lab

lab title

AWS API Gateway
V1.00

Course title

BackSpace Academy
AWS Certified Associate

BackSpace

Table of Contents

Contents

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.

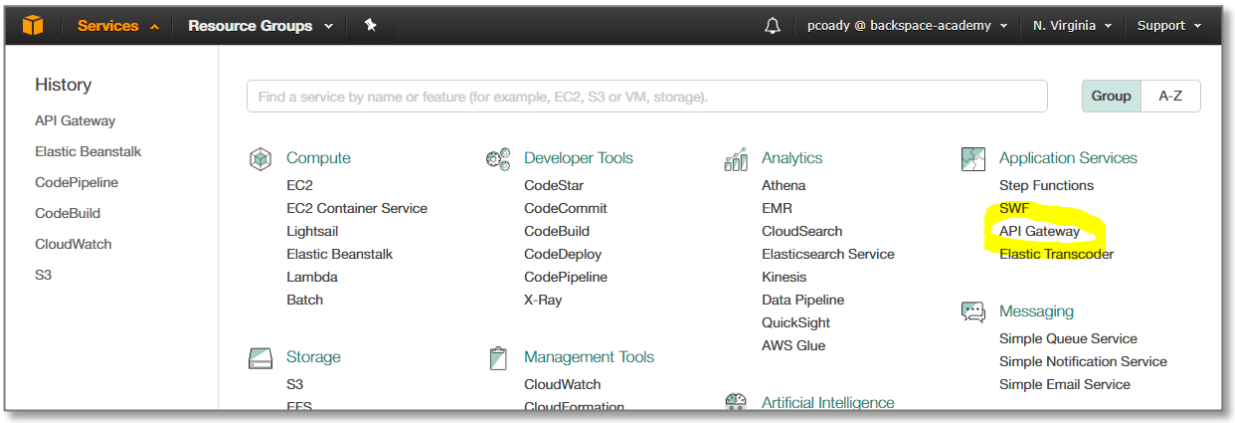
Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

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).

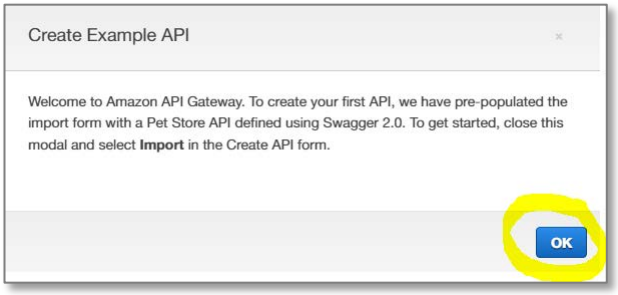
Create the REST API

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



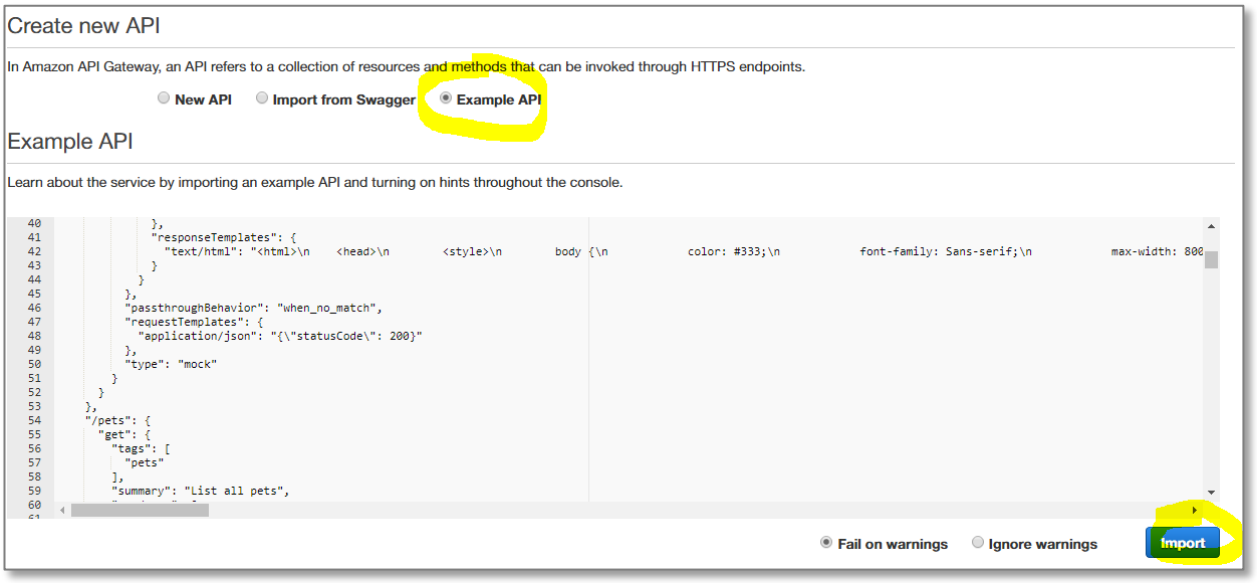
Click “Get Started”

Click “OK”

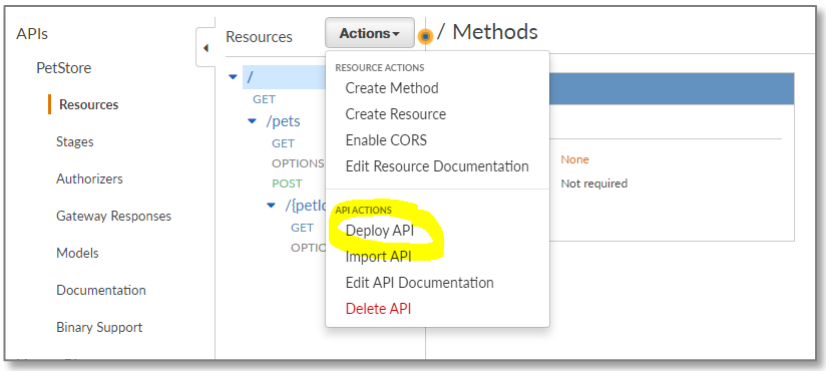


Select “Example API”

Click “Import”



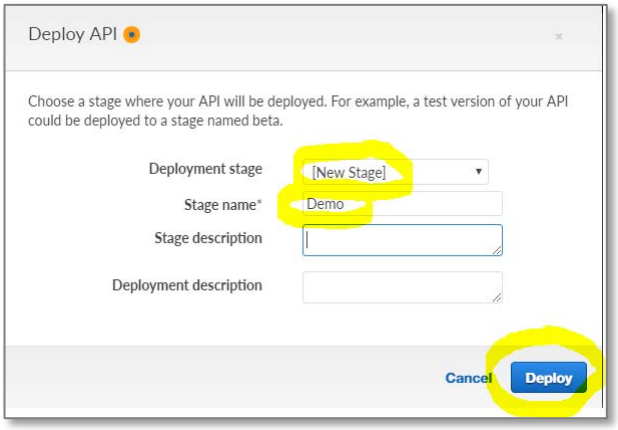
Click “Actions” – “Deploy API”



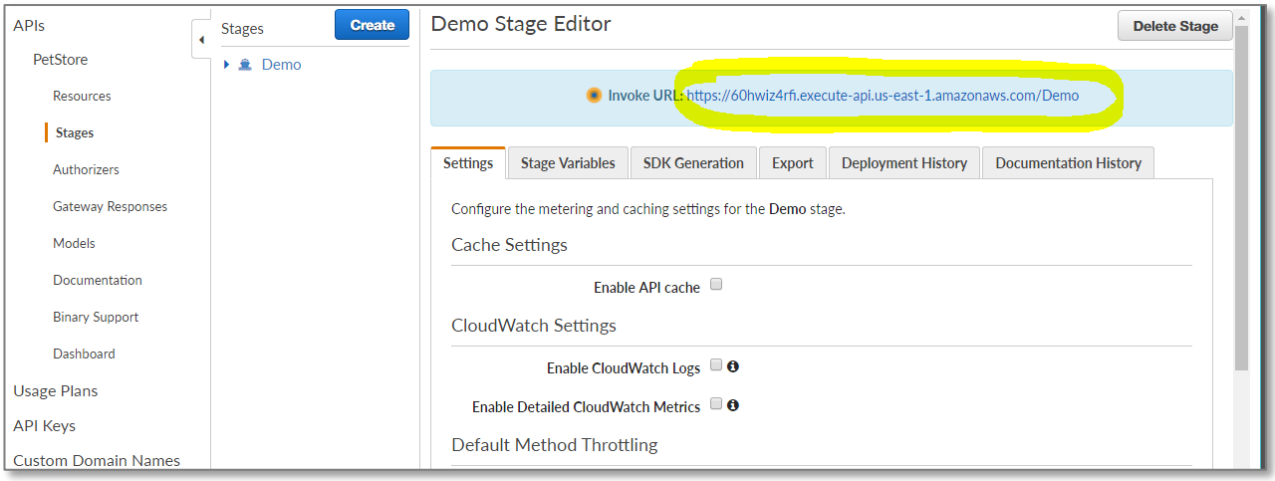
Select “Deployment stage” – “New Stage”

Give the stage a name

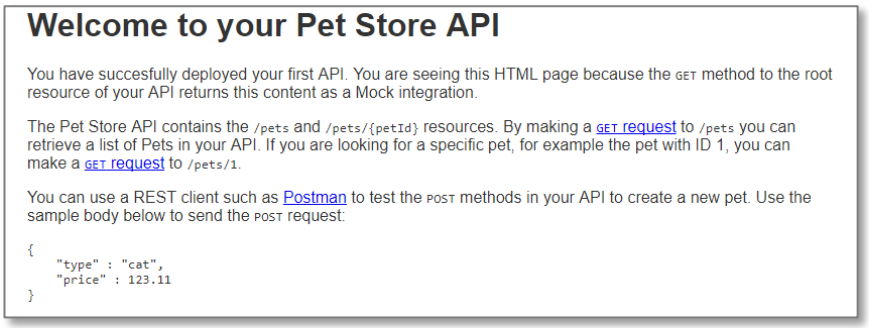
Click “Deploy”



Click on Invoke URL



You will see a welcome page for your new API

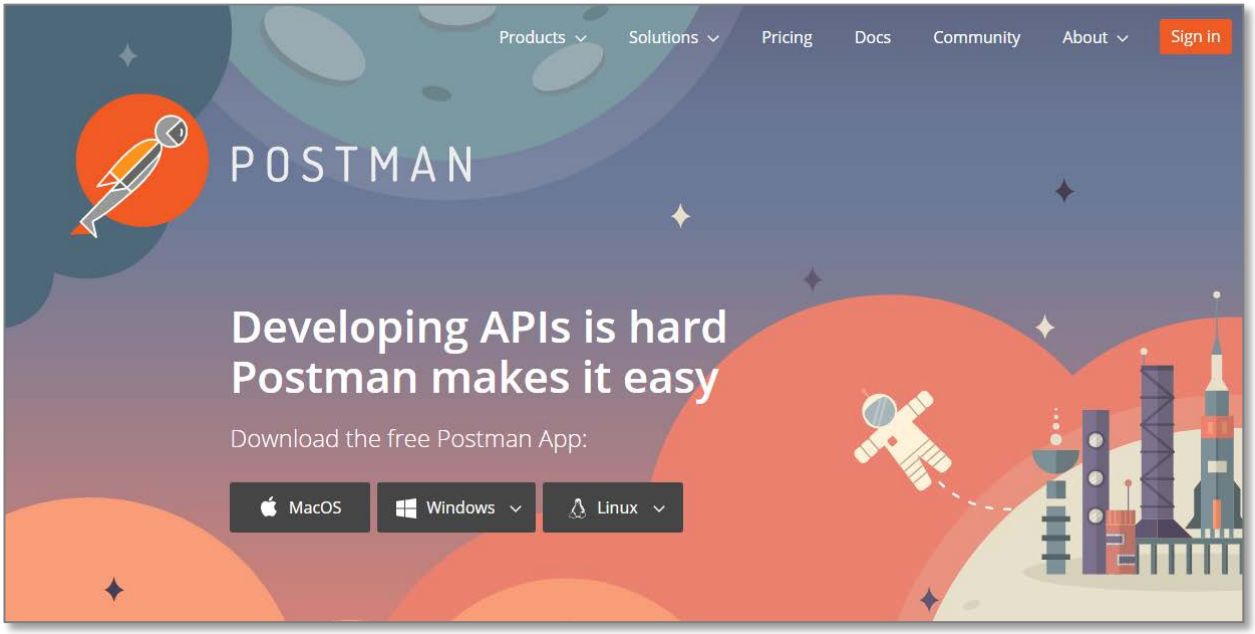


Sending HTTP Requests to your REST API

We will use PostMan to send requests to our REST API

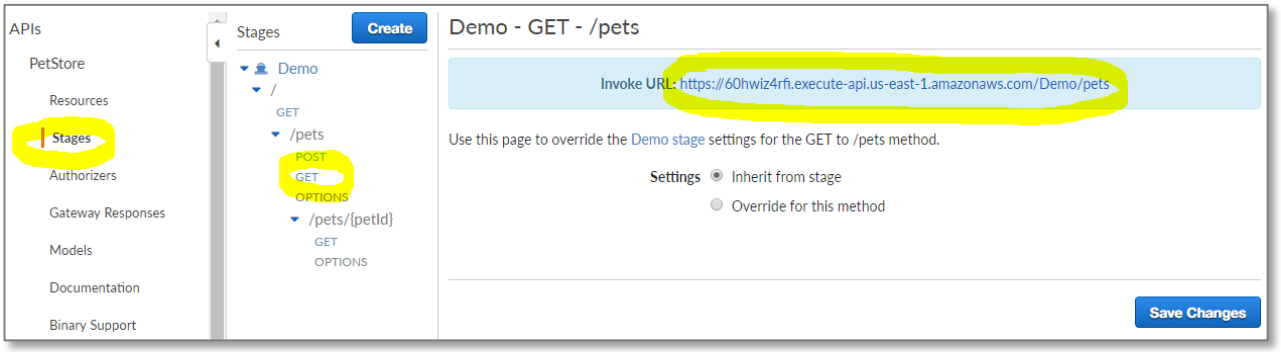
Go to the PostMan website and install PostMan

https://www.getpostman.com/



Go to the API Gateway console

Copy the Invoke URL for a GET request to /pets



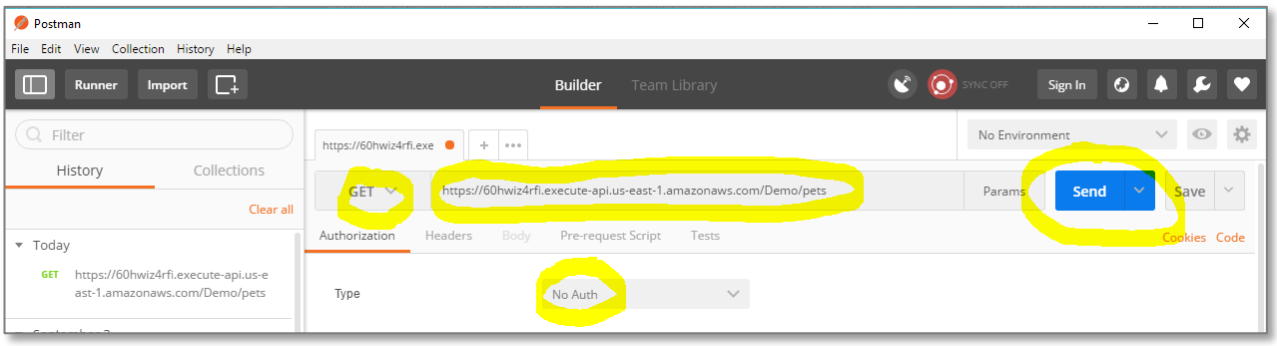
Open PostMan

Select “GET”

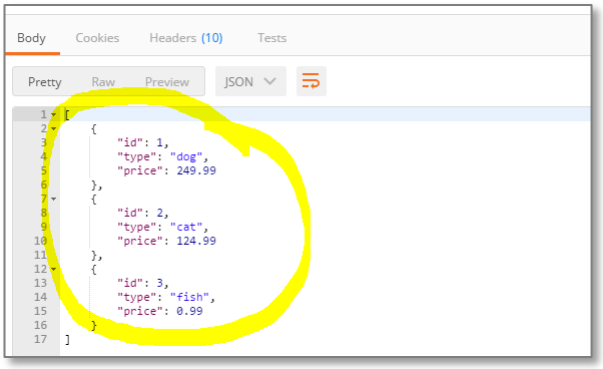
Paste in the URL

Select “No Auth”

Click “Send”

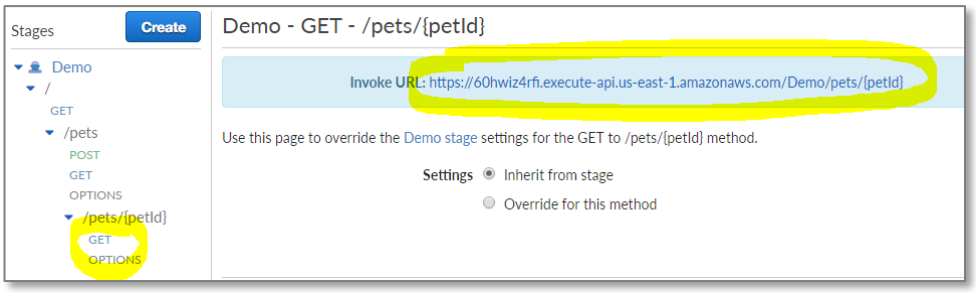


Your API will respond with JSON containing all the Pet entries.



Go Back to the API Gateway console

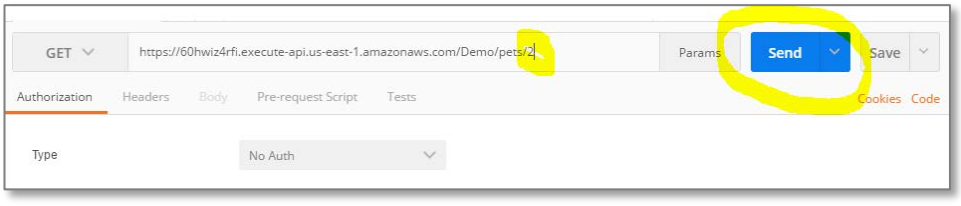
Copy the invoke URL for a GET request to /pets/{petId}



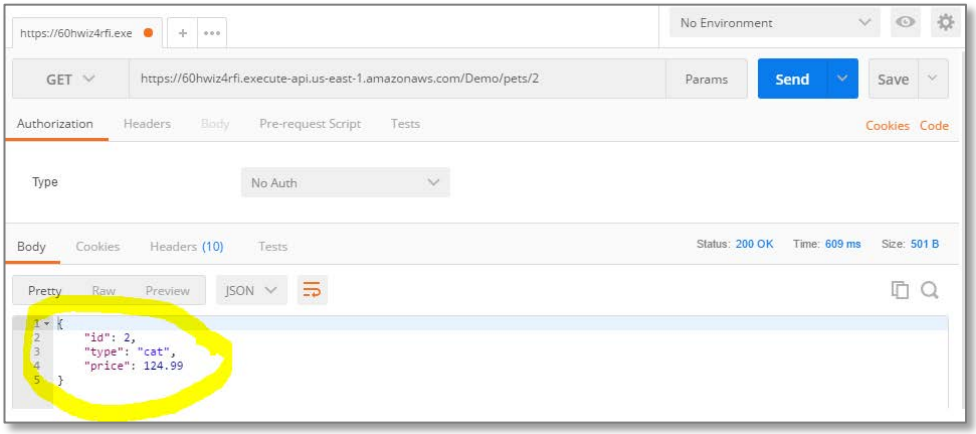
Go back to Postman

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



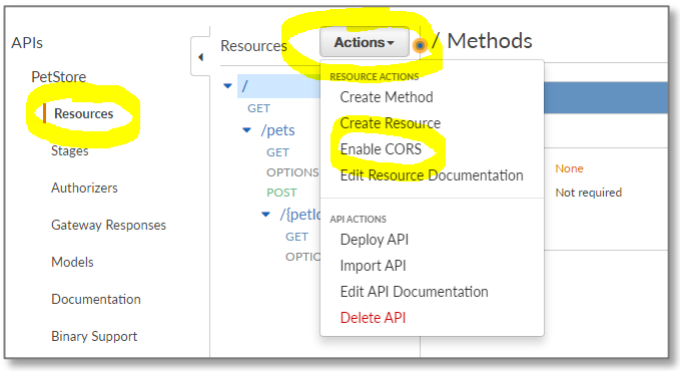
Creating a Dynamic Serverless App with S3 and API Gateway

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

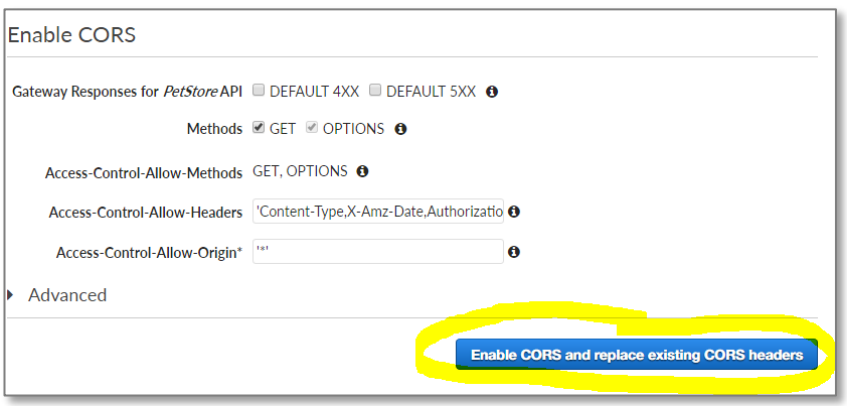
Enable Cross Origin Resource Sharing (CORS) for your API

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”



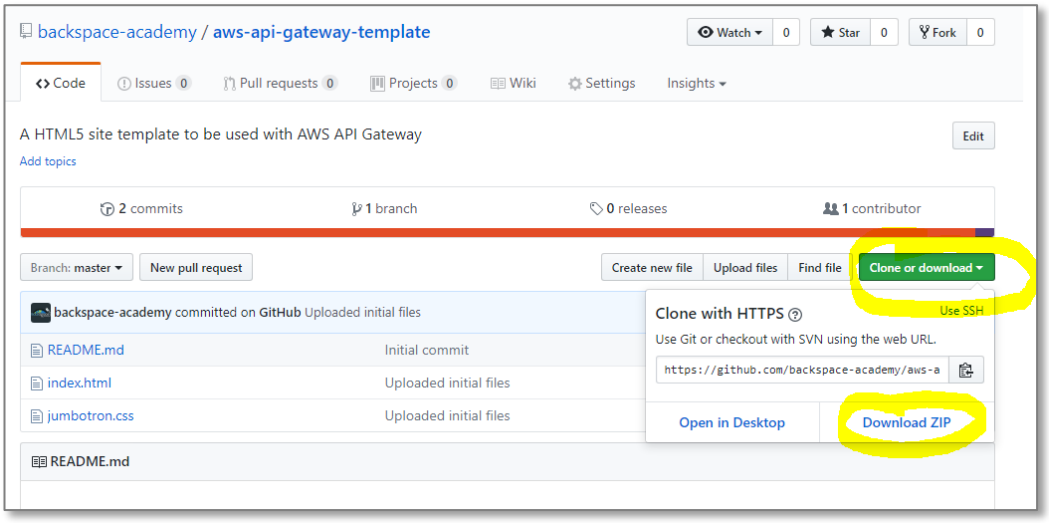
Click “Enable CORS and replace existing CORS headers”



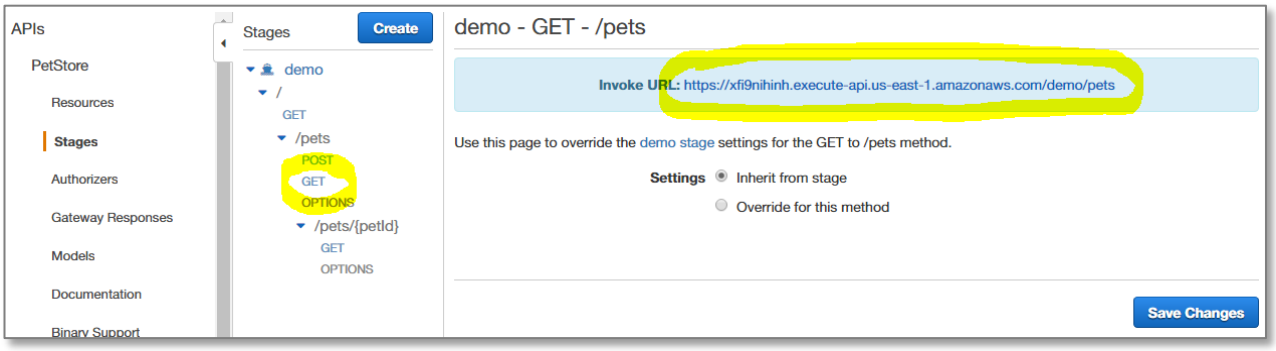
Create a Static Website

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):

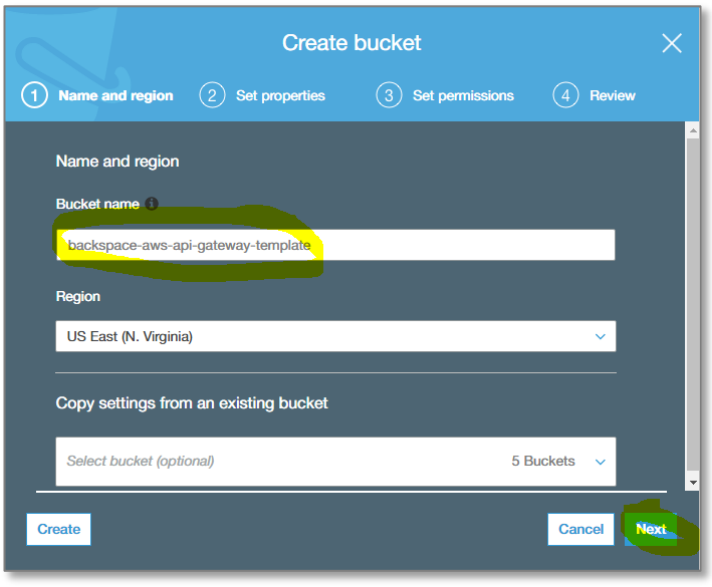


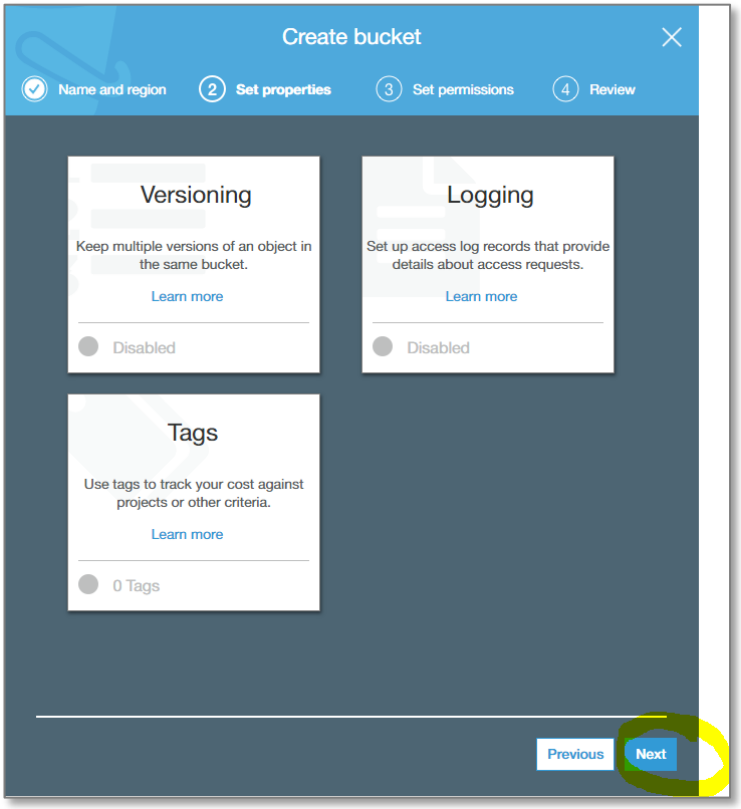

```
// Send request to API
function getPet(){
        // Put your endpoint here, not this!!!!
        var apiEndpoint = 'https://1234567890.execute-api.us-east-1.amazonaws.com/Demo/pets/';
        var petId = $('#petId').val();
        var numPets = 3;
        if (petId>numPets) {
                alert('Invalid entry - exceeds number of pets');
```

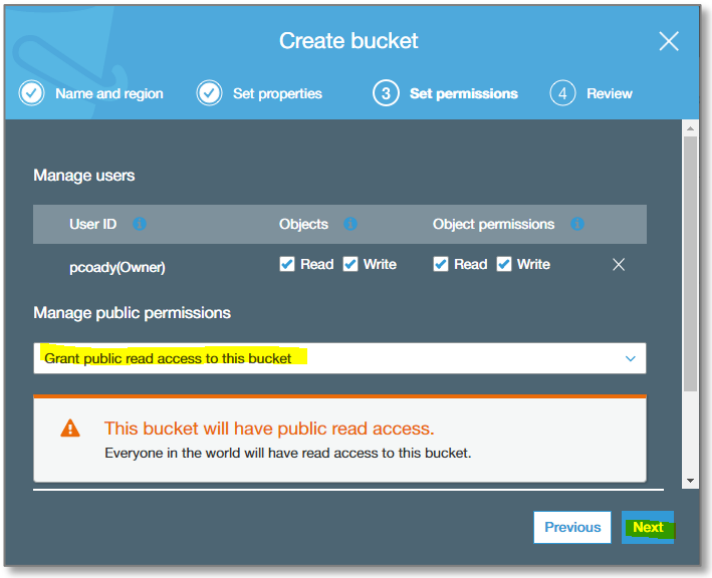
```
        }
        else if (petId<1) {
                alert('Invalid entry - must be greater than zero');
        }
        else{
                // Send the request
                var apiRequest = apiEndpoint + petId;
                $.get(apiRequest, function(data, status){
                  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

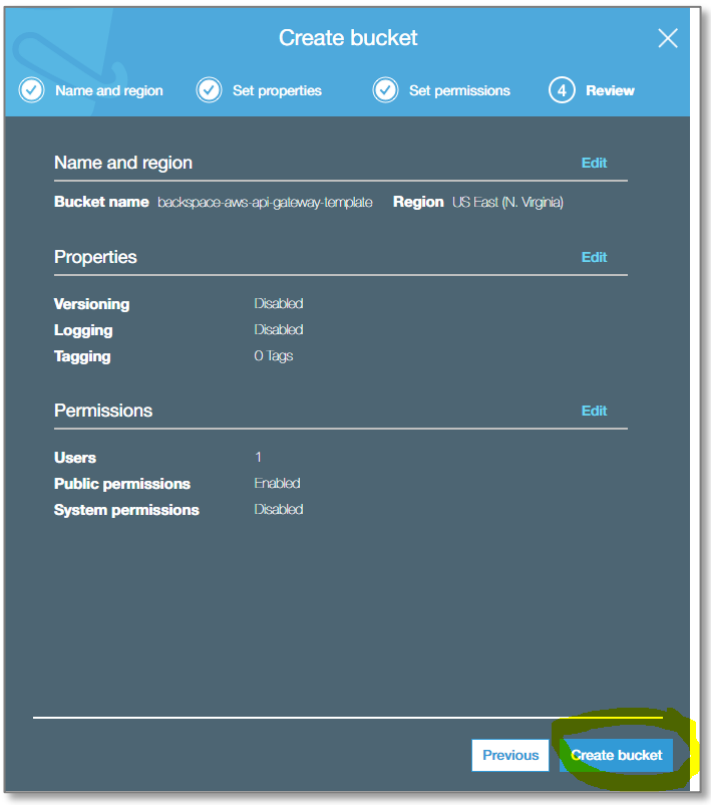
Make sure you are in US-East (N. Virginia) region. From the AWS console select “S3” from the Storage services.

Create a public bucket.

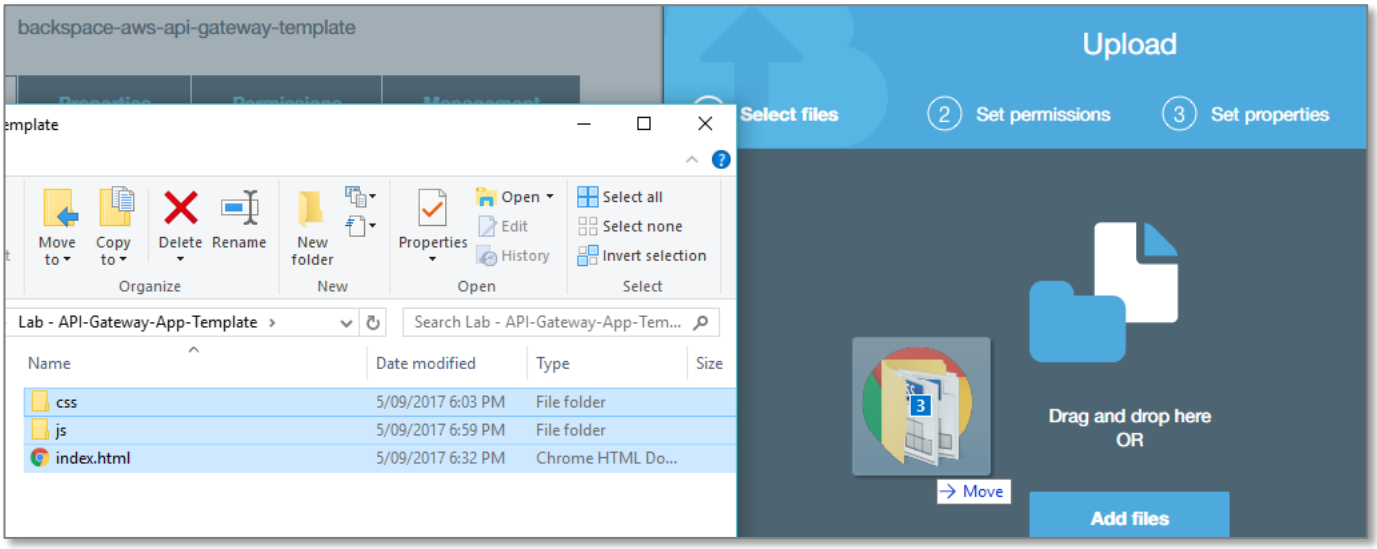




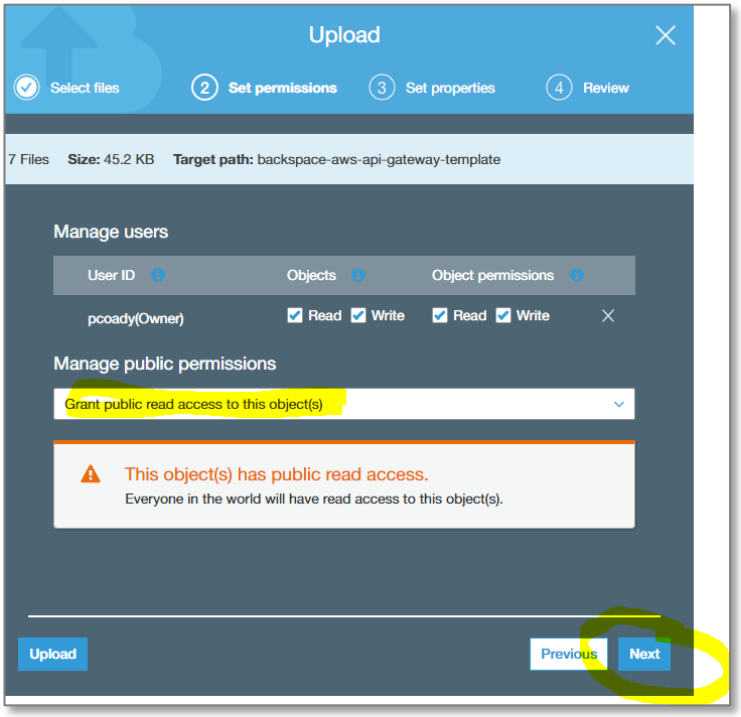




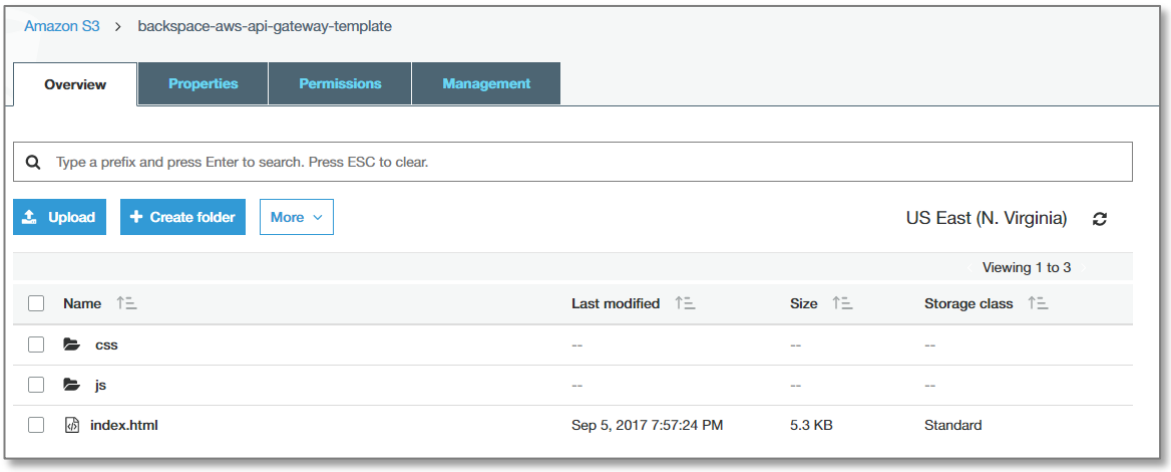
Unzip the template application and upload to the bucket (with drag and drop).



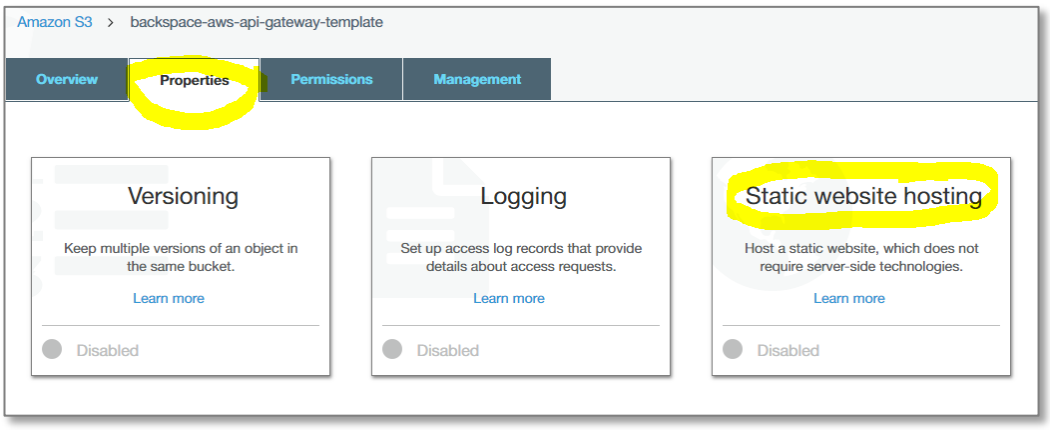
Select public permissions



Click next to start uploading

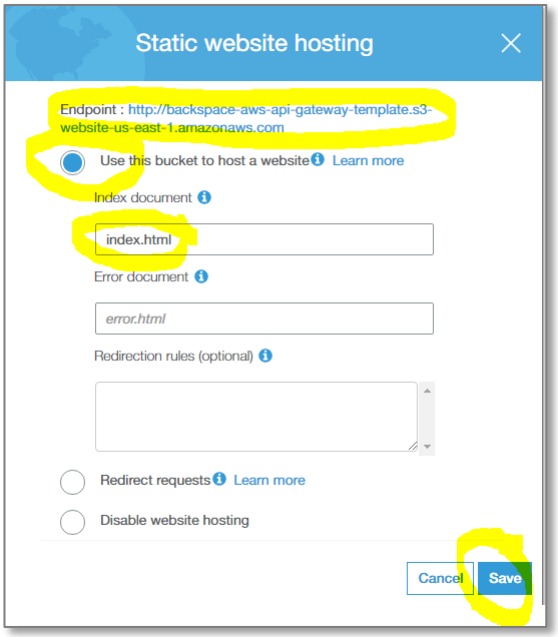


Enable static website hosting for the bucket



Copy the website endpoint

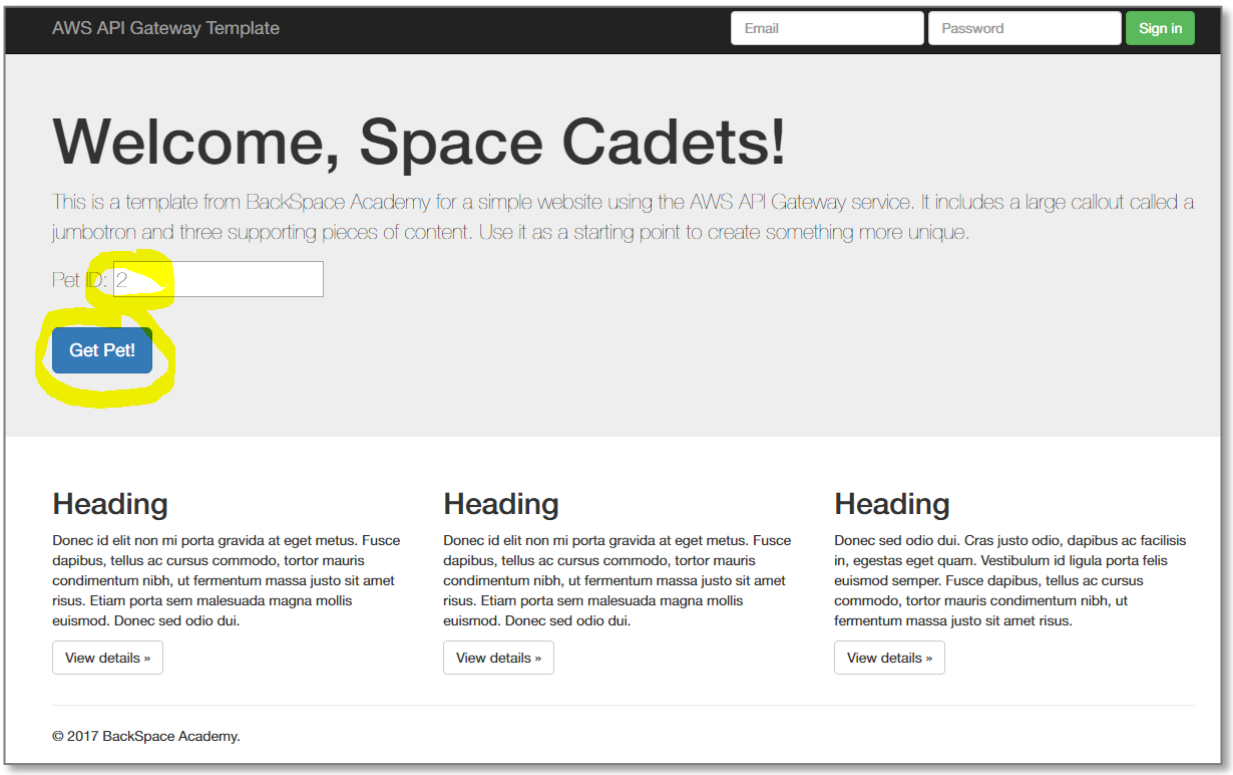
Enter the Index document



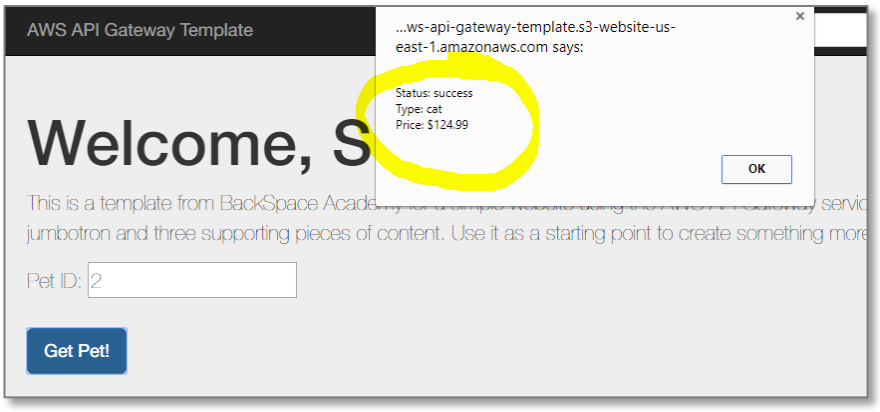
Go to the web site endpoint in your browser.

Enter a number between 1 and 3 for Pet ID

Click “Get Pet!”



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

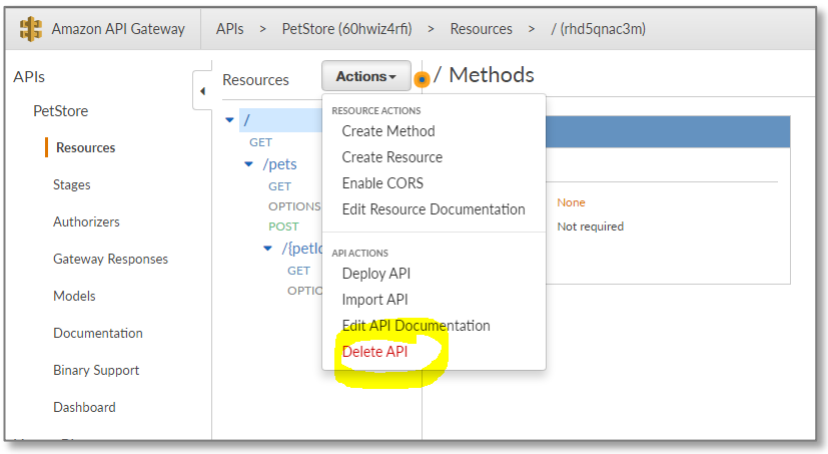


Clean Up

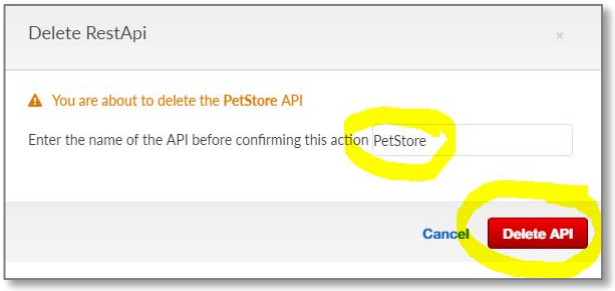
Now that we have finished the lab we can delete the resources to avoid costs.

Go back to the API Gateway console

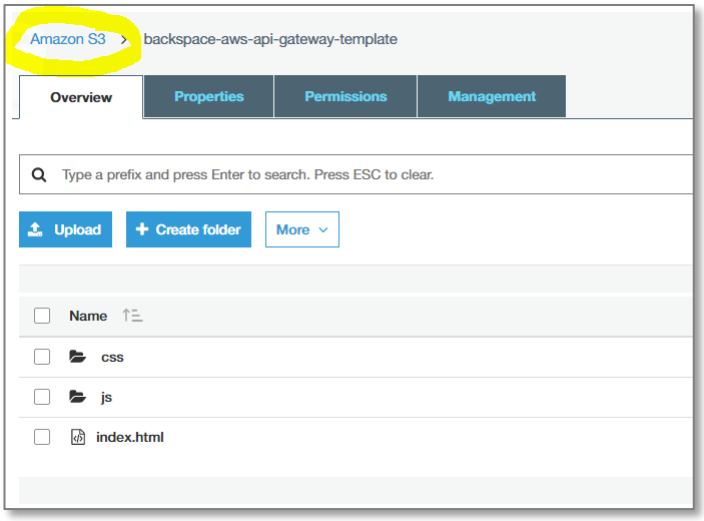
Select “Actions” – “Delete API”



Enter API name and delete



Go back to the Amazon S3 console dashboard



Select the bucket and delete