AWS API Gateway

2024-03

Renato Matos
renato.matos79@gmail.com

Rest API

A Representational State Transfer (REST) API **is a type of web service architecture that allows different systems to communicate with each other over the internet** broken down into key components:

Resources: In a REST API, resources represent the data entities or objects that clients can interact with. Each resource is identified by a unique Uniform Resource Identifier (URI), commonly known as an endpoint. For example, `/users` might represent a collection of user resources, and `/users/{id}` might represent a specific user identified by their ID.

HTTP Methods: RESTful APIs use standard HTTP methods to perform operations on resources. The most commonly used HTTP methods in REST APIs are:

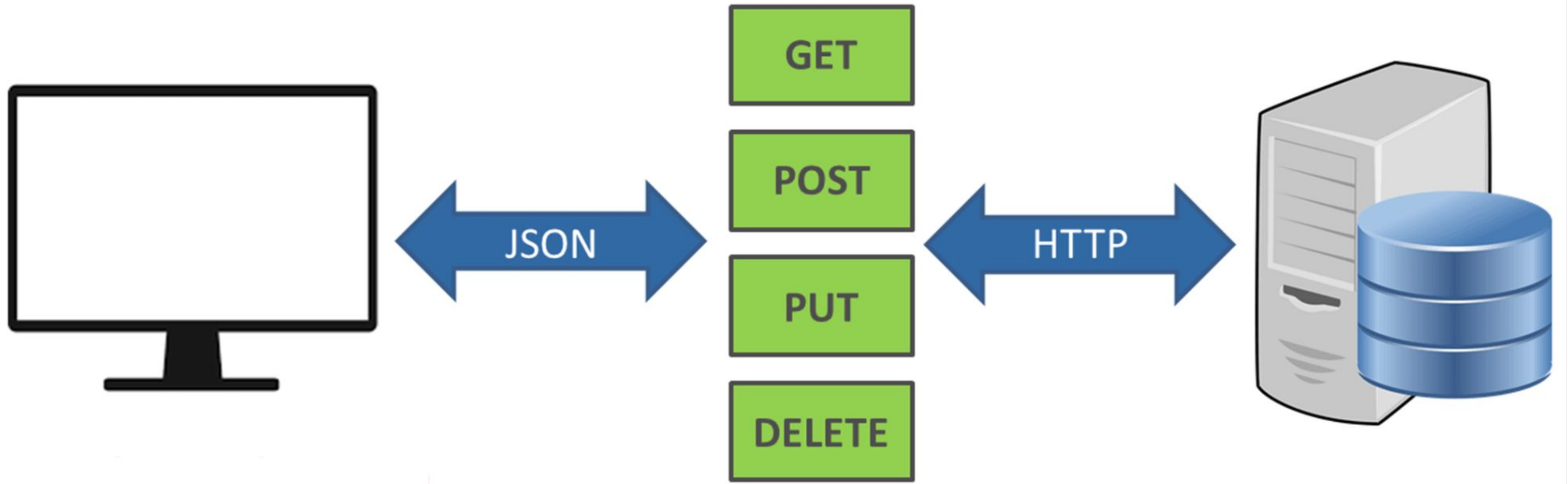
- GET: Retrieve a representation of a resource.
- POST: Create a new resource.
- PUT: Update an existing resource.
- DELETE: Delete a resource.
- PATCH: Partially update a resource.

These HTTP methods map to CRUD (Create, Read, Update, Delete) operations on resources.

Statelessness: REST APIs are stateless, meaning that each request from a client to the server must contain all the information necessary to understand and process the request. The server does not maintain any client state between requests.

Representation: Resources in a REST API are typically represented using different media types such as JSON (JavaScript Object Notation) or XML (eXtensible Markup Language). Clients and servers communicate by exchanging representations of resources, which contain data in a structured format.

Rest API



Client sends a **request** HTTP methods Server sends a **response**

API Gateway

Build and Deploy APIs Quickly: PI Gateway allows you to quickly create RESTful APIs or WebSocket APIs without the need for infrastructure setup or management

Scalability and High Availability: API Gateway automatically scales to handle any amount of traffic, ensuring high availability and reliability of your APIs

Integration with AWS Services: AWS Lambda, AWS S3, AWS DynamoDB, AWS Cognito, and more

Security and Access Control: API Gateway provides features for securing your APIs, including authentication, authorization, and encryption. You can use AWS Identity and Access Management (IAM), OAuth 2.0, and Amazon Cognito to control access to your APIs and protect sensitive data

Monitoring and Logging: API Gateway offers built-in monitoring and logging capabilities that allow you to track API usage, performance metrics, and errors in real-time

Cost-effective Pricing Model: API Gateway offers a pay-as-you-go pricing model, where you only pay for the API requests and data transfer out of the service.

Support for API Versioning and Lifecycle Management: API Gateway supports API versioning and lifecycle management, allowing you to easily manage multiple versions of your APIs, roll back changes, and deprecate outdated versions without disrupting existing clients.

Developer-Friendly Features: API Gateway provides features such as API documentation, SDK generation, and testing tools that make it easier for developers to consume and integrate with your APIs.

https://docs.aws.amazon.com/apigateway/latest/developerguide/limits.html

API Gateway

HTTP API:

- HTTP API is a new type of API Gateway introduced by AWS, designed to offer low-latency, low-cost API integrations for HTTP-based APIs.
- It provides a lightweight and cost-effective solution for building APIs with simple use cases.
- HTTP APIs support core API Gateway features such as request/response transformation, CORS (Cross-Origin Resource Sharing), JWT (JSON Web Tokens) authorizers, and rate limiting.
- HTTP APIs do not support all features available in REST APIs, but they are optimized for serverless architectures and simple API use cases.

REST API:

- REST API is the traditional API Gateway offered by AWS, which supports a broader range of features and customization options compared to HTTP APIs.
- It follows the principles of Representational State Transfer (REST) architecture, allowing clients to interact with resources using standard HTTP methods and representations.
- REST APIs support more advanced features such as custom domain names, API keys, IAM (Identity and Access Management) authorization, request/response validation, and stage variables.
- REST APIs are suitable for building complex APIs with multiple endpoints, advanced security requirements, and integration with various AWS services.

REST API Private:

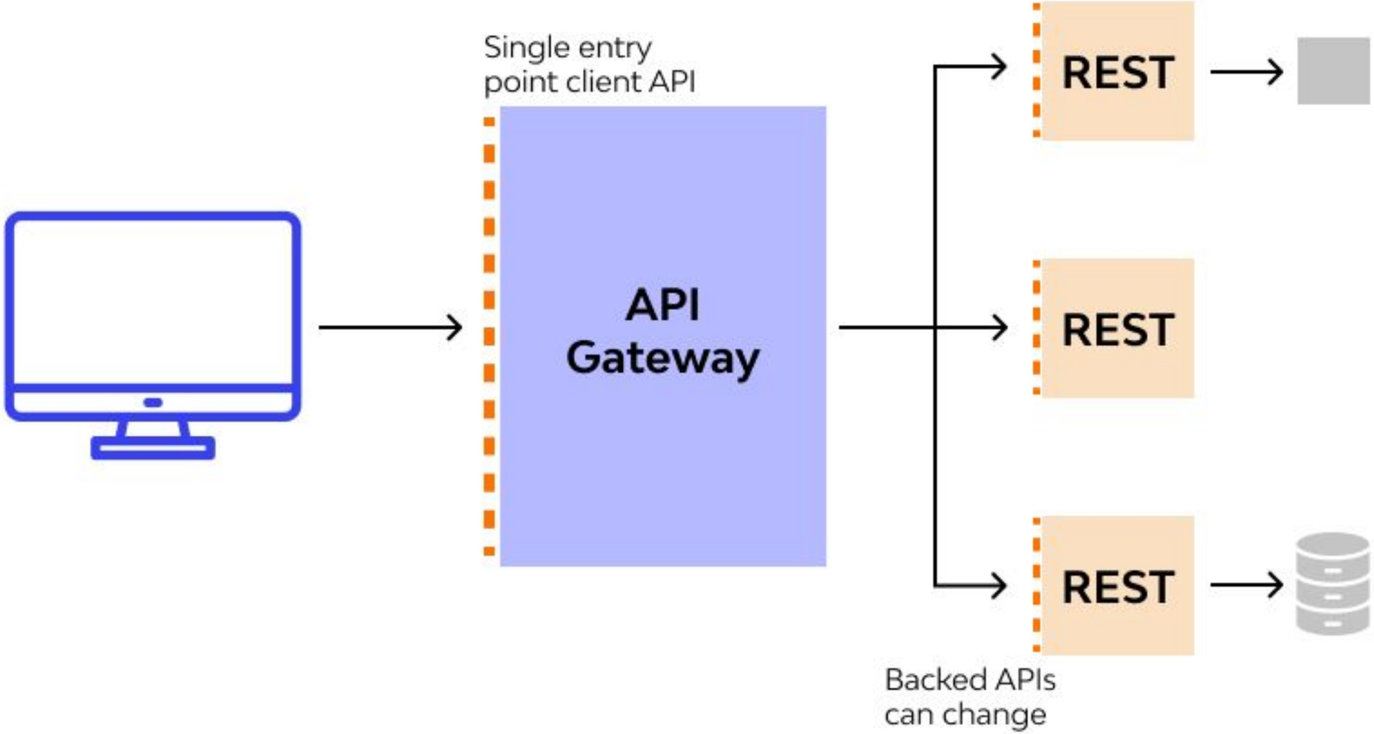
- REST API Private is a variation of REST API that allows you to expose your API privately within a VPC (Virtual Private Cloud) without exposing it to the public internet.
- It is suitable for scenarios where you want to restrict access to your API to specific VPC resources or securely integrate with internal systems.
- REST API Private supports all the features available in REST APIs, including custom domain names, authorization mechanisms, and request/response handling.
- It provides enhanced security and isolation by leveraging private integration with AWS resources or using AWS PrivateLink to access resources within a VPC.

WebSocket APIs:

- WebSocket APIs are used for real-time, bidirectional communication between clients and servers over a single, long-lived connection.
- Unlike HTTP APIs and REST APIs, which use request/response model, WebSocket APIs enable full-duplex communication, allowing both clients and servers to send messages asynchronously.
- WebSocket APIs support features such as route management, message transformation, authorization, and integration with AWS Lambda for serverless processing.
- They are suitable for building applications that require real-time updates, such as chat applications, gaming platforms, or collaborative editing tools.

https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api-vs-rest.html

API Gateway


Single entry
point client API
API
Gateway
REST
REST
REST
Backed APIs
can change

API Gateway

{LAB}
Time

API Gateway

01. Building our first API

API Gateway

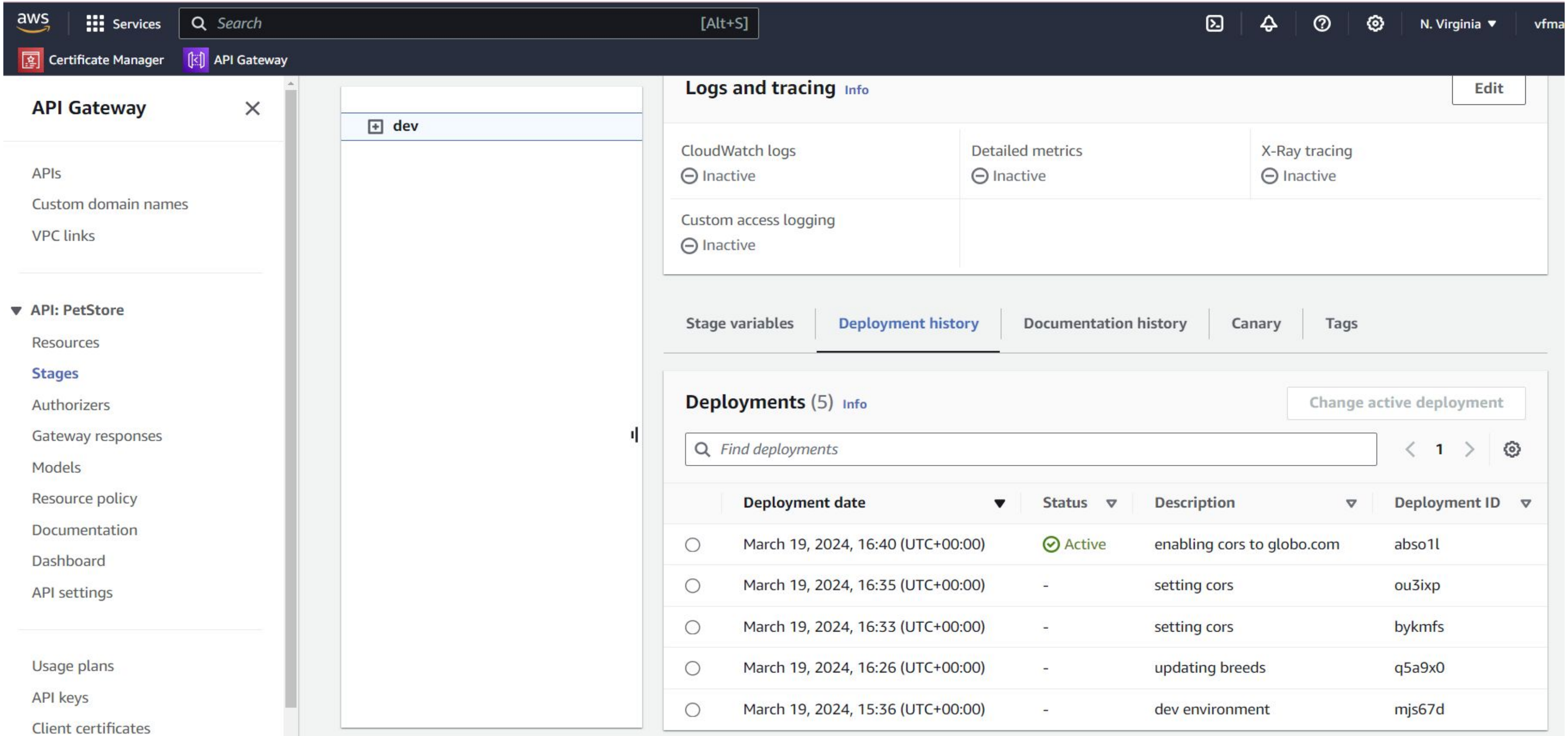
02.Exploring the flow

API Gateway

03.Providing mock data

04.Deploying to DEV env

API Gateway



Logs and tracing Info

CloudWatch logs	Detailed metrics	X-Ray tracing
⊖ Inactive	⊖ Inactive	⊖ Inactive

Custom access logging
⊖ Inactive

Deployments (5) Info

	Deployment date	Status	Description	Deployment ID
○	March 19, 2024, 16:40 (UTC+00:00)	✓ Active	enabling cors to globo.com	abso1l
○	March 19, 2024, 16:35 (UTC+00:00)	-	setting cors	ou3ixp
○	March 19, 2024, 16:33 (UTC+00:00)	-	setting cors	bykmfs
○	March 19, 2024, 16:26 (UTC+00:00)	-	updating breeds	q5a9x0
○	March 19, 2024, 15:36 (UTC+00:00)	-	dev environment	mjs67d

API Gateway

05.Using lambda function to replace the mock content

06.Enabling CORS

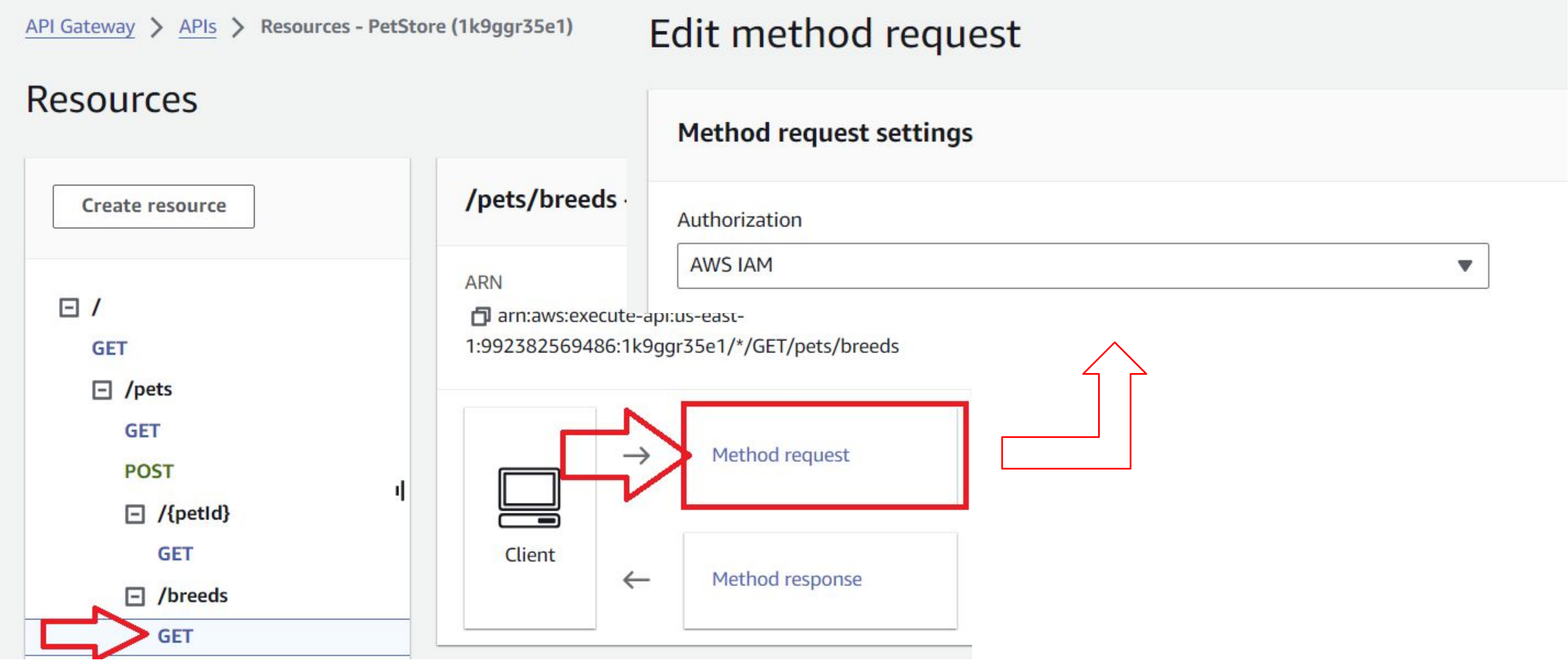
API Gateway

Paste this code below using **console browser** and press **RETURN**, after that run the code typing **main()**

```
function main() {
     console.log("requesting breeds")
     $.ajax({
       url: "https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds",
       success: function(data) {
          console.log(data)
       }
     })
}
```

07.Enabling IAM Authentication

API Gateway



API Gateway > APIs > Resources - PetStore (1k9ggr35e1)

Resources

/pets/breeds -

ARN

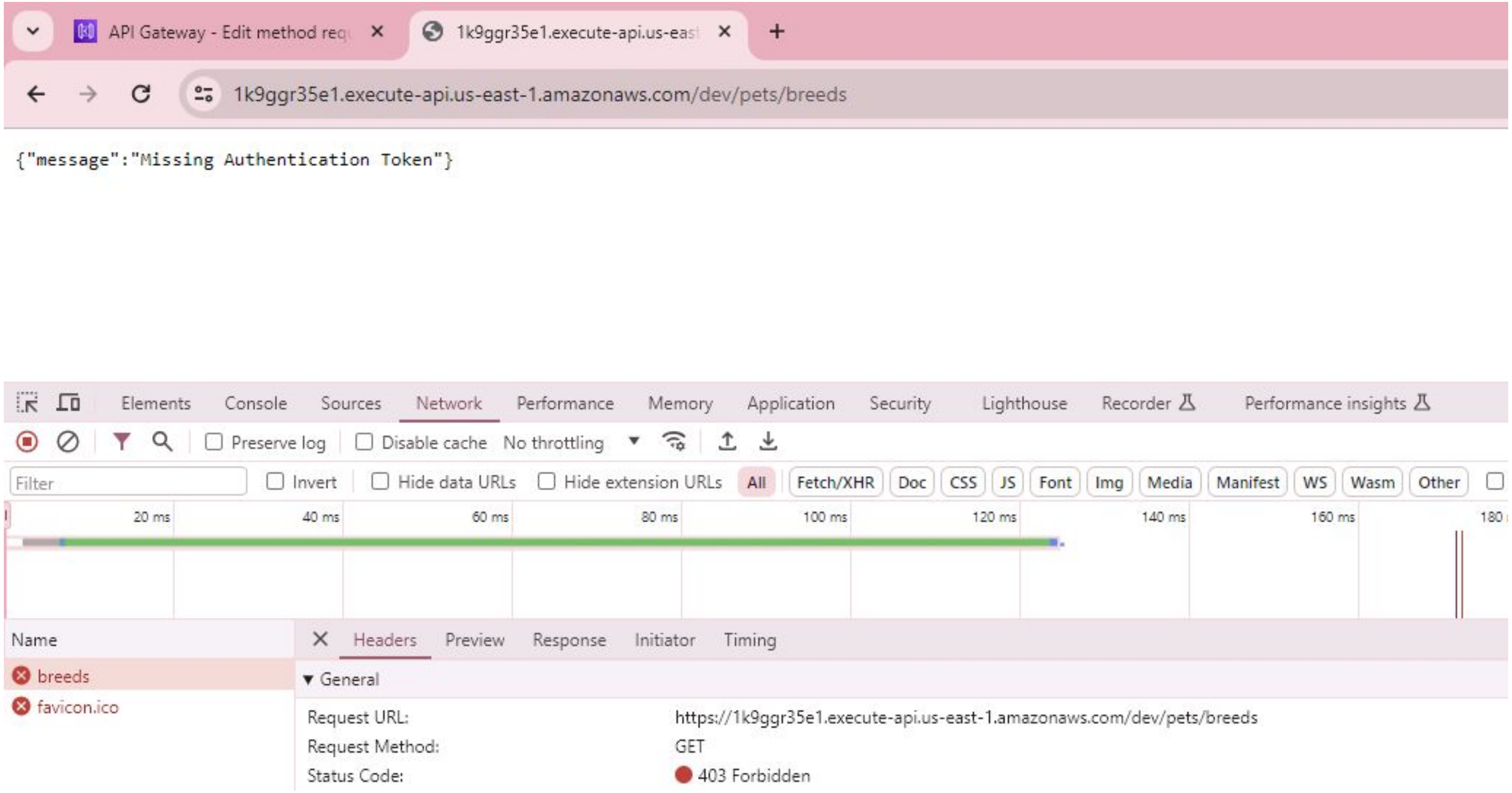
arn:aws:execute-api:us-east-1:992382569486:1k9ggr35e1/*/GET/pets/breeds

Edit method request

Method request settings

Authorization: AWS IAM

API Gateway



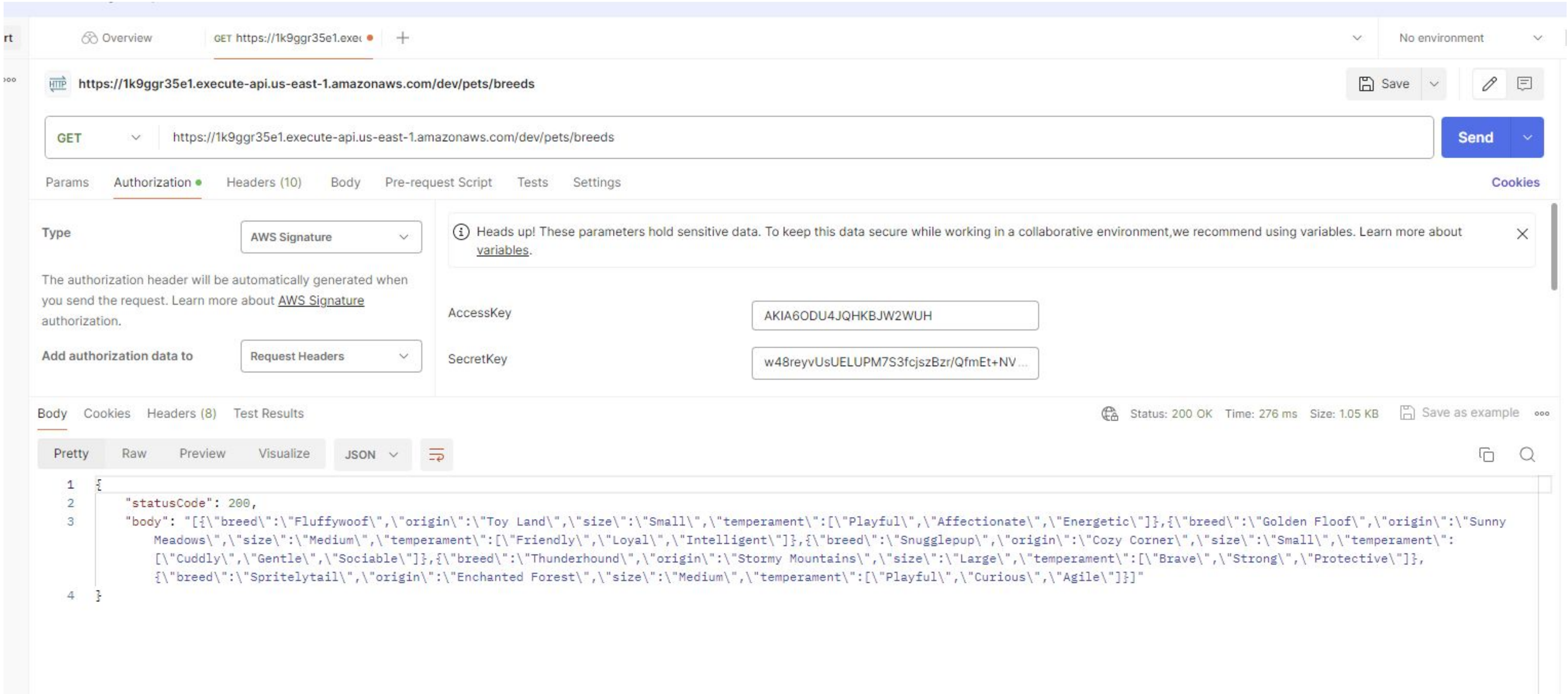
```
{"message":"Missing Authentication Token"}
```

Request URL: https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

Request Method: GET

Status Code: ● 403 Forbidden

API Gateway



https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

GET https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

Type: AWS Signature

The authorization header will be automatically generated when you send the request. Learn more about AWS Signature authorization.

Add authorization data to: Request Headers

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment,we recommend using variables. Learn more about variables.

AccessKey: AKIA6ODU4JQHKBJW2WUH

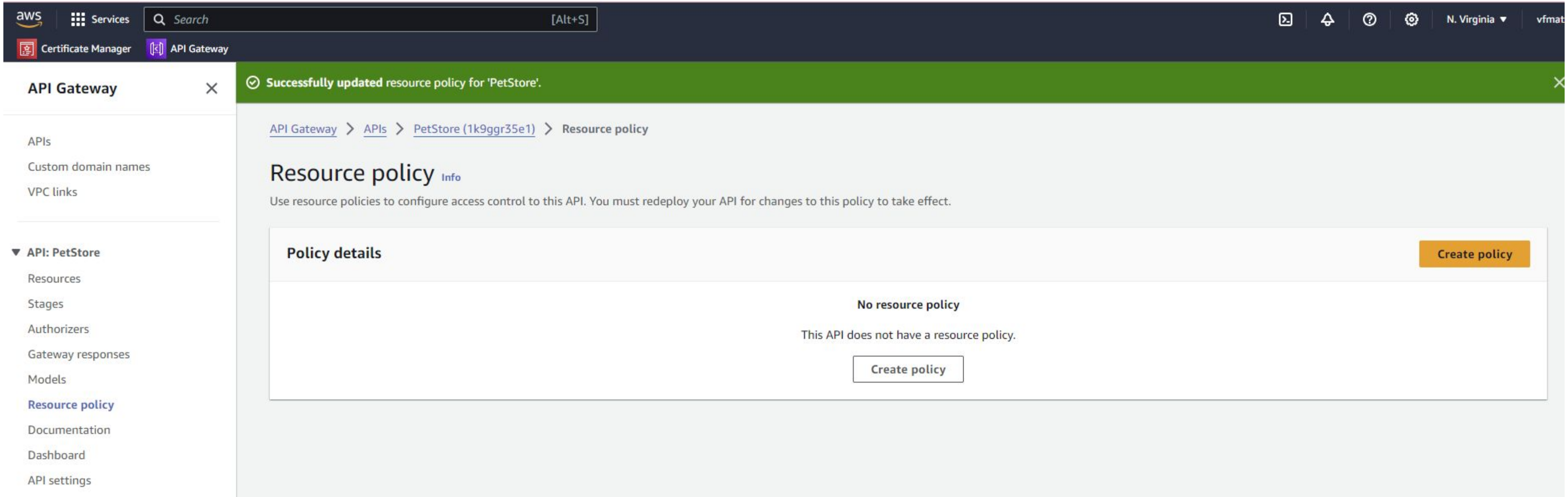
SecretKey: w48reyvUsUELUPM7S3fcjszBzr/QfmEt+NV ...

Status: 200 OK Time: 276 ms Size: 1.05 KB

```
{
    "statusCode": 200,
    "body": "[{\"breed\":\"Fluffywoof\",\"origin\":\"Toy Land\",\"size\":\"Small\",\"temperament\":[\"Playful\",\"Affectionate\",\"Energetic\"]},{\"breed\":\"Golden Floof\",\"origin\":\"Sunny
        Meadows\",\"size\":\"Medium\",\"temperament\":[\"Friendly\",\"Loyal\",\"Intelligent\"]},{\"breed\":\"Snugglepup\",\"origin\":\"Cozy Corner\",\"size\":\"Small\",\"temperament\":
        [\"Cuddly\",\"Gentle\",\"Sociable\"]},{\"breed\":\"Thunderhound\",\"origin\":\"Stormy Mountains\",\"size\":\"Large\",\"temperament\":[\"Brave\",\"Strong\",\"Protective\"]},
        {\"breed\":\"Spritelytail\",\"origin\":\"Enchanted Forest\",\"size\":\"Medium\",\"temperament\":[\"Playful\",\"Curious\",\"Agile\"]}]"
}
```

08.Resource Policy

API Gateway



Successfully updated resource policy for 'PetStore'.

Resource policy

Use resource policies to configure access control to this API. You must redeploy your API for changes to this policy to take effect.

Policy details

No resource policy

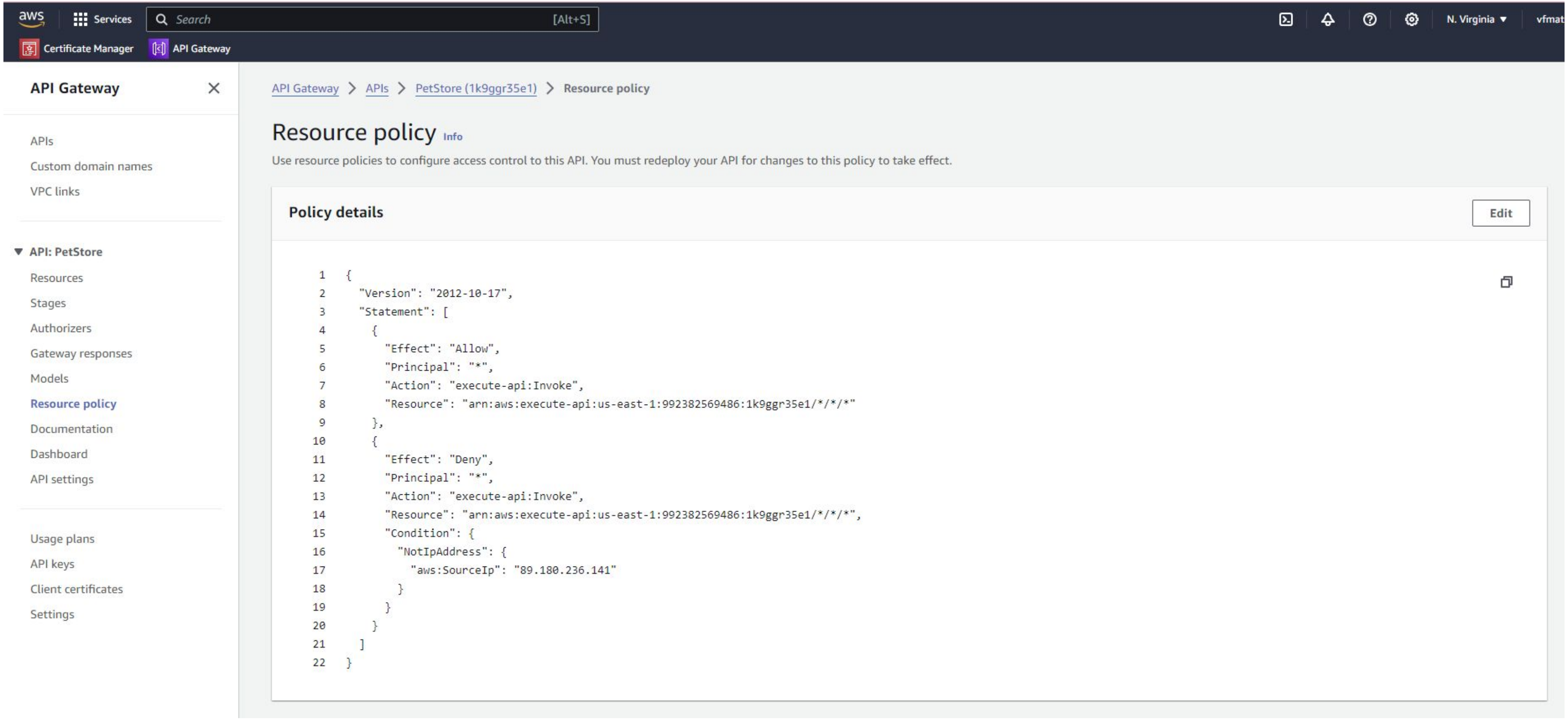
This API does not have a resource policy.

API Gateway

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "arn:aws:execute-api:us-east-1:992382569486:1k9ggr35e1/*/*/*"
    },
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "arn:aws:execute-api:us-east-1:992382569486:1k9ggr35e1/*/*/*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "89.180.236.241"
        }
      }
    }
  ]
}
```

https://repost.aws/knowledge-center/api-gateway-resource-policy-access

API Gateway



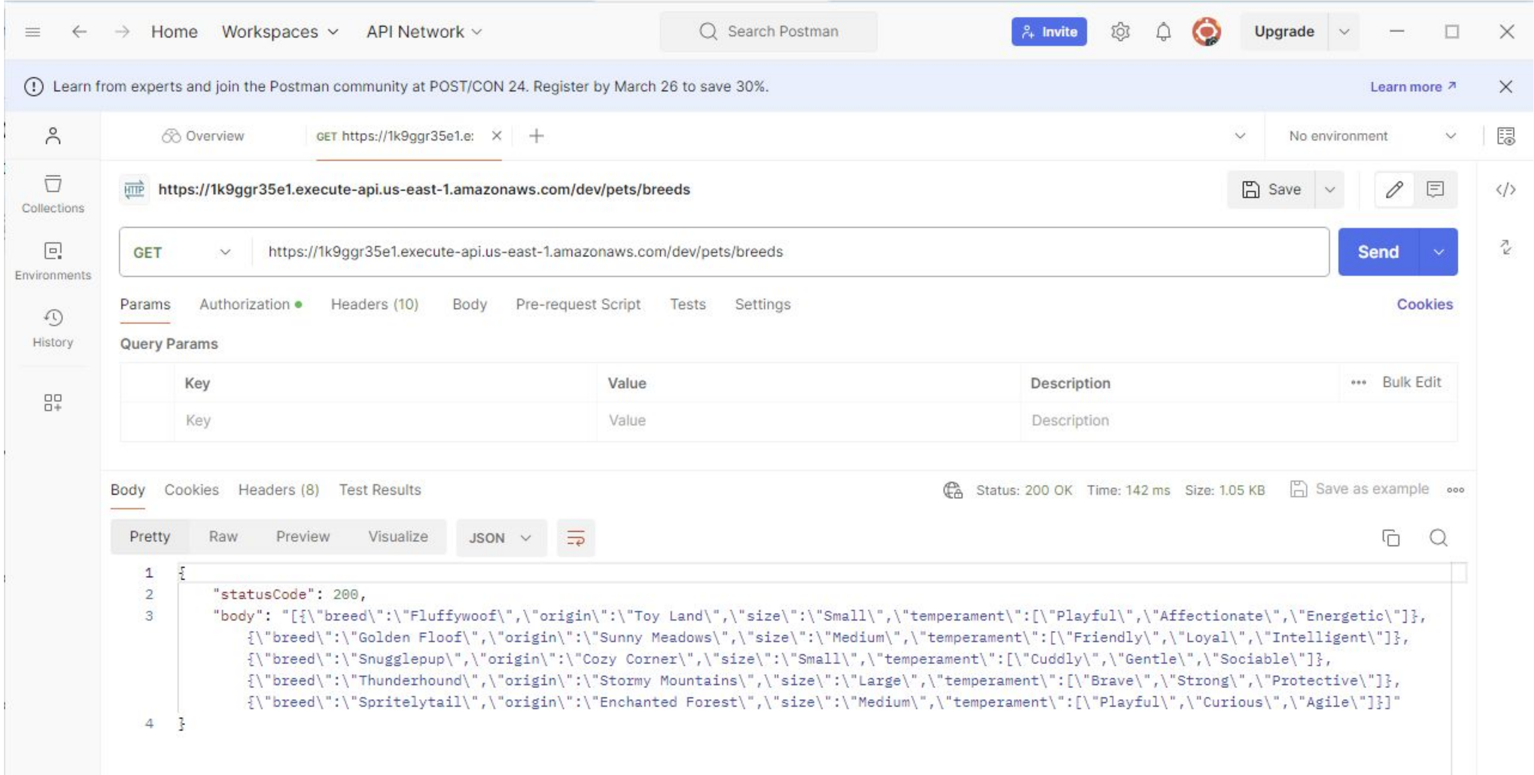
Resource policy

Use resource policies to configure access control to this API. You must redeploy your API for changes to this policy to take effect.

Policy details

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "arn:aws:execute-api:us-east-1:992382569486:1k9ggr35e1/*/*/*"
    },
    {
      "Effect": "Deny",
      "Principal": "*",
      "Action": "execute-api:Invoke",
      "Resource": "arn:aws:execute-api:us-east-1:992382569486:1k9ggr35e1/*/*/*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "89.180.236.141"
        }
      }
    }
  ]
}
```

API Gateway



https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

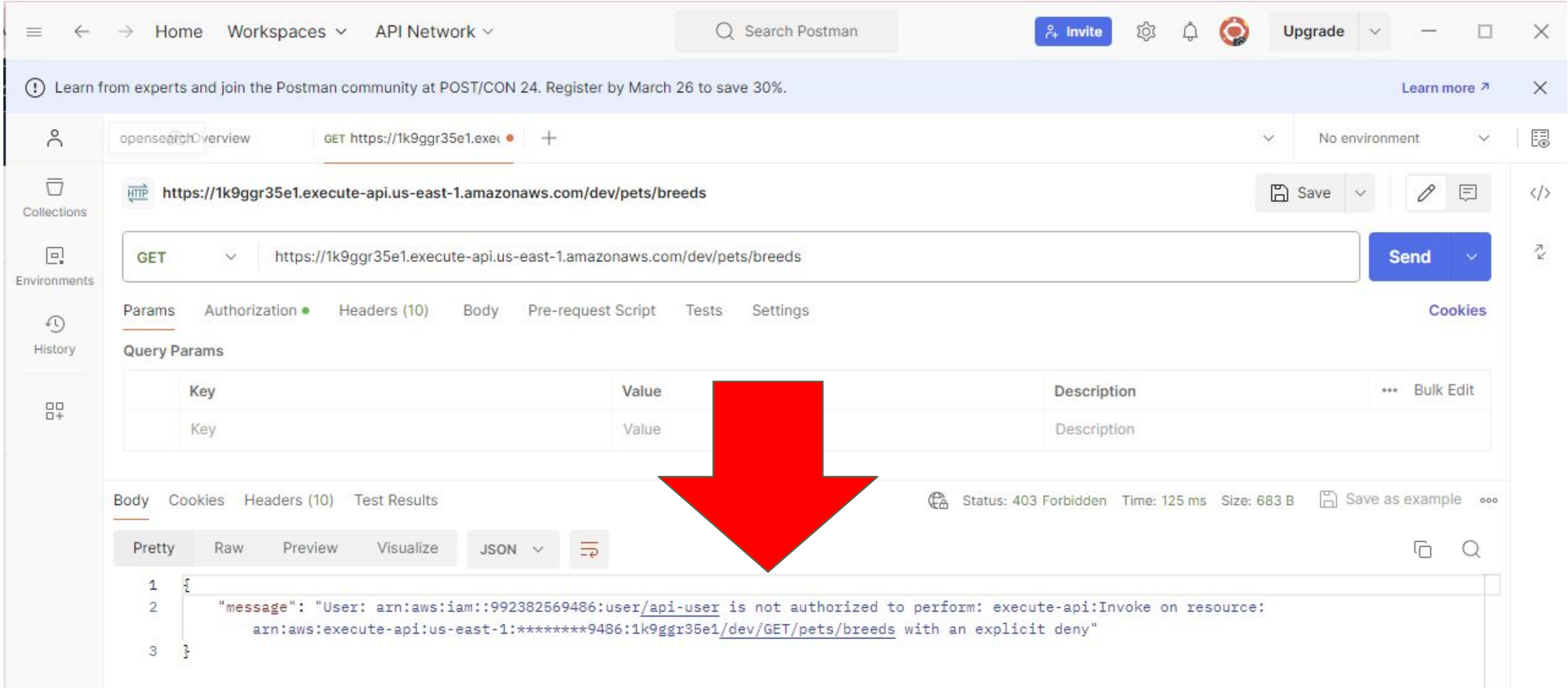
GET https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

Key	Value	Description
Key	Value	Description

Status: 200 OK Time: 142 ms Size: 1.05 KB

```
{
    "statusCode": 200,
    "body": "[{\"breed\":\"Fluffywoof\",\"origin\":\"Toy Land\",\"size\":\"Small\",\"temperament\":[\"Playful\",\"Affectionate\",\"Energetic\"]},
        {\"breed\":\"Golden Floof\",\"origin\":\"Sunny Meadows\",\"size\":\"Medium\",\"temperament\":[\"Friendly\",\"Loyal\",\"Intelligent\"]},
        {\"breed\":\"Snugglepup\",\"origin\":\"Cozy Corner\",\"size\":\"Small\",\"temperament\":[\"Cuddly\",\"Gentle\",\"Sociable\"]},
        {\"breed\":\"Thunderhound\",\"origin\":\"Stormy Mountains\",\"size\":\"Large\",\"temperament\":[\"Brave\",\"Strong\",\"Protective\"]},
        {\"breed\":\"Spritelytail\",\"origin\":\"Enchanted Forest\",\"size\":\"Medium\",\"temperament\":[\"Playful\",\"Curious\",\"Agile\"]}]"
}
```

API Gateway



Learn from experts and join the Postman community at POST/CON 24. Register by March 26 to save 30%. Learn more ↗

GET https://1k9ggr35e1.exe

https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

GET https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

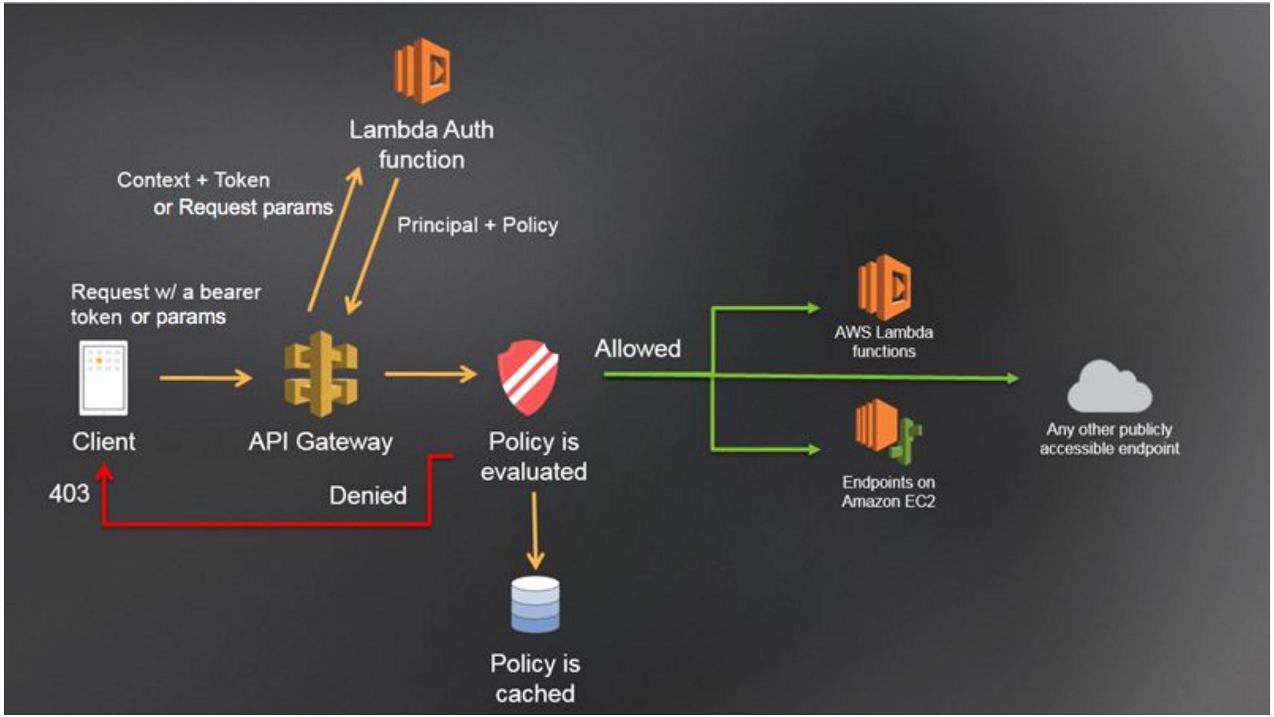
Status: 403 Forbidden Time: 125 ms Size: 683 B

```
{
    "message": "User: arn:aws:iam::992382569486:user/api-user is not authorized to perform: execute-api:Invoke on resource:
        arn:aws:execute-api:us-east-1:********9486:1k9ggr35e1/dev/GET/pets/breeds with an explicit deny"
}
```

09.Lambda Authorizers

API Gateway

A *Lambda authorizer* (formerly known as a *custom authorizer*) **is an API Gateway feature** that uses a **Lambda function** to control access to your API.



https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-use-lambda-authorizer.html

API Gateway

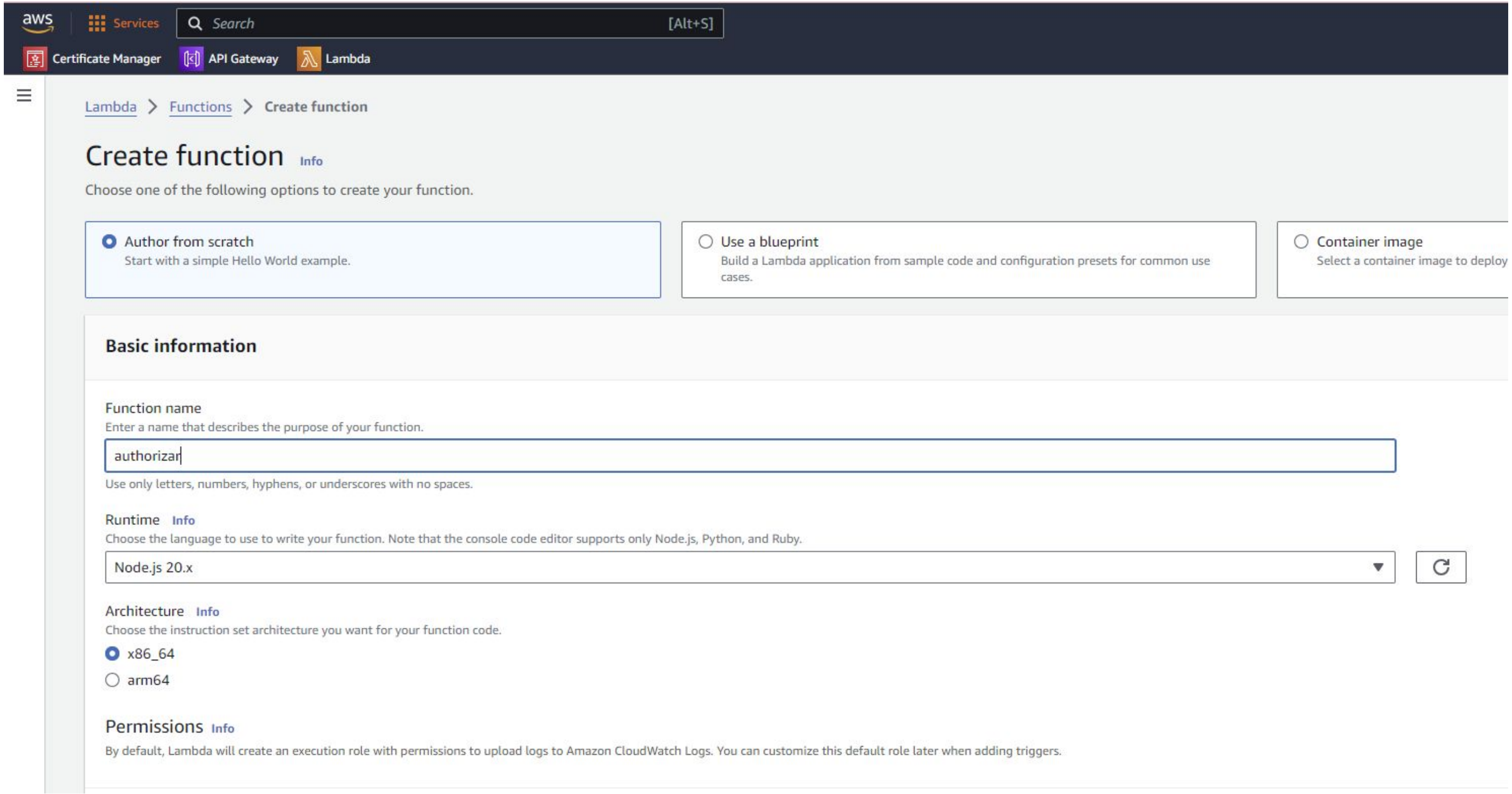
Token-Based Authorizer:

- Token-based authorizers are used when clients provide tokens (such as JWT - JSON Web Tokens) as part of the request, typically in the Authorization header.
- The Lambda authorizer extracts the token from the request, validates its authenticity (e.g., verifies the signature, expiration), and optionally checks the token against an external identity provider (e.g., AWS Cognito, OAuth provider).
- If the token is valid and authorized, the authorizer returns an IAM policy allowing access to the requested resource.
- Token-based authorizers are **commonly used for authentication purposes, where clients provide tokens to prove their identity and access rights**.

Request-Based Authorizer:

- Request-based authorizers are used when clients provide additional information in the request payload, headers, or query parameters to determine access control.
- The Lambda authorizer receives the entire request and examines its content to determine whether the requester is authorized to access the resource.
- Based on the information provided in the request, the authorizer constructs an IAM policy allowing or denying access to the requested resource.
- Request-based authorizers are more flexible and can accommodate various authentication and authorization mechanisms beyond token-based authentication. **For example, they can perform custom logic based on request attributes, user attributes, or external data sources**.

API Gateway



Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy

Basic information

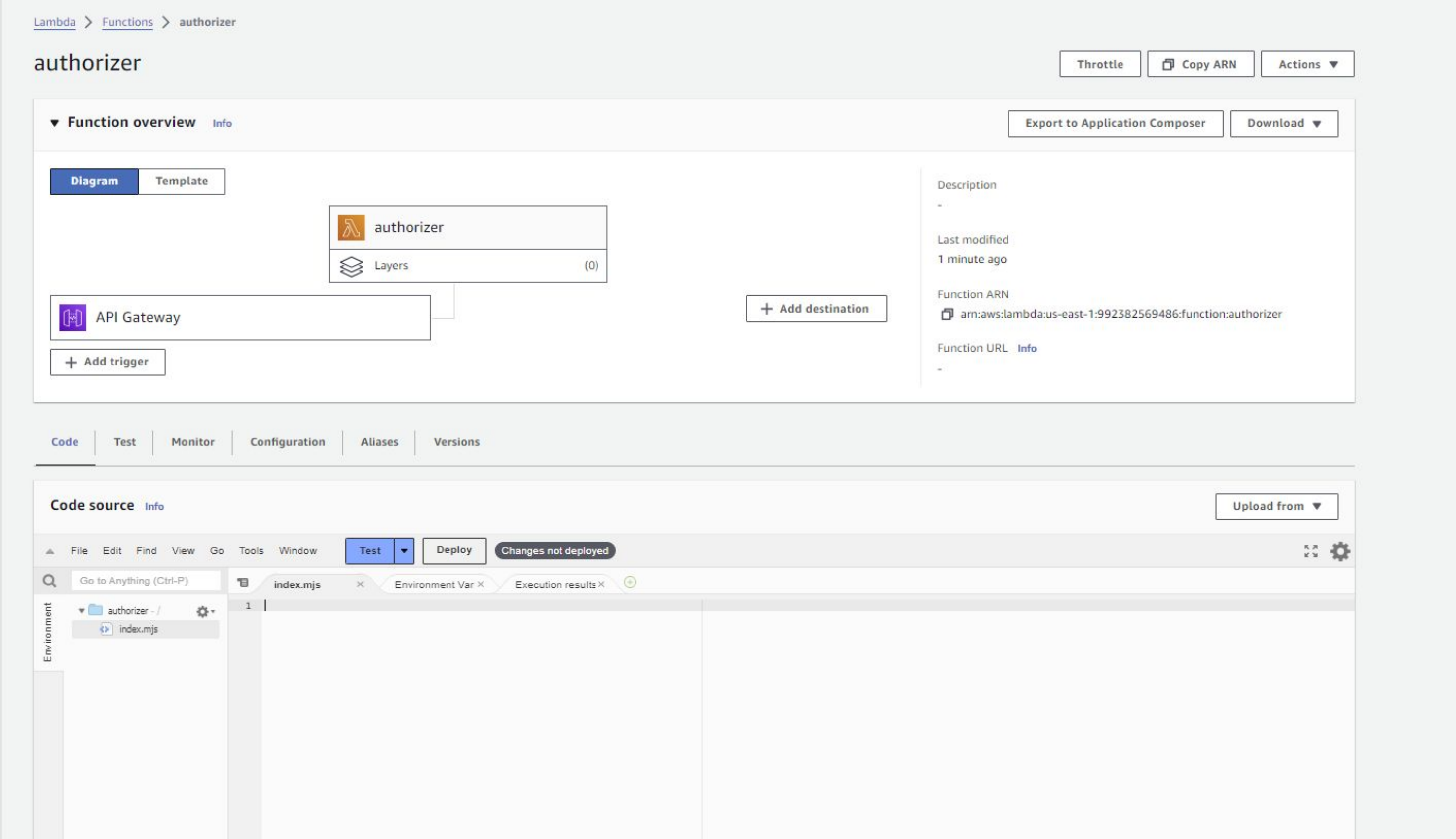
Function name
Enter a name that describes the purpose of your function.
authorizar
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Node.js 20.x

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

API Gateway



Lambda > Functions > authorizer

authorizer

Function overview Info

Description
-

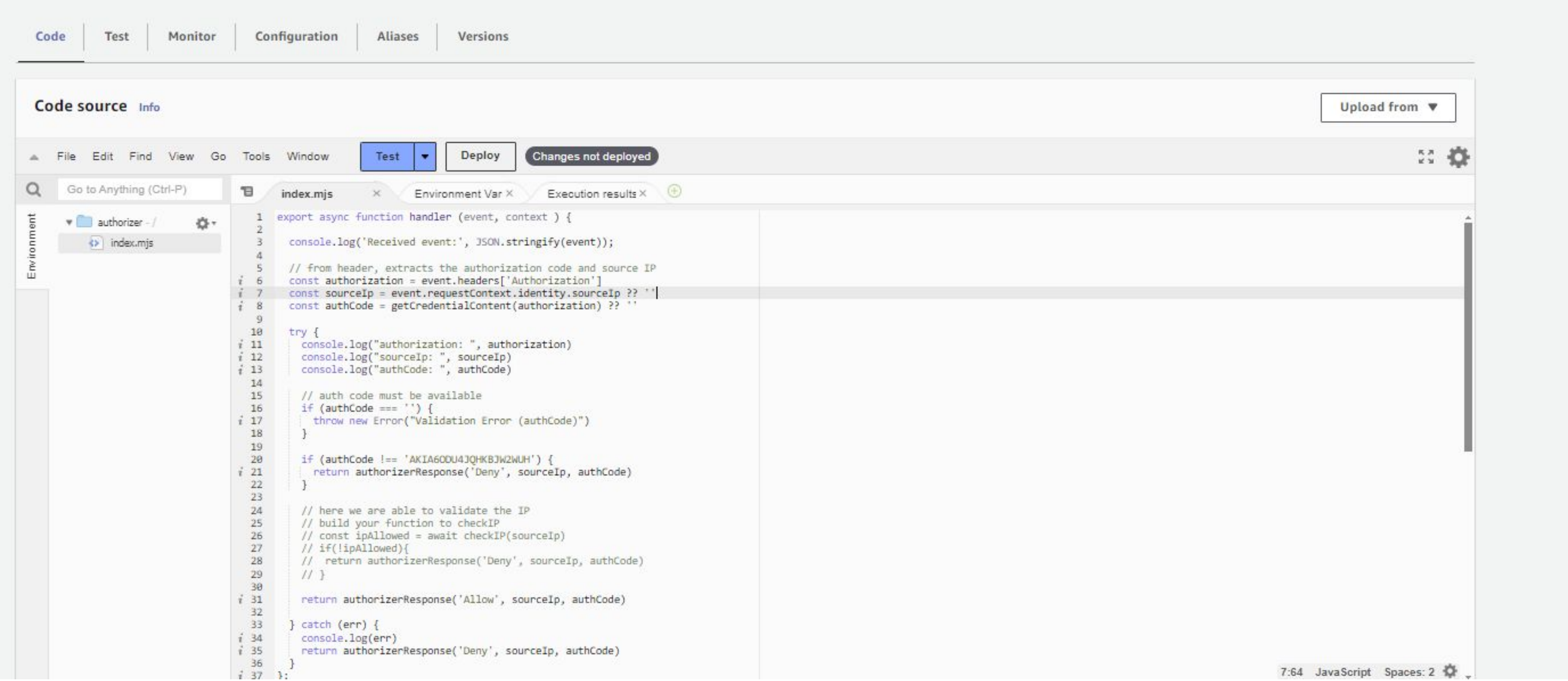
Last modified
1 minute ago

Function ARN
arn:aws:lambda:us-east-1:992382569486:function:authorizer

Function URL Info
-

Code source Info

API Gateway



```
export async function handler (event, context ) {

  console.log('Received event:', JSON.stringify(event));

  // from header, extracts the authorization code and source IP
  const authorization = event.headers['Authorization']
  const sourceIp = event.requestContext.identity.sourceIp ?? ''
  const authCode = getCredentialContent(authorization) ?? ''

  try {
    console.log("authorization: ", authorization)
    console.log("sourceIp: ", sourceIp)
    console.log("authCode: ", authCode)

    // auth code must be available
    if (authCode === '') {
      throw new Error("Validation Error (authCode)")
    }

    if (authCode !== 'AKIA6ODU4JQHKBJW2WUH') {
      return authorizerResponse('Deny', sourceIp, authCode)
    }

    // here we are able to validate the IP
    // build your function to checkIP
    // const ipAllowed = await checkIP(sourceIp)
    // if(!ipAllowed){
    //  return authorizerResponse('Deny', sourceIp, authCode)
    // }

    return authorizerResponse('Allow', sourceIp, authCode)

  } catch (err) {
    console.log(err)
    return authorizerResponse('Deny', sourceIp, authCode)
  }
};
```

API Gateway

```
export async function handler (event, context ) {
  // from header, extracts the authorization code and source IP
  const authorization = event.headers['Authorization']
  const sourceIp = event.requestContext.identity.sourceIp ?? ''
  const authCode = getCredentialContent(authorization) ?? ''
  try {
    // auth code must be available
    if (authCode === '') {
      throw new Error("Validation Error (authCode)")
    }
    // forcing authCode to be a valid user 'AKIA6ODU4JQHKBJW2WUH'
    if (authCode !== 'AKIA6ODU4JQHKBJW2WUH') {
      return authorizerResponse('Deny', sourceIp, authCode)
    }

    return authorizerResponse('Allow', sourceIp, authCode)

  } catch (err) {
    console.log(err)
    return authorizerResponse('Deny', sourceIp, authCode)
  }
};
```

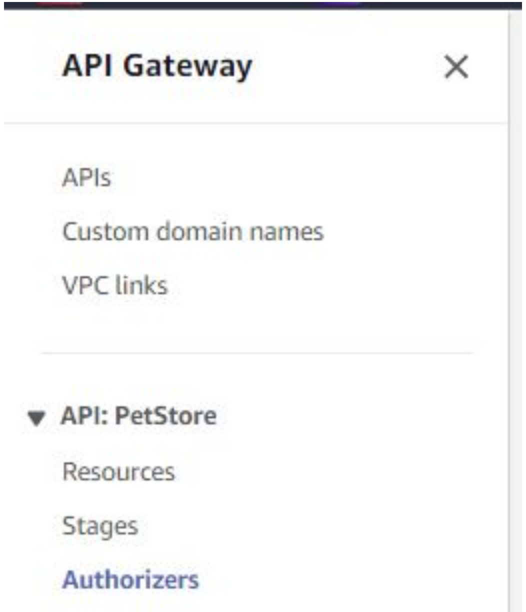
API Gateway

```
// Using REGEX extracts the credential content
function getCredentialContent(header) {
  // Use regular expression to extract the credential content
  const match = header.match(/Credential=([^/]+)/);

  // Check if the match is found
  if (match && match.length > 1) {
    return match[1];
  } else {
    return null
  }
}
```

```
// build the final Policy (Allow)
function authorizerResponse (effect, sourceIp, authCode) {
  const response = {
    principalId: sourceIp,
    context: {
      "sourceIp": sourceIp,
      "authCode": authCode
    },
    policyDocument: {
        Version: '2012-10-17',
        Statement: [
          {
              Action: 'execute-api:Invoke',
              Effect: effect,
              Resource: '*',
          },
        ],
    },
  }
  return response
}
```

API Gateway



API Gateway ✕

APIs

Custom domain names

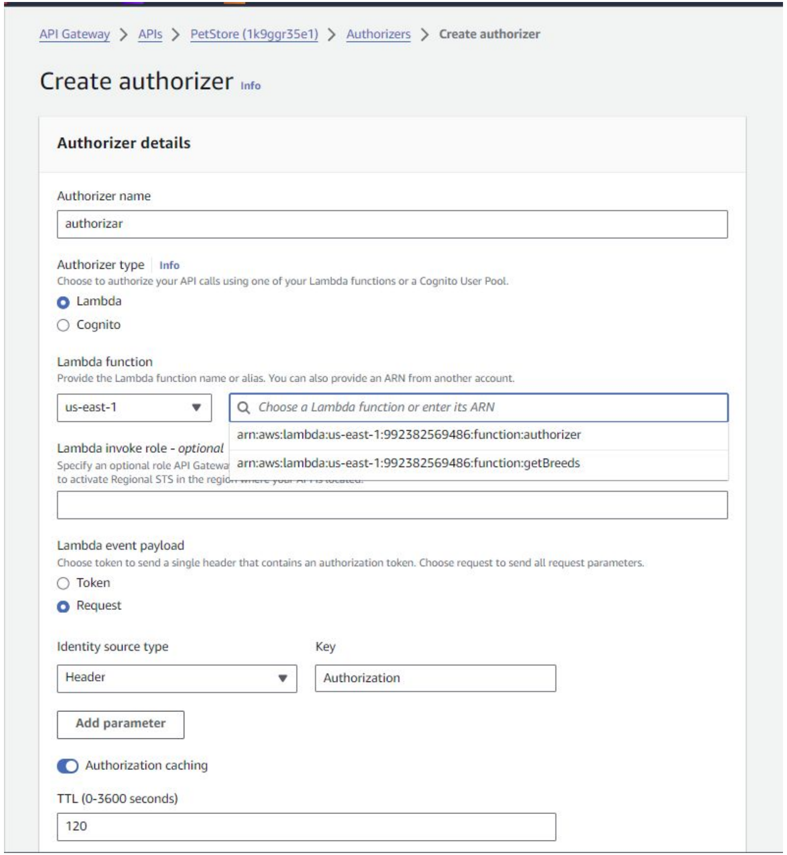
VPC links

▼ API: PetStore

Resources

Stages

Authorizers



API Gateway > APIs > PetStore (1k9ggr35e1) > Authorizers > Create authorizer

Create authorizer Info

Authorizer details

Authorizer name

authorizar

Authorizer type Info

Choose to authorize your API calls using one of your Lambda functions or a Cognito User Pool.

● Lambda

○ Cognito

Lambda function

Provide the Lambda function name or alias. You can also provide an ARN from another account.

us-east-1 ▼ | Choose a Lambda function or enter its ARN

arn:aws:lambda:us-east-1:992382569486:function:authorizer

arn:aws:lambda:us-east-1:992382569486:function:getBreeds

Lambda invoke role - *optional*

Specify an optional role API Gatewa

to activate Regional STS in the regio

Lambda event payload

Choose token to send a single header that contains an authorization token. Choose request to send all request parameters.

○ Token

● Request

Identity source type	Key
Header ▼	Authorization

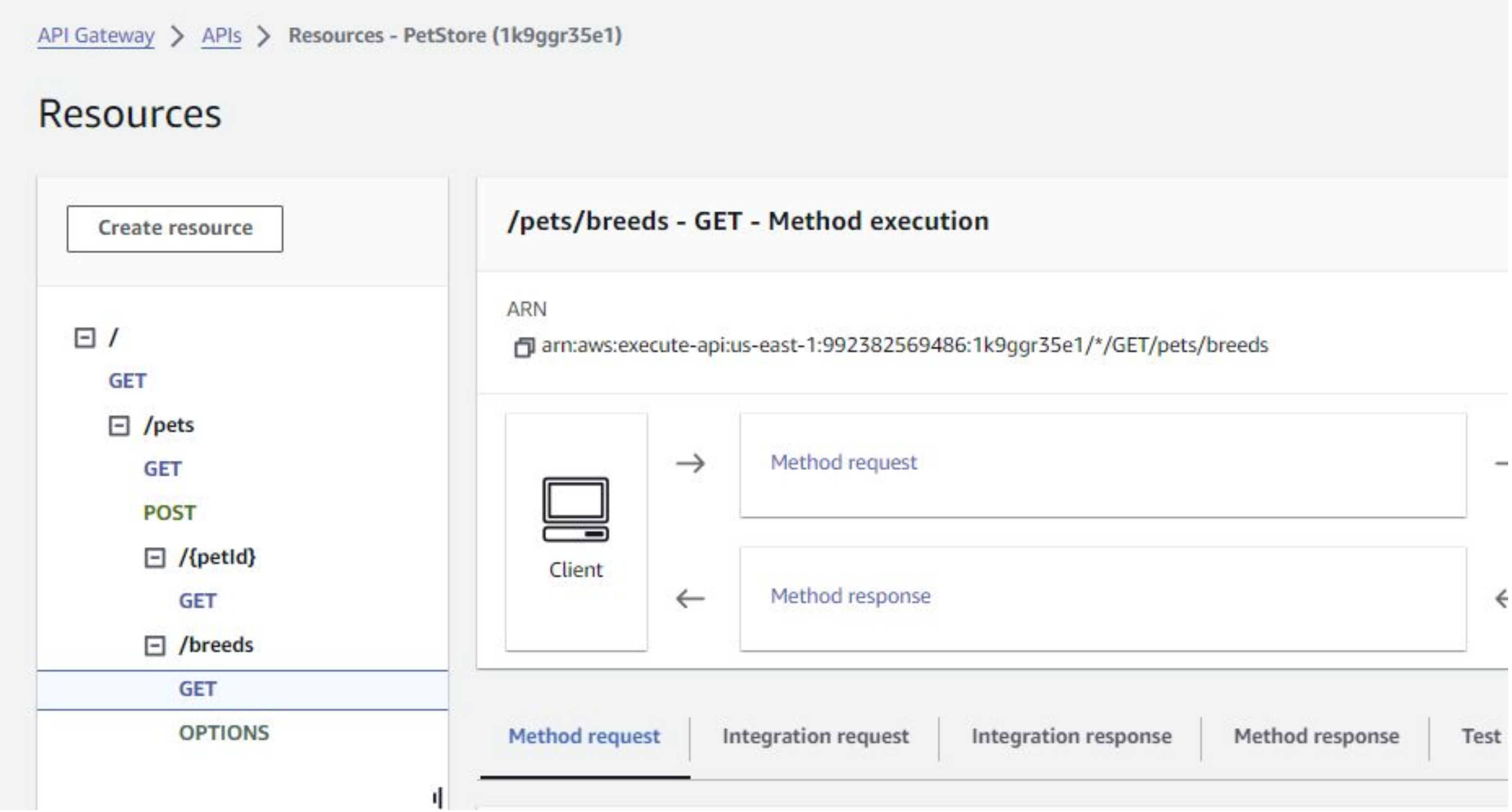
Add parameter

Authorization caching

TTL (0-3600 seconds)

120

API Gateway



API Gateway > APIs > Resources - PetStore (1k9ggr35e1)

Resources

Create resource

- /
  - GET
  - /pets
    - GET
    - POST
    - /{petId}
      - GET
    - /breeds
      - GET
      - OPTIONS

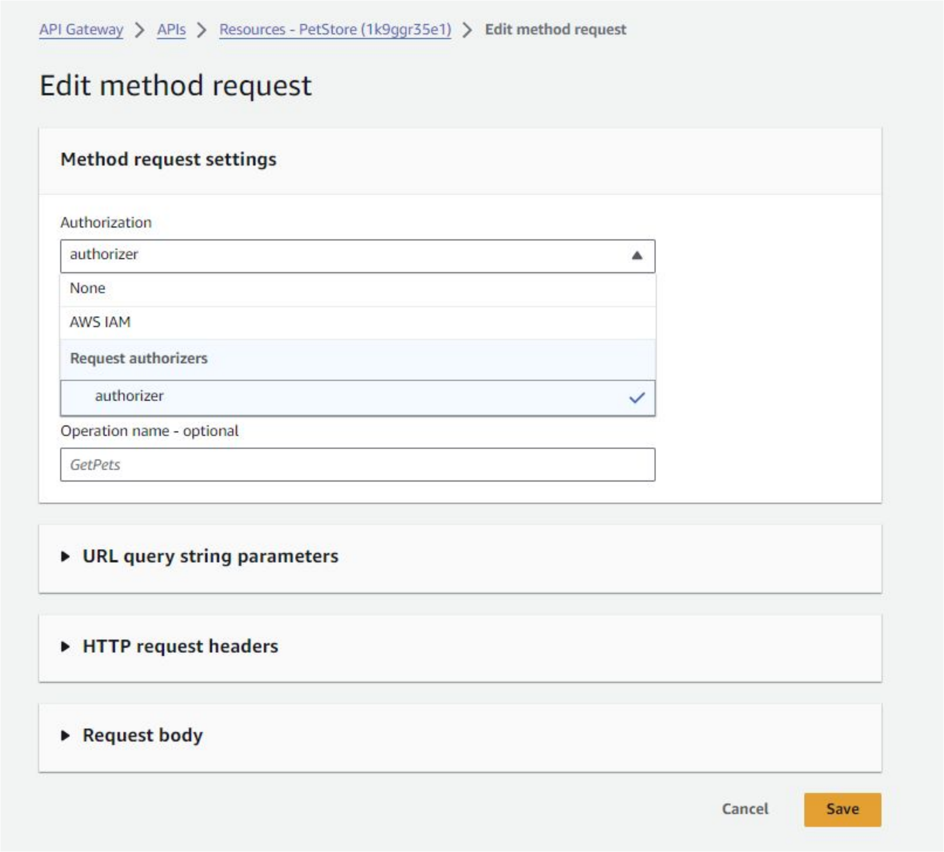
/pets/breeds - GET - Method execution

ARN

arn:aws:execute-api:us-east-1:992382569486:1k9ggr35e1/*/GET/pets/breeds

Method request | Integration request | Integration response | Method response | Test

API Gateway



API Gateway > APIs > Resources - PetStore (1k9ggr35e1) > Edit method request

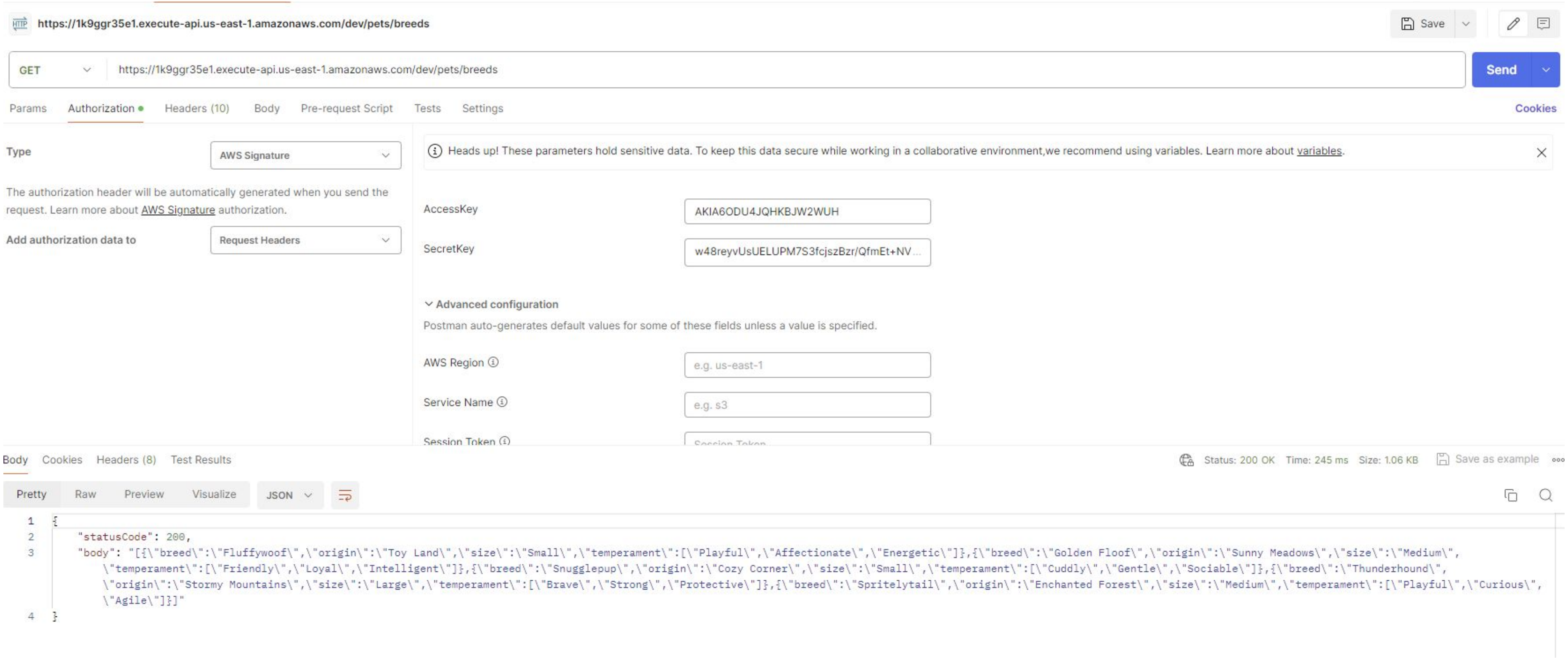
Edit method request

Method request settings

Authorization: authorizer

Operation name - optional: *GetPets*

API Gateway



https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

GET https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

Params | Authorization ● | Headers (10) | Body | Pre-request Script | Tests | Settings

Type: AWS Signature

The authorization header will be automatically generated when you send the request. Learn more about AWS Signature authorization.

Add authorization data to: Request Headers

ⓘ Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment,we recommend using variables. Learn more about variables.

AccessKey: AKIA6ODU4JQHKBJW2WUH

SecretKey: w48reyvUsUELUPM7S3fcjszBzr/QfmEt+NV...

Advanced configuration

Postman auto-generates default values for some of these fields unless a value is specified.

AWS Region ⓘ: e.g. us-east-1

Service Name ⓘ: e.g. s3

Session Token ⓘ: Session Token

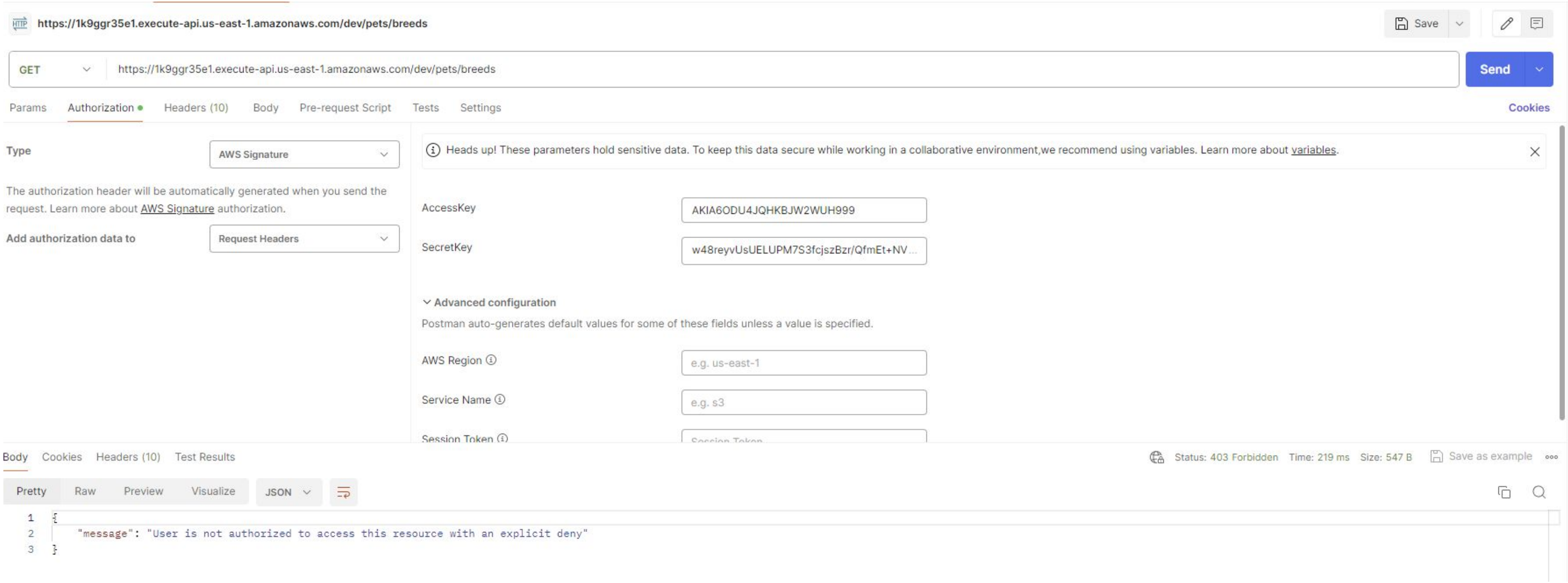
Body | Cookies | Headers (8) | Test Results

Status: 200 OK Time: 245 ms Size: 1.06 KB

Pretty | Raw | Preview | Visualize | JSON

```
{
    "statusCode": 200,
    "body": "[{\"breed\":\"Fluffywoof\",\"origin\":\"Toy Land\",\"size\":\"Small\",\"temperament\":[\"Playful\",\"Affectionate\",\"Energetic\"]},{\"breed\":\"Golden Floof\",\"origin\":\"Sunny Meadows\",\"size\":\"Medium\",
        \"temperament\":[\"Friendly\",\"Loyal\",\"Intelligent\"]},{\"breed\":\"Snugglepup\",\"origin\":\"Cozy Corner\",\"size\":\"Small\",\"temperament\":[\"Cuddly\",\"Gentle\",\"Sociable\"]},{\"breed\":\"Thunderhound\",
        \"origin\":\"Stormy Mountains\",\"size\":\"Large\",\"temperament\":[\"Brave\",\"Strong\",\"Protective\"]},{\"breed\":\"Spritelytail\",\"origin\":\"Enchanted Forest\",\"size\":\"Medium\",\"temperament\":[\"Playful\",\"Curious\",
        \"Agile\"]}]"
}
```

API Gateway



https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

GET https://1k9ggr35e1.execute-api.us-east-1.amazonaws.com/dev/pets/breeds

Type: AWS Signature

The authorization header will be automatically generated when you send the request. Learn more about AWS Signature authorization.

Add authorization data to: Request Headers

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment,we recommend using variables. Learn more about variables.

AccessKey: AKIA6ODU4JQHKBJW2WUH999

SecretKey: w48reyvUsUELUPM7S3fcjszBzr/QfmEt+NV...

Advanced configuration

Postman auto-generates default values for some of these fields unless a value is specified.

AWS Region: e.g. us-east-1

Service Name: e.g. s3

Session Token: Session Token

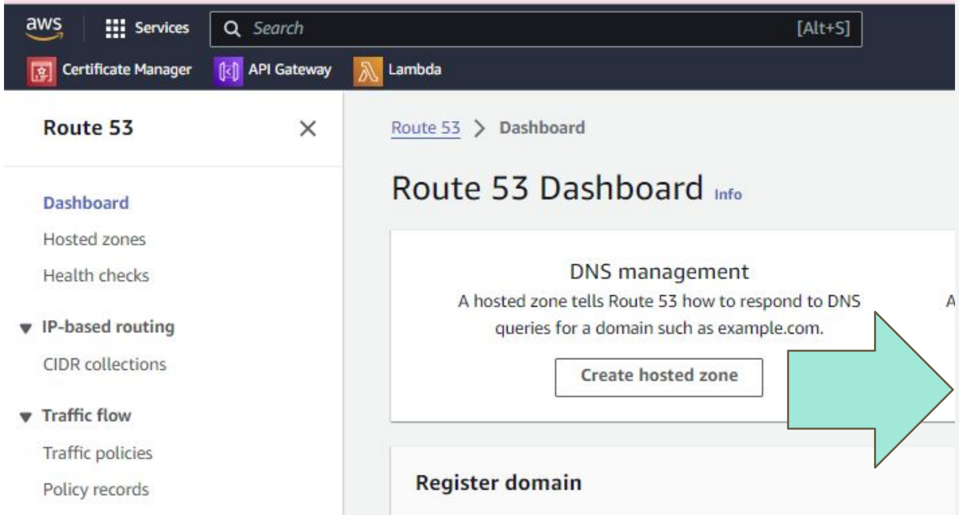
Body

Status: 403 Forbidden Time: 219 ms Size: 547 B

```
{
    "message": "User is not authorized to access this resource with an explicit deny"
}
```

10.Custom Domain

API Gateway



Route 53 ×

Dashboard

Hosted zones

Health checks

▼ IP-based routing

CIDR collections

▼ Traffic flow

Traffic policies

Policy records

Route 53 > Dashboard

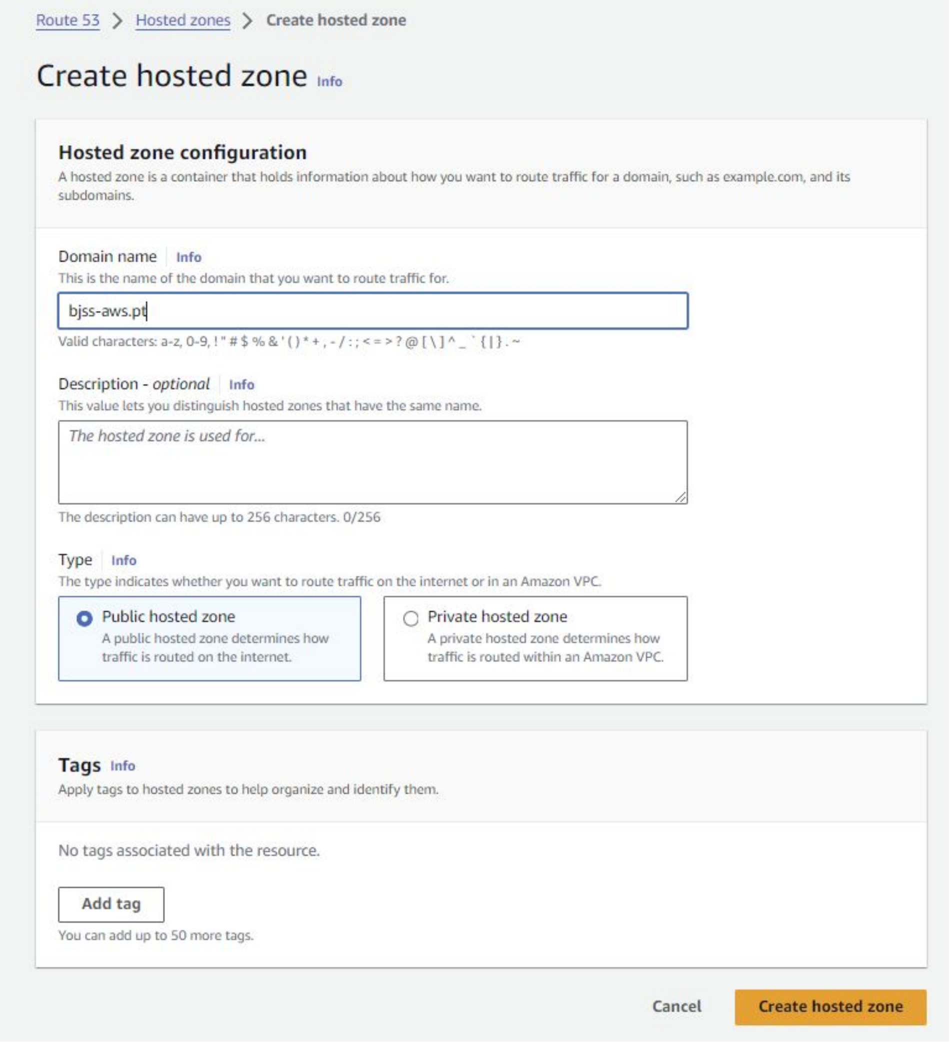
Route 53 Dashboard Info

DNS management

A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.

Create hosted zone

Register domain



Route 53 > Hosted zones > Create hosted zone

Create hosted zone Info

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name Info

This is the name of the domain that you want to route traffic for.

bjss-aws.pt

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Description - *optional* Info

This value lets you distinguish hosted zones that have the same name.

The hosted zone is used for...

The description can have up to 256 characters. 0/256

Type Info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

● Public hosted zone
A public hosted zone determines how traffic is routed on the internet.

○ Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

Tags Info

Apply tags to hosted zones to help organize and identify them.

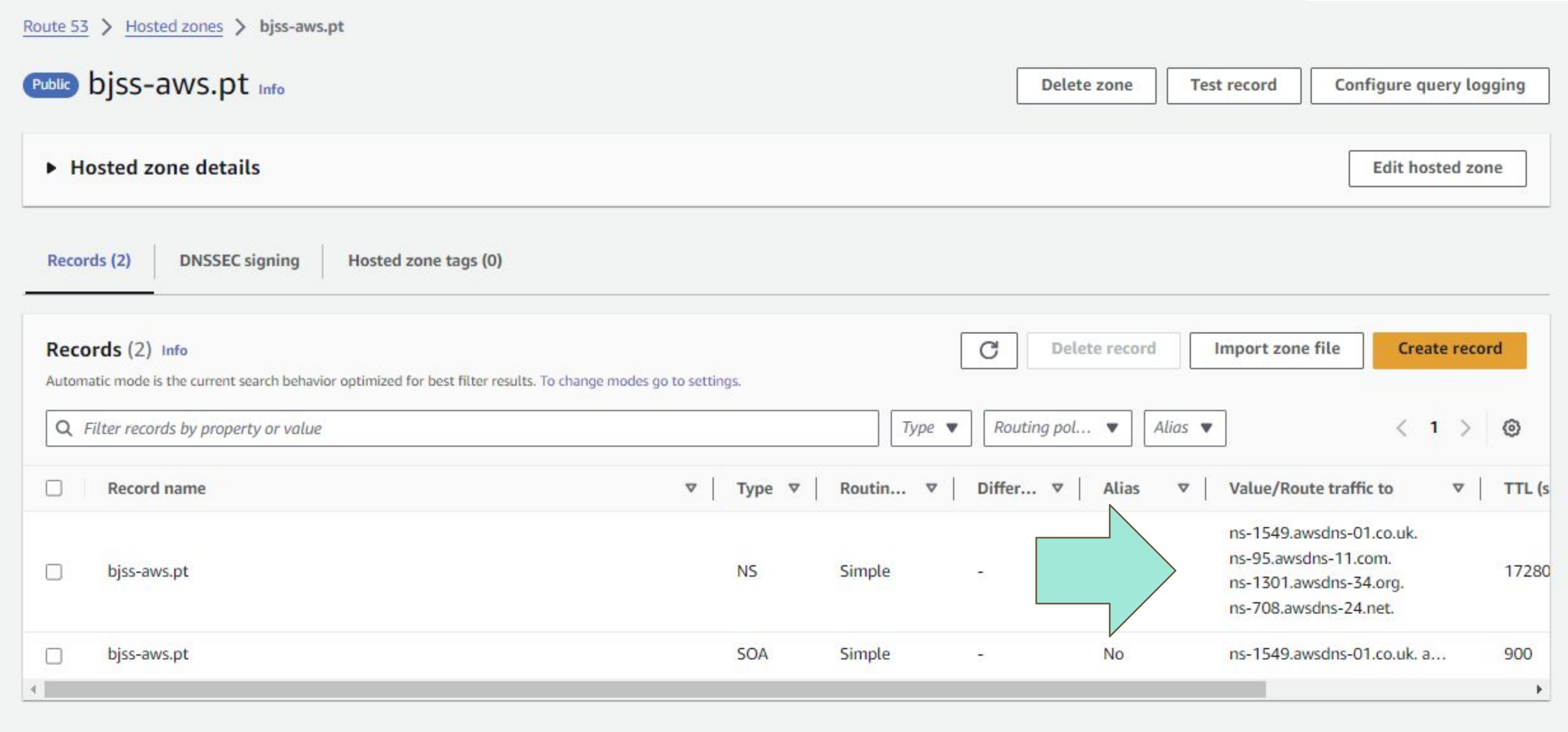
No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel Create hosted zone

API Gateway

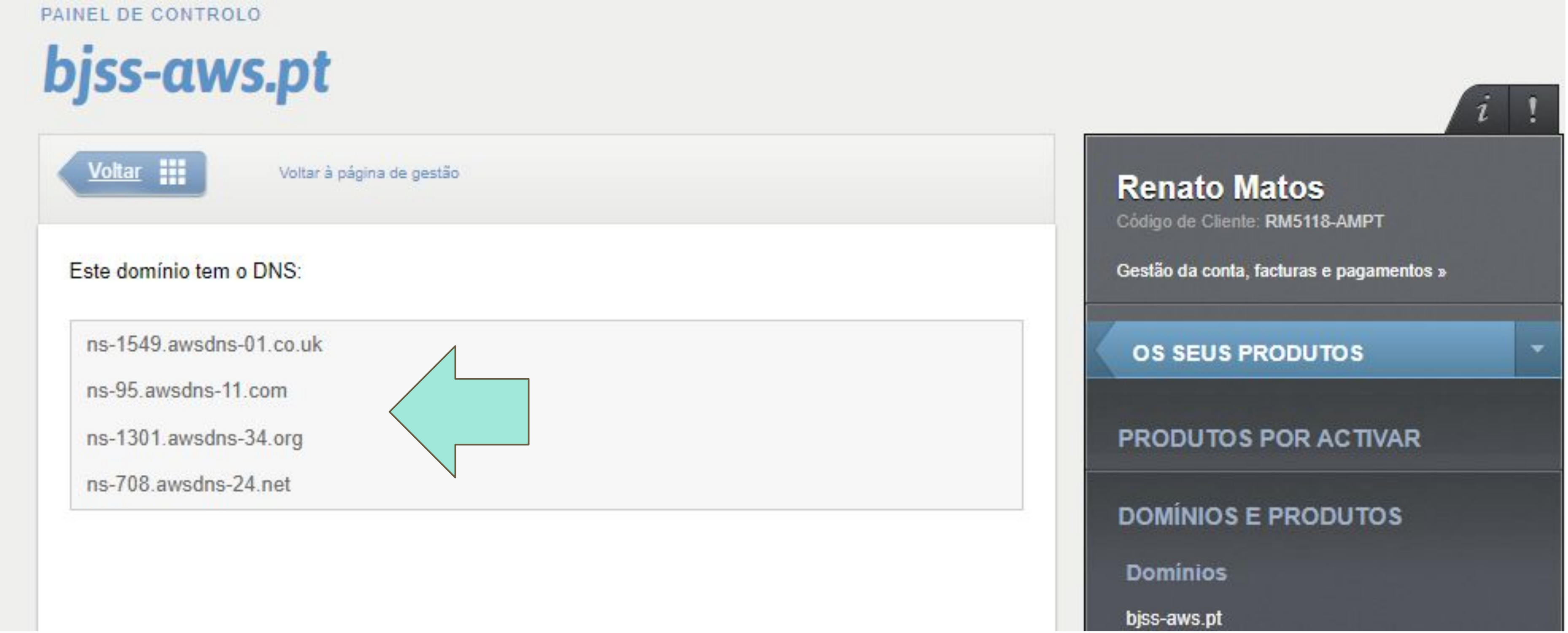


Records (2) Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

□	Record name	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL (s
□	bjss-aws.pt	NS	Simple	-		ns-1549.awsdns-01.co.uk. ns-95.awsdns-11.com. ns-1301.awsdns-34.org. ns-708.awsdns-24.net.	17280
□	bjss-aws.pt	SOA	Simple	-	No	ns-1549.awsdns-01.co.uk. a...	900

API Gateway



PAINEL DE CONTROLO

bjss-aws.pt

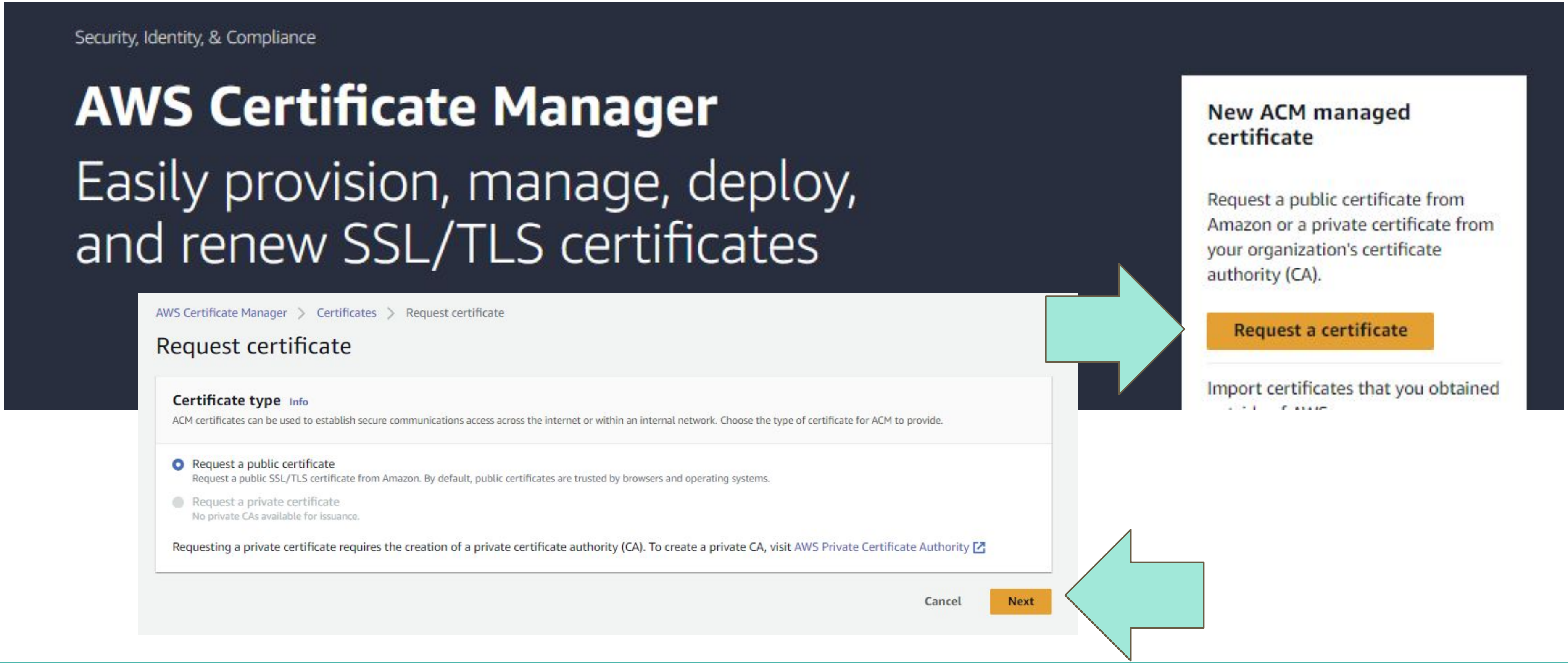
Este domínio tem o DNS:

ns-1549.awsdns-01.co.uk
ns-95.awsdns-11.com
ns-1301.awsdns-34.org
ns-708.awsdns-24.net

Renato Matos

Código de Cliente: RM5118-AMPT

API Gateway



Security, Identity, & Compliance

AWS Certificate Manager

Easily provision, manage, deploy, and renew SSL/TLS certificates

New ACM managed certificate

Request a public certificate from Amazon or a private certificate from your organization's certificate authority (CA).

Import certificates that you obtained

AWS Certificate Manager > Certificates > Request certificate

Request certificate

Certificate type Info

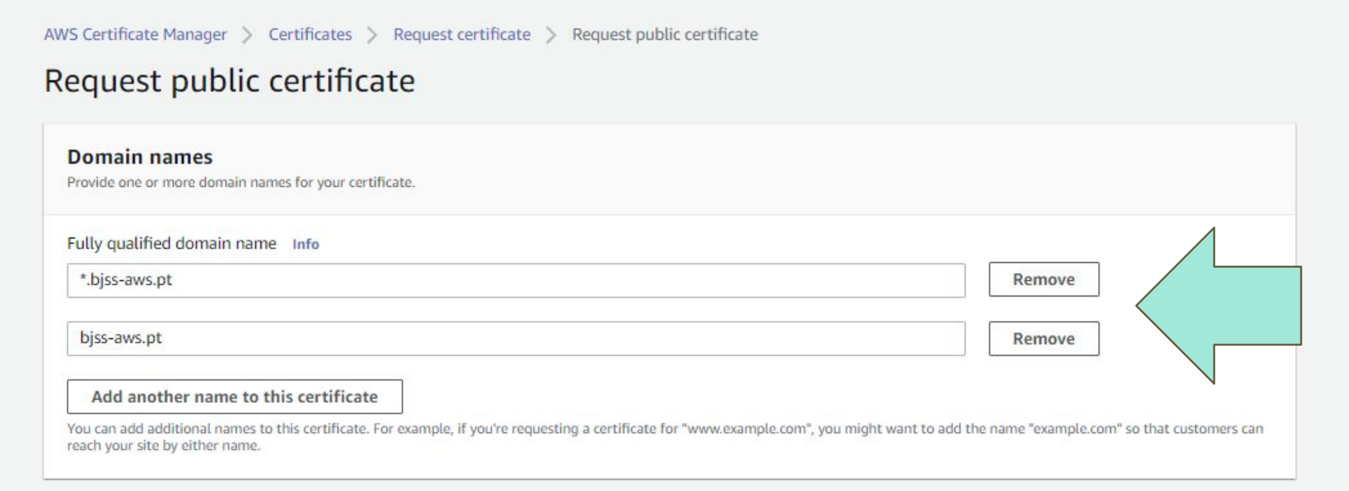
ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

● Request a public certificate
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

○ Request a private certificate
No private CAs available for issuance.

Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit AWS Private Certificate Authority

API Gateway



AWS Certificate Manager > Certificates > Request certificate > Request public certificate

Request public certificate

Domain names

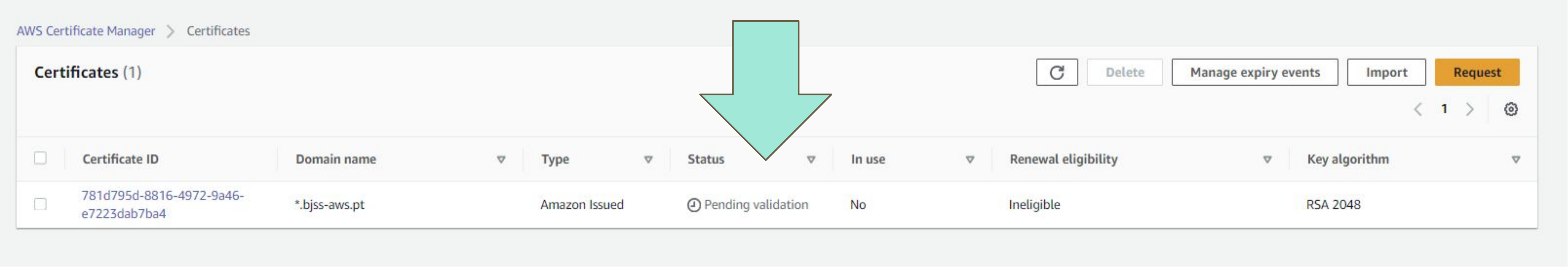
Provide one or more domain names for your certificate.

Fully qualified domain name Info

*.bjss-aws.pt

bjss-aws.pt

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.



AWS Certificate Manager > Certificates

Certificates (1)

□	Certificate ID	Domain name	Type	Status	In use	Renewal eligibility	Key algorithm
□	781d795d-8816-4972-9a46-e7223dab7ba4	*.bjss-aws.pt	Amazon Issued	Pending validation	No	Ineligible	RSA 2048

API Gateway

API Gateway

https://github.com/renatomatos79/bjss-aws-api-gateway

renato.matos79@gmail.com