



lab



lab title

**AWS API Gateway  
V1.00**



Course title

**BackSpace Academy  
AWS Certified Associate**



# Table of Contents

## Contents

|  |    |
|--|----|
| Table of Contents.....   | 1  |
| About the Lab .....  | 2  |
| Creating a REST API .....                                      | 3  |
| Create the REST API .....                                      | 3  |
| Sending HTTP Requests to your REST API.....                    | 5  |
| Creating a Dynamic Serverless App with S3 and API Gateway..... | 9  |
| Enable Cross Origin Resource Sharing (CORS) for your API.....  | 9  |
| Create a Static Website .....                                  | 10 |
| Clean Up.....  | 16 |

## About the Lab

**Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.**

These lab notes are to support the hands on instructional videos of the AWS API Gateway section of the AWS Certified Associate Course.

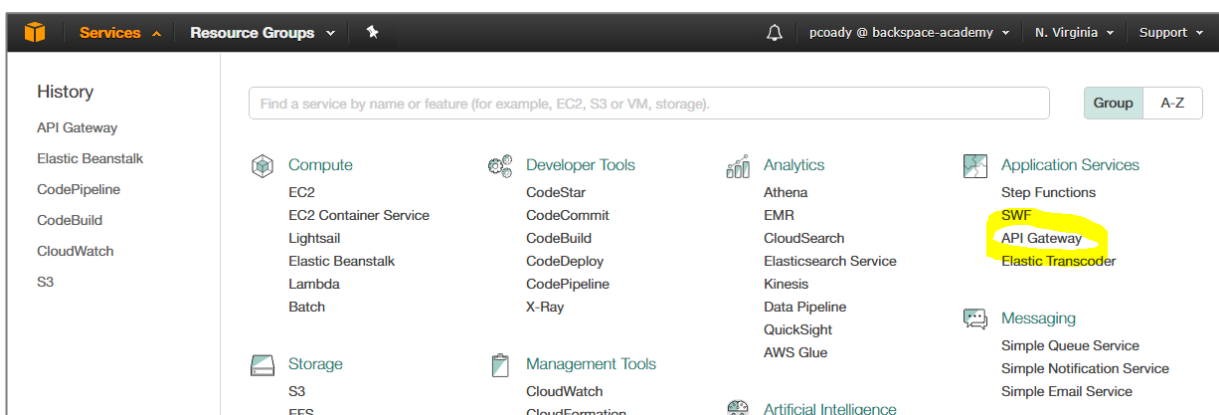
**Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the latest version with any updates or corrections.**

# ▶ Creating a REST API

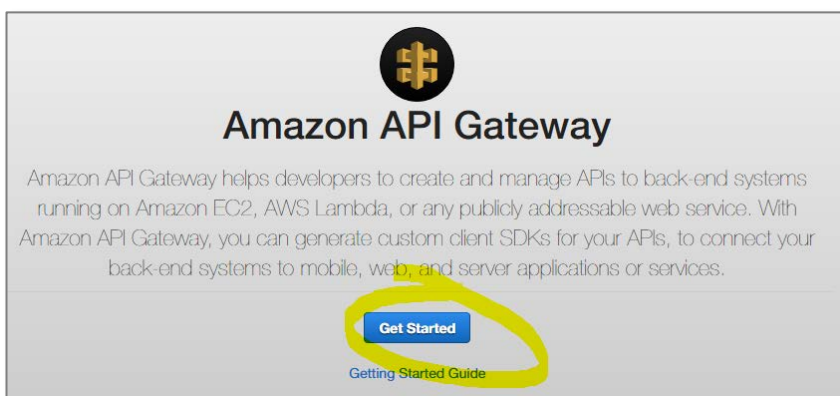
In this section, we will use the AWS API Gateway Service to create a highly available and fault tolerant REST Application Programming Interface (API).

Create the REST API

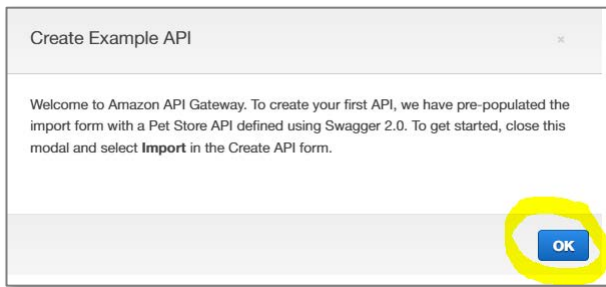
Make sure you are in US-East (N. Virginia) region. From the AWS console select “API Gateway” from the Application services.



Click “Get Started”

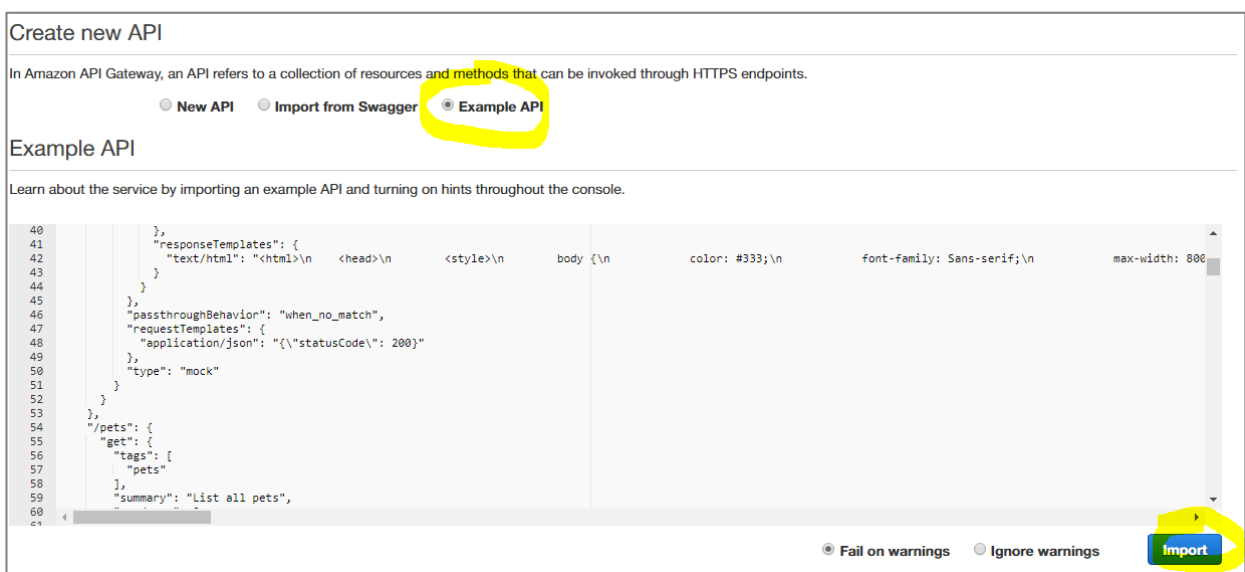


Click “OK”

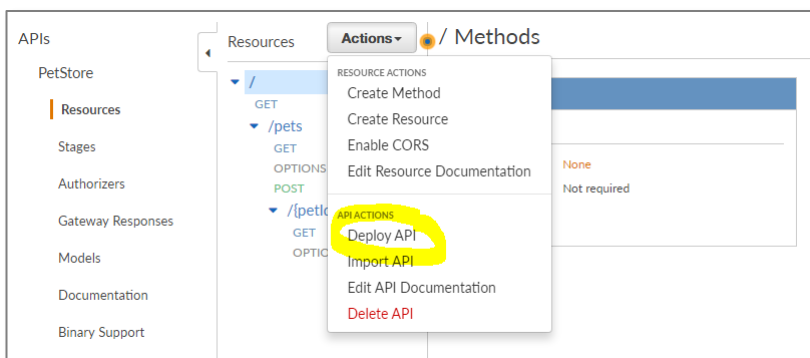


Select "Example API"

Click "Import"



Click "Actions" – "Deploy API"



Select "Deployment stage" – "New Stage"

Give the stage a name

Click "Deploy"

Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Deployment stage: [New Stage]

Stage name\*: Demo

Stage description:

Deployment description:

Cancel Deploy

Click on Invoke URL

APIs

PetStore

Resources

Stages

Authorizers

Gateway Responses

Models

Documentation

Binary Support

Dashboard

Usage Plans

API Keys

Custom Domain Names

Stages

Create

Demo

Demo Stage Editor

Delete Stage

Invoke URL: <https://60hwiz4rfi.execute-api.us-east-1.amazonaws.com/Demo>

Settings Stage Variables SDK Generation Export Deployment History Documentation History

Configure the metering and caching settings for the Demo stage.

Cache Settings

Enable API cache ☐

CloudWatch Settings

Enable CloudWatch Logs ☐ ⓘ

Enable Detailed CloudWatch Metrics ☐ ⓘ

Default Method Throttling

You will see a welcome page for your new API

## Welcome to your Pet Store API

You have successfully deployed your first API. You are seeing this HTML page because the GET method to the root resource of your API returns this content as a Mock integration.

The Pet Store API contains the /pets and /pets/{petId} resources. By making a [GET request](#) to /pets you can retrieve a list of Pets in your API. If you are looking for a specific pet, for example the pet with ID 1, you can make a [GET request](#) to /pets/1.

You can use a REST client such as [Postman](#) to test the POST methods in your API to create a new pet. Use the sample body below to send the POST request:

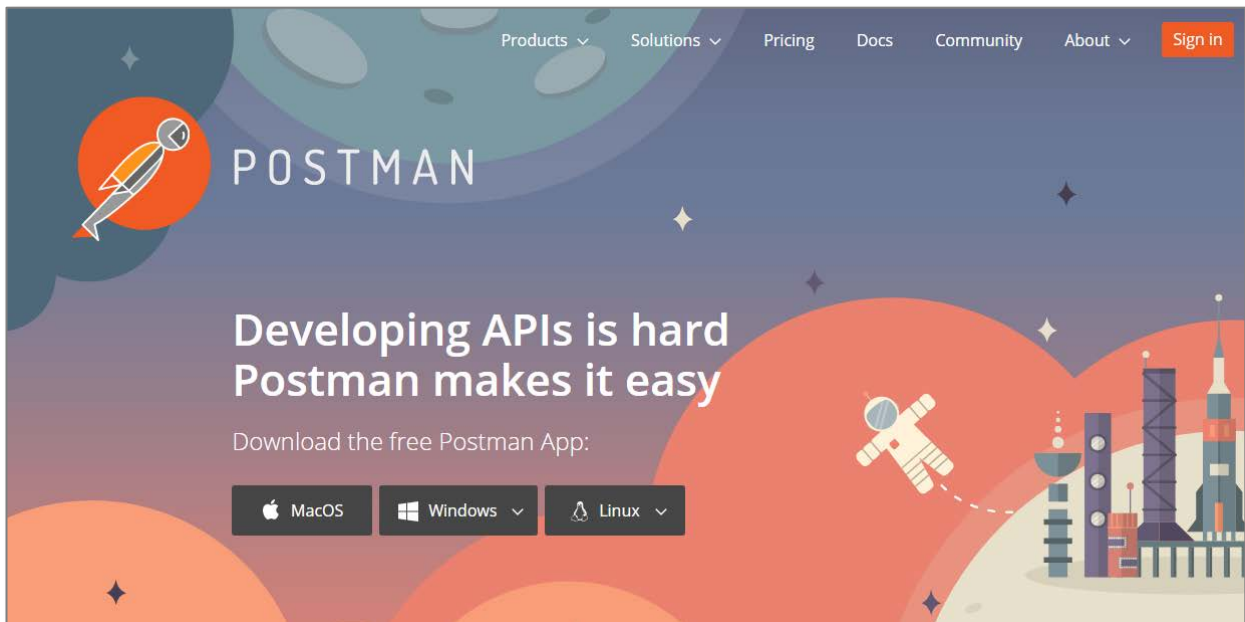
```
{
  "type": "cat",
  "price": 123.11
}
```

## Sending HTTP Requests to your REST API

We will use PostMan to send requests to our REST API

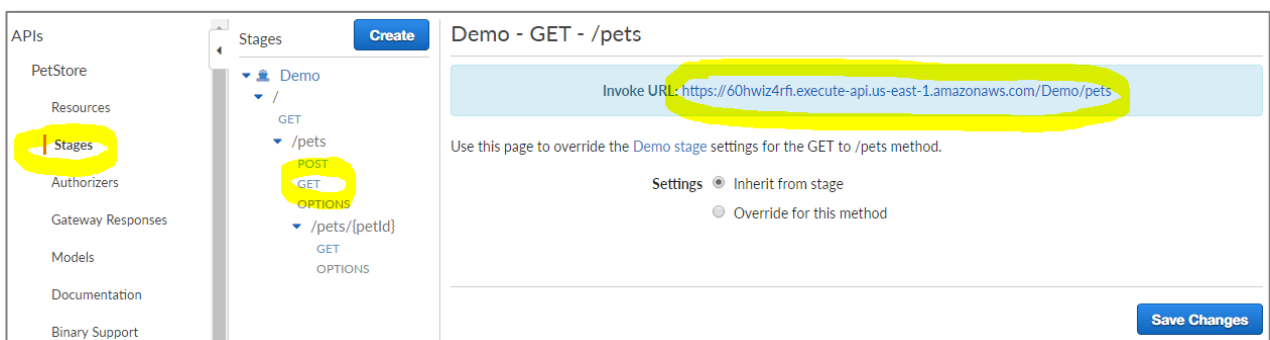
Go to the PostMan website and install PostMan

<https://www.getpostman.com/>



Go to the API Gateway console

Copy the Invoke URL for a GET request to /pets



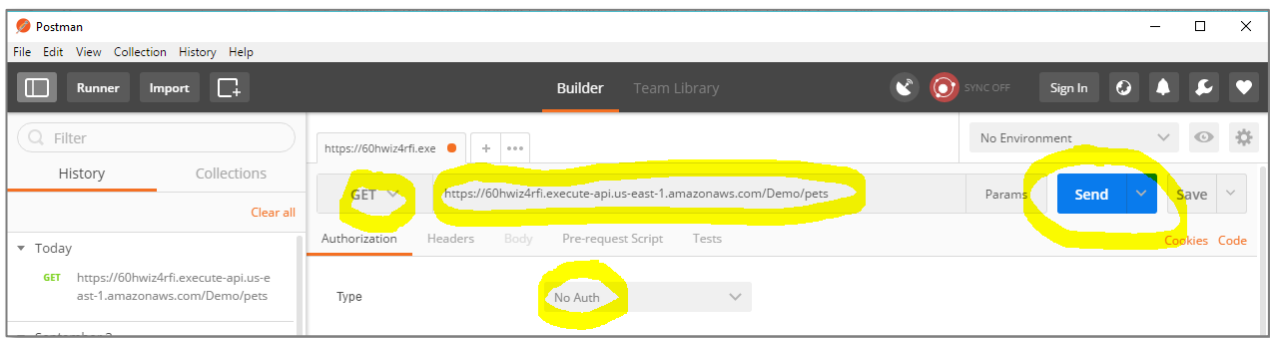
Open PostMan

Select "GET"

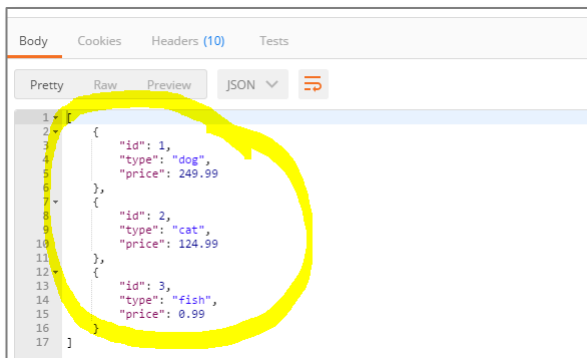
Paste in the URL

Select "No Auth"

Click "Send"



Your API will respond with JSON containing all the Pet entries.



Go Back to the API Gateway console

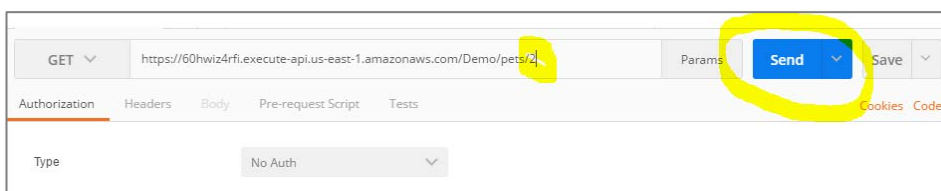
Copy the invoke URL for a GET request to /pets/{petId}



Go back to Postman

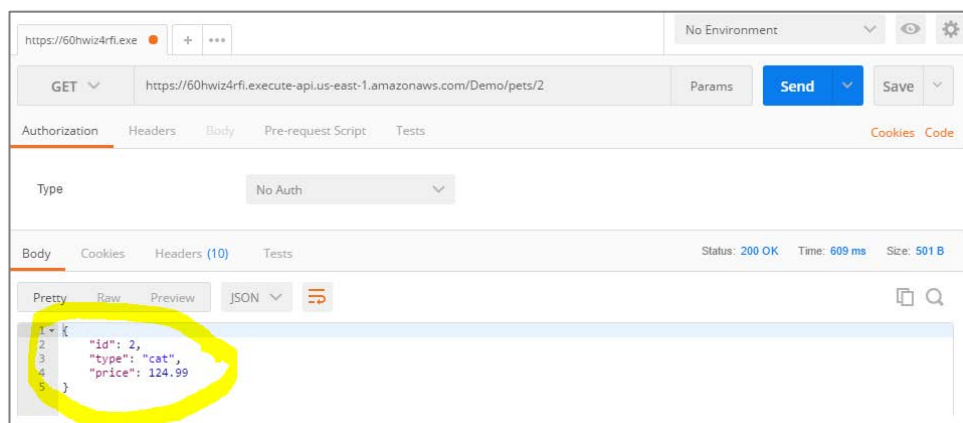
Paste in the URL

Remove the {PetID} from the end and replace with a number for a pet



The API will return the JSON for the pet id





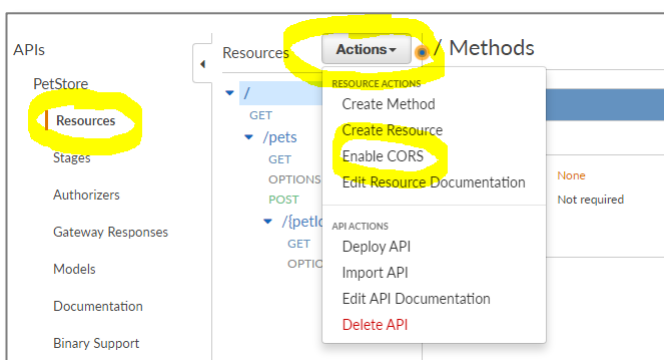
# **Creating** a Dynamic Serverless App with S3 and API Gateway

In this section, we will use the REST Application Programming Interface (API) we created to deliver dynamic content with a static website.

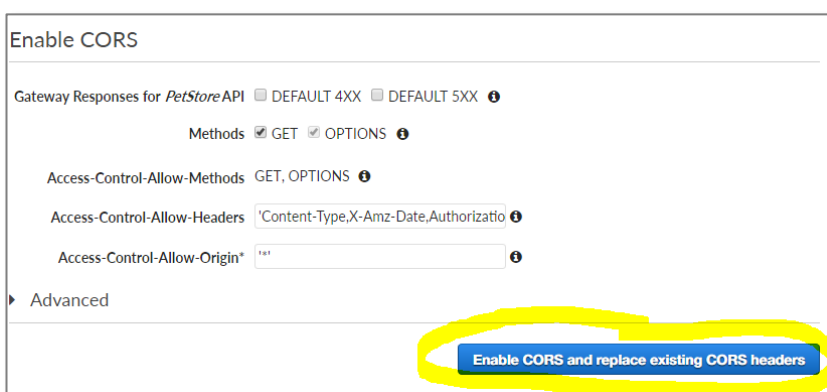
Enable Cross Origin Resource Sharing (CORS) for your API

When your API's resources receive requests from a domain other than the API's own domain, you must enable cross-origin resource sharing (CORS) for selected methods on the resource. Otherwise the action will be blocked by the browser.

Go to API Gateway console and Select “Resources” – “Actions” – “Enable CORS”



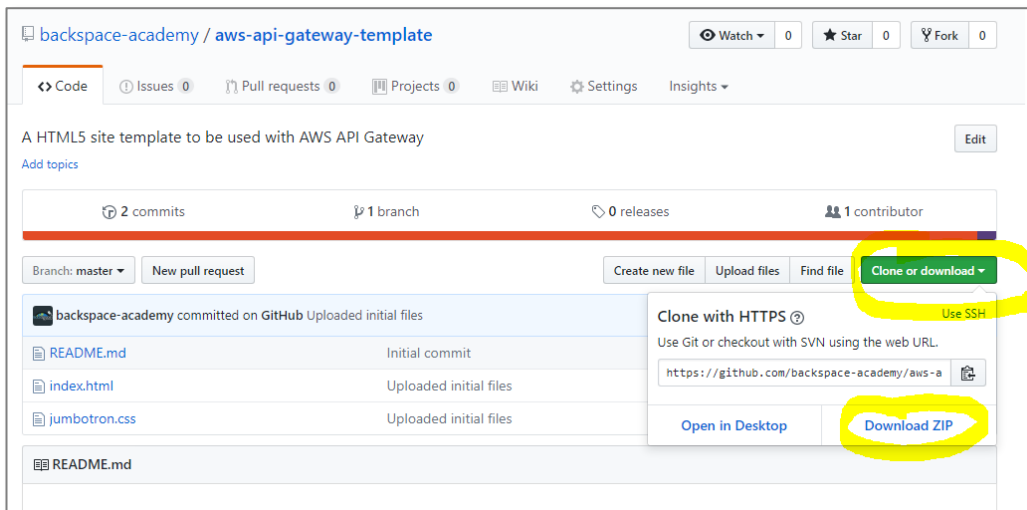
Click “Enable CORS and replace existing CORS headers”



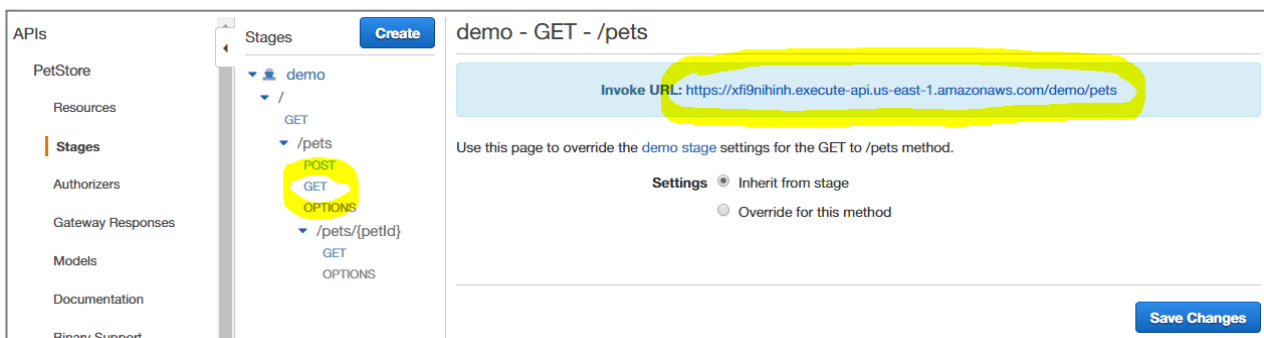
## Create a Static Website

Go to GitHub.com and download the zip file for the template application.

<https://github.com/backspace-academy/aws-api-gateway-template>



Open js/app.js and edit the getPet function with your API endpoint for requesting pets (make sure /pets/ is on the end):



```
// Send request to API
```

```
function getPet(){
```

```
    // Put your endpoint here, not this!!!!
```

```
    var apiEndpoint = 'https://1234567890.execute-api.us-east-1.amazonaws.com/Demo/pets/';
```

```
    var petId = $('#petId').val();
```

```
    var numPets = 3;
```

```
    if (petId>numPets) {
```

```
        alert('Invalid entry - exceeds number of pets');
```

```

    }
    else if (petId < 1) {
        alert('Invalid entry - must be greater than zero');
    }
    else{
        // Send the request
        var apiRequest = apiEndpoint + petId;
        $.get(apiRequest, function(data, status){
            // Display the response
            alert( '\nStatus: ' + status + '\nType: ' + data.type + '\nPrice: $' + data.price);
        });
    }
}

```

Make sure you are in US-East (N. Virginia) region. From the AWS console select “S3” from the Storage services.

Create a public bucket.

The screenshot shows the 'Create bucket' wizard in the AWS Management Console. The first step, 'Name and region', is active. The 'Bucket name' field contains 'backspace-aws-api-gateway-template' and is highlighted with a yellow circle. The 'Region' dropdown is set to 'US East (N. Virginia)'. Below, the 'Copy settings from an existing bucket' section shows a dropdown for 'Select bucket (optional)' with '5 Buckets' indicated. At the bottom, the 'Next' button is highlighted with a green circle.

Create bucket

1 Name and region

2 Set properties

3 Set permissions

4 Review

Versioning

Keep multiple versions of an object in the same bucket.

[Learn more](#)

☐ Disabled

Logging

Set up access log records that provide details about access requests.

[Learn more](#)

☐ Disabled

Tags

Use tags to track your cost against projects or other criteria.

[Learn more](#)

☐ 0 Tags

Previous

Next

Create bucket

1 Name and region

2 Set properties

3 Set permissions

4 Review

Manage users

| User ID       | Objects  | Object permissions   |
|---------------|--|--|
| pcoady(Owner) | <input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write | <input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write |

×

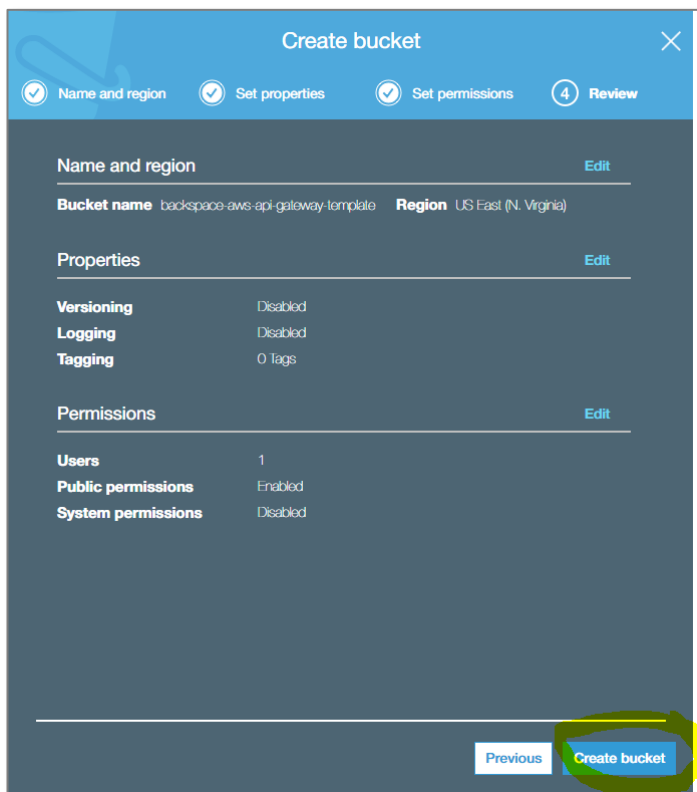
Manage public permissions

Grant public read access to this bucket

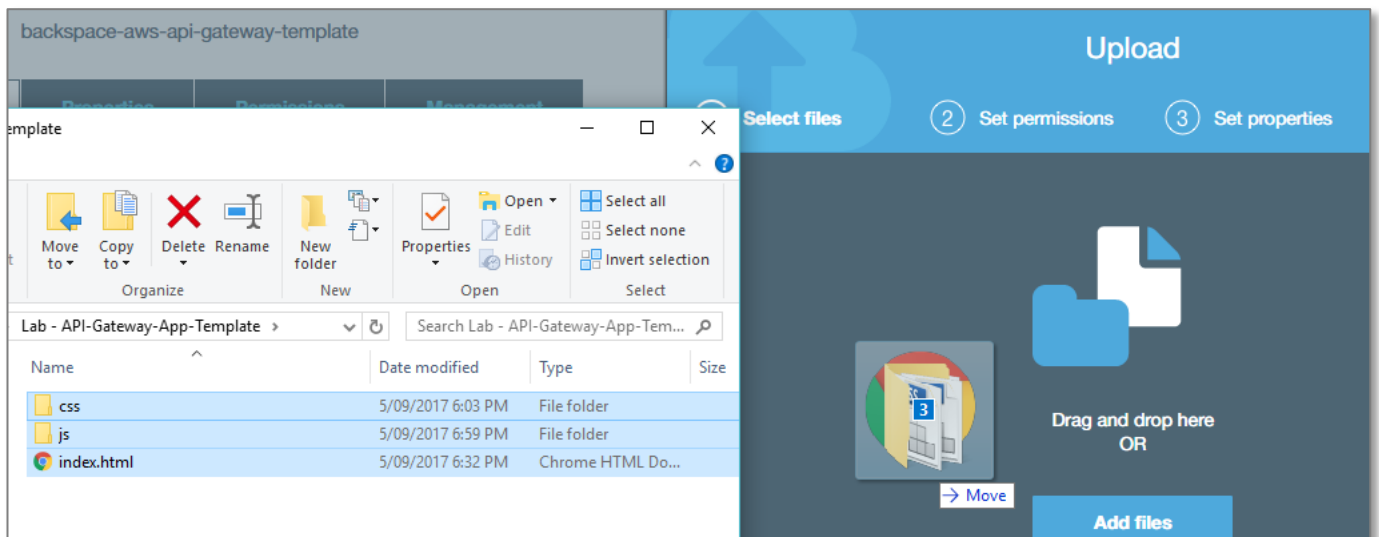
This bucket will have public read access.  
Everyone in the world will have read access to this bucket.

Previous

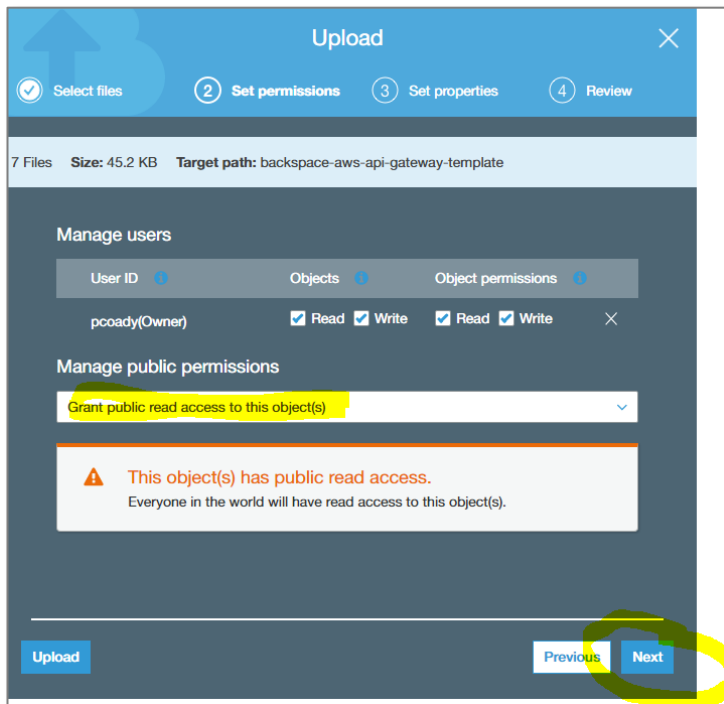
Next



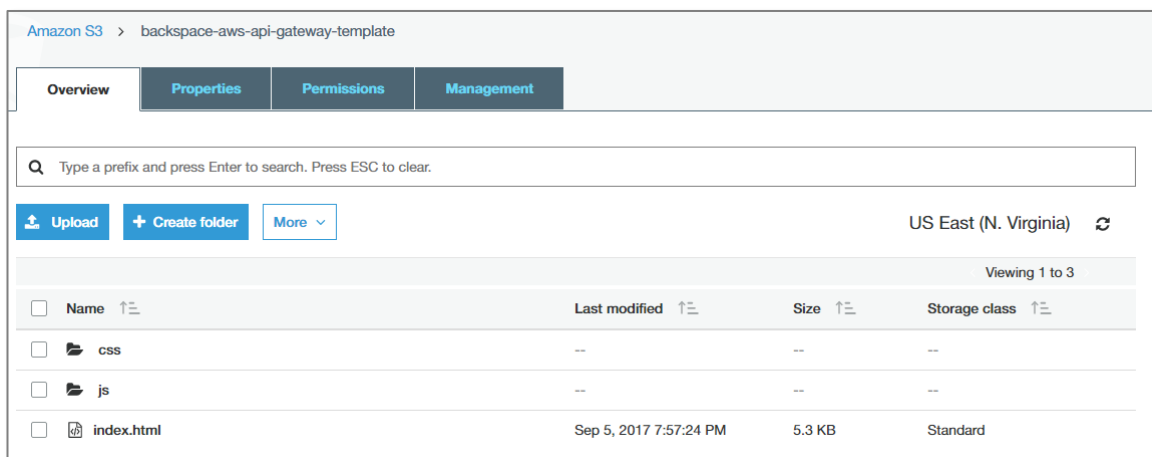
Unzip the template application and upload to the bucket (with drag and drop).



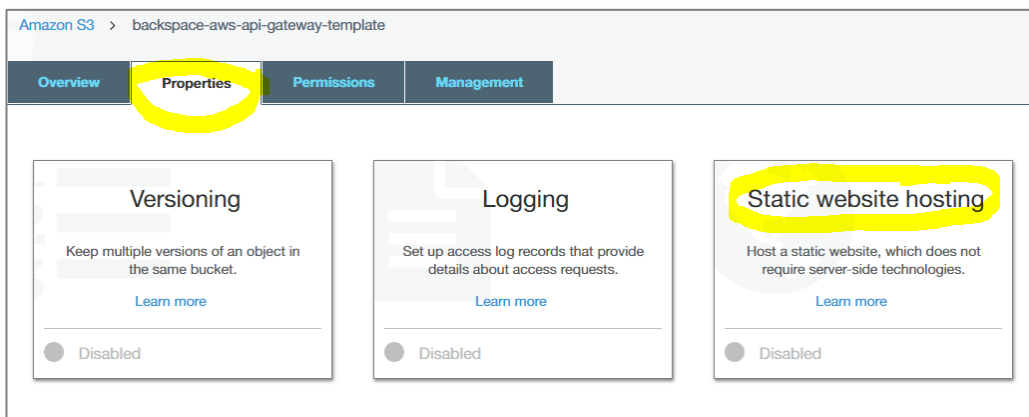
Select public permissions



Click next to start uploading

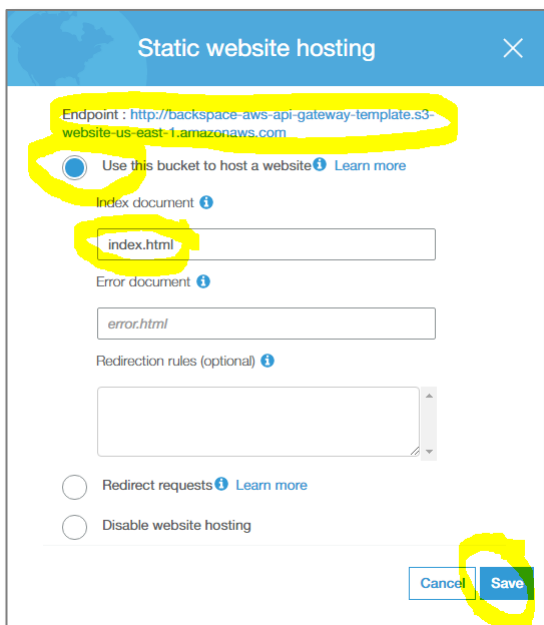


Enable static website hosting for the bucket



Copy the website endpoint

Enter the Index document



Go to the web site endpoint in your browser.

Enter a number between 1 and 3 for Pet ID

Click "Get Pet!"



AWS API Gateway Template

Email Password Sign in

# Welcome, Space Cadets!

This is a template from BackSpace Academy for a simple website using the AWS API Gateway service. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Pet ID: 2

Get Pet!

## Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

## Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

View details »

## Heading

Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

View details »

© 2017 BackSpace Academy.

The app will return the details for that pet ID and the status of the request.

AWS API Gateway Template

...ws-api-gateway-template.s3-website-us-east-1.amazonaws.com says:

Status: success  
Type: cat  
Price: \$124.99

OK

# Welcome, S

This is a template from BackSpace Academy for a simple website using the AWS API Gateway service. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Pet ID: 2

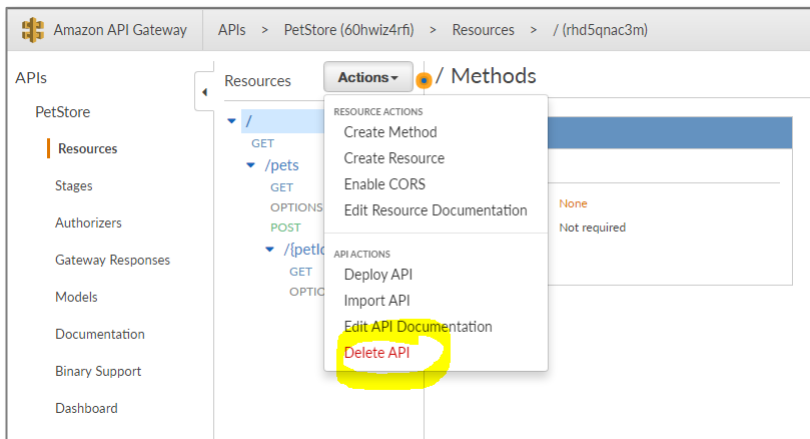
Get Pet!

## Clean Up

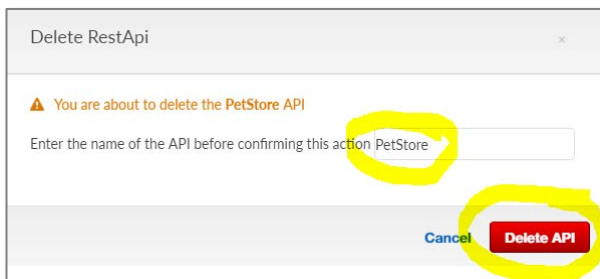
Now that we have finished the lab we can delete the resources to avoid costs.

Go back to the API Gateway console

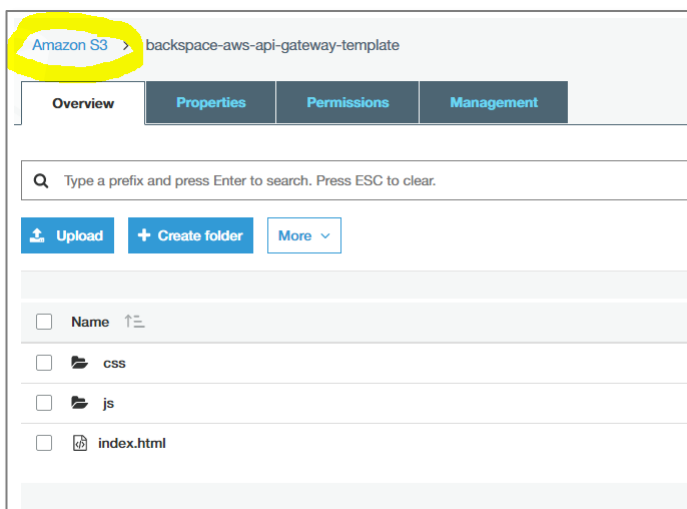
Select "Actions" – "Delete API"



Enter API name and delete



Go back to the Amazon S3 console dashboard



Select the bucket and delete

