

Aula 02.3 - Service Proxy

Criando a fila sqs

1. Crie uma fila no sqs colocando o nome 'demoqueue', deixe os valores default e clique em 'Create Queue'

Region: Asia Pacific (Sydney)

Queue Name: demoqueue

Configure your new queue by setting queue attributes (optional).

Default Visibility Timeout: 30 seconds

Message Retention Period: 4 days

Maximum Message Size: 256 KB

Delivery Delay: 0 seconds

Receive Message Wait Time: 0 seconds

Dead Letter Queue Settings

Use Redrive Policy: ☐

Dead Letter Queue:

Maximum Receives:

Cancel Create Queue

2. Copie o ARN da sua fila. É a terceira informação da sua aba 'Details' quando a fila esta selecionada.

Criando a Role do Api Gateway

3. Abra outra aba e vá para o menu de IAM. Clique em 'Policies' no menu ao lado esquerdo
4. Clique em 'Create policie'

Policy name	Type	Attachments	Description
AdministratorAccess	Job function	2	Provides full access to AWS services and resources.
AlexaForBusinessDeviceSetup	AWS managed	0	Provide device setup access to AlexaForBusiness services
AlexaForBusinessFullAccess	AWS managed	0	Grants full access to AlexaForBusiness resources and access to related

5. Na aba 'JSON' preencha com o seguinte código, trocando 'sqs-arn' pelo arn da fila que criou. E clique em 'Review Policy'

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Resource": [
      "sqs-arn"
    ],
    "Action": [
      "sqs:SendMessage",
      "sqs:ReceiveMessage"
    ]
  }
]
```

6. Nomeie a Policy e clique em 'Create policy'

Review policy

Name*

Use alphanumeric and '+=, @-_' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.

Summary

Service ▼

Access level

Resource

Request condition

Allow (1 of 139 services) [Show remaining 138](#)

[SQS](#)

Limited: Read, Write

QueueName | string like | demoqueue

None

7. Agora vamos criar a role para o apigateway, clique na opção 'Roles' no meu esquerdo

8. Clique em 'Create role'

9. Selecione 'Api Gateway' em clique em 'Next: Permissions'

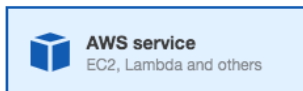
Create role

1

2

3

Select type of trusted entity



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

Choose the service that will use this role

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

API Gateway	Config	Elastic Beanstalk	Lambda	SNS
AppSync	DMS	Elastic Container Service	Lex	SWF
Application Auto Scaling	Data Pipeline	Elastic Transcoder	Machine Learning	SageMaker
Auto Scaling	DeepLens	ElasticLoadBalancing	MediaConvert	Service Catalog
Batch	Directory Service	Glue	OpsWorks	Step Functions
CloudFormation	DynamoDB	Greengrass	RDS	Storage Gateway
CloudHSM	EC2	GuardDuty	Redshift	
CloudWatch Events	EC2 - Fleet	Inspector	Rekognition	
CodeBuild	EMR	IoT	S3	
CodeDeploy	ElastiCache	Kinesis	SMS	

10. clique em 'Next: Review'

11. De o nome 'demoqueue-apigateway-proxy' e clique em 'create'

12. Clique na role que acabou de criar

13. Clique em 'Attach policy'

[Roles](#) > [demoqueue-apigateway-proxy](#)

Summary

Role ARN	arn:aws:iam::799348652687:role/demoqueue-apigateway-proxy
Role description	Allows API Gateway to push logs to CloudWatch Logs. Edit
Instance Profile ARNs	Edit
Path	/
Creation time	2018-05-19 18:30 UTC-0300
Maximum CLI/API session duration	1 hour (3,600 seconds) Edit

[Permissions](#) [Trust relationships](#) [Access Advisor](#) [Revoke sessions](#)

[Attach policy](#) Attached policies: 1

Policy name	Policy type
AmazonAPIGatewayPushToCloudWatchLogs	AWS managed policy

14. Procure pela policy que criou, selecione, e clique em 'Attach policy'

Attach policy

Filter: [Policy type](#)

Policy name	Type
<input type="checkbox"/> demoqueue-apigateway-proxy	Customer managed

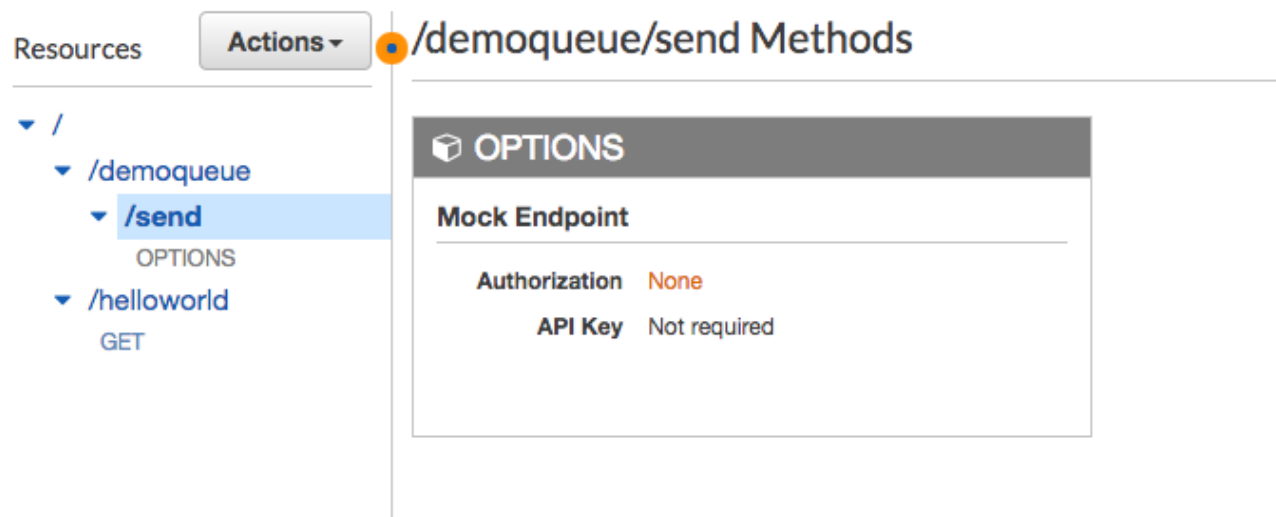
15. A role tem que ficar como na imagem



Criando os recursos no Api Gateway

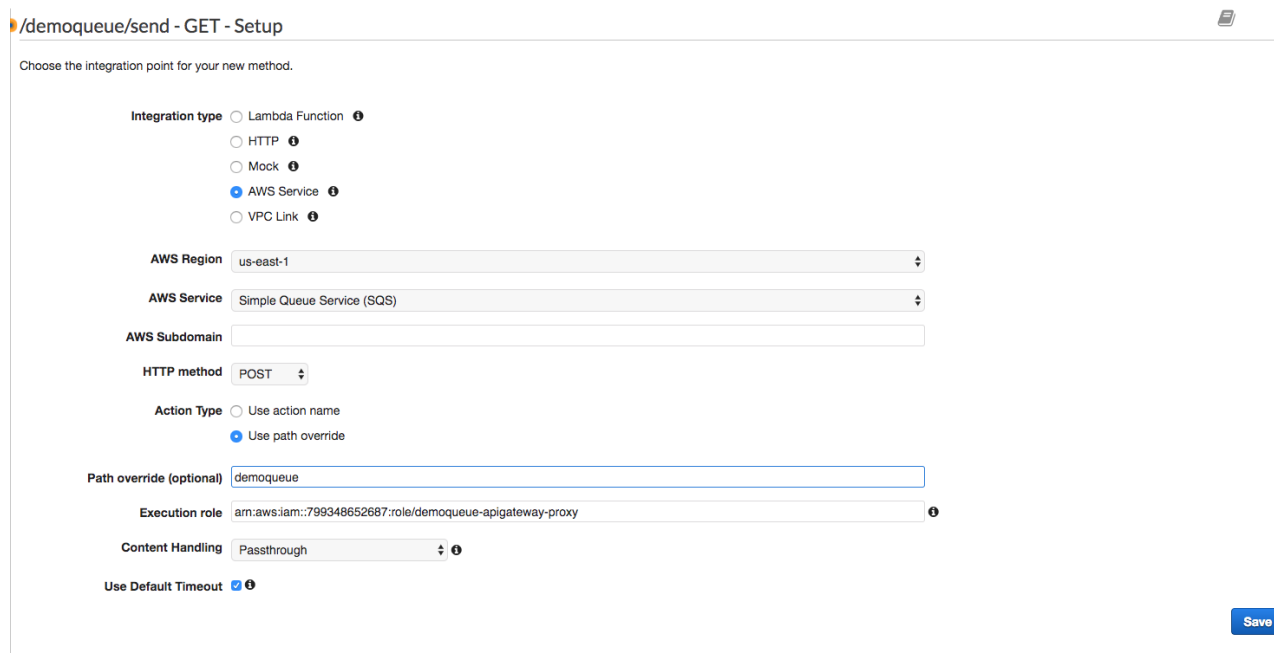
16. Crie um recurso chamado 'demoqueue'

17. No recurso 'demoqueue' crie outro chamado 'send'

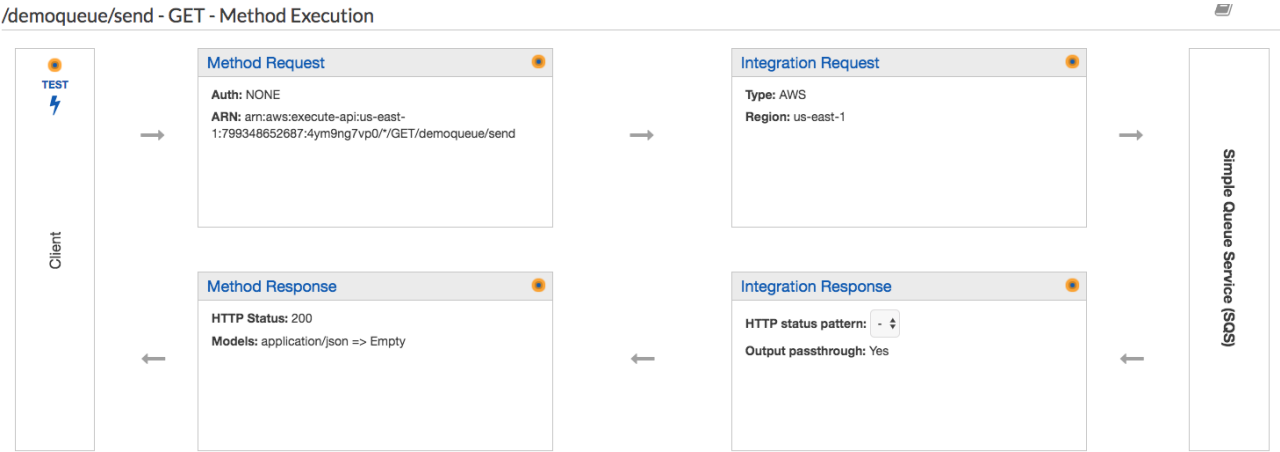


18. No recurso 'send' crie um método do tipo 'GET'

19. Preencha com os dados como a imagem, não esquece de pegar o arn da role que criou, depois clique em 'Save'



20. Clique em 'Method Request'



21. Deixe as query Strings como na imagem, salve, e volte a tela a anterior

← Method Execution /demoqueue/send - GET - Method Request

Provide information about this method's authorization settings and the parameters it can receive.



Settings


Authorization NONE ⓘ

Request Validator NONE ⓘ

API Key Required false ⓘ

▼ URL Query String Parameters ⓘ

Name	Required ⚠	Caching	
MessageBody	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 


 [Add query string](#)

22. Clique me 'Integration Request'





23. Deixe o Query String como na imagem abaixo, salve e volte a tela anterior


▼ URL Path Parameters

Name	Mapped from ⓘ	Caching	
No path parameters			

 [Add path](#)

▼ URL Query String Parameters

Name	Mapped from ⓘ	Caching	
Action	'SendMessage'	<input type="checkbox"/>	 
MessageBody	method.request.querystring.MessageBody	<input type="checkbox"/>	 

 [Add query string](#)

24. Faça o deploy da aplicação o stage já criado.

25. Teste sua api, quando chamar o path /demoqueue/send?MessageBody=SuaMensagem, você deve receber código 200 de resposta juntamente com um json. Se olhar em sua fila sqs verá que tem um item como na imagem.

Filter by Prefix:

Name	Queue Type	Content-Based Deduplication	Messages Available	Messages in Flight
demoqueue	Standard	N/A	1	0