Manuel du développeur

AWS App Runner

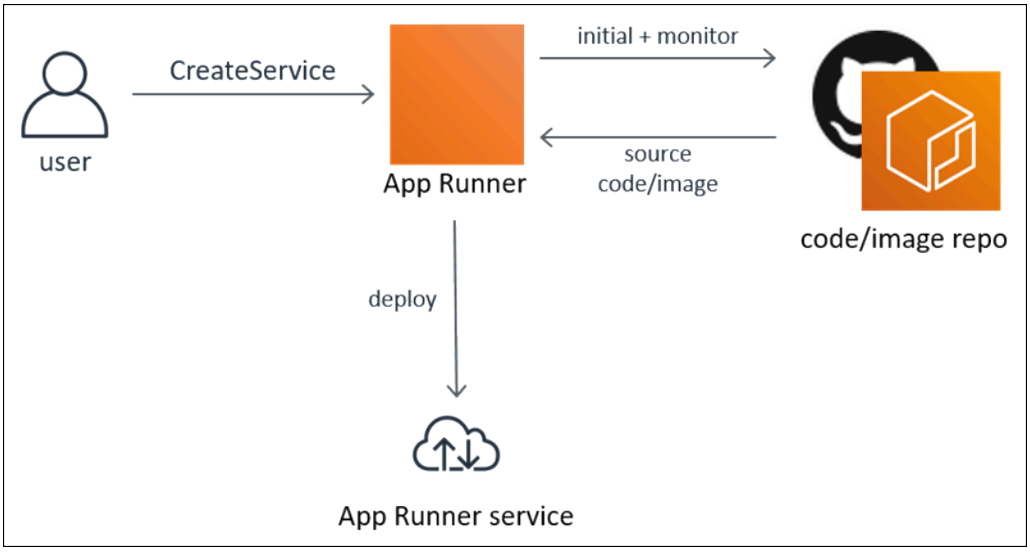
AWS App Runner: Manuel du développeur

Table of Contents

Présentation d'AWS App Runner

AWS App Runnerest unAWSqui fournit un moyen rapide, simple et économique de déployer à partir du code source ou d'une image de conteneur directement vers une application Web évolutive et sécurisée dans leAWSCloud. Vous n'avez pas besoin d'apprendre les nouvelles technologies, de décider du service de calcul à utiliser ou de savoir comment provisionner et configurerAWSAWS.

App Runner se connecte directement à votre référentiel de code ou d'images. Il fournit un pipeline d'intégration et de livraison automatique avec des opérations entièrement gérées, des performances élevées, une évolutivité et une sécurité.



Pour qui est App Runner ?

Si vous êtesdéveloppeur, vous pouvez utiliser App Runner pour simplifier le processus de déploiement d'une nouvelle version de votre référentiel de code ou d'images.

Pouréquipes d'opérations, App Runner permet des déploiements automatiques chaque fois qu'un commit est envoyé dans le référentiel de code ou qu'une nouvelle version d'image de conteneur est envoyée dans le référentiel d'images.

Accès au Exécuteur d'applications

Vous pouvez définir et configurer vos déploiements de service App Runner à l'aide de l'une des interfaces suivantes :

- App Runner— Fournit une interface Web pour gérer vos services App Runner.
- API Exécuteur de applications— Fournit une API RESTful pour effectuer des actions App Runner. Pour de plus amples informations, veuillez consulter Référence d'API AWS App Runner.
- AWSInterface de ligne de commande (AWS CLI)— Fournit des commandes pour une large gamme deAWS, notamment Amazon VPC, et est prise en charge par Windows, macOS et Linux/UNIX. Pour plus d'informations, consultez AWS Command Line Interface.
- AWSKits SDK— fournissent des API spécifiques au langage et se chargent de nombreux détails de connexion, tels que le calcul des signatures, la gestion des nouvelles tentatives de demande et le traitement des erreurs. Pour plus d'informations, consultez Kits SDK AWS.

Tarifs pour App Runner

App Runner offre un moyen économique d'exécuter votre application. Vous ne payez que pour les ressources que votre service App Runner consomme. Votre service réduit à moins d'instances de calcul lorsque le trafic des demandes est plus lent. Vous contrôlez les paramètres d'évolutivité : le nombre le plus bas et le plus élevé d'instances provisionnées et la charge la plus élevée qu'une instance gère.

Pour plus d'informations sur le dimensionnement automatique App Runner, consultezthe section called “Mise à l'échelle automatique”.

Pour en savoir plus sur la tarification, consultez Tarification AWS App Runner.

Quelle est la prochaine étape ?

Découvrez comment commencer à utiliser App Runner dans les rubriques suivantes :

- Configuration de— Effectuez les étapes préalables à l'utilisation d'App Runner.
- Mise en route— Déployez votre première application sur App Runner.

Configuration de App Runner

Si vous êtes un nouveauAWS, remplissez les conditions préalables d'installation répertoriées sur cette page avant de commencer à utiliserAWS App Runner.

Pour ces procédures d'installation, vous utilisez leAWS Identity and Access Management(IAM). Pour obtenir des informations complètes sur IAM, consultez les ressources de référence suivantes :

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

Créer un Compte AWS

Lorsque vous vous inscrivez avecAWS, vous obtenez un numéro de compte avec accès à tous les services queAWSOffres, y comprisAWS App Runner.

Si vous avez déjà unCompte AWSPassez ensuite à la condition préalable suivante.

Si vous n'avez pas deAWSProcédez de la manière suivante pour en créer un.

Pour créer un compte AWS

1. Ouvrez https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

Création d'un utilisateur IAM

Pour accéder à unAWS, vous fournissez les informations d'identification. Ces informations d'identification déterminentauthentification(qui vous êtes) etauthorization(quelles autorisations vous avez pour effectuer des actions surAWSRessources).

Lorsque vous créez pour la première fois un Amazon Web Services (AWS), vous commencez avec une seule identité de connexion. Cette identité a un accès complet à tous les services et ressources AWS du compte. Cette identité est appelée laAWScompteUtilisateur racine. Lorsque vous vous connectez, saisissez l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte.

> **Important**
>
> Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes, y compris pour les tâches administratives. Respectez plutôt la bonne pratique qui consiste à avoir recours à l'utilisateur racine uniquement pour créer le premier utilisateur IAM. Ensuite, mettez en sécurité les informations d'identification de l'utilisateur racine et utilisez-les uniquement pour effectuer certaines tâches de gestion des comptes et des services. Pour afficher les tâches qui nécessitent que vous vous connectiez en tant qu'utilisateur racine, consultezTâches qui nécessitent des informations d'identification utilisateur racine.

Pour plus d'informations sur les informations d'identification de l'utilisateur racine et de l'utilisateur IAM, consultezCompte AWSinformations d'identification utilisateur racine et informations d'identification de l'utilisateur IAMdans leAWSRéférence générale.

Pour créer un administrateur pour vous-même et ajouter l'utilisateur à un groupe d'administrateurs (console)

1. Connectez-vous à la consoleConsole IAMen tant que propriétaire du compte en choisissantUtilisateur racineet en saisissant votreAWSAdresse e-mail du compte Sur la page suivante, saisissez votre mot de passe.

   > **Note**
   >
   > Nous vous recommandons vivement de respecter la bonne pratique qui consiste à avoir recours à la**`Administrator`**Utilisateur IAM qui suit et verrouille en sécurité les informations d'identification de l'utilisateur racine. Connectez-vous en tant qu'utilisateur racine pour effectuer certaines tâches de gestion des comptes et des services.

2. Dans le panneau de navigation, choisissez Utilisateurs, puis Add user (Ajouter un utilisateur).
3. Dans User name (Nom d'utilisateur), entrez **`Administrator`**.
4. Activez la case à cocher près deAWS Management ConsoleAccès via. Puis, sélectionnez Mot de passe personnalisé, et saisissez votre nouveau mot de passe dans la zone de texte.
5. (Facultatif) Par défaut,AWSLe nouvel utilisateur doit créer un nouveau mot de passe lors de sa première connexion. Décochez la case en regard de User must create a new password at

next sign-in (L'utilisateur doit créer un nouveau mot de passe à sa prochaine connexion) pour autoriser le nouvel utilisateur à réinitialiser son mot de passe une fois qu'il s'est connecté.

6. Choisissez Next (Suivant) Permissions (Autorisations).
7. Sous Set permissions (Accorder des autorisations), choisissez Add user to group (Ajouter un utilisateur au groupe).
8. Choisissez Create group.
9. Dans la boîte de dialogue Create group (Créer un groupe), pour Group name (Nom du groupe), tapez **`Administrators`**.
10. ChoisissezStratégies de filtre, puis sélectionnezAWSmanagé - fonction de travailpour filtrer le contenu de la table.
11. Dans la liste des stratégies, cochez la case AdministratorAccess. Choisissez ensuite Create group.

> **Note**
>
> Vous devez activer l'accès de l'utilisateur et du rôle IAM à la facturation avant de pouvoir utiliser le`AdministratorAccess`Autorisations pour accéder à la consoleAWS Billing and Cost Managementconsole Pour ce faire, suivez les instructions de l'étape 1 du didacticiel portant sur comment déléguer l'accès à la console de facturation.

12. De retour dans la liste des groupes, activez la case à cocher du nouveau groupe. Choisissez Refresh si nécessaire pour afficher le groupe dans la liste.
13. Choisissez Next (Suivant) Tags (Balises).
14. (Facultatif) Ajoutez des métadonnées à l'utilisateur en associant les balises sous forme de paires clé-valeur. Pour de plus amples informations sur l'utilisation des balises dans IAM, veuillez consulter Balisage des utilisateurs et des rôles IAM dans le Guide de l'utilisateur IAM.
15. Choisissez Next (Suivant) VérificationPour afficher la liste des appartenances de groupe à ajouter au nouvel utilisateur. Une fois que vous êtes prêt à continuer, choisissez Create user.

Vous pouvez utiliser ce même processus pour créer d'autres groupes et utilisateurs et pour accorder à vos utilisateurs l'accès aux ressources de votre compte AWS. Pour en savoir plus sur l'utilisation des stratégies qui limitent les autorisations utilisateur à desAWS, consultezGestion de l'accèsandExemples de stratégies.

> **Important**
>
> Protéger votreCompte AWS. N'envoyez ni partagez vos informations d'identification avec des personnes extérieures à votre organisation. Aucun véritable représentant d'Amazon ne vous demandera jamais de lui fournir vos informations d'identification.

Une fois votre utilisateur IAM créé, utilisez ses informations d'identification pour vous connecter à la consoleAWS Management Console. Pour de plus amples informations, veuillez consulterComment les utilisateurs IAM se connectent à votre compteCompte AWSdans leGuide de l'utilisateur IAM.

Créer une clé d'accès pour votre utilisateur IAM

Les clés d'accès se composent d'un ID de clé d'accès et d'une clé d'accès secrète, qui permettent de signer des demandes par programmation que vous envoyez àAWS. Si vous n'avez pas de clés d'accès, vous pouvez les créer à partir de la consoleAWS Management Console. À titre de bonne pratique, n'utilisez pas laAWSLes clés d'accès utilisateur racine d'un compte pour toute tâche où elle n'est pas requise. C'est laCréer un utilisateur IAM administrateuravec des clés d'accès pour vous-même.

La seule fois où vous pouvez afficher ou télécharger la clé d'accès secrète est lorsque vous créez les clés. Vous ne pourrez pas les récupérer plus tard. Toutefois, vous pouvez créer de nouvelles clés d'accès à tout moment. Vous devez également disposer des autorisations permettant d'effectuer les opérations IAM nécessaires. Pour de plus amples informations, veuillez consulterAutorisations requises pour accéder aux ressources IAMdans leGuide de l'utilisateur IAM.

Pour créer des clés d'accès pour un utilisateur IAM

1. Connectez-vous à la AWS Management Console et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, sélectionnez Users.
3. Choisissez le nom utilisateur dont vous souhaitez créer les clés d'accès, puis choisissez l'optionInformations d'identification SécuritéOnglet.
4. DansClés d'accès, choisissezCréer une clé d'accès.
5. Pour afficher la nouvelle key pair d'accès, choisissezAfficher. Vous ne pourrez pas accéder à la clé d'accès secrète à nouveau une fois que cette boîte de dialogue se sera fermée. Vos informations d'identification s'afficheront comme suit :

- ID de clé d'accès : AKIAIOSFODNN7EXAMPLE
- Clé d'accès secrète : wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

6. Pour télécharger la paire de clés, choisissez Télécharger le fichier .csv. Stockez les clés dans un emplacement sûr. Vous ne pourrez pas accéder à la clé d'accès secrète à nouveau une fois que cette boîte de dialogue se sera fermée.

   Préservez la confidentialité des clés afin de protéger votre compteAWSet ne les envoyez jamais par e-mail. Ne les divulguez à personne hors de votre entreprise, même si vous recevez une demande qui semble provenir d'AWS ou d'Amazon.com. Aucun véritable représentant d'Amazon ne vous demandera jamais de lui fournir votre clé secrète.

7. Après avoir téléchargé le `.csv`fichier, choisissezFermer. Lorsque vous créez une clé d'accès, la paire de clés est activée par défaut et vous pouvez utiliser la paire immédiatement.

Voir aussi

- En quoi consiste IAM ?dans leGuide de l'utilisateur IAM
- AWSInformations d'identification de sécuritéinAWSRéférence générale

Quelle est la prochaine étape ?

Vous avez terminé les étapes préalables. Pour déployer votre première application sur App Runner, consultezMise en route.

Mise en route d'd'App Runner

AWS App Runnerest unAWSqui fournit un moyen rapide, simple et économique de transformer une image de conteneur existante ou un code source directement en un service Web en cours d'exécution dans leAWS Cloud.

Ce tutoriel traite de la façon dont vous pouvez utiliserAWS App Runnerpour déployer votre application sur un service App Runner. Il traite de la configuration du code source et du déploiement, de la génération du service et de l'exécution du service. Il montre également comment déployer une version de code, modifier la configuration et afficher les journaux. Enfin, le didacticiel montre comment nettoyer les ressources que vous avez créées tout en suivant les procédures du didacticiel.

Rubriques

Prerequisites

Avant de lancer le didacticiel, assurez-vous d'effectuer les opérations suivantes :

1. Suivez les étapes de configuration dansConfiguration de.
2. Création d'unGitHubSi vous n'en avez pas déjà un. Si vous débutez avec GitHub, consultezMise en route d'GitHubdans leDocs GitHub.
3. Création d'un référentiel dans votre compte GitHub. Ce didacticiel utilise le nom du référentiel`python-hello`. Créez des fichiers dans le répertoire racine du référentiel, avec les noms et le contenu spécifiés dans les exemples suivants.

Fichiers pour le**python-hello**Exemple de référentiel

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)
```

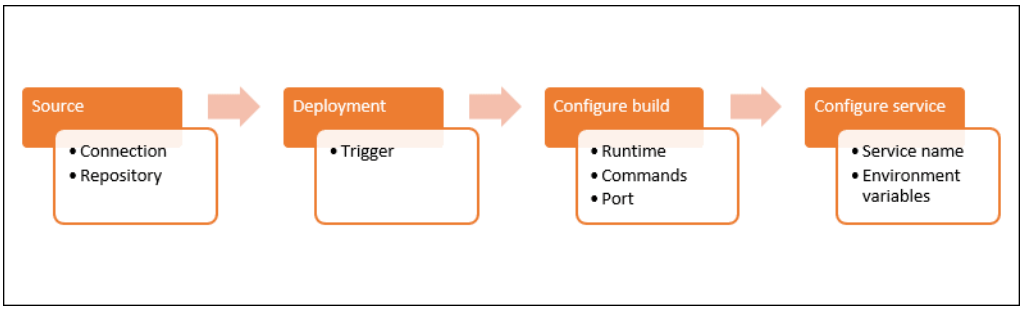
Étape 1 : Création d'un service App Runner

Dans cette étape, vous créez un service App Runner basé sur l'exemple de référentiel de code source que vous avez créé sur GitHub dans le cadre dethe section called “Prerequisites”. L'exemple

contient un site Web Python simple. Voici les principales étapes que vous effectuez pour créer un service :

1. Configurez votre code source.
2. Configurer le déploiement source.
3. Configurer la génération d'applications.
4. Configurez votre service.
5. Examinez et confirmez.

Le schéma suivant décrit les étapes de création d'un service App Runner :

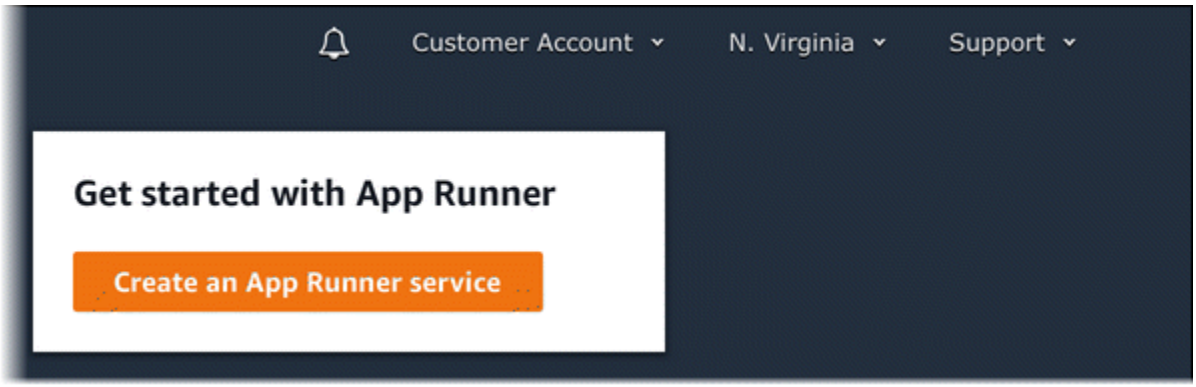


Pour créer un service App Runner basé sur un référentiel de code source

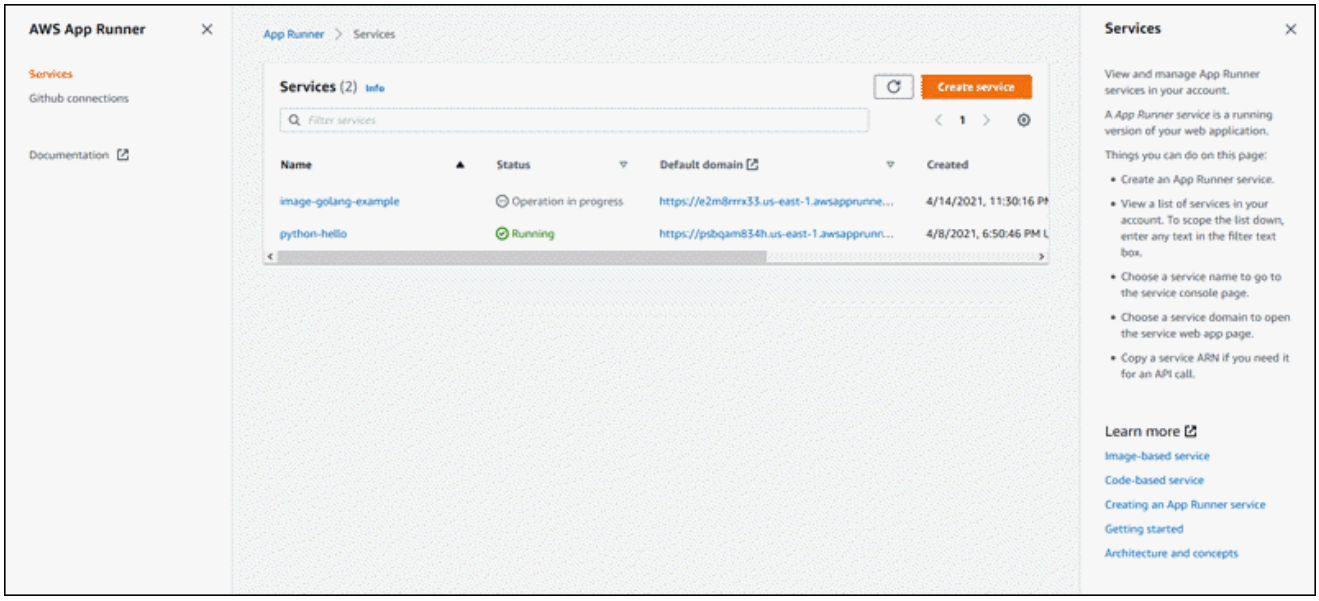
1. Configurez votre code source.

   a. Ouverture d'Console App Runner, et dans leRégions, sélectionnez votreRégion AWS.

   b. Si l'icôneCompte AWSn'a pas encore de services App Runner, la page d'accueil de la console s'affiche. ChoisissezCréation d'un service App Runner.



   Si l'icôneCompte AWSdispose de services existants, leServicesLa page comprenant une liste de vos services s'affiche. Choisissez Créer un service.



c. Dans la pageSource et déploiement, dans laSourceSection, pourType de référentiel, choisissezRéférentiel de code source.

d. UNDERConnect au GitHubchoisirAjouter un nouveau, puis, si vous y êtes invité, saisissez vos informations d'identification GitHub.

> **Note**
>
> Les étapes suivantes pour installer le kitAWSConnecteur pour GitHub à votre compte GitHub sont des étapes uniques. Vous pouvez réutiliser la connexion pour créer plusieurs services App Runner basés sur les référentiels de ce compte. Lorsque vous disposez d'une connexion existante, sélectionnez-la et passez à la sélection du référentiel.

e. DansInstallAWSConnecteur pour GitHub, si vous y êtes invité, choisissez le nom de votre compte GitHub.

f. Si vous êtes invité à autoriser leAWSConnecteur pour GitHub, choisissezAutoriser les connexions AWS.

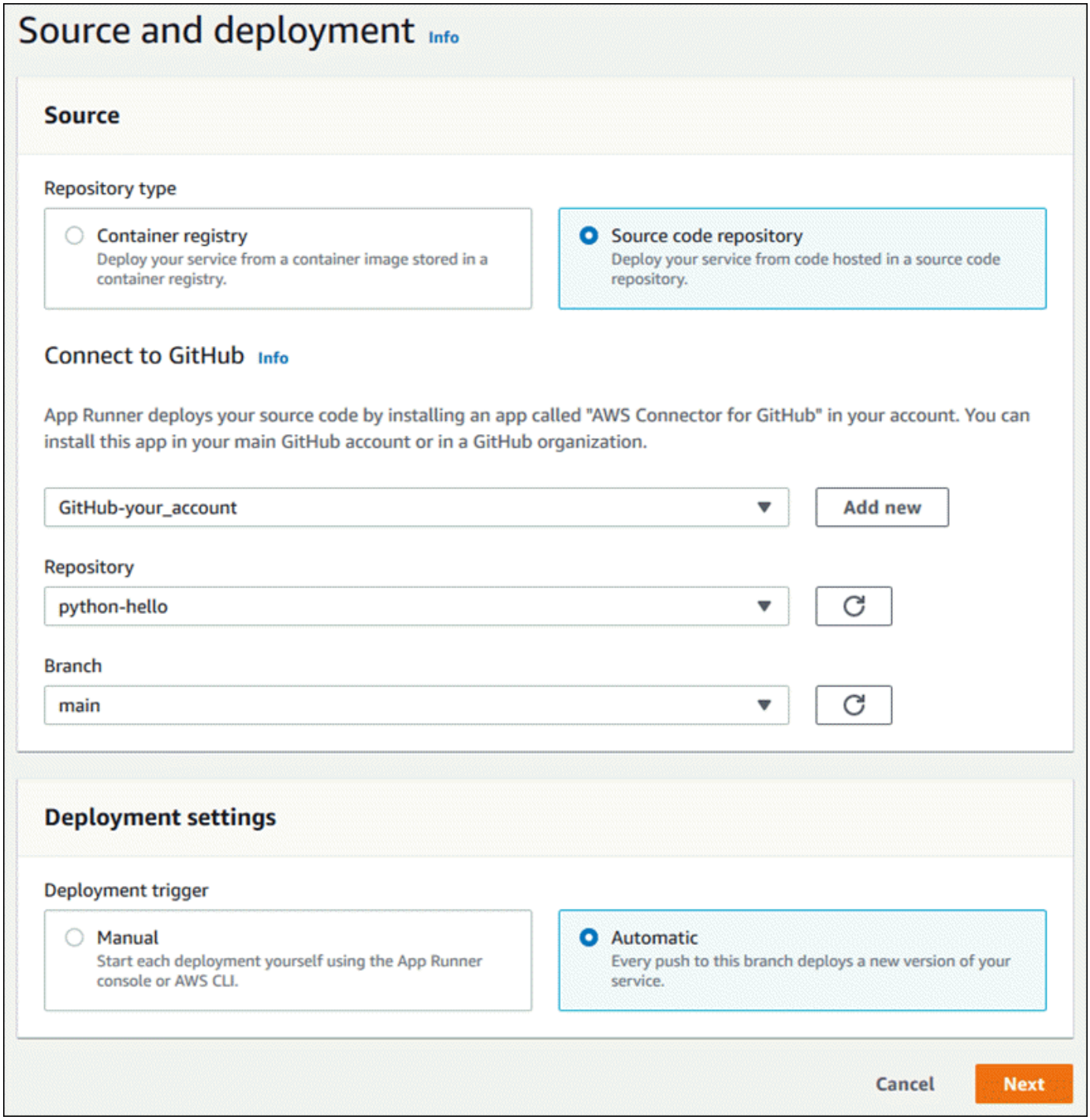
g. Choisissez Installer.

Le nom de votre compte apparaît en tant queCompte/organisation GitHub. Vous pouvez désormais choisir un référentiel dans votre compte.

h. PourRéférentiel., choisissez l'exemple de référentiel que vous avez créé,`python-hello`. PourBranche, choisissez le nom de branche par défaut de votre référentiel (par exemple,Principal).

2. Configurez vos déploiements : DansParamètres de déploiementSection, choisissezAutomatic, puisSuivant.

> **Note**
>
> Avec le déploiement automatique, chaque nouvelle validation dans votre référentiel déploie automatiquement une nouvelle version de votre service.



3. Configurer la génération d'applications.

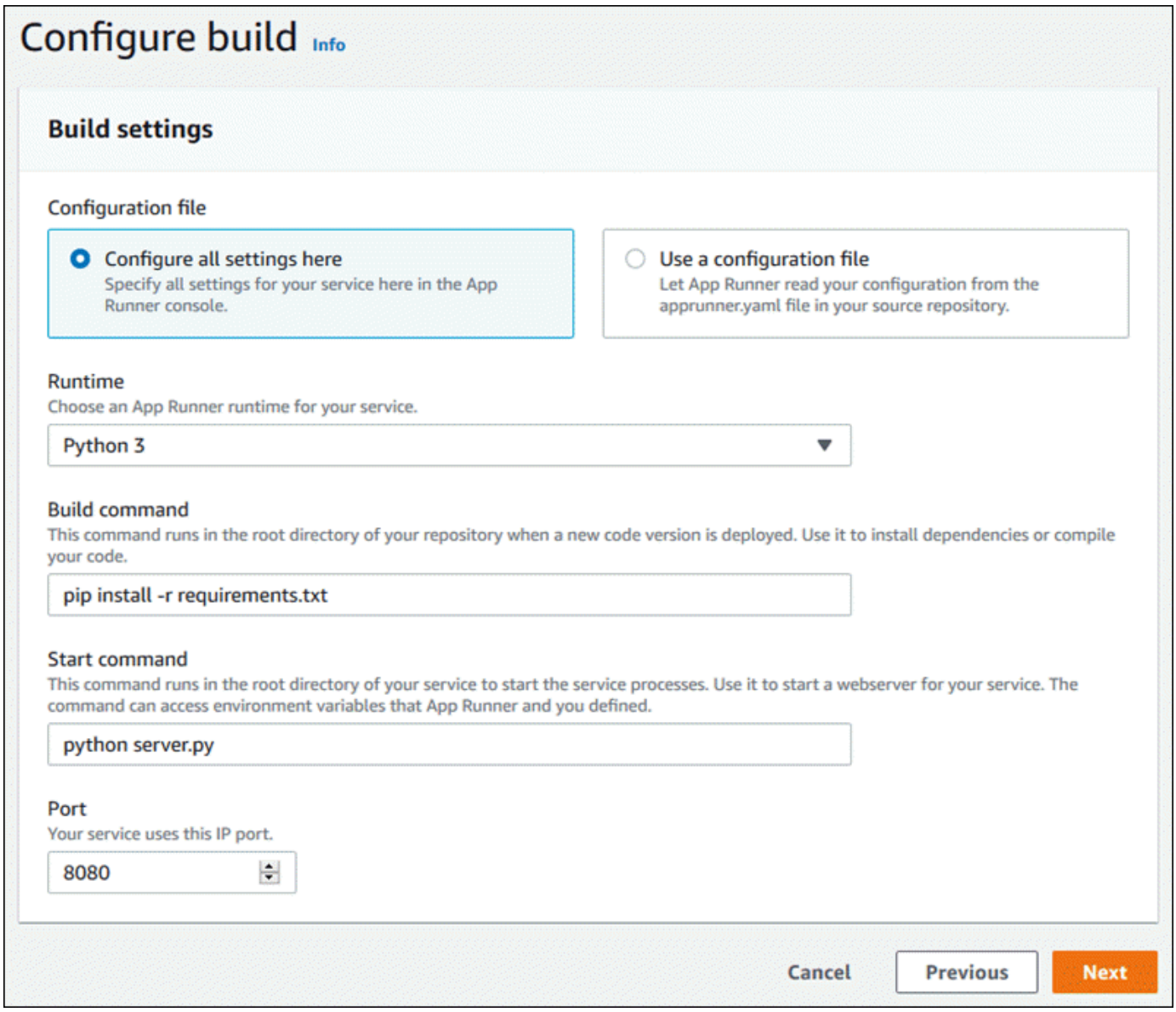
   a. Dans la pageConfigurer la génération, pourFichier de configuration, choisissezConfigurez tous les paramètres ici.

   b. Fournissez les paramètres de génération suivants :

- Runtime (Exécution)— ChoisissezPython 3.
- Commande de build— Saisir**`pip install -r requirements.txt`**.
- Commande de démarrage— Saisir**`python server.py`**.
- Port— Saisir**`8080`**.

c. Choisissez Next (Suivant).

> **Note**
>
> Le moteur d'exécution Python 3 construit une image Docker en utilisant une image Python 3 de base et votre exemple de code Python. Il lance ensuite un service qui exécute une instance de conteneur de cette image.



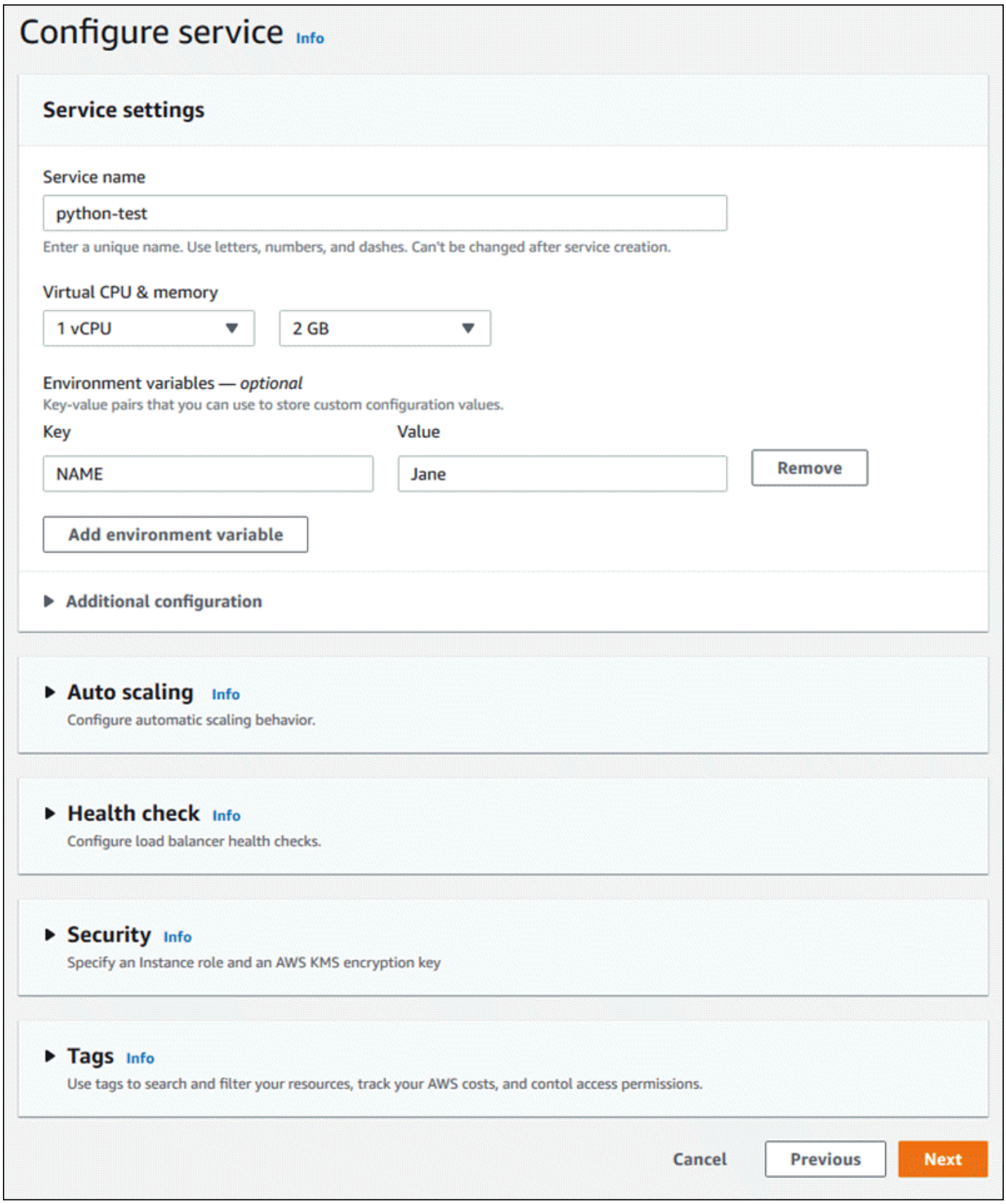
4. Configurez votre service.

   a. Dans la pageConfigurer le service, dans laParamètres de service, entrez un nom de service.

   b. UNDEREnvironment variables (Variables d'environnement), ajoutez une seule variable d'environnement. PourClé, saisissez**`NAME`**, et pourValeur, saisissez n'importe quel nom (par exemple, votre prénom).

> **Note**
>
> L'exemple d'application lit le nom que vous avez défini dans cette variable d'environnement et affiche le nom sur sa page Web.

c. Choisissez Next (Suivant).


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. Dans la pageVérifier et créer, vérifiez tous les détails que vous avez entrés, puis choisissezCréation et déploiement.

   Si le service est créé avec succès, la console affiche le tableau de bord du service, avec unPrésentation du servicedu nouveau service.

6. Vérifiez que votre service est en cours d'exécution.

   a. Sur la page du tableau de bord du service, attendez que le serviceÉtatestEn cours d'exécution.

   b. Cliquez sur l'ongletDomaine par défaut : il s'agit de l'URL du site Web de votre service.

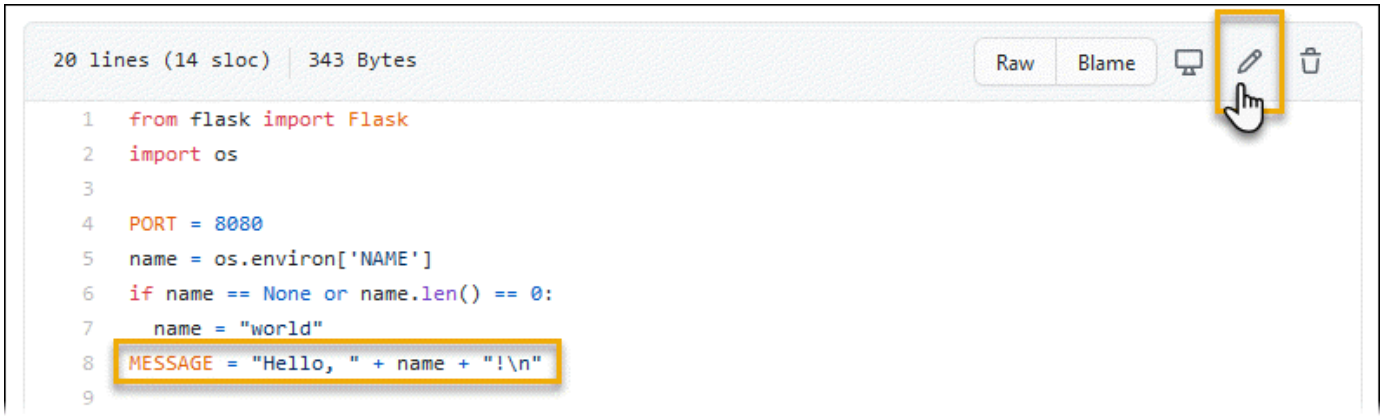
      Une page Web affiche : Bonjour,***`Votre nom`*** !

Étape 2 : Modifier votre code de service

Au cours de cette étape, vous modifiez votre référentiel de code source. La fonctionnalité CI/CD App Runner crée et déploie automatiquement la modification sur votre service.

Pour modifier votre code de service

1. Accédez à votre exemple de référentiel GitHub.
2. Choisissez le nom de fichier`server.py`pour accéder à ce fichier.
3. ChoisissezModifier ce fichier(icône de crayon).
4. Dans l'expression affectée à la variable`MESSAGE`, modifiez le texte`Hello`sur`Good morning`.



5. Choisissez Valider les modifications.

   Le nouveau commit commence à se déployer. Sur la page du tableau de bord du service, le serviceÉtatModifications apportées àOpération en cours.

6. Attendez que le déploiement soit terminé. Sur la page du tableau de bord du service, le serviceÉtatdevrait revenir àEn cours d'exécution.

7. Vérifiez que le déploiement a réussi : actualisez l'onglet du navigateur où la page Web de votre service est affichée.

   La page affiche désormais le message modifié : Bonjour,***Votre nom*** !

Étape 3 : Réalisation d'une modification de la configuration

Au cours de cette étape, vous modifiez le kit**NAME**valeur de la variable d'environnement, pour démontrer un changement de configuration de service.

Pour afficher les journaux de votre service

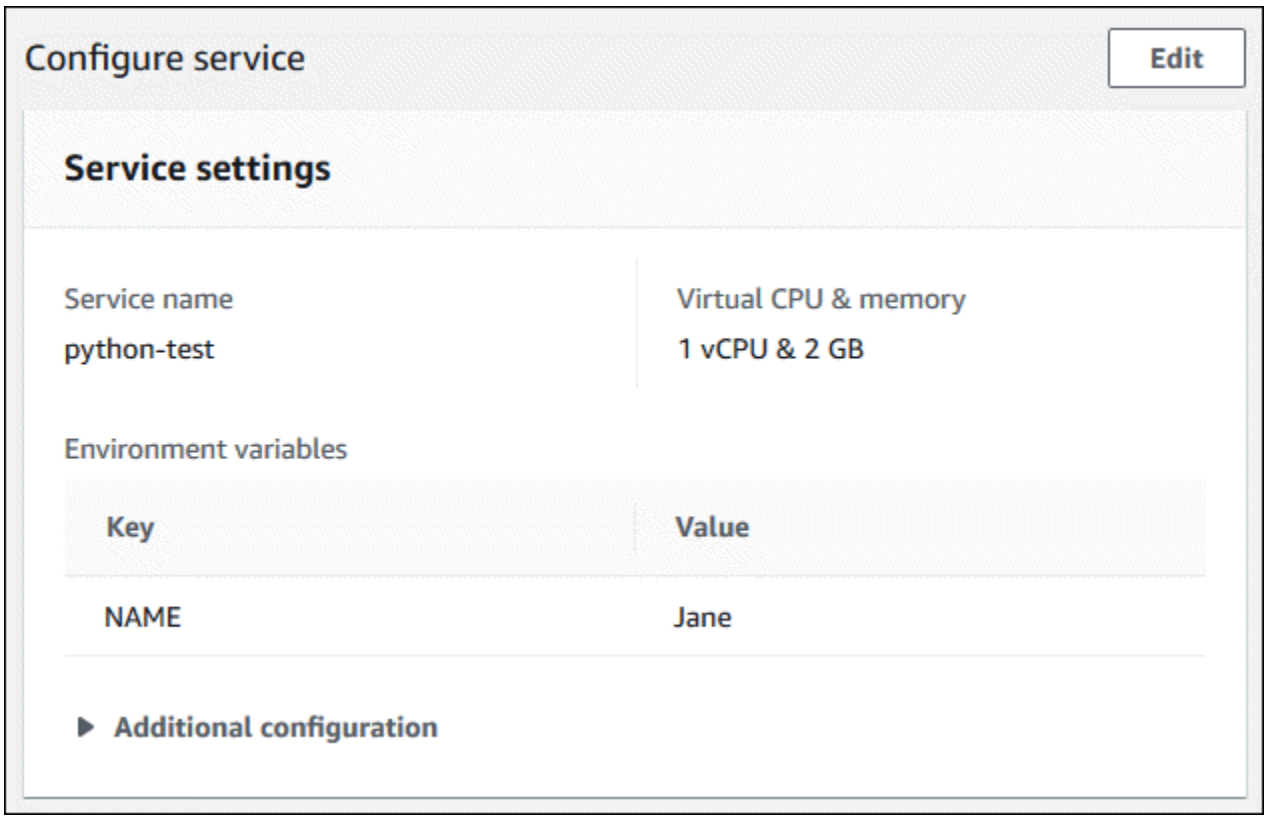
1. Ouverture d'Console App Runner, et dans leRégions, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.

3. Sur la page du tableau de bord du service, choisissez le kitConfigurationOnglet.

   La console affiche les paramètres de configuration de votre service dans plusieurs sections.

4. DansConfigurer le serviceSection, choisissezModifier.



5. Pour la variable d'environnement avec la clé**NAME**, remplacez la valeur par un nom différent.
6. Choisissez Apply changes.

   App Runner lance le processus de mise à jour. Sur la page du tableau de bord du service, le serviceÉtatModifications apportées àOpération en cours.
7. Attendez que la mise à jour soit terminée. Sur la page du tableau de bord du service, le serviceÉtatdevrait revenir àEn cours d'exécution.
8. Vérifiez que la mise à jour est réussie : actualisez l'onglet du navigateur où la page Web de votre service est affichée.

   La page affiche désormais le nom modifié : Bonjour,***Nouveau nom*** !

Étape 4 : Afficher les journaux de votre service

Dans cette étape, vous utilisez la console App Runner pour afficher les journaux de votre service App Runner. App Runner diffuse les journaux sur Amazon CloudWatch Logs (CloudWatch Logs) et les affiche sur le tableau de bord de votre service. Pour plus d'informations sur les journaux App Runner, consultezthe section called “Journaux (journaux CloudWatch Logs)”.

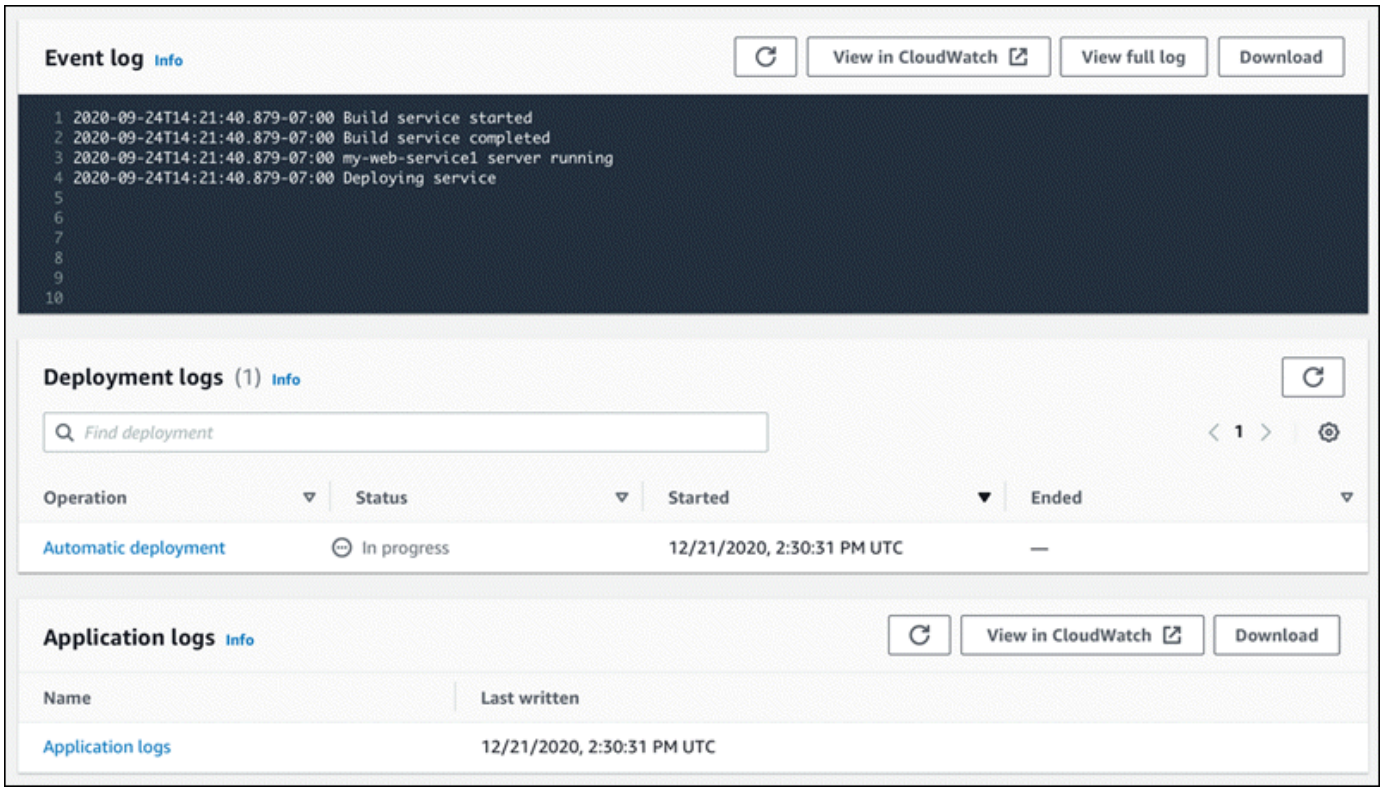
Pour afficher les journaux de votre service

1. Ouverture d'Console App Runner, et dans leRégions, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. Sur la page du tableau de bord du service, choisissez le kitJournauxOnglet.

   La console affiche quelques types de journaux dans plusieurs sections :

   - Event Log (Journal des événements)— Activité dans le cycle de vie de votre service App Runner. La console affiche les événements les plus récents.
   - Journaux de déploiement— Déploiements de référentiel source vers votre service App Runner ; La console affiche un flux de journal distinct pour chaque déploiement.
   - Journaux d'application— Sortie de l'application Web déployée sur votre service App Runner. La console combine la sortie de toutes les instances en cours d'exécution dans un flux de journaux unique.



4. Pour rechercher des déploiements spécifiques, étendez la liste du journal de déploiement en entrant un terme de recherche. Vous pouvez rechercher n'importe quelle valeur apparaissant dans le tableau.
5. Pour afficher le contenu d'un journal, choisissezAfficher le journal complet(journal des événements) ou le nom du flux de journal (journaux de déploiement et d'application).
6. ChoisissezTéléchargementpour télécharger un journal. Pour un flux de journal de déploiement, sélectionnez d'abord un flux de journaux.
7. ChoisissezAfficher dans CloudWatchpour ouvrir la console CloudWatch et utiliser toutes ses fonctionnalités pour explorer vos journaux de service App Runner. Pour un flux de journal de déploiement, sélectionnez d'abord un flux de journaux.

> **Note**
>
> La console CloudWatch est particulièrement utile si vous souhaitez afficher les journaux d'applications d'instances spécifiques au lieu du journal d'applications combiné.

Étape 5 : Nettoyage

Vous avez maintenant appris à créer un service App Runner, à afficher les journaux et à apporter quelques modifications. Au cours de cette étape, vous supprimez le service pour supprimer les ressources dont vous n'avez plus besoin.

Pour supprimer votre service

1. Sur la page du tableau de bord des services, choisissezActions, puisSuppression de service.
2. Dans la boîte de dialogue de confirmation, entrez le texte demandé, puis choisissezSupprimer.

   Résultat: La console accède à laServices. Le service que vous venez de supprimer affiche un état deDELETING. Peu de temps plus tard, il disparaît de la liste.

Envisagez également de supprimer la connexion GitHub que vous avez créée dans le cadre de ce didacticiel. Pour plus d'informations, consultez the section called “Connexions”.

Quelle est la prochaine étape ?

Maintenant que vous avez déployé votre premier service App Runner, consultez les rubriques suivantes :

- Architecture et concepts relatifs à— L'architecture, les principaux concepts etAWSressources liées à App Runner.
- Service basé sur l'imageandService basé sur le code— Les deux types de source d'application que App Runner peut déployer.
- Développement pour App Runner— Ce que vous devez savoir lors du développement ou de la migration du code d'application pour le déploiement vers App Runner.
- Exécuteur d'applications— Gérez et surveillez votre service à l'aide de la console App Runner.
- Gestion de votre service— Gérez le cycle de vie de votre service App Runner.
- Journalisation et surveillanceSurveillez votre service App Runner en affichant les mesures, en lisant les journaux et en suivant les appels d'actions de service.
- Fichier de configuration App Runner : méthode basée sur la configuration pour spécifier des options pour le comportement de génération et d'exécution de votre service App Runner.
- API Exécuteur d'applications— Utilisez l'interface de programmation d'applications (API) App Runner pour créer, lire, mettre à jour et supprimer des ressources App Runner.
- Sécurité— Les différentes façons dontAWSet vous assurez la sécurité du cloud pendant que vous utilisez App Runner et d'autres services.

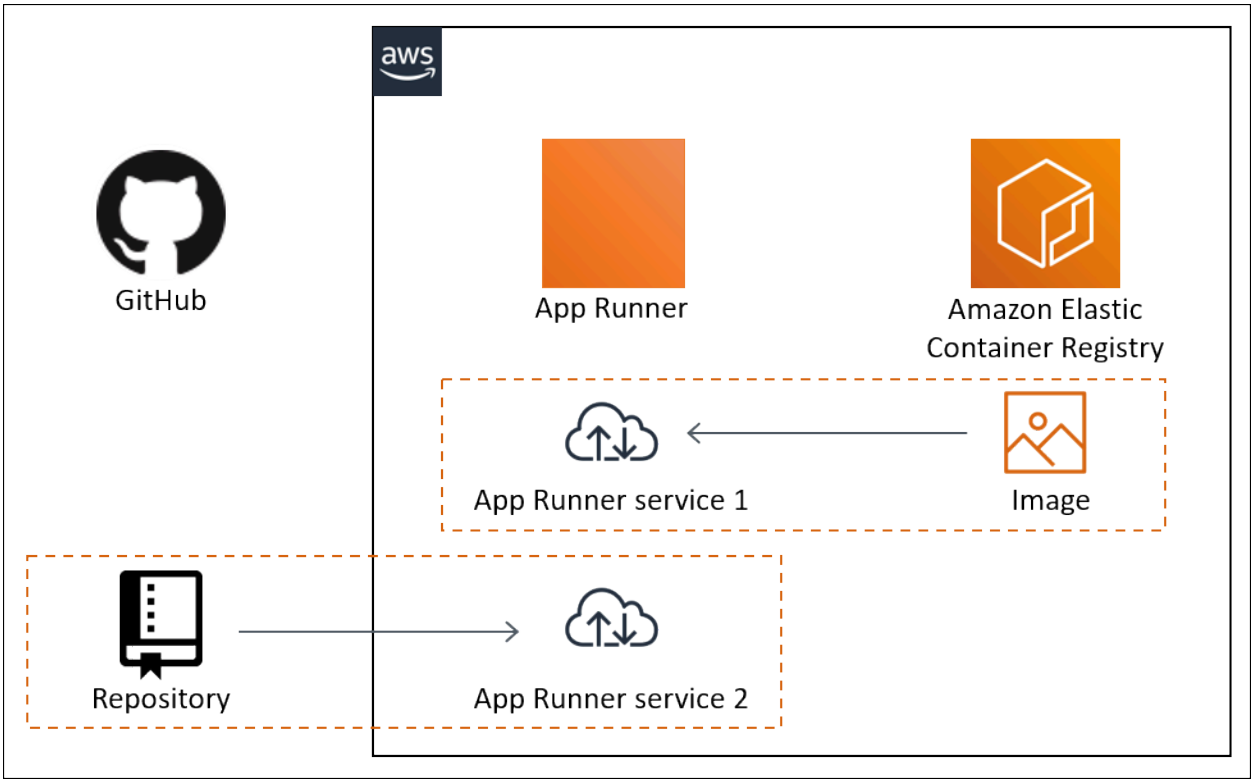
Architecture et concepts d'App Runner

AWS App Runnerprend votre code source ou votre image source à partir d'un référentiel, et crée et gère un service Web en cours d'exécution pour vous dans laAWS Cloud. En règle générale, vous devez appeler une seule action App Runner,CreateService, pour créer votre service.

Avec un référentiel d'images source, vous fournissez une image de conteneur prête à l'emploi que App Runner peut déployer pour exécuter votre service Web. Avec un référentiel de code source, vous pouvez fournir du code et des instructions pour la création et l'exécution d'un service Web conçu pour l'un des nombreux environnements d'exécution gérés par App Runner.

Pour le moment, App Runner peut récupérer votre code source à partir d'unGitHub, ou récupérez votre image source à partir deAmazon Elastic Container Registry (Amazon ECR)dans votreCompte AWS.

Le diagramme suivant présente l'architecture de service App Runner. Dans le diagramme, il existe deux exemples de services : l'un déploie du code source à partir de GitHub et l'autre déploie une image source à partir d'Amazon ECR.



Concepts de l'application

Voici les concepts clés liés à votre service Web qui s'exécute dans App Runner :

- Exécuteur de l'application— UnAWSqu'App Runner utilise pour déployer et gérer votre application en fonction de son référentiel de code source ou de son image de conteneur. Un service App Runner est une version en cours d'exécution de votre application. Pour plus d'informations sur la création d'un service, consultezthe section called “Création”.
- Type de source— Type de référentiel source que vous fournissez pour le déploiement de votre service App Runner :Code sourceouimage source.
- Fournisseur de référentiel— Service de référentiel qui contient la source de votre application (par exemple,GitHubouAmazon ECR).
- Connexion App Runner— UnAWSqui permet à App Runner d'accéder à un compte fournisseur de référentiel (par exemple, un compte GitHub ou une organisation). Pour plus d'informations sur les connexions, consultez the section called “Connexions”.
- Runtime (Exécution)— Image de base pour le déploiement d'un référentiel de code source. App Runner fournit une variété deenvironnements d'exécution géréspour différents environnements de programmation. Pour plus d'informations, consultez Service basé sur le code.
- Déploiement— Action qui applique une version de votre référentiel source (code ou image) à un service App Runner. Le premier déploiement vers le service se produit dans le cadre de la création du service. Les déploiements ultérieurs peuvent se produire de l'une des deux manières suivantes :
  - Déploiements— Capacité CI/CD. Vous pouvez configurer un service App Runner pour qu'il génère automatiquement (pour le code source) et déploie chaque version de votre application telle qu'elle apparaît dans le référentiel. Il peut s'agir d'un nouveau commit dans un référentiel de code source ou d'une nouvelle version d'image dans un référentiel d'images source.
  - Déploiements— Déploiement vers votre service App Runner que vous démarrez explicitement.
- Nom de domaine personnalisé— Domaine que vous associez à votre service App Runner. Les utilisateurs de votre application Web peuvent utiliser ce domaine pour accéder à votre service Web au lieu du sous-domaine App Runner par défaut. Pour plus d'informations, consultez the section called “Noms de domaine personnalisés”.
- Maintenance— Activité qu'App Runner effectue occasionnellement sur l'infrastructure qui exécute votre service App Runner. Lorsque la maintenance est en cours, l'état du service passe temporairement à`OPERATION_IN_PROGRESS`(Opération en coursDans la console) pendant quelques minutes. Les actions sur votre service (par exemple, déploiement, mise à jour de

configuration, pause/reprise ou suppression) sont bloquées pendant cette période. Réessayez l'action quelques minutes plus tard, lorsque l'état du service revient à`RUNNING`.

> **Note**
>
> Si votre action échoue, cela ne signifie pas que votre service App Runner est en panne. Votre application est active et continue de traiter les demandes. Il est peu probable que votre service connaisse un temps d'arrêt.

En particulier, App Runner migre votre service s'il détecte des problèmes dans le matériel sous-jacent hébergeant le service. Pour éviter tout temps d'arrêt de service, App Runner déploie votre service sur un nouvel ensemble d'instances et déplace le trafic vers celles-ci (un déploiement bleu-vert). Vous pourriez occasionnellement constater une légère augmentation temporaire des frais.

Ressources App Exécuteur

Lorsque vous utilisez App Runner, vous créez et gérez quelques types de ressources dans votreCompte AWS. Ces ressources sont utilisées pour accéder à votre code et gérer vos services.

Le tableau suivant donne une vue d'ensemble de ces ressources :

Nom de la ressource	Description
Service	Représente une version en cours d'exécution de votre application. Une grande partie du reste de ce guide décrit les types de service, la gestion, la configuration et la surveillance. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:service/` *`service-name`* `[/`*`service-id`* `]`
Connection	Fournit à vos services App Runner un accès à des référentiels privés stockés auprès de fournisseurs tiers. Existe en tant que ressource distincte pour le partage entre plusieurs services. Pour plus d'informations sur les connexions, consultez the section called “Connexions”. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:connection/` *`connection-name`* `[/`*`connection-id`* `]`

Nom de la ressource	Description
AutoScalingConfiguration	Fournit à vos services App Runner des paramètres qui contrôlent la mise à l'échelle automatique de votre application. Existe en tant que ressource distincte pour le partage entre plusieurs services. Pour plus d'informations sur la mise à l'échelle automatique, consultezthe section called “Mise à l'échelle automatique”. ARN: `arn:aws:apprunner:` *`region`*`:`*`account-id`*` :autoscalingconfiguration/` *`config-name`* `[/`*`config-revision`* `[/`*`config-id`* `]]`

Quotas de ressources App Exécuteur

AWSimpose des quotas (également connus sous le nom de limites) sur votre compte pourAWSUtilisation des ressources dans chaqueRégion AWS. Le tableau suivant répertorie les quotas relatifs aux ressources App Exécuteur. Les quotas sont également répertoriés dansAWS App RunnerPoints de terminaison et quotasdans leAWSRéférence générale.

Continus de ressources	Description	Valeur par défaut	Ajustable ?
Services	Nombre maximal de services que vous pouvez créer dans votre compte pour chaqueRégion AWS.	10	✓ Oui
Connections	Nombre maximal de connexions que vous pouvez créer dans votre compte pour chaqueRégion AWS. Vous pouvez partager une connexion unique entre plusieurs services.	10	✓ Oui
Auto scaling configurations—noms	Le nombre maximal de noms uniques que vous pouvez avoir dans les configurations de mise à l'échelle automatique que vous créez dans votre compte pour chaqueRégion AWS. Vous	10	✓ Oui

Continus de ressources	Description	Valeur par défaut	Ajustable ?
	pouvez utiliser une configuration de mise à l'échelle automatique dans plusieurs services.		
Auto scaling configurations—révisions pour chaque nom	Nombre maximal de révisions de configuration automatique que vous pouvez créer dans votre compte pour chaqueRégion AWSPour chaque nom unique. Vous pouvez utiliser une révision de configuration de mise à l'échelle automatique dans plusieurs services.	10	✕ Non

La plupart des quotas sont ajustables et vous pouvez demander une augmentation de quota pour eux. Pour de plus amples informations, veuillez consulterDemande d'augmentation de quotadans le Guide de l'utilisateur Service Quotas.

Service App Runner basé sur une image source

Vous pouvez utiliserAWS App Runnerpour créer et gérer des services basés sur deux types de sources de services fondamentalement différents :Code sourceandimage source. Quel que soit le type de source, App Runner s'occupe du démarrage, de l'exécution, de la mise à l'échelle et de l'équilibrage de charge de votre service. Vous pouvez utiliser la fonctionnalité CI/CD d'App Runner pour suivre les modifications apportées à votre image ou code source. Lorsque App Runner découvre une modification, il génère automatiquement (pour le code source) et déploie la nouvelle version sur votre service App Runner.

Ce chapitre traite des services basés sur une image source. Pour plus d'informations sur les services basés sur le code source, consultezService basé sur le code.

Aimage sourceest une image de conteneur publique ou privée stockée dans un référentiel d'images. Vous pointez App Runner vers une image, et il démarre un service exécutant un conteneur basé sur cette image. Aucune étape de construction n'est nécessaire. Vous fournissez plutôt une image prête à être déployée.

Fournisseurs d'images référentiel

App Runner prend en charge les fournisseurs de référentiel d'images suivants :

- Amazon Elastic Container Registry (Amazon ECR)— Stocke des images privées dans votreCompte AWS.
- Amazon Elastic Container Registry Public— Stocke les images lisibles publiquement.

Déploiement à partir d'Amazon ECR

Amazon ECRstocke les images dans les référentiels. Il y a des dépôts privés et publics. Pour déployer votre image sur un service App Runner à partir d'un référentiel privé, App Runner doit être autorisé à lire votre image depuis Amazon ECR. Pour donner cette autorisation à App Runner, vous devez fournir à App Runner unRôle d'accès. Il s'agit d'unAWS Identity and Access Management(IAM) qui dispose des autorisations d'action Amazon ECR requises. Lorsque vous utilisez la console App Runner pour créer le service, vous pouvez choisir un rôle existant dans votre compte. Vous pouvez également utiliser la console IAM pour créer un rôle personnalisé, ou choisir pour la console App Runner de créer un rôle pour vous en fonction des stratégies gérées.

Lorsque vous utilisez l'API App Runner ou leAWS CLIVous pouvez effectuer un processus en deux étapes. Tout d'abord, vous utilisez la console IAM pour créer un rôle d'accès. Vous pouvez utiliser une stratégie gérée fournie par App Runner ou entrer vos propres autorisations personnalisées. Ensuite, vous fournissez le rôle d'accès lors de la création du service à l'aide de l'outilCreateServiceAction d'API.

Pour plus d'informations sur la création du service App Runner, consultezthe section called “Création”.

Déploiement à partir d'Amazon ECR Public

Amazon ECR Publicstocke des images lisibles publiquement. Voici les principales différences entre Amazon ECR et Amazon ECR Public que vous devez connaître dans le contexte des services App Runner :

- Les images publiques Amazon ECR sont lisibles publiquement. Vous n'avez pas besoin de fournir un rôle d'accès lorsque vous créez un service basé sur une image Amazon ECR Public.
- App Runner ne prend pas en charge le déploiement automatique pour les images Amazon ECR Public.

Service App Runner basé sur le code source

Vous pouvez utiliserAWS App Runnerpour créer et gérer des services basés sur deux types de sources de services fondamentalement différents :Code sourceandimage source. Quel que soit le type de source, App Runner s'occupe du démarrage, de l'exécution, de la mise à l'échelle et de l'équilibrage de charge de votre service. Vous pouvez utiliser la fonctionnalité CI/CD d'App Runner pour suivre les modifications apportées à votre image ou code source. Lorsque App Runner découvre une modification, il génère automatiquement (pour le code source) et déploie la nouvelle version sur votre service App Runner.

Ce chapitre traite des services basés sur le code source. Pour plus d'informations sur les services basés sur une image source, voirService basé sur l'image.

Le code source est le code d'application que App Runner construit et déploie pour vous. Vous pointez App Runner vers un référentiel de code source et choisissez unEnvironnement d'exécution. App Runner crée une image basée sur l'image de base du moteur d'exécution et le code de votre application. Il démarre ensuite un service qui exécute un conteneur basé sur cette image.

App Runner fournit une langue spécifiqueenvironnements d'exécution gérés. Chacun de ces runtimes construit une image de conteneur à partir de votre code source et ajoute des dépendances d'exécution de langage dans votre image. Vous n'avez pas besoin de fournir de configuration de conteneur et d'instructions de construction telles qu'un fichier Dockerfile.

Les sous-thèmes de ce chapitre traitent des différentsEnvironnements d’exécutionpris en charge par App Runner : le moteur d'exécution Dockerfile générique et leenvironnements d'exécution géréspour différents environnements de programmation.

Rubriques

Fournisseur de référentiel de code source

App Runner déploie votre code source en le lisant à partir d'un référentiel de code source. App Runner prend en charge un fournisseur de référentiel de code source : GitHub.

Déploiement à partir de GitHub

Pour déployer votre code source dans un service App Runner à partir d'un fichier deGitHub, App Runner établit une connexion à GitHub. Si votre référentiel est privé (c'est-à-dire qu'il n'est pas accessible publiquement sur GitHub), vous devez fournir à App Runner les détails de connexion. Lorsque vous utilisez la console App Runner pourCréation d'un service, vous fournissez des informations de connexion dans le cadre de la procédure de création de service.

Lorsque vous utilisez l'API App Runner ou leAWS CLI, une connexion est une ressource distincte. Tout d'abord, vous créez la connexion à l'aide de la commandeCreateConnectionAction d'API. Ensuite, vous fournissez l'ARN de la connexion lors de la création du service à l'aide de l'outilCreateServiceAction d'API.

Pour plus d'informations sur la création de service App Runner, consultez.the section called “Création”. Pour plus d'informations sur vos connexions App Runner, consultez.the section called “Connexions”.

Exécutions gérées par App Runner

App Runner fournit des environnements d'exécution gérés pour divers environnements de programmation. Chaque exécution gérée facilite la création et l'exécution de conteneurs basés sur un langage de programmation ou un environnement d'exécution particulier. Lorsque vous utilisez un moteur d'exécution géré, App Runner commence par une image d'exécution gérée. Cette image est basée sur leImage Docker Amazon Linuxet contient un paquet d'exécution de langage ainsi que des outils et des paquets de dépendance populaires. App Runner utilise cette image d'exécution gérée comme image de base et ajoute votre code d'application pour créer une image Docker. Il déploie ensuite cette image pour exécuter votre service Web dans un conteneur.

Vous spécifiez un moteur d'exécution pour votre service App Runner lorsque vousCréation d'un serviceà l'aide de la console App Runner ou de laCreateServiceAPI. Vous pouvez également spécifier un moteur d'exécution dans le cadre de votre code source. Utilisation de l'`runtime`Mot clé dans un fichier deFichier de configuration App Runnerque vous incluez dans votre référentiel de code. La convention d'attribution de noms d'un moteur d'exécution géré est*`<language-name><major-version>`*.

App Runner met à jour le moteur d'exécution de votre service vers la dernière version de chaque déploiement ou mise à jour de service. Si votre application nécessite une version spécifique d'un moteur d'exécution géré, vous pouvez la spécifier à l'aide de la commande`runtime-version`Mot clé dans le fichier deFichier de configuration App Runner. Spécifiez une version mineure en

tant que*<major>*.*<minor>*pour verrouiller les versions principales et mineures (App Runner met à jour uniquement les versions de correctifs). Spécifiez un niveau de correctif particulier comme*<major>*.*<minor>*.*<patch>*pour verrouiller votre service sur une version d'exécution spécifique (App Runner ne met jamais à jour le moteur d'exécution).

Utilisation du moteur d'exécution géré Python

AWS App Runnerfournit un moteur d'exécution géré par Python. facilite la création et l'exécution de ces conteneurs avec des applications Web basées sur Python. Lorsque vous utilisez le moteur d'exécution Python, App Runner démarre avec une image d'exécution Python gérée. Cette image est basée sur laImage Docker Amazon Linuxet contient le paquet d'exécution Python et quelques outils et paquets de dépendances populaires. App Runner utilise cette image d'exécution gérée comme image de base et ajoute votre code d'application pour créer une image Docker. Il déploie ensuite cette image pour exécuter votre service Web dans un conteneur.

Vous spécifiez un moteur d'exécution pour votre service App Runner lorsque vousCréation d'un serviceà l'aide de la console App Runner ou de laCreateServiceAPI. Vous pouvez également spécifier un moteur d'exécution dans le cadre de votre code source. Utilisation de l'`runtime`Mot clé dans unFichier de configuration App Runnerque vous incluez dans votre référentiel de code. La convention d'attribution de noms d'un moteur d'exécution géré est*<language-name> <major-version>*.

Pour connaître les noms d'exécution Python valides, consultezthe section called “Informations sur la version”.

App Runner met à jour le moteur d'exécution de votre service vers la dernière version de chaque déploiement ou mise à jour de service. Si votre application nécessite une version spécifique d'un moteur d'exécution géré, vous pouvez la spécifier à l'aide de l'outil`runtime-version`Mot clé dans laFichier de configuration App Runner. Spécifiez une version mineure en tant que*<major>*.*<minor>*pour verrouiller les versions principales et mineures (App Runner ne met à jour que les versions de correctifs). Spécifiez un niveau de correctif particulier en tant que*<major>*.*<minor>*.*<patch>*pour verrouiller votre service sur une version d'exécution spécifique (App Runner ne met jamais à jour le moteur d'exécution).

Rubriques

- Configuration d'exécution Python
- Exemples d'exécution Python
- Informations sur la version d'exécution Python

Configuration d'exécution Python

Lorsque vous choisissez un moteur d'exécution géré, vous devez également configurer, au minimum, les commandes de génération et d'exécution. Vous les configurez alors quecreatingouactualisation en coursvotre service App Runner. Il existe différentes façons de le faire :

- Utilisation de la console App Runner— Spécifie les commandes dans la zoneConfiguration de la générationdu processus de création ou de l'onglet de configuration.
- Utilisation de l'API App Runner— AppelezCreateServiceouUpdateService. Spécifiez les commandes à l'aide de la`BuildCommand`and`StartCommand`membres de laCodeConfigurationValuesType de données.
- Utilisation d'uneFichier de configuration : spécifiez une ou plusieurs commandes de construction dans un maximum de trois phases de construction, et une seule commande d'exécution qui sert à démarrer votre application. Il existe d'autres paramètres de configuration facultatifs.

La fourniture d'un fichier de configuration est facultative. Lorsque vous créez un service App Runner à l'aide de la console ou de l'API, vous spécifiez si App Runner obtient vos paramètres de configuration directement lors de la création ou à partir d'un fichier de configuration.

Exemples d'exécution Python

Les exemples suivants illustrent les fichiers de configuration App Runner pour la création et l'exécution d'un service Python. Le dernier exemple est le code source d'une application Python complète que vous pouvez déployer sur un service d'exécution Python.

Fichier de configuration Python minimal

Cet exemple montre un fichier de configuration minimal que vous pouvez utiliser avec le moteur d'exécution géré Python. Pour connaître les hypothèses qu'App Runner fait avec un fichier de configuration minimal, consultezthe section called “Exemples de fichiers de configuration”.

Example apprunner.yaml

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

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

Fichier de configuration Python étendu

Cet exemple montre l'utilisation de toutes les clés de configuration avec le moteur d'exécution géré Python.

Example apprunner.yaml

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

Source d'application Python de bout en bout

Cet exemple montre le code source d'une application Python complète que vous pouvez déployer sur un service d'exécution Python.

Example requirements.txt

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

Example server.py

```
from flask import Flask
import os

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

app = Flask(__name__)


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


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

Example apprunner.yaml

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

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

Informations sur la version d'exécution Python

Cette rubrique répertorie tous les détails des versions d'exécution Python prises en charge par App Runner.

Python 3

Detail	Description
Nom de l'exécution	python3
Versions mineures	3.7, 3.8
Forfaits inclus	python, pip, setuptools, roue, virtualenv

Utilisation du moteur d'exécution géré Node.js

AWS App Runnerfournit un moteur d'exécution géré Node.js. Le runtime facilite la création et l'exécution de conteneurs avec des applications web basées sur Node.js. Lorsque vous utilisez le runtime Node.js, App Runner démarre avec une image d'exécution Node.js gérée. Cette image est basée sur laImage Amazon Linux Dockeret contient le package d'exécution Node.js et quelques outils. App Runner utilise cette image d'exécution gérée comme image de base et ajoute votre code d'application pour créer une image Docker. Il déploie ensuite cette image pour exécuter votre service Web dans un conteneur.

Vous spécifiez un moteur d'exécution pour votre service App Runner lorsque vousCréation d'un serviceà l'aide de la console App Runner ou de laCreateServiceAPI. Vous pouvez également spécifier un moteur d'exécution dans le cadre de votre code source. Utilisation de l'`runtime`dans un mot-cléFichier de configuration App Runnerque vous incluez dans votre référentiel de code. La convention d'attribution de noms d'un moteur d'exécution géré est*`<language-name> <major-version>`*.

Pour connaître les noms d'exécution Node.js valides, consultezthe section called “Informations sur la version”.

App Runner met à jour le moteur d'exécution de votre service vers la dernière version de chaque déploiement ou mise à jour de service. Si votre application nécessite une version

spécifique d'un moteur d'exécution géré, vous pouvez la spécifier à l'aide de l'outil`runtime-version`ondans la boîte de dialogueFichier de configuration App Runner. Spécifiez une version mineure en tant que*<major>*.*<minor>*pour verrouiller les versions principales et mineures (App Runner ne met à jour que les versions de correctifs). Spécifiez un niveau de correctif particulier comme*<major>*.*<minor>*.*<patch>*pour verrouiller votre service sur une version d'exécution spécifique (App Runner ne met jamais à jour le moteur d'exécution).

Rubriques

- Configuration de runtime Node.js
- Exemples de runtime Node.js
- Informations sur la version de runtime Node.js

Configuration de runtime Node.js

Lorsque vous choisissez un moteur d'exécution géré, vous devez également configurer, au minimum, les commandes de génération et d'exécution. Vous les configurez alors quecreatingouactualisation en coursvotre service App Runner. Il existe différentes façons de le faire :

- Utilisation de la console App Runner— Spécifie les commandes dans la zoneConfiguration de la générationdu processus de création ou de l'onglet de configuration.
- Utilisation de l'API App Runner— AppelezCreateServiceouUpdateService. Spécifiez les commandes à l'aide de la`BuildCommand`and`StartCommand`Membres de laCodeConfigurationValuesType de données.
- Utilisation d'uneFichier de configuration : spécifiez une ou plusieurs commandes de construction dans un maximum de trois phases de construction, et une seule commande d'exécution qui sert à démarrer votre application. Il existe d'autres paramètres de configuration facultatifs.

La fourniture d'un fichier de configuration est facultative. Lorsque vous créez un service App Runner à l'aide de la console ou de l'API, vous spécifiez si App Runner obtient vos paramètres de configuration directement lors de la création ou à partir d'un fichier de configuration.

Avec le runtime Node.js spécifiquement, vous pouvez également configurer la build et l'exécution à l'aide d'un fichier JSON nommé`package.json`dans la racine de votre référentiel source. À l'aide de ce fichier, vous pouvez configurer la version du moteur Node.js, les packages de dépendance et diverses commandes (applications de ligne de commande). Les gestionnaires de paquets tels que npm ou yarn interprètent ce fichier comme entrée pour leurs commandes.

Exemples :

- npm installinstalle les paquets définis par ledependenciesanddevDependenciesnœud danspackage.json.
- npm startounpm run startexécute la commande définie parscripts/startnœud danspackage.json.

Voici un exemple de fichier package.json.

package.json

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

Pour plus d'informations surpackage.json, voirGuide de package.jsonsur le site web Node.js.

Tips

- Si vos recettespackage.jsondéfinit un fichierstart, vous pouvez l'utiliser commerundans votre fichier de configuration App Runner, comme l'illustre l'exemple suivant.

Example

package.json

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

apprunner.yaml

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

- Lorsque vous exécuteznpm installdans votre environnement de développement, npm crée le fichier`package-lock.json`. Ce fichier contient un instantané des versions du paquet npm qui viennent d'être installées. Par la suite, lorsque npm installe des dépendances, il utilise ces versions exactes. De même, le fil crée`yarn.json`. Envoyez ces fichiers dans votre référentiel de code source pour vous assurer que votre application est installée avec les versions des dépendances avec lesquelles vous l'avez développée et testée.
- Vous pouvez également utiliser un fichier de configuration App Runner pour configurer la version Node.js et la commande start. Lorsque vous procédez ainsi, ces définitions remplacent celles de`package.json`. Un conflit entre`node`version dans`package.json`et l'`runtime-version`dans le fichier de configuration App Runner provoque l'échec de la phase de construction App Runner.

Exemples de runtime Node.js

Les exemples suivants illustrent les fichiers de configuration App Runner pour la création et l'exécution d'un service Node.js.

Fichier de configuration Node.js minimal

Cet exemple montre un fichier de configuration minimal que vous pouvez utiliser avec le moteur d'exécution géré Node.js. Pour connaître les hypothèses qu'App Runner fait avec un fichier de configuration minimal, consultezthe section called “Exemples de fichiers de configuration”.

Example apprunner.yaml

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

Fichier de configuration Node.js étendu

Cet exemple montre l'utilisation de toutes les clés de configuration avec le moteur d'exécution géré 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"
```

Application Node.js avec Grunt

Cet exemple montre comment configurer une application Node.js développée avec Grunt. Grognementsest un coureur de tâches JavaScript en ligne de commande. Il exécute des tâches répétitives et gère l'automatisation des processus pour réduire les erreurs humaines. Les plugins Grunt et Grunt sont installés et gérés à l'aide de npm. Vous configurez Grunt en incluant le`Gruntfile.js`dans la racine de votre référentiel source.

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
```

Informations sur la version de runtime Node.js

Cette rubrique répertorie tous les détails des versions d'exécution Node.js prises en charge par App Runner.

Node.js 12

Detail	Description
Nom de l’exécution	nodejs12
Versions mineures	latest
Forfaits inclus	nodejs (y compris npm), fils

Développement d'un code d'application pour App Runner

Ce chapitre traite des informations d'exécution et des directives de développement que vous devez prendre en compte lors du développement ou de la migration du code d'application pour le déploiement versAWS App Runner.

Informations d'exécution

Que vous fournissiez une image de conteneur ou que App Runner en crée une pour vous, App Runner exécute votre code d'application dans une instance de conteneur. Voici quelques aspects clés de l'environnement d'exécution de l'instance de conteneur.

- Support Framework— App Runner prend en charge toute image qui implémente une application Web. Il est indépendant du langage de programmation que vous choisissez et du serveur d'applications Web ou du framework que vous utilisez, le cas échéant. Pour votre commodité, nous fournissons des runtimes gérés spécifiques à la langue afin de rationaliser le processus de création d'applications et la création d'images abstraites.
- Demandes Web— Votre instance de conteneur doit écouter les requêtes HTTP, sur le port 8080 par défaut. Pour de plus amples informations sur la configuration de votre service, veuillez consulterthe section called “Configuration”. Vous n'avez pas besoin d'implémenter la gestion du trafic sécurisé HTTPS. App Runner nécessite le trafic HTTPS entrant et termine HTTPS avant de transmettre des demandes à votre instance de conteneur.
- Apps sans état— App Runner ne garantit pas la persistance de l'état au-delà de la durée du traitement d'une seule requête Web entrante.
- Stockage— App Runner implémente le système de fichiers dans votre instance de conteneur en tant questockage éphémère. Les fichiers sont transitoires. Par exemple, elles ne persistent pas lorsque vous arrêtez et reprenez votre service App Runner. Plus généralement, les fichiers ne sont pas garantis de persister au-delà du traitement d'une seule demande, dans le cadre de la nature apatride de votre application. Toutefois, les fichiers stockés prennent une partie de l'allocation de stockage de votre service App Runner pendant toute la durée de leur durée de vie.

  > **Note**
  >
  > Bien que les fichiers de stockage éphémères puissent ne pas persister entre les requêtes, ilsParfoispersist-t-il. Cela peut être utile dans certaines situations. Par exemple, lorsque vous traitez une demande, vous pouvez mettre en cache les fichiers téléchargés par

votre application si de futures demandes peuvent en avoir besoin. Cela peut accélérer le traitement des demandes futures, mais ne peut pas garantir les gains de vitesse. Votre code ne doit pas supposer qu'un fichier qui a été téléchargé dans une requête précédente existe toujours.
Pour une mise en cache garantie à l'aide d'un stockage de données en mémoire à haut débit et à faible latence, utilisez un service tel queAmazon ElastiCache.

- Environment variables (Variables d'environnement)— Par défaut, App Runner rend le`PORT`disponible dans votre instance de conteneur. Vous pouvez configurer la valeur de la variable avec des informations de port et ajouter des variables et des valeurs d'environnement personnalisées. Pour de plus amples informations sur la configuration de votre service, veuillez consulterthe section called “Configuration”.
- Rôle d'instance— Si votre code d'application fait des appels àAWS, à l'aide des API de service ou de l'un desAWSSDK, créez un rôle d'instance à l'aide deAWS Identity and Access Management(IAM). Ensuite, attachez-le à votre service App Runner lorsque vous le créez. Include AllAWSles autorisations d'action de service requises par votre code dans votre rôle d'instance. Pour plus d'informations, consultez the section called “Rôle d'instance”.

Consignes d'élaboration de code

Tenez compte de ces directives lors du développement du code pour une application Web App Runner.

- Conception d'un code sans état— Concevez l'application Web que vous déployez sur votre service App Runner pour qu'elle soit sans état. Votre code doit supposer qu'aucun état ne persiste au-delà de la durée du traitement d'une seule requête Web entrante.
- Supprimer les fichiers temporaires— Lorsque vous créez des fichiers, ils sont stockés sur un système de fichiers et prennent en charge une partie de l'allocation de stockage de votre service. Pour éviter les erreurs de stockage, ne conservez pas de fichiers temporaires pendant des périodes prolongées. Équilibre la taille du stockage avec la vitesse de traitement des demandes lors de la prise de décisions de mise en

Utilisation de la console App Runner

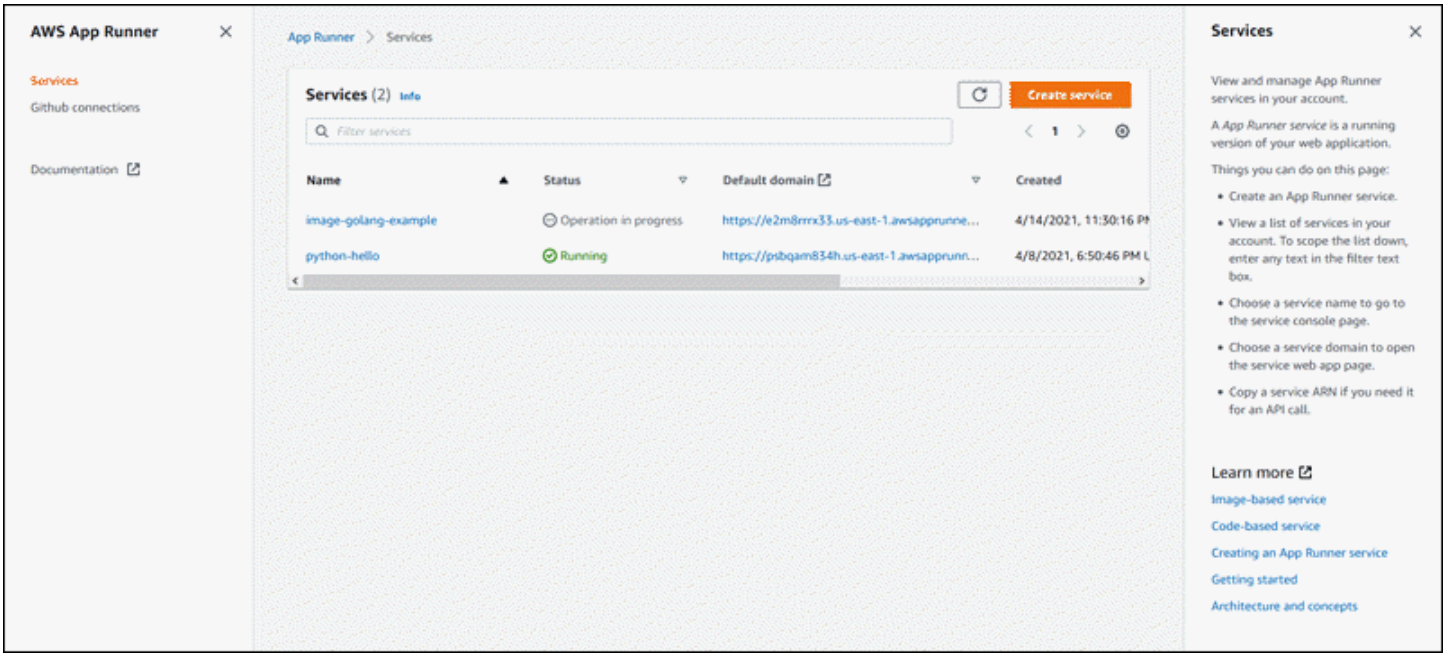
Utilisation de l'AWS App Runnerpour créer, gérer et surveiller vos services App Runner et les ressources associées, telles que les connexions. Vous pouvez afficher les services existants, en créer de nouveaux et configurer un service. Vous pouvez afficher l'état d'un service App Runner ainsi que les journaux, surveiller l'activité et suivre les mesures. Vous pouvez également accéder au site Web de votre service ou à votre référentiel source.

Les sections suivantes décrivent la disposition et les fonctionnalités de la console et vous pointent vers des informations connexes.

Présentation globale de la console

La console App Runner comporte trois zones. De gauche à droite :

- Volet de navigation— Volet latéral qui peut être réduit ou développé. Utilisez-le pour choisir la page de console de niveau supérieur que vous souhaitez utiliser.
- Volet de contenu— Partie principale de la page de la console. Utilisez-le pour afficher des informations et effectuer vos tâches.
- Volet Aide— Un volet latéral pour plus d'informations. Développez-le pour obtenir de l'aide sur la page sur laquelle vous êtes. Ou choisissez unInfossur une page de console pour obtenir de l'aide contextuelle.



Page des services

La .Servicesrépertorie les services App Runner dans votre compte. Vous pouvez étendre la liste vers le bas à l'aide de la zone de texte du filtre.

Pour accéder à laServicespage

1. Ouverture d'Exécuteur d'applications, et dans leRégions, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices.

Ce que vous pouvez faire ici :

- Création d'un service App Runner. Pour plus d'informations, consultez the section called “Création”.
- Choisissez un nom de service pour accéder à la page de la console du tableau de bord de service.
- Choisissez un domaine de service pour ouvrir la page de l'application Web du service.

Page du tableau de bord du service

Vous pouvez afficher des informations sur un service App Runner et les gérer à partir de la page dashbaord du service. En haut de la page, vous pouvez voir le nom du service.

Pour accéder au tableau de bord des services, accédez auServices(voir la section précédente), puis choisissez votre service App Runner.

La .Présentation du servicefournit des informations de base sur le service App Runner et votre application. Ce que vous pouvez faire ici :

- Afficher les détails du service, tels que l'état, l'état et l'ARN.
- Accédez à .Domaine par défaut : le domaine fourni par App Runner pour l'application Web exécutée dans votre service. Il s'agit d'un sous-domaine dans le`awsapprunner.com`appartenant à App Runner.
- Accédez au référentiel source déployé sur le service.
- Démarrez un déploiement de référentiel source vers votre service.
- Suspendre, reprendre et supprimer votre service.

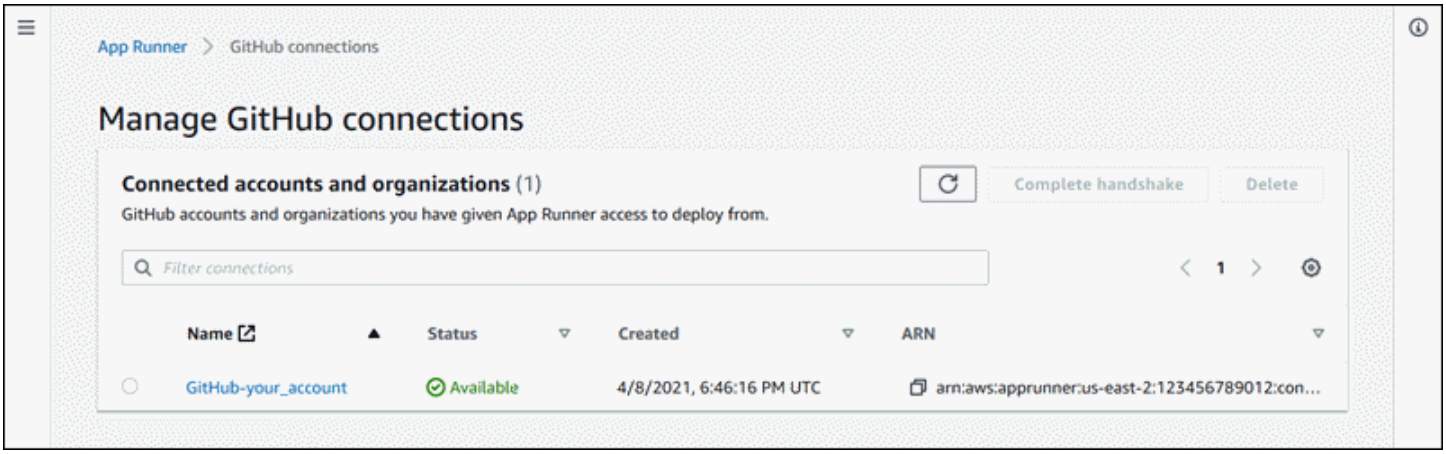
Les onglets sous la vue d'ensemble du service sont pour le servicemanagementandSurveillance.

La page Connexions GitHub

La .Connexions GitHubrépertorie les connexions App Runner à GitHub dans votre compte. Vous pouvez étendre la liste vers le bas à l'aide de la zone de texte du filtre. Pour plus d'informations sur les connexions, consultez the section called “Connexions”.

Pour accéder à laConnexions GitHubpage

1. Ouverture d'Exécuteur d'applications, et dans leRégions, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezConnexions GitHub.



Ce que vous pouvez faire ici :

- Affichez une liste des connexions GitHub dans votre compte. Pour étendre la liste vers le bas, entrez n'importe quel texte dans la zone de texte du filtre.
- Choisissez un nom de connexion pour accéder au compte ou à l'organisation GitHub associé.
- Sélectionnez une connexion pour terminer la négociation d'une connexion que vous venez d'établir (dans le cadre de la création d'un service) ou pour supprimer la connexion.

Gestion du cycle de vie de votre service App Runner

Ce chapitre décrit comment gérer le cycle de vie de votreAWS App Runnerservice Dans ce chapitre, vous apprendrez à créer, configurer et supprimer un service, à déployer de nouvelles versions d'application sur votre service et à gérer les connexions. Vous apprenez également à contrôler la disponibilité de votre service Web en mettant en pause et en reprenant votre service.

Rubriques

Création d'un service App Runner

AWS App Runnerautomatise le processus de passage d'une image de conteneur ou d'un référentiel de code source à un service Web en cours d'exécution qui évolue automatiquement. Vous pointez App Runner vers votre image ou votre code source, en spécifiant seulement un petit nombre de paramètres requis. App Runner construit votre application si nécessaire, provisionne les ressources de calcul et déploie votre application pour les exécuter.

Lorsque vous créez un service, App Runner crée unServiceRessource. Dans certains cas, il se peut que vous ayez besoin de fournir unconnexionRessource. Si vous utilisez la console App Runner, la console crée implicitement la ressource de connexion. Pour plus d'informations sur les types de ressources App Runner, consultezthe section called “Ressources App Exécuteur”. Ces types de ressources ont des quotas qui sont associés à votre compte dans chaqueRégion AWS. Pour plus d'informations, consultez the section called “Quotas de ressources App Exécuteur”.

Il existe des différences subtiles dans la procédure de création d'un service en fonction du type de source et du fournisseur. Cette rubrique présente des procédures entièrement distinctes pour créer ces différents types de sources afin que vous puissiez suivre celle qui correspond le mieux à votre

situation. Pour obtenir une procédure de démarrage de base comprenant un exemple de code, consultezMise en route.

Prerequisites

Avant de créer votre service App Runner, veillez à effectuer les actions suivantes :

- Suivez les étapes de l'installation dansConfiguration de.
- Préparez votre source d'application. Vous pouvez utiliser un référentiel de code dansGitHubou une image de conteneur dansAmazon Elastic Container Registry (Amazon ECR)pour créer un service App Runner.

Création d'un service

Cette section décrit le processus de création des deux types de service App Runner : basé sur le code source et basé sur une image de conteneur.

Créer un service à partir d'un référentiel de code GitHub

Les sections suivantes expliquent comment créer un service App Runner lorsque votre source est un référentiel de code dansGitHub. Lorsque vous utilisez GitHub, App Runner doit se connecter à l'organisation ou au compte GitHub. Par conséquent, vous devez aider à établir cette connexion. Pour plus d'informations sur les connexions App Runner, consultezthe section called “Connexions”.

Lorsque vous créez le service, App Runner crée une image Docker contenant votre code d'application et vos dépendances. Il lance ensuite un service qui exécute une instance de conteneur de cette image.

Rubriques

- Création d'un service à partir de code à l'aide de la console App Runner
- Création d'un service à partir de code à l'aide de l'API App Runner ouAWS CLI

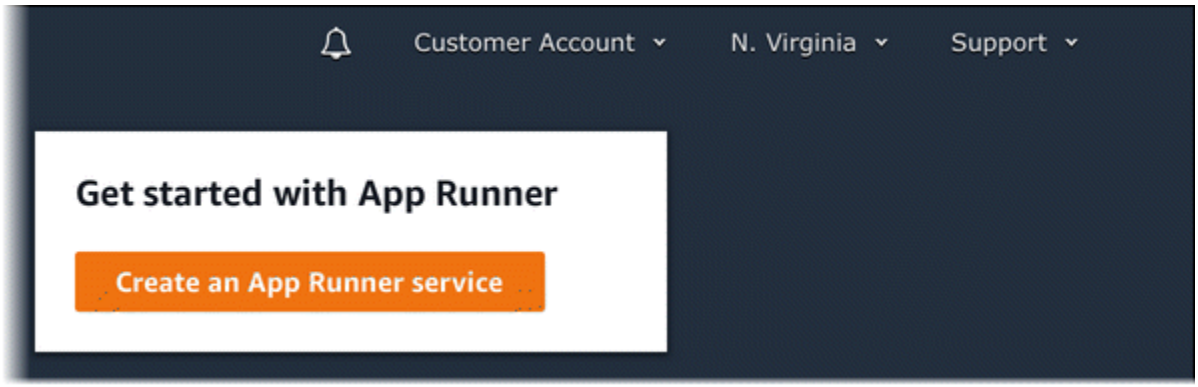
Création d'un service à partir de code à l'aide de la console App Runner

Pour créer un service App Runner à l'aide de la console

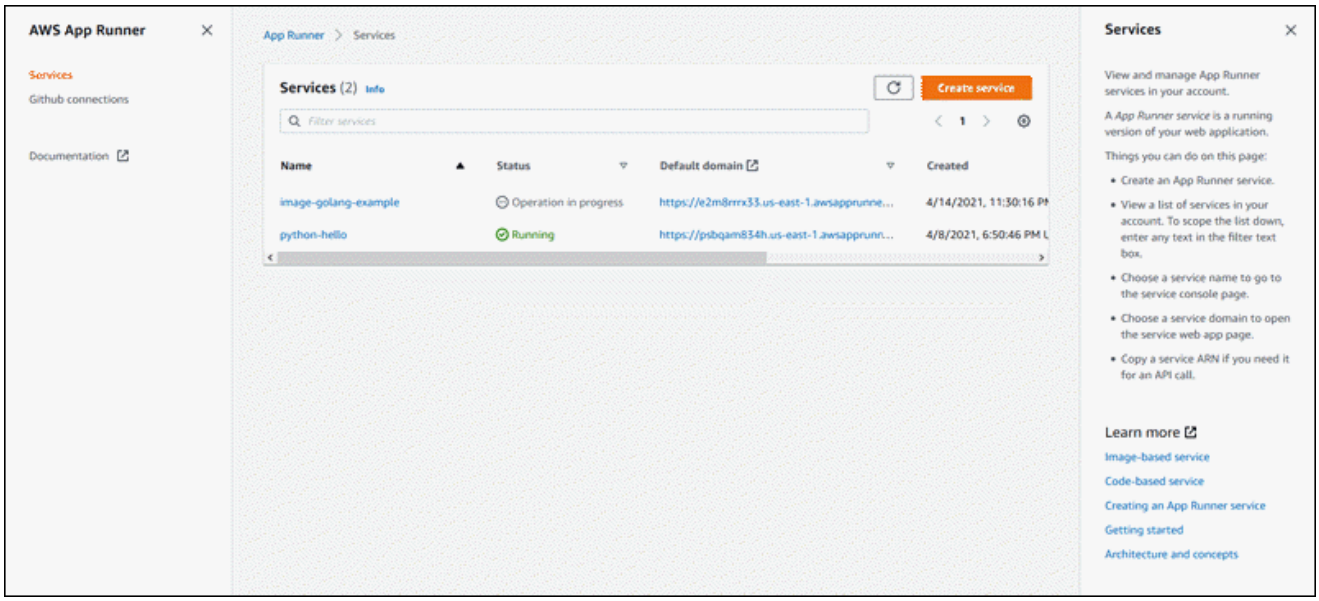
1. Configurez votre code source.

    a. Ouverture d'Console App Runner, et dans leRégions, sélectionnez votreRégion AWS.

b. Si l'icôneCompte AWSn'a pas encore de services App Runner, la page d'accueil de la console s'affiche. ChoisissezCréation d'un service App Runner.



Si l'icôneCompte AWSdispose de services existants, leServicesUne liste de vos services s'affiche. Choisissez Créer un service.



c. Dans la pageSource et déploiement, dans laSourceSection, pourType de référentiel, choisissezRéférentiel de code source.

d. PourConnect à GitHub, sélectionnez un compte ou une organisation GitHub que vous avez déjà utilisé ou choisissezAjouter un nouveau. Ensuite, passez par le processus de fourniture de vos informations d'identification GitHub et choisissez un compte ou une organisation GitHub auquel vous connecter.

e. PourRéférentiel., sélectionnez le référentiel qui contient votre code d'application.

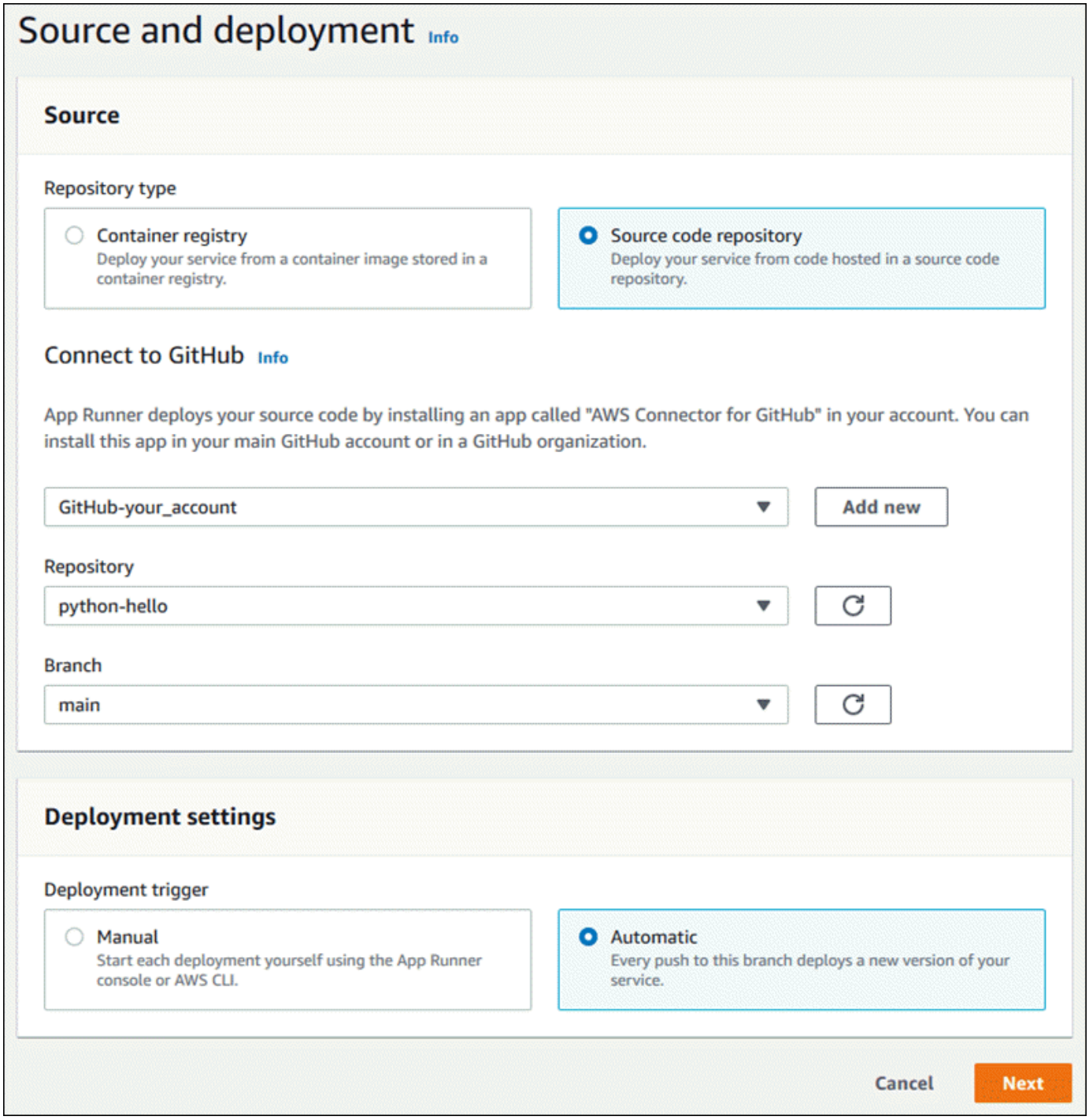
f. PourBranche, sélectionnez la branche que vous souhaitez déployer.

2. Configurez vos déploiements.

a. DansParamètres de déploiementSection, choisissezManuelleouAutomatic.

   Pour plus d'informations sur les méthodes de déploiement, consultezthe section called "Méthodes de déploiement".

b. Choisissez Next (Suivant).



3. Configurez la génération de l'application.

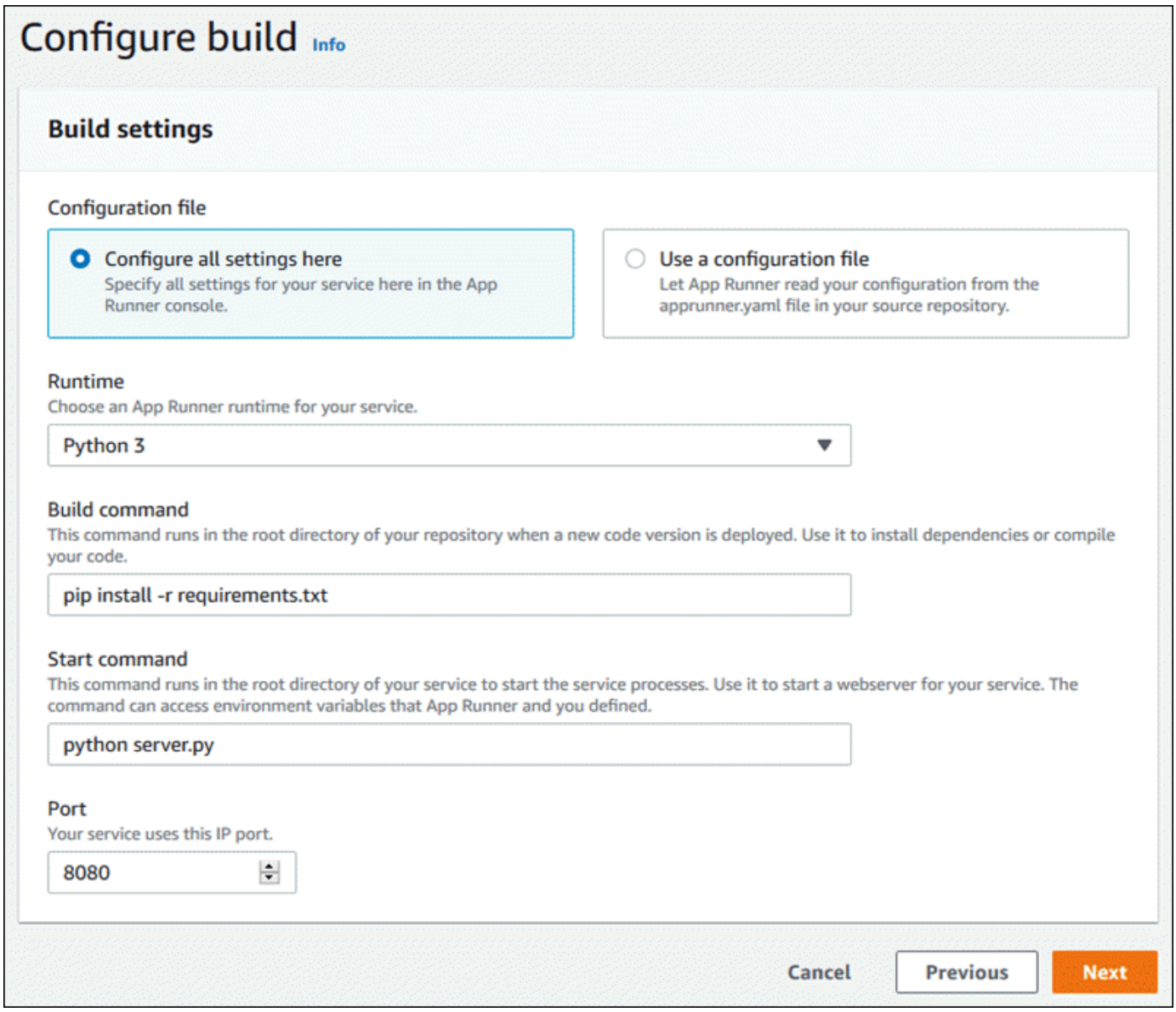
   a. Dans la pageConfiguration de la générationPage, pourFichier de configuration, choisissezConfigurez tous les paramètres icisi votre référentiel ne contient pas de fichier de configuration App Runner, ouUtilisation d'un fichier de configurationSi c'est le cas.

      > **Note**
      >
      > Un fichier de configuration App Runner est un moyen de maintenir votre configuration de build dans le cadre de votre source d'application. Lorsque vous en fournissez un, App Runner lit certaines valeurs du fichier et ne vous permet pas de les définir dans la console.

   b. Fournissez les paramètres de génération suivants :

      - Runtime (Exécution) : choisissez un moteur d'exécution géré spécifique pour votre application.
      - Commande Build— Entrez une commande qui crée votre application à partir de son code source. Il peut s'agir d'un outil spécifique à une langue ou d'un script fourni avec votre code.
      - Commande de démarrage— Entrez la commande qui démarre votre service Web.
      - Port— Entrez le port IP que votre service Web écoute.

   c. Choisissez Next (Suivant).



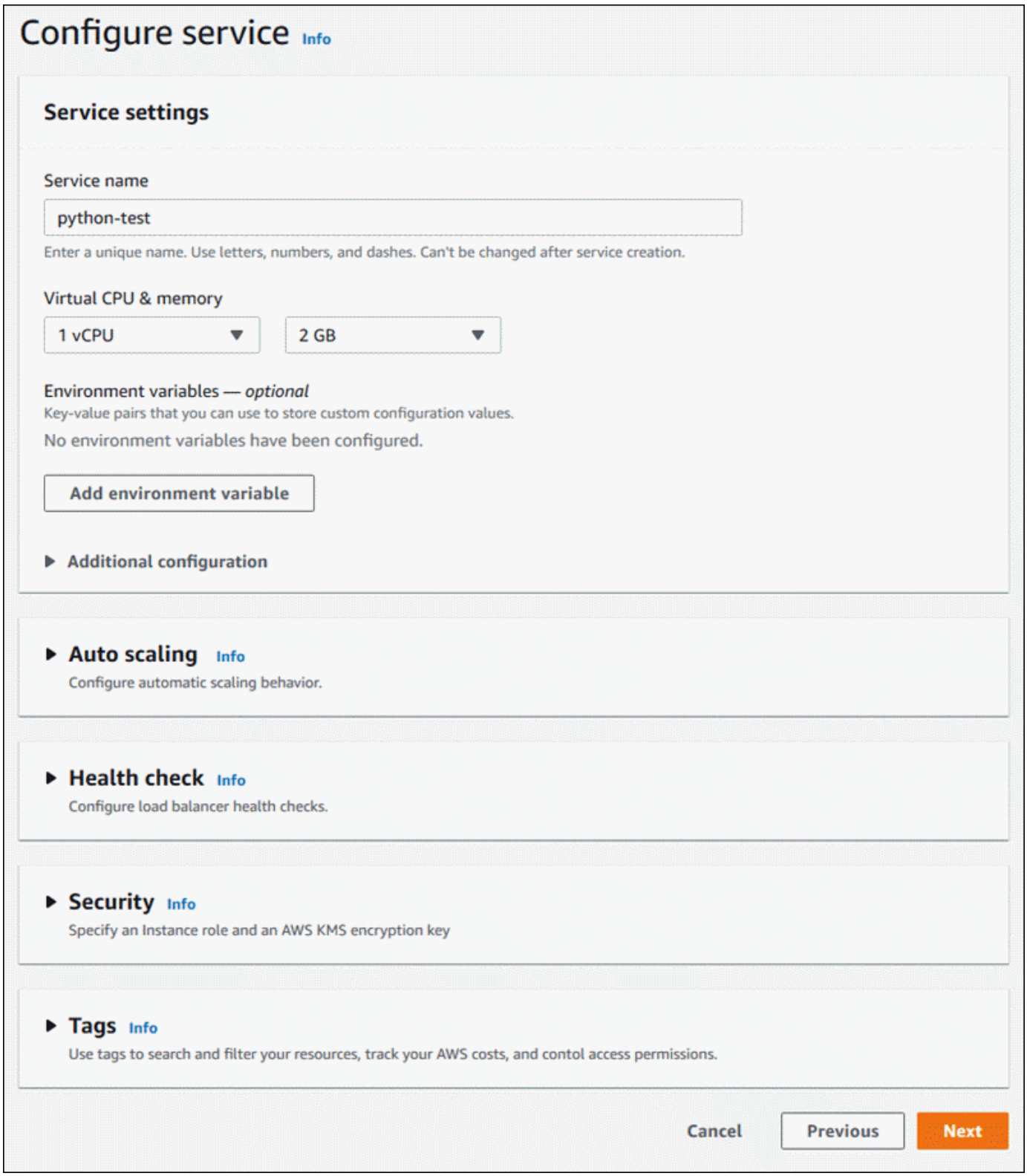
4. Configurez votre service.

   a. Dans la pageConfigurer le service, dans laParamètres du service, entrez un nom de service.

   > **Note**
   >
   > Tous les autres paramètres de service sont facultatifs ou ont des valeurs par défaut fournies par la console.

   b. Vous pouvez modifier ou ajouter d'autres paramètres afin de répondre aux exigences de votre application.

c. Choisissez Next (Suivant).



5. Dans la pageVérifier et créer, vérifiez tous les détails que vous avez entrés, puis choisissezCréation et déploiement.

   Résultat: Si la création du service réussit, la console doit afficher le tableau de bord du service, avec unPrésentation du serviceLe nouveau service.

6. Vérifiez que votre service est en cours d'exécution.

   a. Sur la page du tableau de bord du service, attendez que le serviceÉtatestEn cours d'exécution.
   b. Cliquez sur l'ongletDomaine par défaut : il s'agit de l'URL du site Web de votre service.
   c. Utilisez votre site Web et vérifiez qu'il fonctionne correctement.

Création d'un service à partir de code à l'aide de l'API App Runner ouAWS CLI

Pour créer un service à l'aide de l'API App Runner ouAWS CLI, appelez la`CreateService`Action d'API. Pour obtenir des informations et un exemple, veuillez consulter.CreateService. Si c'est la première fois que vous créez un service à l'aide d'une organisation ou d'un compte GitHub spécifique, commencez par appelerCreateConnection. Cela établit une connexion entre App Runner et l'organisation ou le compte GitHub. Pour plus d'informations sur les connexions App Runner, consultezthe section called “Connexions”.

La création de votre service démarre si l'appel renvoie une réponse réussie avec unServiceobjet affichage`"Status": "CREATING"`.

Pour obtenir un exemple d'appel, veuillez consulter.Création d'un service de référentiel de code sourcedans leAWS App RunnerAPI Reference

Création d'un service à partir d'une image Amazon ECR

Les sections suivantes expliquent comment créer un service App Runner lorsque votre source est une image de conteneur stockée dansAmazon ECR. Amazon ECR est unAWSservice. Par conséquent, pour créer un service basé sur une image Amazon ECR, vous fournissez à App Runner un rôle d'accès contenant les autorisations d'action Amazon ECR nécessaires.

> **Note**
>
> Un rôle d'accès n'est pas requis si votre image est stockée dans Amazon ECR Public, où les images sont accessibles au public.

Lors de la création du service, App Runner lance un service qui exécute une instance de conteneur de l'image que vous fournissez. Il n'y a pas de phase de construction dans ce cas.

Rubriques

- Création d'un service à partir d'une image à l'aide de la console App Runner
- Création d'un service à partir d'une image à l'aide de l'API App Runner ouAWS CLI

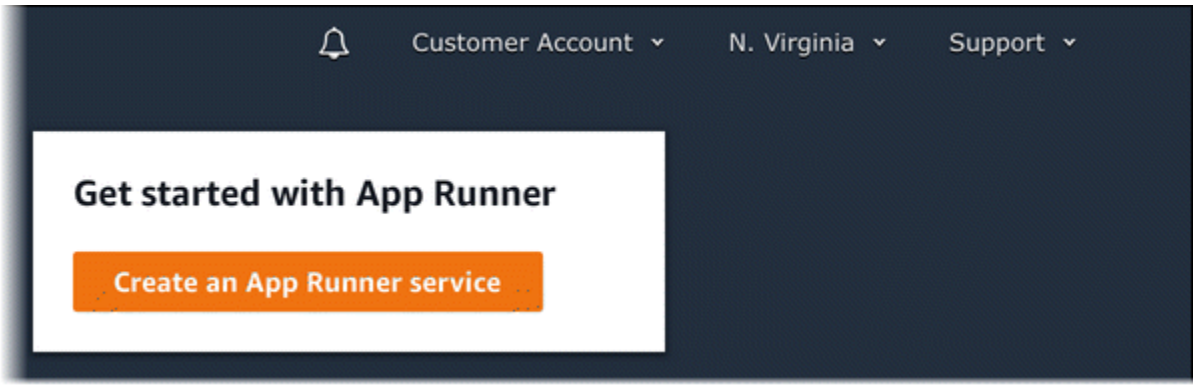
Création d'un service à partir d'une image à l'aide de la console App Runner

Pour créer un service App Runner à l'aide de la console

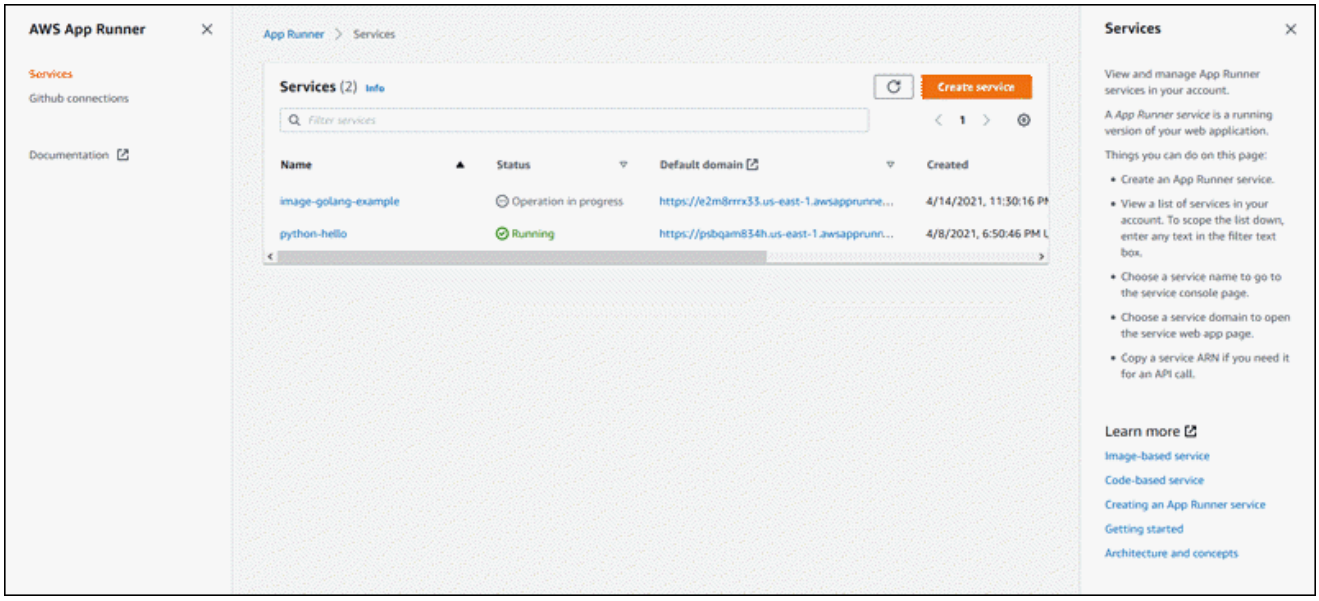
1. Configurez votre code source.

   a. Ouverture d'Console App Runner, et dans leRégions, sélectionnez votreRégion AWS.

   b. Si l'icôneCompte AWSn'a pas encore de services App Runner, la page d'accueil de la console s'affiche. ChoisissezCréation d'un service App Runner.



   Si l'icôneCompte AWSdispose de services existants, leServicesUne liste de vos services s'affiche. Choisissez Créer un service.



c. Dans la pageSource et déploiement, dans laSourceSection, pourType de référentiel, choisissezRegistre de conteneur.

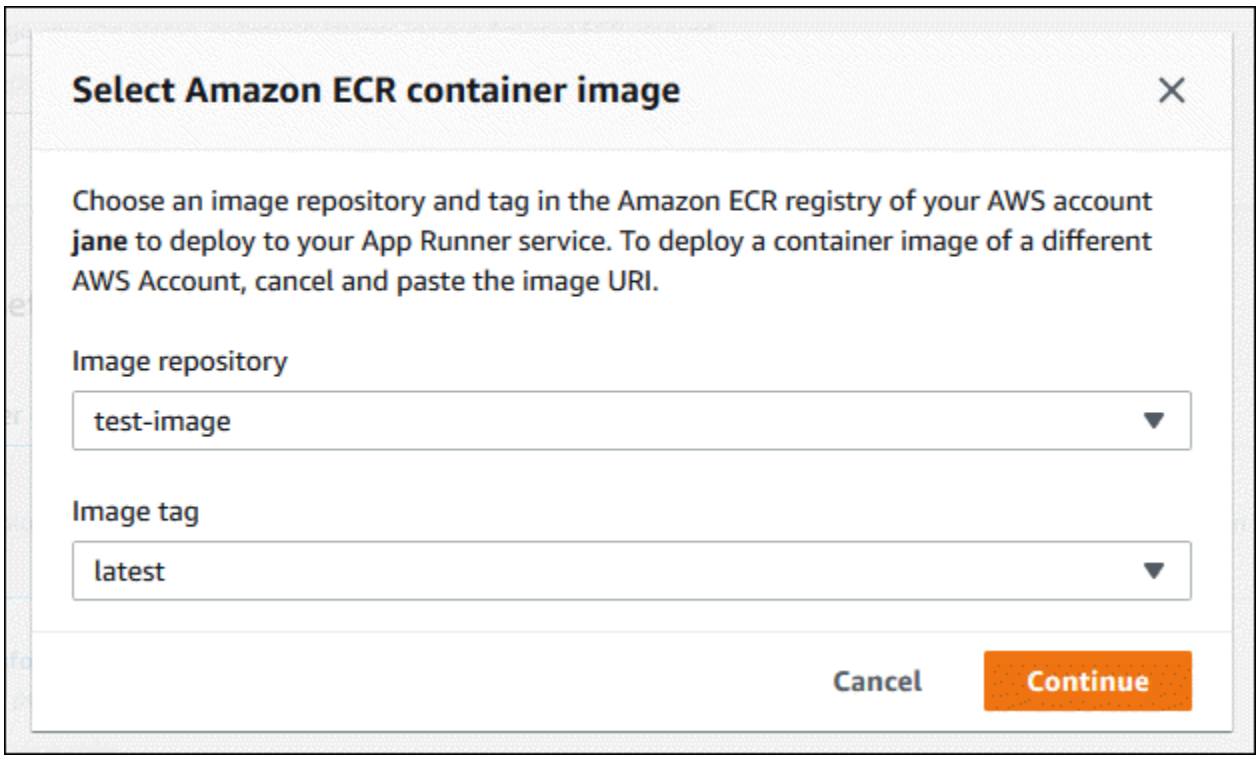
d. PourFournisseur, choisissez le fournisseur dans lequel votre image est stockée :

- Amazon ECR— Une image privée stockée dans Amazon ECR dans votreCompte AWS.
- Amazon ECR Public— Image lisible publiquement stockée dans Amazon ECR Public.

e. PourURI d'image de conteneur, choisissezParcourir.

f. DansSélectionner l'image du conteneur Amazon ECR, pourDépôt d'images, sélectionnez le référentiel qui contient votre image.

g. PourBalise d'image, sélectionnez la balise d'image spécifique que vous voulez déployer, par exemplelatest, puisContinuer.



2. Configurez vos déploiements.

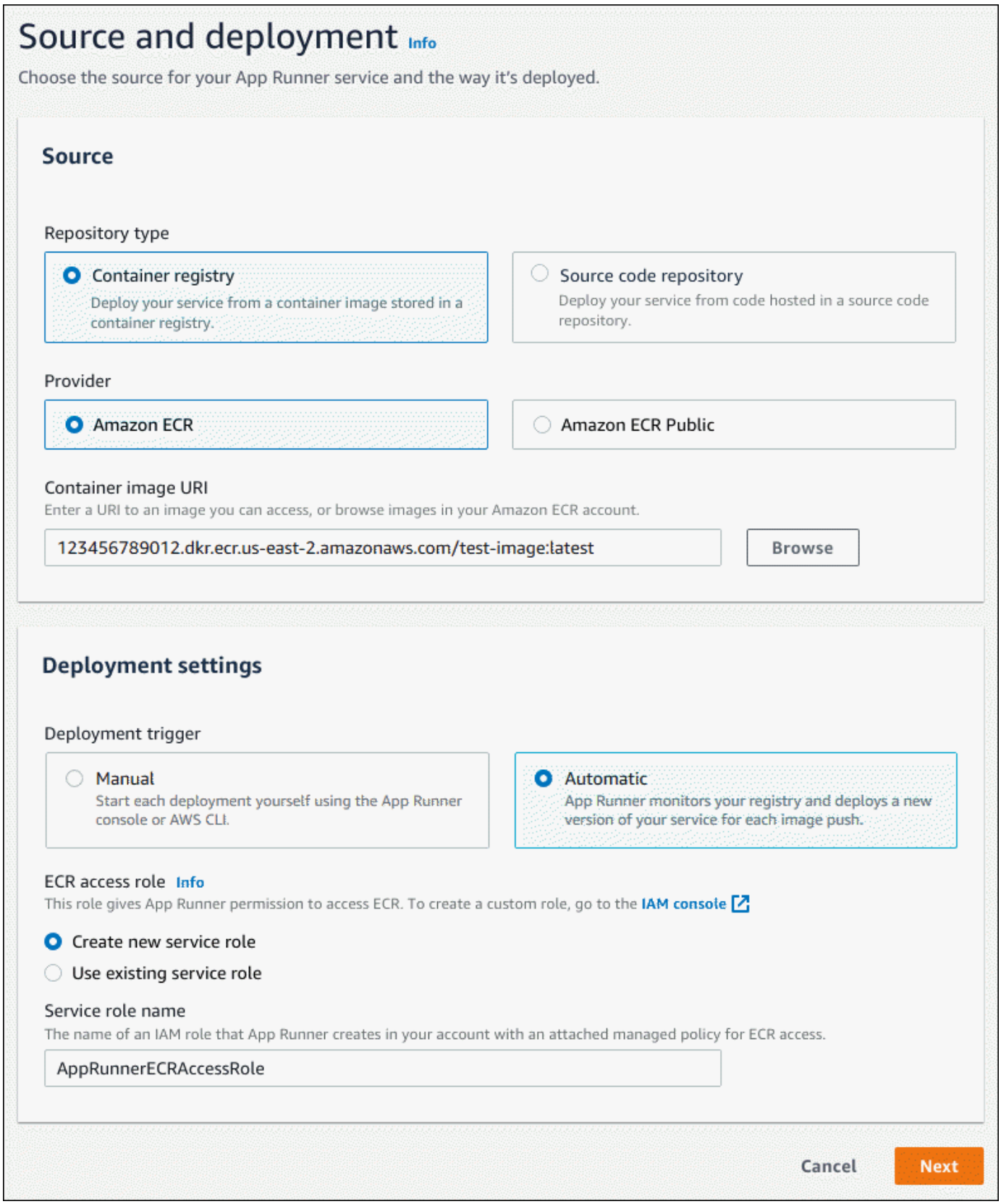
   a. DansParamètres de déploiementSection, choisissezManuelleouAutomatic.

      Pour plus d'informations sur les méthodes de déploiement, consultezthe section called “Méthodes de déploiement”.

      > **Note**
      >
      > App Runner ne prend pas en charge le déploiement automatique pour les images Amazon ECR Public.

   b. [Amazon ECRfournisseur] PourRôle d'accès ECR, choisissez un rôle de service existant dans votre compte ou choisissez de créer un nouveau rôle. Si vous utilisez le déploiement manuel, vous pouvez également choisir d'utiliser le rôle d'utilisateur IAM au moment du déploiement.

   c. Choisissez Next (Suivant).


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

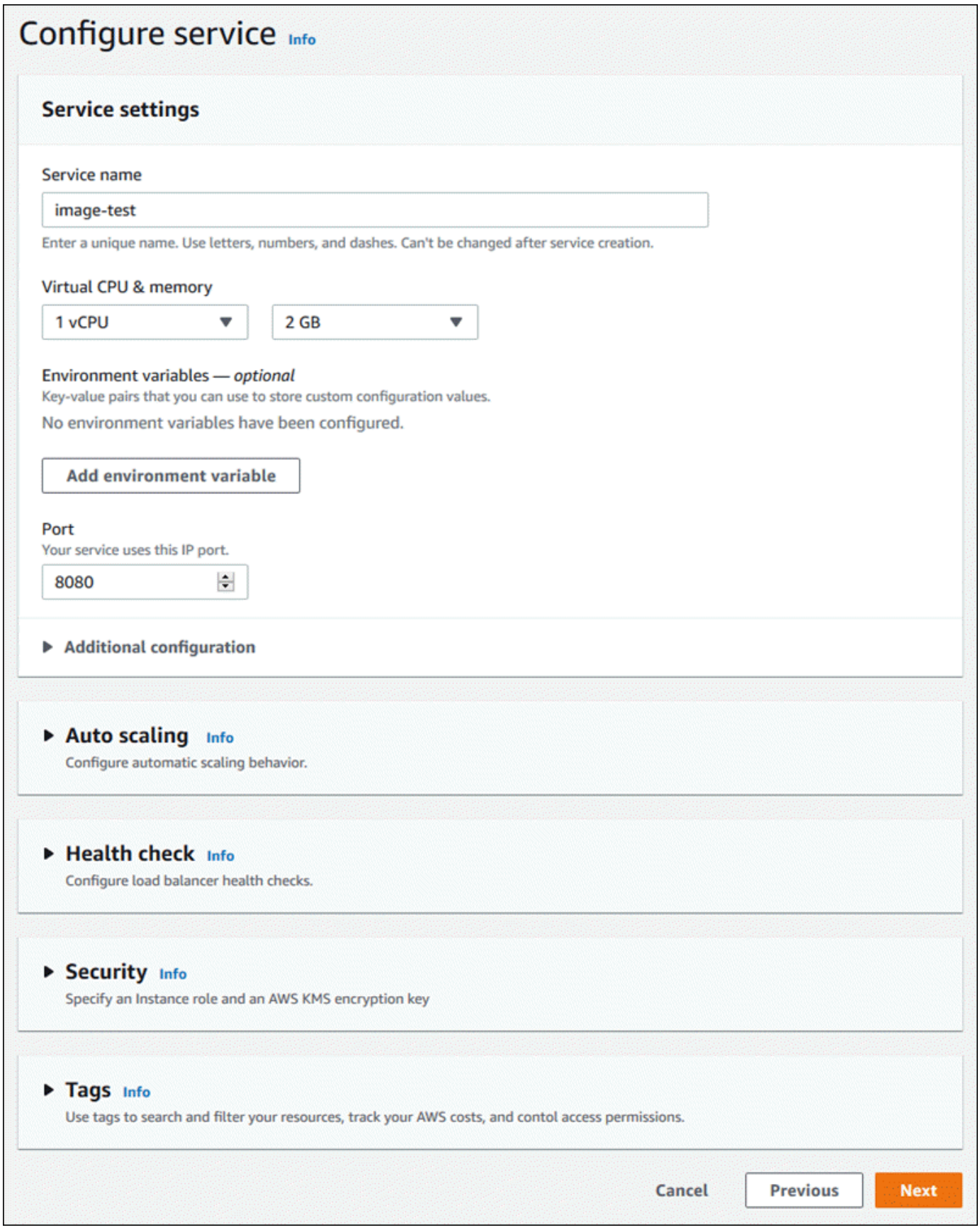
3. Configurez votre service.

   a. Dans la pageConfigurer le service, dans laParamètres du service, entrez un nom de service et le port IP que votre site Web de service écoute.

   > **Note**
   >
   > Tous les autres paramètres de service sont facultatifs ou ont des valeurs par défaut fournies par la console.

   b. (Facultatif) Modifiez ou ajoutez d'autres paramètres en fonction des besoins de votre application.

   c. Choisissez Next (Suivant).


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. Dans la pageVérifier et créer, vérifiez tous les détails que vous avez entrés, puis choisissezCréation et déploiement.

   Résultat: Si la création du service réussit, la console doit afficher le tableau de bord du service, avec unPrésentation du serviceLe nouveau service.

5. Vérifiez que votre service est en cours d'exécution.

   a. Sur la page du tableau de bord du service, attendez que le serviceÉtatestEn cours d'exécution.

   b. Cliquez sur l'ongletDomaine par défaut : il s'agit de l'URL du site Web de votre service.

   c. Utilisez votre site Web et vérifiez qu'il fonctionne correctement.

Création d'un service à partir d'une image à l'aide de l'API App Runner ouAWS CLI

Pour créer un service à l'aide de l'API App Runner ouAWS CLI, appelez laCreateServiceAction d'API.

La création de votre service démarre si l'appel renvoie une réponse réussie avec unServiceobjet affichage`"Status": "CREATING"`.

Pour obtenir un exemple d'appel, veuillez consulter.Création d'un service de référentiel d'images sourcedans leAWS App RunnerAPI Reference

Lorsque la création du service échoue

Si votre tentative de création d'un service App Runner échoue, le service affiche un état de`CREATE_FAILED`(échec de la créationsur la console).

Votre tentative de création d'un service peut échouer en raison de problèmes dans le code d'application, le processus de génération ou la configuration, parce que vous avez atteint des quotas de ressources ou en raison de problèmes temporaires avec leAWSque votre service doit utiliser. Pour résoudre un problème, nous recommandons de procéder comme suit. Tout d'abord, lisez les événements de service et les journaux pour savoir ce qui a causé l'échec. Ensuite, apportez les modifications nécessaires à votre code ou à votre configuration. Enfin, supprimez un ou plusieurs services si vous avez atteint votre quota de service. Ensuite, après avoir terminé toutes ces étapes, essayez de créer à nouveau le service.

> **Important**
>
> Le service défaillant n'est pas utilisable. Vous ne payez pas d'autres frais supplémentaires au-delà de la tentative de création initiale. Cependant, App Runner ne supprime pas automatiquement le service ayant échoué et il est toujours pris en compte dans votre quota de service. Lorsque vous avez terminé l'analyse de l'échec, assurez-vous de supprimer le service ayant échoué.

Déploiement d'une nouvelle version de l'application sur App Runner

Lorsque vousCréation d'un serviceinAWS App Runner, vous configurez une source d'application : une image de conteneur ou un référentiel source. App Runner provisionne des ressources pour exécuter votre service et déploie votre application sur ces ressources.

Cette rubrique décrit les moyens de redéployer la source de votre application vers votre service App Runner lorsqu'une nouvelle version devient disponible. Il peut s'agir d'une nouvelle version d'image dans le référentiel d'images ou d'un nouveau commit dans le référentiel de code. App Runner propose deux méthodes de déploiement vers un service :AutomatiquementandManuelle.

Méthodes de déploiement

App Runner fournit les méthodes suivantes pour contrôler la façon dont les déploiements d'applications sont initiés.

Déploiement automatique

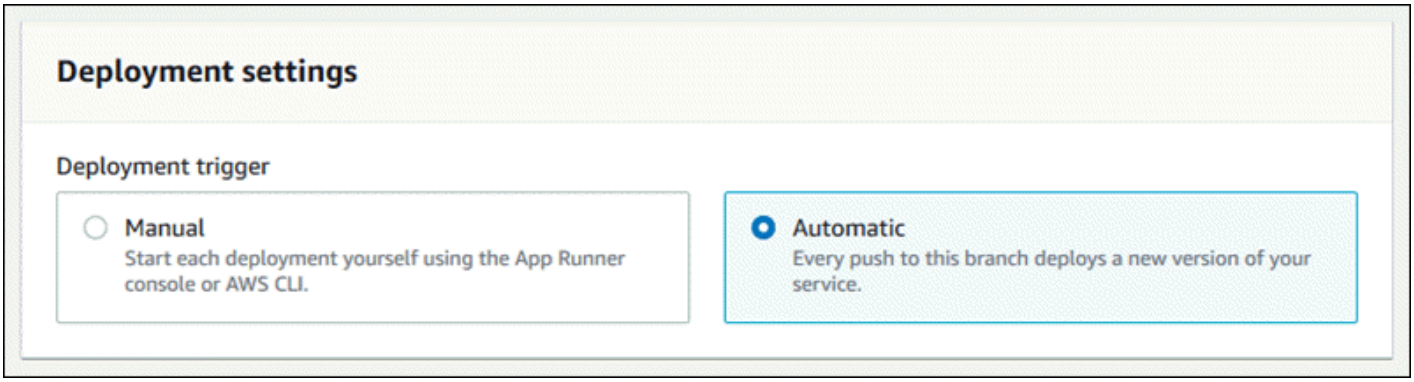
Utilisez le déploiement automatique lorsque vous souhaitez un comportement d'intégration et de déploiement continus (CICD) pour votre service. App Runner surveille votre image ou votre référentiel de code. Chaque fois que vous poussez une nouvelle version d'image dans votre référentiel d'images, ou un nouveau commit dans votre référentiel de code, App Runner la déploie automatiquement sur votre service sans autre action de votre côté.

Déploiement manuel

Utilisez le déploiement manuel lorsque vous souhaitez initier explicitement chaque déploiement vers votre service. Vous initiez un déploiement si le référentiel que vous avez configuré pour votre service a une nouvelle version que vous souhaitez déployer. Pour plus d'informations, consultez the section called “Déploiement manuel”.

Vous pouvez configurer la méthode de déploiement de votre service de plusieurs façons :

- Console— Pour un nouveau service que vous créez ou pour un service existant, dans la sectionParamètres de déploiementSection duSource et déploiementPage de configuration, choisissezManuelleouAutomatic.



- API ouAWS CLI— Dans un appel àCreateServiceouUpdateService, définissez la propriété`AutoDeploymentsEnabled`membre de laSourceConfigurationParamètre à`False`pour un déploiement manuel ou`True`pour un déploiement automatique.

Déploiement manuel

Avec le déploiement manuel, vous devez initier explicitement chaque déploiement vers votre service. Lorsqu'une nouvelle version de l'image ou du code de votre application est prête à être déployée, vous pouvez consulter les sections suivantes pour savoir comment effectuer un déploiement à l'aide de la console et de l'API.

Déployer une version d'application à l'aide de la console App Runner

Déploiement à l'aide de la console App Runner

1. Ouverture d'Console d'exécution d'applications, et dans leRégions, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. Choisissez Deploy (Déployer).

Résultat: Le déploiement de la nouvelle version démarre. Sur la page du tableau de bord du service, le serviceÉtatModifications apportées àOpération en cours.

4. Attendez que le déploiement soit terminé. Sur la page du tableau de bord du service, le serviceÉtatdevrait revenir àEn cours d'exécution.
5. Pour vérifier que le déploiement a réussi, sur la page du tableau de bord des services, sélectionnez l'optionDomaine par défaut : il s'agit de l'URL du site Web de votre service. Inspectez ou interagissez avec votre application Web et vérifiez votre modification de version.

Déployer une version d'application à l'aide de l'API App Runner ouAWS CLI

Pour déployer à l'aide de l'API App Runner ouAWS CLI, appelez la commandeStartDeploymentAction d'API. Le seul paramètre à passer est votre ARN de service. Vous avez déjà configuré l'emplacement source de votre application lorsque vous avez créé le service, et App Runner peut trouver la nouvelle version. Votre déploiement démarre si l'appel renvoie une réponse réussie.

Configuration d'un service App Runner

Lorsque vousCréation d'unAWS App RunnerService, vous définissez différentes valeurs de configuration. Vous pouvez modifier certains de ces paramètres de configuration après la création du service. D'autres paramètres peuvent être appliqués uniquement lors de la création du service et ne peuvent pas être modifiés par la suite. Cette rubrique décrit la configuration de votre service à l'aide de l'API App Runner, de la console App Runner et d'un fichier de configuration App Runner.

Configurez votre service à l'aide de l'API App Runner ouAWS CLI

L'API définit les paramètres qui peuvent être modifiés après la création du service. La liste suivante décrit les actions, les types et les limites pertinents.

- UpdateServiceaction : peut être appelée après la création pour mettre à jour certains paramètres de configuration.
  - Peut être mis à jour— Vous pouvez mettre à jour les paramètres dans la`SourceConfiguration`,`InstanceConfiguration`, et`HealthCheckConfiguration`Paramètres. Cependant, dans`SourceConfiguration`, vous ne pouvez pas changer votre type de source de code en image ou dans l'inverse. Vous devez fournir le même paramètre de référentiel que vous avez fourni lorsque vous avez créé le service. C'est soit`CodeRepository`ou`ImageRepository`.

Vous pouvez également mettre à jourAutoScalingConfigurationArn, l'ARN de la ressource de configuration de mise à l'échelle automatique associée au service.

- Impossible de mettre à jour— Vous ne pouvez pas modifier laServiceNameandEncryptionConfigurationqui sont disponibles dans la fenêtreCreateServiceaction. Ils ne peuvent pas être modifiés après leur création. La .UpdateServicen'inclut pas ces paramètres.
- API et fichier— Vous pouvez définir la propriétéConfigurationSourceParamètre de la propriétéCodeConfiguration(utilisé pour les référentiels de code source dans le cadre deSourceConfiguration) àRepository. Dans ce cas, App Runner ignore les paramètres de configuration dansCodeConfigurationValues, et lit ces paramètres à partir d'unFichier de configurationDans votre référentiel. Si vous définissezConfigurationSourcesurAPI, App Runner obtient tous les paramètres de configuration de l'appel API et ignore le fichier de configuration, même s'il en existe un.

- TagResourceaction : peut être appelée après la création de votre service pour ajouter des balises au service ou mettre à jour les valeurs des balises existantes.
- UntagResourceaction : peut être appelée après la création de votre service pour supprimer des balises du service.

Configurez votre service à l'aide de la console App Runner

La console utilise l'API App Runner pour appliquer les mises à jour de configuration. Les règles de mise à jour imposées par l'API, telles que définies dans la section précédente, déterminent ce que vous pouvez configurer à l'aide de la console. Certains paramètres qui étaient disponibles lors de la création du service ne sont pas disponibles pour modification ultérieure. En outre, si vous décidez d'utiliser unFichier de configuration, des paramètres supplémentaires sont masqués dans la console et App Runner les lit à partir du fichier.

Pour configurer votre service

1. Ouverture d'Application Runner, et dans leRégionsDans la liste, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. Dans la page du tableau de bord des services, choisissezConfigurationOnglet.

Résultat: La console affiche les paramètres de configuration actuels de votre service dans plusieurs sections : Source et déploiement,Configuration de la génération, etConfigurer le service.

4. Pour mettre à jour les paramètres de n'importe quelle catégorie, choisissezModifier.
5. Dans la page d'édition de la configuration, apportez les modifications souhaitées, puis choisissezEnregistrez les modifications.

Configurer votre service à l'aide d'un fichier de configuration App Runner

Lorsque vous créez ou mettez à jour un service App Runner, vous pouvez demander à App Runner de lire certains paramètres de configuration à partir d'un fichier de configuration que vous fournissez dans votre référentiel source. Ce faisant, vous pouvez gérer les paramètres liés à votre code source sous contrôle source, ainsi que le code lui-même. Le fichier de configuration fournit également certains paramètres avancés que vous ne pouvez pas définir à l'aide de la console ou de l'API. Pour plus d'informations, consultez Fichier de configuration App Runner.

Gestion des connexions App Runner

Lorsque vousCréation d'un serviceinAWS App Runner, vous configurez une source d'application : une image de conteneur ou un référentiel source stocké avec un fournisseur. Si un référentiel stocké auprès d'un fournisseur tiers est privé (non lisible publiquement), App Runner doit établir une connexion authentifiée et autorisée avec le fournisseur. Ensuite, App Runner peut lire votre référentiel et le déployer sur votre service. App Runner ne nécessite pas d'établissement de connexion lorsque vous créez un service qui accède au code stocké dans votreCompte AWSou dans un emplacement de code public.

App Runner conserve les informations de connexion dans une ressource appeléeconnexion. App Runner nécessite une ressource de connexion lorsque vous créez un service qui nécessite des informations de connexion tierce. Voici quelques informations importantes sur les connexions :

- Fournisseurs— App Runner nécessite actuellement des ressources de connexion avecGitHub.
- Partagé— Vous pouvez utiliser une ressource de connexion pour créer plusieurs services App Runner qui utilisent le même compte de fournisseur de référentiel.
- Gestion des ressources— Dans App Runner, vous pouvez créer et supprimer des connexions. Toutefois, vous ne pouvez pas modifier une connexion existante.

- Continota de ressources— Les ressources de connexion ont un quota défini associé à votreCompte AWSDans chaqueRégion AWS. Si vous atteignez ce quota, vous devrez peut-être supprimer une connexion avant de pouvoir vous connecter à un nouveau compte fournisseur. Vous pouvez supprimer une connexion avec App Runnerconsoleou API. Pour plus d'informations, consultez the section called “Quotas de ressources App Exécuteur”.

Gérer les connexions à l'aide de la console App Runner

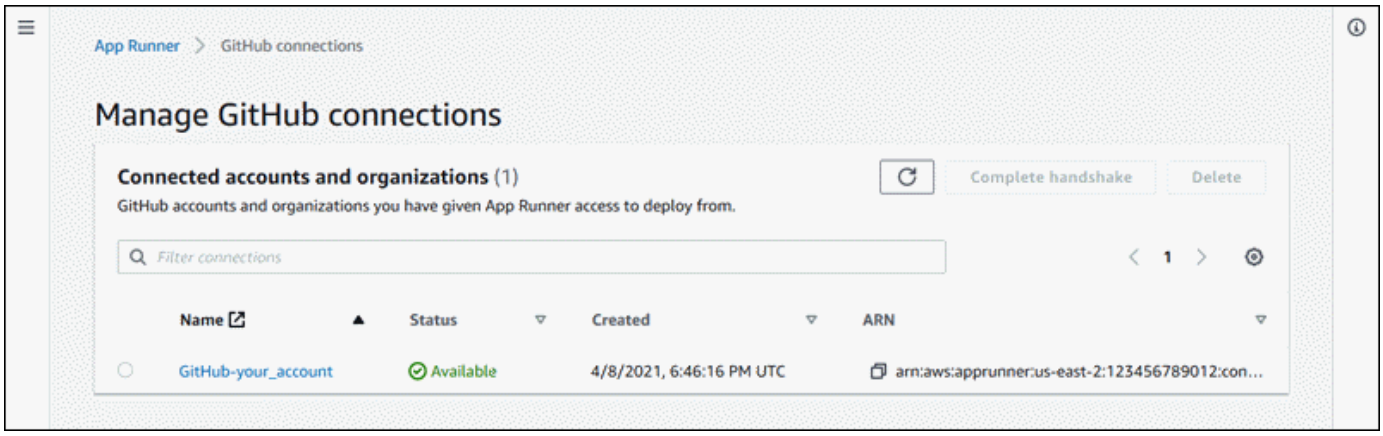
Lorsque vous utilisez la console App Runner pourCréation d'un service, vous fournissez les détails de la connexion. Vous n'avez pas besoin de créer explicitement une ressource de connexion. Dans la console, vous pouvez choisir de vous connecter à un compte GitHub auquel vous vous êtes déjà connecté, ou de vous connecter à un nouveau compte. Si nécessaire, App Runner crée une ressource de connexion pour vous. Pour une nouvelle connexion, certains fournisseurs (par exemple, GitHub) exigent que vous remplissiez une poignée de main d'authentification avant de pouvoir utiliser la connexion. La console vous guide tout au long de ce processus.

La console dispose également d'une page pour gérer vos connexions existantes. Vous pouvez terminer la poignée de main d'authentification pour une connexion si vous ne l'avez pas fait lors de la création de votre service. Vous pouvez également supprimer les connexions que vous n'utilisez plus. La procédure suivante décrit la gestion des connexions GitHub.

Pour gérer les connexions GitHub dans votre compte

1. Ouverture d'console App Runner, et dans leRégions, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezConnexions GitHub.

   La console affiche ensuite une liste des connexions GitHub dans votre compte.



3. Vous pouvez maintenant effectuer l'une des opérations suivantes avec n'importe quelle connexion de la liste :

   - Ouvrir un compte ou une organisation GitHub— Choisissez le nom de la connexion.
   - Liaison d'authentification complète— Sélectionnez la connexion, puis choisissezLiaison complète. La console vous guide tout au long du processus d'authentification.
   - Supprimer une connexion— Sélectionnez la connexion, puis choisissezSupprimer. Suivez les instructions de l'invite de suppression.

Gérez les connexions à l'aide de l'API App Runner ouAWS CLI

Vous pouvez utiliser les actions d'API App Runner suivantes pour gérer vos connexions.

- CreateConnection— Crée une connexion à un compte fournisseur de référentiel. Une fois la connexion créée, vous devez effectuer manuellement la poignée de main d'authentification à l'aide de la console App Runner. Ce processus est expliqué dans la section précédente.
- ListConnexions— Retourne la liste des connexions App Runner associées à votreCompte AWS.
- DeleteConnection— Supprime une connexion. Vous devrez peut-être supprimer les connexions inutiles si vous atteignez le quota de connexion pour votreCompte AWS.

Gestion de la mise à l'échelle automatique App Runner

AWS App Runnermet automatiquement à l'échelle les ressources de calcul (instances) vers le haut ou vers le bas pour votre application App Runner. La mise à l'échelle automatique fournit une gestion adéquate des demandes lorsque le trafic entrant est élevé et réduit votre coût lorsque le trafic ralentit. Vous pouvez configurer quelques paramètres pour ajuster le comportement de mise à l'échelle automatique de votre service.

App Runner conserve les paramètres de mise à l'échelle automatique dans une ressource appeléeAutoScalingConfiguration. Vous pouvez fournir une ressource de configuration de mise à l'échelle automatique lorsque vous créez ou mettez à jour un service. La console App Runner en crée un pour vous lorsque vous créez un nouveau service App Runner. La fourniture d'une configuration de mise à l'échelle automatique est facultative. Si vous n'en fournissez pas, App Runner fournit une configuration de mise à l'échelle automatique par défaut avec les valeurs recommandées.

Une configuration de mise à l'échelle automatique a unnomet un numéro numériquerévision. Plusieurs révisions d'une configuration ont le même nom et des numéros de révision différents.

Vous pouvez utiliser différents noms de configuration pour différents scénarios de mise à l'échelle automatique, tels que haute disponibilité ou faible coût. Pour chaque nom, vous pouvez ajouter plusieurs révisions pour affiner les paramètres d'un scénario spécifique.

Voici quelques informations importantes sur les configurations de mise à l'échelle automatique :

- Paramètres— Procédure à suivre :
  - Simultanéité max.— Nombre maximal de demandes simultanées qu'une instance traite. Lorsque le nombre de demandes simultanées dépasse ce quota, App Runner augmente le service.
  - Taille max.— Nombre maximal d'instances que votre service peut mettre à l'échelle. Au plus, ce nombre d'instances acheminent activement le trafic pour votre service.
  - Taille Min— Nombre minimal d'instances que App Runner fournit pour votre service. Le service a toujours au moins ce nombre d'instances provisionnées. Certains d'entre eux servent activement le trafic. Le reste d'entre eux (instances provisionnées et inactives) est une réserve de capacité de calcul rentable, prête à être activée rapidement. Vous payez pour l'utilisation de la mémoire de toutes les instances provisionnées. Vous payez pour l'utilisation du processeur uniquement du sous-ensemble actif.

    App Runner double temporairement le nombre d'instances provisionnées pendant les déploiements, afin de conserver la même capacité pour l'ancien et le nouveau code.
- Révision— La première configuration que vous créez avec un nom obtient le numéro de révision 1. Les configurations suivantes portant le même nom obtiennent des numéros de révision consécutifs (commençant par 2). Vous pouvez associer votre service App Runner à une révision de configuration de mise à l'échelle automatique spécifique ou à la dernière révision de configuration.
- Partagé : vous pouvez partager une seule ressource de configuration de mise à l'échelle automatique sur plusieurs services App Runner. Ceci est utile s'ils ont des exigences de mise à l'échelle similaires. En particulier, vous pouvez configurer plusieurs services pour qu'ils utilisent tous la dernière version d'une configuration en spécifiant le nom de la configuration mais en ne spécifiant pas de révision. Ce faisant, tous les services que vous avez configurés de cette manière reçoivent les mises à jour de configuration de mise à l'échelle automatique lorsque vous mettez à jour le service. Pour plus d'informations sur les modifications de configuration, consultezthe section called “Configuration”.
- Gestion des ressources— Vous pouvez utiliser App Runner pour créer et supprimer des configurations de mise à l'échelle automatique. Vous ne pouvez pas mettre à jour directement une configuration. Au lieu de cela, vous pouvez créer une nouvelle révision d'un nom de configuration existant pour mettre à jour efficacement la configuration.

> **Note**
>
> Pour le moment, vous ne pouvez créer une configuration qu'avec une seule révision dans la console App Runner. Pour créer d'autres révisions et supprimer des configurations, utilisez App RunnerAPI.

- Quota de ressources— Il existe des quotas définis pour le nombre de noms de configuration uniques et de révisions que vous pouvez avoir pour vos ressources de configuration de mise à l'échelle automatique dans chaqueRégion AWS. Si vous atteignez ces quotas, vous devez supprimer un nom de configuration ou au moins certaines de ses révisions avant de pouvoir en créer d'autres. Utiliser l'App RunnerAPIpour les supprimer. Pour plus d'informations, consultez the section called “Quotas de ressources App Exécuteur”.

Gérer la mise à l'échelle automatique avec la console App Runner

Lorsque vousCréation d'un servicedans la console App Runner, vous pouvez utiliser la configuration de mise à l'échelle automatique par défaut ou une configuration personnalisée. Pour utiliser une configuration personnalisée, choisissez une configuration existante ou indiquez un nouveau nom et des paramètres. S'il s'agit d'une nouvelle configuration, App Runner crée une nouvelle ressource de configuration de mise à l'échelle automatique pour vous, puis l'associe à votre nouveau service.

Gérez la mise à l'échelle automatique à l'aide de l'API App Runner ouAWS CLI

Vous pouvez utiliser des actions d'API App Runner suivantes pour gérer vos configurations de mise à l'échelle automatique.

- CreateAutoScalingConfiguration— Crée une nouvelle configuration de mise à l'échelle automatique ou une révision d'une configuration existante.
- ListAutoScalingConfigurations— Retourne la liste des configurations de mise à l'échelle automatique associées à votreCompte AWS, avec des informations récapitulatives.
- DescribeAutoScalingConfiguration— Retourne une description complète d'une configuration de mise à l'échelle automatique.
- DeleteAutoScalingConfiguration— Supprime une configuration de mise à l'échelle automatique. Vous pouvez supprimer une révision spécifique ou la dernière révision active. Vous devrez peut-

être supprimer les configurations de mise à l'échelle automatique inutiles si vous atteignez le quota de configuration de mise à l'échelle automatique pour votreCompte AWS.

Gestion des noms de domaine personnalisés pour un service App Runner

Lorsque vous créez un nomAWS App Runner, App Runner lui attribue un nom de domaine. Il s'agit d'un sous-domaine dans le`awsapprunner.com`appartenant à App Runner. Il peut être utilisé pour accéder à l'application Web qui s'exécute dans votre service.

Si vous possédez un nom de domaine, vous pouvez l'associer à votre service App Runner. Une fois que App Runner a validé votre nouveau domaine, il peut être utilisé pour accéder à votre application en plus du domaine App Runner. Vous pouvez associer jusqu'à cinq domaines personnalisés.

> **Note**
>
> Si vous le souhaitez, vous pouvez inclure l'option`www`sous-domaine de votre domaine. Cependant, cela est actuellement uniquement pris en charge dans l'API. La console d'App Runner ne prend pas en charge cette option.

Lorsque vous associez un domaine personnalisé à votre service, App Runner vous fournit un ensemble d'enregistrements de validation de certificat. Ajoutez-les à votre système de noms de domaine (DNS) afin que App Runner puisse valider que vous possédez ou contrôlez le domaine. En outre, ajoutez les enregistrements CNAME ou ALIAS à votre DNS pour cibler le domaine App Runner. Vous devez ajouter un enregistrement pour le domaine personnalisé et un autre pour le`www`, si vous avez choisi cette option. Ensuite, attendez que l'état du domaine personnalisé devienneActifdans la console App Runner. Cela prend généralement plusieurs minutes (mais peut prendre 24 à 48 heures). À ce stade, votre domaine personnalisé est validé et App Runner commence à acheminer le trafic de ce domaine vers votre application Web.

Vous pouvez spécifier un domaine à associer à votre service App Runner de la manière suivante :

- Un domaine racine— Par exemple, les ventes`example.com`. Vous pouvez éventuellement associer`www.example.com`Dans le cadre de la même opération.
- Sous-domaine— Par exemple, les ventes`login.example.com`ou`admin.login.example.com`. Vous pouvez éventuellement associer le`www`Dans le cadre de la même opération.

- Caractère générique— Par exemple, les ventes`*.example.com`. Vous ne pouvez pas utiliser l'option`www`Dans ce cas. Vous pouvez spécifier un caractère générique uniquement en tant que sous-domaine immédiat d'un domaine racine, et uniquement seul (ce ne sont pas des spécifications valides :`login*.example.com`,`*.login.example.com`). Cette spécification générique associe tous les sous-domaines immédiats et n'associe pas le domaine racine lui-même (le domaine racine devrait être associé dans une opération distincte).

Une association de domaine plus spécifique remplace une association moins spécifique. Par exemple,`login.example.com`Remplacements`*.example.com`. Le certificat et CNAME de l'association plus spécifique sont utilisés.

L'exemple suivant montre comment utiliser plusieurs associations de domaine personnalisées :

1. Associer`example.com`avec la page d'accueil de votre service. Activer la gem`www`d'associer également`www.example.com`.
2. Associer`login.example.com`avec la page de connexion de votre service.
3. Associer`*.example.com`avec une page personnalisée « introuvable ».

Vous pouvez dissocier (dissocier) un domaine personnalisé de votre service App Runner. Lorsque vous dissociez un domaine, App Runner arrête de router le trafic de ce domaine vers votre application Web. Vous devez supprimer les enregistrements de ce domaine de votre DNS.

App Runner crée en interne des certificats qui suivent la validité du domaine. Ils sont stockés dansAWS Certificate Manager(ACM). App Runner ne supprime pas ces certificats pendant sept jours après qu'un domaine est dissocié de votre service ou après la suppression du service.

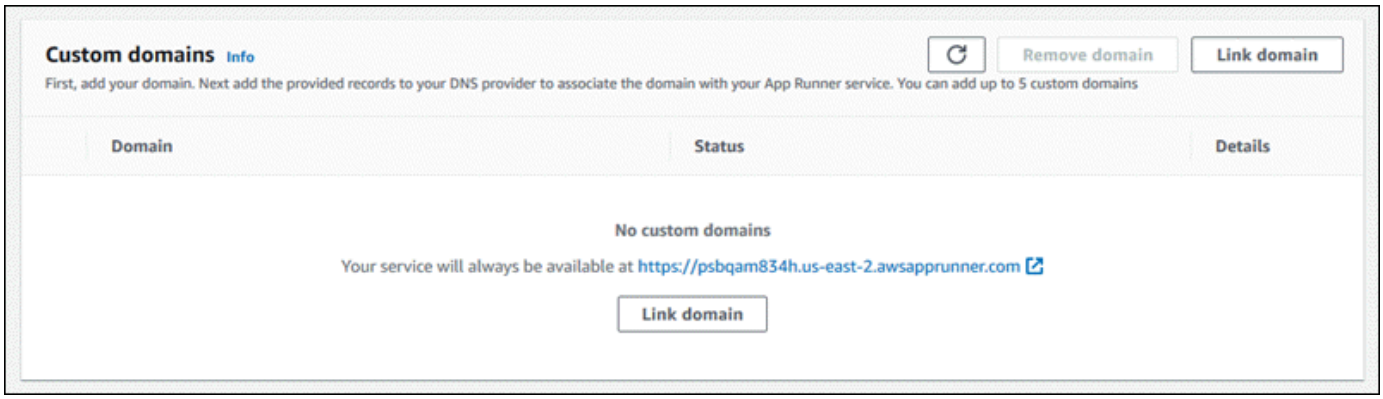
Gérer les domaines personnalisés à l'aide de la console App Runner

Pour associer (lier) un domaine personnalisé à l'aide de la console App Runner

1. Ouverture d'Console d'exécution d'applications, et dans leRégions, sélectionnez votre intentionRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. Dans la page du tableau de bord des services, choisissez l'optionDomaines personnalisésOnglet.

La console affiche les domaines personnalisés associés à votre service ouAucun domaine personnalisé.



4. Dans la pageDomaines personnalisés, choisissezLiaison du domaine.
5. DansLiaison du domaine personnalisé, entrez un nom de domaine, puis choisissezAfficher la configuration DNS.
6. Suivez les instructions sur la sectionConfigurer DNSpour démarrer le processus de validation du domaine.
7. Lorsque l'état du domaine passe àActif, vérifiez que le domaine fonctionne pour le routage du trafic en y naviguant.

Pour dissocier (dissocier) un domaine personnalisé à l'aide de la console App Runner

1. Dans la pageDomaines personnalisés, sélectionnez la vignette du domaine que vous souhaitez dissocier, puis choisissezUnlink du domaine.
2. DansUnlink du domaine, vérifiez l'action en sélectionnantUnlink du domaine.

Gérer les domaines personnalisés à l'aide de l'API App Runner ouAWS CLI

Pour associer un domaine personnalisé à votre service à l'aide de l'API App Runner ouAWS CLI, appelez l'intentionAssociateCustomDomainAction d'API. Lorsque l'appel réussit, il renvoie unCustomDomainqui décrit le domaine personnalisé associé à votre service. L'objet doit afficher un état de`CREATING`, et contient une liste des produitsCertificateValidationRecordobjets. Ce sont des enregistrements que vous pouvez ajouter à votre DNS.

Pour dissocier un domaine personnalisé de votre service à l'aide de l'API App Runner ouAWS CLI, appelez l'intentionDisassociateCustomDomainAction d'API. Lorsque l'appel réussit, il renvoie

unCustomDomainqui décrit le domaine personnalisé qui est en cours de dissociation de votre service. L'objet doit afficher un état de`DELETING`.

Mise en pause et reprise d'un service App Runner

Si vous devez désactiver temporairement votre application Web et arrêter l'exécution du code, vous pouvez suspendre votreAWS App Runnerservice. App Runner réduit la capacité de calcul du service à zéro.

Lorsque vous êtes prêt à exécuter à nouveau votre application, vous pouvez reprendre votre service App Runner. App Runner fournit une nouvelle capacité de calcul, y déploie votre application et exécute l'application. Votre source d'application n'est pas redéployée et aucune génération n'est nécessaire. Au contraire, App Runner reprend avec votre version actuellement déployée. Votre application conserve son domaine App Runner.

⚠ Important

- Lorsque vous mettez votre service en pause, votre application perd son état. Par exemple, tout stockage éphémère utilisé par votre code est perdu. Pour votre code, la mise en pause et la reprise de votre service équivaut à un déploiement vers un nouveau service.
- Si vous suspendez un service en raison d'un défaut dans votre code (par exemple, un bug découvert ou un problème de sécurité), vous ne pouvez pas déployer une nouvelle version avant de reprendre le service.

  Par conséquent, nous vous recommandons de garder le service en cours d'exécution et de revenir à votre dernière version d'application stable à la place.
- Lorsque vous reprenez votre service, App Runner déploie la dernière version d'application utilisée avant la mise en pause du service. Si vous avez ajouté de nouvelles versions source depuis la mise en pause de votre service, App Runner ne les déploie pas automatiquement, même si le déploiement automatique est sélectionné. Par exemple, supposons que vous avez de nouvelles versions d'image dans le référentiel d'images ou de nouvelles validations dans le référentiel de code. Ces versions ne sont pas automatiquement déployées.

  Pour déployer une version plus récente, effectuez un déploiement manuel ou ajoutez une autre version à votre référentiel source après la reprise de votre service App Runner.

Mise en pause et suppression comparées

Pausevotre service App Runner àtemporairementDésactivez-le. Seules les ressources de calcul sont terminées et vos données stockées (par exemple, l'image de conteneur avec la version de votre application) restent intactes. Reprise rapide de votre service : votre application est prête à être déployée sur de nouvelles ressources de calcul. Votre domaine App Runner reste le même.

Supprimervotre service App Runner àEn permanenceSupprimez. Vos données stockées sont supprimées. Si vous avez besoin de recréer le service, App Runner doit récupérer votre source à nouveau, et aussi la construire s'il s'agit d'un référentiel de code. Votre application Web obtient un nouveau domaine App Runner.

Lorsque votre service est mis en pause

Lorsque vous mettez votre service en pause et qu'il se trouve dans lePaused, il répond différemment aux demandes d'action, y compris les appels d'API ou les opérations de console. Lorsqu'un service est mis en pause, vous pouvez toujours effectuer des actions App Runner qui ne modifient pas la définition ou la configuration du service d'une manière qui affecte son exécution. En d'autres termes, si une action modifie le comportement, l'échelle ou d'autres caractéristiques d'un service en cours d'exécution, vous ne pouvez pas effectuer cette action sur un service en pause.

Les listes suivantes fournissent des informations sur les actions d'API que vous pouvez ou ne pouvez pas effectuer sur un service en pause. Les opérations équivalentes de la console sont également autorisées ou refusées.

Actions que vouscaneffectuer sur un service en pause

- *`List*`*and*`Describe*`*Actions— Actions qui ne lisent que les informations.
- *`DeleteService`*— Vous pouvez toujours supprimer un service.
- *`TagResource`*,*`UntagResource`*— Les balises sont associées à un service, mais ne font pas partie de sa définition et n'affectent pas son comportement d'exécution.

Actions que vouscannoteffectuer sur un service en pause

- *`StartDeployment`*Actions(ou unDéploiement manuelUtilisation de la console)
- *`UpdateService`*(ou une modification de configuration à l'aide de la console, à l'exception des modifications de balisage)

- *CreateCustomDomainAssociations*, *DeleteCustomDomainAssociations*
- *CreateConnection*, *DeleteConnection*

Suspendre et reprendre votre service à l'aide de la console App Runner

Pour interrompre votre service à l'aide de la console App Runner

1. Ouverture d'Configurer dans l'application Exécuteur, et dans leRégions, sélectionnez votre intention.Région AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. ChoisissezActions, puis.Pause.

   Sur la page du tableau de bord du service, le serviceÉtatModifications apportées àOpération en cours, puis passe àPaused. Votre service est maintenant suspendu.

Pour reprendre votre service à l'aide de la console App Runner

1. ChoisissezActions, puis.Resume.

   Sur la page du tableau de bord du service, le serviceÉtatModifications apportées àOpération en cours.
2. Attendez que le service reprenne. Sur la page du tableau de bord du service, le serviceÉtatchange de nouveau àEn cours d'exécution.
3. Pour vérifier que la reprise du service est réussie, sur la page du tableau de bord du service, sélectionnez l'optionExécuteur d'applicationsValeur . C'est l'URL du site Web de votre service. Vérifiez que votre application Web s'exécute correctement.

Suspendre et reprendre votre service à l'aide de l'API App Runner ouAWS CLI

Pour suspendre votre service à l'aide de l'API App Runner ouAWS CLI, appelez l'intentionPauseServiceAction d'API. Si l'appel renvoie une réponse réussie avec unServiceobjet affichage`"Status": "OPERATION_IN_PROGRESS"`, App Runner commence à interrompre votre service.

Pour reprendre votre service à l'aide de l'API App Runner ouAWS CLI, appelez l'intentionResumeServiceAction d'API. Si l'appel renvoie une réponse réussie avec unServiceobjet affichage`"Status": "OPERATION_IN_PROGRESS"`, App Runner commence à reprendre votre service.

Suppression d'un service App Runner

Lorsque vous souhaitez mettre fin à l'application Web qui s'exécute dans votreAWS App RunnerVous pouvez supprimer le service. La suppression d'un service arrête le service Web en cours d'exécution, supprime les ressources sous-jacentes et supprime les données associées.

Vous pouvez supprimer un service App Runner pour une ou plusieurs des raisons suivantes :

- Vous n'avez plus besoin de l'application web— Par exemple, il est retiré ou il s'agit d'une version de développement que vous avez fini d'utiliser.
- Vous avez terminé le quota de service d'App Runner— Vous souhaitez créer un nouveau service dans le mêmeRégion AWSet que vous avez atteint le quota associé à votre compte. Pour plus d'informations, consultez the section called “Quotas de ressources App Exécuteur”.
- Considérations de sécurité ou de confidentialité— Vous souhaitez que App Runner supprime les données qu'il stocke pour votre service.

En cours ou en supprimant

Pausevotre service App Runner àtemporairementDésactivez-le. Seules les ressources de calcul sont terminées et vos données stockées (par exemple, l'image de conteneur avec la version de votre application) restent intactes. Reprise rapide de votre service : votre application est prête à être déployée sur de nouvelles ressources de calcul. Votre domaine App Runner reste le même.

Supprimervotre service App Runner àEn permanenceSupprimez-le. Vos données stockées sont supprimées. Si vous devez recréer le service, App Runner doit récupérer votre source à nouveau, et aussi la construire s'il s'agit d'un référentiel de code. Votre application Web obtient un nouveau domaine App Runner.

Que supprime App Runner ?

Lorsque vous supprimez votre service, App Runner supprime certains éléments associés et n'en supprime pas d'autres. Les listes suivantes fournissent les détails.

Éléments que App Runner supprime :

- Image de conteneur— Copie de l'image que vous avez déployée ou de l'image créée par App Runner à partir de votre code source. Il est stocké dans Amazon Elastic Container Registry (Amazon ECR) en utilisantComptes AWSappartenant à App Runner.
- Configuration du service— Paramètres de configuration associés à votre service App Runner. Ils sont stockés dans Amazon DynamoDB en utilisantComptes AWSappartenant à App Runner.

Éléments que App Runner ne supprime pas :

- Connexion— Il se peut que vous ayez une connexion associée à votre service. Une connexion App Runner est une ressource distincte qui peut être partagée entre plusieurs services App Runner. Si vous n'avez plus besoin de la connexion, vous pouvez la supprimer explicitement. Pour plus d'informations, consultez the section called “Connexions”.
- Certificats de domaine personnalisés— Si vous liez des domaines personnalisés à un service App Runner, App Runner crée en interne des certificats qui suivent la validité du domaine. Ils sont stockés dansAWS Certificate Manager(ACM). App Runner ne supprime pas le certificat pendant sept jours après qu'un domaine est dissocié de votre service ou après la suppression du service. Pour plus d'informations, consultez the section called “Noms de domaine personnalisés”.

Supprimer votre service à l'aide de la console App Runner

Pour supprimer votre service à l'aide de la console App Runner

1. Ouverture d'Console d'application Runner, et dans leRégionsDans la liste, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. Choisissez Actions, puis choisissez Delete.

   La console vous fait accéder auServices. Le service supprimé affiche leOpération en cours, puis le service disparaît de la liste. Votre service est maintenant supprimé.

Supprimez votre service à l'aide de l'API App Runner ouAWS CLI

Pour supprimer votre service à l'aide de l'API App Runner ouAWS CLI, appelez l'intentionDeleteServiceAction d'API. Si l'appel renvoie une réponse réussie avec unServiceobjet affichage`"Status": "OPERATION_IN_PROGRESS"`, App Runner commence à supprimer votre service.

Journalisation et surveillance pour votre service App Runner

AWS App Runners'intègre à plusieursAWSpour vous fournir une suite complète d'outils de journalisation et de surveillance pour votre service App Runner. Rubriques de ce chapitre décrivent ces fonctionnalités.

Rubriques

Suivi de l'activité de service App Runner

AWS App Runnerutilise une liste d'opérations pour suivre l'activité de votre service App Runner. Une opération représente un appel asynchrone à une action API, par exemple la création d'un service, la mise à jour d'une configuration et le déploiement d'un service. Les sections suivantes vous expliquent comment suivre l'activité dans la console App Runner et à l'aide de l'API.

Suivi de l'activité du service App Runner dans la console

La console App Runner affiche votre activité de service App Runner et offre d'autres moyens d'explorer les opérations.

Pour afficher l'activité de votre service

1. Ouverture d'console App Runner, et dans leRégionsDans la liste, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. Dans la page du tableau de bord du service, choisissez l'optionActivitéS'il n'est pas déjà choisi.

   La console affiche la liste des opérations.

4. Pour rechercher des opérations spécifiques, étendez la liste en entrant un terme de recherche. Vous pouvez rechercher n'importe quelle valeur apparaissant dans la table.
5. Choisissez une opération répertoriée pour afficher ou télécharger le journal associé.

Récupération des opérations de service App Runner à l'aide de l'API App Runner ouAWS CLI

La .ListOperationsL'action, étant donné l'Amazon Resource Name (ARN) d'un service App Runner, renvoie la liste des opérations qui se sont produites sur ce service. Chaque élément de liste contient un ID d'opération et quelques détails de suivi.

Affichage des journaux App Runner diffusés dans les CloudWatch Logs

Vous pouvez utiliser Amazon CloudWatch Logs pour surveiller, stocker et accéder à des fichiers journaux que vos ressources dans diversAWSles services génèrent. Pour de plus amples informations, veuillez consulterGuide de l'utilisateur Amazon CloudWatch Logs.

AWS App Runnercollecte la sortie de vos déploiements d'applications et de votre service actif et la transmet dans CloudWatch Logs. Les sections suivantes répertorient les flux de journaux App Runner et vous montrent comment les afficher dans la console App Runner.

Flux de journaux et groupes de journaux App Runner

CloudWatch Logs conserve les données de journal dans les flux de journaux qu'il organise en groupes de journaux. AFlux de journauxest une séquence d'événements de journaux depuis une source spécifique. Un groupe de journaux est un groupe de flux de journaux qui partagent les mêmes paramètres de conservation, de surveillance et de contrôle d'accès.

App Runner définit deux groupes de journaux CloudWatch Logs, chacun avec plusieurs flux de journaux, pour chacun de vos services App Runner dans votreCompte AWS.

Journaux de service

Le groupe de journaux de service contient la sortie de journalisation générée par App Runner lorsqu'il gère votre service App Runner et agit dessus.

log group name	Exemple
`/aws/apprunner/ service-name /service-id /service`	`/aws/apprunner/python-test/ ac7ec8b51ff34746bcb6654e0bcb23da/service`

Dans le groupe de journaux de service, App Runner crée un flux de journal des événements pour capturer l'activité dans le cycle de vie de votre service App Runner. Par exemple, il peut s'agir de lancer votre application ou de la suspendre.

En outre, App Runner crée un flux de journal pour chaque opération asynchrone longue durée liée à votre service. Le nom du flux de journal reflète le type d'opération et l'ID d'opération spécifique.

ADéploiementest un type d'opération. Les journaux de déploiement contiennent la sortie de journalisation des étapes de génération et de déploiement effectuées par App Runner lorsque vous créez un service ou déployez une nouvelle version de votre application. Les noms de flux du journal de déploiement commencent par`deployment/`et se terminent par l'ID de l'opération qui effectue le déploiement. Cette opération est soit unCreateServicepour le déploiement initial de l'application ou un appelStartDeploymentpour chaque déploiement ultérieur.

Dans un journal de déploiement, chaque message de journal commence par un préfixe :

- `[AppRunner]`— Sortie générée par App Runner pendant le déploiement.
- `[Build]`— Sortie de vos propres scripts de construction.

Nom du flux de journaux	Exemple
`events`	Non applicable (nom fixe)
`operation-type /operation-id`	`deployment/c2c8eeedea164f459cf78f12a8953390`

Journaux d'application

Le groupe de journaux d'application contient la sortie de votre code d'application en cours d'exécution.

log group name	Exemple
`/aws/apprunner/ service-name /service-id /application`	`/aws/apprunner/python-test/ ac7ec8b51ff34746bcb6654e0bcb23da/ application`

Dans le groupe de journaux d'application, App Runner crée un flux de journal pour chaque instance (unité de mise à l'échelle) exécutant votre application.

Nom du flux de journaux	Exemple
`instance/ instance-id`	`instance/1a80bc9134a84699b7 b3432ebeebb591`

Affichage des journaux App Runner dans la console

La console App Runner affiche un résumé de tous les journaux de votre service et vous permet de les afficher, d'explorer et de les télécharger.

Pour afficher les journaux de votre service

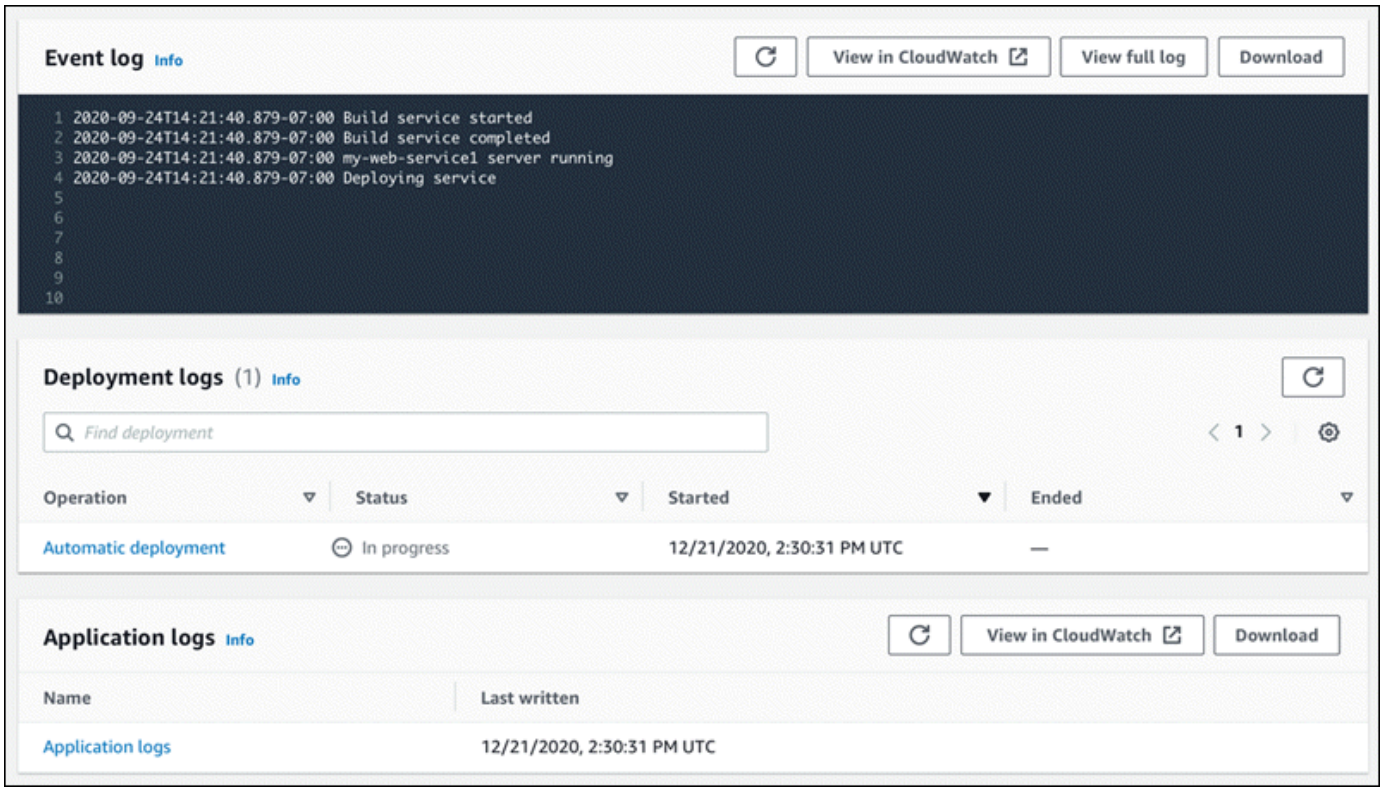
1. Ouverture d'console App Runner, et dans leRégionsDans la liste, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. Dans la page du tableau de bord des services, choisissez l'optionJournauxOnglet.

   La console affiche quelques types de journaux dans plusieurs sections :

   - Event Log (Journal des événements)— Activité dans le cycle de vie de votre service App Runner. La console affiche les événements les plus récents.
   - Journaux de déploiement— Déploiements de référentiel source vers votre service App Runner ; La console affiche un flux de journal distinct pour chaque déploiement.

- Journaux d'application— Sortie de l'application Web déployée sur votre service App Runner. La console combine la sortie de toutes les instances en cours d'exécution dans un flux de journaux unique.



4. Pour rechercher des déploiements spécifiques, étendez la liste du journal de déploiement en entrant un terme de recherche. Vous pouvez rechercher toutes les valeurs qui apparaissent dans la table.
5. Pour afficher le contenu d'un journal, choisissezAfficher le journal complet(journal des événements) ou le nom du flux de journal (journaux de déploiement et d'application).
6. ChoisissezTéléchargementpour télécharger un journal. Pour un flux de journal de déploiement, sélectionnez d'abord un flux de journaux.
7. ChoisissezAfficher dans CloudWatchpour ouvrir la console CloudWatch et utiliser toutes ses fonctionnalités pour explorer vos journaux de service App Runner. Pour un flux de journal de déploiement, sélectionnez d'abord un flux de journaux.

> **Note**
>
> La console CloudWatch est particulièrement utile si vous souhaitez afficher les journaux d'applications d'instances spécifiques au lieu du journal d'applications combiné.

Affichage des mesures de service App Runner signalées à CloudWatch

Amazon CloudWatch surveille vos Amazon Web Services (AWS) et les applications que vous exécutez surAWSEn temps réel. Vous pouvez utiliser CloudWatch pour recueillir et suivre les métriques, qui sont des variables que vous pouvez mesurer pour vos ressources et applications. Vous pouvez également l'utiliser pour créer des alarmes qui surveillent les métriques. Lorsqu'un certain seuil est atteint, CloudWatch envoie des notifications ou apporte automatiquement des modifications aux ressources surveillées. Pour de plus amples informations, veuillez consulterGuide de l'utilisateur Amazon CloudWatch..

AWS App Runnercollecte une variété de mesures qui vous offrent une meilleure visibilité sur l'utilisation, les performances et la disponibilité de vos services App Runner. Certaines mesures suivent les instances individuelles qui exécutent votre service Web, tandis que d'autres sont au niveau de service global. Les sections suivantes répertorient les métriques App Runner et vous expliquent comment les afficher dans la console App Runner.

Métriques App Runner

App Runner collecte les mesures suivantes relatives à votre service et les publie sur CloudWatch dans le`AWS/AppRunner`.

Métriques au niveau de l'instancesont collectées pour chaque instance (unité de mise à l'échelle) individuellement.

Qu'est-ce qui est mesuré ?	Métrique	Description
CPU utilization	`CPUUtilization`	Utilisation moyenne de l'UC pendant les périodes d'une minute.

Qu'est-ce qui est mesuré ?	Métrique	Description
Memory utilization	`MemoryUtilization`	Utilisation moyenne de la mémoire pendant les périodes d'une minute.

Métriques du niveau de servicesont collectées pour le service dans son intégralité.

Qu'est-ce qui est mesuré ?	Métriques	Description
HTTP request count	`Requests`	Le nombre de requêtes HTTP que le service a reçues.

HTTP status counts	`2xxStatusResponses`
	
	`4xxStatusResponses`
	
	`5xxStatusResponses`
		Le nombre de requêtes HTTP qui ont renvoyé chaque état de réponse, regroupées par catégorie (2XX, 4XX, 5XX).

HTTP request latency	`RequestLatency`	Temps nécessaire à votre service Web pour traiter les requêtes HTTP.
Instance counts	`ActiveInstances`	Le nombre d'instances qui traitent les requêtes HTTP pour votre service.

Affichage des métriques App Runner dans la console

La console App Runner affiche graphiquement les mesures collectées par App Runner pour votre service et fournit d'autres moyens de les explorer.

> **Note**
>
> Pour le moment, la console affiche uniquement les mesures de service. Pour afficher les mesures d'instance, utilisez la console CloudWatch.

Pour afficher les journaux de votre service

1. Ouverture d'Exécuteur d'applications, et dans leRégions, sélectionnez votreRégion AWS.
2. Dans le volet de navigation, choisissezServices, puis choisissez votre service App Runner.

   La console affiche le tableau de bord de service avec unPrésentation du service.
3. Dans la page du tableau de bord du service, choisissezMétriquesOnglet.

   La console affiche un ensemble de graphiques de mesures.
4. Choisissez une durée (par exemple,12h) pour définir les graphiques de métriques de la période récente de cette durée.
5. ChoisissezAjouter au tableau de borden haut de l'une des sections du graphique, ou utilisez le menu de n'importe quel graphique, pour ajouter les mesures pertinentes à un tableau de bord de la console CloudWatch pour plus d'informations.

Gestion des événements App Runner dans EventBridge

Avec Amazon EventBridge, vous pouvez configurer des règles basées sur des événements qui surveillent un flux de données en temps réel à partir de votreAWS App Runnerpour certains modèles. Lorsqu'un modèle d'une règle est mis en correspondance, EventBridge lance une action dans une cible telle queAWS Lambda, Amazon ECS,AWS Batchet Amazon SNS. Par exemple, vous pouvez définir une règle pour l'envoi de notifications par e-mail en signalant une rubrique Amazon SNS chaque fois qu'un déploiement vers votre service échoue. Vous pouvez également définir une fonction Lambda pour avertir un canal Slack chaque fois qu'une mise à jour de service échoue. Pour de plus amples informations sur EventBridge, veuillez consulter.Guide de l'utilisateur Amazon EventBridge.

App Runner envoie les types d'événements suivants à EventBridge

- Changement de statut de service— Modification de l'état d'un service App Runner. Par exemple, un statut de service a été modifié en`DELETE_FAILED`.

- Changement de statut de l'opération— Modification de l'état d'une longue opération asynchrone sur un service App Runner. Par exemple, un service a commencé à être créé, une mise à jour de service terminée avec succès ou un déploiement de service terminé avec des erreurs.

Création d'une règle EventBridge pour agir sur les événements App Runner

Un EventBridgeeventest un objet qui définit certains champs EventBridge standard, tels que la sourceAWSet le type de détail (événement), ainsi qu'un ensemble de champs spécifique à l'événement avec les détails de l'événement. Pour créer une règle EventBridge, utilisez la console EventBridge pour définir unModèle d'événement(quels événements doivent parier suivis) et spécifiez unaction ciblée(ce qui devrait être fait sur un match). Un modèle d'événement est similaire aux événements auxquels il correspond. Vous spécifiez un sous-ensemble de champs à mettre en correspondance, et pour chaque champ, vous spécifiez une liste de valeurs possibles. Cette rubrique fournit des exemples d'événements et de modèles d'événements App Runner.

Pour de plus amples informations sur la création de règles EventBridge, veuillez consulterCréation d'une règle pour unAWSServicedans leGuide de l'utilisateur Amazon EventBridge.

> **Note**
>
> Prise en charge de certains servicesModèles prédéfinisdans EventBridge. Cela simplifie la façon dont un modèle d'événement est créé. Vous sélectionnez des valeurs de champ dans un formulaire et EventBridge génère le modèle pour vous. Pour le moment, App Runner ne prend pas en charge les modèles prédéfinis. Vous devez entrer le motif en tant qu'objet JSON. Vous pouvez utiliser les exemples de cette rubrique comme point de départ.

Exemples d'événements App Runner

Voici quelques exemples d'événements que App Runner envoie à EventBridge.

- Événement de changement d'état de service. Plus précisément, un service qui a changé de`OPERATION_IN_PROGRESS`à la`RUNNING`État.

```
{
  "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"
  }
}
```

- Événement de changement d'état de l'opération. Plus précisément, un`UpdateService`opération terminée avec succès.

```
{
  "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"
  }
}
```

Exemples de modèle d'événement App Runner

Les exemples suivants illustrent les modèles d'événements que vous pouvez utiliser dans les règles EventBridge pour correspondre à un ou plusieurs événements App Runner. Un modèle d'événement est similaire à un événement. Incluez uniquement les champs que vous souhaitez faire correspondre et fournissez une liste au lieu d'un scalaire à chacun d'eux.

- Correspondre à tous les événements de changement d'état de service pour les services d'un compte spécifique, où le service n'est plus dans`RUNNING`État.

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

- Correspondre à tous les événements de changement d'état d'opération pour les services d'un compte spécifique, où l'opération a échoué.

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

Référence de l'événement App Runner

Changement de statut de service

Un événement de changement d'état de service peut avoir`detail-type`défini sur`Service Status Change`. Il peut avoir les champs de détail et les valeurs suivantes :

```
"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"
```

Changement de statut d'opération

Un événement de changement de statut d'opération peut avoir`detail-type`défini sur`Service Operation Status Change`. Il peut avoir les champs de détail et les valeurs suivantes :

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

Le tableau suivant répertorie tous les codes d'état possibles et les messages associés.

État	Message
`CreateServiceStarted`	Création de service démarrée.
`CreateServiceCompletedSuccessfully`	La création de service a abouti.
`CreateServiceFailed`	La création de service a échoué. Pour plus d'informations, consultez Journaux de service.
`DeleteServiceStarted`	La suppression du service a démarré.

État	Message
`DeleteServiceCompletedSuccessfully`	La suppression du service a abouti.
`DeleteServiceFailed`	Échec de la suppression du service.
`UpdateServiceStarted`	
`UpdateServiceCompletedSuccessfully`	La mise à jour du service a abouti. Une nouvelle application et une nouvelle configuration sont déployées.
	La mise à jour du service a abouti. Nouvelle configuration est déployée.
`UpdateServiceFailed`	Mise à jour du service échoué. Pour plus d'informations, consultez Journaux de service.
`DeploymentStarted`	Déploiement démarré.
`DeploymentCompletedSuccessfully`	Le déploiement a abouti.
`DeploymentFailed`	Le déploiement a échoué. Pour plus d'informations, consultez Journaux de service.
`PauseServiceStarted`	La pause du service démarrée.
`PauseServiceCompletedSuccessfully`	La pause du service a abouti.
`PauseServiceFailed`	La pause du service a échoué.
`ResumeServiceStarted`	Reprise du service démarrée.
`ResumeServiceCompletedSuccessfully`	La reprise du service a abouti.
`ResumeServiceFailed`	La reprise du service a échoué.

Journalisation des appels d'API App Runner avecAWS CloudTrail

App Runner est intégré àAWS CloudTrail, un service qui fournit un enregistrement des actions réalisées par un utilisateur, un rôle ou unAWSdans App Runner. CloudTrail capture tous les appels d'API pour App Runner en tant qu'événements. Les appels capturés incluent les appels depuis la console App Runner et les appels de code aux opérations de l'API App Runner. Si vous créez un journal de suivi, vous pouvez diffuser en continu les événements CloudTrail dans un compartiment Amazon S3, y compris les événements pour App Runner. Si vous ne configurez pas de journal de suivi, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Event history (Historique des événements). Avec les informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à App Runner, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur et la date de la demande, ainsi que d'autres détails.

Pour plus d'informations sur CloudTrail, consultezAWS CloudTrailGuide de l'utilisateur.

Informations sur App Runner dans CloudTrail

CloudTrail est activé sur votreCompte AWSLorsque vous créez le compte. Quand une activité a lieu dans App Runner, cette activité est enregistrée dans un événement CloudTrail avec d'autresAWSévénements de services dansHistorique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votreCompte AWS. Pour de plus amples informations, veuillez consulter Affichage des événements avec l'historique des événements CloudTrail.

Pour un registre permanent des événements dans votreCompte AWS, y compris les événements pour App Runner, créez un journal de suivi. Un journal de suivi permet à CloudTrail de livrer des fichiers journaux dans un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à tous lesRégions AWS. Le journal de suivi consigne les événements de toutes les régions dansAWSet livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autresAWSPour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour plus d'informations, consultez ce qui suit :

- Présentation de la création d'un journal de suivi
- Intégrations et services pris en charge par CloudTrail
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers journaux CloudTrail de plusieurs régions et Réception de fichiers journaux CloudTrail de plusieurs comptes

Toutes les actions App Runner sont enregistrées par CloudTrail et sont documentées dans leAWS App RunnerRéférence d'API. À titre d'exemple, les appels vers les actions `CreateService`, `DeleteConnection` et `StartDeployment` génèrent des entrées dans les fichiers journaux CloudTrail.

Chaque événement ou entrée du journal contient des informations sur la personne qui a généré la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification utilisateur racine ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour de plus amples informations, consultez l'élément userIdentity CloudTrail.

Présentation des entrées des fichiers journaux App Runner

Un journal de suivi est une configuration qui permet la livraison d'événements sous forme de fichiers journaux vers un compartiment Amazon S3 que vous spécifiez. Les fichiers journaux CloudTrail peuvent contenir une ou plusieurs entrées de journal. Un événement représente une demande individuelle émise à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure de l'action, ainsi que les paramètres de la demande. Les fichiers journaux CloudTrail ne constituent pas une série ordonnée retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de journal CloudTrail qui illustre l'action `CreateService`.

> **Note**
>
> Pour des raisons de sécurité, certaines valeurs de propriété sont caviardées dans les journaux et remplacées par le texte`HIDDEN_DUE_TO_SECURITY_REASONS`. Cela empêche l'exposition involontaire d'informations secrètes. Cependant, vous pouvez toujours voir que ces propriétés ont été transmises dans la requête ou renvoyées dans la réponse.

Exemple d'entrée de journal CloudTrail pour**`CreateService`**Action Exécuteur d'applications


```
{
  "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"
}
```

Définition des options de service App Runner à l'aide d'un fichier de configuration

> **Note**
>
> Les fichiers de configuration ne sont applicables qu'àservices basés sur le code source. Vous ne pouvez pas utiliser de fichiers de configuration avecservices basés sur l'image.

Lorsque vous créez unAWS App Runnerservice utilisant un référentiel de code source,AWS App Runnernécessite des informations sur la création et le démarrage de votre service. Vous pouvez fournir ces informations chaque fois que vous créez un service à l'aide de la console ou de l'API App Runner. Sinon, vous pouvez définir des options de service à l'aide d'unFichier de configuration. Les options que vous spécifiez dans un fichier font partie de votre référentiel source, et toutes les modifications apportées à ces options sont suivies de la même manière que le suivi des modifications apportées au code source. Vous pouvez utiliser le fichier de configuration App Runner pour spécifier plus d'options que la prise en charge par l'API. Vous n'avez pas besoin de fournir un fichier de configuration si vous avez seulement besoin des options de base prises en charge par l'API.

Le fichier de configuration App Runner est un fichier YAML nomméapprunner.yamlDans le répertoire racine du référentiel de votre application. Il fournit des options de construction et d'exécution pour votre service. Les valeurs de ce fichier indiquent à App Runner comment créer et démarrer votre service, et fournissent un contexte d'exécution tel que les paramètres réseau et les variables d'environnement.

Le fichier de configuration App Runner n'inclut pas les paramètres opérationnels, tels que le processeur et la mémoire.

Pour accéder à des exemples de fichiers de configuration d'App Runner, consultezthe section called “Exemples”. Pour obtenir un guide de référence complet, consultezthe section called “Référence”.

Rubriques

- Exemples de fichiers de configuration App Runner
- Référence du fichier de configuration App Runner

Exemples de fichiers de configuration App Runner

> **Note**
>
> Les fichiers de configuration ne sont applicables qu'àservices basés sur le code source. Vous ne pouvez pas utiliser de fichiers de configuration avecservices basés sur des images.

Les exemples suivants illustrentAWS App RunnerFichiers de configuration. Certains sont minimes et ne contiennent que les paramètres requis. D'autres sont terminées, y compris toutes les sections du fichier de configuration. Pour obtenir une présentation des fichiers de configuration App Runner, consultezFichier de configuration App Runner.

Exemples de fichiers de configuration

Fichier de configuration minimal

Avec un fichier de configuration minimal, App Runner fait les hypothèses suivantes :

- Aucune variable d'environnement personnalisée n'est nécessaire pendant la construction ou l'exécution.
- La dernière version d'exécution est utilisée.
- Le numéro de port par défaut et la variable d'environnement de port sont utilisés.

Example apprunner.yaml

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

Fichier de configuration complet

Cet exemple montre l'utilisation de toutes les clés de configuration avec un moteur d'exécution géré.

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"
```

Pour obtenir des exemples de fichiers de configuration d'exécution gérés spécifiques, consultez la sous-rubrique d'exécution spécifique sousService basé sur le code.

Référence du fichier de configuration App Runner

> **Note**
>
> Les fichiers de configuration ne sont applicables qu'àservices basés sur le code source. Vous ne pouvez pas utiliser de fichiers de configuration avecservices basés sur l'image.

Cette rubrique est un guide de référence complet sur la syntaxe et la sémantique d'unAWS App RunnerFichier de configuration. Pour obtenir une présentation des fichiers de configuration d'App Runner, consultezFichier de configuration App Runner.

Le fichier de configuration App Runner est un fichier YAML. Nommez-le`apprunner.yaml`et placez-le dans le répertoire racine du référentiel de votre application.

Présentation de la structure

Le fichier de configuration App Runner est un fichier YAML. Nommez-le`apprunner.yaml`et placez-le dans le répertoire racine du référentiel de votre application.

Le fichier de configuration App Runner contient les parties principales suivantes :

- Section supérieure— Contient des clés de niveau supérieur
- Section de génération— Configure l'étape de génération
- Section Exécuter— Configure le stage d'exécution

Section supérieure

Les clés en haut du fichier fournissent des informations générales sur le fichier et votre moteur d'exécution de service. Les clés suivantes sont disponibles :

- `version`–Obligatoire. Version du fichier de configuration App Runner. Idéalement, utilisez la version la plus récente.

  Syntaxe

  ```
  version: version
  ```

  Example

  ```
  version: 1.0
  ```

- `runtime`–Obligatoire. Nom du moteur d'exécution utilisé par votre application. Les runtimes suivants sont actuellement disponibles :

  `python3`, `nodejs12`

> **Note**
>
> La convention d'attribution de noms d'un moteur d'exécution géré est *<language-name><major-version>*.

Syntaxe

```
runtime: runtime-name
```

Example

```
runtime: python3
```

Section de génération

La section build configure l'étape de génération du déploiement du service App Runner. Vous pouvez spécifier des commandes de génération et des variables d'environnement. Les commandes de construction sont requises.

La section commence par le`build:`, et possède les sous-clés suivantes :

- `commands`–Obligatoire. Spécifie les commandes exécutées par App Runner au cours des différentes phases de construction. inclut les sous-clés suivantes :
  - `pre-build`–Facultatif. Commandes qu'App Runner exécute avant la génération. Par exemple, installeznpmdépendances ou bibliothèques de test.
  - `build`–Obligatoire. Les commandes exécutées par App Runner pour créer votre application. Par exemple, utilisezpipenv.
  - `post-build`–Facultatif. Commandes que App Runner exécute après la génération. Par exemple, utilisez Maven pour packager des artefacts de génération dans un fichier JAR ou WAR, ou exécutez un test.

Syntaxe

```
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–Facultatif. Spécifie les variables d'environnement personnalisées pour la phase de construction. Défini en tant que mappages scalaires nom-valeur. Vous pouvez faire référence à ces variables par leur nom dans vos commandes de génération.

  Syntaxe

```
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"
```

Section Exécuter

La section Exécuter configure l'étape d'exécution du conteneur du déploiement de l'application App Runner. Vous pouvez spécifier la version d'exécution, la commande de démarrage, le port réseau et les variables d'environnement.

La section commence par le`run:`, et possède les sous-clés suivantes :

- `runtime-version`–Facultatif. Spécifie une version d'exécution que vous souhaitez verrouiller pour votre service App Runner.

  Par défaut, seule la version majeure est verrouillée. App Runner utilise les dernières versions mineures et correctifs disponibles pour l'exécution sur chaque déploiement ou mise à jour de service. Si vous spécifiez des versions majeures et mineures, les deux deviennent verrouillées et App Runner ne met à jour que les versions de correctifs. Si vous spécifiez des versions majeures, mineures et correctifs, votre service est verrouillé sur une version d'exécution spécifique et App Runner ne la met jamais à jour.

  Syntaxe

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

  Example

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

- `command`–Obligatoire. Commande utilisée par App Runner pour exécuter votre application une fois la génération de l'application terminée.

  Syntaxe

  ```
  run:
    command: command
  ```

- `network`–Facultatif. Spécifie le port que votre application écoute. Elle inclut les éléments suivants :
  - `port`–Facultatif. Le cas échéant, il s'agit du numéro de port que votre application écoute. La valeur par défaut est `8080`.
  - `env`–Facultatif. Si cela est spécifié, App Runner transmet le numéro de port au conteneur dans cette variable d'environnement, en plus (et non au lieu de) transmettre le même numéro de port dans la variable d'environnement par défaut,`PORT`. En d'autres termes, si vous spécifiez`env`, App Runner transmet le numéro de port dans deux variables d'environnement.

  Syntaxe

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

  Example

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

- `env`–Facultatif. Définition de variables d'environnement personnalisées pour l'étape d'exécution. Défini en tant que mappages scalaires nom-valeur. Vous pouvez faire référence à ces variables par leur nom dans votre environnement d'exécution.

  Syntaxe

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

  Example

```
run:
  env:
```

```
    - name: MY_VAR_EXAMPLE
      value: "example"
```

L'API App Runner

La .AWS App RunnerL'interface de programmation d'application (API) est une API RESTful permettant de faire des requêtes au service App Runner. Vous devez utiliser l'API afin de créer, décrire, mettre à jour et supprimer les ressources App Runner dans votreCompte AWS.

Vous devez appeler l'API directement dans votre code d'application ou utiliser l'un desAWSKits SDK. Pour les scripts de ligne de commande, utilisez la commandeAWS CLIpour appeler le service App Runner. Pour plus d'informations surAWSles outils pour développeurs, consultezOutils sur lequel s'appuyerAWS.

Pour obtenir des informations de référence d'API complètes, consultez laAWS App RunnerAPI Reference.

Sécurité dans App Runner

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous-même. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud–AWSest responsable de la protection de l'infrastructure qui exécuteAWSservicesAWS Cloud.AWSVous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de conformité AWS. Pour de plus d'informations sur les programmes de conformité qui s'appliquent àAWS App RunnerVoir,AWSServices concernés par le programme de conformité.
- Sécurité dans le cloud— Votre responsabilité est déterminée par laAWSque vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise,et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'App Runner. Les rubriques suivantes vous montrent comment configurer App Runner pour qu'elle réponde à vos objectifs de sécurité et de conformité. Vous pouvez également apprendre à utiliser d'autresAWSqui vous aident à surveiller et à sécuriser vos ressources App Runner.

Rubriques

Protection des données dans App Runner

La .AWS Modèle de responsabilité partagéees'applique à la protection des données dansAWS App Runner. Comme il est décrit dans ce modèle,AWSest responsable de la protection de l'infrastructure mondiale qui exécute tous lesAWSLe nuage. La gestion du contrôle de votre contenu hébergé sur cette infrastructure est de votre responsabilité. Ce contenu comprend les tâches de configuration et de gestion de la sécurité pourAWSque vous utilisez. Pour de plus amples informations sur la confidentialité des données, veuillez consulter FAQ sur la confidentialité des données. Pour de plus amples informations sur la protection des données en Europe, veuillez consulterAWSModèle de responsabilité partagée et RGPDArticle de blog surAWSBillet de sécurité.

À des fins de protection des données, nous vous recommandons de protégerAWSet configurez des comptes d'utilisateur individuels avecAWS Identity and Access Management(IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multi-facteurs (MFA) avec chaque compte.
- Utilisez SSL/TLS pour communiquer avec des ressources AWS. Nous recommandons TLS 1.2 ou version ultérieure.
- Configurez l'API et la consignation des activités utilisateur avec AWS CloudTrail.
- Utilisez des solutions de chiffrement AWS, ainsi que tous les contrôles de sécurité par défaut au sein des services AWS.
- Utilisez des services de sécurité gérés avancés tels que Amazon Macie, qui contribuent à la découverte et à la sécurisation des données personnelles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés FIPS 140-2 lorsque vous accédez à AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour de plus amples informations sur les points de terminaison FIPS disponibles, consultez Federal Information Processing Standard (FIPS) 140-2.

Nous vous recommandons vivement de ne jamais placer d'informations identifiables sensibles, telles que les numéros de compte de vos clients, dans des champs de formulaire comme Name (Nom). Cela s'applique aussi lorsque vous utilisez App Runner ou d'autresAWSà l'aide de la console, de l'API,AWS CLI, ouAWSKits SDK. Toutes les données que vous entrez dans App Runner ou d'autres services peuvent être récupérées pour insertion dans les journaux de diagnostic. Lorsque vous fournissez une URL à un serveur externe, n'incluez pas les informations d'identification non chiffrées dans l'URL pour valider votre demande adressée au serveur.

Pour consulter d'autres rubriques sur la sécurité App Runner, consultezSécurité.

Rubriques

- Protection des données à l'aide du chiffrement
- Confidentialité du trafic inter-réseaux
- Utilisation d'App Runner avec des points de terminaison de VPC

Protection des données à l'aide du chiffrement

AWS App Runnerlit la source de votre application (image source ou code source) à partir d'un référentiel que vous spécifiez et la stocke pour déploiement sur votre service. Pour plus d'informations, consultez Architecture et concepts relatifs à.

La protection des données se réfère à la protection des donnéesEn transit(au fur et à mesure qu'il se déplace vers et depuis App Runner) etau repos(alors qu'il est stocké dansAWScentres de données).

Pour de plus amples informations sur la protection des données, veuillez consulterthe section called “Protection des données”.

Pour consulter d'autres rubriques sur la sécurité App Runner, consultezSécurité.

Chiffrement en transit

Vous pouvez obtenir la protection des données en transit de deux manières : chiffrer la connexion à l'aide du protocole TLS (Transport Layer Security) ou utiliser le chiffrement côté client (où l'objet est chiffré avant d'être envoyé). Les deux méthodes sont valides pour protéger les données de votre application. Pour sécuriser la connexion, chiffez-la à l'aide de TLS chaque fois que votre application, ses développeurs et administrateurs, ainsi que ses utilisateurs finaux, envoient ou reçoivent des objets. App Runner configure votre application pour recevoir du trafic via TLS.

Le chiffrement côté client n'est pas une méthode valide pour protéger l'image source ou le code que vous fournissez à App Runner pour déploiement. App Runner a besoin d'accéder à la source de votre application, de sorte qu'elle ne peut pas être chiffrée. Par conséquent, veillez à sécuriser la connexion entre votre environnement de développement ou de déploiement et App Runner.

Chiffrement au repos et gestion des clés

Pour protéger les données de votre application au repos, App Runner chiffre toutes les copies stockées de l'image source ou du groupe source de votre application. Lorsque vous créez un service

App Runner, vous pouvez fournir une clé principale client (CMK). Si vous en fournissez un, App Runner utilise la clé que vous avez fournie pour chiffrer votre source. Si vous n'indiquez pas, App Runner utilise unAWSClé CMK gérée à la place.

Pour plus d'informations sur les paramètres de création de service App Runner, consultezCreateService. Pour obtenir des informations surAWS Key Management Service(AWS KMS), consultezAWS Key Management ServiceManuel du développeur.

Confidentialité du trafic inter-réseaux

App Runner utilise Amazon Virtual Private Cloud (Amazon VPC) pour créer des limites entre les ressources de votre application App Runner et contrôler le trafic entre celles-ci, votre réseau sur site et Internet. Pour de plus amples informations sur la sécurité Amazon VPC, veuillez consulterConfidentialité du trafic inter-réseau dans Amazon VPCdans leGuide de l'utilisateur Amazon VPC.

Pour plus d'informations sur la sécurisation des demandes à App Runner à l'aide d'un point de terminaison de VPC,the section called “Points de terminaison d'un VPC”.

Pour de plus amples informations sur la protection des données, veuillez consulterthe section called “Protection des données”.

Pour consulter d'autres rubriques sur la sécurité App Runner, consultezSécurité.

Utilisation d'App Runner avec des points de terminaison de VPC

VosAWSpeut intégrerAWS App RunnerServices avec d'autresAWSen cours d'exécution dans unAmazon Virtual Private Cloud (Amazon VPC). Certaines parties de votre application peuvent faire des requêtes à App Runner depuis le VPC. Par exemple, vous pouvez utiliserAWS CodePipelinepour déployer en continu sur votre service App Runner. Une façon d'améliorer la sécurité de votre application consiste à envoyer ces requêtes App Runner (et les requêtes à d'autresAWS) sur un point de terminaison VPC.

Un point de terminaison de VPC permet une connexion privée entre votre VPC et les services AWS pris en charge ou les services de point de terminaison de VPC alimentés par AWS PrivateLink sans nécessiter une passerelle Internet, un périphérique NAT, une connexion VPN ou une connexion AWS Direct Connect.

Les ressources de votre VPC n'utilisent pas d'adresses IP publiques pour interagir avec les ressources App Runner. Le trafic entre votre VPC et App Runner ne quitte pas le réseau Amazon.

Pour de plus amples informations sur les points de terminaison de VPC, veuillez consulterPoints de terminaison d'un VPCdans leAWS PrivateLinkGuide.

Compatible avec App RunnerAWS PrivateLink, qui fournit une connectivité privée à App Runner et élimine l'exposition du trafic à l'internet public. Pour permettre à votre application d'envoyer des demandes à App Runner en utilisantAWS PrivateLink, vous configurez un type de point de terminaison VPC appeléPoint de terminaison d'un VPC d'interface(point de terminaison de l'interface). Pour de plus amples informations, veuillez consulterPoints de terminaison d'un VPC d'interface (AWS PrivateLink)dans leAWS PrivateLinkGuide.

> **Note**
>
> L'application Web de votre service App Runner s'exécute dans un VPC fourni et configuré par App Runner. Ce VPC est public, il est connecté à Internet public. App Runner ne prend pas en charge l'exécution de votre application dans un VPC privé ou la création d'un point de terminaison VPC pour celui-ci.

Configuration d'un point de terminaison de VPC pour App Runner

Pour créer le point de terminaison d'un VPC d'interface pour le service App Runner dans votre VPC, suivez lesCréation d'un point de terminaison d'interfacedans l'AWS PrivateLinkGuide. Pour Service Name (Nom du service), choisissez `com.amazonaws.`*`region`*`.apprunner`.

VPC de confidentialité réseau

> **Important**
>
> L'utilisation d'un point de terminaison VPC pour App Runner ne garantit pas que tout le trafic provenant de votre VPC reste hors d'Internet. Le VPC peut être public (connecté à Internet), et certaines parties de votre solution peuvent ne pas utiliser les points de terminaison VPC pour rendreAWSAppels d'API. Par exemple,AWSpeuvent appeler d'autres services à l'aide de leurs points de terminaison publics. Si la confidentialité du trafic est requise pour la solution dans votre VPC, lisez cette section.

Pour garantir la confidentialité du trafic réseau dans votre VPC, prenez en considération les éléments suivants :

- Activer le nom DNS— Certaines parties de votre application peuvent toujours envoyer des demandes à App Runner via Internet à l'aide de la`apprunner.region.amazonaws.com`Point de terminaison public. Si votre VPC est configuré avec un accès Internet public, ces demandes réussissent sans aucune indication pour vous. Pour empêcher cela, assurez-vous que Enable DNS name (Activer le nom DNS) est activé lors de la création du point de terminaison (true par défaut). Cela ajoute une entrée DNS dans votre VPC qui mappe le point de terminaison du service public au point de terminaison de VPC d'interface.
- Configuration des points de terminaison de VPC pour des services supplémentaires— Votre solution peut envoyer des requêtes à d'autresAWSServices . Par exemple,AWS CodePipelinepeut envoyer des requêtes àAWS CodeBuild. Configurez des points de terminaison de VPC pour ces services et activez les noms de DNS sur ces points de terminaison.

Utilisation des stratégies de point de terminaison pour contrôler l'accès avec des points de terminaison de VPC

Par défaut, un point de terminaison de VPC permet un accès complet au service auquel il est associé. Lorsque vous créez ou modifiez un point de terminaison de VPC pour App Runner, vous pouvez attacher unStratégie de point de terminaisonà elle.

Une stratégie de point de terminaison est unAWS Identity and Access Management(IAM) qui contrôle l'accès du point de terminaison au service spécifié. La stratégie de point de terminaison est spécifique au point de terminaison. Elle est distincte des stratégies IAM d'utilisateur ou d'instance que votre environnement peut avoir et ne les remplace pas. Pour plus d'informations sur la création et l'utilisation des stratégies de point de terminaison de VPC, consultezContrôle de l'accès aux services avec les points de terminaison d'un VPCdans leAWS PrivateLinkGuide.

L'exemple suivant autorise l'accès en lecture seule à partir du point de terminaison VPC vers App Runner. Il s'agit de droits d'accès minimaux lors de l'utilisation du point de terminaison VPC. Les entités principales peuvent disposer d'autorisations supplémentaires via d'autres stratégies.

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

```
    }
  ]
}
```

Identity and Access Management pour App Runner

AWS Identity and Access Management(IAM) est unAWSqui aide un administrateur à contrôler en toute sécurité l'accès àAWSAWS. Les administrateurs IAM contrôlent qui peut êtreAuthentification(connecté) etAuthorized(disposent d'autorisations) pour utiliser les ressources App Runner. IAM est unAWSService que vous pouvez utiliser sans frais supplémentaires.

Pour consulter d'autres rubriques de sécurité App Runner, consultezSécurité.

Rubriques

- Audience
- Authentification avec des identités
- Gestion de l'accès à l'aide de stratégies
- Fonctionnement d'App Runner avec IAM
- Exemples de stratégies basées sur l'identité d'App Runner
- Utilisation de rôles liés à un service pour App Runner
- Stratégies gérées AWS pour AWS App Runner
- Résolution des problèmes liés à Identity and Access

Audience

UtilisationAWS Identity and Access Management(IAM) diffère selon le travail que vous effectuez dans App Runner.

Utilisateur du service- Si vous utilisez le service App Runner pour effectuer votre tâche, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Plus vous utiliserez de fonctionnalités App Runner pour effectuer votre travail, plus vous pourrez avoir besoin d'autorisations supplémentaires. En comprenant la gestion des accès, vous n’aurez aucun mal à demander les bonnes autorisations à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans App Runner, consultezRésolution des problèmes liés à Identity and Access.

Administrateur de service- Si vous êtes le responsable des ressources App Runner dans votre entreprise, vous bénéficiez probablement d'un accès total à App Runner. C'est à vous de déterminer

les fonctionnalités et les ressources d'App Runner auxquelles vos employés pourront accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec App Runner, veuillez consulterFonctionnement d'App Runner avec IAM.

Administrateur IAM- Si vous êtes un administrateur IAM, vous souhaiterez peut-être obtenir des détails sur la façon dont vous pouvez écrire des stratégies pour gérer l'accès à App Runner. Pour voir des exemples de stratégies basées sur l'identité App Runner que vous pouvez utiliser dans IAM, veuillez consulterExemples de stratégies basées sur l'identité d'App Runner.

Authentification avec des identités

L'authentification correspond au processus par lequel vous vous connectez à AWS via vos informations d'identification. Pour plus d'informations sur la connexion à l'aide duAWS Management Console, voirConnectez-vous à la consoleAWS Management Consoleen tant qu'utilisateur IAM ou utilisateur racinedans leGuide de l'utilisateur IAM.

Vous devez êtreAuthentification(connecté àAWS) en tant queAWSUn utilisateur racine de compte, un utilisateur IAM ou en endossant un rôle IAM. Vous pouvez également utiliser l'authentification de connexion unique de votre entreprise ou vous connecter par le biais de Google ou de Facebook. Dans ces cas, votre administrateur aura précédemment configuré une fédération d’identités avec des rôles IAM. Lorsque vous accédez à AWS avec des informations d’identification d’une autre entreprise, vous assumez indirectement un rôle.

Pour vous connecter directement auAWS Management Console, utilisez votre mot de passe avec votre adresse e-mail d'utilisateur racine ou votre nom d'utilisateur IAM. Vous pouvez accéder àAWSEn utilisant vos clés d'accès d'utilisateur racine ou IAM.AWSfournit un kit de développement logiciel (SDK) et des outils de ligne de commande pour signer de manière cryptographique votre requête avec vos informations d'identification Si vous n'utilisez pas les outils AWS, vous devez signer la requête vous-même. Pour ce faire, utilisez Signature Version 4, un protocole permettant d’authentifier les demandes d'API entrantes. Pour plus d'informations sur l'authentification des demandes, consultezProcessus de Signature Version 4dans leAWSRéférence générale.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être également fournir des informations de sécurité supplémentaires. Par exemple, AWS vous recommande d'utiliser l'authentification multi-facteurs (MFA) pour améliorer la sécurité de votre compte. Pour en savoir plus, consultezUtilisation de l'authentification multi-facteurs (MFA) dansAWSdans leGuide de l'utilisateur IAM.

Utilisateur racine d'un compte AWS

Lorsque vous créez un compte AWS, vous commencez avec une seule identité de connexion disposant d'un accès complet à tous les services et ressources AWS du compte. Cette identité est appeléeAWScompteUtilisateur racineet elle est accessible après connexion à l'aide de l'adresse e-mail et du mot de passe utilisés pour la création du compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes, y compris pour les tâches administratives. Respectez plutôt la bonne pratique qui consiste à avoir recours à l'utilisateur racine uniquement pour créer le premier utilisateur IAM. Ensuite, mettez en sécurité les informations d'identification de l'utilisateur racine et utilisez-les uniquement pour effectuer certaines tâches de gestion des comptes et des services.

Utilisateurs et groupes IAM

UnUtilisateur IAMest une identité au sein de votreAWSqui dispose d'autorisations spécifiques pour une seule personne ou application. Un utilisateur IAM peut disposer d'informations d'identification à long terme, comme un nom d'utilisateur et un mot de passe ou un ensemble de clés d'accès. Pour découvrir comment générer des clés d'accès, veuillez consulter Gestion des clés d’accès pour les utilisateurs IAM dans le Guide de l’utilisateur IAM. Lorsque vous générez des clés d'accès pour un utilisateur IAM, veillez à afficher et enregistrer la paire de clés de manière sécurisée. Vous ne pourrez plus récupérer la clé d'accès secrète à l'avenir. Au lieu de cela, vous devrez générer une nouvelle paire de clés d'accès.

Un groupe IAM est une identité qui concerne un ensemble d’utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu’un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

UnRôle IAMest une identité au sein de votreAWSqui dispose d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne

en particulier. Vous pouvez temporairement endosser un rôle IAM dans leAWS Management Consoleparchangement de rôles. Vous pouvez obtenir un rôle en appelant une opération d'API AWS CLI ou AWS à l'aide d'une URL personnalisée. Pour de plus amples informations sur les méthodes d'utilisation des rôles, veuillez consulter Utilisation des rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Autorisations utilisateur IAM temporaires : Un utilisateur IAM peut endosser un rôle IAM pour accepter différentes autorisations temporaires concernant une tâche spécifique.
- Accès d'utilisateurs fédérés— Au lieu de créer un utilisateur IAM, vous pouvez utiliser des identités existantes provenant deAWS Directory Service, votre annuaire d'utilisateurs d'entreprise ou un fournisseur d'identité web. On parle alors d'utilisateurs fédérés. AWS attribue un rôle à un utilisateur fédéré lorsque l'accès est demandé via un fournisseur d'identité. Pour de plus amples informations sur les utilisateurs fédérés, veuillez consulter Utilisateurs fédérés et rôles dans le Guide de l'utilisateur IAM.
- Accès comptes multiples : Vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (mandataire de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès entre plusieurs comptes. Toutefois, certains services AWS vous permettent d'attacher une stratégie directement à une ressource (au lieu d'utiliser un rôle en tant que proxy). Pour en savoir plus sur la différence entre les rôles et les stratégies basées sur les ressources pour l'accès comptes multiples, veuillez consulter Différence entre les rôles IAM et les stratégies basées sur les ressources dans le Guide de l'utilisateur IAM.
- Accès interservice— Un peuAWSutilisent des fonctionnalités dans d'autresAWSServices . Par exemple, lorsque vous effectuez un appel dans un service, il est courant pour ce service d'exécuter des applications dans Amazon EC2 ou de stocker des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du mandataire, un rôle de service ou un rôle lié au service.
  - Autorisations principales— Lorsque vous utilisez un utilisateur ou un rôle IAM pour exécuter des actions dansAWS, vous êtes considéré comme un principal. Les stratégies accordent des autorisations au mandataire. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui déclenche une autre action dans un autre service. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour savoir si une action nécessite d'autres actions dépendantes dans une stratégie, veuillez consulterActions, ressources et clés de condition pourAWS App Runnerdans leRéférence de l'autorisation de service.
  - Rôle de service : Il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Les rôles de service fournissent un accès uniquement au sein de votre compte et

ne peuvent pas être utilisés pour accorder l'accès à des services dans d'autres comptes. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour de plus amples informations, veuillez consulterCréation d'un rôle pour la délégation d'autorisations à unAWSServicedans leGuide de l'utilisateur IAM.

- Rôle lié à un service— Un rôle lié à un service est un type de rôle de service lié à unAWSservice. Le service peut assumer le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre compte IAM et sont la propriété du service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications qui s'exécutent sur Amazon EC2- Vous pouvez utiliser un rôle IAM pour gérer des informations d'identification temporaires pour les applications s'exécutant sur une instance EC2 et effectuant des opérations d'AWS CLIouAWSDemandes d'API Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un rôle AWS à une instance EC2 et le rendre disponible à toutes les applications associées, vous pouvez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour de plus amples informations, veuillez consulter Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l’utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, veuillez consulter Quand créer un rôle IAM (au lieu d’un utilisateur) dans le Guide de l’utilisateur IAM.

Gestion de l'accès à l'aide de stratégies

Vous contrôlez l'accès dansAWSEn créant des stratégies et en les attachant à des identités IAM ouAWSAWS. Une stratégie est un objet dans AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit ses autorisations. Vous pouvez vous connecter en tant qu'utilisateur racine ou IAM ou vous pouvez endosser un rôle IAM. Lorsque vous faites ensuite une demande,AWSévalue les stratégies liées basées sur l'identité ou les ressources. Les autorisations dans les stratégies déterminent si la demande est autorisée ou refusée. La plupart des stratégies sont stockées dans AWS en tant que documents JSON. Pour de plus amples informations sur la structure et le contenu des documents de stratégie JSON, veuillez consulter Présentation des stratégies JSON dans le Guide de l’utilisateur IAM.

Les administrateurs peuvent utiliserAWSStratégies JSON pour spécifier qui a accès à quoi. Cela signifie : quel mandataire peut effectuer des actions sur quel type de ressources et dans quelles conditions.

Chaque entité IAM (utilisateur ou rôle) démarre sans autorisation. En d'autres termes, par défaut, les utilisateurs ne peuvent rien faire, pas même changer leurs propres mots de passe. Pour autoriser un utilisateur à effectuer une opération, un administrateur doit associer une stratégie d'autorisations à ce dernier. Il peut également ajouter l'utilisateur à un groupe disposant des autorisations prévues. Lorsqu'un administrateur accorde des autorisations à un groupe, tous les utilisateurs de ce groupe se voient octroyer ces autorisations.

Les stratégies IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une stratégie qui autorise l'action `iam:GetRole`. Un utilisateur avec cette stratégie peut obtenir des informations utilisateur à partir de l'AWS Management Console, de l'AWS CLI ou de l'API AWS.

Stratégies basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces stratégies contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une stratégie basée sur l'identité, veuillez consulter Création de stratégies IAM dans le Guide de l'utilisateur IAM.

Les stratégies basées sur l'identité peuvent être classées comme étant des stratégies en ligne ou des stratégies gérées. Les stratégies en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les stratégies gérées sont des stratégies autonomes que vous pouvez lier à plusieurs utilisateurs, groupes et rôles de votre compte AWS. Les stratégies gérées incluent les stratégies gérées par AWS et les stratégies gérées par le client. Pour découvrir comment choisir entre une stratégie gérée et une stratégie en ligne, veuillez consulter Choix entre les stratégies gérées et les stratégies en ligne dans le Guide de l'utilisateur IAM.

Stratégies basées sur les ressources

Les stratégies basées sur les ressources sont des documents de stratégie JSON que vous attachez à une ressource. Des stratégies basées sur les ressources sont par exemple, les stratégies de confiance de rôle IAM et des stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les stratégies basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la stratégie, cette dernière définit quel type d'actions un mandataire spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un mandataire dans une stratégie basée sur les ressources. Les mandataires peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ouAWSServices .

Les stratégies basées sur les ressources sont des stratégies en ligne situées dans ce service. Vous ne pouvez pas utiliserAWSStratégies gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels mandataires (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d’accès sont similaires aux stratégies basées sur les ressources, bien qu'elles n'utilisent pas le format de document de stratégie JSON.

Amazon S3,AWS WAFet Amazon VPC sont des exemples de services qui prennent en charge les listes de contrôle d'accès. Pour en savoir plus sur les listes de contrôle d'accès, veuillez consulter Présentation des listes de contrôle d’accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de stratégie

AWS prend en charge d'autres types de stratégies moins courantes. Ces types de stratégies peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de stratégies plus courants.

- Limite d'autorisations : Une limite d'autorisations est une fonction avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une stratégie basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations obtenues représentent la combinaison des stratégies basées sur l'identité de l'entité et de ses limites d'autorisations. Les stratégies basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces stratégies remplace l'autorisation. Pour de plus amples informations sur les limites d'autorisations, veuillez consulter Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Stratégies de contrôle de service (SCP)- Les stratégies de contrôle de service qui spécifient les autorisations maximales pour une organisation ou une unité d'organisation dansAWS Organizations.AWS Organizationsest un service permettant de regrouper et de gérer de façon centralisée plusieursAWScomptes détenus par votre entreprise. Si vous activez toutes les fonctions d'une organisation, vous pouvez appliquer les stratégies de contrôle de service (SCP) à l'un ou à l'ensemble de vos comptes. La politique de contrôle de service limite les autorisations pour les entités dans les comptes membres, y compris chaqueAWSUtilisateur racine d'un compte Pour plus d'informations sur Organizations et les SCP, consultezFonctionnement des stratégies de contrôle de servicedans leAWS OrganizationsGuide de l'utilisateur.

- Stratégies de session : Les stratégies de session sont des stratégies avancées que vous passez en tant que paramètre lorsque vous programmez afin de créer une session temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de la session obtenue sont une combinaison des stratégies basées sur l'identité de l'utilisateur ou du rôle et des stratégies de session. Les autorisations peuvent également provenir d'une stratégie basée sur les ressources. Un refus explicite dans l'une de ces stratégies remplace l'autorisation. Pour de plus amples informations, veuillez consulter Stratégies de session dans le Guide de l'utilisateur IAM.

Plusieurs types de stratégie

Lorsque plusieurs types de stratégies s'appliquent à la requête, les autorisations obtenues sont plus compliquées à comprendre. Pour savoir commentAWSdétermine s'il convient d'autoriser une demande en présence de plusieurs types de stratégies, consultezLogique d'évaluation des stratégiesdans leGuide de l'utilisateur IAM.

Fonctionnement d'App Runner avec IAM

Avant d'utiliser IAM pour gérer l'accès àAWS App Runner, vous devez comprendre quelles sont les fonctionnalités IAM qui peuvent être utilisées avec App Runner. Pour obtenir une vue globale de la façon dont App Runner et d'autresAWSfonctionnent avec IAM, consultezAWSServices qui fonctionnent avec IAMdans leGuide de l'utilisateur IAM.

Pour consulter d'autres rubriques de sécurité App Runner, consultezSécurité.

Rubriques

- Stratégies basées sur l'identité d'App Runner
- Stratégies basées sur les ressources App Runner
- Autorisation basée sur les balises App Runner
- Autorisations utilisateur App Runner
- Rôles IAM du Runner

Stratégies basées sur l'identité d'App Runner

Avec les stratégies IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. App Runner prend en charge des actions, ressources et clés de condition spécifiques. Pour

en savoir plus sur tous les éléments que vous utilisez dans une stratégie JSON, veuillez consulter Références des éléments de stratégie JSON IAM dans le Guide de l'utilisateur IAM.

Actions

Les administrateurs peuvent utiliserAWSStratégies JSON pour spécifier qui a accès à quoi. Cela signifie : quel mandataire peut effectuer des actions sur quel type de ressources et dans quelles conditions.

L'élément `Action` d'une stratégie JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une stratégie. Les actions de stratégie possèdent généralement le même nom que l'opération d'API AWS associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une stratégie. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Les actions de stratégie dans App Runner utilisent le préfixe suivant avant l'action : `apprunner:`. Par exemple, accorder à une personne l'autorisation nécessaire afin d'exécuter une instance Amazon EC2 avec Amazon EC2`RunInstances`, vous incluez l'opération`ec2:RunInstances`dans leur politique. Les déclarations de stratégie doivent inclure un élément `Action` ou `NotAction`. App Runner définit son propre ensemble d'actions qui décrivent les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

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

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante :

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

Pour afficher la liste des actions App Runner, consultezActions définies parAWS App Runnerdans leRéférence de l'autorisation de service.

Resources

Les administrateurs peuvent utiliserAWSStratégies JSON pour spécifier qui a accès à quoi. Cela signifie : quel mandataire peut effectuer des actions sur quel type de ressources et dans quelles conditions.

L'élément de stratégie JSON `Resource` indique le ou les objets pour lesquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour les actions qui sont compatibles avec un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Les ressources App Runner ont la structure ARN suivante :

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

Pour plus d'informations sur le format des ARN, consultezAmazon Resource Names (ARN) etAWSEspaces de noms du servicedans leAWSRéférence générale.

Par exemple, pour spécifier l'élément`my-service`Dans votre instruction, utilisez l'ARN suivant :

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

Pour spécifier tous les services qui appartiennent à un compte spécifique, utilisez le caractère générique (*) :

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

Certaines actions App Runner, telles que celles qui permettent de créer des ressources, ne peuvent pas être exécutées sur une ressource spécifique. Dans ce cas, vous devez utiliser le caractère générique (*).

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

Pour voir la liste des types de ressources App Runner et de leurs ARN, veuillez consulterRessources définies parAWS App Runnerdans leRéférence de l'autorisation de service. Pour en savoir plus sur les actions avec laquelle vous pouvez spécifier l'ARN de chaque ressource, veuillez consulterActions définies parAWS App Runner.

Clés de condition

Les administrateurs peuvent utiliserAWSStratégies JSON pour spécifier qui a accès à quoi. Cela signifie : quel mandataire peut effectuer des actions sur quel type de ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition tels que les signes égal ou inférieur à, pour faire correspondre la condition de la stratégie aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une opération `OR` logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour de plus amples d'informations, veuillez consulter Éléments d'une stratégie IAM : variables et balises dans le Guide de l'utilisateur IAM.

AWSprend en charge des clés de condition globales et des clés de condition spécifiques à un service. Pour afficher toutesAWSClés de condition globalesAWSClés de contexte de condition globalesdans leGuide de l'utilisateur IAM.

App Runner prend en charge l'utilisation de certaines clés de condition globales. Pour afficher toutesAWSClés de condition globalesAWSClés de contexte de condition globaledans leGuide de l'utilisateur IAM.

App Runner définit un ensemble de clés de condition spécifiques à un service. En outre, App Runner prend en charge le contrôle d'accès basé sur les balises, qui est implémenté à l'aide de clés de condition Pour de plus amples informations, veuillez consulter the section called “Autorisation basée sur les balises App Runner”.

Pour afficher la liste des clés de condition App Runner, consultezClés de condition pourAWS App Runnerdans leRéférence de l'autorisation de service. Pour en savoir plus sur les actions et ressources avec lesquelles vous pouvez utiliser une clé de condition, veuillez consulterActions définies parAWS App Runner.

Examples

Pour voir des exemples de stratégies basées sur l'identité App Runner, veuillez consulterExemples de stratégies basées sur l'identité d'App Runner.

Stratégies basées sur les ressources App Runner

App Runner ne prend pas en charge les stratégies basées sur les ressource.

Autorisation basée sur les balises App Runner

Vous pouvez attacher des balises aux ressources App Runner ou transmettre des balises dans une requête à App Runner. Pour contrôler l'accès basé sur des balises, vous devez fournir les informations des balises dans l'élément de condition d'une stratégie en utilisant les clés de condition `apprunner:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou `aws:TagKeys`. Pour plus d'informations sur le balisage des ressources App Runner, consultezthe section called “Configuration”.

Pour afficher un exemple de stratégie basée sur l'identité permettant de limiter l'accès à une ressource basée sur les balises de cette ressource, veuillez consulter Contrôle de l'accès aux services App Runner en fonction des balises.

Autorisations utilisateur App Runner

Pour utiliser App Runner, les utilisateurs IAM ont besoin des autorisations pour les actions App Runner. Une façon courante d'accorder des autorisations aux utilisateurs consiste à attacher une

stratégie à des utilisateurs ou groupes IAM. Pour plus d'informations sur la gestion des autorisations des utilisateurs, consultezModification des autorisations pour un utilisateur IAMdans leGuide de l'utilisateur IAM.

App Runner fournit deux stratégies gérées que vous pouvez attacher à vos utilisateurs.

- `AWSAppRunnerReadOnlyAccess`— Octroi d'autorisations pour répertorier et visualiser les informations relatives aux ressources App Runner.
- `AWSAppRunnerFullAccess`— Octroie des autorisations à toutes les actions App Runner.

Pour un contrôle plus précis des autorisations utilisateur, vous pouvez créer une stratégie personnalisée et l'attacher à vos utilisateurs. Pour plus d'informations, consultez .Création de stratégies IAMdans leGuide de l'utilisateur IAM.

Pour obtenir des exemples de stratégies utilisateur, consultezthe section called “Stratégies utilisateur”.

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": "*"
    }
  ]
}
```

Rôles IAM du Runner

UnRôle IAMest une entité au sein de votreCompte AWSqui dispose d'autorisations spécifiques.

Rôles liés à un service

Les rôles liés à un service permettent aux services AWS d'accéder à des ressources dans d'autres services pour effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre

compte IAM et sont la propriété du service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

App Runner prend en charge les rôles liés à un service. Pour plus d'informations sur la création ou la gestion des rôles liés à un service App Runner, veuillez consulterthe section called “Utilisation des rôles liés à un service”.

Rôles de service

Cette fonctionnalité permet à un service d'endosser un rôle de service en votre nom. Ce rôle autorise le service à accéder à des ressources d'autres services pour effectuer une action en votre nom. Les rôles de service s'affichent dans votre compte IAM et sont la propriété du compte. Cela signifie qu'un administrateur IAM peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

App Runner prend en charge quelques rôles de service.

Rôle d'accès

Le rôle d'accès est un rôle utilisé par App Runner pour accéder aux images dans Amazon Elastic Container Registry (Amazon ECR) dans votre compte. Elle est requise pour accéder à une image dans Amazon ECR et n'est pas requise avec Amazon ECR Public. Avant de créer un service basé sur une image dans Amazon ECR, utilisez IAM pour créer un rôle de service et utilisez l'outil`AWSAppRunnerServicePolicyForECRAccess`stratégie gérée en elle. Vous pouvez ensuite passer ce rôle à App Runner lorsque vous appelez la méthodeCreateServicedans la consoleAuthenticationConfiguration(membre de laSourceConfiguration) ou lors de l'utilisation de la console App Runner pour créer un service.

AWSAppRunnerServicePolicyForECRAccess

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

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

> **Note**
>
> Si vous créez votre propre stratégie personnalisée pour votre rôle d'accès, veillez à spécifier`"Resource": "*"`pour la`ecr:GetAuthorizationToken`action. Les jetons peuvent être utilisés pour accéder à n'importe quel registre Amazon ECR auquel vous avez accès.

Lorsque vous créez votre rôle d'accès, veillez à ajouter une stratégie d'approbation qui déclare le principal de service App Runner`build.apprunner.amazonaws.com`en tant qu'entité de confiance.

Stratégie d'approbation pour un rôle d'accès

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

Si vous utilisez la console App Runner pour créer un service, la console peut automatiquement créer un rôle d'accès pour vous et le choisir pour le nouveau service. La console répertorie également d'autres rôles dans votre compte et vous pouvez sélectionner un rôle différent si vous le souhaitez.

Rôle d'instance

Le rôle d'instance est un rôle facultatif que App Runner utilise pour fournir des autorisations àAWSque votre code d'application appelle. Avant de créer un service App Runner, utilisez IAM pour créer un rôle de service avec les autorisations dont votre code d'application a besoin. Vous pouvez

ensuite passer ce rôle à App Runner dans laCreateServiceou lors de l'utilisation de la console App Runner pour créer un service.

Lorsque vous créez votre rôle d'instance, veillez à ajouter une stratégie d'approbation qui déclare le principal de service App Runner`tasks.apprunner.amazonaws.com`en tant qu'entité de confiance.

Stratégie d'approbation pour un rôle d'instance

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

Si vous utilisez la console App Runner pour créer un service, la console répertorie les rôles dans votre compte et vous pouvez sélectionner le rôle que vous avez créé à cette fin.

Pour plus d'informations sur la création d'un service, consultezthe section called “Création”.

> **Note**
>
> Les autorisations que le rôle d'instance doit fournir dépendent entièrement de votre application. Votre code peut ne pas appelerAWSAPI, et dans ce cas, vous n'avez pas besoin de fournir un rôle d'instance à App Runner.
> Dans le cas où votre code appelleAWSAPI, nous n'avons aucun moyen d'anticiper quels appels ils sont. Par conséquent, nous ne fournissons aucune stratégie gérée pour les rôles d'instance. Vous devez explicitement inclure les autorisations requises dans votre rôle d'instance ou créer votre propre stratégie personnalisée et l'utiliser dans le rôle d'instance.

Exemples de stratégies basées sur l'identité d'App Runner

Par défaut, les utilisateurs et rôles IAM ne sont pas autorisés à créer ou modifierAWS App RunnerAWS. Ils ne peuvent pas non plus exécuter des tâches à l'aide de AWS Management

Console, AWS CLI ou de l'API AWS. Un administrateur IAM doit créer des stratégies IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces stratégies aux utilisateurs ou aux groupes IAM ayant besoin de ces autorisations.

Pour savoir comment créer une stratégie IAM basée sur l'identité à l'aide de ces exemples de documents de stratégie JSON, veuillez consulter Création de stratégies dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Pour consulter d'autres rubriques de sécurité App Runner, consultezSécurité.

Rubriques

- Bonnes pratiques en matière de stratégies
- Stratégies utilisateur
- Contrôle de l'accès aux services App Runner en fonction des balises

Bonnes pratiques en matière de stratégies

Les stratégies basées sur l'identité sont très puissantes. Elles déterminent si une personne peut créer, consulter ou supprimer des ressources App Runner dans votre compte. Ces actions peuvent entraîner des frais pour votre compte AWS. Lorsque vous créez ou modifiez des stratégies basées sur l'identité, suivez ces instructions et recommandations :

- Commencez à utiliserAWSStratégies gérées— Pour commencer à utiliser App Runner rapidement, utilisezAWSStratégies gérées pour accorder à vos employés les autorisations dont ils ont besoin. Ces stratégies sont déjà disponibles dans votre compte et sont gérées et mises à jour par AWS. Pour de plus amples informations, veuillez consulterCommencez à utiliser les autorisations avecAWSStratégies gérées dans leGuide de l'utilisateur IAM.
- Accorder le privilège le plus faible : Lorsque vous créez des stratégies personnalisées, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Commencez avec un minimum d'autorisations et accordez-en d'autres si nécessaire. Cette méthode est plus sûre que de commencer avec des autorisations trop permissives et d'essayer de les restreindre plus tard. Pour de plus amples informations, veuillez consulter Accorder les privilèges les plus faibles possible dans le Guide de l'utilisateur IAM.
- Activer la MFA pour les opérations confidentielles : Pour plus de sécurité, demandez aux utilisateurs IAM d'utiliser la Multi-Factor Authentication (MFA) pour accéder à des ressources ou à

des opérations d'API confidentielles. Pour de plus amples informations, veuillez consulterUtilisation de l'authentification multi-facteurs (MFA) dansAWSdans leGuide de l'utilisateur IAM.

- Utiliser des conditions de stratégie pour davantage de sécurité : Définissez les conditions dans lesquelles vos stratégies basées sur l'identité autorisent l'accès à une ressource, dans la mesure où cela reste pratique. Par exemple, vous pouvez rédiger les conditions pour spécifier une plage d'adresses IP autorisées d'où peut provenir une demande. Vous pouvez également écrire des conditions pour autoriser les requêtes uniquement à une date ou dans une plage de temps spécifiée, ou pour imposer l'utilisation de SSL ou de MFA. Pour de plus amples informations, veuillez consulterÉléments de stratégie JSON IAM : Conditiondans leGuide de l'utilisateur IAM.

Stratégies utilisateur

Pour accéder à la console App Runner, les utilisateurs IAM doivent disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources App Runner dans votreCompte AWS. Si vous créez une stratégie basée sur l'identité qui est plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les utilisateurs dotés de cette stratégie.

App Runner fournit deux stratégies gérées que vous pouvez attacher à vos utilisateurs.

- `AWSAppRunnerReadOnlyAccess`— Octroi d'autorisations pour répertorier et visualiser les informations relatives aux ressources App Runner.
- `AWSAppRunnerFullAccess`— Octroie des autorisations à toutes les actions App Runner.

Pour vous assurer que les utilisateurs peuvent utiliser la console App Runner, attachez, au minimum, le`AWSAppRunnerReadOnlyAccess`Stratégie gérée pour les utilisateurs. Vous pouvez joindre le fichier`AWSAppRunnerFullAccess`ou ajouter des autorisations supplémentaires spécifiques pour permettre aux utilisateurs de créer, modifier et supprimer une ressource. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l’utilisateur IAM.

Vous n'avez pas besoin d'accorder les autorisations minimales de console pour les utilisateurs qui effectuent des appels uniquement à l'interface AWS CLI ou API AWS. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API que vous souhaitez autoriser les utilisateurs à effectuer.

Les exemples suivants illustrent des stratégies utilisateur personnalisées. Vous pouvez les utiliser comme points de départ pour définir vos propres stratégies utilisateur personnalisées. Copiez l'exemple, ou supprimez des actions, étendez les ressources et ajoutez des conditions.

Exemple : stratégie utilisateur de gestion de la console et des connexions

Cet exemple de stratégie active l'accès à la console et permet la création et la gestion des connexions. Il n'autorise pas la création et la gestion du service App Runner. Il peut être associé à un utilisateur dont le rôle est de gérer l'accès au service App Runner aux ressources de code source.

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

Exemple : stratégies utilisateur utilisant des clés de condition

Les exemples de cette section illustrent les autorisations conditionnelles qui dépendent de certaines propriétés de ressource ou de certains paramètres d'action.

Cet exemple de stratégie permet de créer un service App Runner mais refuse l'utilisation d'une connexion nomméeprod.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateAppRunnerServiceWithNonProdConnections",
      "Effect": "Allow",
      "Action": "apprunner:CreateService",
      "Resource": "*",
```

```
      "Condition": {
        "ArnNotLike": {
          "apprunner:ConnectionArn": "arn:aws:apprunner:*:*:connection/prod/*"
        }
      }
    }
  ]
}
```

Cet exemple de stratégie permet de mettre à jour un service App Runner nommé`preprod`uniquement avec une configuration de mise à l'échelle automatique nommée`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/*"
        }
      }
    }
  ]
}
```

Contrôle de l'accès aux services App Runner en fonction des balises

Vous pouvez utiliser des conditions dans votre stratégie basée sur l'identité pour contrôler l'accès aux ressources App Runner en fonction des balises. Cet exemple montre comment créer une stratégie qui permet de supprimer un service App Runner. Toutefois, l'autorisation est accordée uniquement si la balise de service `Owner` a la valeur du nom d'utilisateur de cet utilisateur. Cette stratégie accorde également les autorisations nécessaires pour réaliser cette action sur la console.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

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

Vous pouvez rattacher cette stratégie aux utilisateurs IAM de votre compte. Si un utilisateur nommé`richard-roe`tente de supprimer un service App Runner, le service doit être balisé`Owner=richard-roe`ou`owner=richard-roe`. Dans le cas contraire, l'utilisateur se voit refuser l'accès. La clé de condition de balise `Owner` correspond à la fois à `Owner` et à `owner`, car les noms de clé de condition ne sont pas sensibles à la casse. Pour de plus amples informations, veuillez consulterÉléments de stratégie JSON IAM : Conditiondans leGuide de l'utilisateur IAM.

Utilisation de rôles liés à un service pour App Runner

AWS App Runnerutilise le traitementAWS Identity and Access Management(IAM)Rôles liés à un service. Un rôle lié à un service est un type unique de rôle IAM lié directement à App Runner. Les rôles liés à un service sont prédéfinis par App Runner et comprennent toutes les autorisations nécessaires au service pour appeler d'autresAWSServices en votre nom.

Un rôle lié à un service simplifie la configuration d'App Runner, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. App Runner définit les autorisations de ses rôles liés à un service et, sauf définition contraire, seul App Runner peut endosser ses rôles. Les autorisations définies comprennent la stratégie d'approbation et la stratégie d'autorisation. De plus, cette stratégie d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources App Runner sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM et recherchez les services qui comportent Oui dans la colonne Rôle lié à un service. Sélectionnez un Yes (Oui) avec un lien permettant de consulter la documentation du rôle lié à un service, pour ce service.

Pour consulter d'autres rubriques de sécurité App Runner, consultezSécurité.

Autorisations des rôles liés à un service pour App Runner

App Runner utilise le rôle lié à un service nommé.AWSServiceRoleForAprunner.

Ce rôle permet à App Runner d'exécuter les tâches suivantes :

- Transmettre les journaux à des groupes de journaux Amazon CloudWatch Logs.
- Créez des règles Amazon CloudWatch Events pour vous abonner à Amazon Elastic Container Registry (Amazon ECR).

Le rôle lié à un service AWSServiceRoleForApprunner approuve les services suivants pour assumer le rôle :

- `apprunner.amazonaws.com`

La stratégie d'autorisations du rôle lié à un service AWSServiceRoleForApprunner contient toutes les autorisations dont App Runner a besoin pour exécuter des actions en votre nom.

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"
        }
      }
    }
  ]
}
```

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, unn groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour de plus amples informations, veuillez consulter Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour App Runner

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez un service App Runner dans leAWS Management Console, leAWS CLI, ou leAWSAPI, App Runner crée le rôle lié au service pour vous.

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous créez un service App Runner, App Runner crée automatiquement le rôle lié au service.

Modification d'un rôle lié à un service pour App Runner

App Runner ne vous permet pas de modifier le rôle lié à au service AWSServiceRoleForApprunner. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas modifier le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour de plus amples informations, veuillez consulter Modification d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour App Runner

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

Dans App Runner, cela implique de supprimer tous les services App Runner de votre compte. Pour en savoir plus sur la suppression des services App Runner, consultezthe section called “Suppression”.

> **Note**
>
> Si le service App Runner utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM, la consoleAWS CLI, ou leAWSAPI pour supprimer le rôle lié au service AWSServiceRoleForApprunner. Pour de plus amples informations, veuillez consulter Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés à un service App Runner

App Runner prend en charge l'utilisation des rôles liés à un service dans toutes les régions où le service est disponible. Pour de plus amples informations, veuillez consulterAWS App RunnerPoints de terminaison et quotasdans leAWSRéférence générale.

Stratégies gérées AWS pour AWS App Runner

Pour ajouter des autorisations aux utilisateurs, groupes et rôles, il est plus facile d'utiliserAWSplutôt que d'écrire des stratégies vous-même. Il faut du temps et de l'expertise pourCréer des stratégies IAM gérées par le clientqui fournissent à votre équipe uniquement les autorisations dont elle a besoin. Pour démarrer rapidement, utilisez notreAWSStratégies gérées. Ces politiques couvrent les cas d'utilisation courants et sont disponibles dans votreAWS. Pour plus d'informations surAWSStratégies gérées, voirAWSStratégies gérées dans leGuide de l'utilisateur IAM.

AWSmaintenance et mise à jour des servicesAWSStratégies gérées. Vous ne pouvez pas modifier les autorisations dansAWSStratégies gérées. Les services ajoutent occasionnellement des autorisations supplémentaires à unAWSpour prendre en charge les nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la stratégie est attachée. Les services sont les plus susceptibles de mettre à jour unAWSlorsqu'une nouvelle fonctionnalité est lancée ou lorsque de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'unAWS, de sorte que les mises à jour de stratégie ne rompront pas vos autorisations existantes.

En outre,AWSprend en charge des stratégies gérées pour les activités professionnelles qui couvrent plusieurs services. Par exemple, les recettesReadOnlyAccess AWSStratégie gérée fournit un accès en lecture seule à tousAWSservices et ressources. Lorsqu'un service lance une nouvelle fonctionnalité,AWSajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir la liste et la description des stratégies de fonctions professionnelles, consultez la pageAWSStratégies gérées pour les activités professionnelles dans leGuide de l'utilisateur IAM.

Mises à jour App Runner versAWSStratégies gérées

Affiche les détails des mises à jour deAWSpour App Runner depuis que ce service a commencé à suivre ces modifications. Pour obtenir des alertes automatiques sur les modifications apportées à cette page, abonnez-vous au flux RSS de la page d'historique du document App Runner.

Modification	Description	Date
ApprunnerServiceRolePolicy— Nouvelle stratégie	App Runner a ajouté une nouvelle politique permettant à App Runner de passer des appels vers Amazon CloudWatch Logs et Amazon CloudWatch Events au nom des services App Runner.	1er mars 2021
AWSApprunnerReadOnlyAccess— Nouvelle stratégie	App Runner a ajouté une nouvelle stratégie pour permettre aux utilisateurs de répertorier et d'afficher des détails sur les ressources App Runner.	1er mars 2021
AWSApprunnerFullAccess— Nouvelle stratégie	App Runner a ajouté une nouvelle stratégie permettant aux utilisateurs d'effectuer toutes les actions App Runner.	1er mars 2021
AWSApprunnerServicePolicyForecrAccess— Nouvelle stratégie	App Runner a ajouté une nouvelle stratégie pour permettre à App Runner d'accéder aux images Amazon Elastic Container Registry (Amazon ECR) dans votre compte.	1er mars 2021
App Runner a commencé à suivre les modifications	App Runner a commencé à suivre les modifications pour sonAWSStratégies gérées.	1er mars 2021

Résolution des problèmes liés à Identity and Access

Utilisez les informations suivantes pour identifier et résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avecAWS App Runneret IAM.

Pour consulter d'autres rubriques de sécurité App Runner, consultezSécurité.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans App Runner
- Je suis un administrateur et je veux autoriser d'autres utilisateurs à accéder à App Runner
- Je veux permettre à des personnes extérieures à monCompte AWSpour accéder à mes ressources App Runner

Je ne suis pas autorisé à effectuer une action dans App Runner

Si AWS Management Console indique que vous n'êtes pas autorisé à exécuter une action, contactez votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votreAWSNom d'utilisateur et mot de passe.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé`marymajor`tente d'utiliser la console pour afficher des informations détaillées concernant un service App Runner mais ne dispose pas d'`apprunner:DescribeService`Autorisations.

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

Dans ce cas, Mary demande à son administrateur de mettre à jour ses stratégies pour lui permettre d'accéder à*my-example-service*à l'aide de`apprunner:DescribeService`action.

Je suis un administrateur et je veux autoriser d'autres utilisateurs à accéder à App Runner

Pour permettre à d'autres utilisateurs d'accéder à App Runner, vous devez créer une entité IAM (utilisateur ou rôle) pour la personne ou l'application qui a besoin de l'accès. Ils utiliseront les informations d'identification de cette entité pour accéder à AWS. Vous devez ensuite associer une stratégie à l'entité qui leur accorde les autorisations appropriées dans App Runner.

Pour démarrer immédiatement, veuillez consulter Création de votre premier groupe et utilisateur délégué IAM dans le Guide de l’utilisateur IAM.

Je veux permettre à des personnes extérieures à monCompte AWSpour accéder à mes ressources App Runner

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation peuvent utiliser pour accéder à vos ressources. Vous pouvez spécifier la personne à qui vous souhaitez confier le rôle. Pour les services qui prennent en charge les stratégies basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces stratégies pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si App Runner est compatible avec ces fonctionnalités, consultezFonctionnement d'App Runner avec IAM.

- Pour savoir comment donner accès à vos ressources surAWSque vous possédez, consultezOctroi de l'accès à un utilisateur IAM dans un autreAWSCompte que vous possédezdans leGuide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiersAWScomptes, voirOctroi d'un accèsAWScomptes appartenant à des tiersdans leGuide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, veuillez consulter Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des stratégies basées sur les ressources pour l'accès comptes multiples, veuillez consulter Différence entre les rôles IAM et les stratégies basées sur les ressources dans le Guide de l'utilisateur IAM.

Journalisation et surveillance dans App Runner

La surveillance est un encours important pour assurer la fiabilité, la disponibilité et les performances de votreAWS App Runnerservice service service service Collecte des données de surveillance de toutes les parties de votreAWSvous permet de déboguer plus facilement un échec si un échec se produit. App Runner s'intègre à plusieursAWSpour surveiller vos services App Runner et répondre aux éventuels incidents.

Alarmes Amazon CloudWatch

Les alarmes Amazon CloudWatch permettent de surveiller une métrique de service sur une période de temps que vous spécifiez. Si la métrique dépasse un seuil donné pour un nombre donné de périodes, vous recevez une notification.

App Runner collecte une variété de mesures concernant le service dans son ensemble et les instances (unités de mise à l'échelle) qui exécutent votre service Web. Pour plus d'informations, consultez Mesures (CloudWatch).

Journaux d'application

App Runner collecte la sortie de votre code d'application et la diffuse dans Amazon CloudWatch Logs. Ce qui est dans cette sortie est à vous de choisir. Par exemple, vous pouvez inclure des enregistrements détaillés des demandes adressées à votre service Web. Ces enregistrements journaux peuvent s'avérer utiles en cas d'audit de sécurité ou d'audit des accès. Pour plus d'informations, consultez Journaux (journaux CloudWatch Logs).

AWS CloudTrailJournalisation d'actions

App Runner est intégré àAWS CloudTrail, un service qui fournit un enregistrement des actions réalisées par un utilisateur, un rôle ou uneAWSdans App Runner. CloudTrail capture tous les appels d'API pour App Runner en tant qu'événements. Vous pouvez afficher les événements les plus récents dans la console CloudTrail et créer une piste pour permettre la livraison continue des événements CloudTrail vers un compartiment Amazon Simple Storage Service (Amazon S3). Pour plus d'informations, consultez Actions API (CloudTrail).

Validation de la conformité pour App Runner

Les auditeurs tiers évaluent la sécurité et la conformité de AWS App Runner dans le cadre de plusieurs programmes de conformité AWS. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et autres.

Pour savoir si App Runner ou d'autresAWSServices concernés par des programmes de conformité spécifiques, consultezAWSServices concernés par le programme de conformité. Pour obtenir des renseignements généraux, consultez Programmes de conformitéAWS .

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour de plus amples informations, veuillez consulter Téléchargement de rapports dans AWS Artifact.

Votre responsabilité en matière de conformité lorsque vous utilisez des services AWS est déterminée par la sensibilité de vos données, des objectifs de conformité de votre entreprise, ainsi que de la législation et de la réglementation en vigueur. AWS fournit les ressources suivantes pour faciliter le respect de la conformité :

- Guides de démarrage rapide de la sécurité et de la conformité— Ces guides de déploiement proposent des considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de référence dansAWSqui sont axées sur la sécurité et la conformité.
- Livre blanc Architecting for HIPAA Security and Compliance— Ce livre blanc décrit comment les entreprises peuvent utiliserAWSpour créer des applications conformes à la loi HIPAA.

  > **Note**
  >
  > Tous les services ne sont pas conformes à HIPAA.

- AWSRessources de conformité— Cet ensemble de manuels et de guides peut renfermer des informations concernant votre secteur et votre région.

- Évaluation des ressources à l'aide de règlesdans leAWS ConfigManuel du développeur— LeAWS Configpermet d'évaluer comment les configurations de vos ressources se conforment aux pratiques internes, aux normes et aux directives industrielles.
- AWS Security Hub— CeAWSfournit une vue complète de l'état de votre sécurité dansAWSCela vous aide à vérifier votre conformité avec les normes et les bonnes pratiques du secteur de la sécurité.
- AWS Audit Manager— CeAWSvous aide à auditer en continu votreAWSpour simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Pour consulter d'autres rubriques sur la sécurité d'App Runner, consultezSécurité.

Résilience dans App Runner

La .AWSL'infrastructure globale d repose surRégions AWSet zones de disponibilité.Régions AWSfournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations surRégions AWSet zones de disponibilité, consultezAWSInfrastructure mondiale.

AWS App Runner gère et automatise l'utilisation de l'infrastructure AWS globale en votre nom. Lorsque vous utilisez App Runner, vous bénéficiez des mécanismes de disponibilité et de tolérance aux pannes offerts parAWSOffres.

Pour consulter d'autres rubriques sur la sécurité d'App Runner, consultezSécurité.

Sécurité de l'infrastructure dans AWS App Runner

En tant que service géré,AWS App Runnerest protégé par laAWSLes procédures de sécurité du réseau mondial qui sont décrites dans leAmazon Web Services : Présentation des processus de sécuritéLivre blanc.

Vous utilisezAWSPublication d'appels d'API pour accéder à App Runner via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.0 ou une version ultérieure.

Nous recommandons TLS 1.2 ou version ultérieure. Les clients doivent aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un mandataire IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Pour consulter d'autres rubriques de sécurité App Runner, veuillez vous reporterSécurité.

Configuration et analyse des vulnérabilités dans App Runner

AWS et nos clients partagent la responsabilité d'obtenir un niveau élevé de sécurité et de conformité des composants logiciels. Pour de plus amples informations, veuillez consulter AWS Modèle de responsabilité partagée.

Pour consulter d'autres rubriques sur la sécurité des App Runner, consultezSécurité.

Bonnes pratiques de sécurité pour App Runner

AWS App Runner fournit différentes fonctionnalités de sécurité à prendre en compte lorsque vous développez et mettez en œuvre vos propres stratégies de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des considérations utiles et non comme des recommandations.

Pour consulter d'autres rubriques sur la sécurité App Runner, consultezSécurité.

Bonnes pratiques de sécurité préventive

Les contrôles de sécurité préventifs tentent d'éviter les incidents avant qu'ils ne se produisent.

Implémentation d'un accès sur la base du moindre privilège

Exécuteur d'applications fournitAWS Identity and Access Management(IAM) pourUtilisateurs IAMet l'Rôle d'accès. Ces stratégies gérées spécifient toutes les autorisations qui peuvent être nécessaires au bon fonctionnement de votre service App Runner.

Votre application peut ne pas exiger toutes les autorisations de nos stratégies gérées. Vous pouvez les personnaliser et accorder uniquement les autorisations requises pour que vos utilisateurs et votre service App Runner puissent effectuer leurs tâches. C'est particulièrement pertinent pour les stratégies utilisateur, où différents rôles utilisateur peuvent avoir des besoins différents en matière d'autorisations. L'implémentation d'un accès sur la base du moindre privilège est fondamentale pour réduire les risques en matière de sécurité et l'impact que pourraient avoir des d'erreurs ou des actes de malveillance.

Bonnes pratiques de sécurité de détection

Les contrôles de sécurité de détection identifient les violations de sécurité après qu'elles se sont produites. Ils peuvent vous aider à détecter une menace ou un incident de sécurité potentiel.

Mise en œuvre de la surveillance

La surveillance est un enjeu important pour assurer la fiabilité, la sécurité, la disponibilité et les performances de vos solutions App Runner.AWSVous trouverez plusieurs outils et services pour vous aider à surveiller votreAWSServices .

Voici quelques exemples d'éléments à surveiller :

- Métriques Amazon CloudWatch pour App RunnerDéfinissez les alarmes pour les principales métriques App Runner et pour les métriques personnalisées de votre application. Pour de plus amples informations, veuillez consulter Mesures (CloudWatch).
- AWS CloudTrailentriesSuivez les actions qui peuvent avoir un impact sur la disponibilité, comme`PauseService`ou`DeleteConnection`. Pour de plus amples informations, veuillez consulter Actions API (CloudTrail).

AWSGlossaire

Pour la dernière version de l'AWS, consultez la terminologieAWSGlossairedans leAWSRéférence générale.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.