

Índice

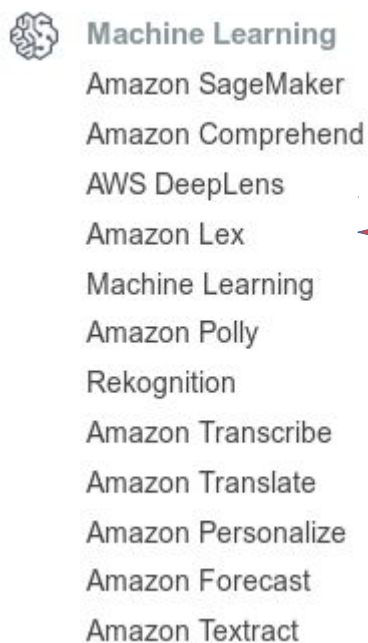
Exercícios práticos de laboratórios.....	3
Hack400 – Criando um Bot no AWS com Amazon Lex.....	3
Hack401 – Configurando as intenções de Bebidas do Pedido (Beverage Intent).....	5
Hack402 – Configurando mensagens de prompt.....	7
Hack403 – Configurando os Utterances(expressões) de pedido.....	7
Hack404 – Criando SlotType.....	8
Hack405 – Criando um Slot.....	9
Hack406 – Criando demais Slot types.....	10
Hack407 – Criando demais slots.....	10
Hack408 – Marcando Slots Obrigatórios.....	11
Hack409 – Realizando o Build do Bot.....	11
Hack410 – Testando o Bot.....	12
Hack412 – Faça mais um teste agora pedindo um item não obrigatório.....	14
Hack413 – Alterando a tentativas mensagem para Error Handling	15
Hack414 – Exportando o bot Lex.....	16
Hack415 – Criando uma credencial de acesso Externo.....	17
Hack416 – Página exemplo em Javascript para atendimento por VOZ.....	18
Hack417 – Criando um Bucket de armazenamento S3.....	19
Hack418 – Instalando o boto SDK.....	21
Hack419 – Executando um exemplo de API com Amazon Polly.....	21

Exercícios práticos de laboratórios.

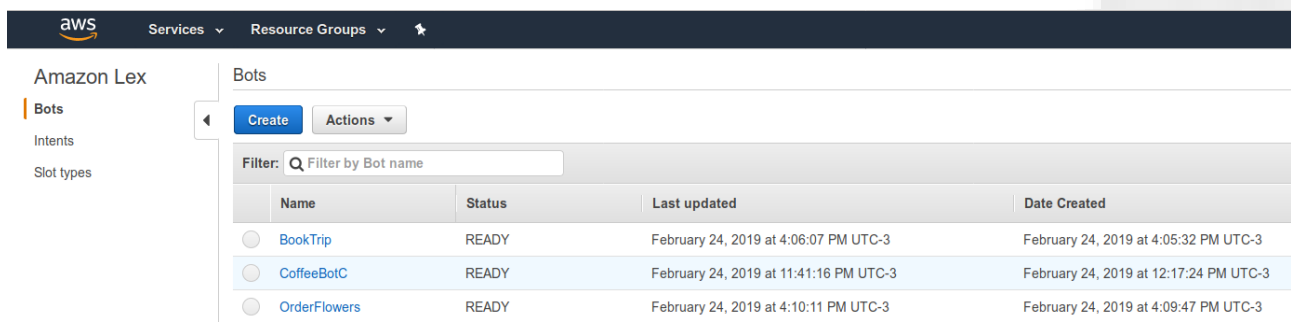
Cria uma pasta em seu ambiente para armazenar estes Hacks.

Hack400 – Criando um Bot no AWS com Amazon Lex.

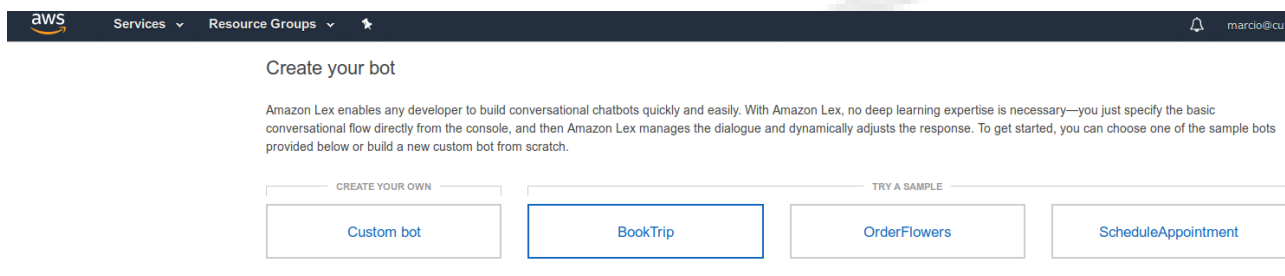
1. Acesse o portal da AWS com sua conta AWS em <https://console.aws.amazon.com/>
2. Clique no menu superior Services e depois em Machine Learning → **Amazon Lex**



3. Clique em CREATE na tela seguinte.



4. Na tela seguinte teremos a opção de criar um bot com base nos exemplos , mas criaremos um customizado (custom) para entendermos todos os conceitos. Clique em **custom bot**



- Informe os parâmetros conforme imagem, a seguir um detalhamento de cada campo:
 Bot name (Nome do seu bot): **coffeeFIAP**
 Output voice(opção de voz, pode testar uma que lhe agrada): **Joanna**
 Session timeout(este e o tempo que o bot tem para atender a requisição): **5 min.**
 Em COPPA marque: **No**

Bot name

Language

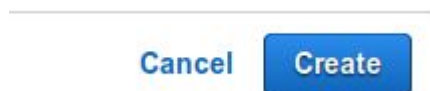
Output voice

Session timeout

IAM role [AWSServiceRoleForLexBots](#)
Automatically created on your behalf

COPPA Please indicate if your use of this bot is subject to the [Children's Online Privacy Protection Act \(COPPA\)](#). [Learn more](#)
☐ Yes ☒ No

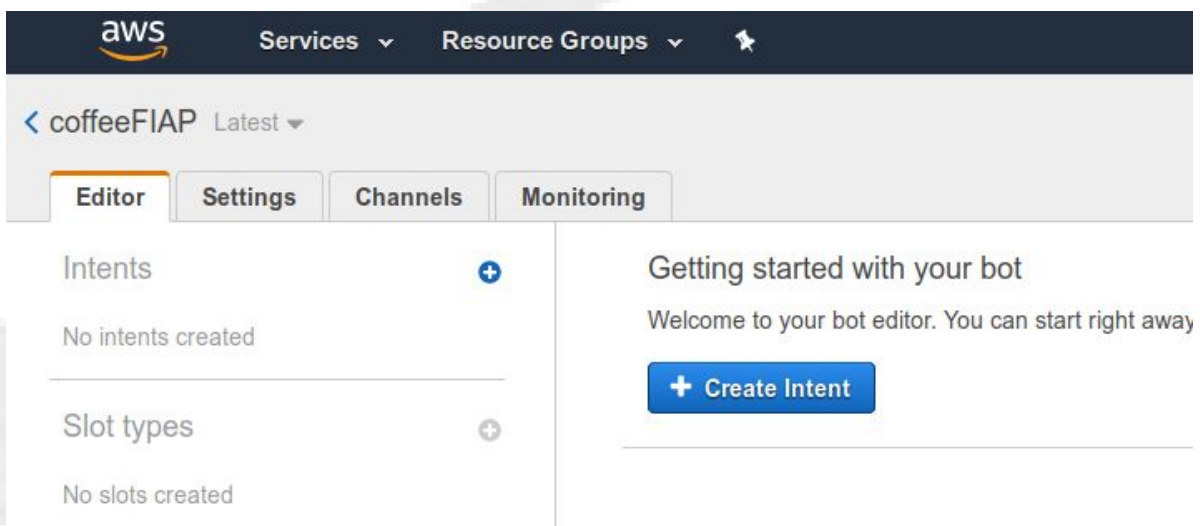
- Clique em Create para criar o bot.



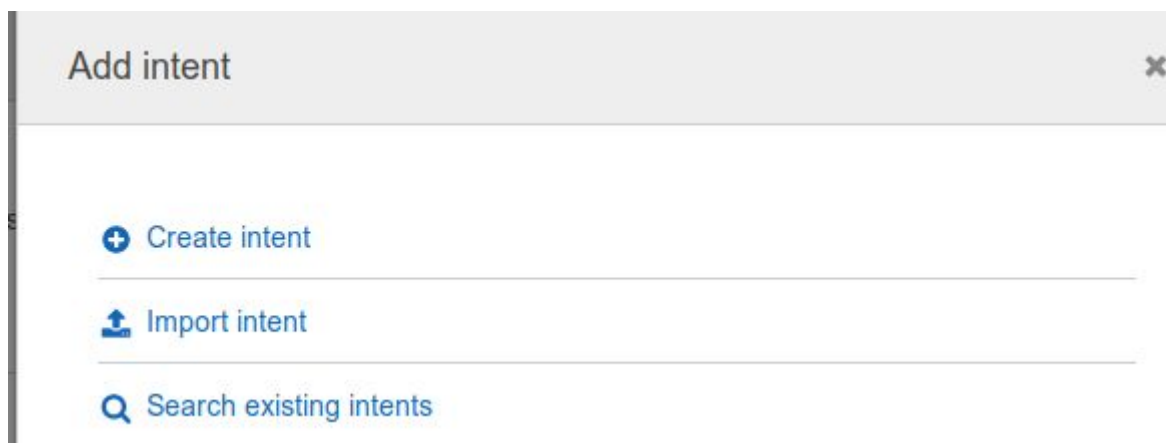
- Com o bot criado será aberta uma tela de editor e agora vamos configurar o bot.

Hack401 – Configurando as intenções de Bebidas do Pedido (Beverage Intent).

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> e selecione nosso bot coffeeFIAP
2. Clique em + **Create Intent**



3. Novamente clique em Create intent



- Informe o nome do intent com `cafeOrderBeverageIntentFIAP` e clique em add

- Será aberta uma tela com diversas configurações a fazer , vamos cadastrar as básicas inicialmente.

- Na opção Lambda *initialization and validation* deixe desmarcada (é o default).

- Na opção *Fulfillment*: marque "**Return parameters to client**" (é o default)

- Salve no botão inferior **Save Intent**

Hack402 – Configurando mensagens de prompt.

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> e selecione nosso bot coffeeFIAP

2. No item **Confirmation prompt**: que é usado para a confirmação do pedido informe:

You'd like me to order a {BeverageFIAPSize}
{BeverageFIAPType}. Is that right?

3. Para cancelar um pedido use o campo **cancel** informe:

Okay. Nothing to order this time. See you next time!

4. Conforme imagem e depois Salve no botão Salve Intent.

▼ Confirmation prompt ⓘ

☒ Confirmation prompt

Confirm

You'd like me to order a {BeverageFIAPSize} {BeverageFIAPType}. Is that right?

Cancel (if the user says "no")

Okay. Nothing to order this time. See you next time!

Hack403 – Configurando os Utterances(expressões) de pedido.

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> e selecione nosso bot coffeeFIAP
2. No campo Sample Utterances adicione linha a linha as opções a seguir:

I would like a {BeverageFIAPSize} {BeverageFIAPType}

Can I get a {BeverageFIAPType}

May I have a {BeverageFIAPSize} {CreamerFIAP} {BeverageFIAPType}

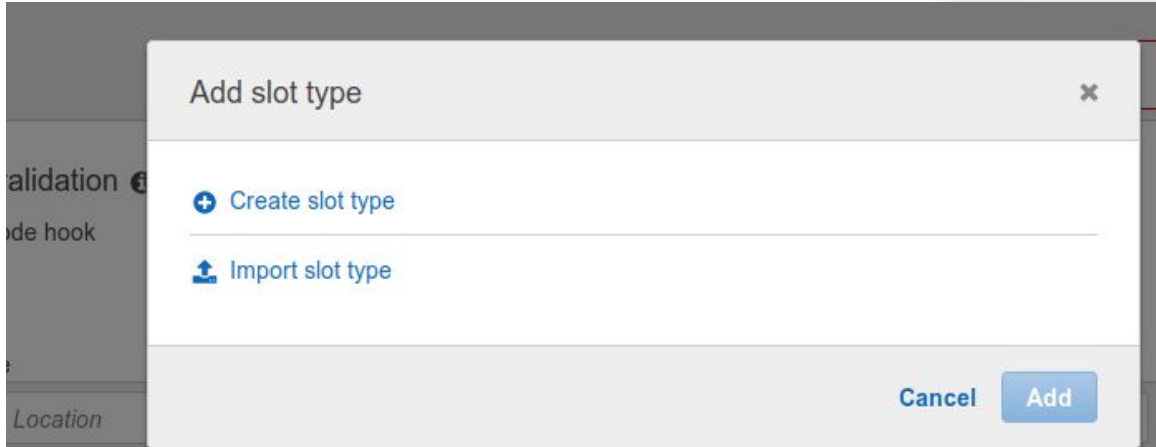
Can I get a {BeverageFIAPSize} {BeverageFIAPTemp} {CreamerFIAP}
{BeverageFIAPType}

Let me get a {BeverageFIAPSize} {CreamerFIAP} {BeverageFIAPType}

3. Salve no botão inferior

Hack404 – Criando SlotType.

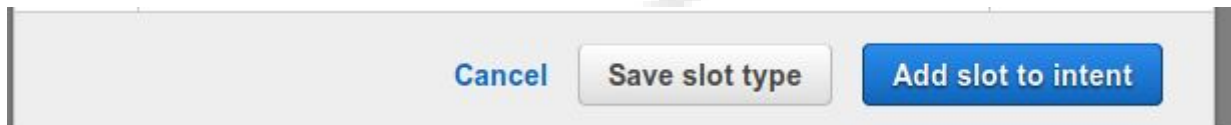
1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> e selecione nosso bot coffeeFIAP
2. No menu a esquerda Slot Type clique no +, e depois em **Create slot type**



3. De o nome do primeiro de `cafeBeverageFIAPType` e adicione todas as opções a seguir no campo VALUE, uma em cada linha.

coffee; cappuccino; latte; mocha; chai; espresso; smoothie

4. Clique em **Save slot Type**



Hack405 – Criando um Slot.

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> , selecione seu Bot coffeeFIAP.
2. No item slot adicione na propriedade
 Name : **BeverageFIAPType**
 Slot Type: **cafeBeverageFIAPType** (selecione pois esta previamente cadastrado)
 E no prompt : **What kind of beverage would you like? For example, mocha, chai, etc.**

Slots ⓘ						
Priority	Required	Name	Slot type	Version	Prompt	Settings
		BeverageFIAPType	cafeBeverageFIAPType		What kind of beverage would you like? For exampl	+

3. Apos informas clique em ENTER ou no botão + no final da linha
4. Veja que ao cadastrar o Slot funciona como uma variável para o ambiente e é recinhecido nos locais com o uso de chaves {slotname}.

ld like to book a flight.

a {BeverageFIAPSize} {CreamerFIAP} {BeverageFIAPType}

a {BeverageFIAPSize} {BeverageFIAPTemp} {CreamerFIAP} {BeverageFIAPType}

a {BeverageFIAPSize} {CreamerFIAP} {BeverageFIAPType}

a {BeverageFIAPType}

a {BeverageFIAPSize} {BeverageFIAPType}

5. Salve nosso intent

Hack406 – Criando demais Slot types

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> , Selecione seu Bot coffeeFIAP
2. Cadastre os 3 Slot Types a seguir, com seus respectivos values

Slot type name	Values (um a um por linha)
cafeBeverageFIAPSize	kids; small; medium; large; extra large; six ounce; eight ounce; twelve ounce; sixteen ounce; twenty ounce
cafeCreamerFIAPType	two percent; skim milk; soy; almond; whole; skim; half and half
cafeBeverageFIAPTemp	kids; hot; iced

3. Salve um a um.

Hack407 – Criando demais slots.

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> , Selecione seu Bot coffeeFIAP.

Name	Slot type	Prompt
BeverageFIAPSize	cafeBeverageFIAPSize	What size? small, medium, large?
CreamerFIAP	cafeCreamerFIAPType	What kind of milk or creamer?
BeverageFIAPTemp	cafeBeverageFIAPTemp	Would you like that iced or hot?

2. Salve o intent.

Hack408 – Marcando Slots Obrigatórios.

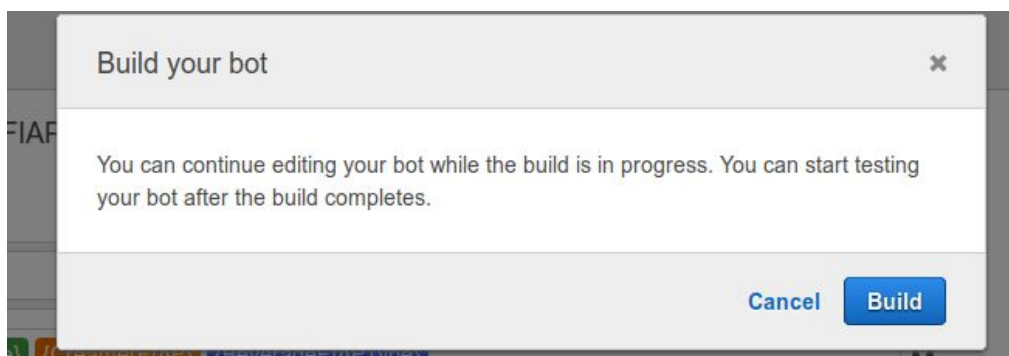
1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> , selecione seu Bot coffeeFIAP.
2. Ao lado do Slots tem um campo required, marque os dois primeiros como obrigatórios. (type e size)

Priority	Required	Name	Slot type	Version	Prompt
		e.g. Location	e.g. AMAZON.US_CITY		e.g. What city?
1.	<input checked="" type="checkbox"/>	BeverageFIAPType	cafeBeverageFIAPType	1	What kind of beverage would you like? For exampl
2.	<input checked="" type="checkbox"/>	BeverageFIAPSize	cafeBeverageFIAPSize	1	What size? small, medium, large?

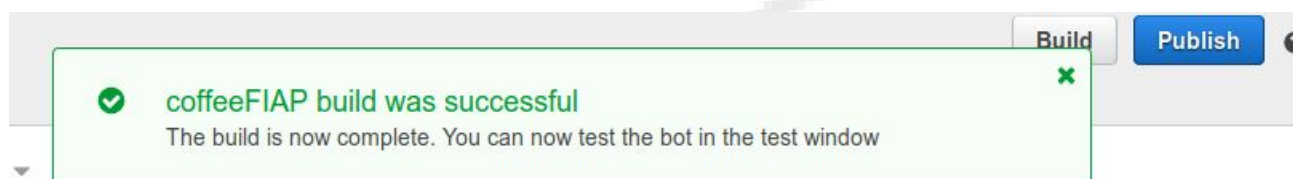
3. Salve o intent.

Hack409 – Realizando o Build do Bot.

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> , selecione seu Bot coffeeFIAP.
2. Clique no botão build confirme e aguarde terminar.



3. Após terminar você receberá uma mensagem de confirmação se tudo estiver correto.



Hack410 – Testando o Bot.

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex>, selecione seu Bot coffeeFIAP.
2. Clique no ícone Test Chat Bot e adicione as frases para realizar seu pedido.

Can i get a mocha

> Test bot (Latest)

✓ Ready. Build complete.

Can i get a mocha

What size? small, medium, large?

3. Veja que o Slot é preenchido com o valor pedido e os demais ainda nulos

Inspect response

Dialog State: ElicitSlot

Hide

☒ Summary ☐ Detail

Intent: cafeOrderBeverageIntentFIAP

Slots (1/4)

BeverageFIAPSize	null
BeverageFIAPTemp	null
BeverageFIAPType	mocha
CreamerFIAP	null

4. Agora responda o size:

medium

> Test bot (Latest)

✓ Ready. Build complete.

can I get a mocha

What size? small, medium, large?

medium

You'd like me to order a medium mocha. Is that right?

5. Veja que o outro slot e preenchido e é requisitado para confirmar e encerrar o pedido.

Inspect response

Dialog State: ConfirmIntent

[Hide](#)

☒ Summary ☐ Detail

Intent: cafeOrderBeverageIntentFIAP

Slots (2/4)

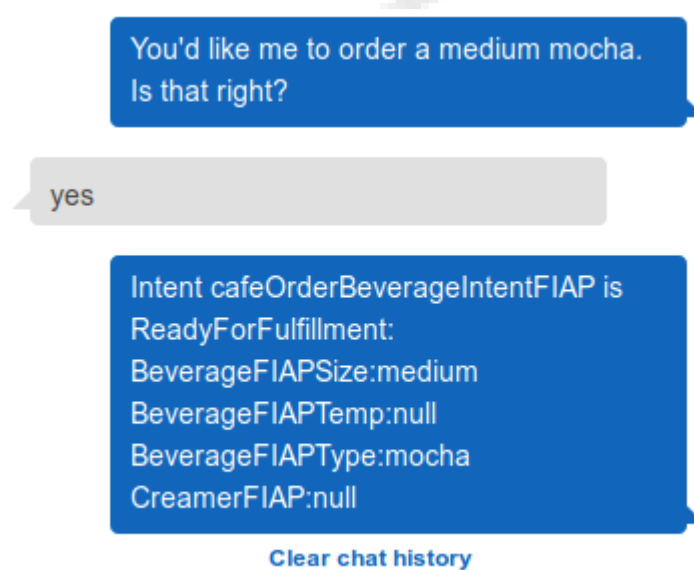
BeverageFIAPSize medium

BeverageFIAPTemp null

BeverageFIAPType mocha

CreamerFIAP null

6. Agora confirme com Yes e os dados do Intent serão apresentados.



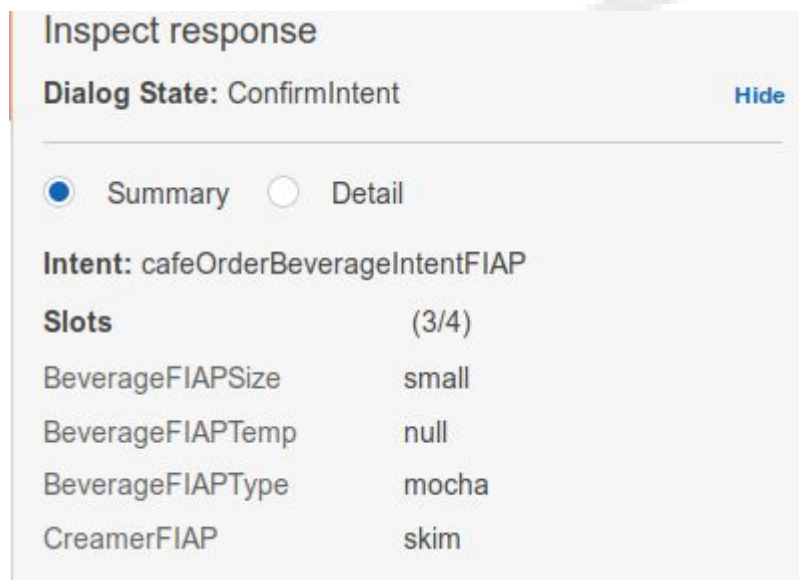
7. Podem ser usadas funções Lambda para que estes dados do Intent possam interagir como uma trigger em qualquer outra aplicação.

Hack412 – Faça mais um teste agora pedindo um item não obrigatório.

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> , Selecione seu Bot coffeeFIAP,
2. Acesse Test chatbot e informe :

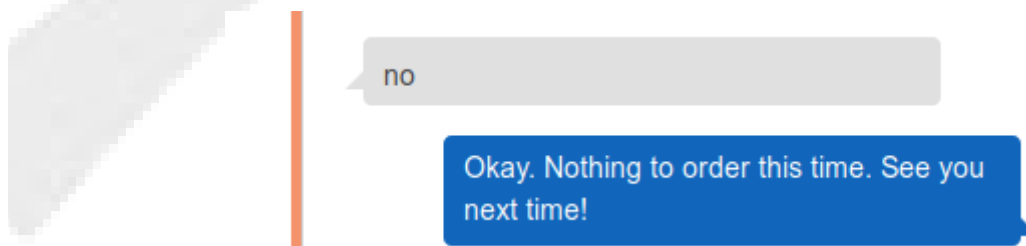
Can i get a small skim mocha

3. Veja que todos os slots são preenchidos (isso claro pois adicionamos na mesma ordem esperada)



4. Não confirme o Pedido.

No



5. Veja que nossa mensagem de cancel e mostrada.

Hack413 – Alterando a tentativas mensagem para Error Handling .

1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex> , Selecione seu Bot coffeeFIAP.
2. Clique no menu lateral em Error Handling
3. Veja que temos a mensagem para quando o Bot não entende o que pedimos e a quantidade de vezes que ele vai solicitar novamente.

Error handling

Clarification prompts

+

×

Maximum number of retries

Hang-up phrase

+

×

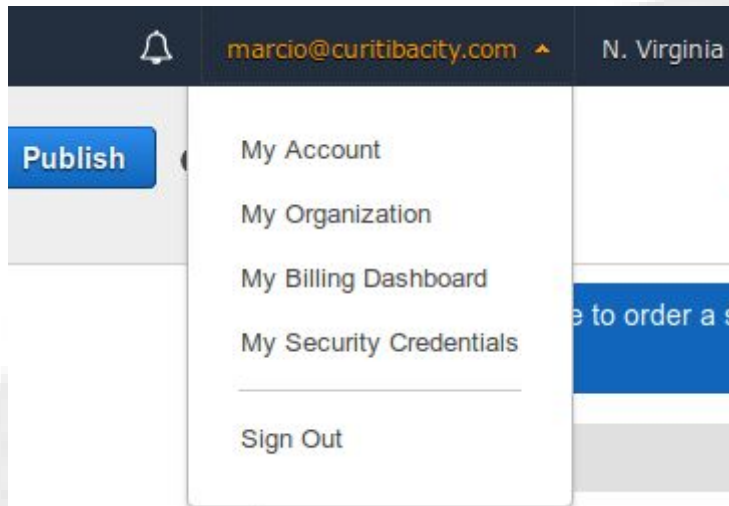
4. Altere frase e realize um build e teste 5 vezes com algo sem sentido (ou não previsto), tipo diga oi ... oi ...oi 5 vezes. Na quinta a mensagem de Hang será emitida

Hack414 – Exportando o bot Lex.

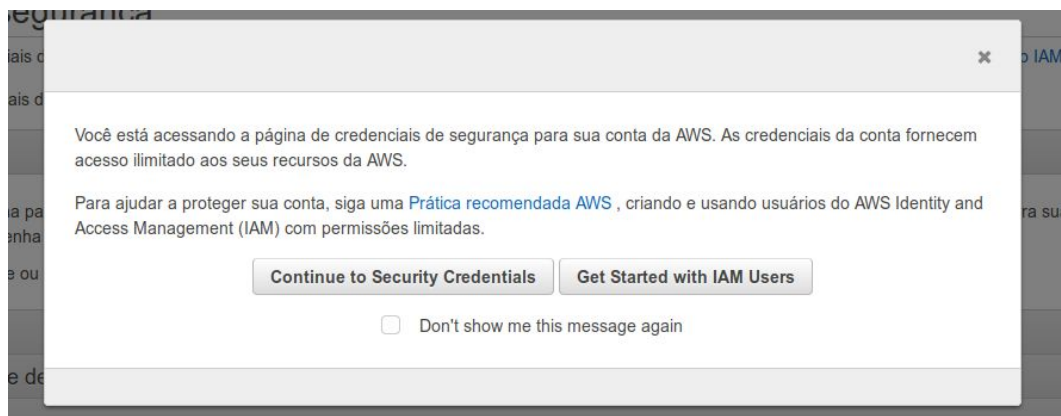
1. Acesse o portal da AWS a Lex <https://console.aws.amazon.com/lex>
2. Selecione seu Bot coffeeFIAP, e em action clique em exporta.
3. Com este arquivo pode importar seu bot em qualquer conta AWS.

Hack415 – Criando uma credencial de acesso Externo.

1. Acesse o portal da AWS
2. Clique no seu login no menu superior e depois em My Security Credentials



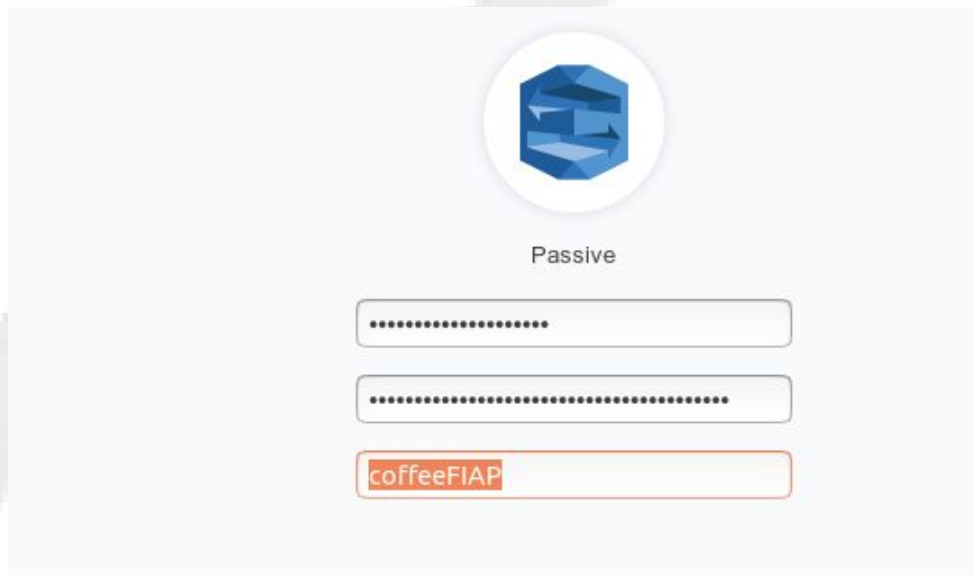
3. Será emitido um alerta sobre as credenciais



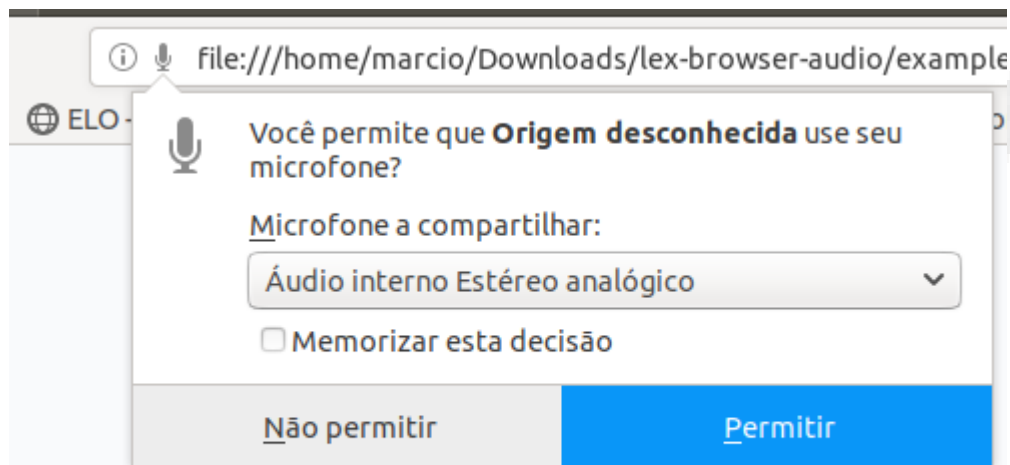
4. Clique em **Continue to Security Credentials**
5. Clique em Chaves de acesso (ID da chave de acesso e a chave de acesso secreta)
6. Clique em **Criar nova chave de acesso**, veja que ao final ele vai pedir para baixar em sua máquina, faça o download do arquivo.
7. Neste arquivo teremos duas informações que vamos usar:
8. **AWSAccessKeyId** e **AWSecretKey**

Hack416 – Página exemplo em Javascript para atendimento por VOZ.

1. Faça o Download de <https://github.com/marciojv/hacks-cognitives-plataforms/blob/master/aws/javascript/lex-browser-audio.zip>
2. Descompacte em qualquer local de sua máquina e acesse o diretório example e clique em index.html
3. Informe os ids que estão no arquivo do hack anterior e informe na pagina assim como o nome do nosso Bot

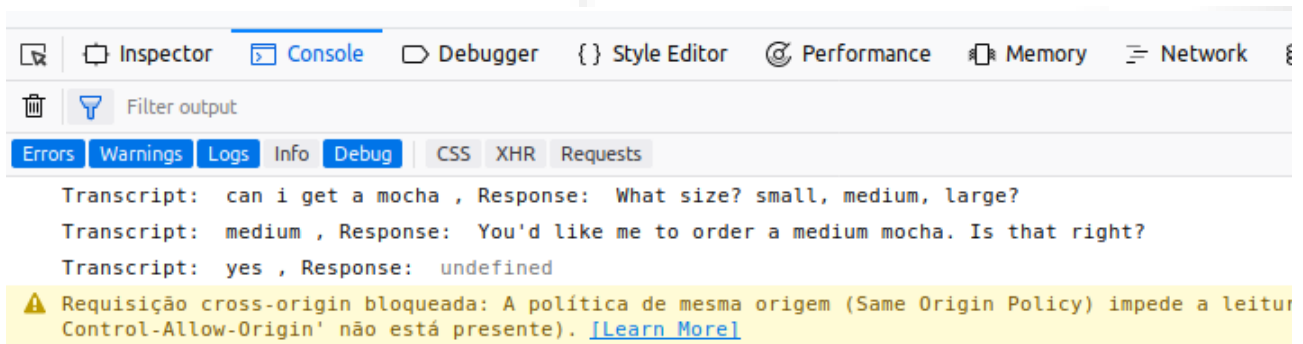


4. Clique no ícone da Amazon Lex , será solicitado para dar acesso ao microfone



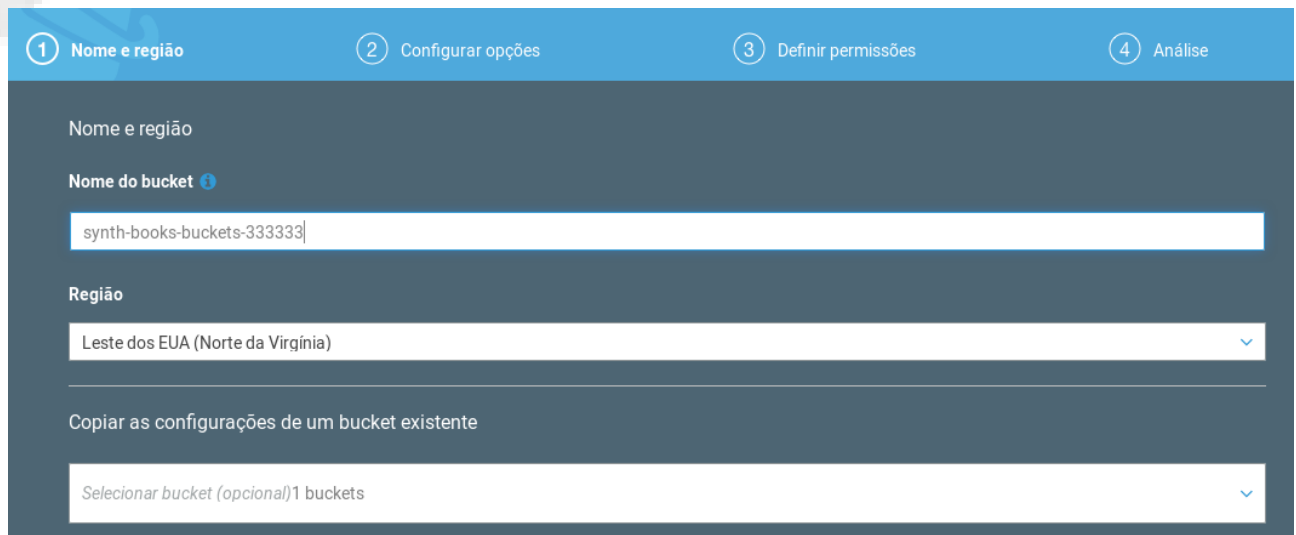
5. Faça o mesmo pedido por voz do teste anterior com o **mocha** e tamanho **medium**.

6. Pode inspecionar no browser os resultados em texto com F12



Hack417 – Criando um Bucket de armazenamento S3.

1. Acesse <https://console.aws.amazon.com> e depois Armazenamento – S3
2. Clique em Criar um bucket
3. De o nome do mesmo de synth-books-buckets-SUAMATRICULA e mantenha a região como Leste dos EUA (código interno = us-east-1) e clique em próximo.



4. As demais configurações mantenha todos como padrão, não altere nenhuma!

- Desmarque as opções para bloquear acesso publico, clique em próximo e depois em criar Bucket.

Observação: é possível conceder acesso a usuários específicos depois de criar o bucket.

Bloquear acesso público (configurações de bucket)

O acesso público é concedido para buckets e objetos por meio de listas de controle de acesso (ACLs), políticas de bucket ou ambas. Para garantir que o acesso público a todos os buckets e objetos do S3 seja bloqueado, ative Bloquear *todo* o acesso público. Essas configurações se aplicam apenas a este bucket. A AWS recomenda que você ative Bloquear *todo* o acesso público, mas, antes de aplicar qualquer uma dessas configurações, verifique se os aplicativos funcionarão corretamente sem acesso público. Se você precisa de algum nível de acesso público para seus buckets ou objetos dentro, é possível personalizar as configurações individuais abaixo de acordo com seus casos de uso específicos de armazenamento. [Saiba mais](#)

☒ **Bloquear *todo* o acesso público**

Ativar essa configuração é o mesmo que ativar todas as quatro configurações abaixo. Cada uma das configurações a seguir são independentes uma da outra.

- ☒ **Bloquear acesso público a buckets e objetos concedidos por meio de novas listas de controle de acesso (ACLs)**
O S3 bloqueará as permissões de acesso público aplicadas a blocos ou objetos recém-adicionados e impedirá a criação de novas ACLs de acesso público para blocos e objetos existentes. Essa configuração não altera nenhuma permissão existente que permita o acesso público aos recursos do S3 usando ACLs.
- ☒ **Bloquear acesso público a buckets e objetos concedidos por meio de qualquer lista de controle de acesso (ACLs)**
O S3 ignorará todas as ACLs que concedem acesso público a buckets e objetos.
- ☒ **Bloquear acesso público a buckets e objetos concedidos por meio de novas políticas de bucket público**
O S3 bloqueará novas políticas de bucket que concedem acesso público a buckets e objetos. Essa configuração não altera nenhuma política existente que permita o acesso público aos recursos do S3.
- ☒ **Bloquear acesso público e entre contas a buckets e objetos por meio de qualquer política de bucket público**
O S3 ignorará o acesso público e entre contas para buckets com políticas que concedem acesso público a buckets e objetos.

- Temos um bucket criado agora.

Nome do bucket	Acesso	Região	Data da criação
<input type="checkbox"/> synth-books-buckets-333333	Objetos podem ser públicos	Leste dos EUA (Norte da Virgínia)	set 19, 2019 4:53:29 PM GMT-0300

Hack418 – Instalando o boto SDK

1. Instale o SDK Python da Amazon, o boto.

pip install boto3

2. Caso tenha problemas de acesso na instalação no boto, instale com o Super Usuário ou administrador do ambiente.

Hack419 – Executando um exemplo de API com Amazon Polly.

1. Crie um diretório qualquer para incluirmos seus códigos fontes em python.

2. Faça um dowload do exemplo em

https://github.com/marciojv/hacks-cognitives-plataformas/blob/master/aws/amazon-polly/python/traduzindo_com_vitoria.py

3. Altere as variáveis com seus códigos de acesso, caso não lembro como gerar releia o hack acima (Criando uma credencial de acesso Externo):

```
ws_access_key_id='adicioneseuidaqui',
aws_secret_access_key='adicionesuachavesecreta',
```

4. Modifique no programa o seu nome de bucket em:

OutputS3BucketName='synth-books-buckets-SUAMATRICULA'

5. Execute o programa exemplo com:

python traduzindo_com_vitoria.py

6. Acesse seu bucket após alguns segundos e analise se deu tudo certo no seu arquivo .mp3

Carregar

+ Criar pasta

Fazer download

Ações

Leste dos EUA (Norte da Virgínia)

Visualização 1 para 8

<input type="checkbox"/>	Nome	Última modificação	Tamanho	Categoria de armazenamento
<input type="checkbox"/>	key.11e92243-0fa3-4c6c-a11f-854126f659a3.mp3	set 19, 2019 4:35:59 PM GMT-0300	33.4 KB	Padrão

7. Sua implementação funcionou!