der aufgelisteten Vorgänge aus, um das zugehörige Protokoll anzuzeigen oder herunterzuladen.

Abrufen von App Runner-Dienstvorgängen mit der App Runner-API oderAWS CLI

DieListOperationsGibt unter Berücksichtigung des Amazon-Ressourcennamen (ARN) eines App Runner-Dienstes eine Liste mit Vorgängen zurück, die für diesen Service aufgetreten sind. Jedes Listenelement enthält eine Vorgangsnummer und einige Sendungsverfolgungsdetails.

Anzeigen von App Runner-Protokollen, die zu CloudWatch Logs gestreamt wurden

Sie können Amazon CloudWatch Logs verwenden, um Protokolldateien zu überwachen, zu speichern und darauf zuzugreifen, die Ihre Ressourcen in verschiedenenAWS-Dienste generieren. Weitere Informationen finden Sie unterAmazon CloudWatch Logs Benutzerhandbuch.

AWS App Runnererfasst die Ausgabe Ihrer Anwendungsbereitstellungen und Ihres aktiven Dienstes und streamt sie zu CloudWatch Logs. In den folgenden Abschnitten werden App Runner-Protokoll-Streams aufgeführt und gezeigt, wie Sie sie in der App Runner-Konsole anzeigen.

App Runner-Protokollgruppen und -streams

CloudWatch Logs speichert Protokolldaten in Protokoll-Streams, die in Protokollgruppen weiter organisiert werden. AProtokolldatenstromEine Abfolge von Protokollereignissen aus einer bestimmten Quelle. Eine Protokollgruppe ist eine Gruppe von Protokollstreams, die dieselben Einstellungen für die Aufbewahrung, Überwachung und Zugriffskontrolle besitzen.

App Runner definiert zwei CloudWatch Logs s-Protokollgruppen mit jeweils mehreren Log-Streams für jeden Ihrer App Runner-Dienste in IhremAWS-Konto.

Serviceprotokolle

Die Dienstprotokollgruppe enthält Protokollierungsausgabe, die von App Runner generiert wird, während sie Ihren App Runner-Dienst verwaltet und darauf reagiert.

Name der Protokollgruppe	Beispiel
`/aws/apprunner/ service-name /service-id /service`	`/aws/apprunner/python-test/ ac7ec8b51ff34746bcb6654e0bc b23da/service`

Innerhalb der Dienstprotokollgruppe erstellt App Runner einen Ereignisprotokoll-Stream, um Aktivitäten im Lebenszyklus Ihres App Runner-Dienstes zu erfassen. Dies könnte beispielsweise Ihre Anwendung starten oder anhalten.

Darüber hinaus erstellt App Runner einen Protokollstream für jeden lang laufenden asynchronen Vorgang, der mit Ihrem Dienst verknüpft ist. Der Name des Protokollstroms spiegelt den Vorgangstyp und die spezifische Vorgangs-ID wider.

A-Bereitstellungist eine Art von Operation. Bereitstellungsprotokolle enthalten die Protokollierungsausgabe der Build- und Bereitstellungsschritte, die App Runner beim Erstellen eines Dienstes oder beim Bereitstellen einer neuen Version der Anwendung ausführt. Namen des Bereitstellungsprotokolls beginnen mit`deployment/`und endet mit der ID des Vorgangs, der die Bereitstellung ausführt. Dieser Vorgang ist entweder einCreateService-Aufruf für die anfängliche Anwendungsbereitstellung oder eineStartDeployment-Aufruf für jede weitere Bereitstellung.

Innerhalb eines Bereitstellungsprotokolls beginnt jede Protokollmeldung mit einem Präfix:

- `[AppRunner]`— Ausgabe, die App Runner während der Bereitstellung generiert.
- `[Build]`— Ausgabe Ihrer eigenen Build-Skripte.

Protokoll-Streamname	Beispiel
`events`	N/A (fester Name)
`operation-type /operation-id`	`deployment/c2c8eeedea164f45 9cf78f12a8953390`

Anwendungsprotokolle

Die Anwendungsprotokollgruppe enthält die Ausgabe des ausgeführten Anwendungscodes.

Name der Protokollgruppe	Beispiel
`/aws/apprunner/ service-name /service-id /application`	`/aws/apprunner/python-test/ ac7ec8b51ff34746bcb6654e0bcb23da/ application`

Innerhalb der Anwendungsprotokollgruppe erstellt App Runner einen Protokollstream für jede Instanz (Skalierungseinheit), auf der Ihre Anwendung ausgeführt wird.

Protokoll-Streamname	Beispiel
`instance/ instance-id`	`instance/1a80bc9134a84699b7 b3432ebeebb591`

Anzeigen von App Runner-Protokollen in der -Konsole

Die App Runner-Konsole zeigt eine Zusammenfassung aller Protokolle für Ihren Dienst an und ermöglicht es Ihnen, sie anzuzeigen, zu durchsuchen und herunterzuladen.

So zeigen Sie Protokolle für Ihren Service an:

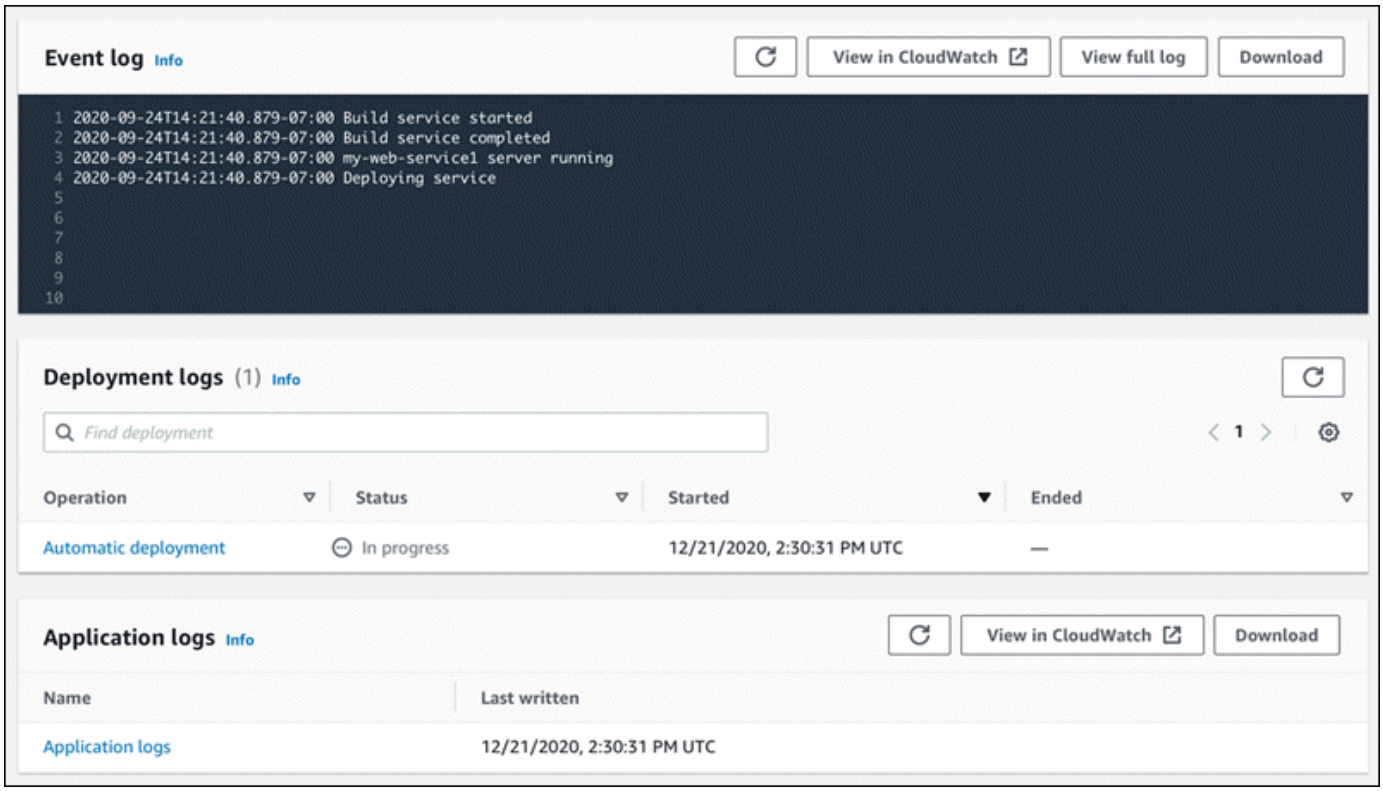
1. Öffnen SieApp Runner-Konsoleund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im Navigationsbereich aus.Services, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Service-Dashboard mit einemServiceübersicht.
3. Wählen Sie auf der Service-Dashboard-Seite die OptionProtokolleRegisterkarte.

   Die Konsole zeigt einige Arten von Protokollen in mehreren Abschnitten an:

   - Ereignisprotokoll— Aktivität im Lebenszyklus Ihres App Runner-Dienstes. Die Konsole zeigt die neuesten Ereignisse an.
   - Bereitstellungsprotokolle— Quell-Repository-Bereitstellungen für Ihren App Runner-Dienst. Die Konsole zeigt für jede Bereitstellung einen separaten Protokolldatenstrom an.

- Anwendungsprotokolle— Die Ausgabe der Webanwendung, die für Ihren App Runner-Dienst bereitgestellt wird. Die Konsole kombiniert die Ausgabe aller ausgeführten Instanzen in einem einzigen Log-Stream.



4. Um bestimmte Bereitstellungen zu suchen, erweitern Sie die Bereitstellungsprotokollliste, indem Sie einen Suchbegriff eingeben. Sie können nach jedem Wert suchen, der in der Tabelle angezeigt wird.
5. Um den Inhalt eines Protokolls anzuzeigen, wählen SieAnzeige des vollständigen Protokolls(Ereignisprotokoll) oder den Namen des Protokolldatenstroms (Bereitstellungs- und Anwendungsprotokolle).
6. Klicken Sie aufHerunterladen, um ein Protokoll herunterzuladen. Wählen Sie für einen Bereitstellungsprotokolldatenstrom zuerst einen Protokolldatenstrom aus.
7. Klicken Sie aufAnzeigen in CloudWatch, um die CloudWatch Konsole zu öffnen und die vollständigen Funktionen zu nutzen, um Ihre App Runner-Service-Protokolle zu erkunden. Wählen Sie für einen Bereitstellungsprotokolldatenstrom zuerst einen Protokolldatenstrom aus.

> **Note**
>
> Die CloudWatch Konsole ist besonders nützlich, wenn Sie Anwendungsprotokolle bestimmter Instanzen anstelle des kombinierten Anwendungsprotokolls anzeigen möchten.

Anzeigen von App Runner-Service-Metriken, die CloudWatch gemeldet wurden

Amazon CloudWatch überwacht Ihre Amazon Web Services (AWS) -Ressourcen und die Anwendungen, die Sie auf ausführenAWSSie können in Echtzeit. Sie können CloudWatch verwenden, um Metriken zu erfassen und nachzuverfolgen, die Variablen sind, die Sie für Ihre Ressourcen und Anwendungen messen können. Sie können es auch verwenden, um Alarme zu erstellen, die Metriken überwachen. Wenn ein bestimmter Schwellenwert erreicht ist, sendet CloudWatch Benachrichtigungen oder nimmt automatisch Änderungen an den überwachten Ressourcen vor. Weitere Informationen finden Sie unterAmazon CloudWatch-Benutzerhandbuch.

AWS App Runnersammelt eine Vielzahl von Metriken, die Ihnen einen besseren Einblick in die Nutzung, Leistung und Verfügbarkeit Ihrer App Runner-Dienste bieten. Einige Metriken verfolgen einzelne Instanzen, die Ihren Webdienst ausführen, während andere sich auf der gesamten Service-Ebene befinden. In den folgenden Abschnitten werden App Runner-Metriken aufgeführt und gezeigt, wie Sie sie in der App Runner-Konsole anzeigen können.

Metriken

App Runner sammelt die folgenden Metriken in Bezug auf Ihren Dienst und veröffentlicht sie in CloudWatch im`AWS/AppRunner`-Namespace.

Metriken auf Instance-Ebenewerden für jede Instanz (Skalierungseinheit) einzeln gesammelt.

Was ist gemessen?	Metrik	Beschreibung
CPU utilization	`CPUUtilization`	Die durchschnittliche CPU-Auslastung während einer Minute.

Was ist gemessen?	Metrik	Beschreibung
Memory utilization	`MemoryUtilization`	Die durchschnittliche Speicherauslastung während einer Minute.

Metriken auf Servicelevelwerden für den gesamten Service erfasst.

Was ist gemessen?	Metriken	Beschreibung
HTTP request count	`Requests`	Die Anzahl von HTTP-Anforderungen, die der Dienst empfangen hat.

HTTP status counts	`2xxStatusResponses`
	`4xxStatusResponses`
	`5xxStatusResponses`
		Die Anzahl von HTTP-Anforderungen, die jeden Antwortstatus zurückgegeben haben, gruppiert nach Kategorie (2XX, 4XX, 5XX).

HTTP request latency	`RequestLatency`	Die Zeit, die der Webdienst zum Verarbeiten von HTTP-Anforderungen benötigt hat.
Instance counts	`ActiveInstances`	Die Anzahl von Instances, die HTTP-Anforderungen für Ihren Service verarbeiten.

Anzeigen von App Runner-Metriken in der -Konsole

Die App Runner-Konsole zeigt grafisch die Metriken an, die App Runner für Ihren Dienst erfasst, und bietet weitere Möglichkeiten, sie zu erkunden.

> **Note**
>
> Zu diesem Zeitpunkt zeigt die Konsole nur Service-Metriken an. Verwenden Sie die CloudWatch Konsole, um Instanzmetriken anzuzeigen.

So zeigen Sie Protokolle für Ihren Service an:

1. Öffnen SieApp Runnerund in derRegionenaus, wählen Sie IhreAWS-Region.
2. Wählen Sie im NavigationsbereichServices, und wählen Sie dann Ihren App Runner-Dienst aus.

   Die Konsole zeigt das Service-Dashboard mit einemServicemetrik.
3. Wählen Sie auf der Service-Dashboard-Seite die Option-Metriken-Registerkarte.

   Die Konsole zeigt eine Reihe von Metrikdiagrammen an.
4. Wählen Sie eine Dauer (z. B.12h), um Metrikendiagramme auf den letzten Zeitraum dieser Dauer auszudehnen.
5. Klicken Sie aufAdd to dashboardoben in einem der Diagrammabschnitte oder verwenden Sie das Menü in einem beliebigen Diagramm, um die relevanten Metriken einem Dashboard in der CloudWatch Konsole zur weiteren Untersuchung hinzuzufügen.

Umgang mit App Runner-Ereignissen in EventBridge

Mit Amazon EventBridge können Sie ereignisgesteuerte Regeln einrichten, die einen Stream von Echtzeitdaten von IhremAWS App Runner-Dienst für bestimmte Muster. Wenn ein Muster für eine Regel übereinstimmt, initiiert EventBridge eine Aktion in einem Ziel wieAWS LambdaAmazon ECSAWS Batchund Amazon SNS. Sie können beispielsweise eine Regel für das Versenden von E-Mail-Benachrichtigungen festlegen, indem Sie ein Amazon SNS -Thema signalisieren, wenn eine Bereitstellung für Ihren Service fehlschlägt. Oder Sie können eine Lambda -Funktion einrichten, um einen Slack-Kanal zu benachrichtigen, wenn ein Service-Update fehlschlägt. Weitere Informationen zu EventBridge finden Sie unterBenutzerhandbuch für Amazon EventBridge.

App Runner sendet die folgenden Ereignistypen an EventBridge

- Änderung des Servicestatus— Eine Änderung des Status eines App Runner-Dienstes. Beispielsweise wurde ein Service-Status in`DELETE_FAILED`.

- Statusänderung des Servicestatus— Eine Änderung des Status eines langen, asynchronen Vorgangs an einem App Runner-Dienst. Beispielsweise wurde ein Dienst erstellt, ein Dienstupdate erfolgreich abgeschlossen oder eine Dienstbereitstellung mit Fehlern abgeschlossen.

Erstellen einer EventBridge Regel, um auf App Runner-Ereignisse zu reagieren

Eine EventBridge-Ereignisist ein Objekt, das einige standardmäßige EventBridge Felder definiert, z. B.AWS-Dienst und den Detailtyp (Ereignis) sowie einen ereignisspezifischen Satz von Feldern mit den Ereignisdetails. Um eine EventBridge Regel zu erstellen, verwenden Sie die EventBridge-Konsole, um eineEreignismuster(welche Ereignisse verfolgt werden sollen) und geben Sie einZielaktion(was sollte auf einem Spiel getan werden). Ein Ereignismuster ähnelt den Ereignissen, die es übereinstimmt. Sie geben eine Teilmenge der Felder an, die übereinstimmen sollen, und für jedes Feld geben Sie eine Liste möglicher Werte an. Dieses Thema enthält Beispiele für App Runner-Ereignisse und Ereignismuster.

Weitere Informationen zum Erstellen von EventBridge -Regeln finden Sie unterErstellen einer -Regel für.AWS-ServiceimBenutzerhandbuch für Amazon EventBridge.

> **Note**
>
> Einige Dienste unterstützenvordefinierte Musterin EventBridge. Dies vereinfacht die Erstellung eines Ereignismusters. Sie wählen Feldwerte in einem Formular aus, und EventBridge generiert das Muster für Sie. Zur Zeit unterstützt App Runner keine vordefinierten Muster. Sie müssen das Muster als JSON-Objekt eingeben. Sie können die Beispiele in diesem Thema als Ausgangspunkt verwenden.

Anwendungs-Runner-Ereignisbeispiele

Dies sind einige Beispiele für Ereignisse, die App Runner an EventBridge sendet.

- Änderungsereignis für den Servicestatus. Insbesondere ein Dienst, der sich von der`OPERATION_IN_PROGRESS`auf die`RUNNING`Status.

```
{
  "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"
  }
}
```

- Änderungsereignis für den Änderungsereignis Insbesondere wird ein`UpdateService`-Operation wurde erfolgreich abgeschlossen.

```
{
  "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"
  }
}
```

Beispiele für Anwendungs-Runner-Ereignismuster

Die folgenden Beispiele veranschaulichen Ereignismuster, die Sie in EventBridge Regeln verwenden können, um einem oder mehreren App Runner-Ereignissen zuzuordnen. Ein Ereignismuster ähnelt einem Ereignis. Fügen Sie nur die Felder ein, die Sie abgleichen möchten, und geben Sie eine Liste anstelle eines Skalars an.

- Alle Dienststatusänderungsereignisse für Dienste eines bestimmten Kontos abgleichen, bei denen sich der Dienst nicht mehr in`RUNNING`Status.

```
{
  "detail-type": [ "Service Status Change" ],
  "source": [ "aws.apprunner" ],
  "account": [ "111122223333" ],
  "detail": {
    "PreviousStatus": [ "RUNNING" ]
  }
}
```

- Ordnen Sie alle Vorgangsstatusänderungsereignisse für Dienste eines bestimmten Kontos zu, bei denen der Vorgang fehlgeschlagen ist.

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

App Runner-Ereignisreferenz

Änderung des Servicestatus

Ein Änderungsereignis für den Servicestatus`detail-type`auf`Service Status Change`. Sie hat die folgenden Detailfelder und -werte:

```
"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"
```

Änderungsereignis

Ein Änderungsereignis für den Statusvorgang`detail-type`auf`Service Operation Status Change`. Sie hat die folgenden Detailfelder und -werte:

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

In der folgenden Tabelle finden Sie alle möglichen Statuscodes und zugehörigen Meldungen.

Status	Fehlermeldung
`CreateServiceStarted`	Service-Erstellung wurde gestartet.
`CreateServiceCompletedSuccessfully`	Service-Erstellung wurde erfolgreich abgeschlossen.
`CreateServiceFailed`	Service-Erstellung ist fehlgeschlagen. Details dazu finden Sie unter Dienstprotokolle.
`DeleteServiceStarted`	Servicelöschung wurde gestartet.

Status	Fehlermeldung
`DeleteServiceCompletedSuccessfully`	Servicelöschung wurde erfolgreich abgeschlossen.
`DeleteServiceFailed`	Fehler beim Löschen des Diensts.
`UpdateServiceStarted`	
`UpdateServiceCompletedSuccessfully`	Service-Aktualisierung wurde erfolgreich abgeschlossen. Neue Anwendung und Konfiguration werden bereitgestellt.
	Service-Aktualisierung wurde erfolgreich abgeschlossen. Neue Konfiguration wird bereitgestellt.
`UpdateServiceFailed`	Service-Aktualisierung fehlgeschlagen. Details dazu finden Sie unter Dienstprotokolle.
`DeploymentStarted`	Die Bereitstellung wurde gestartet.
`DeploymentCompletedSuccessfully`	Die Bereitstellung wurde erfolgreich abgeschlossen.
`DeploymentFailed`	Die Bereitstellung ist fehlgeschlagen. Details dazu finden Sie unter Dienstprotokolle.
`PauseServiceStarted`	Service-Pause gestartet.
`PauseServiceCompletedSuccessfully`	Servicepause wurde erfolgreich abgeschlossen.
`PauseServiceFailed`	Service-Pause fehlgeschlagen.
`ResumeServiceStarted`	Servicestatus gestartet.
`ResumeServiceCompletedSuccessfully`	Servicestatus wurde erfolgreich abgeschlossen.
`ResumeServiceFailed`	Dienst-Wiederaufnahme fehlgeschlagen.

Protokollieren von API-Aufrufen von App Runner mitAWS CloudTrail

App Runner ist mitAWS CloudTrailbietet einen Service, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem durchgeführten Aktionen bereitstellt.AWS-Dienst in App Runner. CloudTrail erfasst alle API-Aufrufe für App Runner als Ereignisse. Die erfassten Aufrufe umfassen Aufrufe von App Runner und Code-Aufrufe der App Runner API-Operationen. Wenn Sie einen Trail erstellen, aktivieren Sie die kontinuierliche Bereitstellung von CloudTrail -Ereignissen an einen Amazon S3 Bucket, einschließlich Ereignissen für App Runner. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole trotzdem in Ereignisverlauf anzeigen. Mit den von CloudTrail erfassten Informationen können Sie die an App Runner gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrail finden Sie unterAWS CloudTrailBenutzerhandbuch.

Informationen zu App Runner in CloudTrail

CloudTrail ist auf IhremAWS-KontoWenn Sie das Konto anlegen. Wenn in App Runner eine Aktivität auftritt, wird diese in einem CloudTrail -Ereignis zusammen mit anderenAWSServiceereignisse inEreignisverlauf. Sie können die neusten Ereignisse in IhrerAWS-Konto. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail-Ereignisverlauf.

Für eine fortlaufende Aufzeichnung von Ereignissen in IhrerAWS-KontoErstellen Sie einen Trail, einschließlich Ereignissen für App Runner. Ein Trail ermöglicht es CloudTrail, Protokolldateien in einem Amazon S3-Bucket bereitzustellen. Wenn Sie einen Pfad in der Konsole anlegen, gilt dieser standardmäßig für alleAWS-Regionen. Der Trail protokolliert Ereignisse aus allen -Regionen inAWSDie Protokolldateien werden in den Amazon S3 Bucket ausgegeben, den Sie angeben. Darüber hinaus können Sie andereAWS-Protokollen der CloudTrail -Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- Übersicht zum Erstellen eines Pfads
- Siehe Von CloudTrail unterstützte Dienste und Integrationen.
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail-Protokolldateien aus mehreren Konten

Alle App Runner Aktionen werden von CloudTrail protokolliert und in derAWS App Runner-API-Referenz. Zum Beispiel generieren Aufrufe der Aktionen `CreateService`, `DeleteConnection` und `StartDeployment` Einträge in den CloudTrail-Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Anhand der Identitätsinformationen zur Benutzeridentität können Sie Folgendes bestimmen:

- Ob die Anfrage mit Stammbenutzer- oder AWS Identity and Access Management (IAM)-Anmeldeinformationen ausgeführt wurde.
- Ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen föderierten Benutzer ausgeführt wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter dem CloudTrail userIdentity-Element.

Grundlagen zu -Protokolldateieinträgen von App Runner

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail-Protokolleinträge sind kein geordnetes Stack-Trace der öffentlichen API-Aufrufe und erscheinen daher in keiner bestimmten Reihenfolge.

Das folgende Beispiel zeigt den CloudTrail-Protokolleintrag, der die Aktion `CreateService` demonstriert.

> **Note**
>
> Aus Sicherheitsgründen werden einige Eigenschaftswerte in den Protokollen geschwärzt und durch den Text`HIDDEN_DUE_TO_SECURITY_REASONS`. Dies verhindert eine unbeabsichtigte Offenlegung von geheimen Informationen. Sie können jedoch immer noch sehen, dass diese Eigenschaften in der Anforderung übergeben oder in der Antwort zurückgegeben wurden.

Beispiel für den CloudTrail -Protokolleintrag für**`CreateService`**Aktion „Runner“

```
{
  "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"
}
```

Festlegen von App Runner-Dienstoptionen mithilfe einer Konfigurationsdatei

> **Note**
>
> Konfigurationsdateien gelten nur fürDienste, die auf Quellcode basieren. Sie können keine Konfigurationsdateien mitImage-basierte Dienste.

Wenn Sie eineAWS App Runner-Dienst mithilfe eines Quellcode-RepositoryAWS App Runnererfordert Informationen zum Erstellen und Starten Ihres Dienstes. Sie können diese Informationen jedes Mal bereitstellen, wenn Sie einen Dienst mithilfe der App Runner-Konsole oder API erstellen. Alternativ können Sie Dienstoptionen festlegen, indem Sie eineKonfigurationsdatei. Die Optionen, die Sie in einer Datei angeben, werden Teil des Quell-Repositorys, und alle Änderungen an diesen Optionen werden in ähnlicher Weise verfolgt, wie Änderungen am Quellcode verfolgt werden. Sie können die App Runner-Konfigurationsdatei verwenden, um mehr Optionen anzugeben, als die API unterstützt. Sie müssen keine Konfigurationsdatei bereitstellen, wenn Sie nur die grundlegenden Optionen benötigen, die die API unterstützt.

Die App Runner-Konfigurationsdatei ist eine YAML-Datei mit dem Namen`apprunner.yaml`Sie können im Root-Verzeichnis Ihres Repositorys Ihrer Anwendung verwenden. Es bietet Build- und Laufzeitoptionen für Ihren Service. Die Werte in dieser Datei weisen App Runner an, wie Sie Ihren Dienst erstellen und starten und Laufzeitkontext wie Netzwerkeinstellungen und Umgebungsvariablen bereitstellen.

Die App Runner-Konfigurationsdatei enthält keine Betriebseinstellungen wie CPU und Arbeitsspeicher.

Beispiele für App Runner-Konfigurationsdateien finden Sie unterthe section called “Beispiele”. Eine vollständige Referenzanleitung finden Sie unterthe section called “Referenz”.

Themen

- Beispiele für die App Runner-Konfigurationsdatei
- Referenz für die App Runner-Konfigurationsdatei

Beispiele für die App Runner-Konfigurationsdatei

> **Note**
>
> Konfigurationsdateien gelten nur fürDienste, die auf Quellcode basieren. Sie können keine Konfigurationsdateien mitImage-basierte Dienste.

In den folgenden Beispielen finden SieAWS App Runner-Konfigurationsdateien verwenden. Einige sind minimal und enthalten nur erforderliche Einstellungen. Andere sind vollständig, einschließlich aller Abschnitte der Konfigurationsdatei. Für eine Übersicht über die App Runner-Konfigurationsdateien finden Sie unterApp Runner-Konfigurationsdatei.

Beispielkonfigurationsdatei

Minimaler Konfigurationsdatei

Mit einer minimalen Konfigurationsdatei macht App Runner die folgenden Annahmen:

- Während des Builds oder der Ausführung sind keine benutzerdefinierten Umgebungsvariablen erforderlich.
- Es wird die neueste Laufzeitversion verwendet.
- Die Standard-Portnummer und die Portumgebungsvariable werden verwendet.

Example apprunner.yaml

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

Vollständige Konfigurationsdatei

Dieses Beispiel zeigt die Verwendung aller Konfigurationsschlüssel mit einer verwalteten Laufzeit.

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"
```

Beispiele für bestimmte verwaltete Laufzeitkonfigurationsdateien finden Sie im spezifischen Laufzeit-Unterthema unterCode-basierter Service.

Referenz für die App Runner-Konfigurationsdatei

> **Note**
>
> Konfigurationsdateien gelten nur fürDienste, die auf Quellcode basieren. Sie können keine Konfigurationsdateien mitImage-basierte Dienste.

Dieses Thema ist ein umfassender Referenzhandbuch zur Syntax und Semantik einesAWS App Runner-Konfigurationsdatei erstellen. Eine Übersicht über die App Runner-Konfigurationsdateien finden Sie unterApp Runner-Konfigurationsdatei.

Die App Runner-Konfigurationsdatei ist eine YAML-Datei. Benennen Sie es`apprunner.yaml`Legen Sie sie im Stammverzeichnis des -Repositorys Ihrer Anwendung ab.

Übersicht über die Struktur

Die App Runner-Konfigurationsdatei ist eine YAML-Datei. Benennen Sie es`apprunner.yaml`Legen Sie sie im Stammverzeichnis des -Repositorys Ihrer Anwendung ab.

Die App Runner-Konfigurationsdatei enthält folgende Hauptteile:

- Oberer Abschnitt— Enthält Schlüssel der obersten Ebene
- Build-Bereich— Konfiguriert die Build-Phase
- Abschnitt „Ausführen“— Konfiguriert die Laufzeitphase

Oberer Abschnitt

Die Schlüssel oben in der Datei enthalten allgemeine Informationen über die Datei und Ihre Dienstlaufzeit. Die folgenden Schlüssel sind verfügbar:

- `version`–Erforderlich. Die Version der App Runner-Konfigurationsdatei. Verwenden Sie im Idealfall die neueste Version.

  Syntax

  ```
  version: version
  ```

  Example

  ```
  version: 1.0
  ```

- `runtime`–Erforderlich. Der Name der Laufzeitumgebung, die Ihre Anwendung verwendet. Folgende Laufzeiten stehen derzeit zur Verfügung:

  `python3`, `nodejs12`

> **Note**
>
> Die Namenskonvention einer verwalteten Laufzeit ist*<language-name> <major-version>*.

Syntax

```
runtime: runtime-name
```

Example

```
runtime: python3
```

Build-Bereich

Im Build-Abschnitt wird die Buildphase der App Runner-Dienstbereitstellung konfiguriert. Sie können Build-Befehle und Umgebungsvariablen angeben. Build-Befehle sind erforderlich.

Der Abschnitt beginnt mit dem`build:`und hat die folgenden Unterschlüssel:

- `commands`–Erforderlich. Gibt die Befehle an, die App Runner während verschiedener Buildphasen ausführt. Enthält die folgenden Unterschlüssel:
  - `pre-build`–Optional. Die Befehle, die App Runner vor dem Build ausführt. Installieren Sie zum BeispielnpmAbhängigkeiten oder Testbibliotheken.
  - `build`–Erforderlich. Die Befehle, die App Runner zum Erstellen Ihrer Anwendung ausführt. Verwenden Sie z. B.pipenv.
  - `post-build`–Optional. Die Befehle, die App Runner nach dem Build ausführt. Beispielsweise können Sie Maven nutzen, um Build-Artefakte in eine Jar- oder WAR-Datei zu verpacken, oder führen Sie einen Test aus.

Syntax

```
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–Optional. Gibt benutzerdefinierte Umgebungsvariablen für die Build-Phase an. Definiert als skalare Name-Wert-Zuordnungen. Sie können auf diese Variablen nach Namen in Ihren Build-Befehlen verweisen.

  Syntax

```
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"
```

Abschnitt „Ausführen“

Im Abschnitt „Ausführen“ wird die Containerlaufphase der App Runner-Anwendungsbereitstellung konfiguriert. Sie können Laufzeitversion, Startbefehl, Netzwerkport und Umgebungsvariablen angeben.

Der Abschnitt beginnt mit dem`run:`und hat die folgenden Unterschlüssel:

- `runtime-version`–Optional. Gibt eine Laufzeitversion an, die Sie für Ihren App Runner-Dienst sperren möchten.

  Standardmäßig ist nur die Hauptversion gesperrt. App Runner verwendet die neuesten Neben- und Patchversionen, die für die Laufzeit bei jeder Bereitstellung oder Dienstaktualisierung verfügbar sind. Wenn Sie Haupt- und Nebenversionen angeben, werden beide gesperrt, und App Runner aktualisiert nur Patchversionen. Wenn Sie Haupt-, Neben- und Patchversionen angeben, ist Ihr Dienst für eine bestimmte Laufzeitversion gesperrt, und App Runner aktualisiert ihn niemals.

  Syntax

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

  Example

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

- `command`–Erforderlich. Der Befehl, den App Runner zum Ausführen der Anwendung verwendet, nachdem der Anwendungsaufbau abgeschlossen wurde.

  Syntax

  ```
  run:
    command: command
  ```

- `network`–Optional. Gibt den Port an, den Ihre Anwendung überwacht. Es umfasst Folgendes:

- `port`–Optional. Wenn angegeben, ist dies die Portnummer, die von der Anwendung überwacht wird. Der Standardwert ist `8080`.
- `env`–Optional. Wenn angegeben, übergibt App Runner die Portnummer an den Container in dieser Umgebungsvariablen, zusätzlich zur (nicht statt) Übergabe derselben Portnummer in der standardmäßigen Umgebungsvariablen`PORT`. Mit anderen Worten: Wenn Sie`env`, übergibt App Runner die Portnummer in zwei Umgebungsvariablen.

Syntax

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

Example

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

- `env`–Optional. Definition von benutzerdefinierten Umgebungsvariablen für die Laufphase. Definiert als skalare Name-Wert-Zuordnungen. Sie können in Ihrer Laufzeitumgebung nach Namen auf diese Variablen verweisen.

Syntax

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

Example

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

```
      value: "example"
```

Die App Runner-API

DieAWS App RunnerApplication Programming Interface (API) ist eine RESTful-API für Anfragen an den App Runner-Dienst. Sie können die -API verwenden, um App Runner -Ressourcen in IhrerAWS-Konto.

Sie können die -API direkt in Ihrem Anwendungscode aufrufen, oder Sie können eine derAWS-SDKs zu öffnen. Verwenden Sie für Befehlszeilenskripte dieAWS CLI, um Anrufe an den App Runner-Dienst zu tätigen. Weitere Informationen zuAWSEntwickler-Tools finden Sie unterTools zum AufbauenAWS.

Vollständige API-Referenzinformationen finden Sie in derAWS App Runner-API-Referenz.

Sicherheit in App Runner

Cloud-Sicherheit hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die eingerichtet wurden, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das Modell der übergreifenden Verantwortlichkeit beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud–AWSist zuständig für den Schutz der Infrastruktur, dieAWS-Services in derAWS Cloud.AWSSie erhalten außerdem Services, die Sie sicher verwenden können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Informationen zu den Compliance-Programmen, die fürAWS App Runnerfinden Sie unterAWSCompliance-Programm in Scope.
- Sicherheit in der Cloud— Ihre Verantwortung wird durch dieAWS-Dienst, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

In dieser Dokumentation wird erläutert, wie das Modell der geteilten Verantwortlichkeit bei der Verwendung von App Runner zum Tragen kommt. In den folgenden Themen wird erläutert, wie Sie App Runner so konfigurieren können, dass Ihre Sicherheits- und Compliance-Ziele erreicht werden. Sie erfahren auch, wie Sie andereAWS-Ressourcen überwachen und schützen Sie Ihre App Runner -Ressourcen.

Themen

Datenschutz in App Runner

DieAWS Modell übergreifender Verantwortlichkeitgilt für den Datenschutz inAWS App Runner.
Wie in diesem Modell beschrieben,AWSist zuständig für den Schutz der globalen Infrastruktur, die alleAWSWolke. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Dieser Inhalt enthält die Sicherheitskonfigurations- und -verwaltungstasks für dieAWS-Dienste, die Sie verwenden. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Weitere Informationen zum Datenschutz in Europa finden Sie in derAWSModell übergreifender Verantwortlichkeit und DSGVOBlogbeitrag auf derAWSBlogsicherheit.

Wir empfehlen aus Gründen des Datenschutzes, dass SieAWSund richten Sie individuelle Benutzerkonten mitAWS Identity and Access Management(IAM). So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem sollten Sie die Daten mit folgenden Methoden schützen:

- Verwenden Sie die Multi-Factor Authentication (MFA) für jedes Konto.
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir empfehlen TLS 1.2 oder höher.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie moderne verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu sichern.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern wie z. B. im Feld Name keine sensiblen, identifizierenden Informationen wie Kontonummern von Kunden einzugeben. Dies gilt auch, wenn Sie mit App Runner oder anderenAWS-Dienste mit der Konsole, API,AWS CLI, oderAWS-SDKs. Alle Daten, die Sie in App Runner oder andere Services eingeben, werden möglicherweise in Diagnoseprotokolle aufgenommen. Wenn Sie eine URL für einen externen Server bereitstellen, schließen Sie keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL ein.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Themen

Datenschutz durch Verschlüsselung

AWS App Runnerliest Ihre Anwendungsquelle (Quellbild oder Quellcode) aus einem von Ihnen angegebenen Repository und speichert sie für die Bereitstellung in Ihrem Dienst. Weitere Informationen finden Sie unter Architektur und Konzepte.

Datenschutz bezieht sich auf den Schutz von Daten, währendim Transit(auf Reisen von und nach App Runner) undim Ruhezustand(während es inAWSRechenzentren).

Weitere Informationen zum Datenschutz finden Sie unterthe section called “Datenschutz”.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Verschlüsselung während der Übertragung

Sie können den Datenschutz während der Übertragung auf zwei Arten erreichen: Verschlüsseln Sie die Verbindung mit Transport Layer Security (TLS) oder verwenden Sie die clientseitige Verschlüsselung (wobei das Objekt vor dem Senden verschlüsselt wird). Beide Methoden sind für den Schutz Ihrer Anwendungsdaten gültig. Um die Verbindung zu sichern, verschlüsseln Sie sie mit TLS, wenn Ihre Anwendung, ihre Entwickler und Administratoren und ihre Endbenutzer Objekte senden oder empfangen. App Runner richtet Ihre Anwendung so ein, dass sie Traffic über TLS empfängt.

Die clientseitige Verschlüsselung ist keine gültige Methode zum Schutz des Quellabbilds oder Quellcodes, die Sie App Runner zur Bereitstellung bereitstellen. App Runner benötigt Zugriff auf Ihre Anwendungsquelle, sodass sie nicht verschlüsselt werden kann. Stellen Sie daher sicher, dass Sie die Verbindung zwischen Ihrer Entwicklungs- oder Bereitstellungsumgebung und App Runner sichern.

Verschlüsselung im Ruhezustand und Schlüsselverwaltung

Um die Daten Ihrer Anwendung im Ruhezustand zu schützen, verschlüsselt App Runner alle gespeicherten Kopien Ihres Anwendungsquellabbilds oder Quell-Bundles. Wenn Sie einen App Runner-Service erstellen, können Sie einen Customer Master Key (CMK) bereitstellen. Wenn

Sie einen angeben, verwendet App Runner Ihren bereitgestellten Schlüssel, um Ihre Quelle zu verschlüsseln. Wenn Sie dies nicht tun, verwendet App Runner eineAWS-verwalteter CMK.

Weitere Informationen zu den Parametern der App Runner-Diensterstellung finden Sie unterCreateService. Weitere Informationen zuAWS Key Management Service(AWS KMS) finden Sie unterAWS Key Management ServiceEntwicklerhandbuch.

Richtlinie für den Datenverkehr zwischen Netzwerken

App Runner verwendet Amazon Virtual Private Cloud (Amazon VPC), um Grenzen zwischen Ressourcen in Ihrer App Runner-Anwendung zu erstellen und den Datenverkehr zwischen ihnen, Ihrem lokalen Netzwerk und dem Internet zu steuern. Weitere Informationen zur Amazon VPC -Sicherheit finden Sie unterRichtlinie für den Datenverkehr zwischen Netzwerken in Amazon VPCimAmazon VPC Benutzerhandbuch.

Weitere Informationen zum Sichern von Anforderungen an App Runner mithilfe eines VPC Endpunkts finden Sie unterthe section called “VPC-Endpunkte”.

Weitere Informationen zum Datenschutz finden Sie unterthe section called “Datenschutz”.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Verwenden von App Runner mit VPC Endpunkten

IhreAWS-Anwendung integriertAWS App Runner-Dienstleistungen mit anderenAWS-Dienste, die in einemAmazon Virtual Private Cloud (Amazon VPC). Teile Ihrer Anwendung stellen möglicherweise Anforderungen an App Runner aus der VPC. Sie können beispielsweise Folgendes verwenden:AWS CodePipeline, um fortlaufend in Ihrem App Runner-Dienst bereitzustellen. Eine Möglichkeit, die Sicherheit Ihrer Anwendung zu verbessern, besteht darin, diese App Runner-Anfragen (und Anfragen an andereAWS-Dienste) über einen VPC Endpunkt.

Ein VPC-Endpunkt ermöglicht Ihnen die Herstellung einer privaten Verbindung zwischen Ihrer VPC, unterstützten AWS-Services und VPC-Endpunktservices über AWS PrivateLink, ohne dass ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder eine AWS Direct Connect-Verbindung erforderlich sind.

Die Ressourcen in Ihrer VPC verwenden für die Interaktion mit App Runner-Ressourcen keine öffentlichen IP-Adressen. Der Datenverkehr zwischen Ihrer VPC und App Runner verlässt das Amazon-Netzwerk nicht. Weitere Informationen über VPC Endpunkte finden Sie unterVPC-EndpunkteimAWS PrivateLinkRichtlinie.

Unterstützung von App RunnerAWS PrivateLinkunterstützt. Diese Lösung stellt eine private Verbindung mit App Runner her und verhindert den Transport von Daten über das öffentliche Internet. So aktivieren Sie Ihre Anwendung das Senden von Anfragen an App Runner mithilfe vonAWS PrivateLinkkonfigurieren Sie einen Typ von VPC Endpunkt, der alsSchnittstellen-VPC-Endpunkt(Interface-Endpunkt). Weitere Informationen finden Sie unterSchnittstellen-VPC Endpunkte (AWS PrivateLink)imAWS PrivateLinkRichtlinie.

> **Note**
>
> Die Webanwendung in Ihrem App Runner-Dienst wird in einer VPC ausgeführt, die App Runner bereitstellt und konfiguriert. Diese VPC ist öffentlich — sie ist mit dem öffentlichen Internet verbunden. App Runner unterstützt die Ausführung Ihrer Anwendung in einer privaten VPC oder das Erstellen eines VPC-Endpunkts für sie nicht.

Einrichtung eines VPC Endpunkts für App Runner

Um den Schnittstellen-VPC Endpunkt für den App Runner-Service in Ihrer VPC zu erstellen, folgen Sie derErstellen eines Schnittstellenendpunkts-Prozedur imAWS PrivateLinkRichtlinie. Wählen Sie für Service Name (Servicename) `com.amazonaws.region.apprunner` aus.

Überlegungen zum VPC Datenschutz

> **Important**
>
> Die Verwendung eines VPC Endpunkts für App Runner garantiert nicht, dass der gesamte Datenverkehr Ihrer VPC außerhalb des Internets bleibt. Die VPC ist möglicherweise öffentlich (mit dem Internet verbunden), und einige Teile Ihrer Lösung verwenden möglicherweise keine VPC Endpunkte, umAWSAPI-Aufrufe. Beispiel,AWS-Dienste können andere -Services über ihre öffentlichen Endpunkte aufrufen. Wenn für die Lösung in Ihrer VPC Datenschutz erforderlich ist, lesen Sie diesen Abschnitt.

Beachten Sie Folgendes, um den Datenschutz des Netzwerkverkehrs in Ihrer VPC zu gewährleisten:

- Aktivieren des DNS-Namens— Teile Ihrer Anwendung senden möglicherweise weiterhin Anfragen an App Runner über das Internet mithilfe der`apprunner.region.amazonaws.com`-Endpunkt. Wenn Ihre VPC mit einem Zugang zum öffentlichem Internet konfiguriert ist, werden diese

Anfragen ohne Hinweis auf Sie erfolgreich. Sie können dies durch die Aktivierung von Enable DNS name (DNS-Namen aktivieren) während der Endpunkterstellung verhindern (standardmäßig „true“). Hierdurch wird ein DNS-Eintrag in Ihrer VPC hinzugefügt, der den Endpunkt für den öffentlichen Service dem VPC-Schnittstellenendpunkt zuordnet.

- Konfigurieren von VPC Endpunkten für zusätzliche Services— Ihre Lösung sendet möglicherweise Anfragen an andereAWS-Services. Beispiel,AWS CodePipelinemöglicherweise Anfragen anAWS CodeBuild. Konfigurieren Sie auch für diese Services VPC Endpunkte, und aktivieren Sie DNS-Namen auf diesen Endpunkten.

Verwenden von Endpunktrichtlinien zur Steuerung des Zugriffs mit VPC-Endpunkten

Standardmäßig ermöglicht ein VPC-Endpunkt den vollständigen Zugriff auf den Service, dem er zugeordnet ist. Wenn Sie einen VPC Endpunkt für App Runner erstellen oder ändern, können Sie-Endpunktrichtliniezu ihm.

Eine Endpunkt-Richtlinie ist einAWS Identity and Access Management(IAM) -Ressourcenrichtlinie, die den Zugriff vom Endpunkt aus auf den angegebenen Service steuert. Die Endpunktrichtlinie ist für den jeweiligen Endpunkt spezifisch. Sie ist von allen Benutzer- oder Instance-IAM-Richtlinien, die in Ihrer Umgebung möglicherweise vorhanden sind, getrennt; weder überschreibt sie diese noch ersetzt sie diese. Weitere Informationen zum Erstellen und Verwenden von VPC Endpunktrichtlinien finden Sie unterKontrollieren des Zugriffs auf Services mit VPC-EndpunktenimAWS PrivateLinkRichtlinie.

Im folgenden Beispiel wird der schreibgeschützte Zugriff vom VPC Endpunkt auf App Runner ermöglicht. Dies sind minimale Zugriffsrechte bei Verwendung des VPC Endpunkts. Prinzipale haben möglicherweise zusätzliche Berechtigungen über andere Richtlinien.

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

Identity and Access Management für App Runner

AWS Identity and Access Management(IAM) ist einAWS-Dienst, der einem Administrator hilft, den Zugriff auf zu steuernAWSRessourcen schätzen. IAM-Administratoren steuern, werauthentifiziert(eingeloggt) undautorisiert(über Berechtigungen), um App Runner-Ressourcen zu verwenden. IAM ist einAWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Themen

Audience

Verwendung von durchAWS Identity and Access ManagementDie IAM (IAM) ist von der Art Ihrer Aktivitäten in App Runner abhängig.

Service-BenutzerWenn Sie den App Runner-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie bei Ihrer Arbeit weitere App Runner-Funktionen ausführen, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle verstehen, kann Ihnen dies helfen, die richtigen Berechtigungen von Ihrem Administrator anzufordern. Wenn Sie nicht auf eine Funktion in App Runner zugreifen können, informieren Sie sich unterProblembehandlung bei Identitäts- und Zugriffsbehebung für App Runner.

Service-AdministratorWenn Sie in Ihrem Unternehmen die Verantwortung für App Runner -Ressourcen haben, haben Sie wahrscheinlich vollständigen Zugriff auf App Runner. Ihre Aufgabe besteht darin, die App Runner-Funktionen und -Ressourcen festzulegen, auf die Mitarbeiter zugreifen

können. Sie müssen anschließend bei Ihrem IAM-Administrator entsprechende Änderungen für die Berechtigungen Ihrer Service-Benutzer anfordern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM zu verstehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit App Runner verwenden kann, finden Sie unterFunktionsweise von App Runner mit IAM.

IAM-AdministratorWenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf App Runner verfassen können. Beispiele für identitätsbasierte App Runner -Richtlinien, die Sie in verwenden können, finden Sie unterBeispiele für identitätsbasierte -Richtlinien für App Runner.

Authentifizierung mit Identitäten

Die Authentifizierung ist die Art und Weise, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Weitere Informationen zum Anmelden mit demAWS Management Consolefinden Sie unterAnmelden bei derAWS Management Consoleals IAM-Benutzer oder Stammbenutzerim*IAM-Benutzerhandbuch*.

Sie müssen sein.authentifiziert(angemeldet beiAWS) alsAWS-Kontenstammbenutzer, einen IAM-Benutzer oder eine IAM-Rolle übernehmen. Sie können auch die Single-Sign-on-Authentifizierung Ihres Unternehmens verwenden oder sich sogar über Google oder Facebook anmelden. In diesen Fällen hat Ihr Administrator vorher einen Identitätsverbund unter Verwendung von IAM-Rollen eingerichtet. Wenn Sie mit Anmeldeinformationen eines anderen Unternehmens auf AWS zugreifen, nehmen Sie indirekt eine Rolle an.

Um sich direkt bei derAWS Management ConsoleVerwenden Sie Ihr Passwort mit der E-Mail-Adresse Ihres Stammbenutzers oder dem IAM-Benutzernamen. Sie können aufAWSMit den Zugriffsschlüsseln Ihres Stammbenutzers oder IAM-Benutzers programmgesteuert.AWSstellt SDKs und Befehlszeilentools bereit, mit denen Sie Ihre Anforderung mit Ihren Anmeldeinformationen verschlüsselt signieren können. Wenn Sie keine AWS-Tools verwenden, müssen Sie die Anforderung selbst signieren. Hierzu verwenden Sie Signature Version 4, ein Protokoll für die Authentifizierung eingehender API-Anforderungen. Weitere Informationen zum Authentifizieren von Anforderungen finden Sie unterProzess mit Signatur Version 4im*AWS– Allgemeine Referenz*.

Unabhängig von der von Ihnen verwendeten Authentifizierungsmethode müssen Sie möglicherweise auch zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Factor Authentication (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen hierzu finden Sie unterVerwenden der Multi-Factor Authentication (MFA) inAWSim*IAM-Benutzerhandbuch*.

Stammbenutzer des AWS-Kontos

Wenn Sie ein AWS-Konto neu erstellen, enthält es zunächst nur eine einzelne Anmeldeidentität, die über Vollzugriff auf sämtliche AWS-Services und -Ressourcen im Konto verfügt. Diese Identität wird alsAWSaccountStammbenutzerSie können zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die bzw. das Sie beim Erstellen des Kontos verwendet haben. Wir raten ausdrücklich davon ab, den Stammbenutzer für Alltagsaufgaben einschließlich administrativen Aufgaben zu verwenden. Bleiben Sie stattdessen bei dem bewährten -Verfahren, den Stammbenutzer nur zu verwenden, um Ihren ersten IAM-Benutzer zu erstellen. Anschließend legen Sie die Anmeldedaten für den Stammbenutzer an einem sicheren Ort ab und verwenden sie nur, um einige Konto- und Service-Verwaltungsaufgaben durchzuführen.

IAM-Benutzer und -Gruppen

EinIAM-Benutzerist eine Identität innerhalb IhrerAWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung verfügt. Ein IAM-Benutzer kann langfristige Anmeldeinformationen wie Benutzername und Passwort oder einen Satz von Zugriffsschlüsseln haben. Informationen zum Generieren von Zugriffsschlüsseln finden Sie unter Verwalten von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch. Achten Sie beim Generieren von Zugriffsschlüsseln für einen IAM-Benutzer darauf, dass Sie das Schlüsselpaar anzeigen und sicher speichern. Sie können den geheimen Zugriffsschlüssel später nicht wiederherstellen. Stattdessen müssen Sie ein neues Zugriffsschlüsselpaar generieren.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen permanente langfristige Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM roles

EinIAM-Rolleist eine Identität innerhalb IhrerAWS-Konto mit spezifischen Berechtigungen. Sie ähnelt einem IAM-Benutzer, ist aber nicht mit einer bestimmten Person verknüpft. Sie können eine

IAM-Rolle vorübergehend in der ListeAWS Management ConsolevonWechseln von Rollen. Sie können eine Rolle annehmen, indem Sie eine AWS CLI- oder AWS-API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer kann eine IAM-Rolle annehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Zugriff für verbundene BenutzerStatt einen IAM-Benutzer zu erstellen, können Sie bereits vorhandene Identitäten von verwendenAWS Directory ServiceIhr Unternehmens-Benutzerverzeichnis oder einen Web-Identitätsanbieter. Diese werden als verbundene Benutzer bezeichnet. AWS weist einem verbundenen Benutzer eine Rolle zu, wenn der Zugriff über einen Identitätsanbieter angefordert wird. Weitere Informationen zu verbundenen Benutzern finden Sie unter Verbundene Benutzer und Rollen im IAM-Benutzerhandbuch.
- Kontenübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen sind die primäre Möglichkeit, um kontoübergreifenden Zugriff zu gewähren. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.
- Service-übergreifender Zugriff— EtwasAWS-Dienste verwenden Funktionen in anderenAWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon EC2 ausführt oder Objekte in Amazon S3 speichert. Ein Service kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Prinzipal BerechtigungenWenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in verwendenAWS, werden Sie als Auftraggeber betrachtet. Richtlinien erteilen einem Prinzipal Berechtigungen. Wenn Sie einige Services verwenden, könnten Sie eine Aktion ausführen, die dann eine andere Aktion in einem anderen Service auslöst. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Informationen dazu, ob eine Aktion zusätzliche abhängige Aktionen in einer Richtlinie erfordert, finden Sie unterAktionen, Ressourcen und Bedingungsschlüssel fürAWS App RunnerimService Authorization-Referenz.

- Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Service-Rollen bieten nur Zugriff innerhalb Ihres Kontos und können nicht genutzt werden, um Zugriff auf Services in anderen Konten zu erteilen. Ein IAM-Administrator kann eine Servicerolle in IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unterErstellen einer Rolle zum Delegieren von Berechtigungen an eineAWS-ServiceimIAM-Benutzerhandbuch.
- Serviceverknüpfte RolleEine serviceverknüpfte Rolle ist ein Typ einer Servicerolle, die mit einer verknüpft istAWS-Service. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

- Anwendungen, die auf Amazon EC2 ausgeführt werdenSie können eine IAM-Rolle nutzen, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden undAWS CLIoder .AWS-API-Anforderungen. Das ist empfehlenswerter als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff inAWSRichtlinien erstellen und sie an IAM-Identitäten anfügen oderAWSRessourcen schätzen. Eine Richtlinie ist ein Objekt in AWS, das, einer Identität oder Ressource zugeordnet, deren Berechtigungen definiert. Sie können sich als Root-Benutzer oder IAM-Benutzer anmelden oder eine IAM-Rolle übernehmen. Wenn Sie dann eine Anfrage stellen,AWSwertet die zugehörigen identitätsbasierten oder ressourcenbasierten Richtlinien aus. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren könnenAWSJSON-Richtlinien, um anzugeben, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Eine IAM-Entität (Benutzer oder Rolle) besitzt zunächst keine Berechtigungen. Anders ausgedrückt, können Benutzer standardmäßig keine Aktionen ausführen und nicht einmal ihr Passwort ändern. Um einem Benutzer die Berechtigung für eine Aktion zu erteilen, muss ein Administrator einem Benutzer eine Berechtigungsrichtlinie zuweisen. Alternativ kann der Administrator den Benutzer zu einer Gruppe hinzufügen, die über die gewünschten Berechtigungen verfügt. Wenn ein Administrator einer Gruppe Berechtigungen erteilt, erhalten alle Benutzer in dieser Gruppe diese Berechtigungen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Benutzer informationen über die AWS Management Console, die AWS CLI oder die AWS-API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Eingebundene Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto anfügen können. Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern.

Für die Ressource, an die die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, verbundene Benutzer oderAWS-Services.

Ressourcenbasierte Richtlinien sind eingebundene Richtlinien, die sich in diesem Service befinden. Sie können nicht verwenden.AWS-Richtlinien von IAM in einer ressourcenbasierten Richtlinie.

Zugriffskontrolllisten (ACLs)

Zugriffskontrolllisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3,AWS WAFAmazon VPC und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. Weitere Informationen zu ACLs finden Sie unter Zugriffskontrollliste (ACL) – Übersicht im Amazon Simple Storage Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die resultierenden Berechtigungen sind eine Schnittmenge der identitätsbasierten Richtlinien der Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs)SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) inAWS Organizations.AWS Organizationsist ein Service für die Gruppierung und zentrale Verwaltung mehrererAWS-Konten, die Ihr Unternehmen besitzt. Wenn Sie innerhalb einer Organisation alle Funktionen aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Die SCP beschränkt Berechtigungen für Entitäten in Mitgliedskonten, einschließlich

allerAWSStammbenutzer des -Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unterArbeitsweise von SCPsimAWS OrganizationsBenutzerhandbuch.

- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Weitere Informationen erhalten Sie unterAWSInformationen dazu, ob eine Anforderung zugelassen wird, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unterBewertungslogik für RichtlinienimIAM-Benutzerhandbuch.

Funktionsweise von App Runner mit IAM

Bevor Sie IAM verwenden, um Zugriff aufAWS App RunnerWenn Sie sich darüber informieren, welche IAM-Funktionen Sie mit App Runner verwenden können. So erhalten Sie eine allgemeine Übersicht über App Runner und andereAWS-Dienste mit IAM arbeiten, finden Sie unterAWSServices, die mit IAM funktionierenimIAM-Benutzerhandbuch.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Themen

- Identitätsbasierte Richtlinien für App Runner
- Ressourcenbasierte Richtlinien von App Runner
- Autorisierung basierend auf App Runner-Tags
- App Runner-Benutzerberechtigungen
- IAM-Rollen

Identitätsbasierte Richtlinien für App Runner

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter

denen Aktionen zugelassen oder abgelehnt werden. App Runner unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Actions

Administratoren könnenAWSJSON-Richtlinien, um anzugeben, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS-API-Operation. Es gibt einige Ausnahmen, wie z. B. „nur Berechtigung“-Aktionen, die keine übereinstimmende API-Operation haben. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen in App Runner verwenden das folgende Präfix vor der Aktion: `apprunner:`. Um jemandem beispielsweise die Berechtigung zum Ausführen einer Amazon EC2 Instance mit Amazon EC2 zu erteilen`RunInstances`API-Operation verwenden, schließen Sie die`ec2:RunInstances`Maßnahmen in ihrer Politik. Richtlinienanweisungen müssen entweder ein `Action`- oder ein `NotAction`-Element enthalten. App Runner definiert eine eigene Gruppe von Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service durchführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie folgendermaßen mit Kommas:

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

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

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

Eine Liste der App Runner-Aktionen finden Sie unterVon definierte AktionenAWS App RunnerimService Authorization-Referenz.

Resources

Administratoren könnenAWSJSON-Richtlinien, um anzugeben, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf die die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource`- oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

App Runner-Ressourcen verfügen über die folgende ARN -Struktur:

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

Weitere Informationen zum Format von ARNs finden Sie unterAmazon-Ressourcennamen (ARNs) undAWS-Service-NamespacesimAWS– Allgemeine Referenz.

Um beispielsweise diemy`-service`-Service in Ihrer Anweisung verwenden Sie den folgenden ARN:

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

Um alle Services anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

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

Einige App Runner-Aktionen, z. B. zum Erstellen von Ressourcen, können auf bestimmten Ressourcen nicht ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

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

Eine Liste der App Runner-Ressourcentypen und ihrer ARNs finden Sie unterVon definierte RessourcenAWS App RunnerimService Authorization-Referenz. Informationen dazu, mit welchen Aktionen Sie den ARN der einzelnen Ressourcen angeben können, finden Sie unterVon definierte AktionenAWS App Runner.

Bedingungsschlüssel

Administratoren könnenAWSJSON-Richtlinien, um anzugeben, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Mithilfe des Elements `Condition`(oder des Blocks `Condition`) können Sie die Bedingungen angeben, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet AWS die Bedingung mittels einer logischen `OR`-Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWSunterstützt globale Bedingungsschlüssel und servicespezifische Bedingungsschlüssel. So zeigen Sie alle an:AWSAllgemeine Bedingungsschlüssel finden Sie unterAWSGlobale - BedingungskontextschlüsselimIAM-Benutzerhandbuch.

App Runner unterstützt die Verwendung einiger globaler Bedingungsschlüssel. So zeigen Sie alle an:AWSAllgemeine Bedingungsschlüssel finden Sie unterAWSGlobale - BedingungskontextschlüsselimIAM-Benutzerhandbuch.

App Runner definiert einen Satz servicespezifische Bedingungsschlüssel. Darüber hinaus unterstützt App Runner tag-basierte Zugriffssteuerung, die mit Bedingungsschlüsseln implementiert wird. Details dazu finden Sie unter the section called “Autorisierung basierend auf App Runner-Tags”.

Eine Liste der -Bedingungsschlüssel von App Runner finden Sie unterBedingungsschlüssel fürAWS App RunnerimService Authorization-Referenz. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unterVon definierte AktionenAWS App Runner.

Examples

Beispiele für identitätsbasierte App Runner-Richtlinien finden Sie unterBeispiele für identitätsbasierte -Richtlinien für App Runner.

Ressourcenbasierte Richtlinien von App Runner

App Runner unterstützt keine ressourcenbasierten Richtlinien.

Autorisierung basierend auf App Runner-Tags

Sie können Tags an App Runner-Ressourcen anfügen oder Tags in einer Anforderung an App Runner übergeben. Um den Zugriff basierend auf Tags zu steuern, stellen Sie Tag-Informationen im Bedingungselement einer Richtlinie unter Verwendung der Bedingungsschlüssel `apprunner:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* oder `aws:TagKeys` bereit. Weitere Informationen zum Tagging von App Runner-Ressourcen finden Sie unterthe section called “Konfiguration”.

Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Tags dieser Ressource finden Sie unter Steuern des Zugriffs auf App Runner-Dienste basierend auf Tags.

App Runner-Benutzerberechtigungen

Um App Runner verwenden zu können, benötigen IAM-Benutzer Berechtigungen für App Runner-Aktionen. Eine übliche Methode zum Erteilen von Berechtigungen für Benutzer besteht darin, IAM-Benutzer oder -Gruppen eine Richtlinie anzuhängen. Weitere Informationen zum Verwalten von Benutzerberechtigungen finden Sie unterÄndern von Berechtigungen für einen IAM-BenutzerimIAM-Benutzerhandbuch.

App Runner bietet zwei verwaltete Richtlinien, die Sie an Ihre Benutzer anfügen können.

- `AWSAppRunnerReadOnlyAccess`— Gewährt Berechtigungen zum Auflisten und Anzeigen von Details zu App Runner-Ressourcen.
- `AWSAppRunnerFullAccess`— Erteilt Berechtigungen für alle App Runner-Aktionen.

Um die Benutzerberechtigungen genauer zu steuern, können Sie eine benutzerdefinierte Richtlinie erstellen und sie Ihren Benutzern zuordnen. Details dazu finden Sie unter .Erstellen von IAM-RichtlinienimIAM-Benutzerhandbuch.

Beispiele für Benutzerrichtlinien finden Sie unterthe section called “Benutzerrichtlinien”.

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": "*"
    }
  ]
}
```

IAM-Rollen

EinIAM-Rolleist eine Entität innerhalb IhrerAWS-Kontomit spezifischen Berechtigungen.

Serviceverknüpfte Rollen

Serviceverknüpfte Rollen erlauben AWS-Services den Zugriff auf Ressourcen in anderen Services, um eine Aktion in Ihrem Auftrag auszuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

App Runner unterstützt serviceverknüpfte Rollen. Informationen zum Erstellen oder Verwalten von serviceverknüpften App Runner-Rollen finden Sie unterthe section called “Verwenden von serviceverknüpften Rollen”.

Servicerollen

Diese Funktion ermöglicht einem Service das Annehmen einer Servicerolle in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Services, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem IAM-Konto angezeigt und gehören zum Konto. Dies bedeutet, dass ein IAM-Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Services beeinträchtigen.

App Runner unterstützt einige Dienstrollen.

Rolle für den Zugriff

Die Zugriffsrolle ist eine Rolle, die App Runner für den Zugriff auf Bilder in der Amazon Elastic Container Registry (Amazon ECR) in Ihrem Konto verwendet. Es ist erforderlich, um auf ein Bild in Amazon ECR zuzugreifen und ist für Amazon ECR Public nicht erforderlich. Bevor Sie einen Service basierend auf einem Image in Amazon ECR erstellen, verwenden Sie IAM, um eine Service-Rolle zu erstellen, und verwenden Sie die`AWSAppRunnerServicePolicyForECRAccess`verwaltete Richtlinie darin. Sie können diese Rolle dann an App Runner übergeben, wenn Sie dieCreateService-API in derAuthenticationConfiguration-Struktur (ein Mitglied derSourceConfiguration-Parameter) oder wenn Sie mit der App Runner-Konsole einen Dienst erstellen.

AWSAppRunnerServicePolicyForECRAccess

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

> **Note**
>
> Wenn Sie eine eigene benutzerdefinierte Richtlinie für Ihre Zugriffsrolle erstellen, geben Sie`"Resource": "*"`für`ecr:GetAuthorizationToken`Aktion Token können für den Zugriff auf jede Amazon ECR-Registrierung verwendet werden, auf die Sie Zugriff haben.

Achten Sie beim Erstellen Ihrer Zugriffsrolle darauf, eine Vertrauensrichtlinie hinzuzufügen, die den App Runner-Dienstprinzipal deklariert`build.apprunner.amazonaws.com`als vertrauenswürdige Entität verwenden.

Vertrauensrichtlinie für eine Zugriffsrolle

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

Wenn Sie mit der App Runner-Konsole einen Dienst erstellen, kann die Konsole automatisch eine Zugriffsrolle für Sie erstellen und diese für den neuen Dienst auswählen. Die Konsole listet auch andere Rollen in Ihrem Konto auf, und Sie können eine andere Rolle auswählen, wenn Sie möchten.

Instance-Rolle

Die Instanzrolle ist eine optionale Rolle, die App Runner verwendet, um Berechtigungen fürAWS-Dienstaktionen, die Ihr Anwendungscode aufruft. Bevor Sie einen App Runner-Dienst erstellen, erstellen Sie mithilfe von IAM eine Dienstrolle mit den Berechtigungen, die Ihr Anwendungscode benötigt. Sie können diese Rolle dann an App Runner imCreateService-API oder wenn Sie die App Runner-Konsole verwenden, um einen Dienst zu erstellen.

Achten Sie beim Erstellen der Instanzrolle darauf, eine Vertrauensrichtlinie hinzuzufügen, die den App Runner-Dienstprinzipal deklariert`tasks.apprunner.amazonaws.com`als vertrauenswürdige Entität verwenden.

Vertrauensrichtlinie für eine Instanzrolle

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

Wenn Sie mit der App Runner-Konsole einen Dienst erstellen, listet die Konsole die Rollen in Ihrem Konto auf, und Sie können die Rolle auswählen, die Sie für diesen Zweck erstellt haben.

Weitere Informationen zum Erstellen eines -Service finden Sie unterthe section called “Erstellung”.

> **Note**
>
> Die Berechtigungen, die die Instanzrolle bereitstellen sollte, hängen vollständig von Ihrer Anwendung ab. Ihr Code ruft möglicherweise keineAWSAPIs. In diesem Fall müssen Sie App Runner keine Instanzrolle bereitstellen.
> Für den Fall, dass Ihr CodeAWSAPIs haben wir keine Möglichkeit zu antizipieren, welche Anrufe sie sind. Daher bieten wir keine verwaltete Richtlinie für Instanzrollen an. Sie müssen die erforderlichen Berechtigungen explizit in Ihre Instanzrolle einschließen oder eine eigene benutzerdefinierte Richtlinie erstellen und in der Instanzrolle verwenden.

Beispiele für identitätsbasierte -Richtlinien für App Runner

IAM-Benutzer und -Rollen besitzen standardmäßig keine Berechtigungen zum Erstellen oder Ändern vonAWS App RunnerRessourcen schätzen. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS CLI oder AWS-API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den IAM-Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Themen

- Bewährte Methoden für Richtlinien
- Benutzerrichtlinien
- Steuern des Zugriffs auf App Runner-Dienste basierend auf Tags

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien sind sehr leistungsfähig. Sie können festlegen, ob jemand App Runner-Ressourcen in Ihrem Konto erstellen oder löschen oder auf sie zugreifen kann. Dies kann zusätzliche Kosten für Ihr AWS-Konto verursachen. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mitAWSVon verwaltete Richtlinien— Um App Runner schnell zu verwenden, verwenden SieAWS-Richtlinien, um Ihren Mitarbeitern die Berechtigungen zu gewähren, die sie benötigen. Diese Richtlinien sind bereits in Ihrem Konto verfügbar und werden von AWS gewartet und aktualisiert. Weitere Informationen finden Sie unterErste Schritte mit Berechtigungen mitAWSVon verwaltete RichtlinienimIAM-Benutzerhandbuch.
- Gewähren von geringsten Rechten – Gewähren Sie beim Erstellen benutzerdefinierter Richtlinien nur die Berechtigungen, die zum Ausführen einer Aufgabe erforderlich sind. Beginnen Sie mit einem Mindestsatz von Berechtigungen und gewähren Sie zusätzliche Berechtigungen wie erforderlich. Dies ist sicherer, als mit Berechtigungen zu beginnen, die zu weit gefasst sind, und dann später zu versuchen, sie zu begrenzen. Weitere Informationen finden Sie unter Gewähren von geringsten Rechten im IAM-Benutzerhandbuch.
- Aktivieren von MFA für sensible Vorgänge – Fordern Sie von IAM-Benutzern die Verwendung von Multi-Factor Authentication (MFA), um zusätzliche Sicherheit beim Zugriff auf sensible Ressourcen oder API-Operationen zu bieten. Weitere Informationen finden Sie unterVerwenden der Multi-Factor Authentication (MFA) inAWSimIAM-Benutzerhandbuch.
- Verwenden von Richtlinienbedingungen für zusätzliche Sicherheit – Definieren Sie die Bedingungen, unter denen Ihre identitätsbasierten Richtlinien den Zugriff auf eine Ressource zulassen, soweit praktikabel. Beispielsweise können Sie Bedingungen schreiben, die eine Reihe

von zulässigen IP-Adressen festlegen, von denen eine Anforderung stammen muss. Sie können auch Bedingungen schreiben, die Anforderungen nur innerhalb eines bestimmten Datums- oder Zeitbereichs zulassen oder die Verwendung von SSL oder MFA fordern. Weitere Informationen finden Sie unterIAM-JSON-Richtlinienelemente: BedingungimIAM-Benutzerhandbuch.

Benutzerrichtlinien

IAM-Benutzer benötigen einen Mindestsatz von Berechtigungen, um auf die App Runner-Konsole zugreifen zu können. Diese Berechtigungen müssen Ihnen das Auflisten und Anzeigen von Details zu den App Runner-Ressourcen in IhremAWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die restriktiver als die mindestens erforderlichen Berechtigungen ist, funktioniert die Konsole nicht wie vorgesehen für Benutzer mit dieser Richtlinie.

App Runner bietet zwei verwaltete Richtlinien, die Sie an Ihre Benutzer anfügen können.

- `AWSAppRunnerReadOnlyAccess`— Gewährt Berechtigungen zum Auflisten und Anzeigen von Details zu App Runner-Ressourcen.
- `AWSAppRunnerFullAccess`— Erteilt Berechtigungen für alle App Runner-Aktionen.

Um sicherzustellen, dass Benutzer die App Runner-Konsole verwenden können, fügen Sie mindestens die`AWSAppRunnerReadOnlyAccess`-Richtlinie an die Benutzer. Sie können die`AWSAppRunnerFullAccess`oder fügen Sie bestimmte zusätzliche Berechtigungen hinzu, um Benutzern das Erstellen, Ändern und Löschen von Ressourcen zu ermöglichen. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die den API-Operation entsprechen, die Benutzer ausführen können möchten.

In den folgenden Beispielen finden Sie benutzerdefinierte Benutzerrichtlinien. Sie können sie als Ausgangspunkt für die Definition eigener benutzerdefinierter Benutzerrichtlinien verwenden. Kopieren Sie das Beispiel, oder entfernen Sie Aktionen, erweitern Sie Ressourcen und fügen Sie Bedingungen hinzu.

Beispiel: Benutzerrichtlinie für Konsolen- und Verbindungsverwaltung

Diese Beispielrichtlinie ermöglicht den Konsolenzugriff und ermöglicht die Erstellung und Verwaltung von Verbindungen. Es lässt die Erstellung und Verwaltung von App Runner-Diensten nicht zu. Sie

kann einem Benutzer zugeordnet werden, dessen Aufgabe es ist, den App Runner-Dienstzugriff auf Quellcode-Assets zu verwalten.

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

Beispiel: Benutzerrichtlinien, die Bedingungsschlüssel verwenden

In den Beispielen in diesem Abschnitt werden bedingte Berechtigungen veranschaulicht, die von einigen Ressourceneigenschaften oder Aktionsparametern abhängen.

Diese Beispielrichtlinie ermöglicht das Erstellen eines App Runner-Dienstes, verweigert jedoch die Verwendung einer Verbindung namens`prod`.

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

```
}
```

In dieser Beispielrichtlinie wird das Aktualisieren eines App Runner-Dienstes mit dem Namen`preprod`nur mit einer automatischen Skalierungskonfiguration namens`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/*"
        }
      }
    }
  ]
}
```

Steuern des Zugriffs auf App Runner-Dienste basierend auf Tags

Sie können in Ihrer identitätsbasierten Richtlinie Bedingungen für die Steuerung des Zugriffs auf App Runner-Ressourcen auf der Basis von Tags verwenden. Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen können, die das Löschen eines App Runner-Service ermöglicht. Die Berechtigung wird jedoch nur gewährt, wenn der Wert des `0wner`-Tags der Name des Benutzers ist. Diese Richtlinie gewährt auch die Berechtigungen, die für die Ausführung dieser Aktion auf der Konsole erforderlich sind.

```
{
  "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}"}
      }
    }
  ]
}
```

Sie können diese Richtlinie den IAM-Benutzern in Ihrem Konto zuweisen. Wenn ein Benutzer namens`richard-roe`Um einen App Runner-Service zu löschen, muss der Service mit Tags versehen sein`0wner=richard-roe`oder .`owner=richard-roe`. Andernfalls wird der Zugriff abgelehnt. Der Tag-Schlüssel `0wner` der Bedingung stimmt sowohl mit `0wner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unterIAM-JSON-Richtlinienelemente: BedingungimIAM-Benutzerhandbuch.

Verwenden von serviceverknüpften Rollen für App Runner

AWS App RunnernutztAWS Identity and Access Management(IAM)Service-verknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit App Runner verknüpft ist. Serviceverknüpfte Rollen werden von App Runner vordefiniert und schließen alle Berechtigungen ein, die der -Service zum Aufrufen andererAWS-Dienstleistungen in Ihrem Namen.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von App Runner, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. App Runner definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur App Runner die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen juristischen Stelle von IAM zugeordnet werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre zugehörigen AWS-Ressourcen gelöscht wurden. Dies schützt Ihre App Runner-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS-Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte

Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Berechtigungen von serviceverknüpften Rollen für App Runner

App Runner verwendet die serviceverknüpfte Rolle namensAWSServiceRoleForApprunner.

Mit der Rolle kann App Runner die folgenden Aufgaben ausführen:

- Push von Protokollen an Amazon CloudWatch Logs -Gruppen.
- Erstellen Sie Amazon CloudWatch Events regeln, um Pushs für Amazon Elastic Container Registry (Amazon ECR) zu abonnieren.

Die servicegebundene Rolle AWSServiceRoleForApprunner vertraut, dass die folgenden Services die Rolle übernehmen:

- `apprunner.amazonaws.com`

Die Berechtigungsrichtlinie der servicegebundenen Rolle AWSServiceRoleForApprunner enthält alle Berechtigungen, die App Runner benötigt, um Aktionen in Ihrem Namen durchzuführen.

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"
        }
      }
    }
  ]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für App Runner

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen App Runner-Dienst imAWS Management Console, dieAWS CLI, oder dieAWSerstellt App Runner die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. App Runner erstellt App Runner die serviceverknüpfte Rolle erneut für Sie.

Bearbeiten einer serviceverknüpften Rolle für App Runner

App Runner verhindert die Bearbeitung der servicegebundenen Rolle AWSServiceRoleForApprunner. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für App Runner

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

In App Runner bedeutet dies, dass alle App Runner-Dienste in Ihrem Konto gelöscht werden. Weitere Informationen zum Löschen von App Runner-Diensten finden Sie unterthe section called “Löschung”.

> **Note**
>
> Wenn der App Runner-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, dieAWS CLI, oder dieAWSAPI zum Löschen der serviceverknüpften AWSServiceRoleForAppRunner-Rolle löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte Rollen für App Runner

App Runner unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unterAWS App Runner-Endpunkte und -KontingenteimAWS– Allgemeine Referenz.

Von AWS verwaltete Richtlinien für AWS App Runner

Um -Benutzern, -Gruppen und -Rollen Berechtigungen hinzuzufügen, ist es einfacher,AWSverwalteten Richtlinien, als Richtlinien selbst zu schreiben. Es braucht Zeit und FachwissenKundenverwaltete IAM-Richtlinien erstellenUm Ihrem Team nur die Berechtigungen bereitzustellen, die es benötigt. Um einen schnellen Einstieg zu wünschen, verwenden Sie unserAWSVon verwaltete Richtlinien. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in IhremAWSKonto. Weitere Informationen zuAWSWeitere Informationen finden Sie unterAWSVon verwaltete RichtlinienimIAM-Benutzerhandbuch.

AWSDienste pflegen und aktualisierenAWSVon verwaltete Richtlinien. Sie können die Berechtigungen nicht in ändernAWSVon verwaltete Richtlinien. Dienste fügen gelegentlich zusätzliche Berechtigungen zu einemAWSverwaltete Richtlinie, um neue Funktionen zu unterstützen. Diese Art von Aktualisierung wirkt sich auf alle Identitäten (Benutzer, Gruppen und Rollen) aus, an die die Richtlinie angefügt ist. Dienste aktualisieren am ehesten eineAWSverwaltete Richtlinie, wenn ein neues Feature gestartet wird oder wenn neue Vorgänge verfügbar sind. Dienste entfernen keine Berechtigungen aus einemAWSverwaltete Richtlinie, sodass Richtlinienaktualisierungen Ihre vorhandenen Berechtigungen nicht verletzen.

Außerdem gilt Folgendes:AWSunterstützt verwaltete Richtlinien für Jobfunktionen, die mehrere Services umfassen. Zum Beispiel, dasReadOnlyAccess AWSDiese verwaltete Richtlinie bietet schreibgeschützten Zugriff auf alleAWSund -Ressourcen. Wenn ein Dienst eine neue Funktion startet,AWSfügt schreibgeschützte Berechtigungen für neue Vorgänge und Ressourcen hinzu. Eine Liste und Beschreibungen der Richtlinien für Jobfunktionen finden Sie unterAWSVon verwaltete Richtlinien für Job-FunktionenimIAM-Benutzerhandbuch.

App Runner-Updates fürAWSVon verwaltete Richtlinien

Details zu Updates für anzeigenAWSverwaltete Richtlinien für App Runner, da dieser Dienst mit der Verfolgung dieser Änderungen begonnen hat. Wenn Sie automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der Seite „App Runner Document History“.

Änderung	Beschreibung	Datum
AppRunnerServiceRolePolicy— Neue -Richtlinien	App Runner hat eine neue Richtlinie hinzugefügt, mit der App Runner Anrufe an Amazon CloudWatch Logs und Amazon CloudWatch Events im Auftrag von App Runner Services tätigen kann.	1. März 2021
AWSAPPrunnerReadOnlyAccess— Neue -Richtlinien	App Runner hat eine neue Richtlinie hinzugefügt, mit der Benutzer Details zu App Runner-Ressourcen auflisten und anzeigen können.	1. März 2021
AWSAPPRUNNERFullAccess— Neue -Richtlinien	App Runner hat eine neue Richtlinie hinzugefügt, mit der Benutzer alle App Runner-Aktionen ausführen können.	1. März 2021
AWSAPPrunnerServicePolicyForecraccess— Neue -Richtlinien	App Runner hat eine neue Richtlinie hinzugefügt, mit der App Runner Zugriff auf Amazon Elastic Container Registry (Amazon ECR) - Bilder in Ihrem Konto gestatten kann.	1. März 2021
App Runner hat mit der Verfolgung von Änderungen begonnen	App Runner begann mit der Verfolgung von Änderungen für seineAWSVon verwaltete Richtlinien.	1. März 2021

Problembehandlung bei Identitäts- und Zugriffsbehebung für App Runner

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mitAWS App Runnerund IAM.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Themen

- Ich bin nicht autorisiert, eine Aktion in App Runner auszuführen
- Ich bin Administrator und möchte anderen Personen oder einer Anwendung Zugriff auf App Runner ermöglichen.

- Ich möchte Personen außerhalb meinerAWS-Konto, um auf meine App Runner-Ressourcen zuzugreifen

Ich bin nicht autorisiert, eine Aktion in App Runner auszuführen

Wenn die AWS Management Console Ihnen mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, wenden Sie sich an Ihren Administrator, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen IhrAWS-Benutzername und -Passwort verwenden.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen`marymajor`versucht, die Konsole zum Anzeigen von Details zu einem App Runner-Service zu verwenden, jedoch nicht über`apprunner:DescribeService`-Berechtigungen.

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

In diesem Fall bittet Mary ihren Administrator um die Aktualisierung ihrer Richtlinien, um Zugriff auf die*my-example-service*-Ressource mithilfe der`apprunner:DescribeService`Aktion

Ich bin Administrator und möchte anderen Personen oder einer Anwendung Zugriff auf App Runner ermöglichen.

Um anderen Personen oder einer Anwendung Zugriff auf App Runner zu gewähren, müssen Sie eine IAM-Entität (Benutzer oder Rolle) für die Person oder Anwendung erstellen, die Zugriff benötigt. Diese verwendet dann die Anmeldeinformationen für diese Entität, um auf AWS zuzugreifen. Anschließend müssen Sie der Entität eine Richtlinie anfügen, die dieser die korrekten Berechtigungen in App Runner gewährt.

Informationen zum Einstieg finden Sie unter Erstellen Ihrer ersten delegierten IAM-Benutzer und -Gruppen im IAM-Benutzerhandbuch.

Ich möchte Personen außerhalb meinerAWS-Konto, um auf meine App Runner-Ressourcen zuzugreifen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können angeben, welchen Personen vertraut werden darf, damit diese die Rolle übernehmen können. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen finden Sie hier:

- Informationen dazu, ob App Runner diese Funktionen unterstützt, finden Sie unterFunktionsweise von App Runner mit IAM.
- Um zu erfahren, wie Sie den Zugriff auf Ihre Ressourcen überAWS-Konten, die Sie besitzen, finden Sie unterGewähren des Zugriffs für einen IAM-Benutzer in einem anderenAWS-Konto, das Sie besitzenimIAM-Benutzerhandbuch.
- Informationen dazu, wie Sie Drittanbieter Zugriff auf Ihre Ressourcen bereitstellenAWS-Konten finden Sie unterGewähren von Zugriff aufAWS-Konten von DrittenimIAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Protokollieren und überwachen in App Runner

Überwachung ist wichtig, um die Zuverlässigkeit, Verfügbarkeit und Performance Ihrer aufrechtzuerhalten.AWS App Runner-Service. Erfassung von Überwachungsdaten aus allen Teilen IhrerAWS-Lösung ermöglicht es Ihnen, einen Fehler einfacher zu debuggen, wenn einer auftritt. App Runner lässt sich mit mehrerenAWS-Tools zur Überwachung Ihrer App Runner-Dienste und zur Reaktion auf potenzielle Vorfälle.

Amazon CloudWatch-Alarme

Mit Amazon CloudWatch Alarmen können Sie eine Service-Metrik über einen von Ihnen angegebenen Zeitraum überwachen. Wenn die Metrik einen bestimmten Schwellenwert für eine bestimmte Anzahl von Zeiträumen überschreitet, erhalten Sie eine Benachrichtigung.

App Runner sammelt eine Vielzahl von Metriken über den Dienst als Ganzes und die Instanzen (Skalierungseinheiten), auf denen Ihr Webdienst ausgeführt wird. Weitere Informationen finden Sie unter Metriken (CloudWatch).

Anwendungsprotokolle

App Runner erfasst die Ausgabe Ihres Anwendungscodes und streamt sie zu Amazon CloudWatch Logs. Was in dieser Ausgabe ist, liegt an Ihnen. Sie können beispielsweise

detaillierte Aufzeichnungen von Anforderungen an Ihren Webdienst einfügen. Diese Protokolldatensätze können sich bei Sicherheits- und Zugriffsüberprüfungen als nützlich erweisen. Weitere Informationen finden Sie unter Protokolle (CloudWatch Logs).

AWS CloudTrailAktionsprotokoll

App Runner ist mitAWS CloudTrail, ein Service, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einer durchgeführten Aktionen bereitstellt.AWS-Dienst in App Runner. CloudTrail erfasst alle API-Aufrufe für App Runner als Ereignisse. Sie können die neuesten Ereignisse in der CloudTrail-Konsole anzeigen und einen Trail erstellen, um die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon Simple Storage Service-Bucket (Amazon S3) zu ermöglichen. Weitere Informationen finden Sie unter API-Aktionen (CloudTrail).

Compliance-Validierung für App Runner

Drittanbieter-Auditoren bewerten im Rahmen mehrerer AWS-Compliance-Programme die Sicherheit und Compliance von AWS App Runner. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

So erfahren Sie, ob App Runner oder andereAWS-Dienste befinden, finden Sie unterAWSServices in Scope nach Compliance-Programm. Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

Die Auditberichte von Drittanbietern lassen sich mit AWS Artifact herunterladen. Weitere Informationen finden Sie unter Berichte herunterladen in AWS Artifact.

Ihre Verantwortung in Bezug auf die Compliance bei der Verwendung von AWS-Services ergibt sich aus der Vertraulichkeit der Daten, den Compliance-Zielen des Unternehmens sowie den einschlägigen Gesetzen und Vorschriften. AWS stellt die folgenden Ressourcen zur Sicherstellung der Compliance bereit:

- Kurzanleitungen für Sicherheit und Compliance— In diesen Bereitstellungsleitfäden finden Sie wichtige Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von Basisumgebungen inAWS, die auf Sicherheit und Compliance ausgerichtet sind.
- Whitepaper für HIPAA-Sicherheit und -Compliance— Dieses Whitepaper beschreibt, wie UnternehmenAWS, um HIPAA-konforme Anwendungen zu erstellen.

> **Note**
>
> Nicht alle Services sind HIPAA-konform.

- AWSCompliance-Ressourcen— Diese Sammlung an Arbeitsmappen und Leitfäden könnte für Ihre Branche und Ihren Standort gelten.
- Bewerten von Ressourcen mit RegelnimAWS ConfigEntwicklerhandbuch— DieAWS ConfigDieser -Service bewertet, zu welchem Maß die Konfiguration Ihrer Ressourcen den internen Vorgehensweisen, Branchenrichtlinien und Vorschriften entspricht.
- AWS Security Hub— DieseAWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb vonAWSSie können Ihre Compliance mit Standards und bewährten Methoden der Branche in Bezug auf die Sicherheit überprüfen.
- AWS Audit Manager— DieseAWShilft Ihnen, IhreAWSverwenden, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Ausfallsicherheit im App Runner

DieAWSIm Mittelpunkt der globalen -Infrastruktur stehenAWS-Regionen-Zones.AWS-Regionen-Zones stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zuAWS-RegionenWeitere Informationen finden Sie unterAWSGlobale Infrastruktur.

AWS App Runner verwaltet und automatisiert die Nutzung der globalen AWS-Infrastruktur in Ihrem Namen. Bei der Verwendung von App Runner profitieren Sie von den Verfügbarkeits- und Fehlertoleranzmechanismen, dieAWS-Angebote.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Infrastruktursicherheit in AWS App Runner

Als verwalteter ServiceAWS App Runnerwird durch dieAWS-Verfahren zur Gewährleistung der Netzwerksicherheit von, die im AbschnittAmazon Web Services: Übersicht über Sicherheitsprozesse-Whitepaper.

Sie verwendenAWSSie verwenden von veröffentlichte API-Aufrufe, um über das Netzwerk auf zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.0 oder höher unterstützen. Wir empfehlen TLS 1.2 oder höher. Clients müssen außerdem Cipher Suites mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Weitere -Sicherheitsthemen von App Runner finden Sie unterSicherheit.

Konfigurations- und Schwachstellenanalyse in App Runner

AWS und unseren Kunden haben eine gemeinsame Verantwortung, um ein hohes Maß an Sicherheit und Compliance für Softwarekomponenten zu erzielen. Weitere Informationen finden Sie unter AWS Modell der übergreifenden Verantwortlichkeit.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Bewährte Sicherheitsmethoden für App Runner

AWS App Runner bietet mehrere Sicherheitsfunktionen, die Sie beim Entwickeln und Implementieren Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Weitere Themen zur App Runner-Sicherheit finden Sie unterSicherheit.

Bewährte Methoden für vorbeugende Sicherheitsmaßnahmen

Vorbeugende Sicherheitskontrollen versuchen, Vorfälle zu verhindern, bevor sie auftreten.

Implementieren der geringstmöglichen Zugriffsrechte

App Runner bietetAWS Identity and Access ManagementVerwaltete IAM-Richtlinien fürIAM-Benutzerund dieRolle für den Zugriff. Diese verwalteten Richtlinien geben alle Berechtigungen an, die möglicherweise für den korrekten Betrieb Ihres App Runner-Service erforderlich sind.

Ihre Anwendung benötigt möglicherweise nicht alle Berechtigungen in unseren verwalteten Richtlinien. Sie können sie anpassen und nur die Berechtigungen erteilen, die für die Benutzer und den App Runner-Service zum Ausführen ihrer Aufgaben erforderlich sind. Dies ist besonders relevant für Benutzerrichtlinien, in denen unterschiedliche Benutzerrollen möglicherweise unterschiedliche Berechtigungsanforderungen haben. Die Implementierung der geringstmöglichen Zugriffsrechte ist eine grundlegende Voraussetzung zum Reduzieren des Sicherheitsrisikos und der Auswirkungen, die aufgrund von Fehlern oder böswilligen Absichten entstehen könnten.

Bewährte Methoden für aufdeckende Sicherheitsmaßnahmen

Aufdeckende Sicherheitskontrollen identifizieren Sicherheitsverstöße, nachdem sie aufgetreten sind. Sie können Ihnen dabei helfen, potenzielle Sicherheitsbedrohungen oder -vorfälle zu erkennen.

Implementieren der Überwachung

Überwachung ist wichtig, um die Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer App Runner Lösungen aufrechtzuerhalten.AWSSie bietet verschiedene Tools und Services, die Sie bei der Überwachung IhrerAWS-Services.

Es folgen einige Beispiele für zu überwachende Elemente:

- Amazon CloudWatch Metriken für App Runner— Legen Sie Alarme für wichtige App Runner-Metriken und für die benutzerdefinierten Metriken Ihrer Anwendung fest. Details dazu finden Sie unter Metriken (CloudWatch).
- AWS CloudTrailEinträge-Aktionen, die sich auf die Verfügbarkeit auswirken können, wie z. B.`PauseService`oder .`DeleteConnection`. Details dazu finden Sie unter API-Aktionen (CloudTrail).

AWS-Glossar

Listet die neuesteAWS-Terminologie finden Sie imAWS-GlossarimAWS– Allgemeine Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.