aws CERTIFIED Developer Associate

aws CERTIFIED Solutions Architect Associate

aws CERTIFIED SysOps Administrator Associate

lab

lab title

AWS API Gateway V1.01

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.

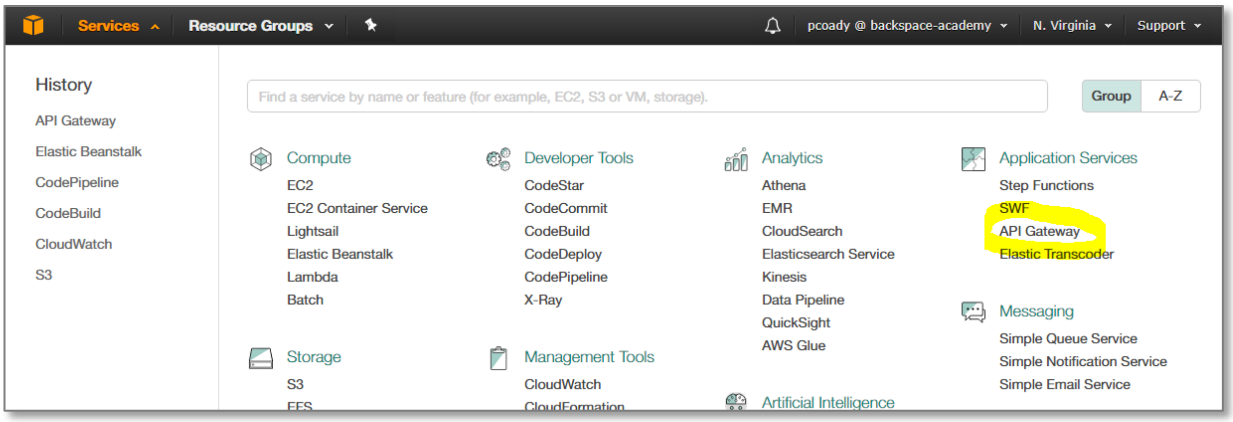
Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

Creating a REST API

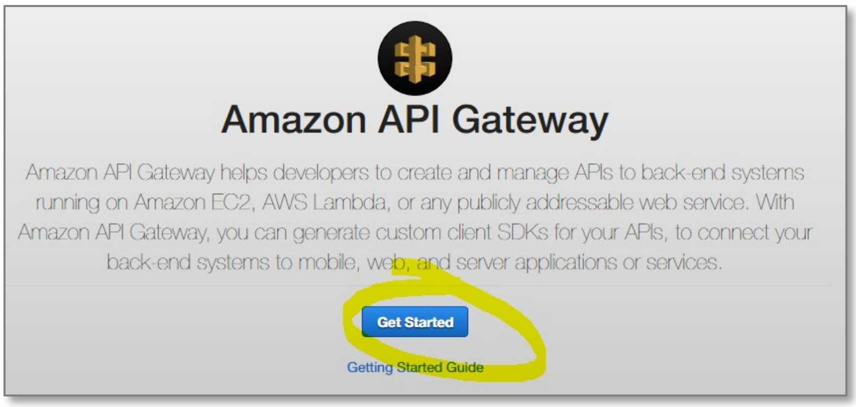
In this section, we will use the AWS API Gateway Service to create a highly available and fault tolerant REST Application Programming Interface (API).

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.



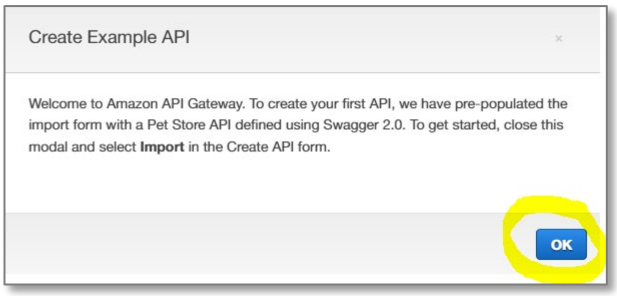
Click “Get Started”.



Amazon API Gateway

Amazon API Gateway helps developers to create and manage APIs to back-end systems running on Amazon EC2, AWS Lambda, or any publicly addressable web service. With Amazon API Gateway, you can generate custom client SDKs for your APIs, to connect your back-end systems to mobile, web, and server applications or services.

Click “OK”

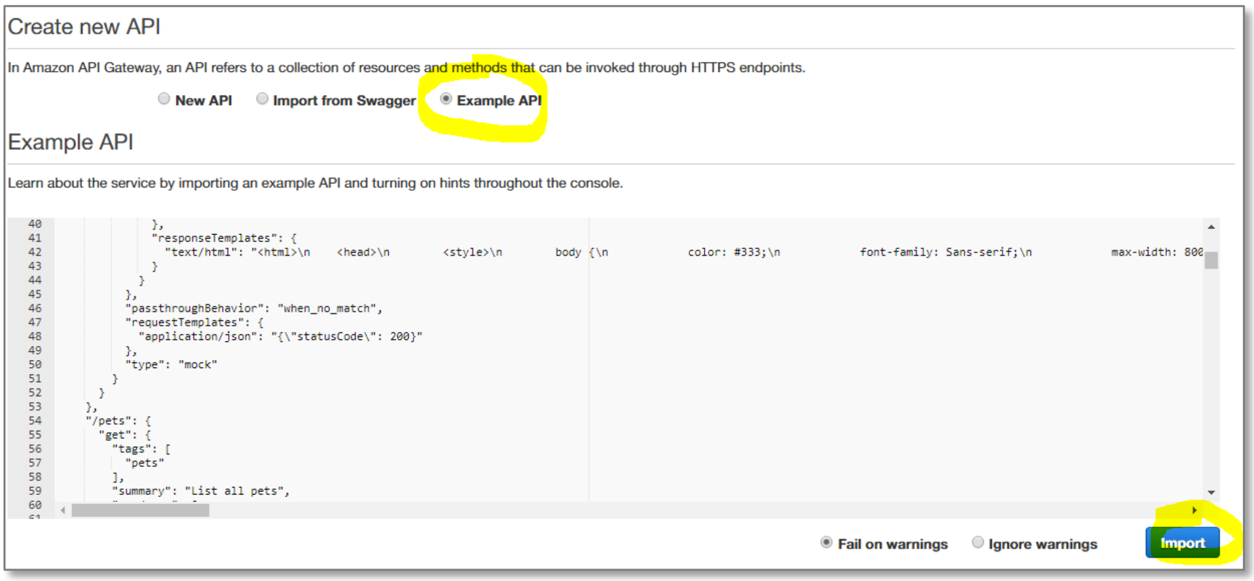


Create Example API

Welcome to Amazon API Gateway. To create your first API, we have pre-populated the import form with a Pet Store API defined using Swagger 2.0. To get started, close this modal and select **Import** in the Create API form.

Select “Example API”

Click “Import”



Create new API

In Amazon API Gateway, an API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

New API [✓ Example API] Import from Swagger

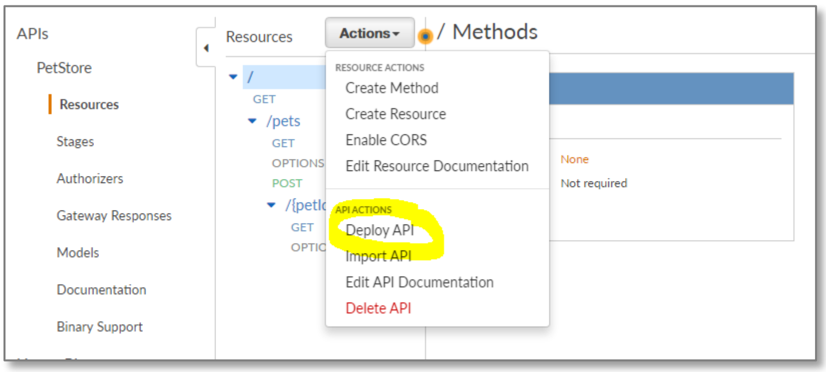
Example API

Learn about the service by importing an example API and turning on hints throughout the console.

```
            },
            "responseTemplates": {
              "text/html": "<html>\n    <head>\n        <style>\n        body {\n            color: #333;\n            font-family: Sans-serif;\n            max-width: 800
            }
          }
        },
        "passthroughBehavior": "when_no_match",
        "requestTemplates": {
          "application/json": "{\"statusCode\": 200}"
        },
        "type": "mock"
      }
    }
  },
  "/pets": {
    "get": {
      "tags": [
        "pets"
      ],
      "summary": "List all pets",
```

[✓ Fail on warnings] Ignore warnings

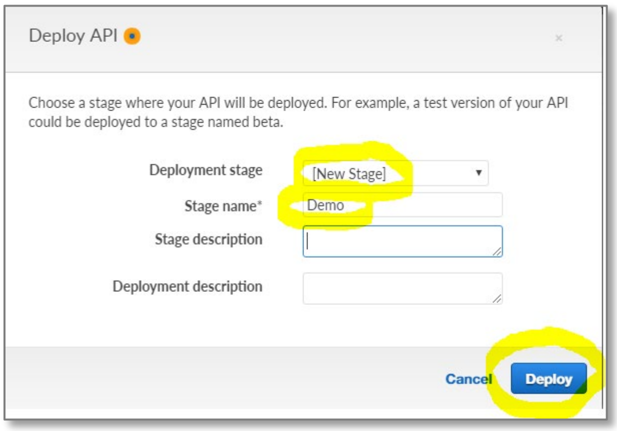
Click “Actions” – “Deploy API”



Select “Deployment stage” – “New Stage”

Give the stage a name

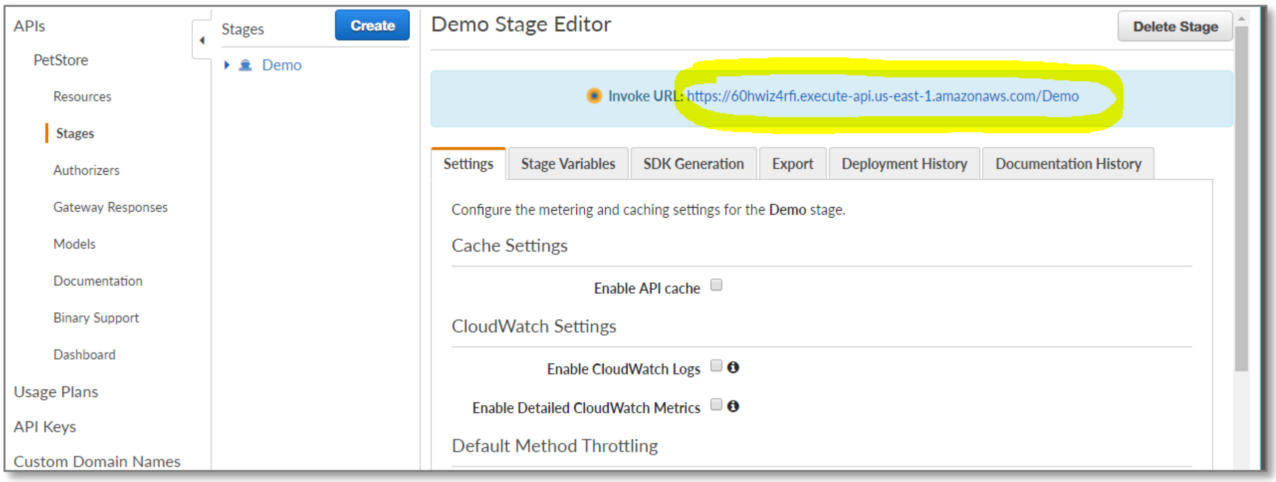
Click “Deploy”



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:

Click on Invoke URL



Demo Stage Editor

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

Configure the metering and caching settings for the **Demo** stage.

Cache Settings

Enable API cache ☐

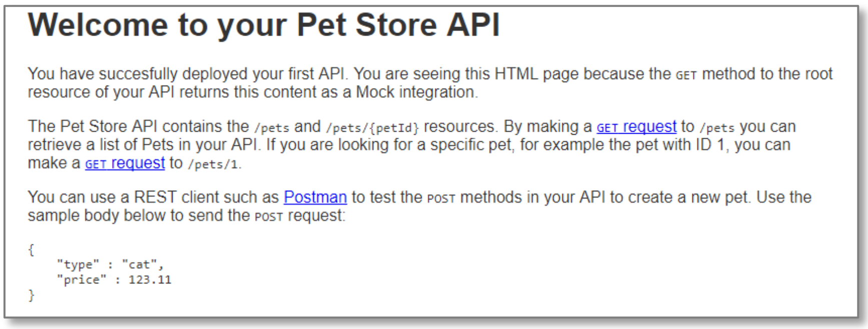
CloudWatch Settings

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 succesfully 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:

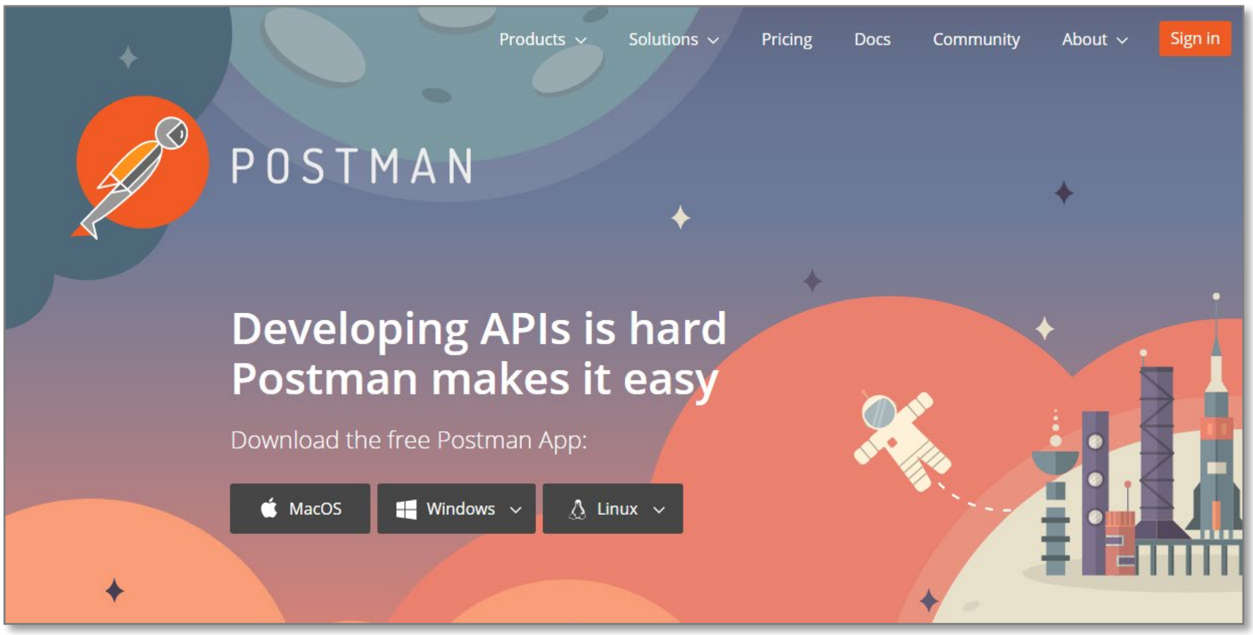
```
{
    "type" : "cat",
    "price" : 123.11
}
```

Sending HTTP Requests to your REST API

We will use PostMan to send requests to our REST API

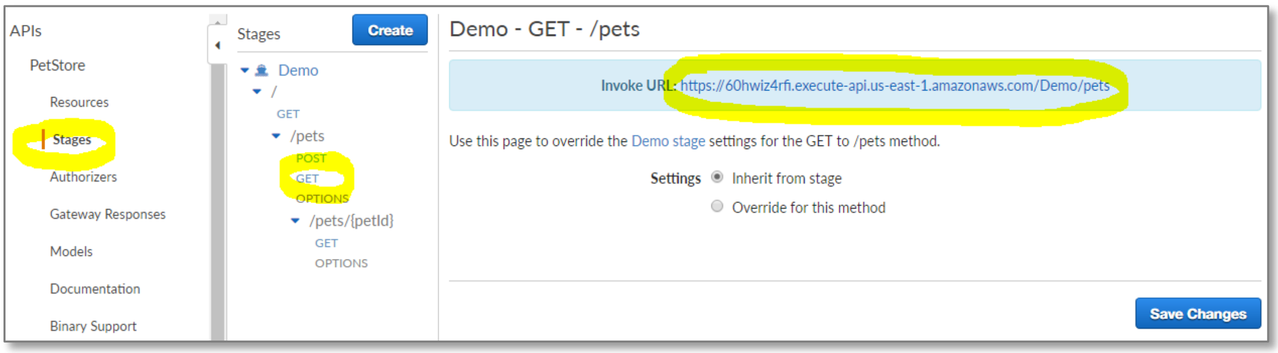
Go to the PostMan website and install PostMan

https://www.getpostman.com/



Go to the API Gateway console

Copy the Invoke URL for a GET request to /pets



Demo - GET - /pets

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

Use this page to override the Demo stage settings for the GET to /pets method.

Settings ● Inherit from stage
○ Override for this method

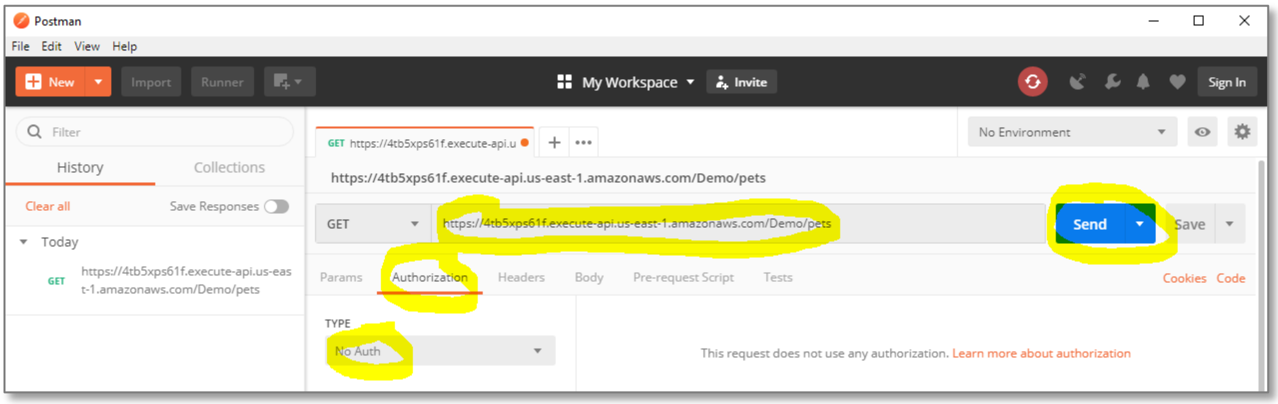
Open PostMan

Select “GET”

Paste in the URL

Click the Authorization tab and select “No Auth”

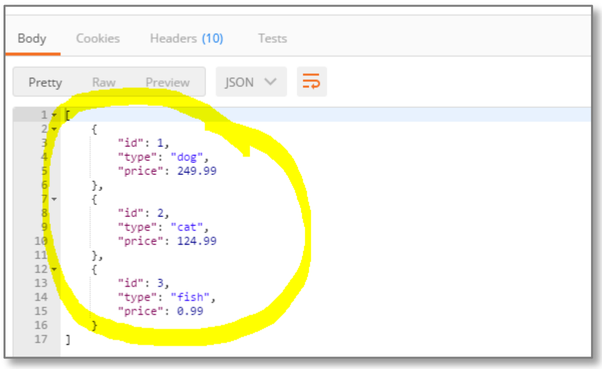
Click “Send”



GET https://4tb5xps61f.execute-api.us-east-1.amazonaws.com/Demo/pets

This request does not use any authorization. Learn more about authorization

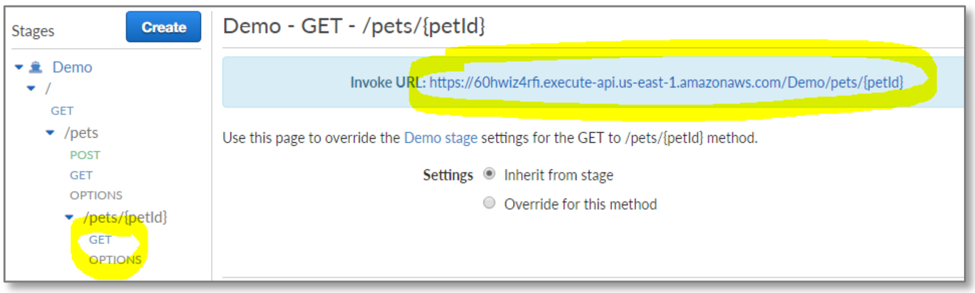
Your API will respond with JSON containing all the Pet entries.



```
[
    {
        "id": 1,
        "type": "dog",
        "price": 249.99
    },
    {
        "id": 2,
        "type": "cat",
        "price": 124.99
    },
    {
        "id": 3,
        "type": "fish",
        "price": 0.99
    }
]
```

Go Back to the API Gateway console

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



Demo - GET - /pets/{petId}

Invoke URL: https://60hwiz4rfi.execute-api.us-east-1.amazonaws.com/Demo/pets/{petId}

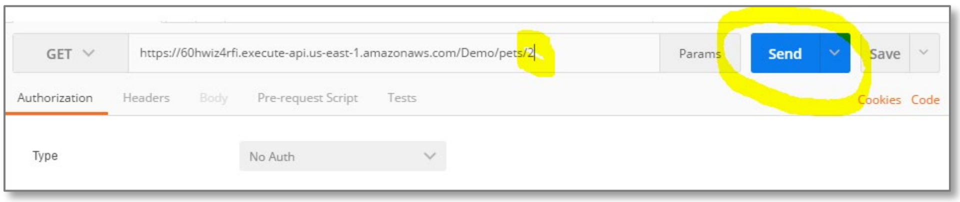
Use this page to override the Demo stage settings for the GET to /pets/{petId} method.

Settings [✓ Inherit from stage] Override for this method

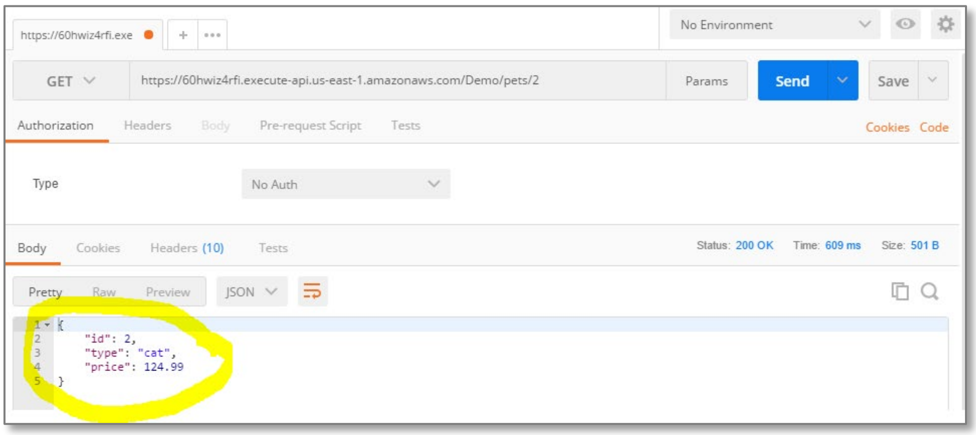
Go back to Postman

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



```
{
    "id": 2,
    "type": "cat",
    "price": 124.99
}
```

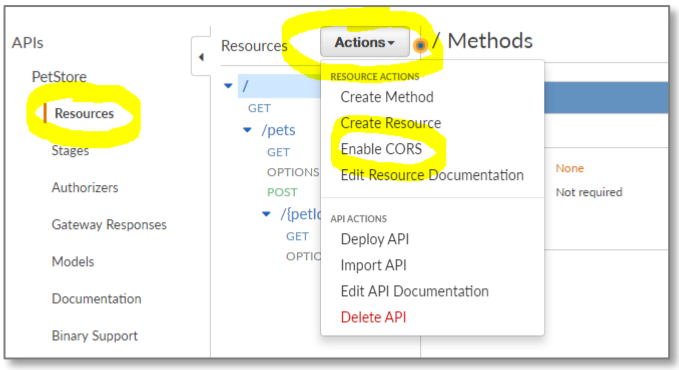
Creating a Dynamic Serverless App with S3 and API Gateway

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

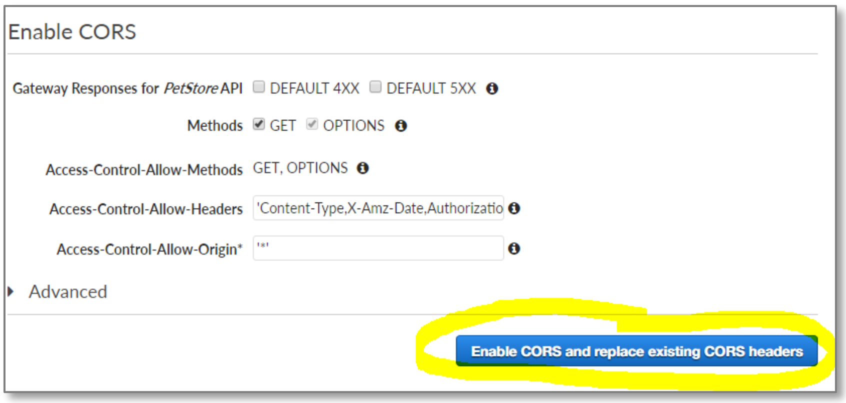
Enable Cross Origin Resource Sharing (CORS) for your API

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”


Enable CORS

Gateway Responses for *PetStore* API ☐ DEFAULT 4XX ☐ DEFAULT 5XX

Methods ☑ GET ☑ OPTIONS

Access-Control-Allow-Methods GET, OPTIONS

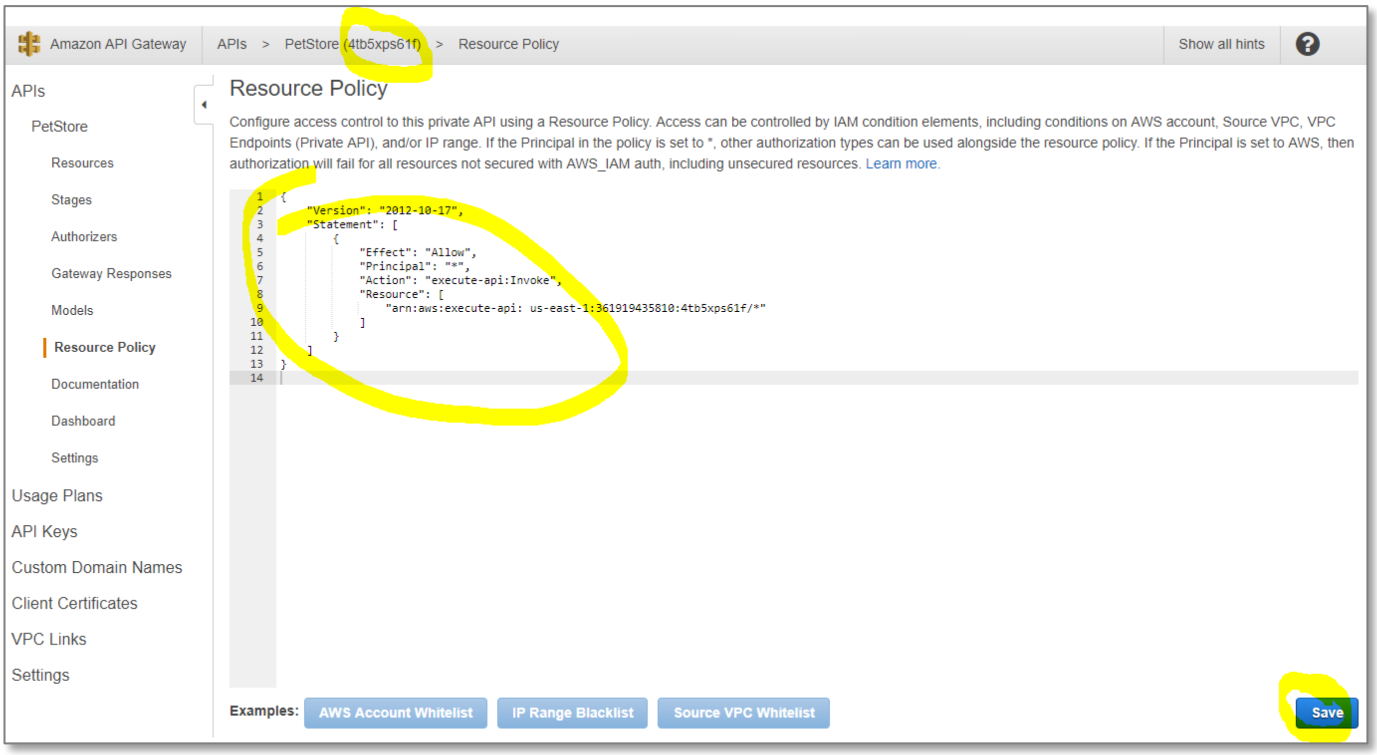
Access-Control-Allow-Headers 'Content-Type,X-Amz-Date,Authorizatio

Access-Control-Allow-Origin* '*'

Add Public Policy to Stage

Create a public policy for the API (put in your account ID and API ID)

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "execute-api:Invoke",
         "Resource": [
            "arn:aws:execute-api: us-east-1:account-id:api-id/*"
         ]
      }
   ]
}
```



Resource Policy

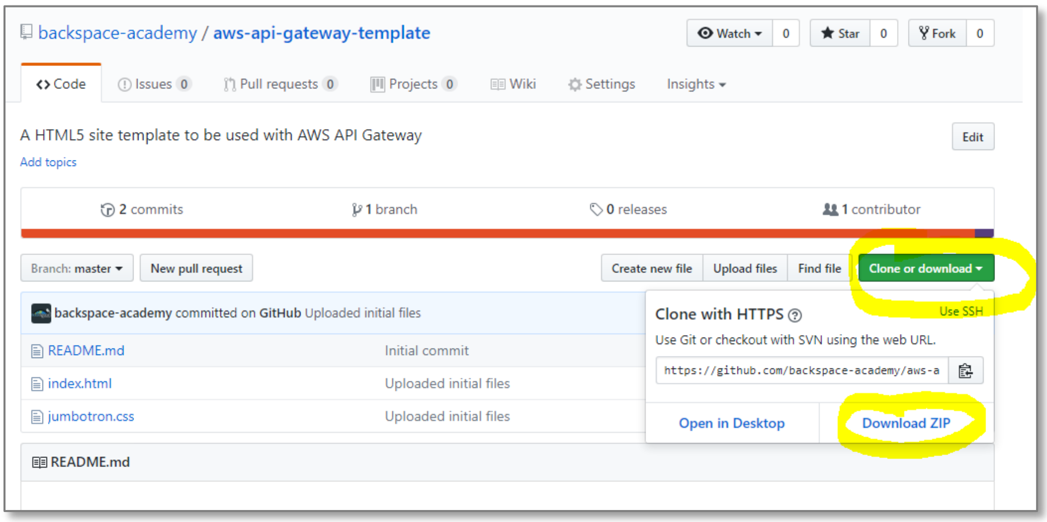
Configure access control to this private API using a Resource Policy. Access can be controlled by IAM condition elements, including conditions on AWS account, Source VPC, VPC Endpoints (Private API), and/or IP range. If the Principal in the policy is set to *, other authorization types can be used alongside the resource policy. If the Principal is set to AWS, then authorization will fail for all resources not secured with AWS_IAM auth, including unsecured resources. Learn more.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "execute-api:Invoke",
            "Resource": [
                "arn:aws:execute-api: us-east-1:361919435810:4tb5xps61f/*"
            ]
        }
    ]
}
```

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



backspace-academy / aws-api-gateway-template

A HTML5 site template to be used with AWS API Gateway

2 commits · 1 branch · 0 releases · 1 contributor

Branch: master

backspace-academy committed on GitHub Uploaded initial files

README.md	Initial commit
index.html	Uploaded initial files
jumbotron.css	Uploaded initial files

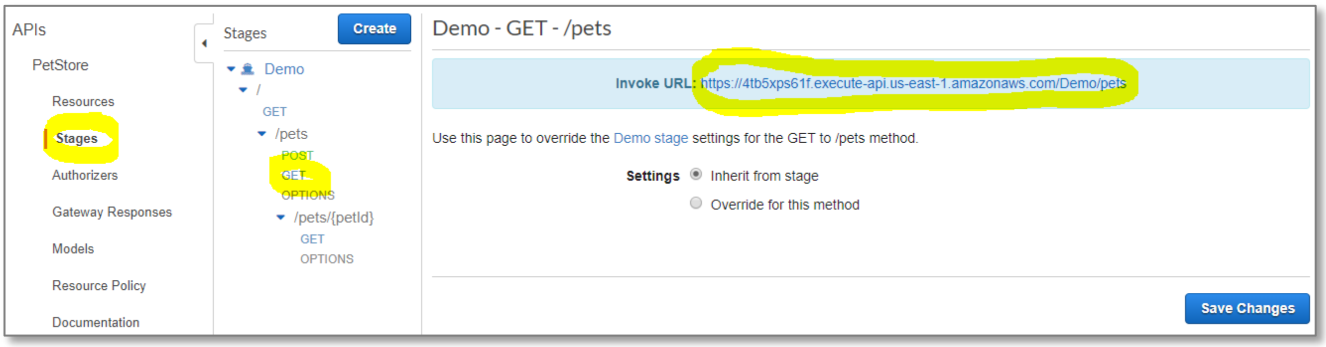
Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/backspace-academy/aws-a

Open in Desktop · Download ZIP

Open js/app.js and edit the getPet function with your API endpoint for requesting pets:



Demo - GET - /pets

Invoke URL: https://4tb5xps61f.execute-api.us-east-1.amazonaws.com/Demo/pets

Use this page to override the Demo stage settings for the GET to /pets method.

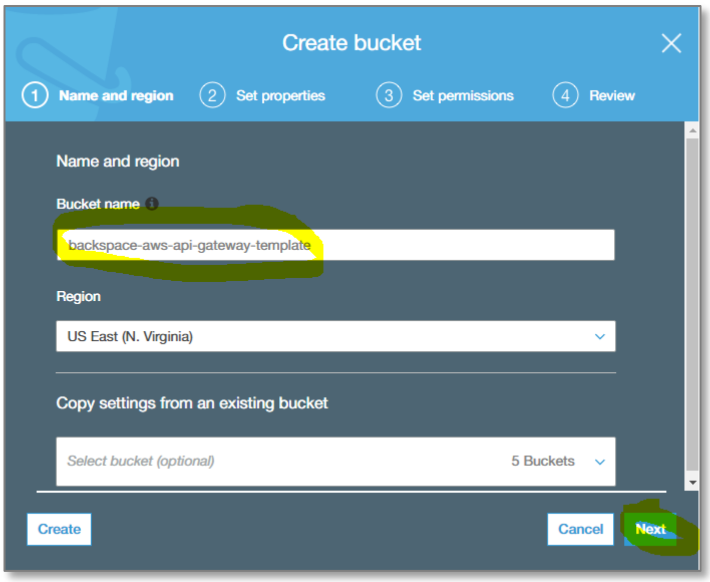
Settings: (●) Inherit from stage ( ) Override for this method

```
// 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.



Name and region

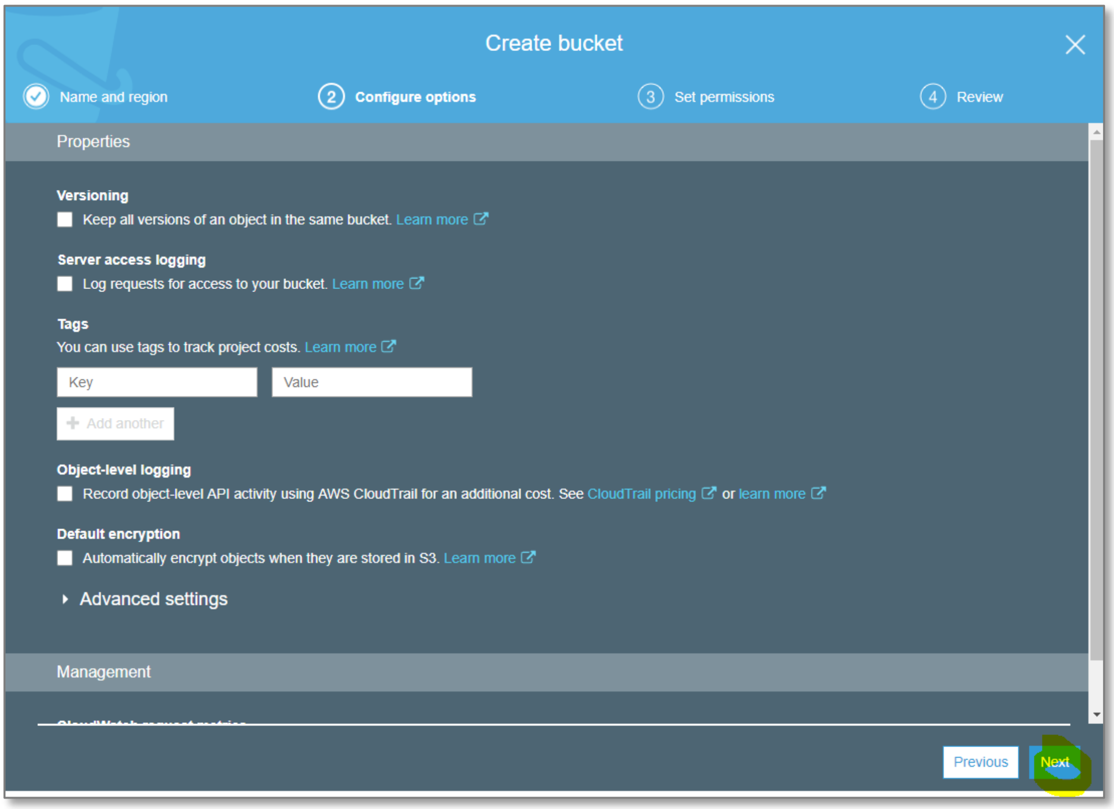
Bucket name: backspace-aws-api-gateway-template

Region: US East (N. Virginia)

Copy settings from an existing bucket

Select bucket (optional) 5 Buckets

Click *Next*



Create bucket

Name and region | 2 Configure options | 3 Set permissions | 4 Review

Properties

Versioning

☐ Keep all versions of an object in the same bucket. Learn more

Server access logging

☐ Log requests for access to your bucket. Learn more

Tags

You can use tags to track project costs. Learn more

Key | Value

Add another

Object-level logging

☐ Record object-level API activity using AWS CloudTrail for an additional cost. See CloudTrail pricing or learn more

Default encryption

☐ Automatically encrypt objects when they are stored in S3. Learn more

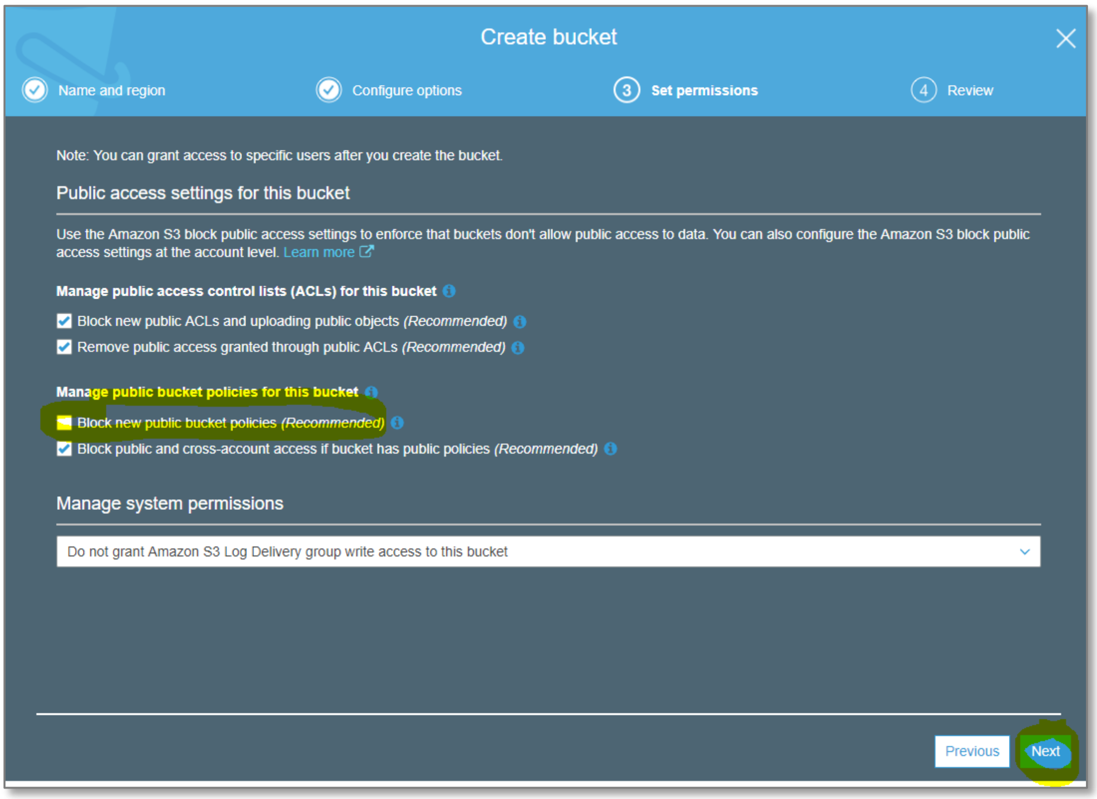
Advanced settings

Management

Previous | Next

Uncheck *Block new public bucket policies (Recommended)*

Click *Next*



Create bucket

Name and region | Configure options | 3 Set permissions | 4 Review

Note: You can grant access to specific users after you create the bucket.

Public access settings for this bucket

Use the Amazon S3 block public access settings to enforce that buckets don't allow public access to data. You can also configure the Amazon S3 block public access settings at the account level. Learn more

Manage public access control lists (ACLs) for this bucket

☑ Block new public ACLs and uploading public objects *(Recommended)*

☑ Remove public access granted through public ACLs *(Recommended)*

Manage public bucket policies for this bucket

☐ Block new public bucket policies *(Recommended)*

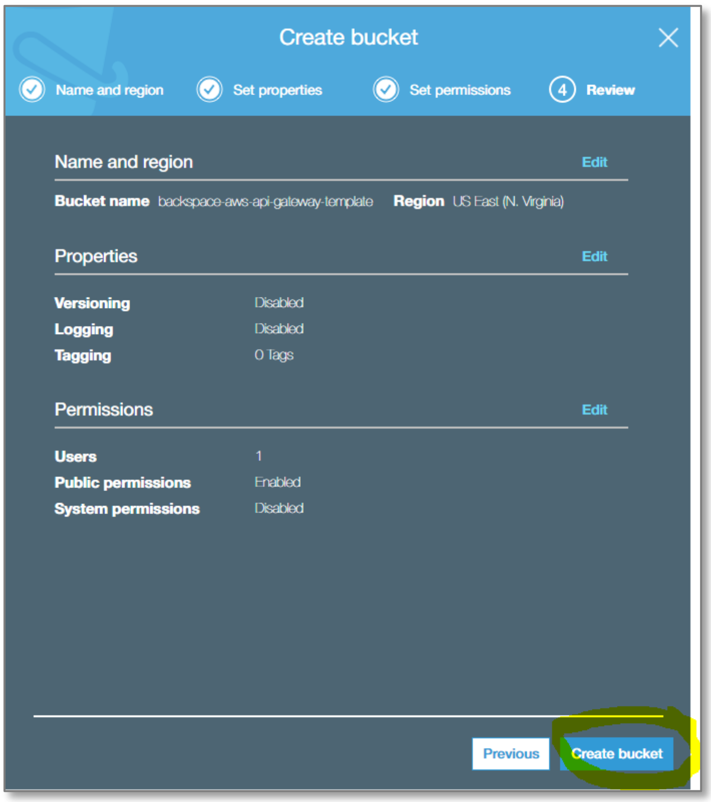
☑ Block public and cross-account access if bucket has public policies *(Recommended)*

Manage system permissions

Do not grant Amazon S3 Log Delivery group write access to this bucket

Previous | Next

Click *Create bucket*



Name and region

Bucket name backspace-aws-api-gateway-template **Region** US East (N. Virginia)

Properties

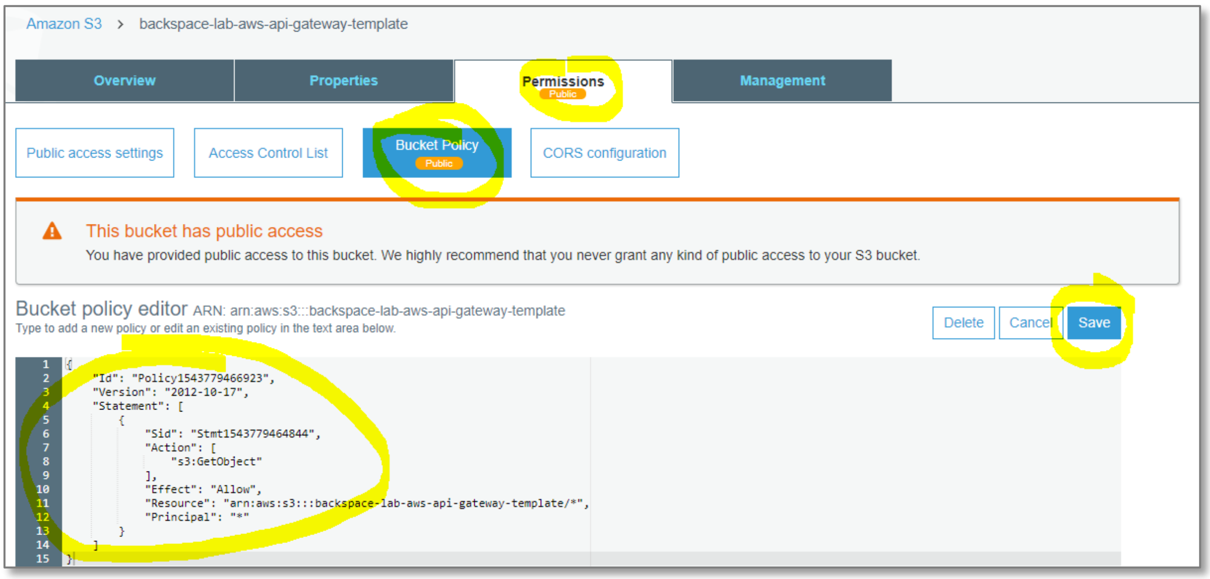
Versioning	Disabled
Logging	Disabled
Tagging	0 Tags

Permissions

Users	1
Public permissions	Enabled
System permissions	Disabled

Add a public bucket policy (change bucket name to yours)

```
{
  "Id": "Policy1543779466923",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1543779464844",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::your_bucket_name_goes_here/*",
      "Principal": "*"
    }
  ]
}
```


Amazon S3 › backspace-lab-aws-api-gateway-template

This bucket has public access

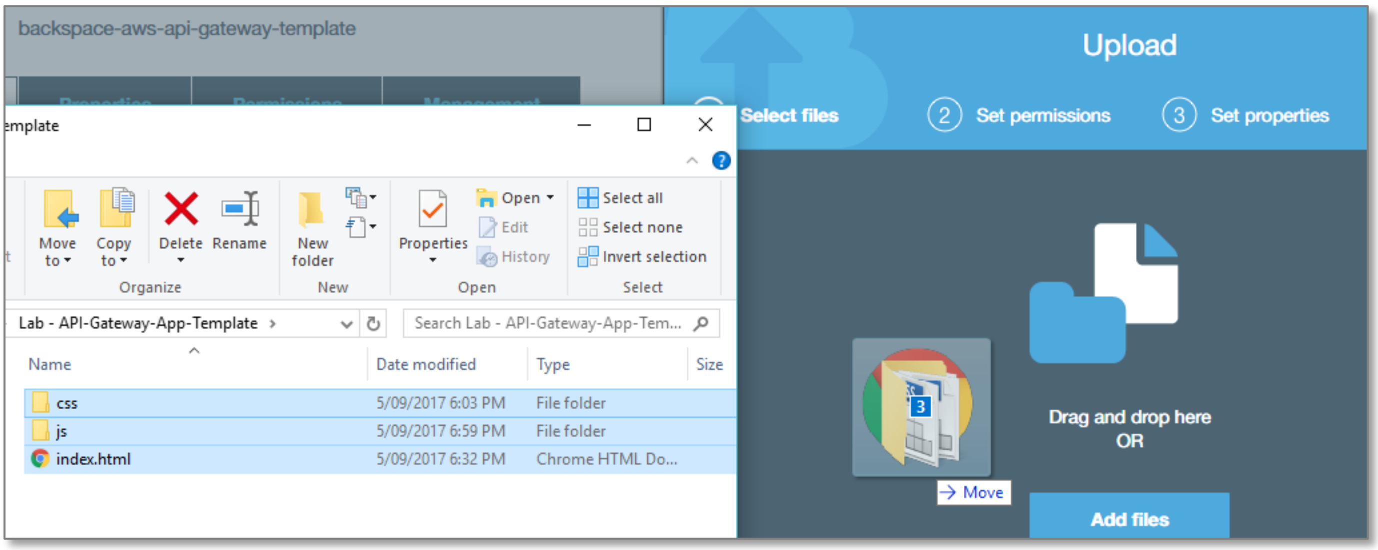
You have provided public access to this bucket. We highly recommend that you never grant any kind of public access to your S3 bucket.

Bucket policy editor ARN: arn:aws:s3:::backspace-lab-aws-api-gateway-template

Type to add a new policy or edit an existing policy in the text area below.

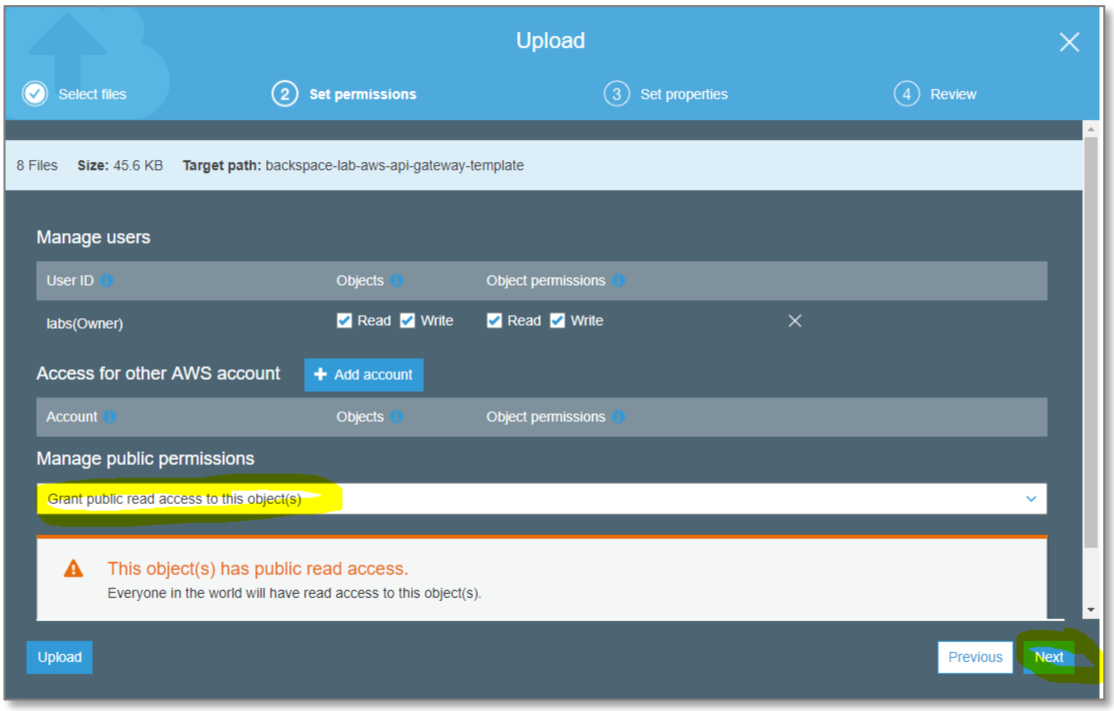
```
{
    "Id": "Policy1543779466923",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1543779464844",
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::backspace-lab-aws-api-gateway-template/*",
            "Principal": "*"
        }
    ]
}
```

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



Name	Date modified	Type	Size
css	5/09/2017 6:03 PM	File folder	
js	5/09/2017 6:59 PM	File folder	
index.html	5/09/2017 6:32 PM	Chrome HTML Do...	

Select public permissions



Upload

Select files | 2 Set permissions | 3 Set properties | 4 Review

8 Files **Size:** 45.6 KB **Target path:** backspace-lab-aws-api-gateway-template

Manage users

User ID	Objects	Object permissions
labs(Owner)	✓ Read ✓ Write	✓ Read ✓ Write

Access for other AWS account + Add account

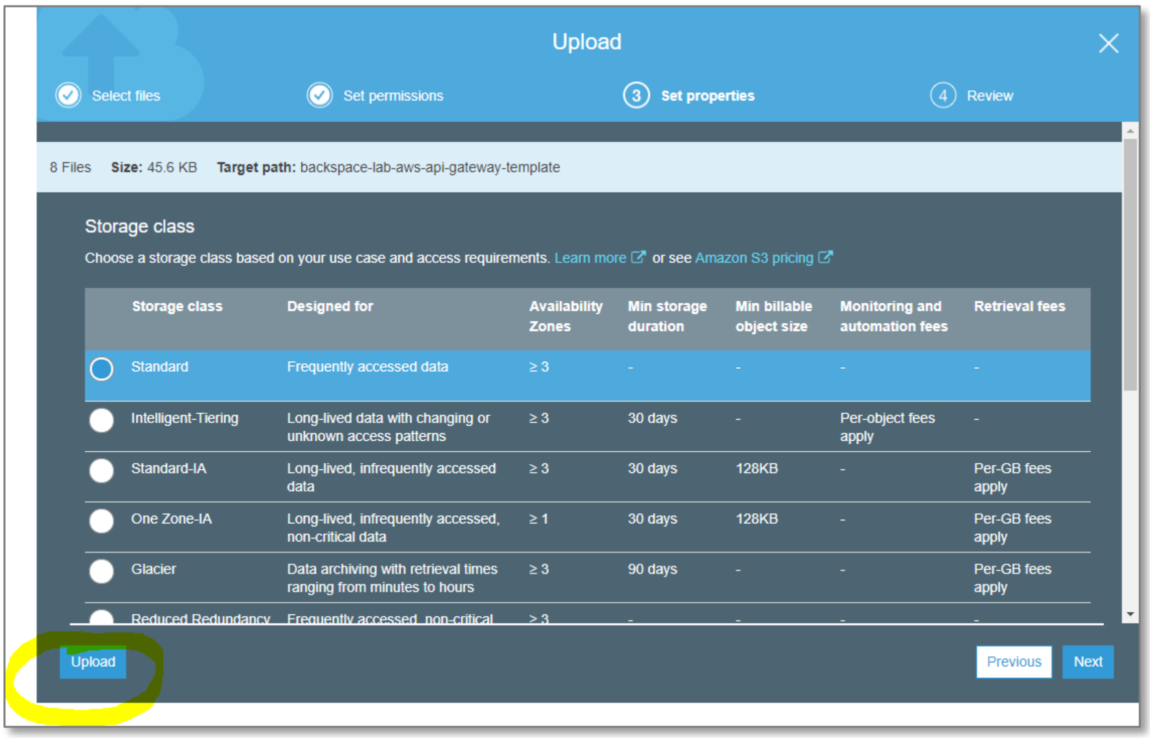
Account	Objects	Object permissions

Manage public permissions

Grant public read access to this object(s)

This object(s) has public read access.
Everyone in the world will have read access to this object(s).

Click *Upload* to start uploading



Upload

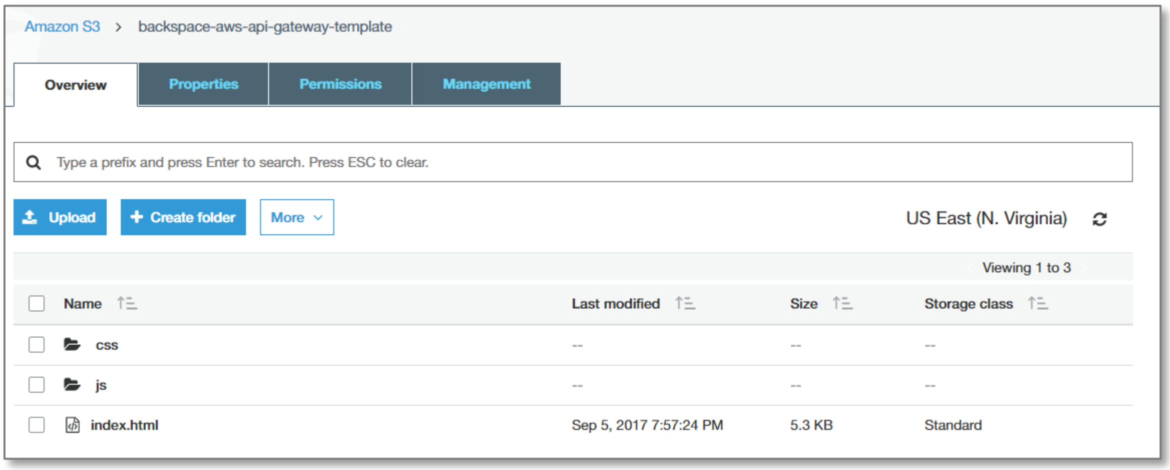
Select files | Set permissions | 3 Set properties | 4 Review

8 Files **Size:** 45.6 KB **Target path:** backspace-lab-aws-api-gateway-template

Storage class

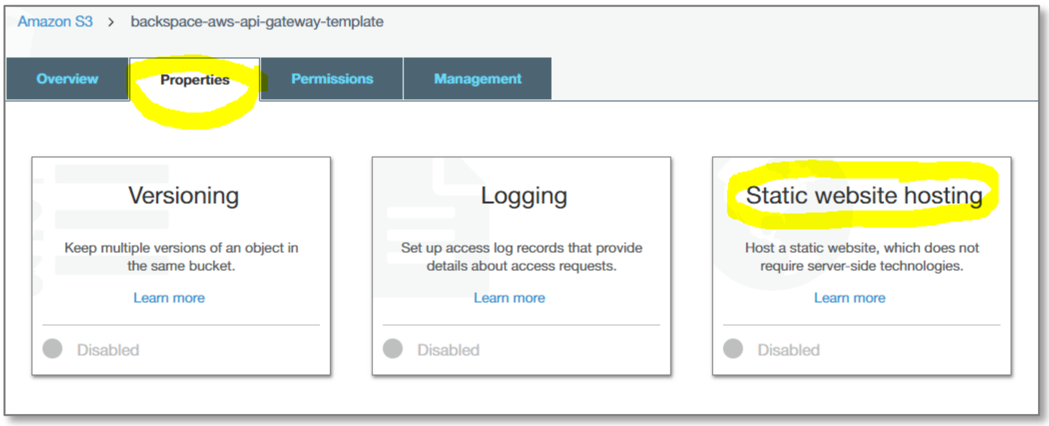
Choose a storage class based on your use case and access requirements. Learn more or see Amazon S3 pricing

	Storage class	Designed for	Availability Zones	Min storage duration	Min billable object size	Monitoring and automation fees	Retrieval fees
●	Standard	Frequently accessed data	≥ 3	-	-	-	-
○	Intelligent-Tiering	Long-lived data with changing or unknown access patterns	≥ 3	30 days	-	Per-object fees apply	-
○	Standard-IA	Long-lived, infrequently accessed data	≥ 3	30 days	128KB	-	Per-GB fees apply
○	One Zone-IA	Long-lived, infrequently accessed, non-critical data	≥ 1	30 days	128KB	-	Per-GB fees apply
○	Glacier	Data archiving with retrieval times ranging from minutes to hours	≥ 3	90 days	-	-	Per-GB fees apply
○	Reduced Redundancy	Frequently accessed, non-critical	≥ 3	-	-	-	


Amazon S3 > backspace-aws-api-gateway-template
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
US East (N. Virginia)
Viewing 1 to 3

Name	Last modified	Size	Storage class
css	--	--	--
js	--	--	--
index.html	Sep 5, 2017 7:57:24 PM	5.3 KB	Standard

Enable static website hosting for the bucket



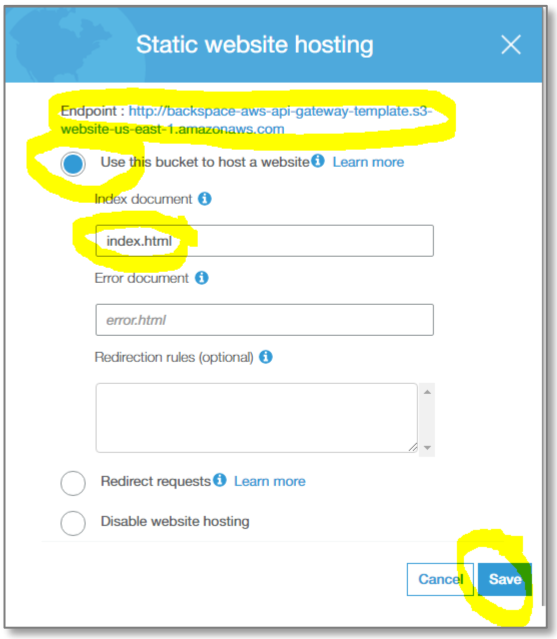
Versioning
Keep multiple versions of an object in the same bucket.
Learn more
Disabled

Logging
Set up access log records that provide details about access requests.
Learn more
Disabled

Static website hosting
Host a static website, which does not require server-side technologies.
Learn more
Disabled

Copy the website endpoint

Enter the Index document



Endpoint : http://backspace-aws-api-gateway-template.s3-website-us-east-1.amazonaws.com

[✓ Use this bucket to host a website] Learn more

Index document: index.html

Error document: error.html

Redirection rules (optional)

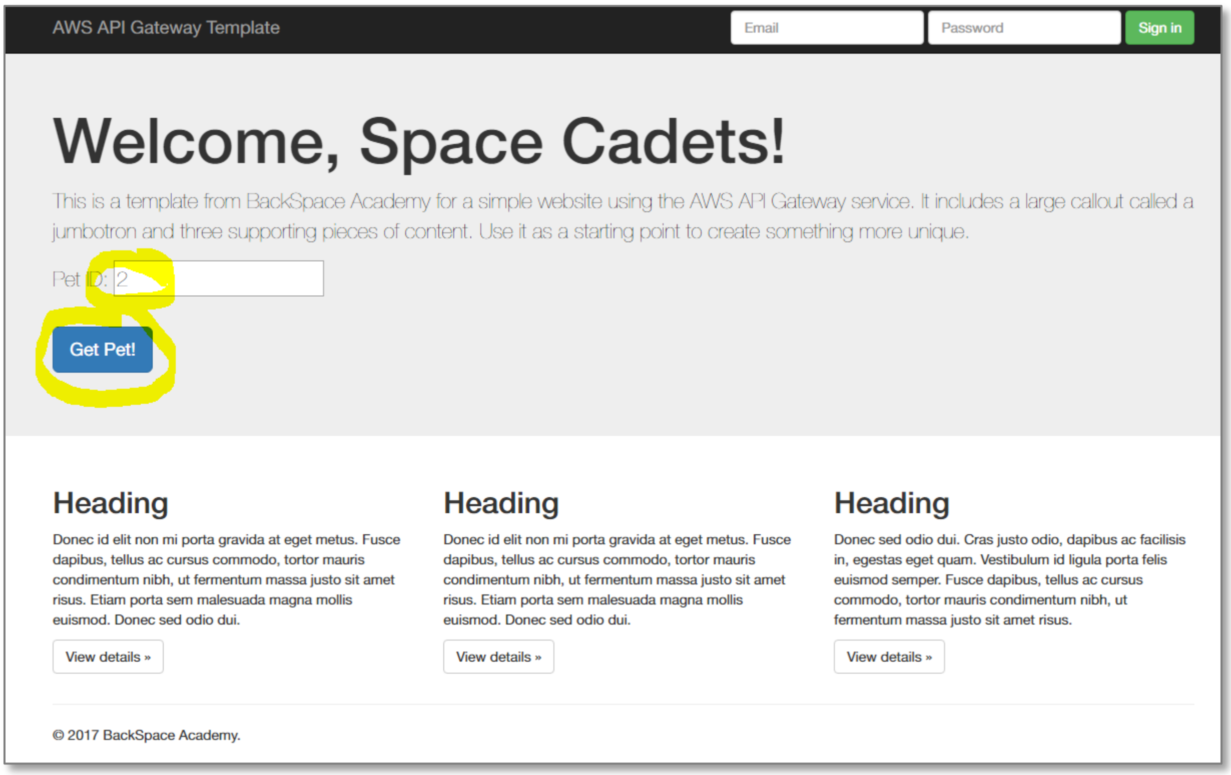
Redirect requests Learn more

Disable website hosting

Go to the web site endpoint in your browser.

Enter a number between 1 and 3 for Pet ID

Click “Get Pet!”



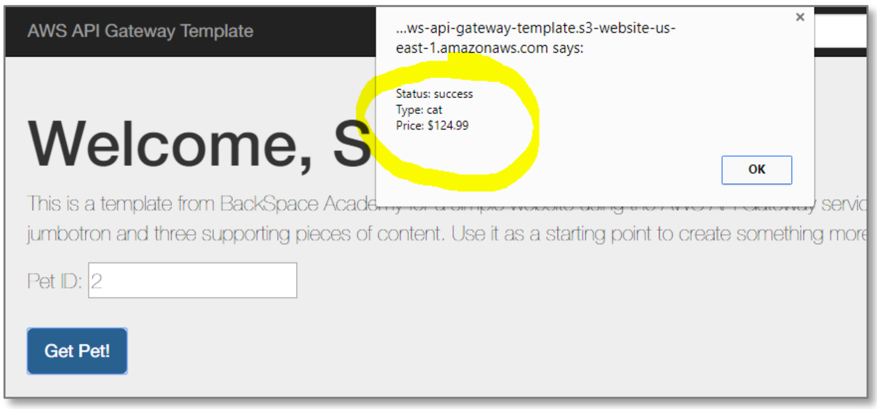
This is a template from BackSpace Academy for a simple website using the AWS API Gateway service. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

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.

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.

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.

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



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

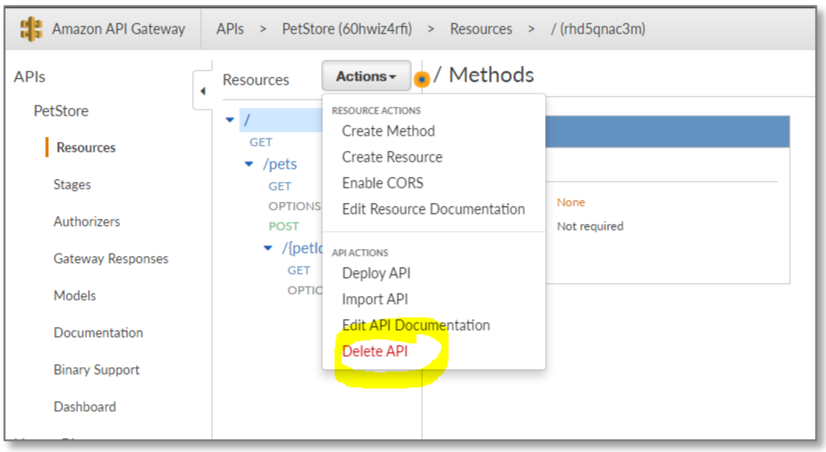
Status: success
Type: cat
Price: $124.99

Clean Up

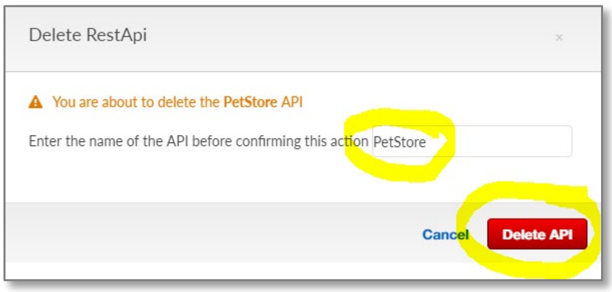
Now that we have finished the lab we can delete the resources to avoid costs.

Go back to the API Gateway console

Select “Actions” – “Delete API”



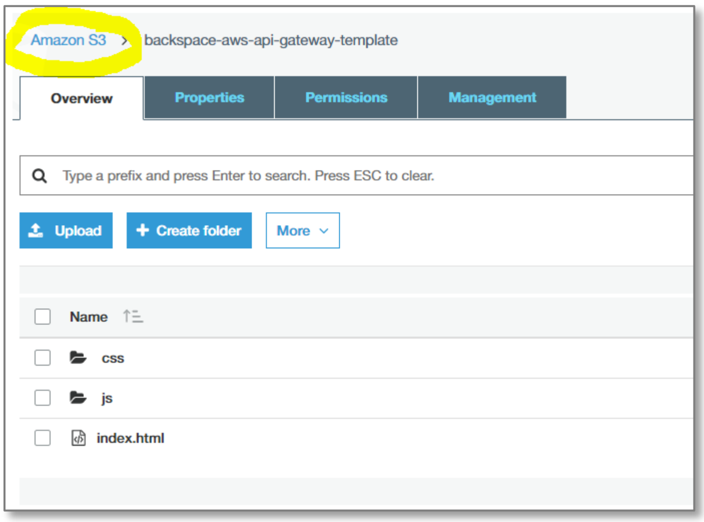
Enter API name and delete



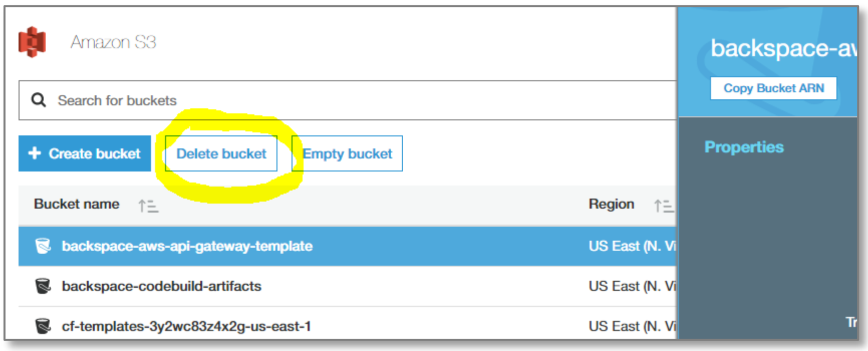
You are about to delete the PetStore API

Enter the name of the API before confirming this action PetStore

Go back to the Amazon S3 console dashboard


Amazon S3 > backspace-aws-api-gateway-template
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
Name
css
js
index.html

Select the bucket and delete



Bucket name	Region
backspace-aws-api-gateway-template	US East (N. Vi
backspace-codebuild-artifacts	US East (N. Vi
cf-templates-3y2wc83z4x2g-us-east-1	US East (N. Vi