Guia do desenvolvedor

AWS App Runner

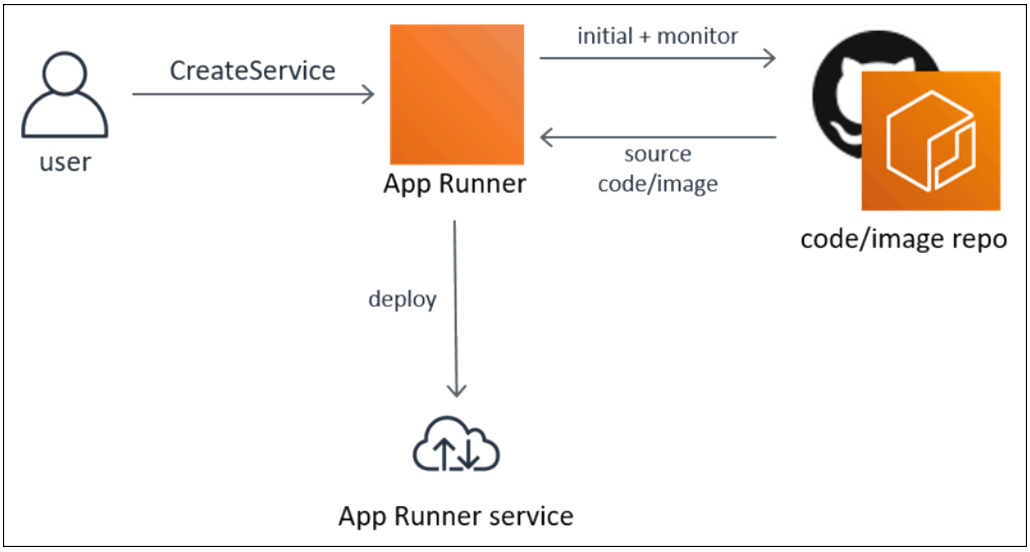
AWS App Runner: Guia do desenvolvedor

Table of Contents

O que é o AWS App Runner?

AWS App Runneré umAWSque fornece uma maneira rápida, simples e econômica de implantar a partir de código-fonte ou de uma imagem de contêiner diretamente em um aplicativo Web escalável e seguro noAWSNuvem. Você não precisa aprender novas tecnologias, decidir qual serviço de computação usar ou saber como provisionar e configurarAWSrecursos da AWS.

App Runner se conecta diretamente ao seu repositório de código ou imagem. Ele fornece uma integração automática e pipeline de entrega com operações totalmente gerenciadas, alto desempenho, escalabilidade e segurança.



Para quem é o App Runner?

Se você for umdesenvolvedor, você pode usar o App Runner para simplificar o processo de implantação de uma nova versão do seu código ou repositório de imagens.

para oequipes de operações, o App Runner habilita implantações automáticas sempre que um commit for enviado para o repositório de código ou uma nova versão de imagem de contêiner for enviada para o repositório de imagens.

Acessar o App Runner

Você pode definir e configurar suas implantações de serviço do App Runner usando qualquer uma das seguintes interfaces:

- Console do App Runner— Fornece uma interface da Web para gerenciar seus serviços do App Runner.
- API do Executor de aplicativos— Fornece uma API RESTful para executar ações do App Runner. Para obter mais informações, consulte Referência da API do AWS App Runner.
- AWSInterface da linha de comando (AWS CLI)— Fornece comandos para um amplo conjunto deAWS, incluindo a Amazon VPC, e é compatível com Windows, macOS e Linux. Para mais informações, consulte AWS Command Line Interface.
- AWSSDKs da— Fornece APIs específicas para a linguagem e cuida de muitos dos detalhes de conexão, como calcular assinaturas, lidar com novas tentativas de solicitação e manusear erros. Para mais informações, consulte AWS SDKs.

Preços para App Runner

O App Runner oferece uma maneira econômica de executar seu aplicativo. Você paga apenas pelos recursos que seu serviço App Runner consome. Seu serviço pode ser reduzido para menos instâncias de computação quando o tráfego de solicitações é mais lento. Você tem controle sobre as configurações de escalabilidade: o menor e o maior número de instâncias provisionadas e a maior carga que uma instância controla.

Para obter mais informações sobre a escalabilidade automática do App Runner, consultethe section called “Auto Scaling”.

Para obter informações sobre a definição de preço, consulte AWS App Runner Pricing (Definição de preço).

E o próximo

Saiba como começar a usar o App Runner nos seguintes tópicos:

- Configuração— Conclua as etapas de pré-requisito para usar o App Runner.
- Conceitos básicos— Implante seu primeiro aplicativo no App Runner.

Configuração para App Runner

Se você é um novoAWS, preencha os pré-requisitos de configuração listados nesta página antes de começar a usarAWS App Runner.

Para estes procedimentos de configuração, utilize o comandoAWS Identity and Access Managementserviço (IAM). Para obter mais informações sobre o IAM, consulte os seguintes materiais de referência:

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

Criar um Conta da AWS.

Quando você se inscreve noAWS, você obtém um número de conta com acesso a todos os serviços queAWSOfertas, incluindoAWS App Runner.

Se você já tiver umConta da AWS, vá para o próximo pré-requisito.

Se você não tiver umAWSConta da, conclua as etapas a seguir para criar uma.

Para se cadastrar em uma conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções online.

   Parte do procedimento de cadastro envolve uma chamada telefônica e a digitação de um código de verificação usando o teclado do telefone.

Criar um usuário do IAM

Para acessar umAWSServiço, forneça as credenciais do. Essas credenciais determinamautenticação(quem você é) eautorização(quais permissões você tem para executar ações noAWSrecursos da AWS).

Quando você cria uma Amazon Web Services (AWS), você começa com uma única identidade de logon. Essa identidade tem acesso completo a todos os serviços e recursos da AWS da conta. Essa

identidade é chamadaAWSaccountUsuário raiz. Ao fazer login, insira o endereço de e-mail e a senha usados para criar a conta.

> **Important**
>
> Recomendamos que não use o usuário raiz para suas tarefas diárias, nem mesmo as administrativas. Em vez disso, siga as práticas recomendadas para o uso do usuário raiz somente a fim de criar seu primeiro usuário do IAM. Depois, armazene as credenciais do usuário raiz com segurança e use-as para executar somente algumas tarefas de gerenciamento de contas e de serviços. Para visualizar as tarefas que exigem que você faça login como usuário raiz, consulteTarefas que exigem credenciais de usuário raiz.

Para obter mais informações sobre o usuário raiz e as credenciais do usuário do IAM, consulteConta da AWScredenciais de usuário raiz e credenciais de usuário do IAMnoAWSReferência geral.

Para criar um usuário administrador para você mesmo e adicionar o usuário a um grupo de administradores (console)

1. Faça login noConsole do IAMcomo o proprietário da conta escolhendoUsuário raize inserindo seuAWSEndereço de e-mail da conta. Na próxima página, insira sua senha.

   > **Note**
   >
   > Recomendamos que você siga as melhores práticas para utilizar o**`Administrator`**Usuário do IAM que segue e armazena as credenciais do usuário raiz com segurança. Cadastre-se como o usuário raiz apenas para executar algumas tarefas de gerenciamento de serviços e contas.

2. No painel de navegação, escolha Usuários e depois Adicionar usuário.
3. Em User name (Nome do usuário), digite **`Administrator`**.
4. Marque a caixa de seleção ao lado deAWS Management ConsoleAcesso do. Então, selecione Custom password (Senha personalizada), e insira sua nova senha na caixa de texto.
5. (Opcional) Por padrão,AWSO requer que o novo usuário crie uma senha ao fazer login pela primeira vez. Você pode desmarcar a caixa de seleção próxima de User must create a new password at next sign-in (O usuário deve criar uma senha no próximo login) para permitir que o novo usuário redefina a senha depois de fazer login.

6. Selecione Next (Próximo): Permissions
7. Em Set permissions (Conceder permissões), escolha Add user to group (Adicionar usuário ao grupo).
8. Escolha Create group (Criar grupo).
9. Na caixa de diálogo Create group (Criar grupo), em Group name (Nome do grupo), digite **`Administrators`**.
10. SelecionePolíticas de filtroE depois selecioneAWSgerenciado - função de trabalhopara filtrar o conteúdo da tabela.
11. Na lista de políticas, marque a caixa de seleção AdministratorAccess. A seguir escolha Criar grupo.

> **Note**
>
> Ative acesso do usuário e da função do IAM ao Faturamento para poder usar o`AdministratorAccess`Permissões para acessar oAWS Billing and Cost Managementconsole do . Para fazer isso, siga as instruções na etapa 1 do tutorial sobre como delegar acesso ao console de faturamento.

12. Suporte a lista de grupos, selecione a caixa de seleção para seu novo grupo. Escolha Refresh (Atualizar) caso necessário, para ver o grupo na lista.
13. Selecione Next (Próximo): Tags.
14. (Opcional) Adicione metadados ao usuário anexando tags como pares de chave-valor. Para obter mais informações sobre como usar tags no IAM, consulte Marcar entidades do IAM no Guia do usuário do IAM.
15. Selecione Next (Próximo): Review (Revisar)Para ver uma lista de associações a grupos a serem adicionadas ao novo usuário. Quando você estiver pronto para continuar, selecione Criar usuário.

Você pode usar esse mesmo processo para criar mais grupos e usuários e conceder aos seus usuários acesso aos recursos de sua conta da AWS. Para saber mais sobre o uso de políticas que restringem as permissões do usuário aAWSrecursos da AWS, consulteGerenciamento de acessoeExemplo de políticas.

> **Important**
>
> Proteger oConta da AWS. Nunca envie ou compartilhe suas credenciais com ninguém fora da sua organização. Ninguém que legitimamente represente a Amazon jamais pedirá suas credenciais a você.

Depois de criar seu usuário do IAM, use suas credenciais para fazer login noAWS Management Console. Para obter mais informações, consulteComo os usuários do IAM fazem login noConta da AWSnoGuia do usuário do IAM.

Criar uma chave de acesso para seu usuário do IAM

As chaves de acesso consistem em um ID de chave de acesso e uma chave de acesso secreta, que são usados para assinar solicitações programáticas feitas aoAWS. Se você não tiver chaves de acesso, você poderá criá-las noAWS Management Console. Como melhor prática, não utilize oAWSChaves de acesso do usuário raiz da conta da para qualquer tarefa para a qual ele não seja necessário. Em vez disso,Criar um novo usuário do IAMcom as chaves de acesso para si mesmo.

A única vez que você pode visualizar ou fazer download da chave de acesso secreta é quando você cria as chaves do. Não será possível recuperá-las posteriormente. No entanto, você pode criar novas chaves de acesso a qualquer momento. Você também deve ter permissão para executar as ações do IAM necessárias. Para obter mais informações, consultePermissões necessárias para acessar os recursos do IAMnoGuia do usuário do IAM.

Para criar chaves de acesso para um usuário do IAM

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Usuários.
3. Escolha o nome do usuário cujas chaves de acesso você deseja criar e escolhaCredenciais de segurançaGuia.
4. NoChaves de acessoSeção, escolhaCriar chave de acesso.
5. Para visualizar o novo key pair de acesso, escolhaMostrar. Você não terá mais acesso à chave de acesso secreta depois que essa caixa de diálogo for fechada. Suas credenciais terão a seguinte aparência:

- ID de chave de acesso: AKIAIOSFODNN7EXAMPLE
- Chave de acesso secreta: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

6. Para baixar o par de chaves, escolha Baixar arquivo .csv. Armazene as chaves em um lugar seguro. Você não terá mais acesso à chave de acesso secreta depois que essa caixa de diálogo for fechada.

   Mantenha a confidencialidade das chaves para proteger suaAWSconta e nunca as envie por e-mail. Não compartilhe as chaves fora da sua organização, mesmo se uma pesquisa parecer vir da AWS ou da Amazon.com. Alguém que legitimamente represente a Amazon jamais pedirá a você sua chave secreta.

7. Depois de baixar o `.csv`, escolhaFechar. Quando você cria uma chave de acesso, o par de chaves é ativo por padrão, e você pode usar o par imediatamente.

Tópicos relacionados

- O que é o IAM?noGuia do usuário do IAM
- AWSCredenciais de segurança daemAWSReferência geral

E o próximo

Você concluiu as etapas de pré-requisito. Para implantar seu primeiro aplicativo no App Runner, consulteConceitos básicos.

Conceitos básicos do App Runner

AWS App Runneré umAWSque fornece uma maneira rápida, simples e econômica de transformar uma imagem de contêiner existente ou código-fonte diretamente em um serviço Web em execução noNuvem AWS.

Este tutorial aborda como você pode usarAWS App Runnerpara implantar seu aplicativo em um serviço App Runner. Ele percorre a configuração do código-fonte e implantação, a compilação do serviço e o tempo de execução do serviço. Ele também mostra como implantar uma versão de código, fazer uma alteração de configuração e exibir logs. Por último, o tutorial mostra como limpar os recursos que você criou ao seguir os procedimentos do tutorial.

Tópicos

Prerequisites

Antes de iniciar o tutorial, verifique se realiza as seguintes ações:

1. Siga as etapas de configuração emConfiguração.
2. Criar umGitHubConta do, caso ainda não tenha uma. Se você é novo no GitHub, consulteConceitos básicos do GitHubnoGitHub Docs.
3. Crie um repositório em sua conta do GitHub. Este tutorial usa o nome do repositório`python-hello`. Crie arquivos no diretório raiz do repositório, com os nomes e o conteúdo especificados nos exemplos a seguir.

Arquivos para o**python-hello**repositório de exemplos

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

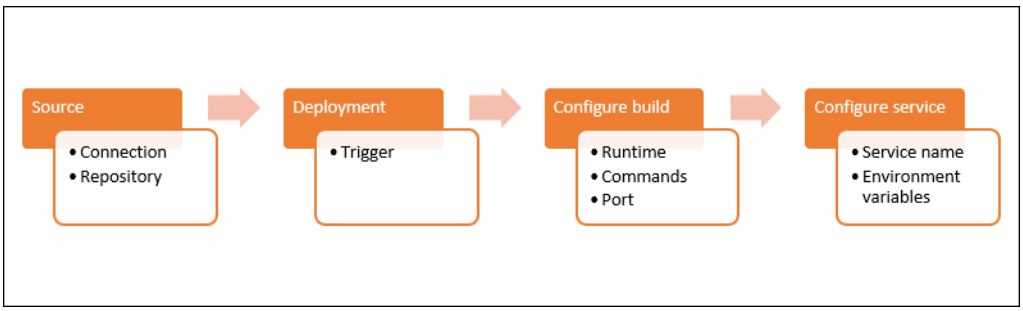
```
from flask import Flask
import os

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)
```

Etapa 1: Criar um serviço do App Runner

Nesta etapa, você cria um serviço App Runner com base no repositório de código-fonte de exemplo que você criou no GitHub como parte dothe section called “Prerequisites”. O exemplo contém um site Python simples. Estas são as principais etapas que você toma para criar um serviço:

1. Configure o código-fonte.
2. Configurar a implantação de origem.
3. Configurar a compilação do aplicativo.
4. Configure seu serviço.
5. Revise e confirme.

O diagrama a seguir descreve as etapas para criar um serviço App Runner:

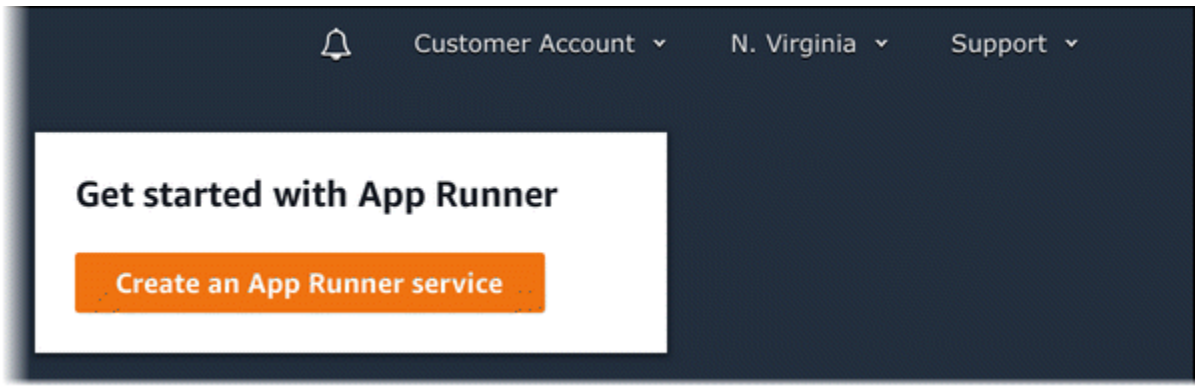


Para criar um serviço App Runner com base em um repositório de código-fonte

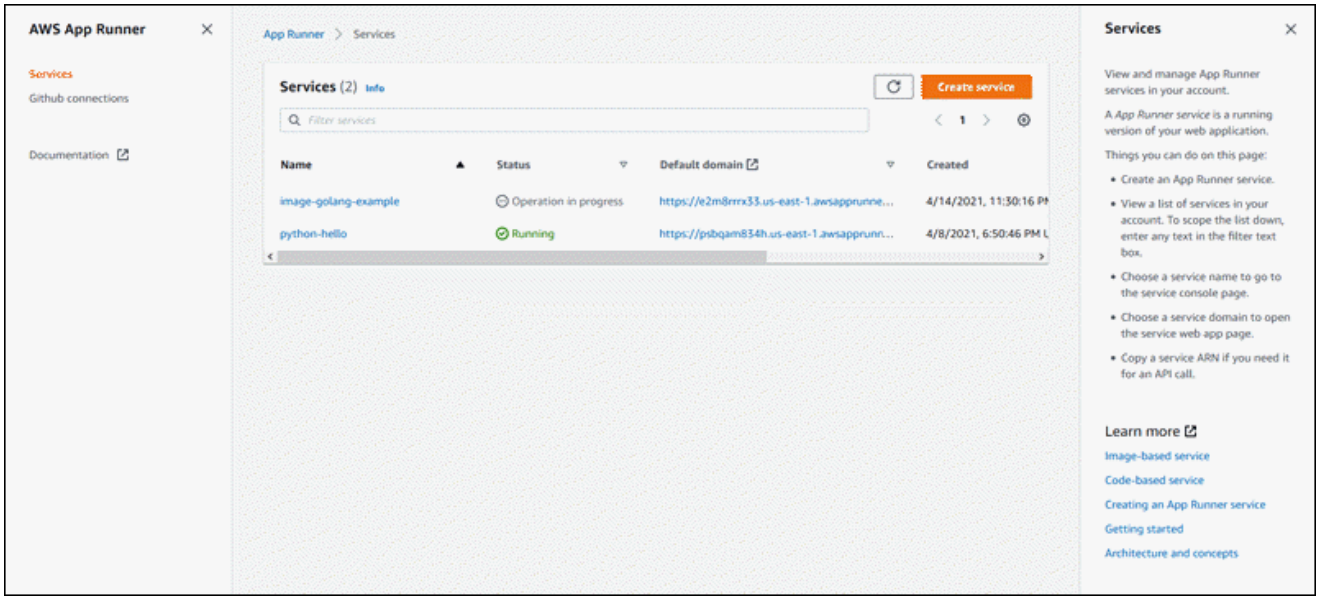
1. Configure o código-fonte.

   a. Abrir oConsole do App Runner, e noRegiões, selecione a suaRegião da AWS.

   b. Se oConta da AWSainda não tiver nenhum serviço do App Runner, a página inicial do console será exibida. SelecioneCriar um serviço do App Runner.



   Se oConta da AWStem serviços existentes, oServiçosCom uma lista dos serviços é exibida. Escolha Create service.



c. NoOrigem e implantação, noOrigem, paraRepository type (Tipo de repositório), escolharepositório de código-fonte.

d. UnderConnect ao GitHubescolhaAdicionar novoe, se solicitado, forneça suas credenciais do GitHub.

> **Note**
>
> As seguintes etapas para instalar oAWSConector do GitHub para sua conta do GitHub são etapas únicas. Você pode reutilizar a conexão para criar vários serviços do App Runner com base nos repositórios dessa conta. Quando você tiver uma conexão existente, escolha-a e pule para a seleção do repositório.

e. NoInstallAWSConector para GitHub, se solicitado, escolha o nome da conta do GitHub.

f. Se solicitado a autorizar oAWSConector para GitHub, escolhaAutorizar conexões da AWS.

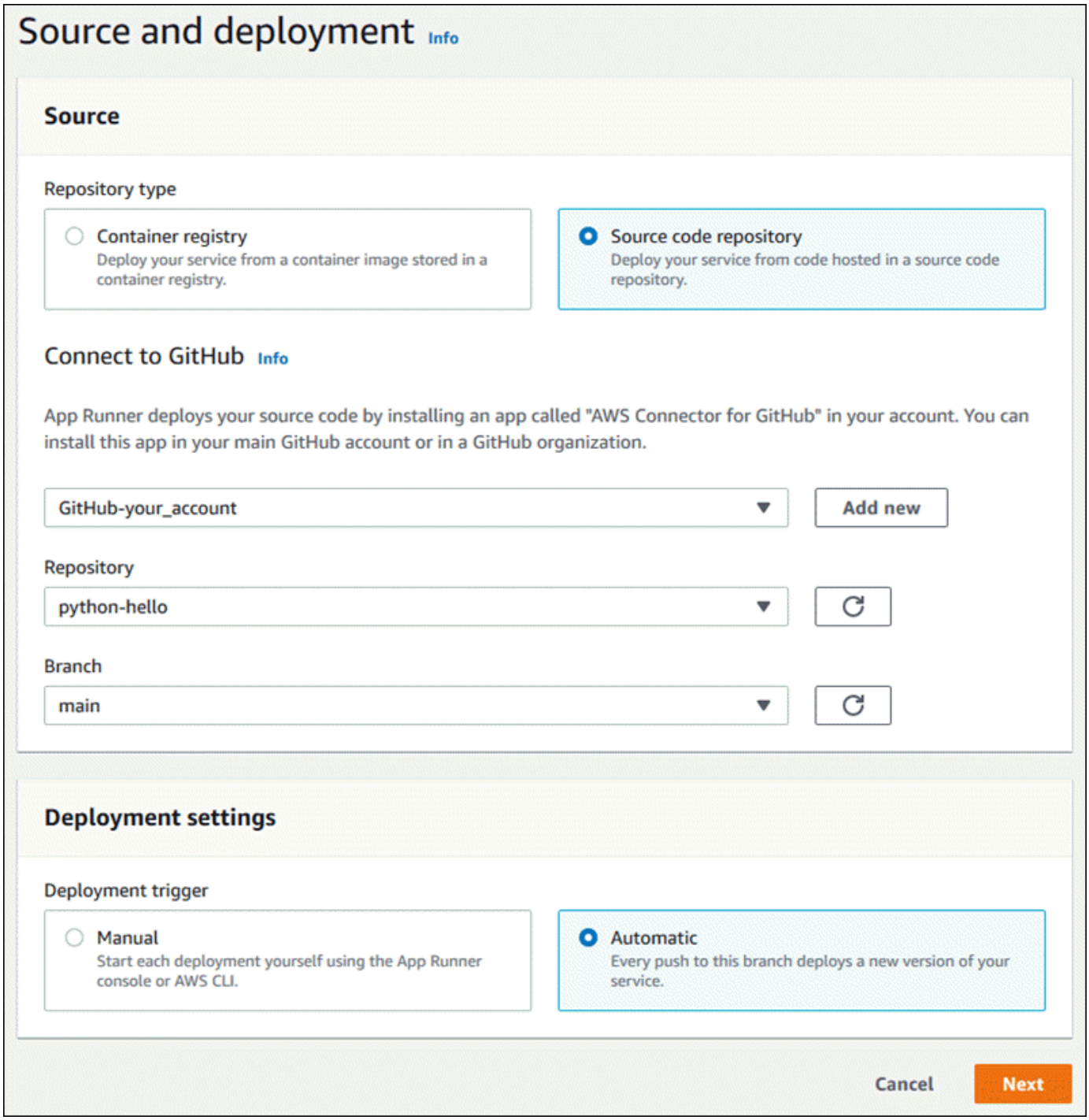
g. Escolha Install.

O nome da sua conta aparece como oConta/organização do GitHub. Agora é possível escolher um repositório na conta do.

h. para oRepositório, escolha o repositório de exemplo que você criou,`python-hello`. para oRamificação, escolha o nome da ramificação padrão do seu repositório (por exemplo,MAIN).

2. Configure suas implantações: NoConfigurações de implantação, selecioneAutomatice, depois, escolhaPróximo.

> **Note**
>
> Com a implantação automática, cada nova confirmação no repositório implanta automaticamente uma nova versão do seu serviço.



3. Configurar a compilação do aplicativo.

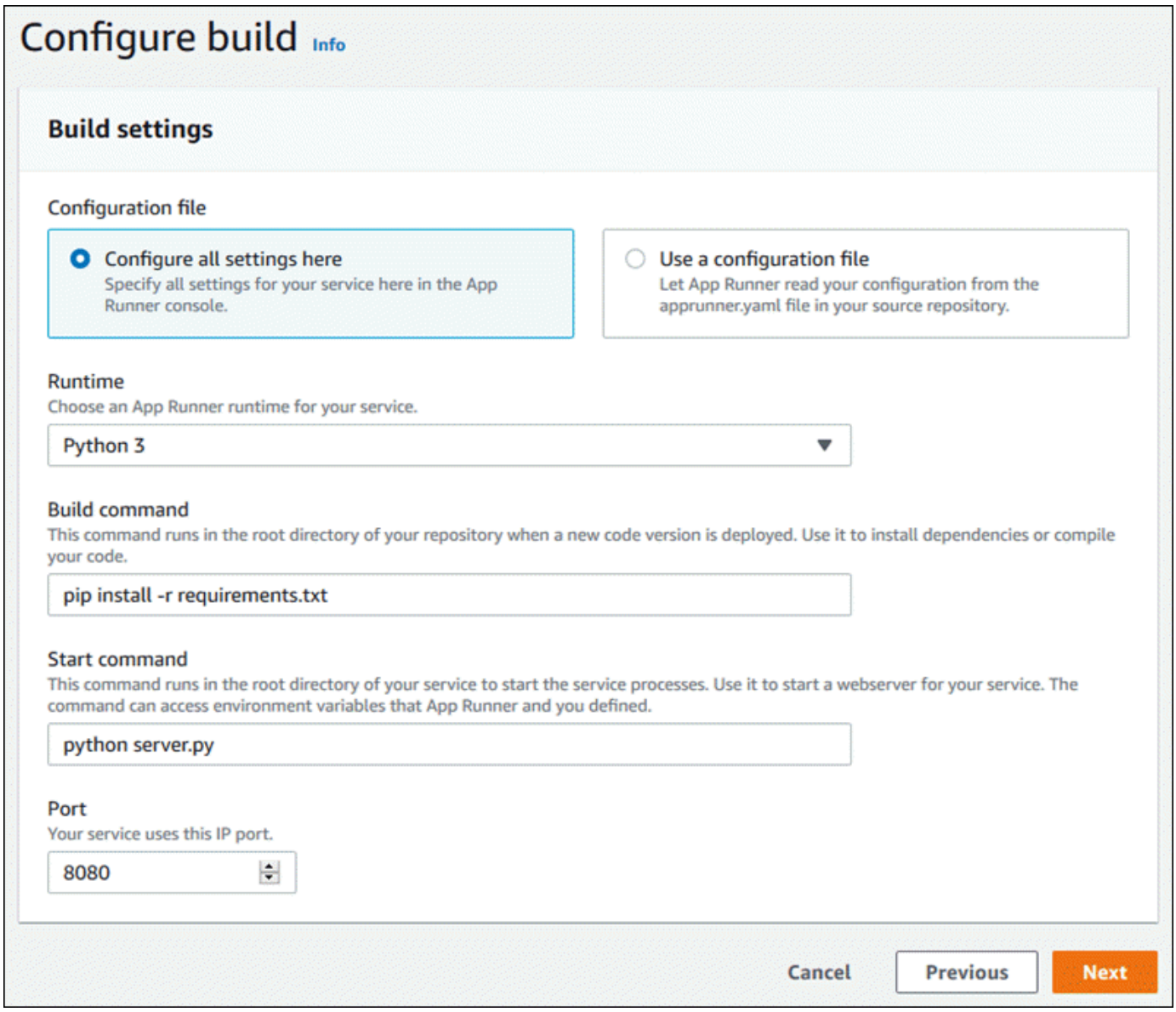
   a. NoConfigurar compilação, paraArquivo de configuração, escolhaConfigurar todas as configurações aqui.

   b. Forneça as seguintes configurações de compilação:

      - Tempo de execução— EscolhaPython 3.
      - Comando para compilação— Digite**`pip install -r requirements.txt`**.
      - Comando para iniciar— Digite**`python server.py`**.
      - Port— Digite**`8080`**.

   c. Escolha Next (Próximo).

> **Note**
>
> O tempo de execução Python 3 constrói uma imagem Docker usando uma imagem Python 3 base e seu exemplo de código Python. Em seguida, ele inicia um serviço que executa uma instância de contêiner desta imagem.



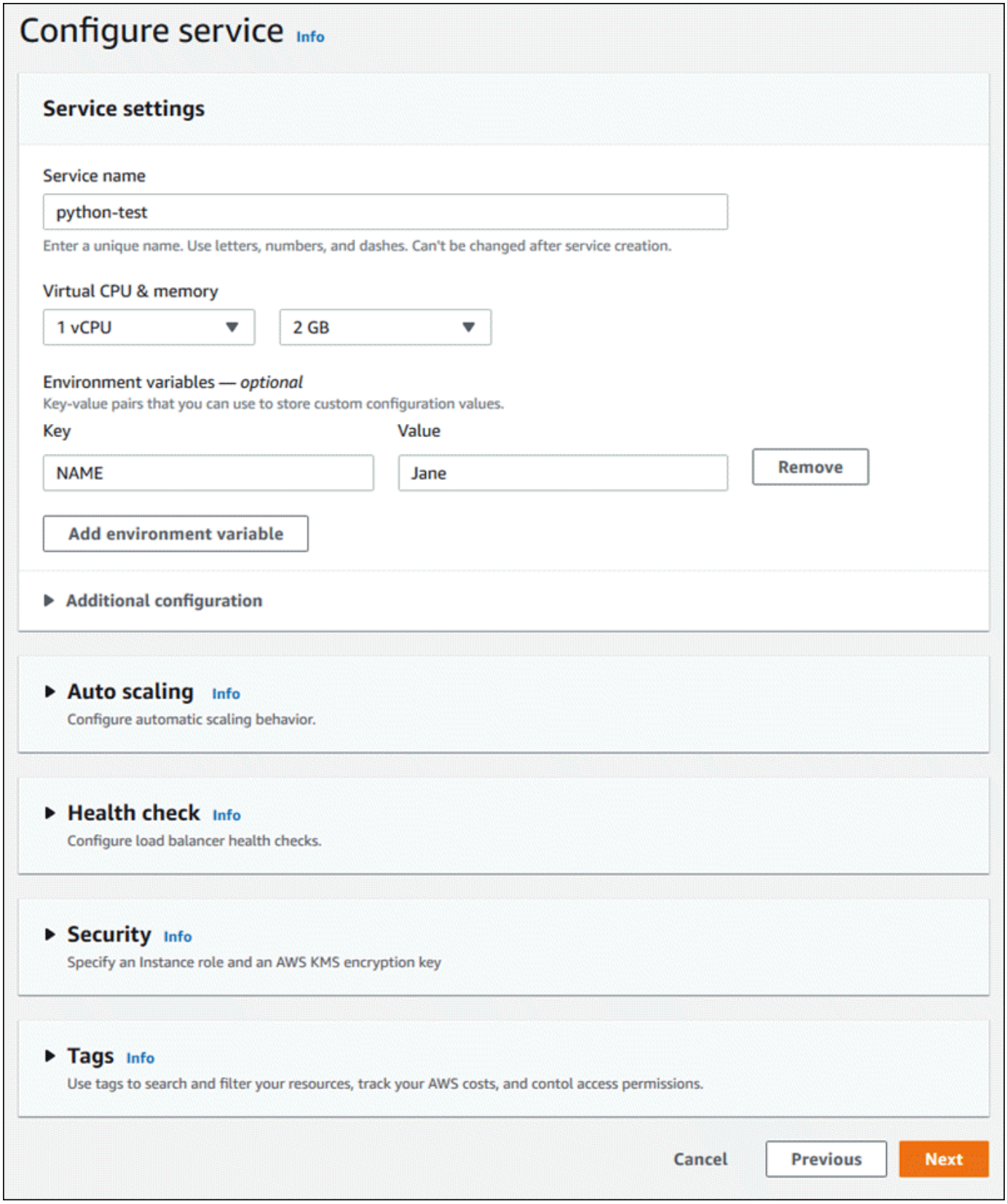
4. Configure seu serviço.

   a. NoConfigurar o serviço, noConfigurações do serviçoInsira um nome de serviço.

   b. UnderVariáveis de ambiente, adicione uma única variável de ambiente. para oKey (Chave), insira**`NAME`**, e paraValorDigite qualquer nome (por exemplo, seu nome).

> **Note**
>
> O aplicativo de exemplo lê o nome definido nesta variável de ambiente e exibe o nome em sua página da Web.

c. Escolha Next (Próximo).


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. NoRevisar e criar, verifique todos os detalhes inseridos e escolhaCriar e implantar.

   Se o serviço for criado com êxito, o console mostrará o painel de serviço, com umVisão geral do serviçodo novo serviço.

6. Verifique se o serviço está em execução.

   a. Na página do painel de serviço, aguarde até que o serviçoStatuséRunning.

   b. Selecione oDomínio padrãovalor—é a URL para o site do seu serviço.

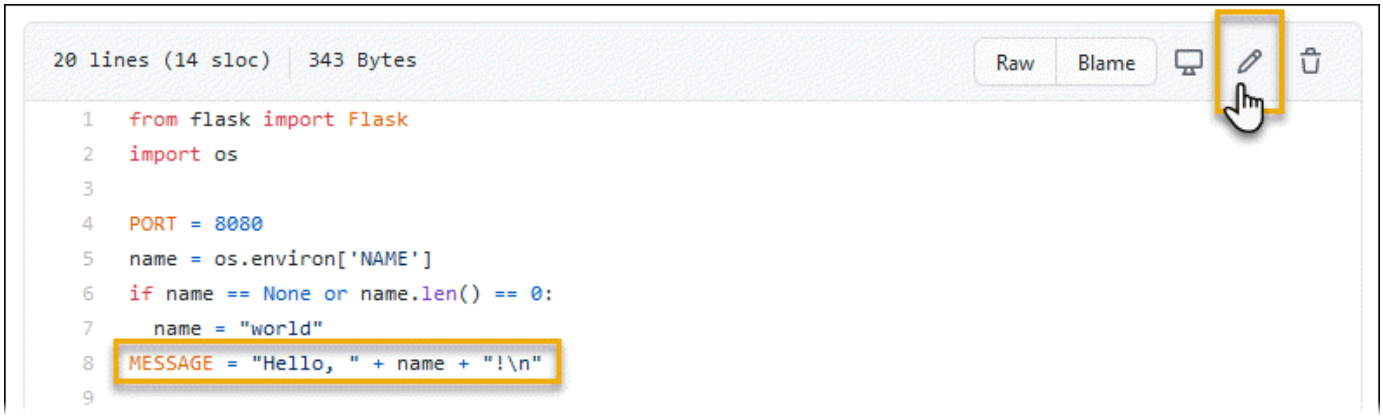
      Uma página da Web exibe: Olá,***Seu nome***!

Etapa 2: Alterar o código de serviço

Nesta etapa, você altera o repositório do código-fonte do. O recurso de CI/CD do App Runner cria e implementa automaticamente a alteração no seu serviço.

Para fazer uma alteração no código de serviço

1. Navegue até seu exemplo do repositório do GitHub.
2. Escolha o nome do arquivo`server.py`para navegar até esse arquivo.
3. SelecioneEditar este arquivo(o ícone de lápis).
4. Na expressão atribuída à variável`MESSAGE`, altere o texto`Hello`para`Good morning`.



5. Escolha Commit changes (Confirmar alterações).

   O novo commit começa a ser implantado. Na página do painel de serviço, o serviçoStatusalterações noOperação em andamento.

6. Aguarde até que a implantação seja encerrada. Na página do painel de serviço, o serviçoStatusdeve mudar de volta paraRunning.

7. Verifique se a implantação foi bem-sucedida: atualize a guia do navegador onde a página da Web do seu serviço é exibida.

   A página agora exibe a mensagem modificada: Bom dia.***Seu nome***!

Etapa 3: Fazer uma alteração de configuração

Nesta etapa, você altera o**NAME**, para demonstrar uma alteração de configuração de serviço.

Para exibir logs do serviço

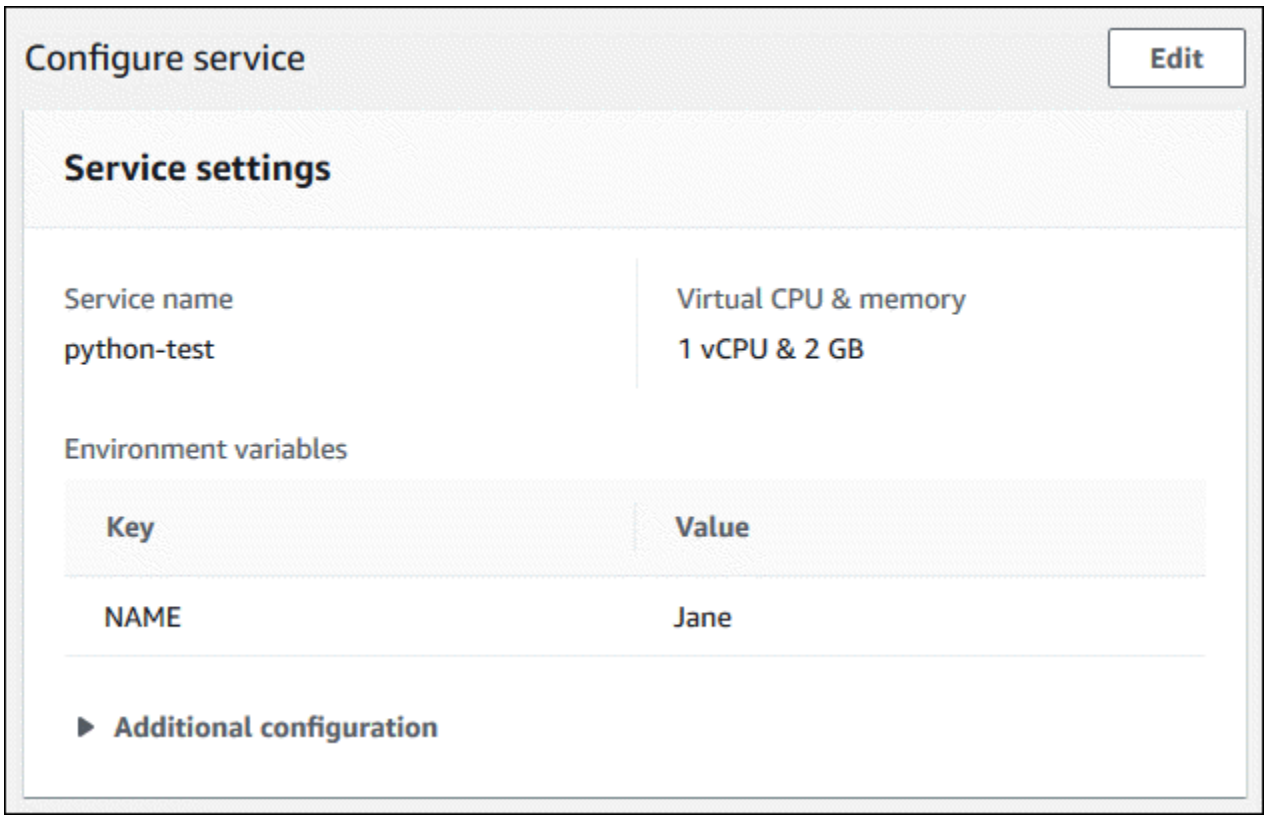
1. Abrir oConsole do App Runner, e noRegiões, selecione a suaRegião da AWS.
2. No painel de navegação, selecioneServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.

3. Na página do painel do serviço, selecione oConfiguração.

   O console exibe suas configurações de serviço em várias seções.

4. NoConfigurar o serviço, selecioneEdite.



5. Para a variável de ambiente com a chave**NAME**, altere o valor para um nome diferente.
6. Selecione Aplicar alterações.

   O App Runner inicia o processo de atualização. Na página do painel de serviço, o serviçoStatusalterações noOperação em andamento.
7. Aguarde o término da atualização. Na página do painel de serviço, o serviçoStatusdeve mudar de volta paraRunning.
8. Verifique se a atualização foi bem-sucedida: atualize a guia do navegador onde a página da Web do seu serviço é exibida.

   A página agora exibe o nome modificado: Bom dia.***Novo nome***!

Etapa 4: Ver registos para o seu serviço

Nesta etapa, você usa o console do App Runner para exibir logs do serviço App Runner. O App Runner transmite logs para o Amazon CloudWatch Logs (CloudWatch Logs) e os exibe no painel do seu serviço. Para obter informações sobre os logs do App Runner, consultethe section called “Logs (CloudWatch Logs)”.

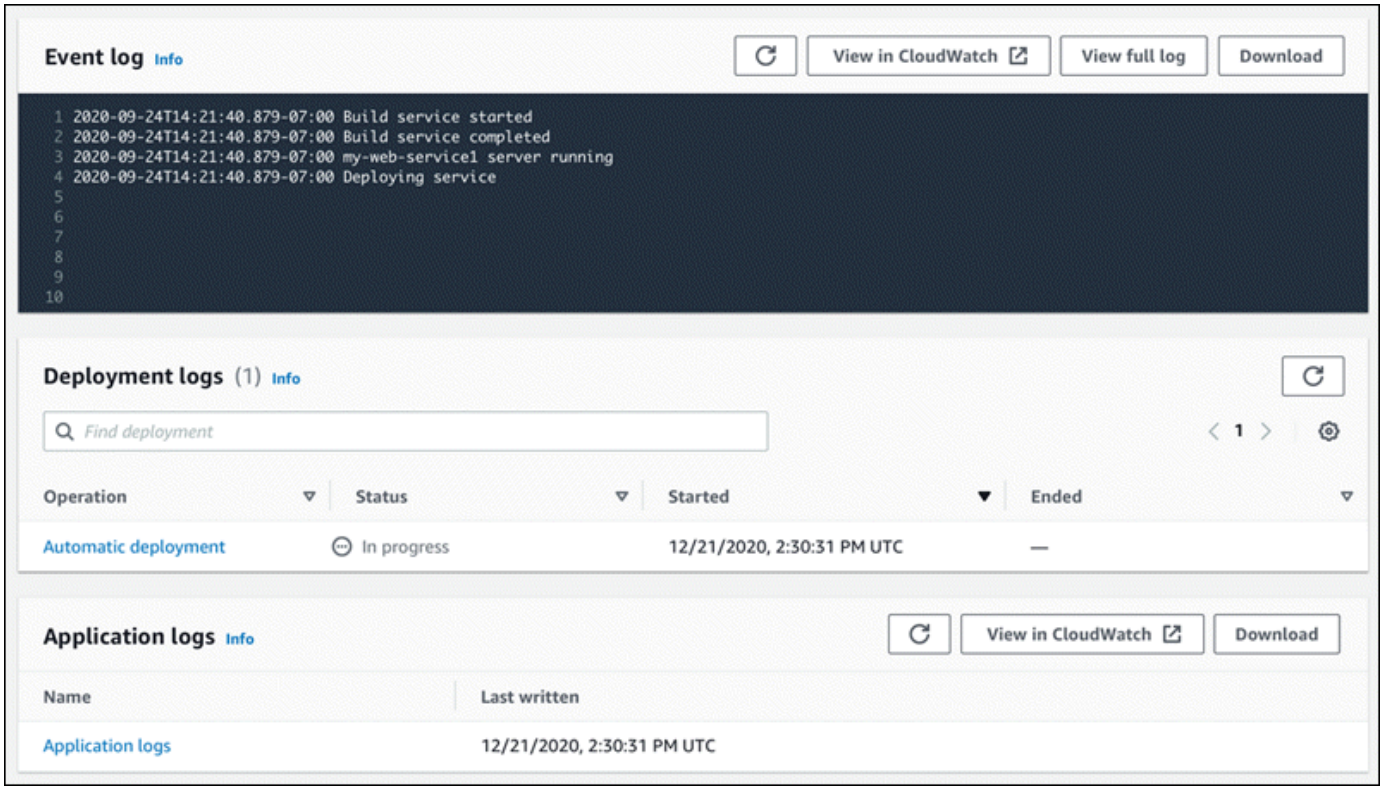
Para exibir logs do serviço

1. Abrir oConsole do App Runner, e noRegiões, selecione a suaRegião da AWS.
2. No painel de navegação, selecioneServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. Na página do painel do serviço, selecione oLogs.

   O console exibe alguns tipos de logs em várias seções:

   - Log de eventos— Atividade no ciclo de vida do seu serviço App Runner. O console exibe os eventos mais recentes.
   - Registros de implantação— Implantações de repositório de origem para o seu serviço App Runner. O console exibe um fluxo de log separado para cada implantação.
   - Logs de aplicativos— A saída do aplicativo Web que é implantado no seu serviço App Runner. O console combina a saída de todas as instâncias em execução em um único fluxo de log.



4. Para localizar implantações específicas, defina o escopo da lista de logs de implantação inserindo um termo de pesquisa. Você pode pesquisar qualquer valor exibido na tabela.

5. Para exibir o conteúdo de um log, escolhaVisualizar log completo(log de eventos) ou o nome do fluxo de log (logs de implantação e aplicativos).
6. SelecioneBaixarpara baixar um log. Para um fluxo de log de implantação, selecione primeiro um fluxo de log.
7. SelecioneVisualizar no CloudWatchpara abrir o console do CloudWatch e usar seus recursos completos para explorar seus logs de serviço do App Runner. Para um fluxo de log de implantação, selecione primeiro um fluxo de log.

> **Note**
>
> O console do CloudWatch é particularmente útil se você quiser exibir logs de aplicativos de instâncias específicas em vez do log de aplicativos combinado.

Etapa 5: Limpar

Agora você aprendeu a criar um serviço App Runner, visualizar logs e fazer algumas alterações. Nesta etapa, exclua o serviço para remover recursos que você não precisa mais.

Para excluir seu serviço

1. Na página do painel do serviço, selecioneAçõese, depois, escolhaExcluir serviço.
2. Na caixa de diálogo de confirmação, insira o texto solicitado e selecioneExcluir.

   Resultado: O console navega até oServiços. O serviço que você acabou de excluir mostra um status deDELETING. Pouco tempo depois, desaparece da lista.

Considere também excluir a conexão do GitHub que você criou como parte deste tutorial. Para mais informações, consulte the section called “Conexões”.

E o próximo

Agora que você implantou seu primeiro serviço App Runner, saiba mais nos seguintes tópicos:

- Arquitetura e conceitos— A arquitetura, os principais conceitos eAWSrecursos relacionados ao App Runner.

- Serviço baseado na imagemeServiço baseado em código— Os dois tipos de origem de aplicativo que o App Runner pode implantar.
- Desenvolvendo para App Runner— Coisas que você deve saber ao desenvolver ou migrar código de aplicativo para implantação para o App Runner.
- Console do Executor de aplicativos— Gerencie e monitore seu serviço usando o console do App Runner.
- Gerenciar seu serviço do— Gerencie o ciclo de vida do seu serviço do App Runner.
- Registro em log e monitoramento— monitore seu serviço App Runner exibindo métricas, lendo logs e rastreando chamadas de ação de serviço.
- Arquivo de configuração do App Runner— Uma maneira baseada em configuração de especificar opções para o comportamento de compilação e tempo de execução do seu serviço App Runner.
- API do Executor de aplicativos— Use a interface de programação de aplicativos (API) do App Runner para criar, ler, atualizar e excluir recursos do App Runner.
- Segurança— As diferentes maneiras queAWSe garante a segurança na nuvem enquanto utiliza o App Runner e outros serviços.

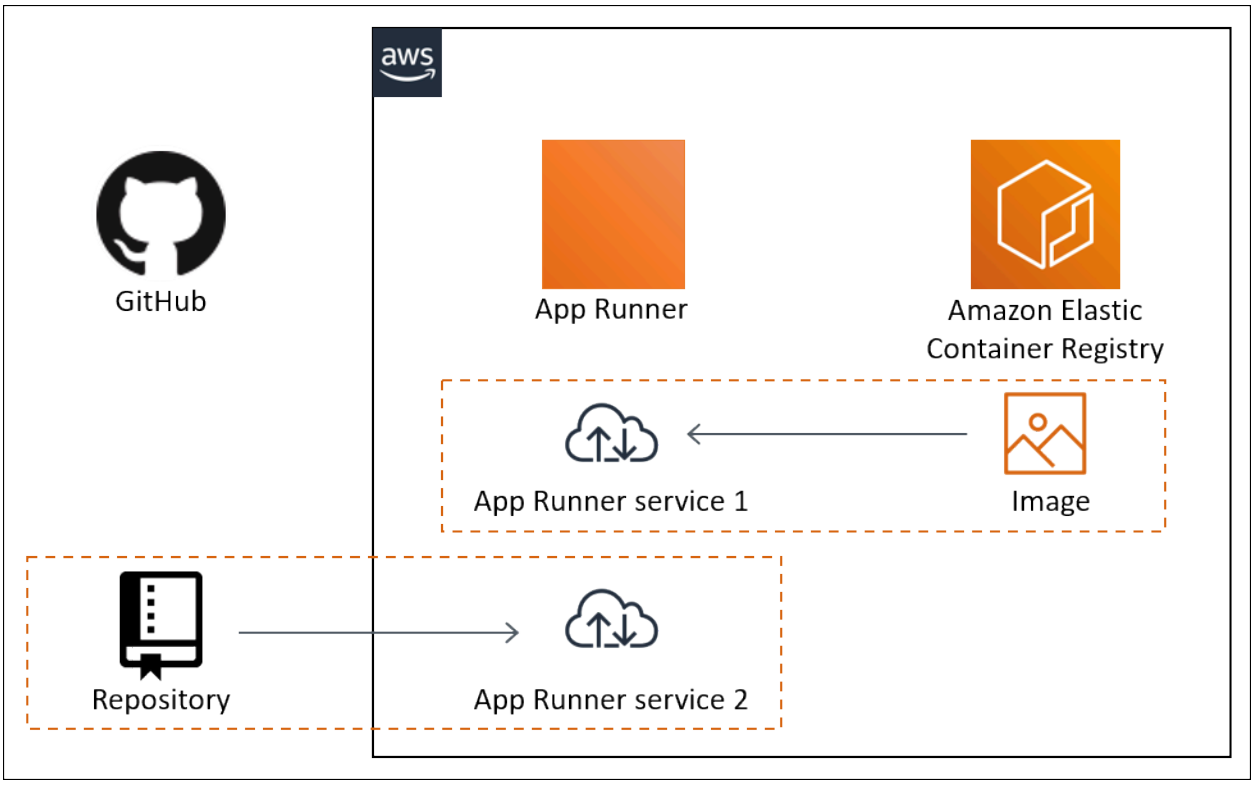
Arquitetura e conceitos do App Runner

AWS App Runnertira seu código-fonte ou imagem de origem de um repositório e cria e mantém um serviço Web em execução para você noNuvem AWS. Normalmente, você precisa chamar apenas uma ação App Runner,CreateService, para criar seu serviço.

Com um repositório de imagens de origem, você fornece uma imagem de contêiner pronta para usar que o App Runner pode implantar para executar seu serviço Web. Com um repositório de código-fonte, você pode fornecer código e instruções para criar e executar um serviço Web projetado para um dos vários ambientes de tempo de execução gerenciados pelo App Runner.

No momento, o App Runner pode recuperar seu código-fonte de umGitHubou recupere sua imagem de origem doAmazon Elastic Container Registry (Amazon ECR)Em seuConta da AWS.

O diagrama a seguir mostra uma visão geral da arquitetura do serviço App Runner. No diagrama, há dois serviços de exemplo: um implanta o código-fonte do GitHub e o outro implanta uma imagem de origem do Amazon ECR.



Conceitos do App Runner

Veja a seguir os principais conceitos relacionados ao seu serviço Web que está sendo executado no App Runner:

- Serviço App Executor— UmAWSque o App Runner usa para implantar e gerenciar seu aplicativo com base em seu repositório de código-fonte ou imagem de contêiner. Um serviço App Runner é uma versão em execução do seu aplicativo. Para obter mais informações sobre como criar um serviço, consultethe section called “Criação”.
- Tipo de origem— O tipo de repositório de origem que você fornece para implantar seu serviço App Runner:código fonteouimagem de origem.
- Provedor de repositório— O serviço de repositório que contém a origem do aplicativo (por exemplo,GitHubouAmazon ECR).
- Conexão App Executor— UmAWSque permite que o App Runner acesse uma conta de provedor de repositório (por exemplo, uma conta ou organização do GitHub). Para obter mais informações sobre conexões, consulte the section called “Conexões”.
- Tempo de execução— Uma imagem base para implantar um repositório de código-fonte. App Runner fornece uma variedade deTempos de execução dopara diferentes ambientes de programação. Para mais informações, consulte Serviço baseado em código.
- Implantação— Uma ação que aplica uma versão do repositório de origem (código ou imagem) a um serviço do App Runner. A primeira implantação no serviço ocorre como parte da criação do serviço. Implantações posteriores podem ocorrer de duas formas:
  - Implantação automática— Um recurso CI/CD. Você pode configurar um serviço App Runner para compilar automaticamente (para código-fonte) e implantar cada versão do aplicativo conforme ele aparece no repositório. Isso pode ser um novo commit em um repositório de código-fonte ou uma nova versão de imagem em um repositório de imagem de origem.
  - Implantação manual— Uma implantação em seu serviço App Runner que você inicia explicitamente.
- Domínio personalizado— Um domínio que você associa ao seu serviço App Runner. Os utilizadores da sua aplicação Web podem utilizar este domínio para aceder ao seu serviço Web em vez do subdomínio predefinido do App Runner. Para mais informações, consulte the section called “Nomes de domínios personalizados”.
- Manutenção— Uma atividade que o App Runner realiza ocasionalmente na infraestrutura que executa seu serviço App Runner. Quando a manutenção está em andamento, o status do serviço muda temporariamente para`OPERATION_IN_PROGRESS`(Operação em andamentoNo console) por

alguns minutos. As ações em seu serviço (por exemplo, implantação, atualização de configuração, pausar/retomar ou exclusão) são bloqueadas durante esse período. Tente a ação novamente alguns minutos depois, quando o status do serviço retornar ao`RUNNING`.

> **Note**
>
> Se a ação falhar, isso não significa que o serviço App Runner esteja inativo. Seu aplicativo está ativo e continua processando solicitações. É improvável que seu serviço experimente algum tempo de inatividade.

Em particular, o App Runner migra seu serviço se detectar problemas no hardware subjacente que hospeda o serviço. Para evitar qualquer tempo de inatividade do serviço, o App Runner implanta seu serviço em um novo conjunto de instâncias e muda o tráfego para elas (uma implantação azul-verde). Você pode ocasionalmente ver um ligeiro aumento temporário nas cobranças.

Recursos do App Executor

Ao usar o App Runner, você cria e gerencia alguns tipos de recursos noConta da AWS. Esses recursos são usados para acessar seu código e gerenciar seus serviços.

A tabela a seguir fornece uma visão geral desses recursos:

Resource name (Nome do recurso)	Descrição
Service	Representa uma versão em execução do seu aplicativo. Grande parte do restante deste guia descreve tipos de serviço, gerenciamento, configuração e monitoramento. Nome de região da Amazon (ARN): `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:service/` *`service-name`* `[/`*`service-id`* `]`
Connection	Fornece aos serviços do App Runner acesso a repositórios privados armazenados com provedores terceirizados. Existe como um recurso separado para compartilhamento entre vários serviços. Para obter mais informações sobre conexões, consulte the section called “Conexões”.

Resource name (Nome do recurso)	Descrição
	Nome de região da Amazon (ARN): `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:connection/` *`connection-name`*`[/`*`connection-id`* `]`
AutoScalingConfiguration	Fornece aos seus serviços App Runner definições que controlam o dimensionamento automático da sua aplicação. Existe como um recurso separado para compartilhamento entre vários serviços. Para obter mais informações sobre o dimensionamento automático, consultethe section called “Auto Scaling”. Nome de região da Amazon (ARN): `arn:aws:apprunner:` *`region`*`:`*`account-id`* `:autoscalingconfiguration/` *`config-name`* `[/`*`config-revision`* `[/`*`config-id`* `]]`

Cotas de recurso App Executor

AWSimpõe algumas cotas (também conhecidas como limites) à sua conta paraAWSUso de recursos em cadaRegião da AWS. A tabela a seguir lista as cotas relacionadas aos recursos do Executor de aplicativos. Cotas também estão listadas emAWS App RunnerEndpoints e cotas donoAWSReferência geral.

Cota de recursos	Descrição	Valor padrão	Ajustável?
Services	O número máximo de serviços que você pode criar em sua conta para cadaRegião da AWS.	10	✓ Sim
Connections	O número máximo de conexões que você pode criar em sua conta para cadaRegião da AWS. Você pode compartilhar uma única conexão entre vários serviços.	10	✓ Sim

Cota de recursos	Descrição	Valor padrão	Ajustável?
Auto scaling configurations—names	O número máximo de nomes exclusivos que você pode ter em configurações de auto scaling que você cria em sua conta para cadaRegião da AWS. Você pode usar uma configuração de Auto Scaling em vários serviços.	10	✓ Sim
Auto scaling configurations—revisões para cada nome	O número máximo de revisões de configuração do Auto Scaling que você pode criar em sua conta para cadaRegião da AWSPara cada nome exclusivo. Você pode usar uma revisão de configuração de Auto Scaling em vários serviços.	10	× Não

A maioria das cotas são ajustáveis e você pode solicitar um aumento da cota para elas. Para obter mais informações, consulteSolicitar um aumento de cotaNo Guia do usuário do Service Quotas.

Serviço App Runner baseado em uma imagem de origem

Você pode usar oAWS App Runnerpara criar e gerenciar serviços com base em dois tipos fundamentalmente diferentes de fonte de serviço:código fonteeimagem de origem. Independentemente do tipo de origem, o App Runner cuida de iniciar, executar, dimensionar e balancear a carga do seu serviço. Você pode usar o recurso CI/CD do App Runner para rastrear alterações em sua imagem de origem ou código. Quando o App Runner descobre uma alteração, ele cria automaticamente (para o código-fonte) e implanta a nova versão no seu serviço App Runner.

Este capítulo discute serviços baseados em uma imagem de origem. Para obter informações sobre serviços baseados no código-fonte, consulteServiço baseado em código.

Aimagem de origemé uma imagem de contêiner pública ou privada armazenada em um repositório de imagens. Você aponta o App Runner para uma imagem e ele inicia um serviço executando um contêiner com base nessa imagem. Nenhuma etapa de construção é necessária. Em vez disso, você fornece uma imagem pronta para implantar.

Provedores de repositório

O App Runner oferece suporte aos seguintes provedores de repositório de imagens:

- Amazon Elastic Container Registry (Amazon ECR)— Armazena imagens privadas no seuConta da AWS.
- Amazon Elastic Container Registry Public (Amazon ECR Public)— Armazena imagens publicamente legíveis.

Implantação do Amazon ECR

Amazon ECRarmazena imagens em repositórios. Existem repositórios privados e públicos. Para implantar sua imagem em um serviço App Runner a partir de um repositório privado, o App Runner precisa de permissão para ler sua imagem no Amazon ECR. Para dar essa permissão ao App Runner, você precisa fornecer ao App Runner umFunção de acesso ao. Este é umAWS Identity and Access ManagementFunção do (IAM) que tem as permissões de ação do Amazon ECR necessárias. Quando utiliza a consola App Runner para criar o serviço, pode escolher uma função existente na sua conta. Como alternativa, você pode usar o console do IAM para criar uma nova função personalizada ou escolher o console do App Runner para criar uma função para você com base em políticas gerenciadas.

Quando você usa a API do App Runner ou oAWS CLI, você conclui um processo de duas etapas. Primeiro, você usa o console do IAM para criar uma função de acesso. Você pode usar uma política gerenciada fornecida pelo App Runner ou inserir suas próprias permissões personalizadas. Em seguida, você fornece a função de acesso durante a criação do serviço usando oCreateServiceAção de API.

Para obter informações sobre a criação do serviço App Runner, consultethe section called “Criação”.

Implantação do Amazon ECR Public

Amazon ECRarmazena imagens publicamente legíveis. Estas são as principais diferenças entre o Amazon ECR e o Amazon ECR Public que você deve estar ciente no contexto dos serviços do App Runner:

- As imagens públicas do Amazon ECR são legíveis publicamente. Você não precisa fornecer uma função de acesso ao criar um serviço baseado em uma imagem pública do Amazon ECR.
- O App Runner não oferece suporte à implantação automática para imagens públicas do Amazon ECR.

Serviço App Runner baseado no código-fonte

Você pode usar oAWS App Runnerpara criar e gerenciar serviços com base em dois tipos fundamentalmente diferentes de fonte de serviço:código fonteeimagem de origem. Independentemente do tipo de origem, o App Runner cuida de iniciar, executar, dimensionar e balancear a carga do seu serviço. Você pode usar o recurso CI/CD do App Runner para rastrear alterações em sua imagem de origem ou código. Quando o App Runner descobre uma alteração, ele cria automaticamente (para o código-fonte) e implanta a nova versão no seu serviço App Runner.

Este capítulo discute serviços baseados no código-fonte. Para obter informações sobre serviços baseados em uma imagem de origem, consulteServiço baseado na imagem.

O código-fonte é o código do aplicativo que o App Runner cria e implementa para você. Você aponta o App Runner para um repositório de código-fonte e escolhe umtempo de execução. O App Runner cria uma imagem baseada na imagem base do tempo de execução e no código do aplicativo. Em seguida, ele inicia um serviço que executa um contêiner com base nessa imagem.

App Runner fornece conveniente idioma específicoTempos de execução do. Cada um desses tempos de execução cria uma imagem de contêiner a partir de seu código-fonte e adiciona dependências de tempo de execução de idioma em sua imagem. Você não precisa fornecer instruções de configuração de contêiner e de compilação, como um arquivo de dockerfile.

Subtópicos deste capítulo discutem os váriosTempos de execução doque o App Runner suporta — o tempo de execução genérico do Dockerfile e oTempos de execução dopara diferentes ambientes de programação.

Tópicos

- Fornecedores de repositório de código-
- Tempo de execução gerenciado do App Runner
- Usando o tempo de execução gerenciado Python
- Usando o tempo de execução gerenciado Node.js

Fornecedores de repositório de código-

O App Runner implanta seu código-fonte lendo-o a partir de um repositório de código-fonte. O App Runner suporta um provedor de repositório de código-fonte: GitHub.

Implantando a partir do GitHub

Para implantar o código-fonte em um serviço do App Runner de umGitHub, o App Runner estabelece uma conexão com o GitHub. Se seu repositório for privado (ou seja, ele não é acessível publicamente no GitHub), você deve fornecer detalhes de conexão ao App Runner. Quando utiliza a consola App Runner paraCriar um serviço, você fornece detalhes de conexão como parte do procedimento de criação do serviço.

Quando você usa a API do App Runner ou oAWS CLI, uma conexão é um recurso separado. Primeiro, crie a conexão usando oCreateConnectionAção de API. Em seguida, você fornece o ARN da conexão durante a criação do serviço usando oCreateServiceAção de API.

Para obter mais informações sobre a criação do serviço do App Runner, consultethe section called “Criação”. Para obter mais informações sobre conexões do App Runner, consultethe section called “Conexões”.

Tempo de execução gerenciado do App Runner

O App Runner fornece tempos de execução gerenciados para vários ambientes de programação. Cada tempo de execução gerenciado facilita a criação e a execução de contêineres com base em uma determinada linguagem de programação ou ambiente de tempo de execução. Quando você usa um tempo de execução gerenciado, o App Runner começa com uma imagem de tempo de execução gerenciada. Esta imagem é baseada noimagem de docker do Amazon Linuxe contém um pacote de tempo de execução de linguagem, bem como algumas ferramentas e pacotes de dependência populares. O App Runner usa essa imagem de tempo de execução gerenciada como uma imagem base e adiciona o código do aplicativo para criar uma imagem do Docker. Em seguida, ela implanta essa imagem para executar seu serviço Web em um contêiner.

Você especifica um tempo de execução para o serviço App Runner quandoCriar um serviçousando o console do App Runner ou oCreateServiceAPI. Você também pode especificar um tempo de execução como parte do código-fonte. Usar `aruntime`Palavra-chave em umArquivo de configuração do App Runnerque você incluir em seu repositório de código. A convenção de nomeação de um tempo de execução gerenciado é*`<language-name> <major-version>`*.

O App Runner atualiza o tempo de execução do seu serviço para a versão mais recente em cada implantação ou atualização de serviço. Se o aplicativo exigir uma versão específica de um tempo de execução gerenciado, você pode especificá-lo usando `oruntime-version`Palavra-chave noArquivo de configuração do App Runner. Especifique uma versão

secundária como*<major>*.*<minor>*para bloquear as versões principal e secundária (o App Runner atualiza apenas versões de patch). Especifique um nível de patch específico como*<major>*.*<minor>*.*<patch>*para bloquear seu serviço em uma versão específica do tempo de execução (App Runner nunca atualiza o tempo de execução).

Usando o tempo de execução gerenciado Python

AWS App Runnerfornece um tempo de execução gerenciado Python. O tempo de execução facilita a compilação e a execução de contêineres com aplicativos da Web baseados em Python. Quando você usa o tempo de execução Python, o App Runner começa com uma imagem de tempo de execução Python gerenciada. Esta imagem é baseada noImagem do Amazon Linux Dockere contém o pacote de tempo de execução Python e algumas ferramentas e pacotes de dependência populares. O App Runner usa essa imagem de tempo de execução gerenciada como uma imagem base e adiciona o código do aplicativo para criar uma imagem do Docker. Em seguida, ela implanta essa imagem para executar seu serviço Web em um contêiner.

Você especifica um tempo de execução para o serviço App Runner quandoCriar um serviçousando o console do App Runner ou oCreateServiceAPI DO. Você também pode especificar um tempo de execução como parte do código-fonte. Usar `aruntime`Palavra-chave em umArquivo de configuração do App Runnerque você incluir em seu repositório de código. A convenção de nomeação de um tempo de execução gerenciado é*<language-name> <major-version>*.

Para obter nomes válidos de tempo de execução Python, consultethe section called “Informações sobre o lançamento”.

O App Runner atualiza o tempo de execução do seu serviço para a versão mais recente em cada implantação ou atualização de serviço. Se o aplicativo exigir uma versão específica de um tempo de execução gerenciado, você pode especificá-lo usando o`runtime-version`Palavra-chave noArquivo de configuração do App Runner. Especifique uma versão secundária como*<major>*.*<minor>*para bloquear as versões principal e secundária (o App Runner atualiza apenas versões de patch). Especifique um nível de patch específico como*<major>*.*<minor>*.*<patch>*para bloquear seu serviço em uma versão específica do tempo de execução (App Runner nunca atualiza o tempo de execução).

Tópicos

- Configuração do tempo de execução Python
- Exemplos de execução do Python

- Informações sobre a versão do tempo de execução

Configuração do tempo de execução Python

Ao escolher um tempo de execução gerenciado, você também deve configurar, no mínimo, comandos de compilação e execução. Você os configura enquantocreatingouatualizarseu serviço App Runner. Há algumas maneiras de fazer isso:

- Usando o console do App Runner— Especifique os comandos na caixaConfigurar compilaçãodo processo de criação ou guia de configuração.
- Usando a API do App RunnerChameCreateServiceouUpdateService. Especifique os comandos usando o`BuildCommand`e`StartCommand`Membros doCodeConfigurationValuestipo de dados.
- Usando umaArquivo de configuração do— especifique um ou mais comandos de compilação em até três fases de compilação e um único comando de execução que sirva para iniciar o aplicativo. Existem definições de configuração opcionais adicionais.

Fornecer um arquivo de configuração é opcional. Ao criar um serviço App Runner usando o console ou a API, você especifica se o App Runner obtém suas configurações diretamente durante a criação ou a partir de um arquivo de configuração.

Exemplos de execução do Python

Os exemplos a seguir mostram arquivos de configuração do App Runner para criar e executar um serviço Python. O último exemplo é o código-fonte para um aplicativo Python completo que você pode implantar em um serviço de tempo de execução Python.

Arquivo de configuração do Python mínimo

Este exemplo mostra um arquivo de configuração mínimo que você pode usar com o runtime gerenciado Python. Para obter as suposições que o App Runner faz com um arquivo de configuração mínimo, consultethe section called “Exemplo de arquivo de configuração”.

Example apprunner.yaml

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

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

Arquivo de configuração do Python estendido

Este exemplo mostra o uso de todas as chaves de configuração com o tempo de execução gerenciado 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"
```

Fonte do aplicativo Python de ponta a ponta

Este exemplo mostra o código-fonte de um aplicativo Python completo que você pode implantar em um serviço de tempo de execução 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
```

Informações sobre a versão do tempo de execução

Este tópico lista os detalhes completos das versões de tempo de execução do Python suportadas pelo App Runner.

Python 3

Detalhes	Descrição
Nome do tempo de execução	Python3
Versões secundárias	3,7, 3,8
Pacotes incluídos	python, pip, setuptools, roda, virtualenv

Usando o tempo de execução gerenciado Node.js

AWS App Runnerfornece um tempo de execução gerenciado Node.js. O tempo de execução facilita a compilação e a execução de contêineres com aplicativos da Web baseados em Node.js. Quando você usa o tempo de execução Node.js, o App Runner começa com uma imagem de tempo de execução Node.js gerenciada. Esta imagem é baseada noImagem de docker do Amazon Linuxe contém o pacote de tempo de execução Node.js e algumas ferramentas. O App Runner usa essa imagem de tempo de execução gerenciada como uma imagem base e adiciona o código do aplicativo para criar uma imagem do Docker. Em seguida, ela implanta essa imagem para executar seu serviço Web em um contêiner.

Você especifica um tempo de execução para o serviço App Runner quandoCriar um serviçousando o console do App Runner ou oCreateServiceAPI. Você também pode especificar um tempo de execução como parte do código-fonte. Usar `aruntime`Palavra-chave em umArquivo de configuração do App Runnerque você incluir em seu repositório de código. A convenção de nomeação de um tempo de execução gerenciado é*<language-name> <major-version>*.

Para obter nomes válidos de runtime Node.js, consultethe section called “Informações sobre a versão”.

O App Runner atualiza o tempo de execução do seu serviço para a versão mais recente em cada implantação ou atualização de serviço. Se o aplicativo exigir uma versão específica

de um tempo de execução gerenciado, você pode especificá-lo usando `oruntime-version`Palavra-chave noArquivo de configuração do App Runner. Especifique uma versão secundária como*<major>*.*<minor>*para bloquear as versões principal e secundária (o App Runner atualiza apenas versões de patch). Especifique um nível de patch específico como*<major>*.*<minor>*.*<patch>*para bloquear seu serviço em uma versão específica do tempo de execução (App Runner nunca atualiza o tempo de execução).

Tópicos

- Configuração do Node.js
- Exemplos do Node.js
- Informações sobre a versão do Node.js

Configuração do Node.js

Ao escolher um tempo de execução gerenciado, você também deve configurar, no mínimo, comandos de compilação e execução. Você os configura enquantocreatingouatualizarseu serviço App Runner. Existem algumas maneiras de fazê-lo:

- Usando o console do App Runner— Especifique os comandos na caixaConfigurar compilaçãodo processo de criação ou guia de configuração.
- Usando a API do App Runner— ChameCreateServiceouUpdateService. Especifique os comandos usando o`BuildCommand`e`StartCommand`Membros doCodeConfigurationValuestipo de dados.
- Usando umaArquivo de configuração— especifique um ou mais comandos de compilação em até três fases de compilação e um único comando de execução que sirva para iniciar o aplicativo. Existem definições de configuração opcionais adicionais.

Fornecer um arquivo de configuração é opcional. Ao criar um serviço App Runner usando o console ou a API, você especifica se o App Runner obtém suas configurações diretamente durante a criação ou a partir de um arquivo de configuração.

Com o tempo de execução Node.js especificamente, você também pode configurar a compilação e o tempo de execução usando um arquivo JSON chamado`package.json`na raiz do repositório de origem. Usando esse arquivo, você pode configurar a versão do mecanismo Node.js, pacotes de dependência e vários comandos (aplicativos de linha de comando). Os gerenciadores de pacotes, como npm ou yarn, interpretam este arquivo como entrada para seus comandos.

Por exemplo:

- npm installinstala pacotes definidos pelodependenciesedevDependenciesNó dopackage.json.
- npm startounpm run startexecuta o comando definido peloscripts/startNó dopackage.json.

Veja a seguir um exemplo de arquivo 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"
  }
}
```

Para obter mais informações sobrepackage.json, consulteGuia package.jsonno site do Node.js.

Tips

- Se suas receitaspackage.jsondefine um arquivostart, você pode usá-lo como umrunno arquivo de configuração do App Runner, como mostra o exemplo a seguir.

Example

package.json

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

apprunner.yaml

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

- Quando você executanpm installem seu ambiente de desenvolvimento, o npm cria o arquivo`package-lock.json`. Este arquivo contém um instantâneo das versões do pacote que o npm acabou de ser instalado. Posteriormente, quando o npm instala dependências, ele usa essas versões exatas. Da mesma forma, o fio cria`yarn.json`. Submeta esses arquivos no repositório de código-fonte para garantir que seu aplicativo esteja instalado com as versões de dependências com as quais você desenvolveu e testou.
- Você também pode usar um arquivo de configuração do App Runner para configurar a versão Node.js e o comando start. Ao fazer isso, essas definições substituem as do`package.json`. Um conflito entre onodeVersão do`package.json`O e `aruntime-version`no arquivo de configuração do App Runner faz com que a fase de compilação do App Runner falhe.

Exemplos do Node.js

Os exemplos a seguir mostram arquivos de configuração do App Runner para criar e executar um serviço Node.js.

Arquivo de configuração do Node.js

Este exemplo mostra um arquivo de configuração mínimo que você pode usar com o tempo de execução gerenciado Node.js. Para obter as suposições que o App Runner faz com um arquivo de configuração mínimo, consultethe section called “Exemplo de arquivo de configuração”.

Example apprunner.yaml

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

Arquivo de configuração do Node.js

Este exemplo mostra o uso de todas as chaves de configuração com o tempo de execução gerenciado 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"
```

Aplicativo Node.js com o Grunt

Este exemplo mostra como configurar um aplicativo Node.js desenvolvido com o Grunt. Gruntté um executor de tarefas JavaScript de linha de comando. Ele executa tarefas repetitivas e gerencia a automação do processo para reduzir o erro humano. Os plugins Grunt e Grunt são instalados e gerenciados usando o npm. Você configura o Grunt incluindo o`Gruntfile.js`Arquivo na raiz do repositório de origem.

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
```

Informações sobre a versão do Node.js

Este tópico lista os detalhes completos das versões de tempo de execução Node.js suportadas pelo App Runner.

Node.js 12

Detalhes	Descrição
Nome do tempo de execução	nodejs12
Versões secundárias	latest
Pacotes incluídos	nodejs (incluindo npm), fios

Desenvolvendo código de aplicativo para App Runner

Este capítulo discute informações de tempo de execução e diretrizes de desenvolvimento que você deve considerar ao desenvolver ou migrar código de aplicativo para implantação paraAWS App Runner.

Informações de tempo de

Se você fornecer uma imagem de contêiner ou o App Runner cria uma para você, o App Runner executa seu código de aplicativo em uma instância de contêiner. Aqui estão alguns aspectos principais do ambiente de tempo de execução da instância do contêiner.

- Suporte do framework— O App Runner é compatível com qualquer imagem que implementa um aplicativo web. É agnóstico para a linguagem de programação que você escolher e para o servidor de aplicativos web ou framework que você usa, se você usar qualquer. Para sua conveniência, fornecemos tempos de execução gerenciados específicos do idioma para simplificar o processo de criação de aplicativos e a criação de imagens abstratas.
- Solicitações da Web— Sua instância de contêiner deve escutar solicitações HTTP, na porta 8080 por padrão. Para obter mais informações sobre como configurar seu serviço, consultethe section called “Configuração”. Não é necessário implementar o processamento do tráfego seguro HTTPS. O App Runner requer tráfego HTTPS de entrada e encerra HTTPS antes de passar solicitações para sua instância de contêiner.
- Aplicativos stateseless— O App Runner não garante a persistência do estado além da duração do processamento de uma única solicitação da Web recebida.
- Armazenamento— O App Runner implementa o sistema de arquivos em sua instância de contêiner comoarmazenamento temporário. Os arquivos são transitórios. Por exemplo, elas não persistem quando você pausar e retomar seu serviço do App Runner. De forma mais geral, os arquivos não têm garantia de persistir além do processamento de uma única solicitação, como parte da natureza sem estado de seu aplicativo. No entanto, os arquivos armazenados ocupam parte da alocação de armazenamento do seu serviço App Runner durante a vida útil.

> **Note**
>
> Embora os arquivos de armazenamento efêmero possam não persistir em todas as solicitações, elesÀs vezesPersistir. Isso pode ser útil em certas situações. Por exemplo, ao lidar com uma solicitação, você pode armazenar em cache os arquivos que seu

aplicativo baixa se solicitações futuras precisarem deles. Isso pode acelerar o tratamento de solicitações futuras, mas não pode garantir os ganhos de velocidade. Seu código não deve assumir que um arquivo que foi baixado em uma solicitação anterior ainda existe. Para cache garantido usando um armazenamento de dados na memória de alta taxa de transferência e baixa latência, use um serviço comoAmazon ElastiCache.

- Variáveis de ambiente— Por padrão, o App Runner torna o`PORT`disponível em sua instância de contêiner. Você pode configurar o valor da variável com informações de porta e adicionar variáveis e valores de ambiente personalizados. Para obter mais informações sobre como configurar seu serviço, consultethe section called “Configuração”.
- Função da instância do— Se o código do seu aplicativo fizer chamadas para qualquerAWS, usando as APIs de serviço ou um dosAWSSDKs, crie uma função de instância usandoAWS Identity and Access Management(IAM). Em seguida, anexe-o ao serviço App Runner quando você criá-lo. Incluir tudoAWSpermissões de ação de serviço que seu código requer em sua função de instância. Para mais informações, consulte the section called “Função da instância do”.

Diretrizes de desenvolvimento do

Considere essas diretrizes ao desenvolver código para um aplicativo Web App Runner.

- Código stateless— Projete o aplicativo Web que você implanta no seu serviço App Runner para ficar sem estado. Seu código deve assumir que nenhum estado persiste além da duração do processamento de uma única solicitação da Web recebida.
- Excluir arquivos temporários— quando você cria arquivos, eles são armazenados em um sistema de arquivos e ocupam parte da alocação de armazenamento de seu serviço. Para evitar erros de falta de armazenamento, não mantenha arquivos temporários por períodos prolongados. Equilibre o tamanho do armazenamento com a velocidade de manipulação de solicitações ao tomar decisões de cache de arquivos.

Usando o console do App Runner

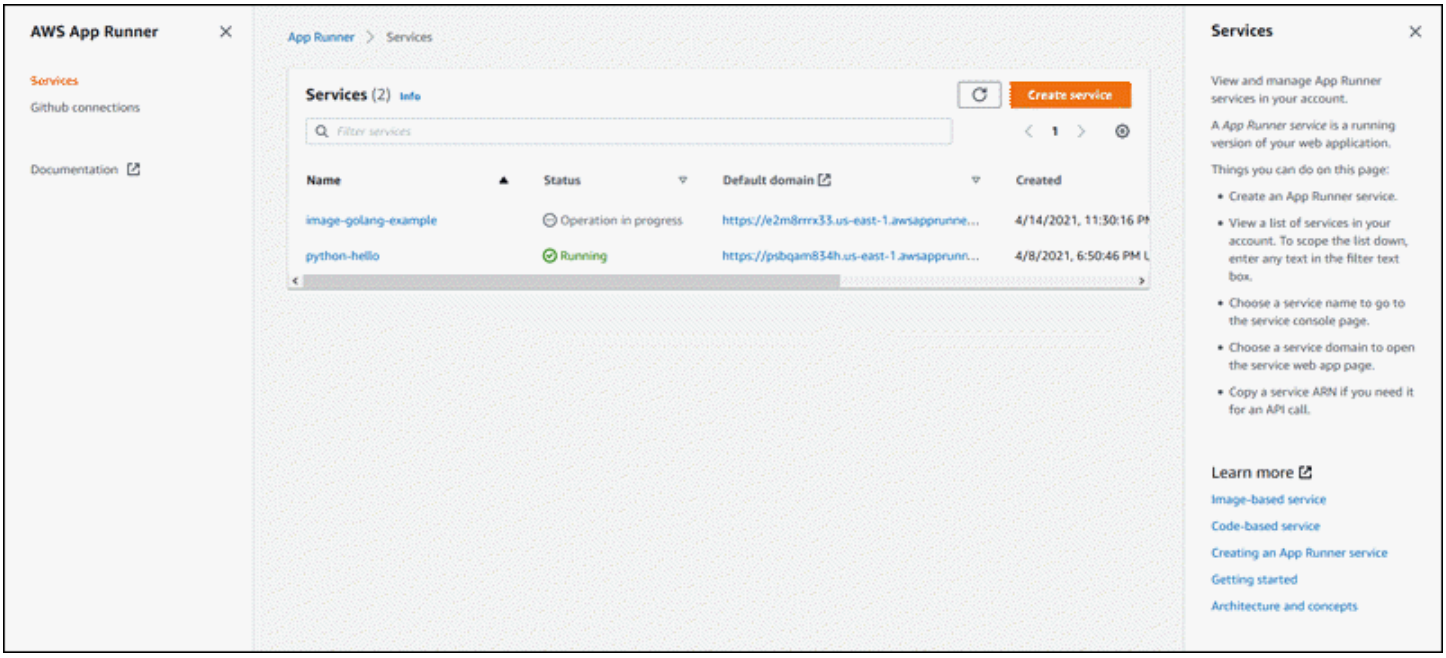
Usar aAWS App Runnerpara criar, gerenciar e monitorar seus serviços do App Runner e recursos relacionados, como conexões. Você pode exibir serviços existentes, criar novos e configurar um serviço. Você pode exibir o status de um serviço App Runner, bem como exibir logs, monitorar atividades e monitorar métricas. Você também pode navegar para o site do seu serviço ou para o repositório de origem.

As seções a seguir descrevem o layout e a funcionalidade do console e apontam para informações relacionadas.

Layout do console

A consola App Runner tem três áreas. Da esquerda para a direita:

- Painel de navegação— Um painel lateral que pode ser recolhido ou expandido. Use-o para escolher a página de console de nível superior que você deseja usar.
- Painel de conteúdo— A parte principal da página do console da. Use-o para exibir informações e executar suas tarefas.
- Painel de ajuda— Um painel lateral para obter mais informações. Expanda-o para obter ajuda sobre a página em que está. Ou escolha qualquerInformaçõesem uma página de console para obter ajuda contextual.



Página Serviços

OServiçoslista os serviços do App Runner em sua conta. Você pode aplicar o escopo da lista para baixo usando a caixa de texto do filtro.

Para acessar oServicespágina

1. Abrir oConsole do Executor de aplicativos, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneServiços.

Coisas que você pode fazer aqui:

- Crie um serviço de Executor de aplicativos. Para mais informações, consulte the section called “Criação”.
- Escolha um nome de serviço para acessar a página do console do painel de controle de serviço.
- Escolha um domínio de serviço para abrir a página do aplicativo Web de serviço.

A página do painel de serviços

Você pode exibir informações sobre um serviço App Runner e gerenciá-lo a partir da página dashbaord do serviço. Na parte superior da página, você pode ver o nome do serviço.

Para acessar o painel de serviço, navegue até oServiços(consulte a secção anterior) e, em seguida, escolha o serviço App Runner.

OVisão geral do serviçofornece detalhes básicos sobre o serviço App Runner e seu aplicativo. Coisas que você pode fazer aqui:

- Exibir detalhes do serviço, como status, integridade e ARN.
- Navegue até o .Domínio padrão— o domínio que o App Runner fornece para o aplicativo Web em execução no seu serviço. Este é um subdomínio no`awsapprunner.com`domínio de propriedade do App Runner.
- Navegue até o repositório de origem implantado no serviço.
- Inicie uma implantação de repositório de origem no seu serviço.
- Pausar, retomar e excluir seu serviço.

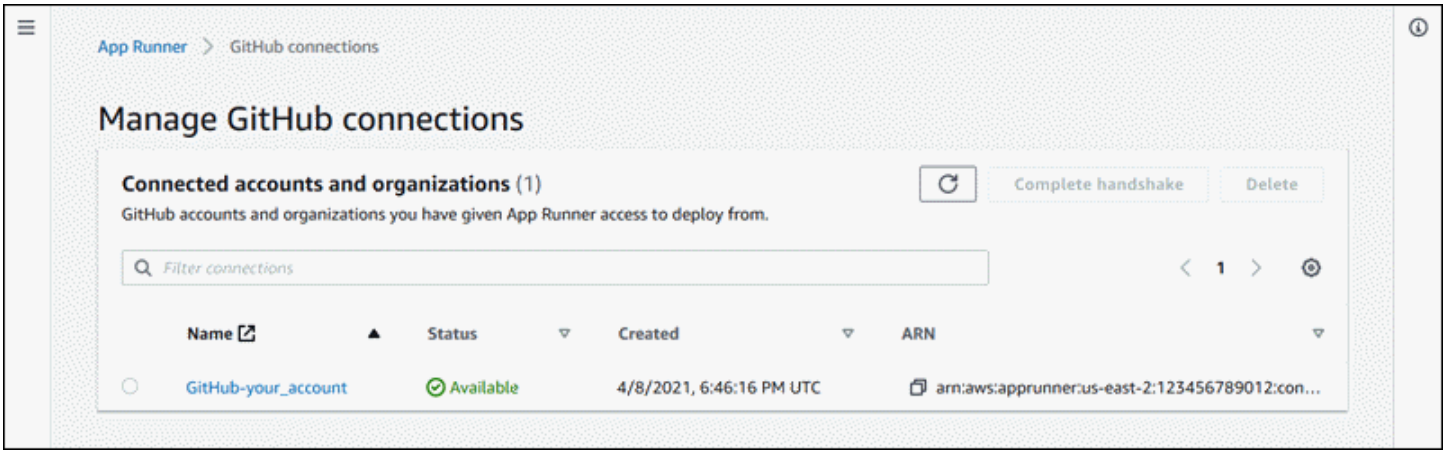
As guias abaixo da visão geral do serviço são para o serviçogerenciamentoeMonitoramento do.

A página de conexões do GitHub

OConexões do GitHublista as conexões do App Runner com o GitHub em sua conta. Você pode aplicar o escopo da lista para baixo usando a caixa de texto do filtro. Para obter mais informações sobre conexões, consulte the section called “Conexões”.

Para acessar oConexões do GitHubpágina

1. Abrir oConsole do Executor de aplicativos, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneConexões do GitHub.



Coisas que você pode fazer aqui:

- Visualize uma lista de conexões do GitHub na sua conta. Para reduzir o escopo da lista, insira qualquer texto na caixa de texto do filtro.
- Escolha um nome de conexão para ir para a conta ou organização do GitHub relacionada.
- Selecione uma conexão para concluir o handshake de uma conexão que você acabou de estabelecer (como parte da criação de um serviço) ou para excluir a conexão.

Gerenciando o ciclo de vida do serviço do App R

Este capítulo descreve como gerenciar o ciclo de vida doAWS App RunnerserviçoServiço do Neste capítulo, você aprenderá como criar, configurar e excluir um serviço, como implantar novas versões de aplicativos em seu serviço e como gerenciar conexões. Você também aprende a controlar a disponibilidade de seu serviço Web pausando e retomando seu serviço.

Tópicos

Criando um serviço App Runner

AWS App Runnerautomatiza o processo de passar de uma imagem de contêiner ou de um repositório de código-fonte para um serviço Web em execução que é dimensionado automaticamente. Você aponta o App Runner para sua imagem de origem ou código, especificando apenas um pequeno número de configurações necessárias. O App Runner cria seu aplicativo, se necessário, provisiona recursos de computação e implanta seu aplicativo para ser executado neles.

Ao criar um serviço, o App Runner cria umaserviçoRecurso. Em alguns casos, talvez seja necessário fornecer umconexãoRecurso. Se você usar o console App Runner, o console criará implicitamente o recurso de conexão. Para obter detalhes sobre os tipos de recursos do App Runner, consultethe section called “Recursos do App Executor”. Esses tipos de recursos têm cotas associadas à sua conta em cadaRegião da AWS. Para mais informações, consulte the section called “Cotas de recurso App Executor”.

Há diferenças sutis no procedimento para criar um serviço dependendo do tipo de origem e do provedor. Este tópico mostra procedimentos totalmente separados para criar esses diferentes tipos

de origem para que você possa seguir o que melhor corresponder à sua situação. Para ver um procedimento básico de início com um exemplo de código, consulteConceitos básicos.

Prerequisites

Antes de criar o serviço do App Runner, conclua as seguintes ações:

- Siga as etapas de configuração emConfiguração.
- Tenha sua fonte de aplicativo pronta. Você pode usar um repositório de código noGitHubou uma imagem de contêiner noAmazon Elastic Container Registry (Amazon ECR)para criar um serviço App Runner.

Criar um serviço

Esta seção explica o processo de criação dos dois tipos de serviço do App Runner: com base no código-fonte e com base em uma imagem de contêiner.

Criar um serviço a partir de um repositório de código do GitHub

As seções a seguir mostram como criar um serviço do App Runner quando o código-fonte é um repositório de código noGitHub. Quando você usa o GitHub, o App Runner precisa se conectar à organização ou conta do GitHub. Portanto, você precisa ajudar a estabelecer essa conexão. Para obter mais informações sobre conexões do App Runner, consultethe section called “Conexões”.

Ao criar o serviço, o App Runner cria uma imagem do Docker que tenha o código do aplicativo e as dependências. Em seguida, ele inicia um serviço que executa uma instância de contêiner desta imagem.

Tópicos

- Criar um serviço a partir do código usando o console do App Runner
- Criar um serviço a partir de código usando a API App Runner ouAWS CLI

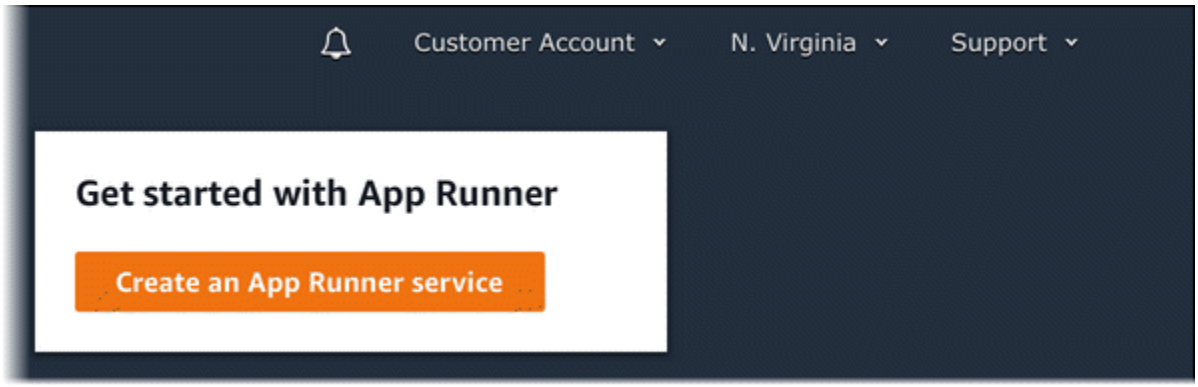
Criar um serviço a partir do código usando o console do App Runner

Para criar um serviço do App Runner usando o console

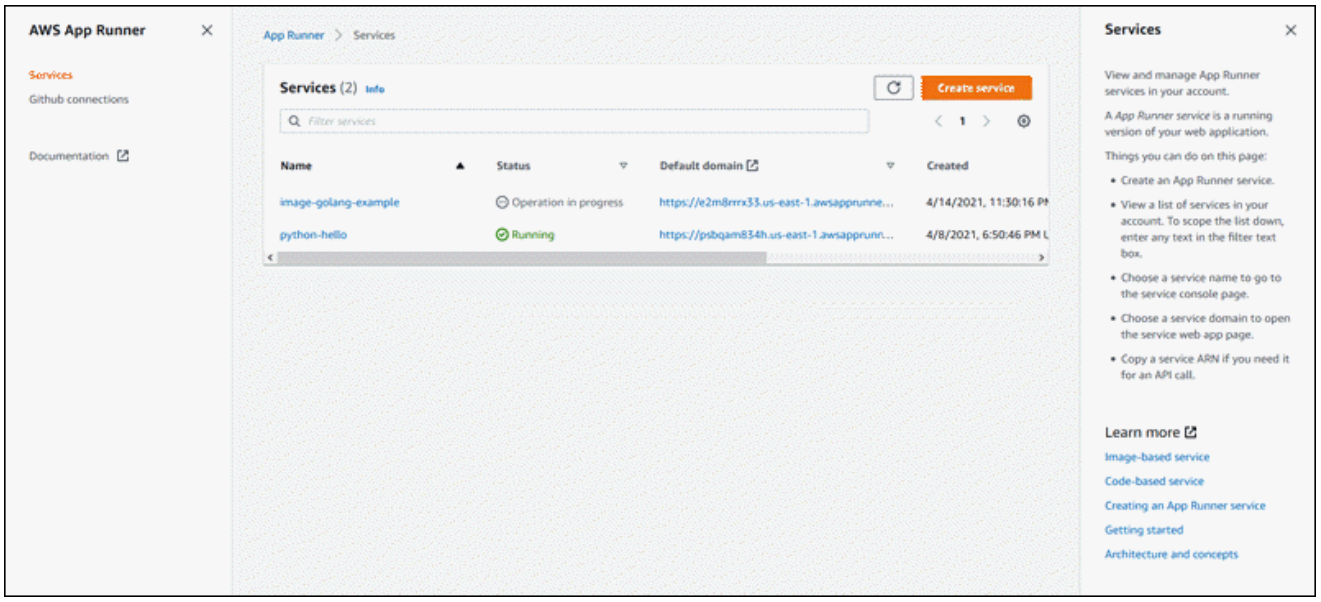
1. Configure o código-fonte.

    a. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.

b. Se oConta da AWSainda não tiver nenhum serviço do App Runner, a página inicial do console será exibida. SelecioneCriar um serviço App Runner.



Se oConta da AWStem serviços existentes, oServiçosÉ exibida uma lista dos serviços do. Escolha Create service.



c. NoOrigem e implantação, noOrigemseção, paraRepository type (Tipo de repositório), escolharepositório de código-fonte.

d. para oConnect ao GitHub, selecione uma conta ou organização do GitHub que você já usou antes ou escolhaAdicionar novo. Em seguida, passe pelo processo de fornecer suas credenciais do GitHub e escolher uma conta ou organização do GitHub para se conectar.

e. para oRepositórioSelecione o repositório que contém o código do aplicativo.

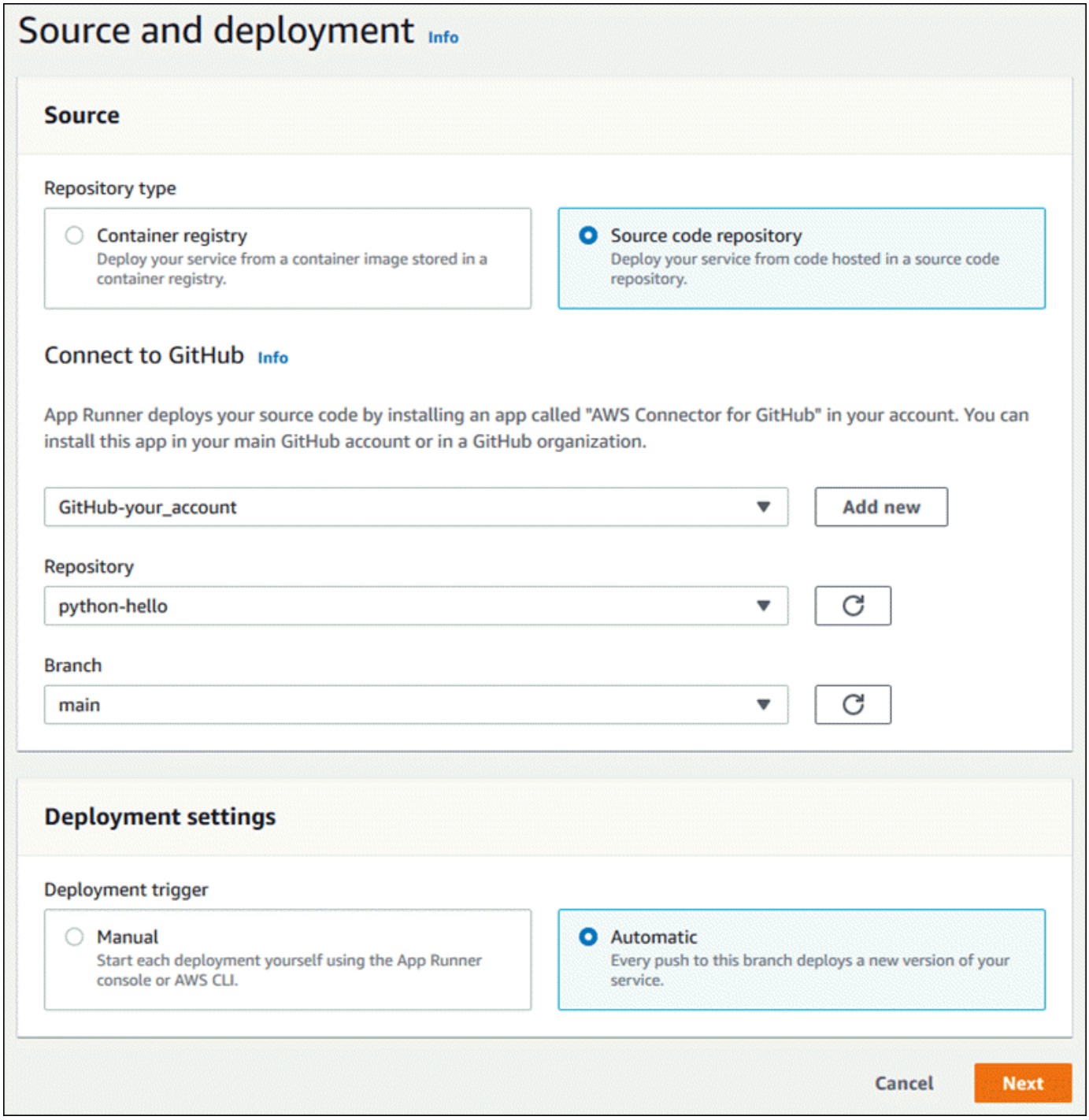
f. para oRamificaçãoSelecione a ramificação que você deseja implantar.

2. Configure suas implantações.

a. NoConfigurações de implantação, selecioneManualouAutomatic.

Para obter mais informações sobre métodos de implantação, consultethe section called "Métodos de implantação".

b. Escolha Next (Próximo).



3. Configure a compilação do aplicativo.

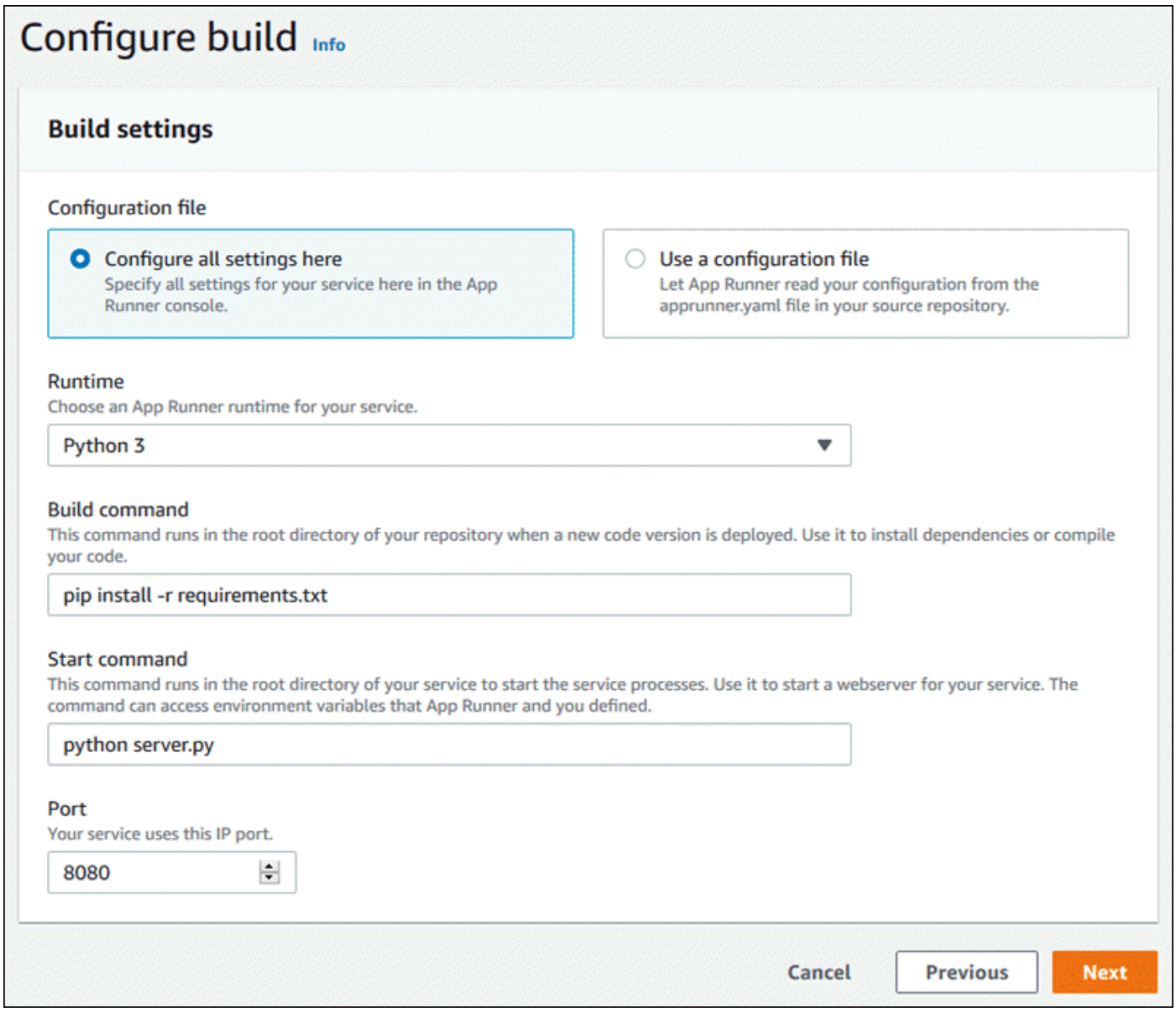
a. NoConfigurar compilação, paraArquivo de configuração, escolhaDefina todas as configurações aquise o repositório não contiver um arquivo de configuração do App Runner, ouUsar um arquivo de configuraçãoSe isso acontecer.

> **Note**
>
> Um arquivo de configuração do App Runner é uma maneira de manter sua configuração de compilação como parte de sua fonte de aplicativo. Quando você fornece um, o App Runner lê alguns valores do arquivo e não permite que você os defina no console.

b. Forneça as seguintes configurações de compilação:

- Tempo de execução— Escolha um tempo de execução gerenciado específico para seu aplicativo.
- Comando Compile— Insira um comando que crie o aplicativo a partir de seu código-fonte. Esta pode ser uma ferramenta específica de idioma ou um script fornecido com seu código.
- Comando para iniciar— Digite o comando que inicia seu serviço Web.
- Port— Insira a porta IP que seu serviço Web escuta.

c. Escolha Next (Próximo).



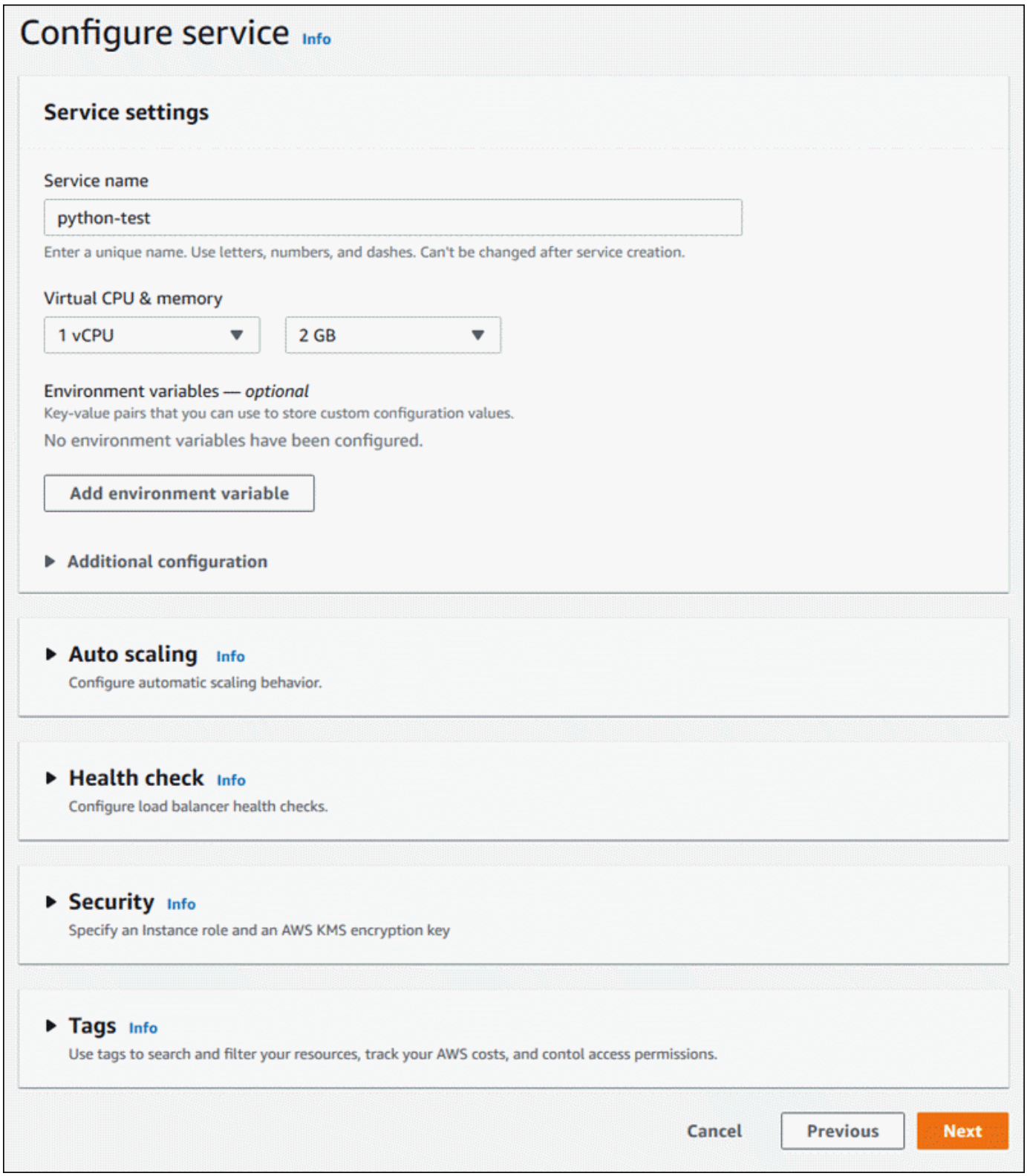
4. Configure seu serviço.

   a. NoConfigurar o serviço, noConfigurações do serviço, digite um nome de serviço.

   > **Note**
   >
   > Todas as outras configurações de serviço são opcionais ou têm padrões fornecidos pelo console.

   b. Opcionalmente, altere ou adicione outras configurações para atender aos requisitos do aplicativo.

c. Escolha Next (Próximo).



5. NoRevisar e criar, verifique todos os detalhes inseridos e escolhaCriar e implantar.

   Resultado: Se a criação do serviço for bem-sucedida, o console deverá mostrar o painel de serviço, com umVisão geral do serviçodo novo serviço.

6. Verifique se o serviço está em execução.

   a. Na página do painel de serviço, aguarde até que o serviçoStatuséRunning.

   b. Selecione oDomínio padrãovalor—é a URL do site do seu serviço.

   c. Use seu site e verifique se ele está funcionando corretamente.

Criar um serviço a partir de código usando a API App Runner ouAWS CLI

Para criar um serviço usando a API do App Runner ouAWS CLI, chame o`CreateService`Ação de API. Para obter mais informações e um exemplo, consulteCreateService. Se esta for a primeira vez que você está criando um serviço usando uma organização ou conta específica do GitHub, comece chamandoCreateConnection. Isso estabelece uma conexão entre o App Runner e a organização ou conta do GitHub. Para obter mais informações sobre conexões do App Runner, consultethe section called “Conexões”.

A criação do serviço é iniciada se a chamada retornar uma resposta bem-sucedida com umServiçoobjeto mostrando`"Status": "CREATING"`.

Para ver uma chamada de exemplo, consulteCriar um serviço de repositório de códigos-fontenoAWS App RunnerReferência de API do

Criar um serviço a partir de uma imagem do Amazon ECR

As seções a seguir mostram como criar um serviço do App Runner quando a origem é uma imagem de contêiner armazenada noAmazon ECR. O Amazon ECR é umAWSserviçoServiço. Portanto, para criar um serviço baseado em uma imagem do Amazon ECR, você fornece ao App Runner uma função de acesso contendo as permissões de ação necessárias do Amazon ECR.

> **Note**
>
> Uma função de acesso não é necessária se sua imagem for armazenada no Amazon ECR Public, onde as imagens estão disponíveis publicamente.

Durante a criação do serviço, o App Runner inicia um serviço que executa uma instância de contêiner da imagem que você fornece. Não há fase de construção neste caso.

Tópicos

- Criar um serviço a partir de uma imagem usando o console App Runner
- Criar um serviço a partir de uma imagem usando a API App Runner ouAWS CLI

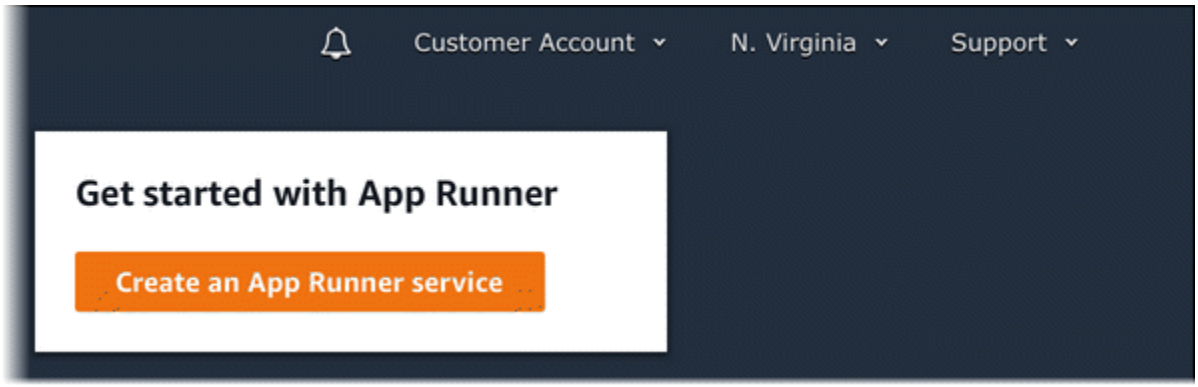
Criar um serviço a partir de uma imagem usando o console App Runner

Para criar um serviço do App Runner usando o console

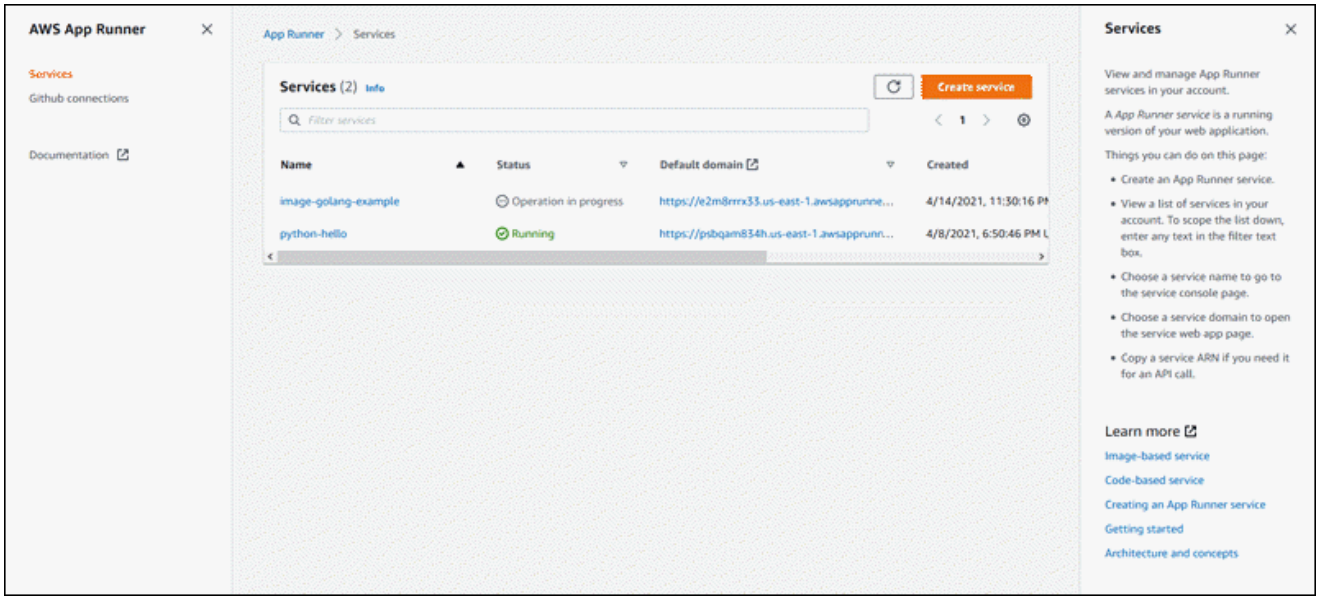
1. Configure o código-fonte.

   a. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.

   b. Se oConta da AWSainda não tiver nenhum serviço do App Runner, a página inicial do console será exibida. SelecioneCriar um serviço App Runner.



   Se oConta da AWStem serviços existentes, oServiçosÉ exibida uma lista dos serviços do. Escolha Create service.



c. NoOrigem e implantação, noOrigemseção, paraRepository type (Tipo de repositório), escolhaRegistro de contêiner.

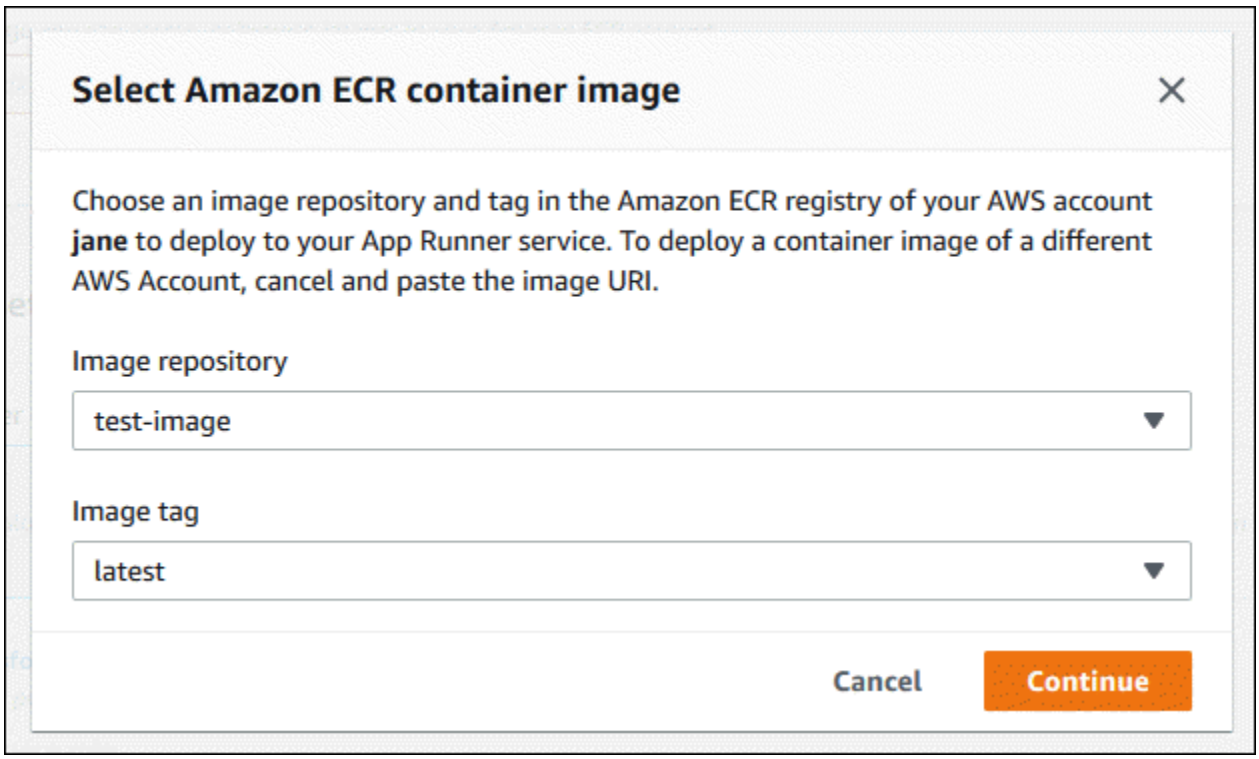
d. para oFornecedor, escolha o provedor onde sua imagem está armazenada:

- Amazon ECR— Uma imagem privada armazenada no Amazon ECR em seuConta da AWS.
- Público do Amazon ECR— Uma imagem publicamente legível armazenada no Amazon ECR Public.

e. para oURI de imagem de, escolhaNavegar.

f. NoSelecionar imagem de contêiner do Amazon ECR, paraRepositório de, selecione o repositório que contém sua imagem.

g. para oTag de imagemSelecione a tag de imagem específica que você deseja implantar, por exemplo,lateste, depois, escolhaContinuar.



2. Configure suas implantações.

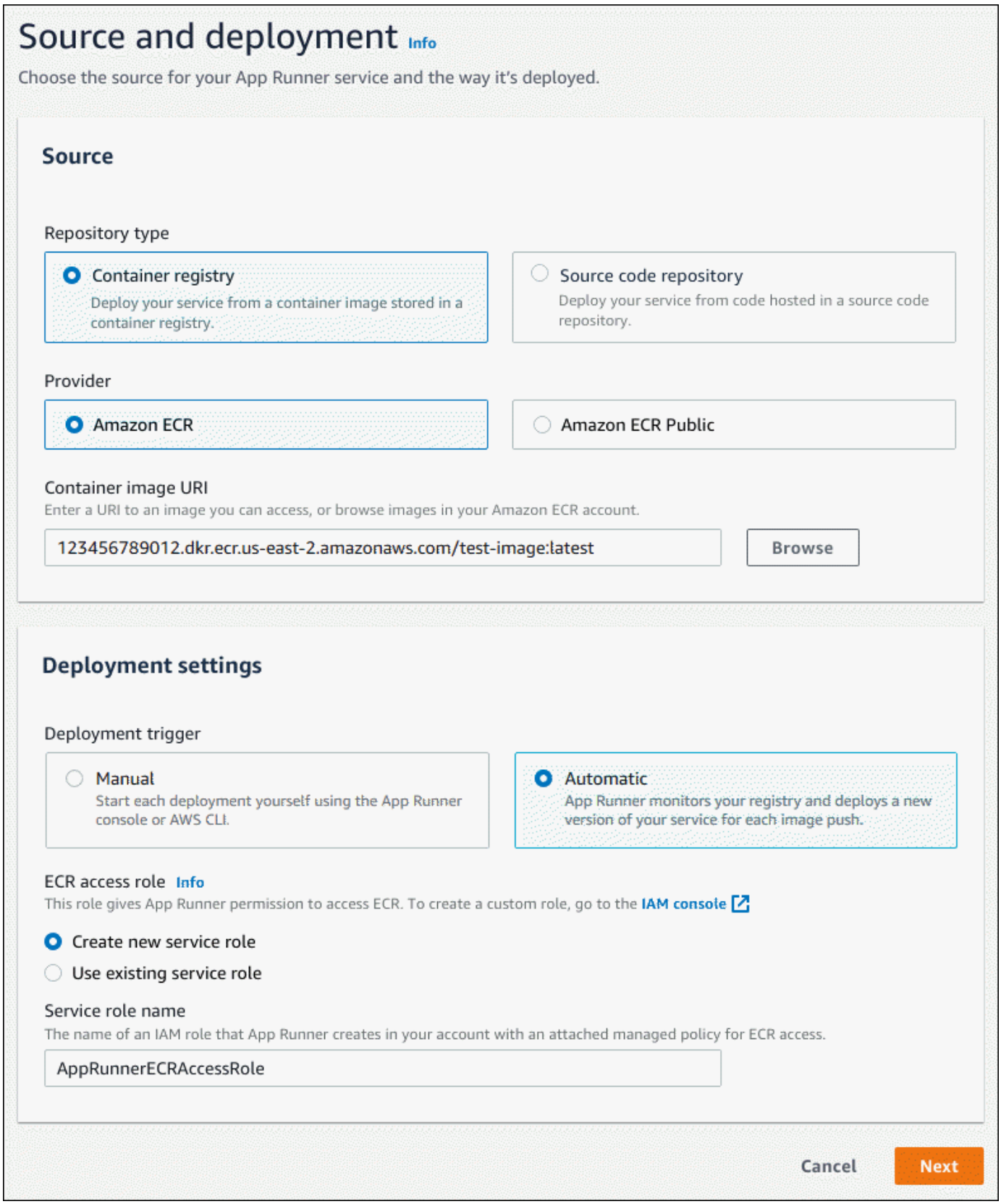
   a. NoConfigurações de implantação, selecioneManualouAutomatic.

      Para obter mais informações sobre métodos de implantação, consultethe section called "Métodos de implantação".

      > **Note**
      >
      > O App Runner não oferece suporte à implantação automática para imagens públicas do Amazon ECR.

   b. [Amazon ECRprovedor] ParaFunção de acesso do ECREscolha uma função de serviço existente na conta ou crie uma nova função. Se você estiver usando a implantação manual, também poderá optar por usar a função de usuário do IAM no momento da implantação.

   c. Escolha Next (Próximo).


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

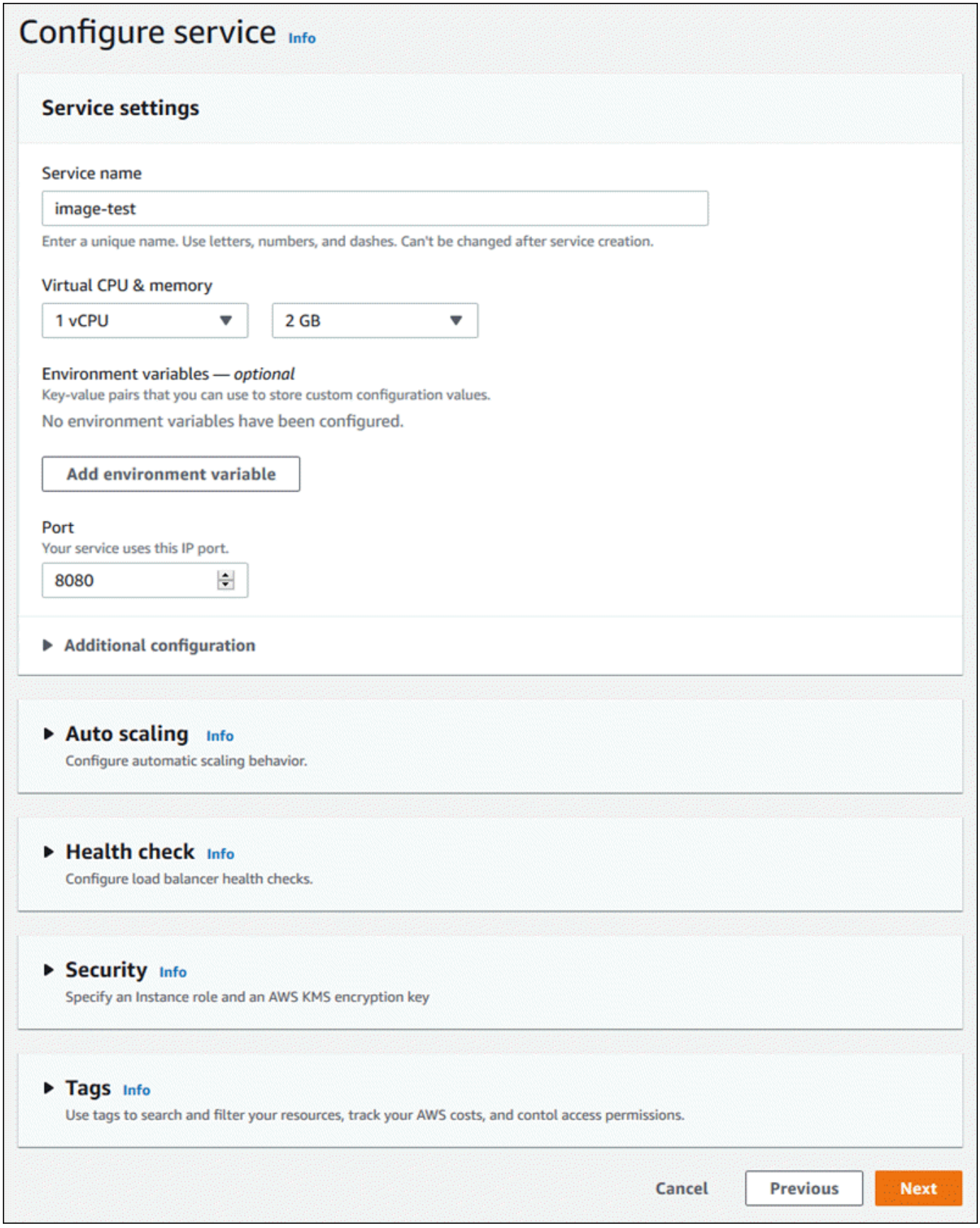
3. Configure seu serviço.

   a. NoConfigurar o serviço, noConfigurações do serviço, insira um nome de serviço e a porta IP que seu site de serviço escuta.

   > **Note**
   >
   > Todas as outras configurações de serviço são opcionais ou têm padrões fornecidos pelo console.

   b. (Opcional) Altere ou adicione outras configurações para atender às necessidades do seu aplicativo.

   c. Escolha Next (Próximo).


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. NoRevisar e criarEm seguida, verifique todos os detalhes que você inseriu e selecioneCriar e implantar.

   Resultado: Se a criação do serviço for bem-sucedida, o console deverá mostrar o painel de serviço, com umVisão geral do serviçodo novo serviço.

5. Verifique se o serviço está em execução.

   a. Na página do painel de serviço, aguarde até que o serviçoStatusséRunning.

   b. Selecione oDomínio padrãovalor—é a URL do site do seu serviço.

   c. Use seu site e verifique se ele está funcionando corretamente.

Criar um serviço a partir de uma imagem usando a API App Runner ouAWS CLI

Para criar um serviço usando a API do App Runner ouAWS CLI, chame oCreateServiceAção de API.

A criação do serviço é iniciada se a chamada retornar uma resposta bem-sucedida com umServiçoobjeto mostrando`"Status": "CREATING"`.

Para ver uma chamada de exemplo, consulteCriar um serviço de repositório de imagens de origemnoAWS App RunnerReferência de API do

Quando falha na criação do serviço

Se a sua tentativa de criar um serviço App Runner falhar, o serviço mostrará um status de`CREATE_FAILED`(Falha na criaçãono console).

Sua tentativa de criar um serviço pode falhar devido a problemas em seu código de aplicativo, processo de compilação ou configuração, porque você atingiu cotas de recursos ou devido a problemas temporários com oAWSque seu serviço precisa usar. Para solucionar uma falha, recomendamos executar as seguintes ações. Primeiro, leia os eventos de serviço e os logs para descobrir o que causou a falha. Em seguida, faça as alterações necessárias em seu código ou configuração. Por último, exclua um ou mais serviços se você atingiu sua cota de serviço. Em seguida, depois de concluir todas estas etapas, tente criar o serviço novamente.

> **Important**
>
> O serviço com falha não é utilizável. Você não terá cobranças adicionais para ele além da tentativa de criação inicial. No entanto, o App Runner não exclui automaticamente o serviço

com falha e ainda conta para sua cota de serviço. Quando terminar de analisar a falha, certifique-se de excluir o serviço com falha.

Implantar uma nova versão do aplicativo no App Runner

Quando vocêCriar um serviçoemAWS App Runner, você configura uma origem do aplicativo — uma imagem de contêiner ou um repositório de origem. O App Runner provisiona recursos para executar seu serviço e implanta seu aplicativo para eles.

Este tópico descreve maneiras de reimplantar a origem do aplicativo no serviço do App Runner quando uma nova versão estiver disponível. Esta pode ser uma nova versão de imagem no repositório de imagens ou uma nova confirmação no repositório de código. O App Runner fornece dois métodos para implantar em um serviço:AutomaticeManual.

Métodos de implantação

O App Runner fornece os seguintes métodos para você controlar como as implantações de aplicativos são iniciadas.

Implantação automática

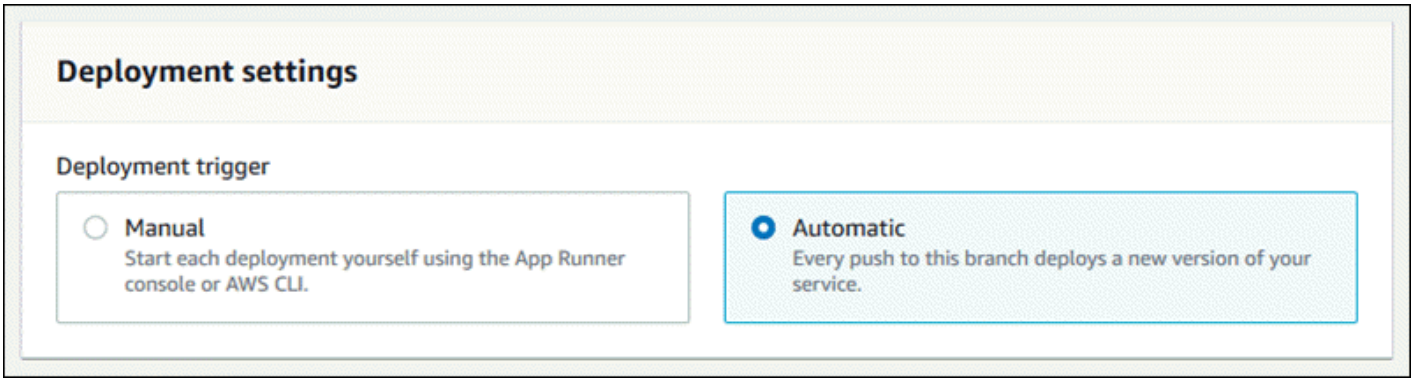
Utilize a implementação automática quando pretender um comportamento de integração e implementação contínuas (CICD) para o seu serviço. App Runner monitora sua imagem ou repositório de código. Sempre que você envia uma nova versão de imagem para o repositório de imagens ou uma nova confirmação para o repositório de código, o App Runner a implanta automaticamente em seu serviço sem mais ações do seu lado.

Implantar manual

Use a implantação manual quando quiser iniciar explicitamente cada implantação em seu serviço. Você inicia uma implantação se o repositório que você configurou para seu serviço tiver uma nova versão que você deseja implantar. Para mais informações, consulte the section called “Implantar manual”.

Você pode configurar o método de implantação para o serviço das seguintes maneiras:

- Console— Para um novo serviço que você está criando ou para um serviço existente, noConfigurações de implantaçãoSeção do doOrigem e implantação, selecioneManualouAutomatic.



- API ouAWS CLI— Em uma chamada para oCreateServiceouUpdateService, defina a propriedade`AutoDeploymentsEnabled`membro doSourceConfigurationparâmetro para`False`para implantação manual ou`True`para implantação automática.

Implantar manual

Com a implantação manual, você precisa iniciar explicitamente cada implantação em seu serviço. Quando você tiver uma nova versão da imagem ou código do aplicativo pronta para implantar, consulte as seções a seguir para saber como executar uma implantação usando o console e a API.

Implantar uma versão do aplicativo usando o console do App Runner

Para implantar usando o console do App Runner

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. Escolha Deploy (Implantar).

   Resultado: A implantação da nova versão é iniciada. Na página do painel de serviço, o serviçoStatusalterações noOperação em andamento.
4. Aguarde até que a implantação seja concluída. Na página do painel de serviço, o serviçoStatusdeve mudar de volta paraRunning.
5. Para verificar se a implantação foi bem-sucedida, na página do painel de serviço, escolha a opçãoDomínio padrãovalor—é a URL do site do seu serviço. Inspecione ou interaja com seu aplicativo Web e verifique sua alteração de versão.

Implante uma versão do aplicativo usando a API do App Runner ouAWS CLI

Para implantar usando a API do App Runner ouAWS CLI, chame oStartDeploymentAção de API. O único parâmetro a ser transmitido é o ARN do serviço. Você já configurou o local de origem do aplicativo quando criou o serviço, e o App Runner pode encontrar a nova versão. Sua implantação será iniciada se a chamada retornar uma resposta bem-sucedida.

Configuração de um serviço App Runner

Quando vocêCrie umAWS App Runnerserviço, defina vários valores de configuração. É possível alterar algumas dessas configurações após criar o serviço. Outras configurações podem ser aplicadas somente durante a criação do serviço e não podem ser alteradas posteriormente. Este tópico discute a configuração do seu serviço usando a API do App Runner, o console do App Runner e um arquivo de configuração do App Runner.

Configure seu serviço usando a API App Runner ouAWS CLI

A API define quais configurações podem ser alteradas após a criação do serviço. A lista a seguir discute as ações, tipos e limitações relevantes.

- UpdateServiceaction — Pode ser chamado após a criação para atualizar algumas definições de configuração.
  - Pode ser atualizado- Você pode atualizar as configurações na guia`SourceConfiguration`,`InstanceConfiguration`, e`HealthCheckConfiguration`Parâmetros. No entanto, em`SourceConfiguration`, você não pode alternar seu tipo de fonte de código para imagem ou o contrário. Você deve fornecer o mesmo parâmetro de repositório que forneceu ao quando criou o serviço. É qualquer um dos dois`CodeRepository`ou`ImageRepository`.

    Você também pode atualizar`AutoScalingConfigurationArn`, o ARN do recurso de configuração de auto scaling associado ao serviço.
  - Não pode ser atualizado— Você não pode alterar o`ServiceName`e`EncryptionConfiguration`que estão disponíveis noCreateServiceAção . Eles não podem ser alterados após serem criados. OUpdateServicenão inclui esses parâmetros.
  - API vs. arquivo— É possível definir o`ConfigurationSource`O parâmetro doCodeConfiguration(usado para repositórios de código-fonte como parte do`SourceConfiguration`) para`Repository`. Neste caso, o App Runner ignora as definições de configuração no`CodeConfigurationValues`e lê essas configurações a partir de

umArquivo de configuraçãoNo repositório. Se você definir`ConfigurationSource`para`API`, o App Runner obtém todas as definições de configuração da chamada de API e ignora o arquivo de configuração, mesmo que exista.

- TagResourceaction — Pode ser chamado depois que seu serviço é criado para adicionar tags ao serviço ou atualizar valores de tags existentes.
- UntagResourceaction — Pode ser chamado depois que seu serviço é criado para remover tags do serviço.

Configure o seu serviço utilizando a consola App Runner

O console usa a API do App Runner para aplicar atualizações de configuração. As regras de atualização impostas pela API, conforme definido na seção anterior, determinam o que você pode configurar usando o console. Algumas configurações que estavam disponíveis durante a criação do serviço não estão disponíveis para modificação posteriormente. Além disso, se você decidir usar umArquivo de configuração, as configurações adicionais ficam ocultas no console e o App Runner as lê do arquivo.

Para configurar seu serviço

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. Na página do painel de serviços, escolha oConfiguraçãoGuia.

   Resultado: O console exibe as definições de configuração atuais do seu serviço em várias seções: Origem e implantação,Configurar compilação, eConfigurar o serviço.
4. Para atualizar as configurações em qualquer categoria, escolhaEdite.
5. Na página de edição da configuração, faça todas alterações desejadas e escolhaSalve as alterações.

Configurar seu serviço usando um arquivo de configuração do App Runner

Ao criar ou atualizar um serviço App Runner, você pode instruir o App Runner a ler algumas definições de configuração de um arquivo de configuração fornecido como parte do repositório de origem. Ao fazer isso, você pode gerenciar as configurações que estão relacionadas ao seu código-

fonte sob controle de código-fonte, juntamente com o próprio código. O arquivo de configuração também fornece determinadas configurações avançadas que você não pode definir usando o console ou a API. Para mais informações, consulte Arquivo de configuração do App Runner.

Gerenciando conexões do App Runner

Quando vocêCriar um serviçoemAWS App Runner, você configura uma origem do aplicativo — uma imagem de contêiner ou um repositório de origem armazenado com um provedor. Se um repositório armazenado com um provedor terceirizado for privado (não legível publicamente), o App Runner precisa estabelecer uma conexão autenticada e autorizada com o provedor. Em seguida, o App Runner pode ler seu repositório e implantá-lo em seu serviço. O App Runner não requer o estabelecimento de conexão quando você cria um serviço que acessa o código armazenado noConta da AWSou em um local de código público.

O App Runner mantém informações de conexão em um recurso chamadoconexão. O App Runner requer um recurso de conexão quando você cria um serviço que precisa de informações de conexão de terceiros. Veja a seguir algumas informações importantes sobre conexões:

- Fornecedores— O App Runner atualmente requer recursos de conexão com oGitHub.
- Compartilhado— Você pode usar um recurso de conexão para criar vários serviços do App Runner que usam a mesma conta de provedor de repositório.
- Gerenciamento de recursos— No App Runner, você pode criar e excluir conexões. No entanto, não é possível modificar uma conexão existente.
- Cota de recurso— Os recursos de conexão têm uma cota definida associada à suaConta da AWSEm cadaRegião da AWS. Se você atingir essa cota, talvez seja necessário excluir uma conexão antes de se conectar a uma nova conta de provedor. Você pode excluir uma conexão usando o App Runnerconsoleou API. Para mais informações, consulte the section called “Cotas de recurso App Executor”.

Gerenciar conexões usando o console App Runner

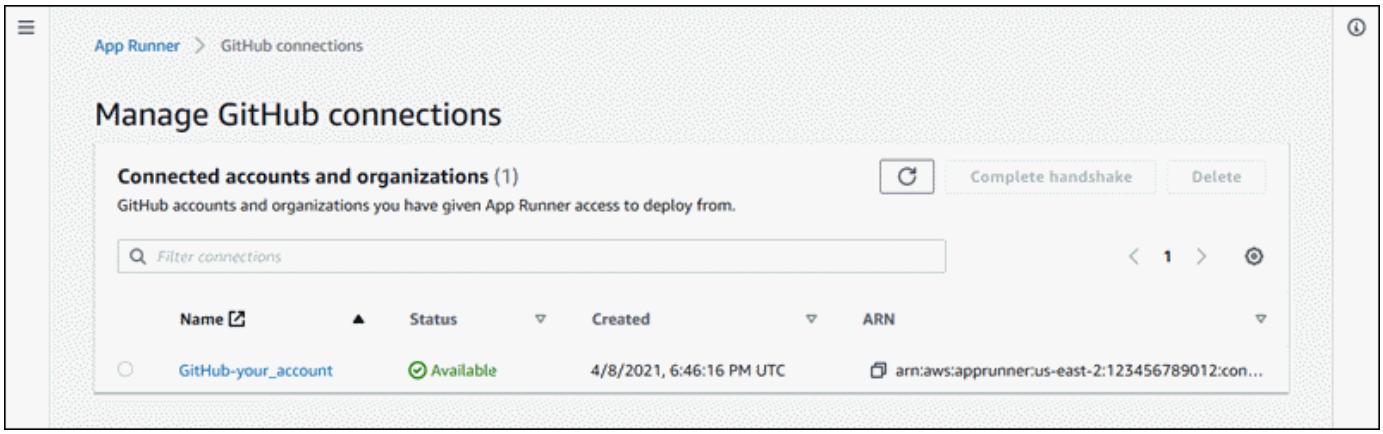
Quando utiliza a consola App Runner paraCriar um serviço, você fornece detalhes da conexão. Não é necessário criar explicitamente um recurso de conexão. No console, você pode optar por se conectar a uma conta do GitHub à qual você já se conectou antes ou se conectar a uma nova conta. Quando necessário, o App Runner cria um recurso de conexão para você. Para uma nova conexão, alguns provedores (por exemplo, GitHub) exigem que você complete um handshake de autenticação antes de poder usar a conexão. O console o orienta por esse processo.

O console também tem uma página para gerenciar suas conexões existentes. Você pode concluir o handshake de autenticação para uma conexão se não tiver feito isso quando criou seu serviço. Você também pode excluir conexões que não está mais sendo usadas. O procedimento a seguir mostra como você pode gerenciar conexões do GitHub.

Para gerenciar conexões do GitHub em sua conta

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneConexões do GitHub.

   O console exibirá uma lista de conexões do GitHub em sua conta.



3. Agora você pode realizar uma das ações a seguir com qualquer conexão na lista:

   - Abrir conta ou organização do GitHub— Selecione o nome da conexão.
   - Handshake de autenticação completo— Selecione a conexão e escolhaHandshake completo. O console o orienta pelo processo de handshake de autenticação.
   - Excluir conexão— Selecione a conexão e escolhaExcluir. Siga as instruções no prompt de exclusão.

Gerencie conexões usando a API do App Runner ouAWS CLI

Você pode usar as seguintes ações da API do App Runner para gerenciar suas conexões.

- CreateConnection— Cria uma conexão com uma conta de provedor de repositório. Após a criação da conexão, você deve concluir manualmente o handshake de autenticação usando o console do App Runner. Esse processo é explicado na seção anterior.
- ListConnections— Retorna uma lista de conexões do App Runner associadas à suaConta da AWS.

- DeleteConnection— Exclui uma conexão. Talvez seja necessário excluir conexões desnecessárias se você atingir a cota de conexão para oConta da AWS.

Gerenciando o dimensionamento automático do App Runner

AWS App Runnerdimensiona automaticamente os recursos de computação (instâncias) para cima ou para baixo para o aplicativo App Runner. O dimensionamento automático fornece tratamento adequado de solicitações quando o tráfego de entrada é alto e reduz o custo quando o tráfego fica lento. Você pode configurar alguns parâmetros para ajustar o comportamento de auto scaling para o seu serviço.

O App Runner mantém as configurações de dimensionamento automático em um recurso chamadoAutoScalingConfiguration. É possível fornecer um recurso de configuração de Auto Scaling ao criar ou atualizar um serviço. O console App Runner cria um para você quando você cria um novo serviço App Runner. Fornecer uma configuração de auto scaling é opcional. Se você não fornecer uma, o App Runner fornecerá uma configuração de auto scaling padrão com valores recomendados.

Uma configuração de auto scaling tem umnome doE um numéricorevisão. Várias revisões de uma configuração têm o mesmo nome e números de revisão diferentes. Você pode usar nomes de configuração diferentes para diferentes cenários de dimensionamento automático, como alta disponibilidade ou baixo custo. Para cada nome, você pode adicionar várias revisões para ajustar as configurações de um cenário específico.

Veja a seguir algumas informações importantes sobre configurações de auto scaling:

- ConfiguraçõesVeja o que você pode configurar:
  - Simultaneidade máxima— O número máximo de solicitações simultâneas processadas por uma instância. Quando o número de solicitações simultâneas excede essa cota, o App Runner aumenta o serviço.
  - Max size— O número máximo de instâncias para as quais o seu serviço pode ser dimensionado. No máximo, esse número de instâncias está servindo ativamente o tráfego para seu serviço.
  - Tamanho mínO número mínimo de instâncias que o App Runner provisiona para o seu serviço. O serviço sempre tem pelo menos esse número de instâncias provisionadas. Alguns deles servem ativamente o tráfego. O restante deles (instâncias provisionadas e inativas) é uma reserva econômica de capacidade computacional, pronta para ser ativada rapidamente. Você paga pelo uso de memória de todas as instâncias provisionadas. Você paga pelo uso da CPU apenas do subconjunto ativo.

O App Runner duplica temporariamente o número de instâncias provisionadas durante as implantações, para manter a mesma capacidade para os códigos antigos e novos.

- Revisões— A primeira configuração criada com um nome obtém o número de revisão 1. As configurações subsequentes com o mesmo nome obtêm números de revisão consecutivos (começando com 2). Você pode associar seu serviço App Runner a uma revisão de configuração de auto scaling específica ou à revisão mais recente da configuração.
- Compartilhado— Você pode compartilhar um único recurso de configuração de Auto Scaling em vários serviços do App Runner. Isso é útil se eles tiverem requisitos de dimensionamento semelhantes. Em particular, você pode configurar vários serviços para que todos usem a versão mais recente de uma configuração especificando o nome da configuração, mas não especificando uma revisão. Ao fazer isso, qualquer um dos serviços que você configurou dessa forma recebe atualizações de configuração de dimensionamento automático quando você atualiza o serviço. Para obter mais informações sobre alterações de configuração do, consultethe section called “Configuração”.
- Gerenciamento de recursos— Você pode usar o App Runner para criar e excluir configurações de auto scaling. Não é possível atualizar diretamente uma configuração. Em vez disso, você pode criar uma nova revisão para um nome de configuração existente para atualizar efetivamente a configuração.

> **Note**
>
> No momento, você só pode criar uma configuração com uma única revisão no console do App Runner. Para criar mais revisões e excluir configurações, use o aplicativo RunnerAPI.

- Cota de recurso— Há cotas definidas para o número de nomes de configuração exclusivos e revisões que você pode ter para seus recursos de configuração de auto scaling em cadaRegião da AWS. Se você atingir essas cotas, você deve excluir um nome de configuração ou pelo menos algumas de suas revisões antes de criar mais. Usar o App RunnerAPIpara excluí-los. Para mais informações, consulte the section called “Cotas de recurso App Executor”.

Gerenciar o dimensionamento automático usando o console do App Runner

Quando vocêCriar um serviçono console do App Runner, você pode usar a configuração padrão de auto scaling ou uma configuração personalizada. Para usar uma configuração personalizada, escolha uma configuração existente ou forneça um novo nome e configurações. Se for uma nova

configuração, o App Runner cria um novo recurso de configuração de auto scaling para você e, em seguida, associa-o ao seu novo serviço.

Gerencie o dimensionamento automático usando a API do App Runner ouAWS CLI

Você pode usar as seguintes ações da API do App Runner para gerenciar suas configurações de auto scaling.

- CreateAutoScalingConfiguration— Cria uma nova configuração de auto scaling ou uma revisão para uma existente.
- ListAutoScalingConfigurations— Retorna uma lista das configurações de auto scaling que estão associadas ao seuConta da AWSCom informações de resumo.
- DescribeAutoScalingConfiguration— Retorna uma descrição completa de uma configuração de auto scaling.
- DeleteAutoScalingConfiguration— Exclui uma configuração de auto scaling. Você pode excluir uma revisão específica ou a revisão ativa mais recente. Talvez seja necessário excluir configurações desnecessárias de auto scaling se você atingir a cota de configuração de auto scaling para o seuConta da AWS.

Gerenciar nomes de domínio personalizados para um serviço App Runner

Quando você cria umAWS App Runner, o App Runner aloca um nome de domínio para ele. Este é um subdomínio no`awsapprunner.com`que pertence ao App Runner. Ele pode ser usado para acessar o aplicativo da Web que está sendo executado em seu serviço.

Se você tem um nome de domínio, pode associá-lo ao serviço do App Runner. Após o App Runner validar seu novo domínio, ele pode ser usado para acessar seu aplicativo além do domínio App Runner. Você pode associar até cinco domínios personalizados.

> **Note**
>
> Opcionalmente, é possível incluir as instruções`www`subdomínio do seu domínio. No entanto, isso tem suporte somente na API. O console do App Runner não oferece suporte a ele.

Quando você associa um domínio personalizado ao seu serviço, o App Runner fornece um conjunto de registros de validação de certificado. Adicione-os ao seu Sistema de Nomes de Domínio (DNS) para que o App Runner possa validar que você possui ou controla o domínio. Além disso, adicione os registros CNAME ou ALIAS ao DNS para direcionar o domínio App Runner. Você precisa adicionar um registro para o domínio personalizado e outro para o`www`, se você escolheu essa opção. Em seguida, aguarde até que o status de domínio personalizado se torne`Ativo`no console do App Runner. Isso normalmente leva vários minutos (mas pode levar de 24 a 48 horas). Nesse ponto, seu domínio personalizado é validado e o App Runner inicia o roteamento do tráfego desse domínio para seu aplicativo Web.

Você pode especificar um domínio para associar ao seu serviço App Runner das seguintes maneiras:

- Um domínio raiz— Por exemplo,`example.com`. Opcionalmente, é possível associar`www.example.com`Também como parte da mesma operação.
- Um subdomínio— Por exemplo,`login.example.com`ou`admin.login.example.com`. Você pode, opcionalmente, associar o`www`Subdomain também como parte da mesma operação.
- Um caractere curinga— Por exemplo,`*.example.com`. Não é possível usar o`www`Nesse caso,. Você pode especificar um curinga somente como o subdomínio imediato de um domínio raiz e somente por conta própria (estas não são especificações válidas:`login*.example.com`,`*.login.example.com`). Essa especificação curinga associa todos os subdomínios imediatos e não associa o próprio domínio raiz (o domínio raiz teria que estar associado em uma operação separada).

Uma associação de domínio mais específica substitui uma menos específica. Por exemplo,`login.example.com`Substituições`*.example.com`. O certificado e CNAME da associação mais específica são usados.

O exemplo a seguir mostra como é possível usar várias associações de domínios personalizados:

1. Associate`example.com`com a página inicial do seu serviço. Habilitar o`www`para associar também`www.example.com`.
2. Associate`login.example.com`com a página de login do seu serviço.
3. Associate`*.example.com`com uma página personalizada “não encontrada”.

Você pode desassociar (desvincular) um domínio personalizado do serviço App Runner. Quando você desvincula um domínio, o App Runner interrompe o roteamento do tráfego desse domínio para seu aplicativo Web. Você deve excluir os registros desse domínio do DNS.

O App Runner cria internamente certificados que controlam a validade do domínio. Eles estão armazenados emAWS Certificate Manager(ACM). O App Runner não exclui esses certificados por sete dias após um domínio ser desassociado do seu serviço ou após o serviço ser excluído.

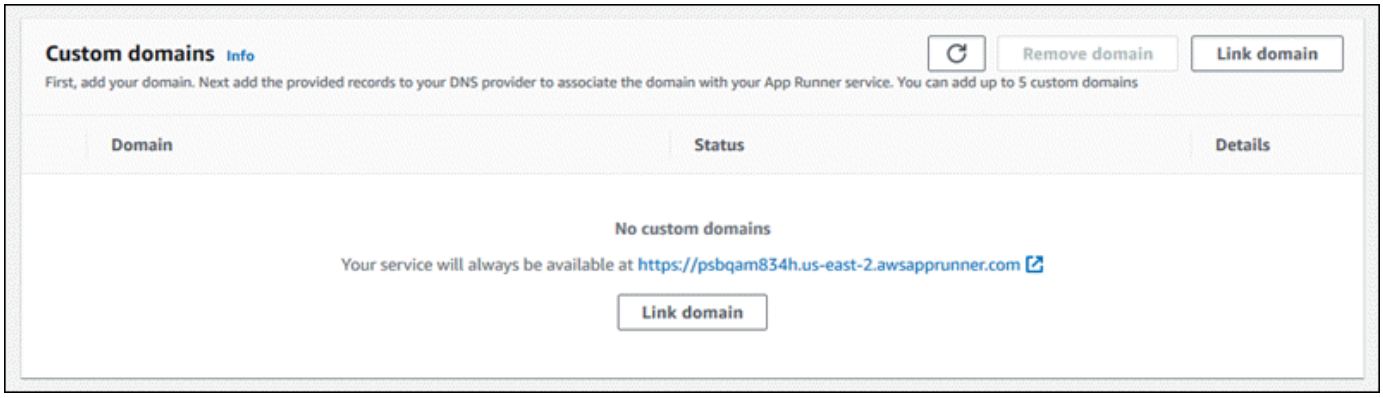
Gerenciar domínios personalizados usando o console App Runner

Para associar (vincular) um domínio personalizado usando o console do App Runner

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. Na página do painel de serviços, selecione as opçõesDomínios personalizadosGuia.

   O console mostra os domínios personalizados associados ao seu serviço ouSem domínios personalizados.



4. NoDomínios personalizados, selecioneLink de domínio.
5. NoLink de domínio personalizado, insira um nome de domínio e selecioneVisualizar a configuração do DNS.
6. Siga as instruções da guiaConfigurar o DNSpara iniciar o processo de validação de domínio.
7. Quando o status do domínio for alterado paraAtivo, verifique se o domínio funciona para rotear tráfego navegando até ele.

Para desassociar (desvincular) um domínio personalizado usando o console do App Runner

1. NoDomínios personalizados, selecione o bloco do domínio que deseja desassociar e selecioneDesvincular domínio.
2. NoDesvincular domínio, verifique a ação escolhendoDesvincular domínio.

Gerencie domínios personalizados usando a API App Runner ouAWS CLI

Para associar um domínio personalizado ao seu serviço usando a API do App Runner ouAWS CLI, chame o métodoAssociateCustomDomainAção de API. Quando a chamada é bem-sucedida, ela retorna umCustomDomainque descreve o domínio personalizado que está sendo associado ao seu serviço. O objeto deve mostrar um status de`CREATING`e contém uma lista deCertificateValidationRecordObjetos. Estes são registros que você pode adicionar ao seu DNS.

Para desassociar um domínio personalizado do seu serviço utilizando a API do App Runner ouAWS CLI, chame o métodoDisassociateCustomDomainAção de API. Quando a chamada é bem-sucedida, ela retorna umCustomDomainque descreve o domínio personalizado que está sendo desassociado do seu serviço. O objeto deve mostrar um status de`DELETING`.

Pausar e retomar um serviço App Runner

Se você precisar desativar seu aplicativo Web temporariamente e impedir que o código seja executado, você pode pausar seuAWS App RunnerserviçoServiço do O App Runner reduz a capacidade de computação do serviço para zero.

Quando estiver pronto para executar seu aplicativo novamente, você poderá retomar seu serviço App Runner. O App Runner provisiona nova capacidade de computação, implanta seu aplicativo e executa o aplicativo. A origem do aplicativo não é reimplantada e nenhuma compilação é necessária. Em vez disso, o App Runner retoma com sua versão atualmente implantada. Seu aplicativo mantém seu domínio App Runner.

> **Important**
>
> - Quando você pausa seu serviço, seu aplicativo perde seu estado. Por exemplo, qualquer armazenamento efêmero usado pelo código é perdido. Para seu código, pausar e retomar seu serviço é o equivalente a implantar em um novo serviço.

- Se você pausar um serviço devido a uma falha em seu código (por exemplo, um bug descoberto ou problema de segurança), não será possível implantar uma nova versão antes de retomar o serviço.

  Portanto, recomendamos que você mantenha o serviço em execução e reverta para sua última versão estável do aplicativo.
- Quando você retoma seu serviço, o App Runner implanta a última versão do aplicativo usada antes de você pausar o serviço. Se você adicionou novas versões de origem desde a pausa do serviço, o App Runner não as implantará automaticamente, mesmo que a implantação automática esteja selecionada. Por exemplo, suponha que você tenha novas versões de imagem no repositório de imagens ou novas confirmações no repositório de código. Essas versões não são implantadas automaticamente.

  Para implantar uma versão mais recente, execute uma implantação manual ou adicione outra versão ao repositório de origem depois de retomar o serviço App Runner.

Pausar e excluir comparadas

Pauseseu serviço App Runner paratemporariamentedesabilitá-lo. Somente os recursos de computação são encerrados e seus dados armazenados (por exemplo, a imagem do contêiner com a versão do aplicativo) permanecem intactos. Retomar seu serviço é rápido — seu aplicativo está pronto para ser implantado em novos recursos de computação. O domínio do App Runner permanecerá o mesmo.

Excluirseu serviço App Runner paraPermanentementeRemove. Seus dados armazenados são excluídos. Se você precisar recriar o serviço, o App Runner precisa buscar seu código-fonte novamente, e também compilá-lo se for um repositório de código. Seu aplicativo web recebe um novo domínio App Runner.

Quando seu serviço está pausado

Quando você pausar seu serviço e ele está noPaused, ele responde de forma diferente a solicitações de ação, incluindo chamadas de API ou operações de console. Quando um serviço é pausado, você ainda pode executar ações do App Runner que não modificam a definição ou a configuração do serviço de uma forma que afete seu tempo de execução. Em outras palavras, se uma ação alterar o comportamento, a escala ou outras características de um serviço em execução, você não poderá executar essa ação em um serviço pausado.

As listas a seguir fornecem informações sobre ações de API que você pode e não pode executar em um serviço pausado. As operações equivalentes do console são permitidas ou negadas da mesma forma.

Ações que vocêcanexecutar em um serviço pausado

- *List** e *Describe** Ações do— Ações que leem somente informações.
- *DeleteService* — Você sempre pode excluir um serviço.
- *TagResource*, *UntagResource* — As tags são associadas a um serviço, mas não fazem parte de sua definição e não afetam seu comportamento de tempo de execução.

Ações que vocêcannotexecutar em um serviço pausado

- *StartDeployment* Ações do(ou umImplementação manualUsar o console)
- *UpdateService* (ou uma alteração de configuração usando o console, exceto para alterações de marcação)
- *CreateCustomDomainAssociations*, *DeleteCustomDomainAssociations*
- *CreateConnection*, *DeleteConnection*

Pausar e retomar seu serviço usando o console do App Runner

Para pausar seu serviço usando o console do App Runner

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. SelecioneAçõese, depois, escolhaPause.

   Na página do painel de serviço, o serviçoStatusalterações noOperação em andamentoe, em seguida, muda paraPaused. Seu serviço está agora pausado.

Para retomar seu serviço usando o console do App Runner

1. SelecioneAçõese, depois, escolhaRetomar.

   Na página do painel de serviço, o serviçoStatusalterações noOperação em andamento.

2. Aguarde até que o serviço seja retomado. Na página do painel de serviço, o serviçoStatusmuda de volta paraRunning.
3. Para verificar se a retomada do serviço foi bem-sucedida, na página do painel de serviços, escolha a opçãoDomínio do App Runnervalue. É a URL do site do seu serviço. Verifique se o aplicativo da Web está funcionando corretamente.

Pausar e retomar seu serviço usando a API do App Runner ouAWS CLI

Para pausar seu serviço usando a API do App Runner ouAWS CLI, chame oPauseServiceAção de API. Se a chamada retornar uma resposta bem-sucedida com umServiçoobjeto mostrando`"Status": "OPERATION_IN_PROGRESS"`, o App Runner começa a pausar seu serviço.

Para retomar seu serviço usando a API do App Runner ouAWS CLI, chame oResumeServiceAção de API. Se a chamada retornar uma resposta bem-sucedida com umServiçoobjeto mostrando`"Status": "OPERATION_IN_PROGRESS"`, o App Runner começa a retomar o seu serviço.

Excluir um serviço App Runner

Quando você deseja encerrar o aplicativo Web que está sendo executado no seuAWS App RunnerÉ possível excluir o serviço. A exclusão de um serviço interrompe o serviço Web em execução, remove os recursos subjacentes e exclui os dados associados.

Convém excluir um serviço de Executor de aplicativos por um ou mais dos seguintes motivos:

- Você não precisa mais do aplicativo da Web— Por exemplo, ele está desativado ou é uma versão de desenvolvimento que você acabou de usar.
- Você atingiu a cota de serviço do App Runner— Você deseja criar um novo serviço no mesmoRegião da AWSe você atingiu a cota associada à sua conta. Para mais informações, consulte the section called “Cotas de recurso App Executor”.
- Considerações de segurança ou privacidade— Você deseja que o App Runner exclua os dados que ele armazena para seu serviço.

Pausando vs. excluindo

Pauseseu serviço App Runner paratemporariamentedesabilitá-lo. Somente os recursos de computação são encerrados e seus dados armazenados (por exemplo, a imagem do contêiner

com a versão do aplicativo) permanecem intactos. Retomar seu serviço é rápido — seu aplicativo está pronto para ser implantado em novos recursos de computação. O domínio do App Runner permanece o mesmo.

Excluirseu serviço App Runner paraPermanentementeRemove. Seus dados armazenados são excluídos. Se você precisar recriar o serviço, o App Runner precisa buscar seu código-fonte novamente, e também compilá-lo se for um repositório de código. Seu aplicativo web recebe um novo domínio App Runner.

O que o App Runner exclui?

Quando você exclui seu serviço, o App Runner exclui alguns itens associados e não exclui outros. As listas a seguir fornecem os detalhes.

Itens que o App Runner exclui:

- Imagem de contêiner— Uma cópia da imagem que você implantou ou da imagem que o App Runner criou a partir do seu código-fonte. Ele é armazenado no Amazon Elastic Container Registry (Amazon ECR) usandoContas da AWSque são de propriedade do App Runner.
- Configuração do serviço— As definições de configuração associadas ao serviço App Runner. Eles são armazenados no Amazon DynamoDB usandoContas da AWSque são de propriedade do App Runner.

Itens que o App Runner não exclui:

- Conexão do— Você pode ter uma conexão associada ao seu serviço. Uma conexão do App Runner é um recurso separado que pode ser compartilhado entre vários serviços do App Runner. Se não precisar mais da conexão, você poderá excluí-la explicitamente. Para mais informações, consulte the section called “Conexões”.
- Certificados de domínio personalizados— Se você vincular domínios personalizados a um serviço App Runner, o App Runner criará internamente certificados que rastreiam a validade do domínio. Eles estão armazenados emAWS Certificate Manager(ACM). O App Runner não exclui o certificado por sete dias após um domínio ser desvinculado do seu serviço ou após o serviço ser excluído. Para mais informações, consulte the section called “Nomes de domínios personalizados”.

Eliminar o serviço utilizando a consola App Runner

Para excluir seu serviço usando o console App Runner

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. Escolha Ações e, em seguida, escolha Excluir.

   O console o leva para oServiços. O serviço excluído exibe oOperação em andamentoe, em seguida, o serviço desaparece da lista. Seu serviço foi excluído agora.

Exclua seu serviço usando a API do App Runner ouAWS CLI

Para excluir seu serviço usando a API do App Runner ouAWS CLI, chame oDeleteServiceAção de API. Se a chamada retornar uma resposta bem-sucedida com umServiçoobjeto mostrando`"Status": "OPERATION_IN_PROGRESS"`, o App Runner começa a excluir seu serviço.

Registro em log e monitoramento para a sua App Runner

AWS App Runnerintegra-se com váriosAWSpara fornecer a você um amplo conjunto de ferramentas de registro e monitoramento para seu serviço App Runner. Tópicos neste capítulo descrevem esses recursos.

Tópicos

Rastreando atividade do serviço App Runner

AWS App Runnerusa uma lista de operações para acompanhar a atividade no seu serviço App Runner. Uma operação representa uma chamada assíncrona para uma ação de API, como criar um serviço, atualizar uma configuração e implantar um serviço. As seções a seguir mostram como rastrear a atividade no console do App Runner e como usar a API do.

Rastreamento da atividade do serviço App Runner no console

O console do App Runner exibe sua atividade de serviço do App Runner e oferece mais maneiras de explorar as operações.

Como visualizar a atividade do serviço

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, escolhaServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. Na página do painel de serviço, escolha aAtividadesSe ainda não estiver aberto.

   O console exibe uma lista de operações.
4. Para localizar operações específicas, faça o escopo da lista inserindo um termo de pesquisa. Você pode pesquisar qualquer valor exibido na tabela.

5. Escolha qualquer operação listada para ver ou baixar o log relacionado.

Recuperando operações do serviço App Runner usando a API App Runner ouAWS CLI

OListOperationsCom o Amazon Resource Name (ARN — Nome de recurso da Amazon) de um serviço de Executor de aplicativos, retorna uma lista de operações que ocorreram neste serviço. Cada item da lista contém um ID de operação e alguns detalhes de rastreamento.

Visualizar logs do App Runner transmitidos para o CloudWatch Logs

Você pode usar o Amazon CloudWatch Logs para monitorar, armazenar e acessar arquivos de log do que seus recursosAWSgeram serviços. Para obter mais informações, consulteAmazon CloudWatch Logs Guia do usuário.

AWS App Runnercoleta a saída de suas implantações de aplicativos e de seu serviço ativo e o transmite para o CloudWatch Logs. As seções a seguir listam os fluxos de log do App Runner e mostram como exibi-los no console do App Runner.

Grupos de log e streams

O CloudWatch Logs mantém os dados de log em fluxos de log que ele organiza ainda mais em grupos de log. AStream de logé uma sequência de eventos de log de uma origem específica. Um grupo de logs é um grupo de fluxos de log que compartilham as mesmas configurações de retenção, monitoramento e controle de acesso.

O App Runner define dois grupos de logs do CloudWatch Logs, cada um com vários fluxos de log, para cada um de seus serviços do App Runner em seuConta da AWS.

Registros de serviço

O grupo de log de serviço contém saída de registro gerada pelo App Runner, pois gerencia seu serviço App Runner e atua sobre ele.

nome do grupo de logs	Exemplo
`/aws/apprunner/` *`service-name`* `/` *`service-id`* `/service`	`/aws/apprunner/python-test/ac7ec8b51ff34746bcb6654e0bcb23da/service`

No grupo de logs de serviço, o App Runner cria um fluxo de log de eventos para capturar atividades no ciclo de vida do seu serviço App Runner. Por exemplo, isso pode ser iniciar seu aplicativo ou pausá-lo.

Além disso, o App Runner cria um fluxo de log para cada operação assíncrona de longa duração relacionada ao seu serviço. O nome do fluxo de log reflete o tipo de operação e o ID de operação específico.

Aimplantaçãoé um tipo de operação. Os logs de implantação contêm a saída de log das etapas de compilação e implantação executadas pelo App Runner quando você cria um serviço ou implanta uma nova versão do aplicativo. Os nomes do fluxo de log de implantação começam com`deployment/`e termine com o ID da operação que executa a implantação. Esta operação é umCreateServicechamada para a implantação inicial do aplicativo ou umStartDeploymentchamar para cada implantação adicional.

Em um log de implantação, cada mensagem de log começa com um prefixo:

- `[AppRunner]`— Saída que o App Runner gera durante a implantação.
- `[Build]`— Saída de seus próprios scripts de compilação.

Nome do fluxo do log	Exemplo
`events`	N/D (nome fixo)
`operation-type` `/` *`operation-id`*	`deployment/c2c8eeedea164f459cf78f12a8953390`

Logs de aplicativos

O grupo de logs de aplicativos contém a saída do código do aplicativo em execução.

nome do grupo de logs	Exemplo
`/aws/apprunner/ service-name /service-id /application`	`/aws/apprunner/python-test/ ac7ec8b51ff34746bcb6654e0bcb23da/ application`

No grupo de logs de aplicativos, o App Runner cria um fluxo de log para cada instância (unidade de dimensionamento) que está executando seu aplicativo.

Nome do fluxo do log	Exemplo
`instance/ instance-id`	`instance/1a80bc9134a84699b7 b3432ebeebb591`

Visualizar logs do App Runner no console do

A consola App Runner apresenta um resumo de todos os registos para o seu serviço e permite-lhe visualizá-los, explorar e transferi-los.

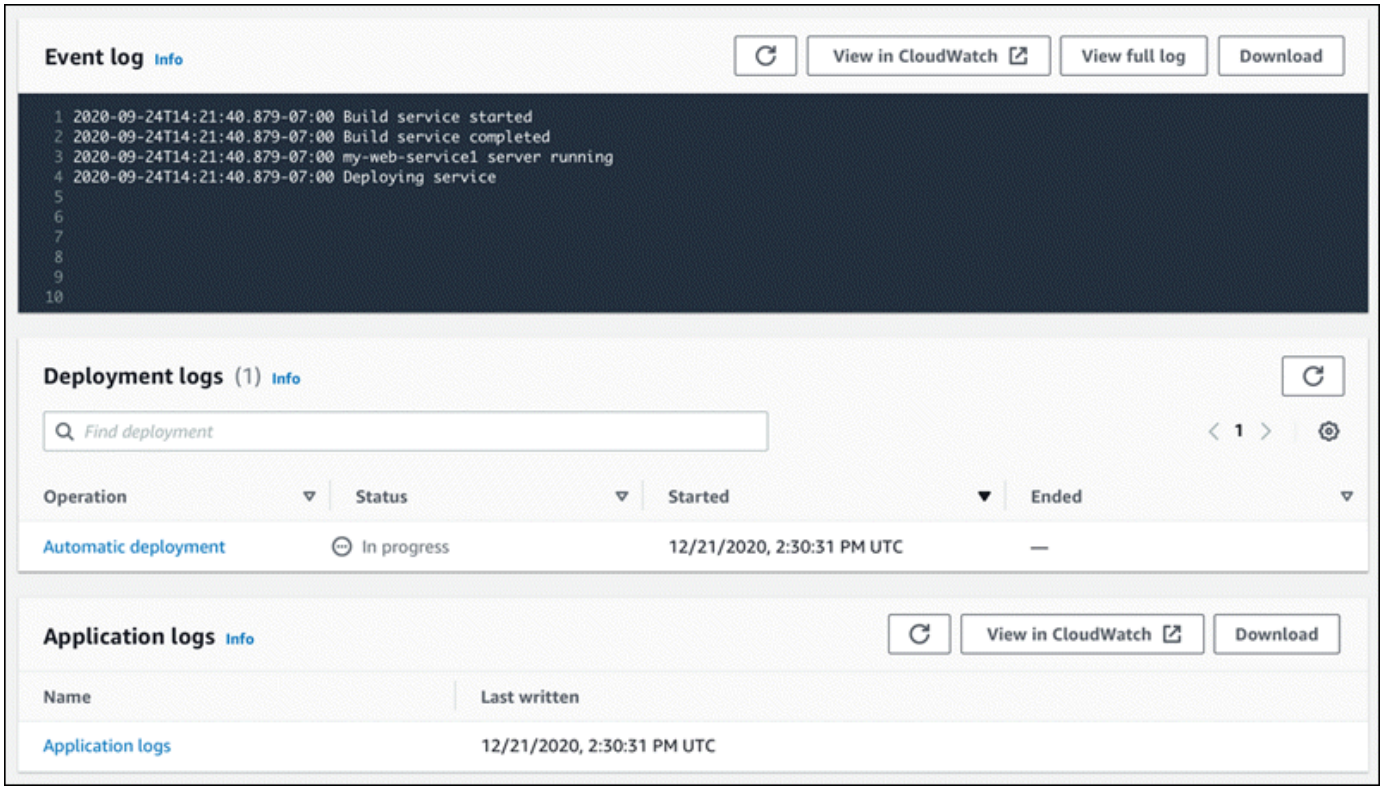
Para visualizar registros do serviço

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, selecioneServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. Na página do painel de serviço, selecione oLogsGuia.

   O console exibe alguns tipos de logs em várias seções:

   - Log de eventos— Atividade no ciclo de vida do seu serviço App Runner. O console exibe os eventos mais recentes.
   - Registros de implantação— Implantações de repositório de origem para o seu serviço App Runner. O console exibe um fluxo de log separado para cada implantação.
   - Logs de aplicativos— A saída do aplicativo Web que é implantado no seu serviço App Runner. O console combina a saída de todas as instâncias em execução em um único fluxo de log.



4. Para localizar implantações específicas, defina o escopo da lista de logs de implantação inserindo um termo de pesquisa. Você pode pesquisar qualquer valor exibido na tabela.
5. Para exibir o conteúdo de um log, escolhaVisualizar registro completo(log de eventos) ou o nome do fluxo de log (logs de implantação e aplicativos).
6. SelecioneBaixarpara fazer o download de um log. Para um fluxo de log de implantação, selecione primeiro um fluxo de log.
7. SelecioneVisualizar no CloudWatchpara abrir o console do CloudWatch e usar seus recursos completos para explorar seus logs de serviço do App Runner. Para um fluxo de log de implantação, selecione primeiro um fluxo de log.

> **Note**
>
> O console do CloudWatch é particularmente útil se você quiser exibir logs de aplicativos de instâncias específicas em vez do log de aplicativos combinado.

Exibindo métricas de serviço do App Runner relatadas ao CloudWatch

O Amazon CloudWatch monitora seus Amazon Web Services (AWS) e os aplicativos que você executa naAWSEm tempo real. Você pode usar o CloudWatch para coletar e monitorar métricas, que são as variáveis que é possível medir para avaliar seus recursos e aplicativos. Você também pode usar o para criar alarmes que observam métricas. Quando um determinado limite é atingido, o CloudWatch envia notificações ou faz alterações automaticamente nos recursos monitorados. Para obter mais informações, consulteGuia do usuário do Amazon CloudWatch.

AWS App Runnercoleta uma variedade de métricas que fornecem maior visibilidade do uso, desempenho e disponibilidade dos serviços do App Runner. Algumas métricas rastreiam instâncias individuais que executam seu serviço Web, enquanto outras estão no nível de serviço geral. As seções a seguir listam as métricas do App Runner e mostram como exibi-las no console do App Runner.

Métricas do App Runner

O App Runner coleta as seguintes métricas relacionadas ao seu serviço e as publica no CloudWatch no`AWS/AppRunner`Namespace.

Métricas nível de instânciasão coletados para cada instância (unidade de escala) individualmente.

O que é medido?	Métrica	Descrição
CPU utilization	`CPUUtilization`	O uso médio da CPU durante períodos de um minuto.
Memory utilization	`MemoryUtilization`	O uso médio de memória durante períodos de um minuto.

Métricas nível de serviçoSão coletadas para cada serviço da.

O que é medido?	Métricas	Descrição
HTTP request count	`Requests`	O número de solicitações HTTP que o serviço recebeu.

O que é medido?	Métricas	Descrição

HTTP status counts	`2xxStatus Responses`
	
	`4xxStatus Responses`
	
	`5xxStatus Responses`
		O número de solicitações HTTP que retornaram cada status de resposta, agrupadas por categoria (2XX, 4XX, 5XX).

HTTP request latency	`RequestLatency`	O tempo que levou seu serviço Web para processar solicitações HTTP.
Instance counts	`ActiveInstances`	O número de instâncias que estão processando solicitações HTTP para o serviço.

Visualizar métricas do App Runner no console do

O console do App Runner exibe graficamente as métricas que o App Runner coleta para seu serviço e fornece mais maneiras de explorá-las.

> **Note**
>
> No momento, o console exibe apenas métricas de serviço. Para exibir métricas de instância, use o console do CloudWatch.

Para exibir logs do serviço

1. Abrir oConsole do App Runner, e noRegiões, selecione suaRegião da AWS.
2. No painel de navegação, escolhaServiçose, em seguida, escolha o serviço App Runner.

   O console exibe o painel de serviço com umVisão geral do serviço.
3. Na página do painel de serviço, escolha oMétricasGuia.

   O console exibe um conjunto de gráficos de métricas.

4. Escolha uma duração (por exemplo,12h) para definir o escopo de gráficos de métricas para o período recente dessa duração.
5. SelecioneAdicionar ao painelna parte superior de uma das seções do gráfico, ou use o menu em qualquer gráfico, para adicionar as métricas relevantes a um painel no console do CloudWatch para uma investigação mais aprofundada.

Manipulando eventos de App Runner no EventBridge

Usando o Amazon EventBridge, você pode configurar regras orientadas por eventos que monitoram um fluxo de dados em tempo real doAWS App Runnerserviço para certos padrões. Quando um padrão para uma regra é correspondido, o EventBridge inicia uma ação em um destino, comoAWS Lambda, Amazon ECS,AWS Batche o Amazon SNS. Por exemplo, você pode definir uma regra para enviar notificações por e-mail sinalizando um tópico do Amazon SNS sempre que uma implantação no seu serviço falhar. Ou você pode definir uma função do Lambda para notificar um canal do Slack sempre que uma atualização do serviço falhar. Para obter mais informações sobre o EventBridge, consulteGuia do usuário do Amazon EventBridge.

O App Runner envia os seguintes tipos de eventos para o EventBridge

- Alteração do estado do— Uma alteração no status de um serviço App Runner. Por exemplo, um status de serviço alterado para`DELETE_FAILED`.
- Alteração do estado da operação do— Uma alteração no status de uma operação longa e assíncrona em um serviço App Runner. Por exemplo, um serviço começou a criar, uma atualização de serviço concluída com êxito ou uma implantação de serviço concluída com erros.

Criar uma regra EventBridge para agir em eventos do App Runner

Um EventBridgeEventoé um objeto que define alguns campos padrão EventBridge, como a fonteAWSe o tipo de detalhe (evento) e um conjunto de campos específico do evento com os detalhes do evento. Para criar uma regra EventBridge, use o console EventBridge para definir umPadrão de evento(quais eventos devem apostar rastreados) e especificar umação de destino(o que deve ser feito em uma partida). Um padrão de evento é semelhante aos eventos que ele corresponde. Especifique um subconjunto de campos a serem correspondidos e, para cada campo, especifique uma lista de valores possíveis. Este tópico fornece exemplos de eventos e padrões de eventos do App Runner.

Para obter mais informações sobre como criar regras do EventBridge, consulteCriar uma regra para umAWSserviçonoGuia do usuário do Amazon EventBridge.

> **Note**
>
> Alguns serviços de suportepadrões predefinidosno EventBridge. Isso simplifica a forma como um padrão de evento é criado. Você seleciona valores de campo em um formulário e EventBridge gera o padrão para você. No momento, o App Runner não suporta padrões predefinidos. Você tem que inserir o padrão como um objeto JSON. Você pode usar os exemplos deste tópico como um ponto de partida.

Exemplos de eventos App Runner

Estes são alguns exemplos de eventos que o App Runner envia para o EventBridge.

- Evento de alteração de status do serviço. Especificamente, um serviço que mudou do`OPERATION_IN_PROGRESS`para o`RUNNING`Status.

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

- Evento de alteração de status da operação. Mais especificamente, um`UpdateService`Operação concluída com êxito.

```
{
  "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"
  }
}
```

Exemplos de padrão de evento App Runner

Os exemplos a seguir demonstram padrões de eventos que você pode usar nas regras do EventBridge para corresponder a um ou mais eventos do App Runner. Um padrão de evento é semelhante a um evento. Inclua apenas os campos que você deseja corresponder e forneça uma lista em vez de um escalar para cada um.

- Corresponder a todos os eventos de alteração de status de serviço para serviços de uma conta específica, onde o serviço não está mais em`RUNNING`Status.

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

```
  }
}
```

- Corresponder todos os eventos de alteração de status da operação para serviços de uma conta específica, onde a operação falhou.

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

Referência de evento do App Runner

Alteração do estado do

Um evento de alteração de status do serviço`detail-type`definido como`Service Status Change`. Ela tem os seguintes campos de detalhes e valores:

```
"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"
```

Alteração do estado da

Um evento de alteração de status da operação tem`detail-type`definido como`Service Operation Status Change`. Ela tem os seguintes campos de detalhes e valores:

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

A tabela a seguir lista todos os códigos de status e mensagens relacionadas.

Status	Message
`CreateServiceStarted`	Criação de serviços iniciada.
`CreateServiceCompletedSuccessfully`	Criação de serviços concluída com êxito.
`CreateServiceFailed`	Falha na criação de serviços. Para obter mais detalhes, consulte logs do serviço.
`DeleteServiceStarted`	Eliminação do serviço iniciada.
`DeleteServiceCompletedSuccessfully`	Eliminação do serviço concluída com êxito.
`DeleteServiceFailed`	Falha na exclusão do serviço.
`UpdateServiceStarted`	
`UpdateServiceCompletedSuccessfully`	Atualização do serviço concluída com êxito. Novo aplicativo e configuração são implantados.
	Atualização do serviço concluída com êxito. Nova configuração é implantada.
`UpdateServiceFailed`	Falha na atualização do serviço. Para obter mais detalhes, consulte logs do serviço.
`DeploymentStarted`	Implementação iniciada.
`DeploymentCompletedSuccessfully`	Implantação concluída com êxito.

Status	Message
`DeploymentFailed`	Falha na implantação. Para obter mais detalhes, consulte logs do serviço.
`PauseServiceStarted`	Pausa do serviço iniciada.
`PauseServiceCompletedSuccessfully`	Pausa do serviço concluída com êxito.
`PauseServiceFailed`	Falha na pausa do serviço.
`ResumeServiceStarted`	Reinício do serviço.
`ResumeServiceCompletedSuccessfully`	Retomada do serviço concluída com êxito.
`ResumeServiceFailed`	Falha no reinício do serviço.

Registrar em log as chamadas de API do AppAWS CloudTrail

O App Runner está integrado aoAWS CloudTrail, um serviço que fornece um registro das ações executadas por um usuário, uma função ou umaAWSno App Runner. O CloudTrail captura todas as chamadas de API para o App Runner como eventos. As chamadas capturadas incluem chamadas do console do App Runner e as chamadas de código para as operações de API do App Runner. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o App Runner. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o App Runner, o endereço IP no qual foi feita a solicitação, quem a fez e quando ela foi feita, além de detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte oAWS CloudTrailGuia do usuário.

Informações sobre o App Runner no CloudTrail

O CloudTrail está habilitado noConta da AWSQuando você cria a conta. Quando ocorre atividade no App Runner, essa atividade é registrada em um evento do CloudTrail junto com outrosAWSeventos de serviço emHistórico de eventos. Você pode visualizar, pesquisar e fazer download de eventos

recentes em seuConta da AWS. Para obter mais informações, consulte Como visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos em suaConta da AWS, incluindo eventos para o App Runner, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas asRegiões da AWS. A trilha registra em log os eventos de todas as regiões noAWSO fornece os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, você pode configurar outrasAWSPara analisar mais profundamente e agir sobre os dados de eventos coletados nos logs do CloudTrail. Para mais informações, consulte:

- Visão geral da criação de uma trilha
- Serviços e integrações compatíveis com o CloudTrail
- Configuração de notificações do Amazon SNS para o CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões e receber arquivos de log do CloudTrail de várias contas

Todas as ações do App Runner são registradas pelo CloudTrail e documentadas naAWS App RunnerReferência de API do. Por exemplo, as chamadas para as APIs `CreateService`, `DeleteConnection` e `StartDeployment` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte o Elemento userIdentity do CloudTrail.

Noções básicas sobre as entradas de arquivos de log do

Uma trilha é uma configuração que permite a entrega de eventos como registros de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação

solicitada, a data e a hora da ação e os parâmetros de solicitação. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação do `CreateService`.

> **Note**
>
> Por motivos de segurança, alguns valores de propriedade são editados nos logs e substituídos pelo texto`HIDDEN_DUE_TO_SECURITY_REASONS`. Isso evita a exposição não intencional de informações secretas. No entanto, você ainda pode ver que essas propriedades foram transmitidas na solicitação ou retornadas na resposta.

Exemplo de entrada de log do CloudTrail para o**`CreateService`**Ação do App Runner

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

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

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

Definir opções de serviço do App Runner usando um arquivo de configuração

> **Note**
>
> Os arquivos de configuração são aplicáveis somente aoserviços baseados no código-fonte. Não é possível usar arquivos de configuração com osserviços baseados em imagem.

Quando você cria umAWS App RunnerServiço usando um repositório de códigos-fonteAWS App Runnerrequer informações sobre como construir e iniciar seu serviço. Você pode fornecer essas informações sempre que criar um serviço usando o console ou API do App Runner. Como alternativa, defina opções de serviço usando umArquivo de configuração. As opções especificadas em um arquivo se tornam parte do repositório de origem e quaisquer alterações nessas opções são acompanhadas de forma semelhante à forma como as alterações no código-fonte são controladas. Você pode usar o arquivo de configuração do App Runner para especificar mais opções do que as suportadas pela API. Você não precisa fornecer um arquivo de configuração se precisar apenas das opções básicas suportadas pela API.

O arquivo de configuração do App Runner é um arquivo YAML chamado`apprunner.yaml`No diretório raiz do repositório do seu aplicativo. Ele fornece opções de compilação e tempo de execução para o seu serviço. Os valores neste arquivo instruem o App Runner como criar e iniciar seu serviço, além de fornecer contexto de tempo de execução, como configurações de rede e variáveis de ambiente.

O arquivo de configuração do App Runner não inclui configurações operacionais, como CPU e memória.

Para obter exemplos de arquivos de configuração do App Runner, consultethe section called “Exemplos”. Para obter um guia de referência completo, consultethe section called “Referência”.

Tópicos

- Exemplos de arquivos de configuração do App Run
- Referência do arquivo de configuração do App Runner

Exemplos de arquivos de configuração do App Run

> **Note**
>
> Os arquivos de configuração são aplicáveis somente aoserviços baseados no código-fonte. Não é possível usar arquivos de configuração do comserviços baseados em imagem.

Os exemplos a seguir demonstramAWS App RunnerArquivos de configuração. Alguns são mínimos e contêm apenas as configurações necessárias. Outros estão completos, incluindo todas as seções do arquivo de configuração. Para obter uma visão geral dos arquivos de configuração do App Runner, consulteArquivo de configuração do App Runner.

Exemplo de arquivo de configuração

Arquivo de configuração mínimo

Com um arquivo de configuração mínimo, o App Runner faz as seguintes suposições:

- Nenhuma variável de ambiente personalizada é necessária durante a compilação ou execução.
- A versão do runtime mais recente é usada.
- O número de porta padrão e a variável de ambiente de porta são usados.

Example apprunner.yaml

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

Arquivo de configuração completo

Este exemplo mostra o uso de todas as chaves de configuração com um tempo de execução gerenciado.

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"
```

Para obter exemplos de arquivos de configuração de runtime gerenciados específicos, consulte o subtópico de tempo de execução específico emServiço baseado em código.

Referência do arquivo de configuração do App Runner

> **Note**
>
> Os arquivos de configuração são aplicáveis somente aoserviços que são baseados no código-fonte. Não é possível usar arquivos de configuração com oserviços baseados em imagem.

Este tópico é um guia de referência abrangente para a sintaxe e semântica de umAWS App RunnerArquivo de configuração. Para uma visão geral dos arquivos de configuração do App Runner, consulteArquivo de configuração do App Runner.

O arquivo de configuração do App Runner é um arquivo YAML. Nomeie`apprunner.yaml`E coloque-o no diretório raiz do repositório do aplicativo.

Visão geral da estrutura

O arquivo de configuração do App Runner é um arquivo YAML. Nomeie`apprunner.yaml`E coloque-o no diretório raiz do repositório do aplicativo.

O arquivo de configuração do App Runner contém as seguintes partes principais:

- Seção principal— Contém chaves de nível superior
- Seção de compilação— Configura o estágio de compilação
- Seção Executar— Configura o estágio de tempo de execução

Seção principal

As chaves na parte superior do arquivo fornecem informações gerais sobre o arquivo e o tempo de execução do serviço. As seguintes chaves estão disponíveis:

- `version`:Obrigatório. A versão do arquivo de configuração do App Runner. Idealmente, use a versão mais recente do.

  Sintaxe

  ```
  version: version
  ```

  Example

  ```
  version: 1.0
  ```

- `runtime`:Obrigatório. O nome do tempo de execução que seu aplicativo usa. Os seguintes tempos de execução estão disponíveis no momento:

  `python3`, `nodejs12`

> **Note**
>
> A convenção de nomeação de um tempo de execução gerenciado é*<language-name><major-version>*.

Sintaxe

```
runtime: runtime-name
```

Example

```
runtime: python3
```

Seção Construir

A seção de compilação configura o estágio de compilação da implantação do serviço App Runner. É possível especificar comandos de build e variáveis de ambiente. Comandos de compilação são necessários.

A seção começa com o`build:`e tem as seguintes subchaves:

- `commands`:Obrigatório. Especifica os comandos que o App Runner executa durante várias fases de compilação. Inclui as seguintes subchaves:
  - `pre-build`:Optional. Os comandos que o App Runner executa antes da compilação. Por exemplo, instalenpmdependências ou bibliotecas de teste.
  - `build`:Obrigatório. Os comandos que o App Runner executa para criar seu aplicativo. Por exemplo, usepipenv.
  - `post-build`:Optional. Os comandos que o App Runner executa depois da compilação. Por exemplo, use o Maven para empacotar artefatos de build em um arquivo JAR ou WAR, ou execute um teste.

Sintaxe

```
build:
  commands:
```

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

Example

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

- env:Optional. Especifica variáveis de ambiente personalizadas para o estágio de construção. Definido como mapeamentos escalares de nome-valor. Você pode se referir a essas variáveis pelo nome em seus comandos de build.

  Sintaxe

```
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"
```

Seção Executar

A seção de execução configura o estágio de execução do contêiner da implantação do aplicativo App Runner. Você pode especificar a versão do tempo de execução, o comando start, a porta de rede e as variáveis de ambiente.

A seção começa com `orun:` e tem as seguintes subchaves:

- `runtime-version`:Optional. Especifica uma versão de tempo de execução que você deseja bloquear para o serviço App Runner.

  Por padrão, apenas a versão principal está bloqueada. O App Runner usa as versões secundárias e de patch mais recentes que estão disponíveis para o tempo de execução em cada implantação ou atualização de serviço. Se você especificar as versões principais e secundárias, ambas ficarão bloqueadas e o App Runner atualizará apenas as versões de patch. Se você especificar versões principais, secundárias e de patch, seu serviço estará bloqueado em uma versão específica do tempo de execução e o App Runner nunca o atualizará.

  Sintaxe

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

  Example

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

- `command`:Obrigatório. O comando que o App Runner usa para executar seu aplicativo depois que ele concluir a compilação do aplicativo.

  Sintaxe

  ```
  run:
    command: command
  ```

- `network`:Optional. Especifica a porta que o aplicativo escuta. Ele inclui o seguinte:
  - `port`:Optional. Se especificado, esse é o número da porta que o aplicativo escuta. O padrão é `8080`.
  - `env`:Optional. Se especificado, o App Runner passa o número da porta para o contêiner nessa variável de ambiente, além de (não em vez de) passar o mesmo número de porta na variável de ambiente padrão,`PORT`. Em outras palavras, se você especificar`env`, o App Runner passa o número da porta em duas variáveis de ambiente.

  Sintaxe

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

  Example

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

- `env`:Optional. Definição de variáveis de ambiente personalizadas para o estágio de execução. Definido como mapeamentos escalares de nome-valor. Você pode se referir a essas variáveis por nome em seu ambiente de tempo de execução.

  Sintaxe

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

  Example

```
run:
  env:
```

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

A API do App Runner

OAWS App Runnerinterface de programação de aplicativos (API) é uma API RESTful para fazer solicitações ao serviço App Runner. Você pode usar a API do para criar, listar, descrever, atualizar e excluir recursos do App Runner noConta da AWS.

Você pode chamar a API do seu código de aplicativo ou usar um dosAWSSDKs. Para scripts de linha de comando, use oAWS CLIpara fazer chamadas para o serviço App Runner. Para obter mais informações sobreAWSferramentas do desenvolvedor, consulteFerramentas em que se baseiaAWS.

Para obter informações de referência de API completas, consulte aAWS App RunnerReferência de API do.

Segurança no App Runner

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de datacenters e arquiteturas de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem:AWSA é responsável por proteger a infraestrutura que executa oAWSServiços da naNuvem AWS.AWSA também fornece serviços que você pode usar com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos Programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam aoAWS App Runner, consulteAWSServiços da no escopo pelo programa de conformidade.
- Segurança na nuvem— Sua responsabilidade é determinada peloAWSque você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o App Runner. Os tópicos a seguir mostram como configurar o App Runner para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outrosAWSServiços da que ajudam a monitorar e proteger os recursos do App Runner.

Tópicos

Proteção de dados no App Runner

OAWS Modelo de responsabilidade compartilhadaAplica-se à proteção de dados noAWS App Runner. Conforme descrito neste modelo,AWSO é responsável pela proteção da infraestrutura global que executa todas asAWSNuvem. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança doAWSque você usa. Para obter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para obter informações sobre a proteção de dados na Europa, consulte oAWSModelo de responsabilidade compartilhada e GDPRPublicação noAWSBlog de segurança.

Para fins de proteção de dados, recomendamos que você proteja oAWSe configure contas de usuário individuais com oAWS Identity and Access Management(IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Recomendamos TLS 1.2 ou posterior.
- Configure a API e o registro em log das atividades do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão nos serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados pessoais armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para obter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como números de conta dos seus clientes, em campos de formato livre, como um campo Name (Nome). Isso inclui quando você trabalha com o App Runner ou outrosAWSusando o console, API,AWS CLI, ouAWSSDKs. Todos os dados inseridos por você no App Runner ou em outros serviços podem ser selecionados para inclusão em logs de diagnóstico. Ao fornecer um URL para um servidor externo, não inclua informações de credenciais no URL para validar a solicitação a esse servidor.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Tópicos

- Proteção de dados usando criptografia
- Privacidade do tráfego entre redes
- Usar o App Runner com VPC endpoints

Proteção de dados usando criptografia

AWS App Runnerlê a origem do aplicativo (imagem de origem ou código-fonte) de um repositório especificado e o armazena para implantação no serviço. Para mais informações, consulte Arquitetura e conceitos.

A proteção de dados refere-se à proteção de dados enquantoEm trânsito(como ele viaja de e para App Runner) eem repouso(enquanto ele é armazenado emAWSdata centers).

Para obter mais informações sobre a proteção de dados, consultethe section called “Proteção de dados”.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Criptografia em trânsito

É possível obter proteção de dados em trânsito de duas maneiras: criptografar a conexão usando Transport Layer Security (TLS) ou usar a criptografia do lado do cliente (onde o objeto é criptografado antes de ser enviado). Ambos os métodos são válidos para proteger os dados do aplicativo. Para proteger a conexão, criptografe-a usando TLS sempre que o aplicativo e seus desenvolvedores, administradores e usuários finais enviarem ou receberem objetos. App Runner configura seu aplicativo para receber tráfego via TLS.

A criptografia do lado do cliente não é um método válido para proteger a imagem de origem ou o código fornecido ao App Runner para implantação. O App Runner precisa de acesso à fonte da aplicação, para que ela não possa ser criptografada. Portanto, certifique-se de proteger a conexão entre o ambiente de desenvolvimento ou de implantação e o App Runner.

Criptografia em repouso e gerenciamento de chaves

Para proteger os dados em repouso do aplicativo, o App Runner criptografa todas as cópias armazenadas da imagem de origem do aplicativo ou do pacote de origem. Ao criar um serviço do

App Runner, você pode fornecer uma chave mestra do cliente (CMK). Se você fornecer um, o App Runner usará sua chave fornecida para criptografar sua fonte. Se você não fornecer uma, o App Runner usa umAWSCMK gerenciada pelo contrário.

Para obter detalhes sobre os parâmetros de criação do serviço App Runner, consulteCreateService. Para obter informações sobre oAWS Key Management Service(AWS KMS), consulte oAWS Key Management ServiceGuia do desenvolvedor.

Privacidade do tráfego entre redes

O App Runner usa o Amazon Virtual Private Cloud (Amazon VPC) para criar limites entre recursos em sua aplicação do App Runner e controlar o tráfego entre eles, sua rede no local e a internet. Para obter mais informações sobre segurança da Amazon VPC, consultePrivacidade do tráfego entre redes na Amazon VPCnoGuia do usuário da Amazon VPC.

Para obter informações sobre como proteger solicitações ao App Runner usando um VPC endpoint, consultethe section called “VPC endpoints”.

Para obter mais informações sobre a proteção de dados, consultethe section called “Proteção de dados”.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Usar o App Runner com VPC endpoints

SuasAWSaplicativo pode integrarAWS App RunnerServiços com outrosAWSem execução em umAmazon Virtual Private Cloud (Amazon VPC). Partes do aplicativo podem fazer solicitações ao App Runner de dentro da VPC. Por exemplo, você pode usar oAWS CodePipelinepara implantar continuamente no seu serviço App Runner. Uma maneira de melhorar a segurança do seu aplicativo é enviar esses pedidos do App Runner (e solicitações para outrosAWS) em um endpoint da VPC.

Um VPC endpoint permite conectar de forma privada sua VPC aos serviços compatíveis da AWS e aos serviços de VPC endpoint habilitados pelo AWS PrivateLink, sem exigir um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect.

Os recursos da sua VPC não usam endereços IP públicos para interagir com os recursos do App Runner. O tráfego entre a VPC e o App Runner não sai da rede da Amazon. Para obter informações completas sobre VPC endpoints, consulteVPC endpointsnoAWS PrivateLinkGuia.

App Runner oferece suporte aoAWS PrivateLinkO oferece conectividade privada ao App Runner e elimina a exposição do tráfego à Internet pública. Para permitir que seu aplicativo envie solicitações

para o App Runner usandoAWS PrivateLink, você configura um tipo de endpoint da VPC conhecido comoVPC endpoint de interface(ponto final da interface). Para obter mais informações, consulteVPC endpoints de interface (AWS PrivateLink)noAWS PrivateLinkGuia.

> **Note**
>
> O aplicativo Web em seu serviço App Runner é executado em uma VPC que o App Runner fornece e configura. Esta VPC é pública — está conectada à Internet pública. O App Runner não oferece suporte à execução do seu aplicativo em uma VPC privada ou à criação de um endpoint da VPC para ele.

Configurar um VPC endpoint para o App Runner

Para criar o VPC endpoint de interface para o serviço do App Runner em sua VPC, siga asCriar um endpoint de interfacenoAWS PrivateLinkGuia. Em Service Name (Nome do serviço), escolha `com.amazonaws.`*`region`*`.apprunner`.

Considerações sobre privacidade de rede

> **Important**
>
> O uso de um endpoint da VPC para App Runner não garante que todo o tráfego de sua VPC permaneça fora da Internet. A VPC pode ser pública (conectada à Internet), e algumas partes da solução podem não usar endpoints da VPC para tornarAWSChamadas de API. Por exemplo,AWSOs serviços da podem chamar outros serviços da usando seus endpoints públicos. Se a privacidade de tráfego for necessária para a solução em sua VPC, leia esta seção.

Para garantir a privacidade do tráfego de rede na VPC, considere o seguinte:

- Habilitar nome DNS— Partes do seu aplicativo ainda podem enviar solicitações ao App Runner pela internet usando `o``apprunner.`*`region`*`.amazonaws.com`O endpoint público. Se a VPC estiver configurada com acesso público à Internet, essas solicitações serão bem-sucedidas sem nenhuma indicação para você. É possível impedir isso habilitando Enable DNS name (Habilitar nome DNS) durante a criação do endpoint (true por padrão). Isso adiciona uma entrada DNS em sua VPC, que mapeia o endpoint público de serviço para o VPC endpoint de interface.

- Configurar VPC endpoints para serviços adicionais— Sua solução pode enviar solicitações para outrosAWSServiços da . Por exemplo,AWS CodePipelinepode enviar solicitações paraAWS CodeBuild. Configure VPC endpoints para esses serviços também e ative nomes DNS nesses endpoints.

Usar políticas de endpoint para controlar o acesso com VPC endpoints

Por padrão, um VPC endpoint permite acesso total ao serviço ao qual está associado. Ao criar ou modificar um VPC endpoint para o App Runner, é possível anexar umPolítica de endpointa ele.

Uma política de ponto de extremidade é umaAWS Identity and Access Management(IAM) que controla o acesso do endpoint ao serviço especificado. A política de endpoint é específica ao endpoint. Ela é separada de outras políticas do IAM de usuário ou instância que seu ambiente possa ter e não as substitui. Para obter detalhes sobre como criar e usar políticas de VPC endpoint, consulteControle do acesso a serviços com VPC endpointsnoAWS PrivateLinkGuia.

O exemplo a seguir permite acesso somente leitura do endpoint da VPC ao App Runner. Esses são direitos de acesso mínimos ao usar o endpoint da VPC. Os principais podem ter permissões adicionais por meio de outras políticas.

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

Identity and Access Management para o App Runner

AWS Identity and Access Management(IAM) é umaAWSO serviço do que ajuda a controlar o acesso aoAWSrecursos da AWS. Os administradores do IAM controlam quem pode serAutenticado(conectado) eautorizado(ter permissões) para usar os recursos do App Runner. O IAM é umAWSO serviço que pode ser usado sem custo adicional.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Tópicos

Audience

Como você usa oAWS Identity and Access ManagementO (IAM) varia, dependendo do trabalho que você realiza no App Runner.

Usuário do serviço— se você usar o serviço App Runner para fazer o trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que usar mais recursos do App Runner forem usados para fazer seu trabalho, talvez sejam necessárias permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no App Runner, consulteSolução de problemas de identidade e acesso.

Administrador de serviçosSe você for o responsável pelos recursos do App Runner na sua empresa, provavelmente terá acesso total ao App Runner. Seu trabalho é determinar quais recursos do App Runner os funcionários devem acessar. Assim, é necessário enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como a empresa pode usar o IAM com o App Runner, consulteComo o App Runner funciona com o IAM.

Administrador do IAM— se você é um administrador do IAM, talvez queira saber detalhes sobre como pode escrever políticas para gerenciar o acesso ao App Runner. Para visualizar exemplos de políticas baseadas em identidade do App Runner do que podem ser usadas no IAM, consulteExemplos de políticas baseadas em identidade do App Runner.

Autenticar com identidades

A autenticação é a forma como você faz login na AWS usando suas credenciais de identidade. Para obter mais informações sobre como fazer login usando oAWS Management Console, consulteFazer login noAWS Management ConsoleComo um usuário do IAM ou usuário raiznoGuia do usuário do IAM.

Você deve serAutenticado(conectado aoAWS) como oAWSO usuário raiz da conta do, um usuário do IAM ou assumindo uma função do IAM. Também é possível usar a autenticação de logon único da sua empresa ou até mesmo fazer login usando o Google ou o Facebook. Nesses casos, o administrador configurou anteriormente federação de identidades usando funções do IAM. Ao acessar a AWS usando credenciais de outra empresa, você estará assumindo uma função indiretamente.

Para iniciar sessão diretamente naAWS Management ConsoleUse sua senha do com o e-mail do usuário raiz ou com o nome do usuário do IAM. Você pode acessar oAWSUsando programaticamente o usuário raiz ou as chaves de acesso dos usuários do IAM.AWSO fornece o SDK e as ferramentas da linha de comando para assinar sua solicitação de forma criptográfica usando suas credenciais. Se você não utilizar as ferramentas da AWS, você deverá assinar a solicitação por conta própria. Faça isso usando o Signature versão 4, um protocolo para autenticação de solicitações de API de entrada. Para obter mais informações sobre autenticação de solicitações, consulteProcesso de assinatura do Signature versão 4noAWSReferência geral.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulteUsar autenticação multifator (MFA) noAWSnoGuia do usuário do IAM.

Usuário raiz da conta da AWS

Ao criar uma conta da AWS, você começa com uma única identidade de login que tenha acesso total a todos os recursos e serviços da AWS na conta. Essa identidade é chamadaAWSaccountUsuário raizE é acessada pelo login com o endereço de e-mail e a senha usados para criar a conta. Recomendamos que não use o usuário raiz para suas tarefas diárias, nem mesmo as administrativas. Em vez disso, siga as práticas recomendadas para o uso do usuário raiz somente a fim de criar seu primeiro usuário do IAM. Depois, armazene as credenciais do usuário raiz com segurança e use-as para executar somente algumas tarefas de gerenciamento de contas e de serviços.

Grupos e usuários do IAM

UmaUsuário do IAMé uma identidade dentro do seuAWSA conta do que tem permissões específicas para uma única pessoa ou um único aplicativo. Um usuário do IAM pode ter credenciais de longo prazo, como um nome de usuário e uma senha ou um conjunto de chaves de acesso. Para saber como gerar chaves de acesso, consulte Gerenciar chaves de acesso para usuários do IAM no Guia do usuário do IAM. Ao gerar chaves de acesso para um usuário do IAM, visualize e salve o par de chaves de maneira segura. Não será possível recuperar a chave de acesso secreta futuramente. Em vez disso, você deverá gerar outro par de chaves de acesso.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e atribuir a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de funções. Um usuário é exclusivamente associado a uma pessoa ou a um aplicativo, mas uma função pode ser assumida por qualquer pessoa que precisar dela. Os usuários têm credenciais permanentes de longo prazo, mas as funções fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de uma função) no Guia do usuário do IAM.

Funções do IAM

UmaIAM role (Função do IAM)é uma identidade dentro do seuAWSConta com permissões específicas. Ela é semelhante a um usuário do IAM, mas não está associada a uma pessoa específica. Você pode assumir temporariamente uma função do IAM noAWS Management Consoleporalternando funções. É possível assumir uma função chamando uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para mais informações sobre métodos para o uso de funções, consulte Usar funções do IAM no Guia do usuário do IAM.

As funções do IAM com credenciais temporárias são úteis nas seguintes situações:

- Permissões temporárias para usuários do IAM: um usuário do IAM pode assumir uma função do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso de usuário federadoEm vez de criar um usuário do IAM, é possível usar identidades existentes doAWS Directory ServiceO, o diretório de usuários da sua empresa ou um provedor de identidades da web. Estes são conhecidos como usuários federados. A AWS atribui uma função a um usuário federado quando o acesso é solicitado por meio de um provedor de identidades. Para

obter mais informações sobre usuários federados, consulte Usuários federados e funções no Guia do usuário do IAM.

- Acesso entre contas: é possível usar uma função do IAM para permitir que alguém (um principal confiável) em outra conta acesse recursos em sua conta. As funções são a principal forma de conceder acesso entre contas. No entanto, alguns serviços da AWS permitem que você anexe uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recurso para acesso entre contas, consulte Como as funções do IAM diferem das políticas baseadas em recurso no Guia do usuário do IAM.
- Acesso entre serviços— AlgunsAWSusam recursos em outrosAWSServiços da . Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando uma função de serviço ou uma função vinculada ao serviço.
  - Permissões principais— Quando você usa um usuário ou uma função do IAM para executar ações noAWS, você é considerado um principal. As políticas concedem permissões a uma entidade principal. Quando você usa alguns serviços, pode executar uma ação que, em seguida, aciona outra ação em outro serviço. Nesse caso, você deve ter permissões para executar ambas as ações. Para ver se uma ação requer ações dependentes adicionais em uma política, consulteAções, recursos e chaves de condição para oAWS App RunnernoReferência de autorização do serviço.
  - Função de serviço: uma função de serviço é uma função do IAM que um serviço assume para realizar ações em seu nome. As funções de serviço fornecem acesso apenas dentro de sua conta e não podem ser usadas para conceder acesso a serviços em outras contas. Um administrador do IAM pode criar, modificar e excluir uma função de serviço do IAM. Para obter mais informações, consulteCriar uma função para delegar permissões a umAWSserviçonoGuia do usuário do IAM.
  - Função vinculada ao serviçoA função vinculada ao serviço é um tipo de função de serviço vinculada a umAWSserviçoServiço O serviço pode assumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.
- Aplicativos em execução no Amazon EC2— é possível usar uma função do IAM para gerenciar credenciais temporárias para aplicativos em execução em uma instância do EC2 que fazem oAWS CLIouAWSSolicitações da API. É preferível fazer isso do que armazenar chaves de acesso na instância do EC2. Para atribuir uma função da AWS a uma instância do EC2 e disponibilizá-la para

todos os seus aplicativos, crie um perfil de instância que esteja anexado à instância. Um perfil de instância contém a função e permite que programas que estão em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Usar uma função do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se deseja usar as funções do IAM, consulte Quando criar uma função do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso noAWSCriando políticas e anexando-as às identidades do IAM ou aoAWSrecursos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou recurso, define suas permissões. Você pode fazer login como o usuário raiz ou um usuário do IAM ou assumir uma função do IAM. Quando você fizer uma solicitação,AWSO avalia as políticas relacionadas baseadas em identidade ou baseadas em recursos do. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas são armazenadas na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar oAWSPolíticas JSON para especificar quem tem acesso a quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

Cada entidade do IAM (usuário ou função) começa sem permissões. Em outras palavras, por padrão, os usuários não podem fazer nada, nem mesmo alterar sua própria senha. Para dar permissão a um usuário para fazer algo, um administrador deve anexar uma política de permissões ao usuário. Ou o administrador pode adicionar o usuário a um grupo que tenha as permissões pretendidas. Quando um administrador concede permissões a um grupo, todos os usuários desse grupo recebem essas permissões.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de funções do AWS Management Console, da AWS CLI ou da API da AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou função do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou função. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e funções em sua conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recurso são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de função do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar um principal em uma política baseada em recursos. Os principais podem incluir contas, usuários, funções, usuários federados ouAWSServiços da .

Políticas baseadas em recursos são políticas em linha que estão localizadas nesse serviço. Você não pode usar oAWSPolíticas gerenciadas do IAM em uma política baseada em recurso.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Amazon S3,AWS WAFO e a Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

A AWS oferece suporte a tipos de política menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou função do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade da entidade e seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou a função no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs)— SCPs são políticas JSON que especificam o máximo de permissões para uma organização ou unidade organizacional (UO) noAWS Organizations.AWS OrganizationsO é um serviço para agrupar e gerenciar centralmente váriosAWSAs contas da que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades em contas-membro, incluindo cadaAWSUsuário raiz da conta da. Para obter mais informações sobre Organizations e SCPs, consulteComo funcionam as SCPsnoAWS OrganizationsGuia do usuário.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para uma função ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou da função e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recurso. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber comoAWSO determina se deve permitir uma solicitação quando vários tipos de política estão envolvidos, consulteLógica da avaliação denoGuia do usuário do IAM.

Como o App Runner funciona com o IAM

Antes de usar o IAM para gerenciar o acesso aoAWS App RunnerVocê deve entender quais recursos do IAM estão disponíveis para uso com o App Runner. Para obter uma visão detalhada de como o App Runner e outrosAWSfuncionam com o IAM, consulteAWSServiços que funcionam com o IAMnoGuia do usuário do IAM.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Tópicos

- Políticas baseadas em identidade do App Runner
- Políticas baseadas em recurso do App Runner
- Autorização baseada em tags do App Runner
- Permissões de usuário do App Run
- Funções do IAM do aplicativo do

Políticas baseadas em identidade do App Runner

Com as políticas baseadas em identidade do API Gateway, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. O App Runner oferece suporte a ações, recursos e chaves de condição específicos. Para conhecer todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Actions

Os administradores podem usar oAWSPolíticas JSON para especificar quem tem acesso a quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Inclua ações em uma política para conceder permissões para executar a operação associada.

As ações de política no App Runner usam o seguinte prefixo antes da ação: `apprunner:`. Por exemplo, conceder permissão a alguém para executar uma instância do Amazon EC2 com o

Amazon EC2`RunInstances`operação da API, você inclui o`ec2:RunInstances`ação em sua política. As instruções de política devem incluir um elemento `Action` ou `NotAction`. O App Runner define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme o seguinte:

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

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação.

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

Para ver uma lista das ações do App Runner, consulteAções definidas peloAWS App RunnernoReferência de autorização do serviço.

Resources

Os administradores podem usar oAWSPolíticas JSON para especificar quem tem acesso a quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando seu Nome de recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Os recursos do App Runner têm a seguinte estrutura ARN:

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

Para obter mais informações sobre o formato de ARNs, consulteNomes de recurso da Amazon (ARNs) eAWSNamespaces de serviços danoAWSReferência geral.

Por exemplo, para especificar omy-serviceEm sua instrução, use o seguinte ARN:

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

Para especificar todos os serviços que pertencem a uma conta específica, use o caractere curinga (*):

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

Algumas ações do App Runner, como as ações para a criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, você deve usar o caractere curinga (*).

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

Para ver uma lista de tipos de recursos do App Runner e seus ARNs, consulteRecursos definidos peloAWS App RunnernoReferência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulteAções definidas peloAWS App Runner.

Chaves de condição

Os administradores podem usar oAWSPolíticas JSON para especificar quem tem acesso a quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco de `Condition`) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam operadores de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único elemento `Condition`, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação

lógica `OR`. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

AWSO oferece suporte a chaves de condição globais e chaves de condição específicas do serviço. Como visualizar todas asAWSChaves de condição globais da, consulteAWSChaves de contexto de condição globais danoGuia do usuário do IAM.

O App Runner oferece suporte ao uso de algumas chaves de condição globais. Como visualizar todas asAWSChaves de condição globais da, consulteAWSChaves de contexto de condição da globaisnoGuia do usuário do IAM.

O App Runner define um conjunto de chaves de condição específicas do serviço. Além disso, o App Runner oferece suporte ao controle de acesso baseado em tags, que é implementado usando chaves de condição. Para obter mais detalhes, consulte the section called “Autorização baseada em tags do App Runner”.

Para ver uma lista das chaves de condição do App Runner, consulteChaves de condição doAWS App RunnernoReferência de autorização do serviço. Para saber com quais ações e recursos que você pode usar uma chave de condição, consulteAções definidas peloAWS App Runner.

Examples

Para visualizar exemplos de políticas baseadas em identidade do App Runner, consulteExemplos de políticas baseadas em identidade do App Runner.

Políticas baseadas em recurso do App Runner

O App Runner não oferece suporte a políticas baseadas em recurso.

Autorização baseada em tags do App Runner

É possível anexar tags aos recursos do App Runner ou transmitir tags em uma solicitação ao App Runner. Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição `apprunner:ResourceTag/`*`key-`*

name, `aws:RequestTag/`*key-name* ou `aws:TagKeys`. Para obter mais informações sobre a marcação de recursos do App Runner, consultethe section called “Configuração”.

Para visualizar um exemplo de política baseada em identidade que visa limitar o acesso a um recurso baseado nas tags desse recurso, consulte Controlar o acesso aos serviços do App Runner com base em tags.

Permissões de usuário do App Run

Para usar o App Runner, os usuários do IAM precisam de permissões para ações do App Runner. Uma maneira comum de conceder permissões aos usuários é anexando uma política a usuários ou grupos do IAM. Para obter mais informações sobre como gerenciar permissões de usuário do, consulteAlterar permissões para um usuário do IAMnoGuia do usuário do IAM.

O App Runner oferece duas políticas gerenciadas da que podem ser anexadas aos usuários do.

- `AWSAppRunnerReadOnlyAccess`- Concede permissões para listar e visualizar detalhes sobre os recursos do App Runner.
- `AWSAppRunnerFullAccess`— Concede permissões a todas as ações do App Runner.

Para um controle mais granular das permissões de usuário, você pode criar uma política personalizada e anexá-la aos usuários. Para obter mais detalhes, consulte .Criando políticas IAMnoGuia do usuário do IAM.

Para obter políticas de usuário do, consultethe section called “Políticas de usuário”.

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": "*"
    }
  ]
```

```
}
```

Funções do IAM do aplicativo do

UmaIAM role (Função do IAM)é uma entidade dentro do seuConta da AWSQue tem permissões específicas.

Funções vinculadas ao serviço

Funções vinculadas ao serviço permitem que os serviços da AWS acessem recursos em outros serviços para concluir uma ação em seu nome. As funções vinculadas ao serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

O App Runner oferece suporte a funções vinculadas ao serviço. Para obter informações sobre como criar ou gerenciar funções vinculadas ao serviço do App Runner, consultethe section called “Uso de funções vinculadas a serviço”.

Funções de serviço

Esse recurso permite que um serviço assuma uma função de serviço em seu nome. A função permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. As funções de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso significa que um administrador do IAM pode alterar as permissões para essa função. Porém, fazer isso pode alterar a funcionalidade do serviço.

App Runner suporta algumas funções de serviço.

Função de acesso ao

A função de acesso é uma função que o App Runner usa para acessar imagens no Amazon Elastic Container Registry (Amazon ECR) em sua conta. É necessário acessar uma imagem no Amazon ECR e não é necessário com o Amazon ECR Public. Antes de criar um serviço baseado em uma imagem no Amazon ECR, use o IAM para criar uma função de serviço e use o`AWSAppRunnerServicePolicyForECRAccess`política gerenciada nele. Você pode então passar essa função para o App Runner ao chamar o métodoCreateServiceAPI do noAuthenticationConfigurationestrutura (um membro doSourceConfiguration) ou ao usar o console do App Runner para criar um serviço.

AWSAppRunnerServicePolicyForECRAccess

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

> **Note**
>
> Se você criar sua própria política personalizada para sua função de acesso, certifique-se de especificar`"Resource": "*"`para `aecr:GetAuthorizationToken`Ação . Os tokens do podem ser usados para acessar qualquer registro do Amazon ECR ao qual você tem acesso.

Ao criar sua função de acesso, certifique-se de adicionar uma política de confiança que declara a entidade de serviço do App Runner`build.apprunner.amazonaws.com`Como uma entidade confiável.

Política de confiança para uma função de acesso

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

```
    }
  ]
}
```

Se você usar o console do App Runner para criar um serviço, o console poderá criar automaticamente uma função de acesso para você e escolhê-la para o novo serviço. O console também lista outras funções em sua conta e você pode selecionar uma função diferente, se quiser.

Função da instância do

A função da instância é uma função opcional que o App Runner usa para fornecer permissões paraAWSações de serviço que o código do aplicativo chama. Antes de criar um serviço App Runner, use o IAM para criar uma função de serviço com as permissões de que seu código de aplicativo precisa. Em seguida, você pode passar essa função para o App Runner na seçãoCreateServiceou ao usar o console do App Runner para criar um serviço.

Ao criar sua função de instância, certifique-se de adicionar uma política de confiança que declara a entidade de serviço do App Runner`tasks.apprunner.amazonaws.com`Como uma entidade confiável.

Política de confiança para uma função de instância

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

Se você usar o console do App Runner para criar um serviço, o console lista as funções em sua conta e você pode selecionar a função que você criou para esse fim.

Para obter mais informações sobre como criar um serviço, consultethe section called “Criação”.

> **Note**
>
> As permissões que a função de instância deve fornecer dependem inteiramente do seu aplicativo. Seu código pode não chamar qualquerAWSe, nesse caso, você não precisa fornecer uma função de instância ao App Runner.
> No caso do seu código chamarAWSAPIs, não temos como antecipar quais chamadas são. Portanto, não fornecemos uma política gerenciada para funções de instância. Você deve incluir explicitamente as permissões necessárias em sua função de instância ou criar sua própria política personalizada e usá-la na função de instância.

Exemplos de políticas baseadas em identidade do App Runner

Por padrão, os usuários e as funções do IAM não têm permissão para criar ou modificarAWS App Runnerrecursos da AWS. Eles também não podem executar tarefas usando o AWS Management Console, a AWS CLI ou uma API da AWS. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e funções permissão para executarem operações de API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Tópicos

- Melhores práticas de políticas
- Políticas de usuário
- Controlar o acesso aos serviços do App Runner com base em tags

Melhores práticas de políticas

As políticas baseadas em identidade são muito eficientes. Elas determinam se alguém pode criar, acessar ou excluir recursos do App Runner em sua conta. Essas ações podem incorrer em custos para sua conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece a usar oAWSPolíticas gerenciadas do— Para começar a usar o App Runner rapidamente, useAWSPolíticas gerenciadas pela para conceder a seus funcionários as permissões de que precisam. Essas políticas já estão disponíveis na conta e são mantidas e atualizadas pela AWS. Para obter mais informações, consulteComece a usar permissões com oAWSPolíticas gerenciadas donoGuia do usuário do IAM.
- Conceder privilégio mínimo: ao criar políticas personalizadas, conceda apenas as permissões necessárias para executar uma tarefa. Comece com um conjunto mínimo de permissões e conceda permissões adicionais conforme necessário. Fazer isso é mais seguro do que começar com permissões que são muito lenientes e tentar restringi-las posteriormente. Para obter mais informações, consulte Conceder privilégio mínimo no Guia do usuário do IAM.
- Habilitar MFA para operações confidenciais: para aumentar a segurança, exija que os usuários do IAM usem Multi-Factor Authentication (MFA) para acessar recursos ou operações de API confidenciais. Para obter mais informações, consulteUsar autenticação multifator (MFA) noAWSnoGuia do usuário do IAM.
- Usar condições de política para segurança adicional: na medida do possível, defina as condições sob as quais suas políticas baseadas em identidade permitem o acesso a um recurso. Por exemplo, você pode gravar condições para especificar um intervalo de endereços IP permitidos do qual a solicitação deve partir. Você também pode escrever condições para permitir somente solicitações em uma data especificada ou período ou para exigir o uso de SSL ou MFA. Para obter mais informações, consulteElementos de política JSON do: CondiçãonoGuia do usuário do IAM.

Políticas de usuário

Para acessar o console do App Runner, os usuários do IAM devem ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes dos recursos do App Runner noConta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para os usuários com essa política do.

O App Runner oferece duas políticas gerenciadas da que podem ser anexadas aos usuários do.

- `AWSAppRunnerReadOnlyAccess`- Concede permissões para listar e visualizar detalhes sobre os recursos do App Runner.
- `AWSAppRunnerFullAccess`— Concede permissões a todas as ações do App Runner.

Para garantir que os usuários possam usar o console do App Runner, conecte, no mínimo, o`AWSAppRunnerReadOnlyAccess`Gerenciamento de políticas para os usuários da. Você pode anexar o`AWSAppRunnerFullAccess`ou adicione permissões adicionais específicas para permitir que os usuários criem, modifiquem e excluam recursos. Para obter mais informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM:

Não é necessário conceder permissões mínimas do console para os usuários que estão fazendo ligações somente com a AWS CLI ou a API do AWS. Em vez disso, permita o acesso somente às ações que correspondem à operação da API que você deseja permitir que os usuários executem.

Os exemplos a seguir demonstram políticas de usuário personalizadas. Você pode usá-los como pontos de partida para definir suas próprias políticas de usuário personalizadas. Copie o exemplo, e/ou remova ações, faça o escopo de recursos e adicione condições.

Exemplo: política de usuário de gerenciamento de console e conexão

Este exemplo de política permite o acesso ao console e permite a criação e o gerenciamento de conexões. Ele não permite a criação e o gerenciamento do serviço App Runner. Ele pode ser anexado a um usuário cuja função é gerenciar o acesso ao serviço do App Runner aos ativos do código-fonte.

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

Exemplo: políticas de usuário que usam chaves de condição

Os exemplos nesta seção demonstram permissões condicionais que dependem de algumas propriedades de recurso ou parâmetros de ação.

Esta política de exemplo permite a criação de um serviço App Runner, mas nega o uso de uma conexão chamada`prod`.

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

Este exemplo de política permite a atualização de um serviço App Runner chamado`preprod`somente com uma configuração de auto scaling chamada`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/*"
        }
      }
    }
  ]
}
```

Controlar o acesso aos serviços do App Runner com base em tags

Você pode usar condições em sua política baseada em identidade para controlar o acesso aos recursos do App Runner baseados em tags. Este exemplo mostra como você pode criar uma política que permite excluir um serviço do App Runner. No entanto, a permissão será concedida somente se a tag `Owner` tiver o valor do nome desse usuário. Essa política também concede as permissões necessárias concluir essa ação no 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}"}
      }
    }
  ]
}
```

É possível anexar essa política aos usuários do IAM na sua conta. Se um usuário chamado`richard-roe`O serviço do App Runner tenta excluir um serviço do App Runner, o serviço deve ser marcado`Owner=richard-roe`ou`owner=richard-roe`. Caso contrário, ele terá o acesso negado. A chave da tag de condição `Owner` corresponde a `Owner` e a `owner` porque os nomes de chaves de condição não fazem distinção entre maiúsculas e minúsculas. Para obter mais informações, consulteElementos de política JSON do: CondiçãonoGuia do usuário do IAM.

Usar funções vinculadas ao serviço para o App Runner

AWS App RunnerO usaAWS Identity and Access Management(IAM)Funções vinculadas ao serviço. A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao App

Runner. As funções vinculadas a serviços são predefinidas pelo App Runner e incluem todas as permissões que o serviço requer para chamar outrosAWSServiços da em seu nome.

Uma função vinculada ao serviço facilita a configuração do App Runner, já que não é preciso adicionar as permissões necessárias manualmente. O App Runner define as permissões das funções vinculadas ao serviço e, exceto se definido em contrário, somente o App Runner pode assumir suas funções. As permissões definidas incluem as políticas de confiança e de permissões, e essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Uma função vinculada ao serviço poderá ser excluída somente após excluir seus recursos relacionados. Isso protege seus recursos do App Runner, pois você não pode remover por engano as permissões para acessar os recursos.

Para obter informações sobre outros serviços que oferecem suporte às funções vinculadas a serviço, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que apresentam Sim na coluna Função vinculada a serviços. Escolha um Sim com um link para exibir a documentação da função vinculada a serviço desse serviço.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Permissões de função vinculada ao serviço para o App Runner

O App Runner usa a função vinculada ao serviço chamadaAWSServiceRoleForAppRunner.

A função permite que o App Runner execute as seguintes tarefas:

- Push logs para grupos de log do Amazon CloudWatch Logs.
- Crie regras de Amazon CloudWatch Events para assinar os push de imagem do Amazon Elastic Container Registry (Amazon ECR).

A função vinculada ao serviço AWSServiceRoleForAPPRUNNER confia nos seguintes serviços para assumir a função:

- `apprunner.amazonaws.com`

A política de permissões da função vinculada ao serviço AWSServiceRoleForAPPRUNNER contém todas as permissões de que o App Runner precisa para concluir ações em seu nome.

AppRunnerServiceRolePolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:PutRetentionPolicy"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:*:*:log-group:/aws/apprunner/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/apprunner/*:log-stream:*"
      ]
    },
    {
      "Sid": "AllowPutRuleForManagedRules",
      "Effect": "Allow",
      "Action": [
        "events: PutRule",
        "events: PutTargets",
        "events: DeleteRule",
        "events: RemoveTargets",
        "events: DisableRule",
        "events: EnableRule"
      ],
      "Resource": "arn:aws:events:*:*:rule/apprunner-*",
      "Condition": {
        "StringEquals": {
          "events:ManagedBy": "apprunner.amazonaws.com",
          "events:source": "aws.ecr",
          "events:detail-type": "ECR Image Action"
        }
      }
    }
```

```
  ]
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Para obter mais informações, consulte Permissões de função vinculada ao serviço no Guia do usuário do IAM.

Criar uma função vinculada ao serviço para o App Runner

Você não precisa criar manualmente uma função vinculada ao serviço. Quando você cria um serviço App Runner noAWS Management Console, oAWS CLI, ou oAWSO App Runner cria uma função vinculada ao serviço para você.

Se você excluir essa função vinculada ao serviço e precisar criá-la novamente, poderá usar esse mesmo processo para recriar a função em sua conta. Quando você cria um serviço App Runner, o App Runner cria a função vinculada ao serviço novamente por você.

Editar uma função vinculada ao serviço para o App Runner

O App Runner não permite que você edite a função vinculada ao serviço AWSServiceRoleForAppRunner. Depois que criar uma função vinculada ao serviço, você não poderá alterar o nome da função, pois várias entidades podem fazer referência a ela. No entanto, você poderá editar a descrição da função usando o IAM. Para obter mais informações, consulte Editar uma função vinculada ao serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço para o App Runner

Se você não precisar mais usar um recurso ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente.

No App Runner, isso significa excluir todos os serviços do App Runner em sua conta da. Para saber mais sobre como excluir serviços do App Runner, consultethe section called “Exclusão”.

> **Note**
>
> Se o serviço App Runner estiver usando a função quando tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Como excluir manualmente a função vinculada ao serviço usando o IAM

Use o console do IAM, oAWS CLI, ou oAWSPara excluir a função vinculada ao serviço AWSServiceRoleForAppRunner. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Regiões compatíveis com funções vinculadas ao serviço do App Runner

O App Runner é compatível com funções vinculadas a serviços em todas as regiões em que o serviço está disponível. Para obter mais informações, consulteAWS App RunnerEndpoints e cotas donoAWSReferência geral.

Políticas gerenciadas pela AWS para o AWS App Runner

Para adicionar permissões a usuários, grupos e funções do, é mais fácil usar oAWSdiretivas gerenciadas do que escrever políticas você mesmo. É necessário tempo e experiência para criar políticas gerenciadas pelo cliente do IAM que fornecem à sua equipe apenas as permissões de que precisam. Para começar a usar rapidamente, use oAWSPolíticas gerenciadas do. Essas políticas cobrem casos de uso comuns e estão disponíveis em seuAWSconta. Para obter mais informações sobreAWSPolíticas gerenciadas do, consulteAWSPolíticas gerenciadas donoGuia do usuário do IAM.

AWSmanutenção e atualização de serviçosAWSPolíticas gerenciadas do. Você não pode alterar as permissões do noAWSPolíticas gerenciadas do. Os serviços ocasionalmente adicionam permissões adicionais a umAWSpara oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. Os serviços são mais propensos a atualizar umAWSquando um novo recurso é iniciado ou quando novas operações ficam disponíveis. Os serviços não removem permissões de umAWSpara que as atualizações de políticas não quebrem suas permissões existentes.

Além disso,AWSO oferece suporte a políticas gerenciadas pela para funções de trabalho que abrangem vários serviços Por exemplo, as receitasReadOnlyAccess AWSA política gerenciada fornece acesso somente leitura a todos osAWSServiços e recursos da. Quando um serviço inicia um novo recurso, oAWSadiciona permissões somente leitura para novas operações e recursos. Para obter uma lista e descrições de políticas de função de trabalho, consulteAWSPolíticas gerenciadas pela para funções de trabalhonoGuia do usuário do IAM.

Atualizações do App Runner paraAWSPolíticas gerenciadas do

Ver detalhes sobre atualizações doAWSpara o App Runner desde que este serviço começou a acompanhar essas alterações. Para obter alertas automáticos sobre alterações a esta página, assine o feed RSS na página de histórico de documentos do App Runner.

Alteração	Descrição	Data
AppRunnerServiceRolePolicyNova política	O App Runner adicionou uma nova política para permitir que o App Runner faça chamadas para o Amazon CloudWatch Logs e Amazon CloudWatch Events em nome dos serviços do App Runner.	1.º de março de 2021
awsApprunnerReadOnlyAccessNova política	O App Runner adicionou uma nova política para permitir que os usuários listem e visualizem detalhes sobre os recursos do App Runner.	1.º de março de 2021
awsApprunnerFullAccessNova política	O App Runner adicionou uma nova política para permitir que os usuários executem todas as ações do App Runner.	1.º de março de 2021
awsApprunnerServicePolicyForecrAccessNova política	O App Runner adicionou uma nova política para permitir que o App Runner acesse imagens do Amazon Elastic Container Registry (Amazon ECR) em sua conta.	1.º de março de 2021
App Runner começou a acompanhar alterações	O App Runner começou a acompanhar as alterações para o seuAWSPolíticas gerenciadas do.	1.º de março de 2021

Solução de problemas de identidade e acesso

Use as seguintes informações para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com oAWS App Runnere IAM.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Tópicos

- Não tenho autorização para executar uma ação no App Runner
- Sou administrador e desejo permitir que outros usuários tenham acesso ao App Runner
- Quero permitir que as pessoas fora da minhaConta da AWSpara acessar meus recursos do App Runner

Não tenho autorização para executar uma ação no App Runner

Se o AWS Management Console informar que você não está autorizado a executar uma ação, entre em contato com o administrador para obter assistência. O administrador é a pessoa que forneceu a você o seuAWSNome de usuário e senha do.

O erro de exemplo a seguir ocorre quando um usuário do IAM chamado`marymajor`O tenta usar o console do para visualizar detalhes sobre um serviço do App Runner, mas não tem o`apprunner:DescribeService`Permissões

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

Neste caso, Mary pede ao administrador para atualizar suas políticas para permitir que ela acesse o*`my-example-service`*recurso usando o`apprunner:DescribeService`Ação .

Sou administrador e desejo permitir que outros usuários tenham acesso ao App Runner

Para permitir que outros usuários acessem o App Runner, é necessário criar uma entidade do IAM (usuário ou função) para a pessoa ou a aplicação que precisa do acesso. Eles usarão as credenciais dessa entidade para acessar a AWS. Você deve anexar uma política à entidade que concede a eles as permissões corretas no App Runner.

Para começar a usar imediatamente, consulte Criar os primeiros usuário e grupo delegados do IAM no Guia do usuário do IAM.

Quero permitir que as pessoas fora da minhaConta da AWSpara acessar meus recursos do App Runner

Você pode criar uma função que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir a função. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte o seguinte:

- Para saber se o App Runner oferece suporte a esses recursos, consulteComo o App Runner funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos emAWScontas que você possui, consulteFornecer acesso a um usuário do IAM em outroAWSConta própriaanoGuia do usuário do IAM.
- Para saber como fornecer acesso aos recursos do a terceirosAWScontas, consulteFornecer acesso aoAWSContas de propriedade de terceirosanoGuia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Como as funções do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Registro em log e monitoramento no App Runner

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e o desempenho doAWS App RunnerserviçoServiço. Coletando dados de monitoramento de todas as partes doAWSA solução permite que você depure mais facilmente uma falha se ocorrer uma. App Runner integra-se com váriosAWSpara monitorar seus serviços do App Runner e responder a incidentes potenciais.

Alarmes do Amazon CloudWatch

Com alarmes do Amazon CloudWatch, você pode observar uma métrica de serviçoServiço durante um período especificado. Se a métrica ultrapassar um limite especificado para um número especificado de períodos, você receberá uma notificação.

O App Runner coleta uma variedade de métricas sobre o serviço como um todo e as instâncias (unidades de escala) que executam seu serviço Web. Para mais informações, consulte Métricas (CloudWatch).

Logs de aplicativos

O App Runner coleta a saída do código do aplicativo e o transmite para o Amazon CloudWatch Logs. O que está nessa saída é com você. Por exemplo, você pode incluir registros detalhados de solicitações feitas ao seu serviço Web. Esses registros de log podem ser úteis em auditorias de segurança e acesso. Para mais informações, consulte Logs (CloudWatch Logs).

AWS CloudTrailAction Logs

O App Runner está integrado aoAWS CloudTrail, um serviço que fornece um registro de ações executadas por um usuário, uma função ou umAWSno App Runner. O CloudTrail captura todas as chamadas à API do App Runner como eventos. Você pode exibir os eventos mais recentes no console do CloudTrail e criar uma trilha para habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon Simple Storage Service (Amazon S3). Para mais informações, consulte Ações de API (CloudTrail).

Validação de conformidade para App Runner

Auditores de terceiros avaliam a segurança e a conformidade do AWS App Runner como parte de vários programas de conformidade da AWS. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para saber se o App Runner ou outroAWSestão no escopo de programas de conformidade específicos, consulteAWSServiços no escopo pelo programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

Você pode fazer download de relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Fazer download de relatórios no AWS Artifact.

Sua responsabilidade com relação à conformidade ao usar os serviços da AWS é determinada pela confidencialidade dos dados, pelos objetivos de conformidade da empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- Guias de início rápido de segurança e conformidadeesses guias de implantação abordam as considerações de arquitetura e fornecem etapas para a implantação de ambientes de linha de base noAWSque são focados em segurança e conformidade.
- Whitepaper de arquitetura for HIPAA Security and Compliance— Este whitepaper descreve como as empresas podem usarAWSPara criar aplicativos compatíveis com HIPAA.

> **Note**
>
> Nem todos os serviços estão em conformidade com a HIPAA.

- AWSRecursos de conformidade— esta coleção de manuais e guias pode ser aplicada ao seu setor e local.
- Avaliar recursos com regrasnoAWS ConfigGuia do desenvolvedor— OAWS Configavalia até que ponto suas configurações de recursos estão em conformidade com práticas internas, diretrizes e regulamentações do setor.
- AWS Security Hub— EsteAWSfornece uma visão abrangente do seu estado de segurança naAWSIsso ajuda você a verificar a sua conformidade com os padrões e as melhores práticas do setor de segurança.
- AWS Audit Manager— EsteAWSajuda-o a auditar continuamente oAWSpara simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Resiliência no App Runner

OAWSA infraestrutura global da é criada com baseRegiões da AWSe Zonas de disponibilidade.Regiões da AWSAs fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, altas taxas de transferência e redes altamente redundantes. Com as zonas de disponibilidade, você pode projetar e operar aplicativos e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobreRegiões da AWSe Zonas de disponibilidade, consulteAWSInfraestrutura global.

O AWS App Runner gerencia e automatiza o uso da infraestrutura global da AWS em seu nome. Ao usar o App Runner, você se beneficia dos mecanismos de disponibilidade e tolerância a falhas oferecidos pelaAWSOfertas.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Segurança da infraestrutura no AWS App Runner

Como um serviço gerenciado, oAWS App Runneré protegido peloAWSProcedimentos de segurança de rede global da descritos noAmazon Web Services: Visão geral dos processos de segurança doWhitepaper.

Você pode usar oAWSChamadas de API publicadas pela para acessar o App Runner pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos TLS 1.2 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Análise de configuração e vulnerabilidade no App Runner

A AWS e nossos clientes compartilham responsabilidade para atingir um alto nível de segurança e conformidade do componente de software. Para obter mais informações, consulte o modelo de responsabilidade compartilhada da AWS.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Melhores práticas de segurança do App Runner

O AWS App Runner fornece vários recursos de segurança a serem considerados no desenvolvimento e na implementação das suas próprias políticas de segurança. As melhores

práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes no seu ambiente, trate-as como considerações úteis, não como requisitos.

Para outros tópicos de segurança do App Runner, consulteSegurança.

Práticas recomendadas de segurança preventiva

Os controles de segurança preventiva tentam evitar incidentes antes que ocorram.

Implemente o privilégio de acesso mínimo

O App Runner forneceAWS Identity and Access ManagementPolíticas gerenciadas do (IAM) paraUsuários do IAMO e aFunção de acesso ao. Essas políticas gerenciadas especificam todas as permissões que podem ser necessárias para a operação correta do serviço do Executor de aplicativos.

O aplicativo pode não exigir todas as permissões em nossas políticas gerenciadas. É possível personalizá-los e conceder apenas as permissões necessárias para que os usuários e o serviço do Executor de aplicativos executem suas tarefas. Isso é especialmente relevante para políticas de usuário, em que diferentes funções de usuário podem ter diferentes necessidades de permissão. A implementação do privilégio de acesso mínimo é fundamental para reduzir o risco de segurança e o impacto que pode resultar de erros ou usuários mal-intencionados.

Práticas recomendadas de segurança de detecção

Os controles de segurança de detecção identificam violações de segurança depois de ocorrerem. Eles podem ajudar a detectar uma possível ameaça ou incidente de segurança.

Implementar o monitoramento

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e o desempenho das suas soluções do App Runner.AWSO fornece várias ferramentas e serviços para ajudar você a monitorar seuAWSServiços da .

Veja a seguir alguns exemplos de itens a serem monitorados:

- Métricas do Amazon CloudWatch para App Runner- defina alarmes para as principais métricas do App Runner e para as métricas personalizadas da sua aplicação. Para obter mais detalhes, consulte Métricas (CloudWatch).

- AWS CloudTrailentries— acompanhe as ações que podem afetar a disponibilidade, como`PauseService`ou`DeleteConnection`. Para obter mais detalhes, consulte Ações de API (CloudTrail).

AWSGlossário

Para a mais recenteAWSterminologia, consulte aAWSGlossárionoAWSReferência geral.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.