



Migração da Aplicação VillasBrands do heroku para aws

Esse tutorial é para migração de uma Aplicacao Rails para AWS *

Passos:

Criar uma conta na aws <https://aws.amazon.com/> ,forneca informações pessoais ,informações de pagamento e verificação de identidade(authenticação multifator).

Configurar IAM (Identity and Access Management)

1. No Console de Gerenciamento da AWS, clique em "Services" (Serviços) e, em seguida, em "IAM" na seção "Security, Identity, & Compliance" (Segurança, Identidade e Conformidade).
2. No painel de navegação à esquerda, clique em "Users" (Usuários) e depois em "Add user" (Adicionar usuário).
3. Preencha o nome do usuário e selecione o tipo de acesso:
 - **Access type:** Selecione "Programmatic access" para permitir o acesso por meio da API da AWS.
 - **Console access:** Selecione "AWS Management Console access" se o usuário precisar de acesso ao console da AWS.

Avance para as permissões e adicione o usuário a um grupo existente ou atribua políticas diretamente. Recomenda-se seguir o princípio do menor privilégio, dando ao usuário apenas as permissões necessárias

Vamos criar um grupo para organizar permissões:

- Clique em "Add user to group" (Adicionar usuário a um grupo).
- Crie um novo grupo (por exemplo, "EC2_RDS_Group").
- Atribua políticas conforme necessário

1. AmazonEC2FullAccess: Para gerenciar instâncias EC2.

2. **AmazonRDSFullAccess:** Para gerenciar bancos de dados RDS.
3. **AmazonVPCFullAccess:** Para gerenciar VPCs e configurações de rede.
4. **AmazonRDSReadOnlyAccess:** Se o usuário só precisa de acesso de leitura ao banco de dados RDS.
5. **AmazonS3FullAccess:** Para acessar e gerenciar buckets do Amazon S3, se aplicável.
6. **AmazonRDSDataFullAccess:** Se o usuário precisa executar consultas e gerenciar dados em bancos de dados RDS.
7. Passe pelas configurações adicionais, como tags, se desejar.
8. Revise as configurações e clique em "Create user" (Criar usuário).
9. Após a criação, será exibida uma mensagem com as informações de acesso do usuário, incluindo a chave de acesso e a chave secreta. Certifique-se de armazenar essas informações com segurança

Configurar AWS CloudTrail

1. No Console de Gerenciamento da AWS, vá para "Services" (Serviços) e selecione "CloudTrail" sob a seção "Management & Governance" (Gerenciamento e Governança).
2. Clique em "Create Trail" (Criar Rastro).
3. Preencha os detalhes do rastro:
 - **Trail Name:** Escolha um nome descritivo para o rastro.
 - **Apply trail to all regions:** Selecione se deseja que o rastro registre eventos de todas as regiões.
 - **Storage Location:** Escolha um bucket do Amazon S3 para armazenar os registros.
4. Configure as opções adicionais, como a criação de logs de dados do Amazon S3 e a configuração de notificações.
5. Clique em "Create" (Criar) para criar o rastro.
6. Após a criação, o CloudTrail começará a registrar eventos na sua conta.

Configurar Notificações de Log do CloudTrail (Opcional)

1. No Console de Gerenciamento da AWS, vá para "Services" (Serviços) e selecione "CloudWatch" sob a seção "Management & Governance".
2. No painel de navegação à esquerda, clique em "Rules" (Regras) sob "EventBridge".
3. Clique em "Create rule" (Criar regra).
4. Configure as condições da regra, como o serviço (CloudTrail) e as condições específicas.
5. Configure a ação da regra, como enviar uma mensagem para um tópico do Amazon SNS.
6. Clique em "Create" (Criar) para criar a regra.

O AWS CloudTrail ajudará na auditoria de atividades na conta, fornecendo informações detalhadas sobre quem fez o quê, quando e onde. Esses logs são valiosos para fins de segurança, conformidade e solução de problemas.

Instalação e Configuração da AWS CLI

Linux: `sudo apt-get install awscli`

Depois de instalar a AWS CLI, configure as credenciais e a região padrão. Abra o terminal ou prompt de comando.

1. Execute o comando: **aws configure**

Forneça as informações solicitadas:

- **AWS Access Key ID:** Sua Access Key ID.
- **AWS Secret Access Key:** Sua Secret Access Key.
- **Default region name:** "US East (N. Virginia)" ou "us-east-1"

Após configurar as credenciais, você pode verificar se a configuração foi bem-sucedida usando o seguinte comando: **aws sts get-caller-identity**

Configurar Região da aws "US East (N. Virginia)" ou "us-east-1"

- **Selecione a Região:** No canto superior direito do console, terá uma lista suspensa que mostra a região atualmente selecionada. Clique nessa lista suspensa.
- **Escolha a Região:** Tera uma lista de todas as regiões da AWS disponíveis. Clique "US East (N. Virginia)" ou "us-east-1"

Configurar VPC e Subnetes:

Criar VPC:

1. No Console de Gerenciamento da AWS, vá para o serviço VPC.
2. No painel de navegação à esquerda, clique em "Your VPCs" (Suas VPCs).
3. Clique no botão "Create VPC" (Criar VPC).
4. Preencha os detalhes da VPC, incluindo o nome, o bloco de endereços IP (CIDR) da VPC e outras configurações, como tags.
5. Clique em "Create VPC" para criar a VPC.

Criar Subnets:

1. Com a VPC criada, clique em "Subnets" no painel de navegação à esquerda.
2. Clique no botão "Create Subnet" (Criar Subnet).

3. Preencha os detalhes da subnet, incluindo o nome, a VPC associada, o bloco de endereços IP da subnet, a disponibilidade da zona (Availability Zone), etc.
4. Clique em "Create" (Criar) para criar a subnet.
5. Repita o processo para criar subnets adicionais conforme necessário.

Criar Grupo de Segurança para Instâncias EC2 (WebAppSecurityGroup):

1. No Console de Gerenciamento da AWS, vá para o serviço EC2.
2. No painel de navegação à esquerda, clique em "Security Groups" (Grupos de Segurança).
3. Clique no botão "Create Security Group" (Criar Grupo de Segurança).
4. Preencha os detalhes do grupo de segurança:
 - Nome do grupo de segurança: WebAppSecurityGroup
 - Descrição: Grupo de Segurança para o Ambiente da Aplicação Web

Regras de Entrada para WebAppSecurityGroup:

1. HTTP (Servidor Web - Nginx, Apache, etc.):
 - Tipo: HTTP
 - Porta: 80
 - Origem: Selecione "Custom" e especifique o IP ou a faixa de IPs desejados.
2. TCP (Servidor de Aplicativos):
 - Tipo: TCP
 - Porta: A porta específica do aplicativo (por exemplo, 3000 se estiver usando um servidor de aplicativos personalizado).
 - Origem: Selecione "Custom" e especifique o IP ou a faixa de IPs desejados.
3. PostgreSQL (Banco de Dados):
 - Tipo: PostgreSQL
 - Porta: 5432 (ou a porta configurada para o PostgreSQL)
 - Origem: IP específico ou VPC (restringir ao mínimo necessário).
4. SSH (Acesso Remoto ao Servidor):
 - Tipo: SSH
 - Porta: 22
 - Origem: Seu IP ou uma faixa de IPs específica para acesso SSH.
5. TCP (Outras Dependências - por exemplo, Redis):
 - Tipo: TCP
 - Porta: A porta específica que a aplicação utiliza (por exemplo, 6379 para o Redis).
 - Origem: IP da instância do Redis ou VPC

Migração do Banco de Dados PostgreSQL do Heroku para o Amazon RDS

1. Criar uma Instância RDS na AWS:

- No Console da AWS, vá para o serviço RDS e clique em "Create database."
- Escolha o tipo de banco de dados desejado (por exemplo, PostgreSQL), selecione a opção "Amazon Aurora with PostgreSQL compatibility" e siga as instruções para criar uma nova instância RDS.(db.t3.medium 4gb)

2. Exportar Dados do Heroku Postgres:

Use a ferramenta **pg_dump** para exportar dados do banco de dados PostgreSQL no Heroku:

```
pg_dump --format=plain -h HOST_HERE -U USER_HERE -p PORT_HERE -W -d  
DB_NAME_HERE > heroku_db.sql
```

- **--format=plain**: Especifica o formato do arquivo de saída como SQL (texto simples).
- **-h**: Nome do host do banco de dados (pode ser encontrado nas configurações do Heroku).
- **-U**: Nome do usuário do banco de dados (pode ser encontrado nas configurações do Heroku).
- **-p**: Número da porta do banco de dados (pode ser encontrado nas configurações do Heroku).
- **-W**: Solicita a senha ao executar o comando.
- **-d**: Nome do banco de dados que você deseja exportar.

Importar Dados para o Amazon RDS:

- Use a ferramenta **pg_restore** para carregar o dump no Amazon RDS PostgreSQL:
- **pg_restore --verbose --clean --no-acl --no-owner -h YOUR_RDS_ENDPOINT -U YOUR_RDS_USERNAME -d YOUR_RDS_DB_NAME -W heroku_db.sql**
 - verbose**: Exibe informações detalhadas durante a restauração.
 - clean**: Remove os objetos existentes no banco de dados de destino antes da restauração.
 - no-acl**: Não restaura as listas de controle de acesso (ACLs).
 - no-owner**: Não restaura os proprietários originais dos objetos.

- **-h:** O endpoint do seu banco de dados RDS (pode ser encontrado nas configurações do RDS).
- **-U:** O nome de usuário do seu banco de dados RDS (pode ser encontrado nas configurações do RDS).
- **-d:** O nome do banco de dados no RDS.
- **-W:** Solicita a senha ao executar o comando.

Nota Importante:

Certifique-se de substituir os marcadores de posição como **HOST_HERE**, **USER_HERE**, **PORT_HERE**, **DB_NAME_HERE**, **YOUR_RDS_ENDPOINT**, **YOUR_RDS_USERNAME** e **YOUR_RDS_DB_NAME** pelos valores específicos do seu ambiente.

Além disso, é importante que as versões do PostgreSQL no Heroku e no Amazon RDS sejam compatíveis. Faça verificações para garantir que não haja diferenças significativas entre as versões.

Conectividade:

1. Configurar as regras de segurança no Grupo de Segurança da VPC (VPC Security Group):
 - No Console da AWS, vá para o serviço RDS e selecione sua instância.
 - Na guia "Connectivity & security," clique no nome do grupo de segurança associado à sua instância.
 - Certifique-se de que as regras de entrada permitam tráfego de sua instância EC2 para o banco de dados RDS.

Type	Protocol	Port Range	Source
PostgreSQL	TCP	5432	0.0.0.0/0

Backups Automáticos:

1. Configurar backups automáticos:
 - No Console da AWS, vá para o serviço RDS e selecione sua instância.
 - Na guia "Backup," ajuste as configurações de backup automático, incluindo a "Retention period" (Período de Retenção).

Exemplo:

- Marque a opção "Enable automatic backups."
- Defina o "Retention period" para o número desejado de dias

Configurar o Amazon S3

1. Criar um Bucket S3:

No Console da AWS, vá para o serviço S3.

- Clique em "Create bucket" (Criar bucket).
- Preencha os detalhes, como o nome do bucket e a região.
- Clique em "Create" (Criar).

2. Configurar Permissões no Bucket:

- Vá até a guia "Permissions" (Permissões) do seu bucket.
- Certifique-se de configurar as permissões corretas, como políticas de cubo ou políticas de acesso a objetos, dependendo das suas necessidades de segurança.

*****Configurar o Active Storage para Usar o S3*****

Configurar volumes de armazenamento necessários EBS(Elastic Block Store)para as EC2

Navegue para as Instâncias EC2:

- No Console da AWS, vá para o serviço "EC2" para acessar as instâncias EC2.

Selecione a Instância EC2 Desejada:

- Selecione a instância EC2 à qual deseja adicionar volumes EBS.

Adicione um Novo Volume EBS:

- Com a instância EC2 selecionada, na parte inferior da tela, role até a seção "Block devices" (Dispositivos de Bloqueio) e clique em "Add new volume" (Adicionar novo volume).

Configure o Tamanho e o Tipo do Volume EBS:

- Configure o tamanho do volume EBS de acordo com as necessidades da aplicação. Selecione o tipo de volume (por exemplo, SSD ou HDD) com base no desempenho necessário.

Opcionalmente, Adicione Tags:

- Pode se adicionar tags para identificar o volume, o que pode ser útil para fins de organização.

Adicionar um Novo Volume EBS:

- Com a instância EC2 selecionada, na parte inferior da tela, role até a seção "Block devices" (Dispositivos de Bloqueio) e clique em "Add new volume" (Adicionar novo volume).

Configurar o Tamanho e o Tipo do Volume EBS:

- Configure o tamanho do volume EBS de acordo com as necessidades da aplicação
- tipo de volume(SSD) tamanho do volume

Opcionalmente, Adicione Tags:

- Você pode adicionar tags para identificar o volume, o que pode ser útil para fins de organização.

Confirme a Adição do Volume:

- Clique em "Add" (Adicionar) para confirmar a adição do volume EBS à instância EC2.

Montar o Volume na Instância:

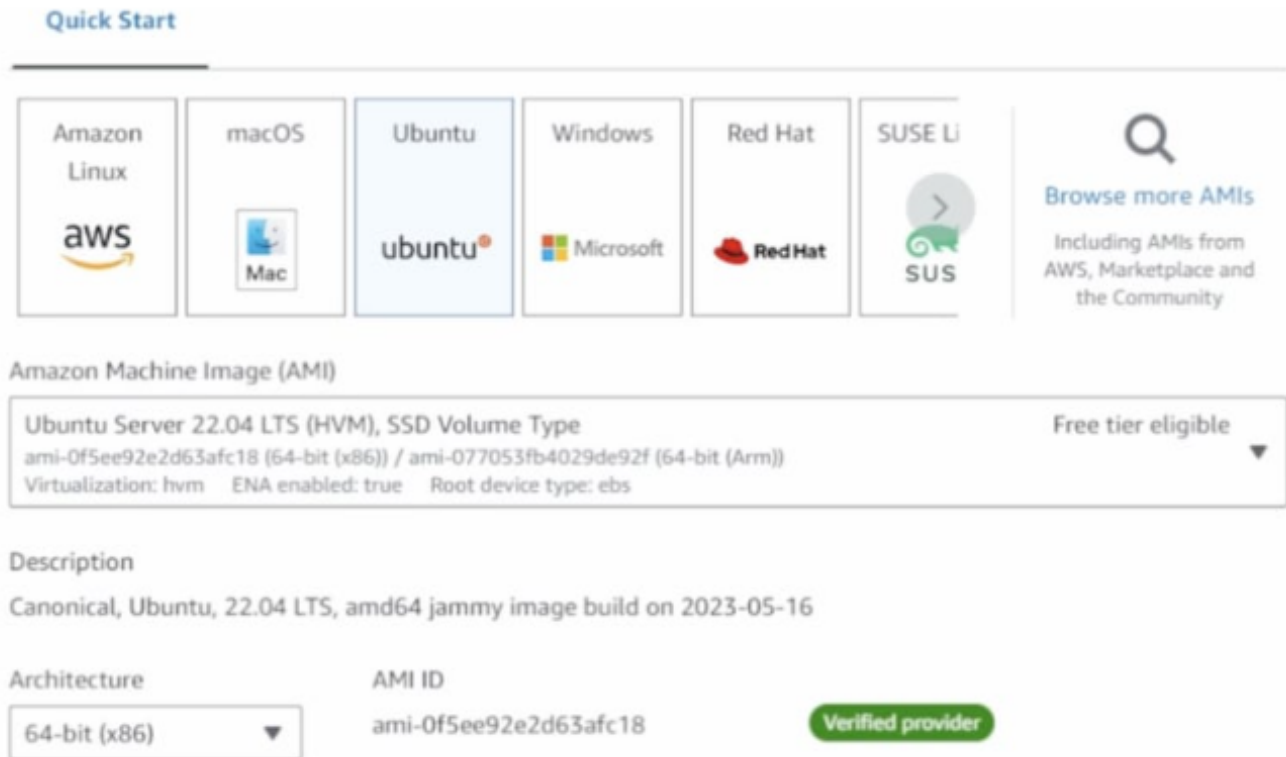
- Conecte-se à instância EC2 usando SSH ou qualquer método de acesso que preferir.
- Use comandos Linux para listar os dispositivos disponíveis e identificar o dispositivo EBS recém-adicionado, como **/dev/xvdf**.
- Crie um sistema de arquivos no volume EBS usando um comando como **sudo mkfs -t ext4 /dev/xvdf**.
- Crie um diretório de montagem, por exemplo, **sudo mkdir /mnt/mydata**.
- Monte o volume EBS no diretório de montagem com um comando como **sudo mount /dev/xvdf /mnt/mydata**.

Configurar a Montagem Automática na Inicialização:

- Abra o arquivo **/etc/fstab** em um editor de texto com privilégios de administrador, como **sudo nano /etc/fstab**.
- Adicione uma linha que especifique o dispositivo EBS, o ponto de montagem e as opções de montagem. Por exemplo:
- **/dev/xvdf /mnt/mydata ext4 defaults,nofail 0 2**
- Isso garantirá que o volume EBS seja montado automaticamente na inicialização da instância EC2.
- .

Lancar uma instancia EC2

- Essa etapa cria um servidor EC2
- No painel de navegação escolha Instances e em seguida launch Instance
- Escolha a AMI(Amazon Machine Image) na guia Quick Start



Selecione o tipo de instancia EC2 com base na necessidade da aplicação 2
instancias **t4g.micro 1 GB**

Adicionar Auto Scaling:

- No Console da AWS, vá para o serviço "Auto Scaling".
- Crie um Novo Grupo de Auto Scaling
- Na página inicial do serviço Auto Scaling, clique em "Grupos de Auto Scaling" no painel de navegação à esquerda.
- Clique no botão "Create Auto Scaling group" (Criar grupo de Auto Scaling).

Configure o Grupo de Auto Scaling

Na tela de configuração do grupo de Auto Scaling, siga as instruções para configurar o grupo.

- **Nome do Grupo:** Dê um nome descritivo ao grupo.
- **Configurações de Lançamento:** Escolha a AMI (Amazon Machine Image) que deseja usar e a instância EC2 que foi criado anteriormente.
- **Adicionar Load Balancer:** vá para o serviço "Elastic Load Balancing".Clique no botão "Create Load Balancer" (Criar Load Balancer).

Configurar o Amazon Route 53 para o Domínio Principal e Subdomínios

Para o Domínio Principal (www.villas.digital):

1. Acesse o Console da AWS e vá para o serviço Amazon Route 53.
2. No painel de navegação à esquerda, clique em "Zonas Hospedadas".
3. Selecione a zona hospedada correspondente ao domínio principal "villas.digital".
4. Crie um registro de Alias (CNAME) para o domínio principal:
 - Clique em "Criar Registro" no canto superior direito.
 - Escolha o tipo de registro "Alias".
 - No campo "Alias Target", selecione o Load Balancer associado ao seu aplicativo (por exemplo, o Load Balancer criado para o seu aplicativo web).
 - Preencha outros detalhes necessários (por exemplo, nome do registro) e clique em "Criar Registros".

Para os Subdomínios (por exemplo, subdominio.villas.digital):

1. Para cada subdomínio, repita os seguintes passos:
 - Clique no nome da zona hospedada "villas.digital".
 - Clique em "Criar Registro" no canto superior direito.
 - Escolha o tipo de registro "Alias".
 - No campo "Alias Target", selecione o Load Balancer apropriado associado a esse subdomínio específico.
 - Preencha outros detalhes necessários (por exemplo, nome do registro) e clique em "Criar Registros".

Importante:

- **Alias Target para Load Balancer:** O campo "Alias Target" deve ser preenchido com o DNS do Load Balancer que deseja associar. Esse DNS é geralmente encontrado na descrição do Load Balancer no Console da AWS.
- **Validação:** Certifique-se de que os Load Balancers estão configurados corretamente para lidar com o tráfego para seus aplicativos.
- **Tempos de Propagação:** Mudanças nos registros DNS podem levar algum tempo para se propagar globalmente. Aguarde alguns minutos a algumas horas.
- Na página de configuração da instância, role para baixo até encontrar a seção "Advanced Details" (Detalhes Avançados).
- No campo "User Data," insira o script de inicialização **para instalar Ruby e Nginx:**

```
#!/bin/bash
apt-get -y update
apt-get -y install ruby
apt-get -y install wget
apt-get -y install nginx

cd /home/ubuntu
wget https://bucket-name.s3.amazonaws.com/latest/install
chmod +x ./install
./install auto
```

bucket-name é o nome do bucket do Amazon S3 que contém os arquivos do CodeDeploy da região escolhida Por exemplo, South America - São Paulo" (sa-east-1) , substitua *bucket-name* por *aws-codedeploy-sa-east-1*

CONFIGURAR A IMPLANTAÇÃO DO CÓDIGO (CI/CD) fazer login no console do Amazon CodeDeploy
<https://console.aws.amazon.com/codesuite/codedeploy/start> na região escolhida
Crie um aplicativo e selecione Amazon EC2 como plataforma de computação e configure um grupo de implantação associado a instancia EC2

The screenshot shows the AWS CodeDeploy console interface. At the top, a green banner indicates "Application created" with a message: "In order to create a new deployment, you must first create a deployment group." To the right of this banner is a button that says "Create a notification rule for this application". Below the banner, the breadcrumb navigation reads "Developer Tools > CodeDeploy > Applications > [Application Name]". There are two buttons: "Notify" with a bell icon and "Delete application". The "Application details" section shows the "Name" of the application and the "Compute platform" set to "EC2/On-premises". Below this, there are three tabs: "Deployments", "Deployment groups" (which is selected and highlighted in orange), and "Revisions". The "Deployment groups" section shows a "Create deployment group" button in orange, along with "View details" and "Edit" buttons. There is a search bar and pagination controls showing "1" of 1 items. At the bottom, a table header is visible with columns: "Last attempted", "Last successful", and "Triquer".

CONFIGURAR O PIPELINE DE CÓDIGO Acesse o AWS Code pipeline ,crie um novo pipeline Configure os detalhes do pipeline incluindo a origem(GITHUB) Selecione o aplicativo Code Deploy e o grupo de implantação

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connected

✔ You have successfully configured the action with the provider.

Repository



⚠ You must select an repository

Branch



⚠ You must select a branch

Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.



GitHub webhooks (recommended)

Use webhooks in GitHub to automatically start my pipeline when a change occurs



AWS CodePipeline

Use AWS CodePipeline to check periodically for changes

Pule a etapa de construção, o Rails não é construído

Add build stage Info

Build - optional

Build provider

This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

AWS CodeBuild

▼

Region

US East (N. Virginia)

▼

Project name

Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

Q

or

Create project

↗

Environment variables - optional

Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#) ↗

Add environment variable

Selecione o aplicativo e o grupo de
implantação criado

https://villasbrands.sharepoint.com/sites/TransformaoDigital/SitePages/Migração-da-Aplicação-no-heroku-para-aws.aspx

13/22

**You cannot skip this stage**

Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

Deploy provider

Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

**Region**

US East (N. Virginia)

**Application name**

Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

**Deployment group**

Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.



Agora, anexe uma função à instância EC2 para dar acesso à implantação de código


Para fazer isso, acesse o painel IAM > Função > Criar Função


Selecione **serviço EC2** e clique em **Próximo: permissões**


Create role


1 2 3 4

Select type of trusted entity


AWS service
 EC2, Lambda and others


Another AWS account
 Belonging to you or 3rd party


Web identity
 Cognito or any OpenID provider


SAML 2.0 federation
 Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway

CodeGuru

EMR

KMS

Rekognition

AWS Backup

CodeStar Notifications

ElastiCache

Kinesis

RoboMaker

Required

Cancel

Next: Permissions

Encontre e selecione: **AmazonS3FullAccess**,
AmazonEC2RoleforAWSCodeDeploy e crie a política embutida abaixo

```
{
  "Versão": "2012-10-17",
  "Declaração": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Permitir",
      "Ação": [
        "ssm:GetParameterHistory", "ssm:GetParametersByPath ",
        "ssm:GetParameters",
        "ssm:GetParameter",
      ],
      "Recurso": "arn:aws:ssm: REGION : ACCOUNT_ID :parameter/*"
    }
  ]
}
```

Altere os parâmetros região e account_id para os seus

Escolha **Próximo: Tags**

Deixe a página **Step Tags** inalterada e escolha **Next: Review**

Insira o nome da sua função e clique em **Criar função**

Vá para o painel do EC2> encontre sua instância do EC2

Selecione sua instância > Ação > Configurações da instância > Anexar/Substituir função do IAM

Crie uma pasta no terminal dependencies_install.sh (pode usar SSH para acessar a instância remotamente com o aws configure no terminal)

```
#!/bin/bash

export PATH=/home/ubuntu/.rvm/gems/ruby-2.5.0/bin:/home/ubuntu/.rvm/gems/ruby-2.5.0@global/bin:/home/ubuntu/.rvm/rubies/ruby-2.5.0/bin:/home/ubuntu/bin:/home/ubuntu/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/ubuntu/.rvm/bin:/home/ubuntu/.rvm/bin

sudo kill -9 $(cat /var/www/my-app/tmp/pids/server.pid)

cd /var/www/my-app/

#sudo apt-get install ruby2.3-dev libffi-dev -y

gem install nokogiri -- use-system-libraries

bundle config build.nokogiri -- use-system-libraries

gem install bundler -- user-install

bundle install
```

torne o script executável `chmod +x dependencies_install.sh` execute o script `./dependencies_install.sh`

O script realizará as seguintes ações:

Define o caminho (PATH) para as gemas Ruby.

Navega até a pasta do aplicativo Rails.

Instala a gem nokogiri com a opção `--use-system-libraries`, o que permite que a gem use bibliotecas do sistema.

Configura a construção da gem nokogiri para usar bibliotecas do sistema.

Instala o Bundler usando a opção `--user-install`.

Executa bundle install para instalar as dependências do Ruby

Crie um arquivo chamado "appspec.yml" na pasta raiz do projeto - Este é um arquivo de configuração usado pelo AWS CodeDeploy para especificar como os arquivos e as permissões devem ser tratados durante o processo de implantação

```
version: 0.0
os: linux
files:
- source: /
  destination: /var/www/my-app/
  permissions:
  - object: /var/www/my-app/
  pattern: "*"
  owner: ubuntu
  group: ubuntu
  mode: 775
  hooks:
  AfterInstall:
  - location: scripts/dependencies_install.sh
    runas: ubuntu
```

Altere o arquivo **appspect.yml** adicionando isto no final:

```
- location: scripts/set_environment_variables.sh
  runas: ubuntu
```

O pipeline de código está funcionando com o github, se alterar qualquer código e enviar para o servidor, verá em execução:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Configurar parâmetros usando o AWS Systems Manager

Armazenamento de parâmetros: o Parameter Store serve para armazenar informações como configurações de aplicativos, chaves de API, senhas de banco de dados, cadeias de conexão e outros dados de configuração. Os parâmetros podem ser organizados em hierarquias e versões, permitindo rastrear e gerenciar alterações ao longo do tempo.

Acesse a página do Systems Manager no AWS Dashboard.

Selecione a opção **Parameter Store** :

AWS Systems Manager

Quick Setup

▼ Operations Management

Explorer New

OpsCenter

CloudWatch Dashboard

Trusted Advisor & PHD

▼ Application Management

Resource Groups

AppConfig New

Parameter Store

Clique em Parameter Store

Insira o nome, selecione o nível padrão, string para dados não seguros e SecureString para dados seguros.

Após a criação do parâmetro no AWS Parameter Store, é necessário desenvolver um código que permita a recuperação desses parâmetros e a configuração das variáveis de ambiente no ambiente de execução da aplicação. Isso é essencial para garantir a segurança e a flexibilidade na gestão das configurações da aplicação.

Crie um arquivo chamado **set_environment_variables.sh**, em uma pasta **de scripts**

```
#!/bin/bash

function set_parameter {
    SSM_PARAM_NAME=$1

    SSM_VALUE=`aws ssm get-parameters -- with-decryption -- names
"${SSM_PARAM_NAME}" -- query 'Parameters[*].Value' -- output text`

    [ "$(eval echo "$"$1)" == "" ] && echo "export
${SSM_PARAM_NAME}=${SSM_VALUE}" >> ~/.bashrc
}

set_parameter "DB_USER"
set_parameter "DB_PASS"
set_parameter "DB_HOST"
```

Altere o arquivo **database.yml** para obter variáveis do ambiente

```
database.yml — Backend

database.yml ●

ig > database.yml

# MySQL. Versions 5.1.10 and up are supported.
#
# Install the MySQL driver
#   gem install mysql2
#
# Ensure the MySQL gem is defined in your Gemfile
#   gem 'mysql2'
#
# And be sure to use new-style password hashing:
#   https://dev.mysql.com/doc/refman/5.7/en/password-hashing.html
#
default: &default
  adapter: mysql2
  encoding: utf8mb4
  charset: utf8mb4
  collation: utf8mb4_unicode_ci
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
  username: <%= ENV['DB_USER'] %>
  password: <%= ENV['DB_PASS'] %>
  host: <%= ENV['DB_HOST'] %>
  port: 3306
```

Referências

<https://docs.aws.amazon.com/codedeploy/latest/userguide/instances-ec2-configure.html>

<https://aws.amazon.com/blogs/mt/the-right-way-to-store-secrets-using-parameter-store/>

