

DAY_1.1

Tarde

OC2023 – SKILL 53 – DIA

Proposto por:
Vitor Hugo Alves Vieira
Jader Lucas de Souza Santos

Introdução

A competição tem um horário fixo para começar e terminar. Você deve decidir como melhor dividir o seu tempo. Por favor, leia atentamente as seguintes instruções.

Quando o tempo da competição terminar, por favor, deixe sua estação funcionando. A avaliação será feita no estado em que se encontra. Nenhuma reinicialização será iniciada e as máquinas desligadas não serão ligadas.

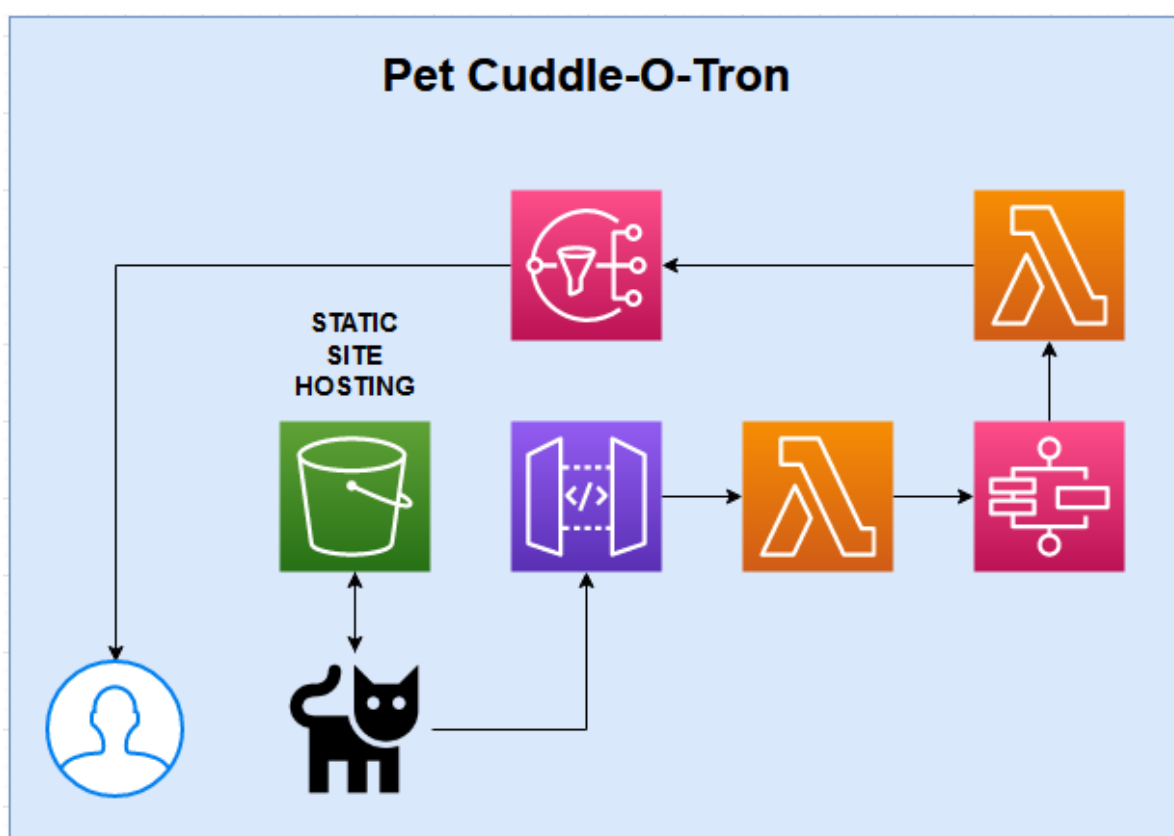
Instruções para os competidores

- Telefones celulares não podem ser usados;
- Leia todo o teste antes de começar;
- Planeje suas tarefas com cuidado;

Tarefa 02: Pet-Cuddle-O-Tron

Introdução

Você foi contratado como um arquiteto de soluções junior e sua primeira demanda é a seguinte: Uma solução desenvolvida para gatinhos em que eles podem requisitar que seus humanos voltem para casa para dar atenção às necessidades de seus mestres. A partir de um site estático no S3, os gatinhos podem agendar um temporizador no “Step Functions” que, quando chega à zero, dispara uma mensagem a partir do SNS. O seguinte modelo de arquitetura deverá ser seguido:



ETAPA 1: Configuração do SNS

A partir do SNS, os gatinhos irão registrar os endereços de e-mail de seus humanos para que eles sejam notificados onde quer que estejam.

- Crie um tópico no SNS com o nome de “PetCuddleOTronTopic”.

Vários gatinhos com vários humanos planejam utilizar esta solução. É preciso garantir que as ordens de um gatinho somente serão enviadas ao seu humano correspondente.

- Implemente um filtro para que as mensagens do SNS somente sejam enviadas para o devido correspondente. (*Informado pelos gatinhos através do site estático, no momento do agendamento, através do parâmetro “email”).

ETAPA 2: Configuração da função de envio de mensagens no Lambda.

Com o tópico SNS criado e seu filtro configurado, agora é preciso provisionar uma função do Lambda que será acionada quando o temporizador da “Step Functions” chegar a zero.

- Crie uma função Lambda para encaminhar mensagens ao SNS:
 - O nome da função deve ser “email_reminder”.
 - A função vai receber um evento do “Step Functions” no seguinte modelo:

```
{"waitSeconds": "120", "message": "SAMPLE MESSAGE", "email": "samplehuman@sample.com"}
```

- Você deve encaminhar a mensagem “message” ao tópico do SNS criado previamente para que seja feito o envio do comunicado somente ao humano especificado em “email”.

ETAPA 3: Implementar e configurar a “state machine” no “Step Functions”

Agora você precisará de configurar o serviço principal da solução, através do “Step Functions” você irá criar um temporizador que quando chega a zero, aciona a função do lambda “email_reminder”.

- Crie uma “state machine” no “Step Functions”.
 - O nome da “state machine” deve ser “PetCuddleOTronMachine”.
 - A “state machine” vai receber um evento do “Lambda” no seguinte modelo:

{“waitSeconds”: “120”, “message”: “SAMPLE MESSAGE”, “email”: “samplehuman@sample.com”}

- A “state machine” deverá aguardar, em segundos, pelo tempo indicado no evento através do parâmetro “waitSeconds”.
- Após aguardar este tempo, você deverá invocar a função “email_reminder”.
- Após estes eventos ocorrerem, você pode terminar a execução da “state machine”.

ETAPA 4: Criação do API Gateway e da função lambda de suporte

Antes de você criar a API Gateway, você precisará criar uma função lambda que iniciará a “state machine” com os parâmetros corretos.

- Crie uma função no lambda que será chamada pelo API Gateway para iniciar a “state machine”.
 - O nome da função deve ser “state_machine_start”.
 - A função vai receber um evento do “API Gateway” no seguinte modelo:

```
{“waitSeconds”: “120”, “message”: “SAMPLE MESSAGE”, “email”: “samplehuman@sample.com”}
```

- Repasse o evento para o “Step Functions”, garantindo que sua execução seja concluída com êxito e a “state machine” consiga acessar os parâmetros necessários para a sua execução.

Os gatinhos são muito exigentes, mas não muito inteligentes. Para garantir a satisfação dos felinos faça com que sua função do lambda também realize o seguinte.

- Garanta que sempre seja retornado no corpo da resposta gerada pelo lambda o seguinte:

```
{"Status": "Success"}
```

Com a função do lambda pronta, agora você terá que configurar o API Gateway.

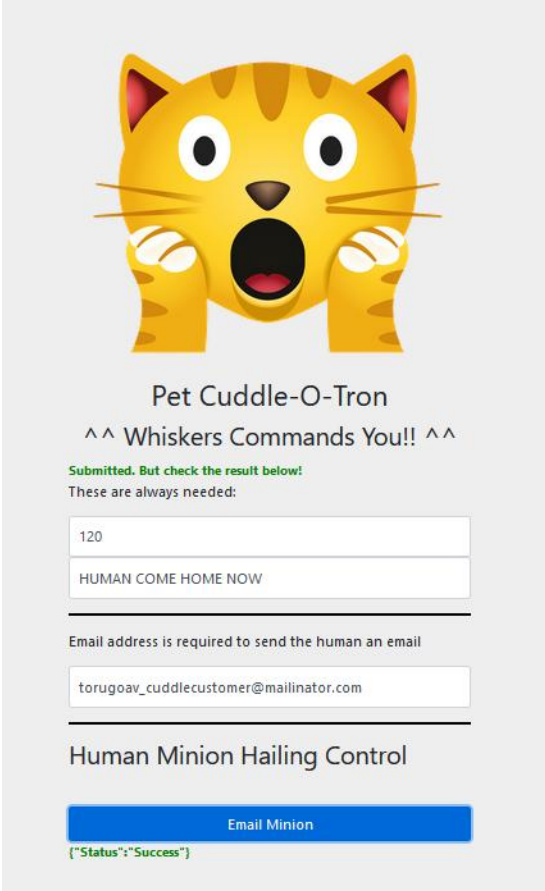
- Crie uma API no API Gateway:
 - O nome da API deve ser “petcuddleotron”.
 - Crie um recurso com o nome de “petcuddleotron”.
 - Dentro do recurso, configure o “petcuddleotron” somente para receber requisições do tipo “POST”.
 - As requisições do tipo “POST” devem acionar a função lambda “state_machine_start”, criada anteriormente.

ETAPA 5: Implementação do “front-end” estático da aplicação.

Com a solução pronta, os gatinhos precisam de uma interface para conseguirem utilizar a aplicação. Agora você irá implantar um site estático no S3 que irá prover a interface necessária para comunicação com o API Gateway.

- Crie um site estático no S3
 - O nome do bucket utilizado deverá começar com “PetCuddleOTronBucket”
 - O conteúdo do site estático está localizado no seguinte arquivo zip “https://learn-cantrill-labs.s3.amazonaws.com/aws-serverless-pet-cuddle-o-tron/serverless_frontend.zip”. Extraia o arquivo e indique as informações da sua API no arquivo “serverless.js” antes de utilizá-lo.

Com isso concluído, os gatinhos devem conseguir acessar o site estático, informar o tempo desejado, uma mensagem personalizada e o email de destino para contatar seus humanos. A inscrição de e-mails no SNS será feita de forma manual, então não é preciso se preocupar com isso. Se tudo estiver certo, os gatinhos conseguirão utilizar a solução como no exemplo abaixo.



Pet Cuddle-O-Tron

^^ Whiskers Commands You!! ^^

Submitted. But check the result below!

These are always needed:

120

HUMAN COME HOME NOW

Email address is required to send the human an email

torugoav_cuddlecustomer@mailinator.com

Human Minion Hailing Control

Email Minion

{\"Status\": \"Success\"}