# AWS Starter Kit

# Introduction:

Welcome to Hack@Brown's cloud starter kit. We will be using Amazon Web Services (AWS) as our cloud platform. This guide will take you through three different services and explain what they are and how they can be used.

# Overview:

This is a quick overview of the three services to give you an idea what each one is about.

- **S3**
  - S3 is an online, bulk storage service for all types of files that are accessible from most devices.
- **EC2**
  - EC2 is a virtual computing environment, known as *instances.* It is essentially a virtual server that is run on Amazon's cloud resources.
- **Lambda**
  - AWS Lambda is a compute service that lets you run code without provisioning or managing servers.

# Setup:

The first step is to create a free account on AWS. Follow this link and the steps that they tell you to set up a Free AWS account.

*Note that it will ask you for a payment method but it will not charge you as long as you do not exceed the free trial.

# S3:

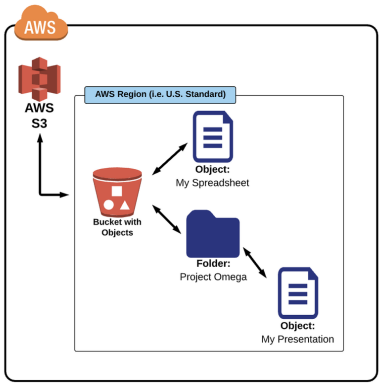
## What is S3?

As the name implies, S3 really is a simple storage service. Generally, it is an online, bulk storage service for all types of files that are accessible from most devices.

In Amazon's words, S3 is a "simple web service that you can use to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any user access to the same highly scalable, reliable, fast, inexpensive infrastructure that Amazon uses to run its own global network of websites. The service aims to maximize benefits of scale and to pass those benefits on to users."

The free tier of S3 includes 5GB of Amazon S3 standard storage, 20,000 get requests, 2,000 put requests, and 15GB of data transfer out each month for one year.

## Navigating the AWS console




You can access S3 under Storage & Content Delivery of AWS. S3 is made up of buckets, folders, and objects. The lowest, or root level folders are called buckets. Within buckets, you can store objects or subfolders, which can also store objects. Objects are just files, and can be stored in buckets *or* subfolders.

## Creating a bucket:

Bucket names must:

- Be 3 to 62 characters long
- Be unique across ALL of AWS
- Only contain lowercase letters, numbers, and hyphens
- Not have the format of an IP address (as in 123.456.7.8)

When you create a bucket from the S3 console, you must specify the geographic region that it will occupy. Any data uploaded to that bucket will be stored in a data center in that region. The best practice is to select the region that is located physically closest to where you are. However, if you are creating a service to provide data to customers in a different region, you should create the bucket in that region.

## Creating a folder within a bucket

Click on the bucket name in your list of buckets. Click the 'Create Folder' button in the top left corner of the page (to the right of the 'Upload button'. You will then be prompted to name this Folder, which does NOT need to follow the same naming conventions as buckets do.

## Uploading/importing an object into a bucket or a folder:

Click on the bucket name in your list of buckets. If you want to add an object into a folder, click on the folder name, but if you want to upload into the bucket itself, stay in the main bucket.

Click the ‘Upload’ button in the top left corner on the page, then press ‘Add Files’ on the window that will pop up. This will prompt you to select files or other objects from your computer. If you want to change properties of the file, select ‘Set Permissions’ in the bottom right corner of the upload window. In this window, you can choose a different storage class (see section below).

Click ‘Start Upload’ in the bottom right corner of the upload window.

## Object Durability and Availability

Durability is the percent chance over one year that a file stored in S3 will NOT be lost, corrupted, or deleted. Availability is the percentage over a one year time period that an object stored in S3 will be accessible (able to be downloaded, modified, etc.).

## Viewing Properties

In your list of all buckets, select a bucket without navigating into it by clicking in the horizontal space to the right of the bucket name. Near the top right corner, select the ‘Properties’ tab. Note: you can also view the properties of a folder or an object.

The top section of Properties displays general information about the bucket, including its name, region, creation date, and owner.

## Important Properties

### Storage Classes

The general classification assigned to an object in S3. Each storage class differs in storage cost, object availability, object durability, and frequency of access to the object. Each object is assigned the standard class by default, but you can change between Standard, RRS, and S3-IA in the ‘Details’ section of properties (see above how to view properties). You can also change a folder’s storage class to change the storage class of every object inside.

- **Standard:** Although it is the default class, it is the most expensive. It has 99.999999999% object durability and 99.99% object availability. Best for objects that you frequently access (on a daily basis) or objects that are extremely important and need high durability.
- **Reduced Redundancy Storage (RRS):** Designed for non-critical, reproducible objects and is less expensive than Standard. This is because it only has 99.99% object durability, though it also has 99.9% object availability. Best for backups or files that can be reproduced.
- **Infrequent Access (S3-IA):** Designed for objects that are not accessed as frequently, but need to be immediately available when you do need access. Like Standard, it has 99.999999999% durability, but only 99.90% availability. It is less expensive than both Standard and RRS. Best for objects that you need infrequently, but need a guarantee of quick access when you need it.

- **Glacier:** Designed for long-term storage, and as the name implies, it may take hours for objects stored in Glacier to be retrieved. It has 99.999999999% durability and is the cheapest S3 storage class. Best for large, archival storage. To move an object to the Glacier class, you need to use object life cycles. The change to Glacier may take 1-2 days.

### Object Lifeycles

A set of rules that automate the migration of an object's storage classes to a different storage class (or deletion), based on a specific time interval. By using an object lifecycle, you can automate the process of changing a file's storage class to meet your usage needs (that you know will change overtime) in order to minimize your S3 storage costs.

- To create a lifecycle policy, navigate to the bucket's properties (see section above) and select the 'Lifecycle' dropdown.
- Click 'Add Rule' and specify whether you want to apply the rule/policy to the entire bucket, or just certain folders or objects within the bucket.
- Select the transitions (between storage classes or deletion) you would like your objects/files to make and the time intervals upon which those transitions should be made.

### Permissions

Allow you to control over who can view, access, and use specific buckets and objects.

- Bucket permissions:
  - Navigate to the bucket's properties (see section above) and select the 'Permissions' dropdown.
  - Click 'Add more permissions' and select which parties these new permissions will apply to.
  - Then, select whether these parties should be able to list (see a list of folders and objects in this bucket), upload/delete (modify the folders and objects in this bucket) and/or view permissions.
- Object permissions:
  - Navigate to the object's properties (see section above) and select the 'Permissions' dropdown.
  - Click 'Add more permissions' and select which parties these new permissions will apply to.
  - Then, select whether these parties should be able to open/download (view/save this object), view permissions, and/or edit permission.

### Object Versioning

Keeps track of and stores all old/new versions of an object so that you can access and use an older version if you need to.

- Versioning is either ON or OFF (default). Once versioning is on, you can 'suspend' versioning, but not fully turn it off. Suspending versioning only prevents versioning going forward, so all previous objects with version will still maintain their older versions. Versioning can only be set on the bucket level and applies to all objects in the bucket.
- To enable versioning, navigate to the bucket's properties (see section above) and select the 'Versioning' dropdown. By default, versioning is off, so you can press 'Enable Versioning' in

this dropdown section. Now, you cannot turn versioning off again, you can only suspend versioning.

- You can reupload the object without worrying about moving the older version. You will need to specify the storage class of that object again, or else the new version will be set to Standard by default. If you suspend versioning and upload then reupload a new object, the older versions will not be saved.
- To view the version of an object, in the bucket or folder where the object is contained, select the object by clicking in the horizontal space to the right of the object (without clicking on its name). In the top middle bar to the right of ‘Versions:’, select ‘Show’. Now you can see a list of the previous versions of that object.

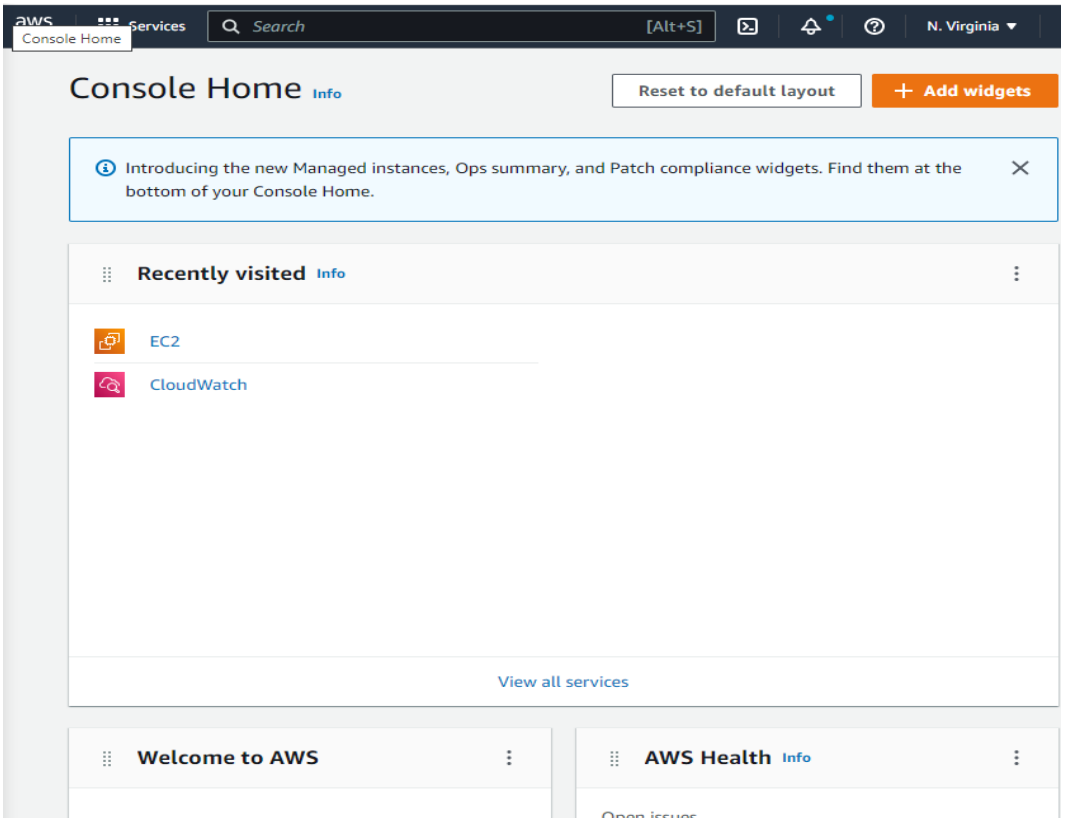
# EC2:

## What is EC2?

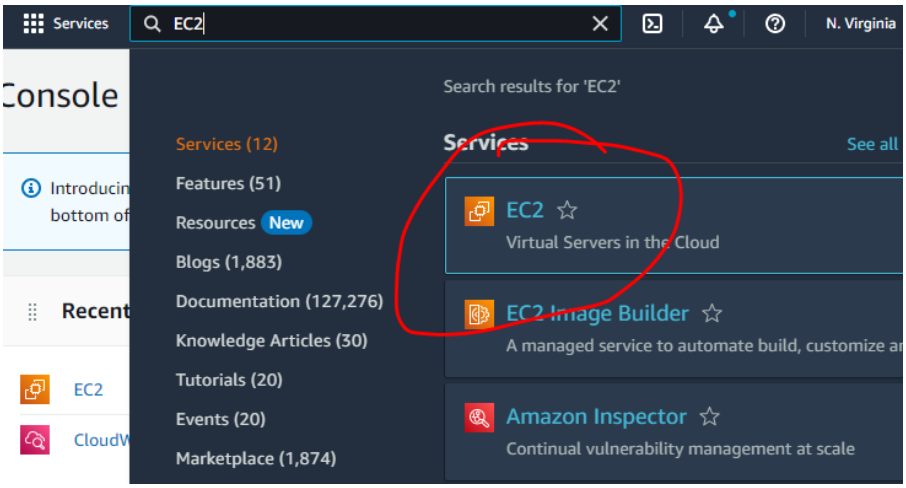
EC2 is a virtual computing environment, known as *instances.* It is essentially a virtual server that is run on Amazon's cloud resources. Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types.* Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software).

## Launching an Instance

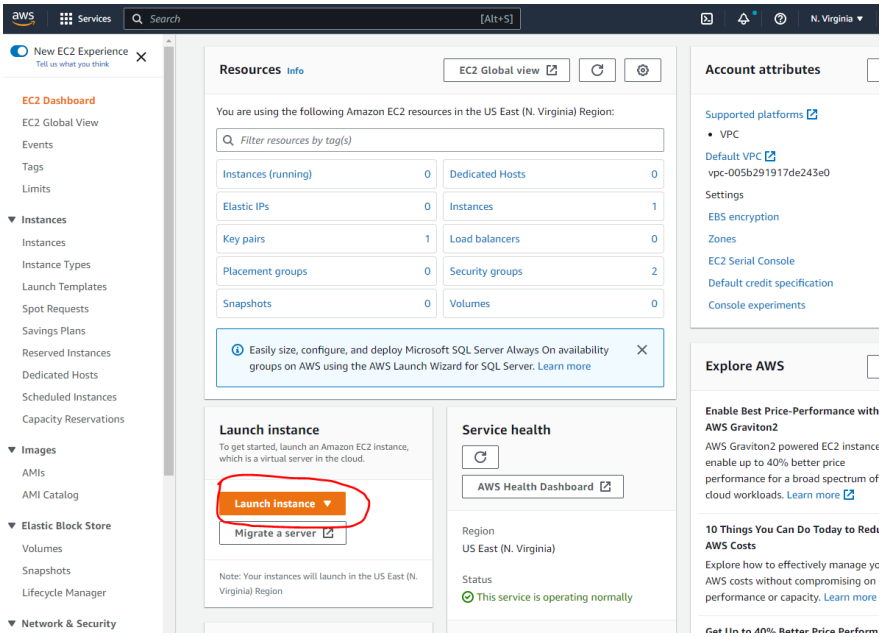
First go to the AWS console:



Then type “EC2” in the search bar at the top and navigate to that page.

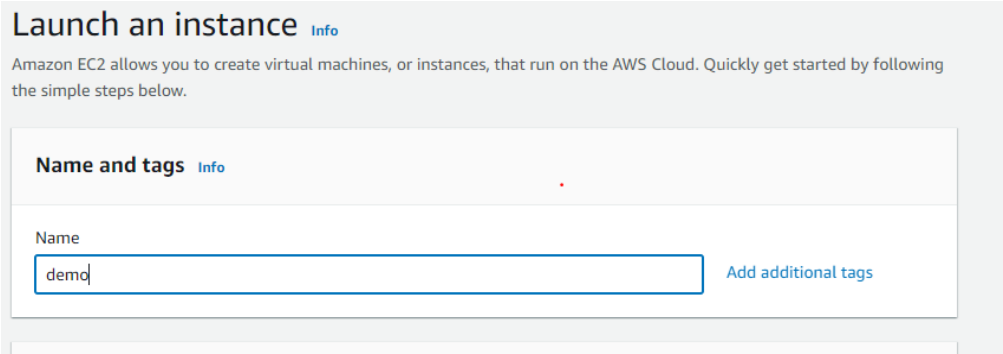



Then Click on the “Launch Instance” button.

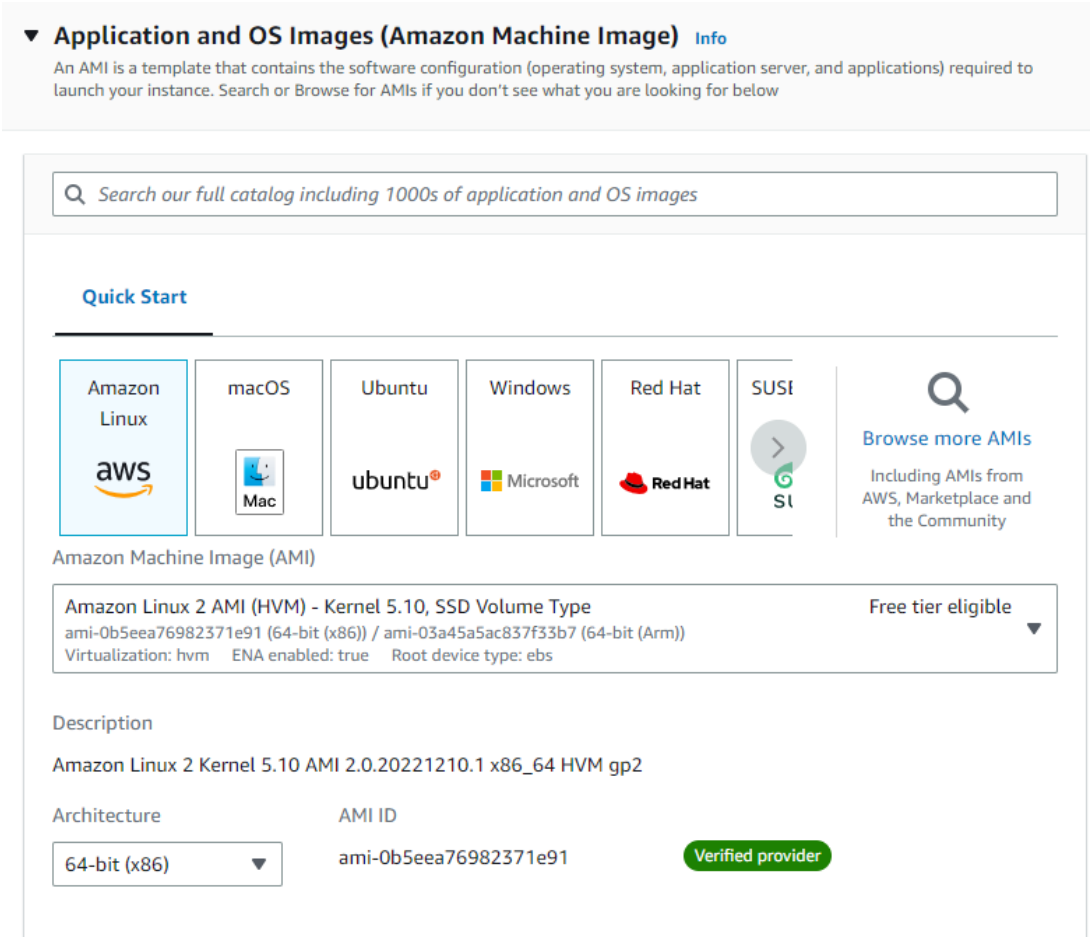


You will then be taken to a Form to fill out to launch an instance:

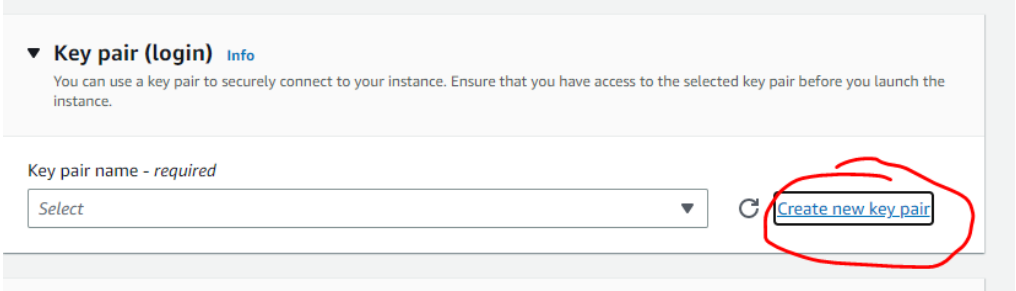
1. The first thing that will need to be filled out is the name of your instance.



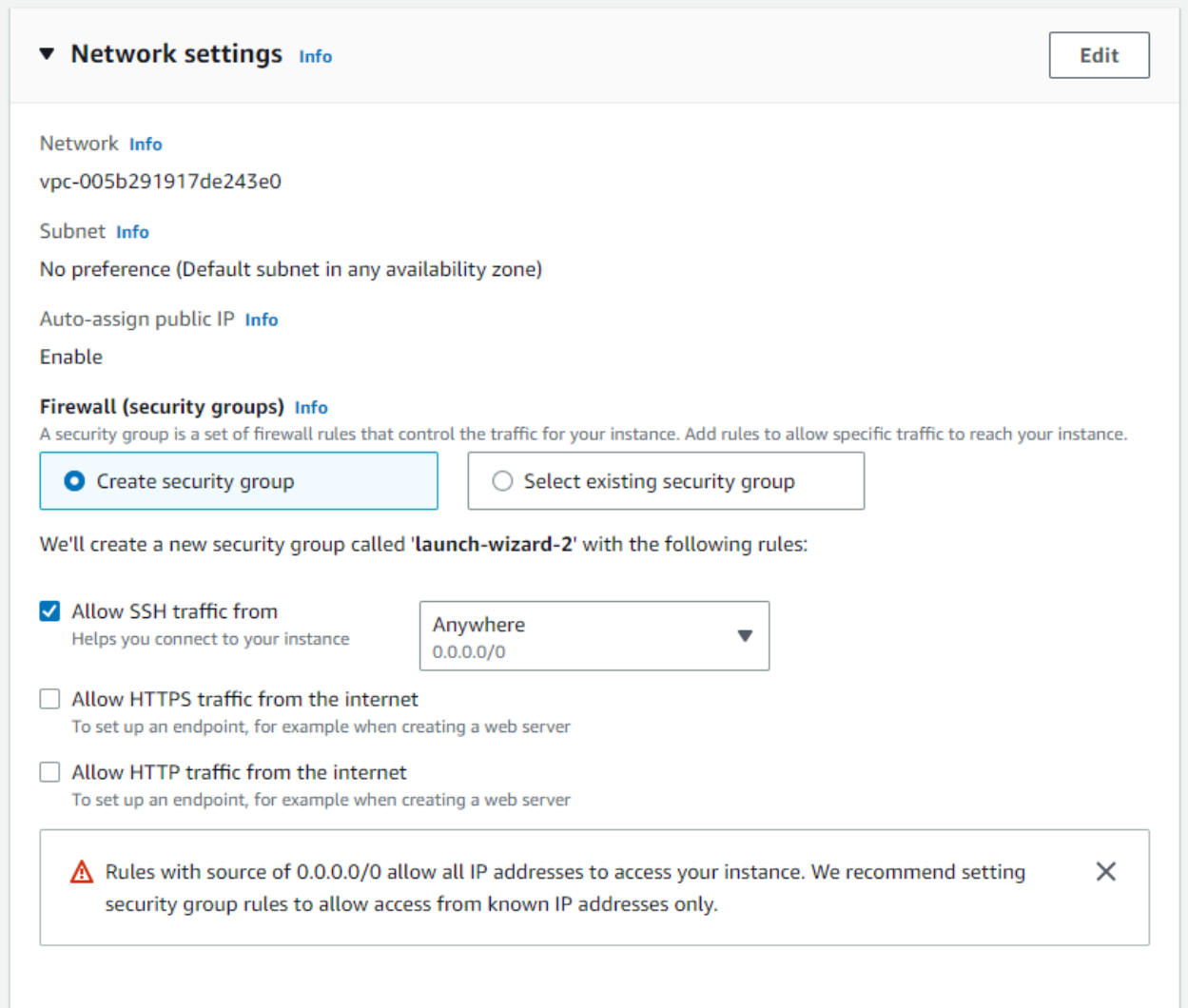

2. Then you will be asked to choose an Amazon Image machine. For this tutorial we will be using the Amazon Linux image because it is “free tier eligible”. There are other images that are Free Tier eligible, but please be sure to choose the free tiers in order to not be charged money.



3. For instance type, we can use the **t2.micro** because it is also on the free tier.
4. For the Key pair section, this will be needed to connect to your instance.
    a. Click on “Create a new key pair”.

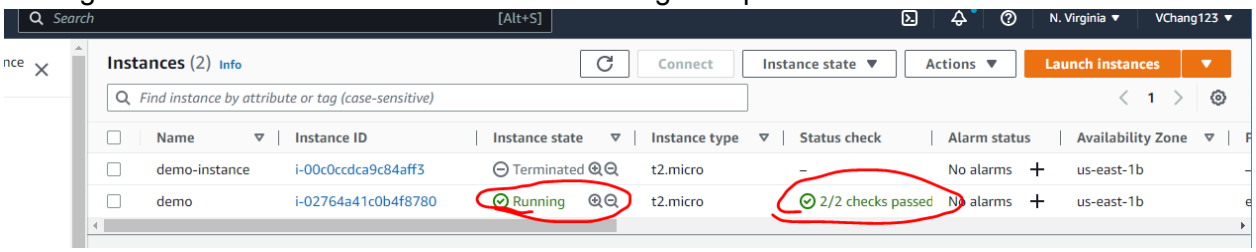



    b. You will see the following form.
    c. Create a name for your key pair, select “RSA”, and choose either the **.pem** or the **.ppk** format, depending on what software you have.



5. For Network settings:
    a. You can leave everything as it is, it will automatically create a new security group that will allow SSH traffic from any IP address.
6. You can adjust the storage as long as it remains within the free tier.
7. Then finally click “Launch Instance”.

It will take some time for the instance to get up and running. You can view all your instances by clicking on the **‘instances’** tab on the left side navigation panel.




When you see that the instance is running and all the checks have passed it is ready to be used.

## Connecting to your instance:

- Depending on which OS you have (Windows / MacOS / etc..)
    - There are different ways to connect your instance.
    - For windows please follow this documentation:
        - https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/EC2_GetStarted.html#ec2-connect-to-instance-windows
    - For other devices, please use this documentation:
        - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html

## Terminate/Stopping your instance:

Finally when you are done using the instance, make sure to delete it or stop it so it does not keep running and potentially cost money.

# Lambda:

## What is Lambda?

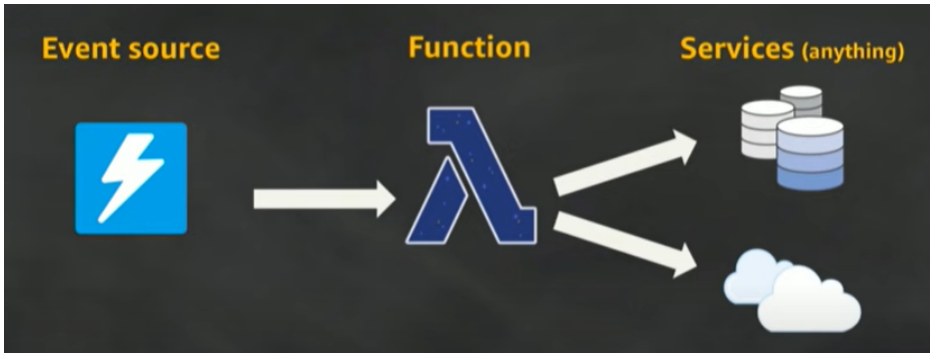
- AWS Lambda is a compute service that lets you run code without provisioning or managing servers.
- With Lambda, you can run code for virtually any type of application or backend service. All you need to do is supply your code in one of the languages that Lambda supports.
- You organize your code into Lambda functions. Lambda runs your function only when needed and scales automatically, from a few requests per day to thousands per second.

## Getting Started:

1. Please make sure you have an AWS account.

## How does Lambda Work?

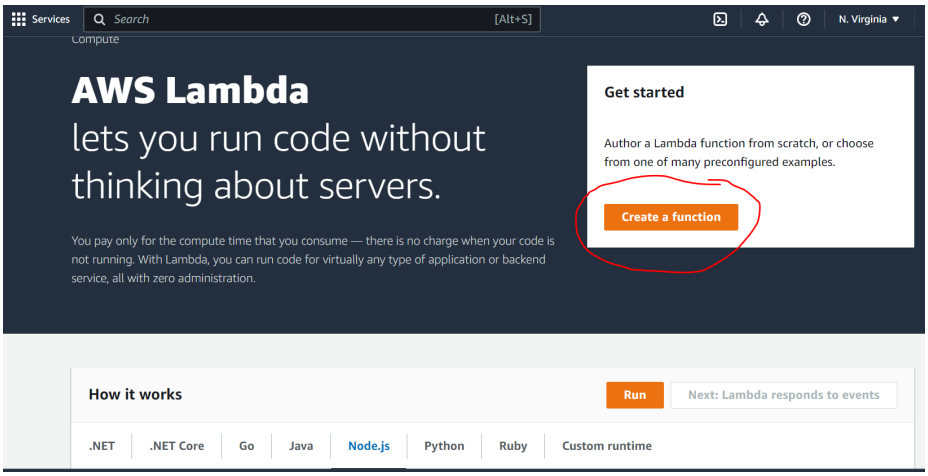
- Lambda functions work by event triggers.
- An event trigger could be a file uploaded to S3, API calls, and much more.
- Once the lambda function receives a trigger, it will then execute the code that is in the function and return its output in a designated area.




## How to use Lambda:

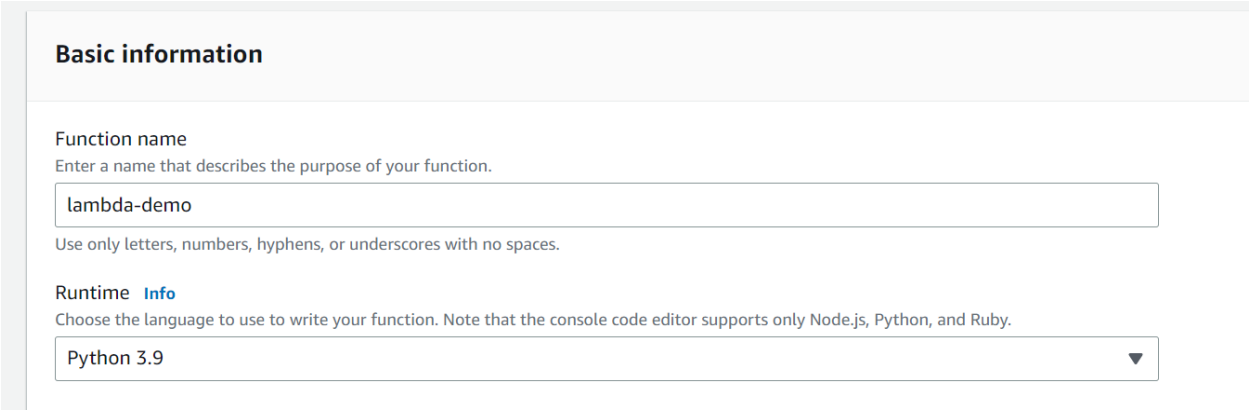
In this demo, we will be explaining how to trigger a lambda function by uploading a file to an S3 bucket.

First go to the lambda console by searching “lambda” in the search bar at the top. Then click on “Create a Function” on the home page.

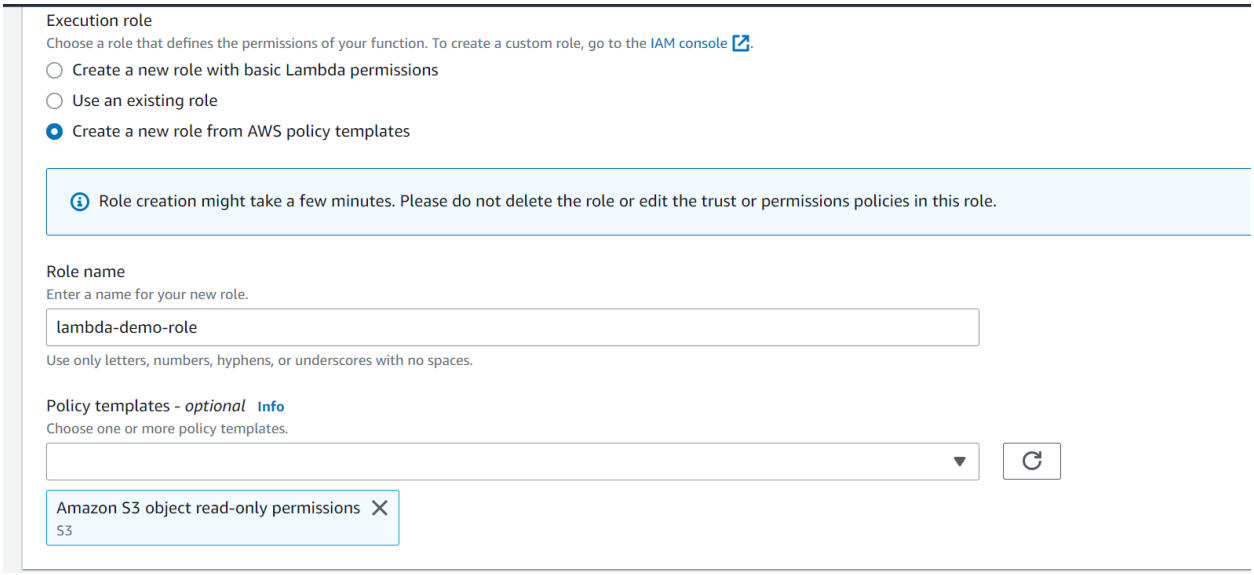



We are then going to create a custom lambda function from scratch.

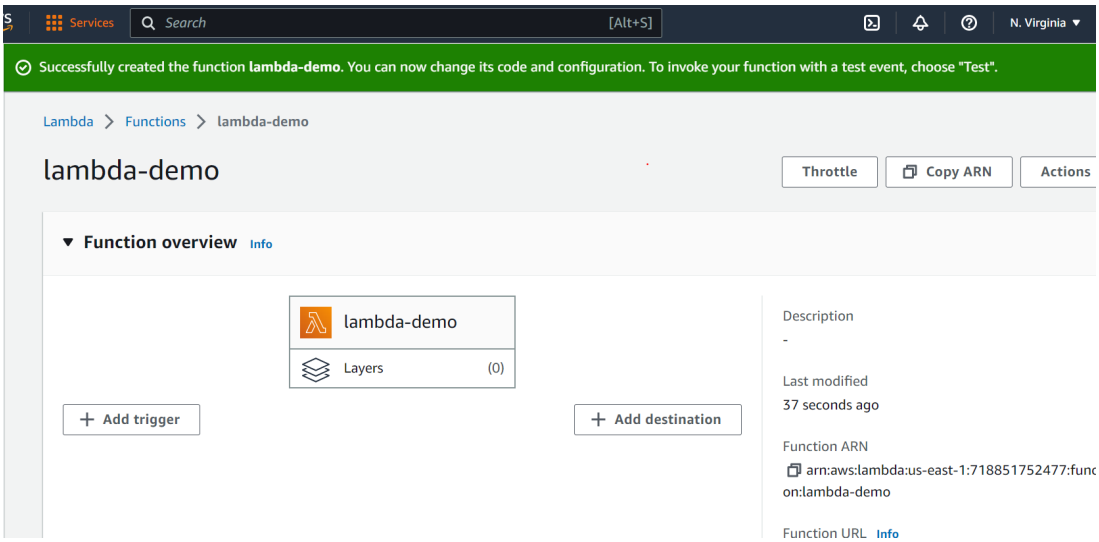
1. First give your function a name, for example, “lambda-demo”.
2. Then choose a language to write your code in. For this demo, we will be using python 3.9.



3. Then because we are going to need to access the S3 bucket, we are going to need to create a new permission group.
    a. Click on the “Change default execution role” dropdown.
    b. Select “Create a new role from AWS policy templates”.
    c. Then give the role a name.
    d. Select “Amazon S3 object read-only permissions”.
    e. And then press create function afterwards.

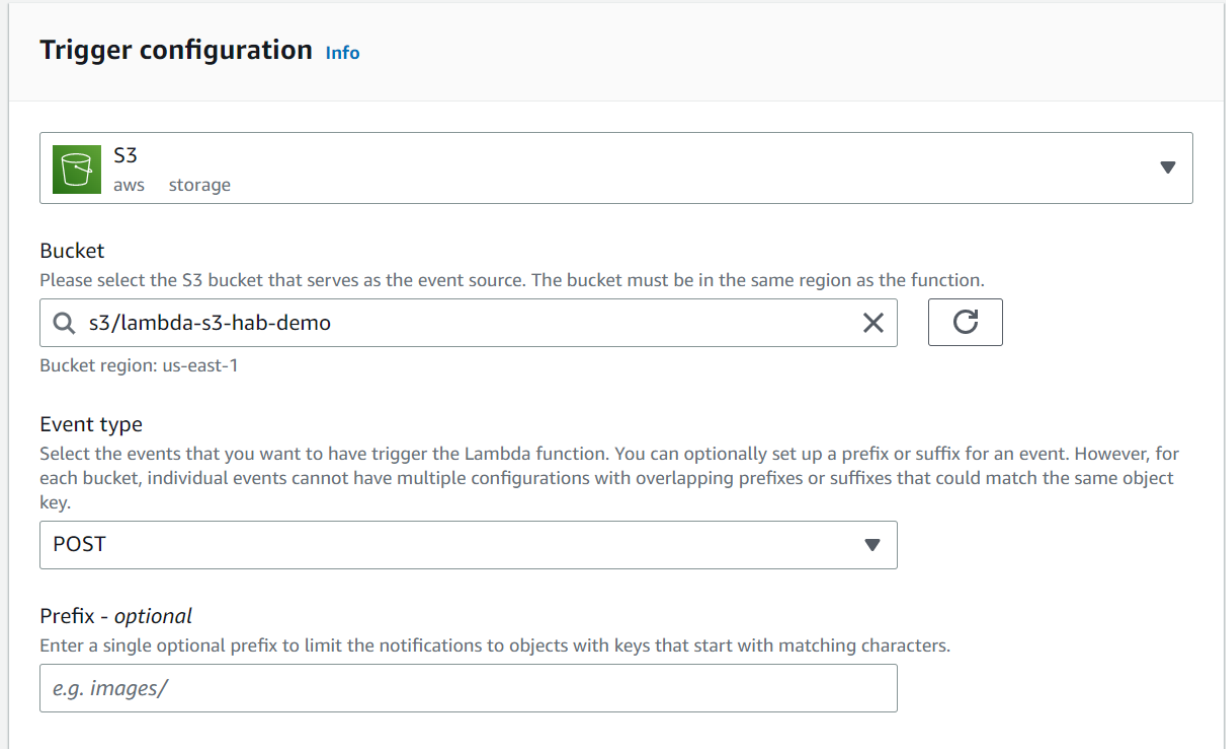


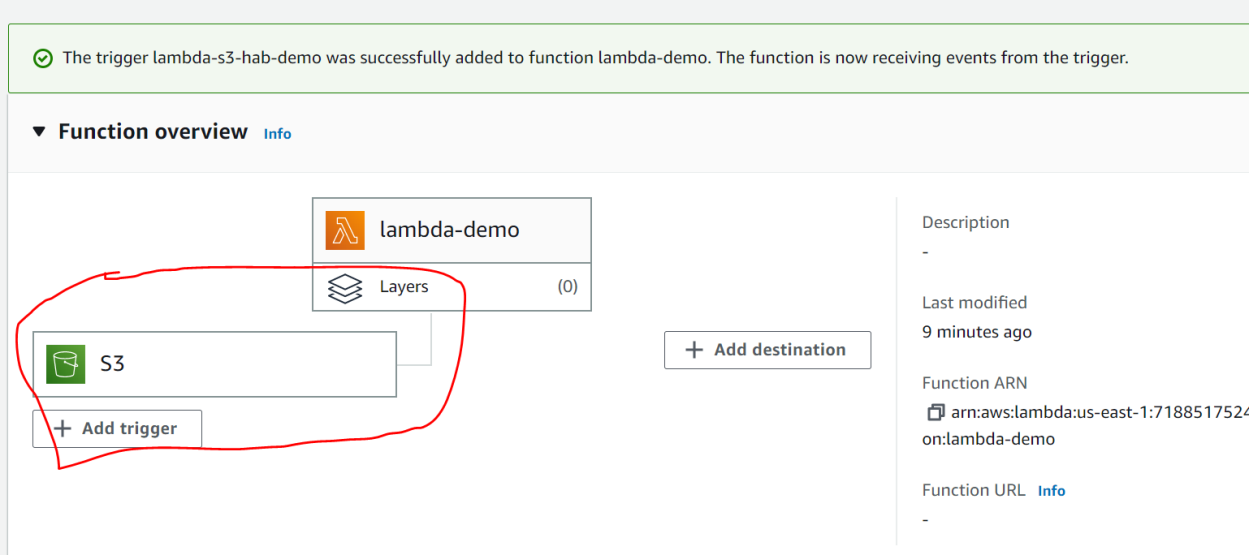

4. You will be brought to a page that looks like this:




5. To add the S3 trigger, press the “add trigger” button.

6. You will be brought to a page that looks like this:
    a. Choose “S3” as the source event.
    b. Select a bucket (if you do not have a bucket, please create one).
    c. Select the event type “POST”.
    d. You can also add prefixes if you have a subfolder within your bucket or suffixes if you want to specify file types that trigger the lambda function.
    e. Then press “Create".
    f. You should see that the lambda function now has a trigger!



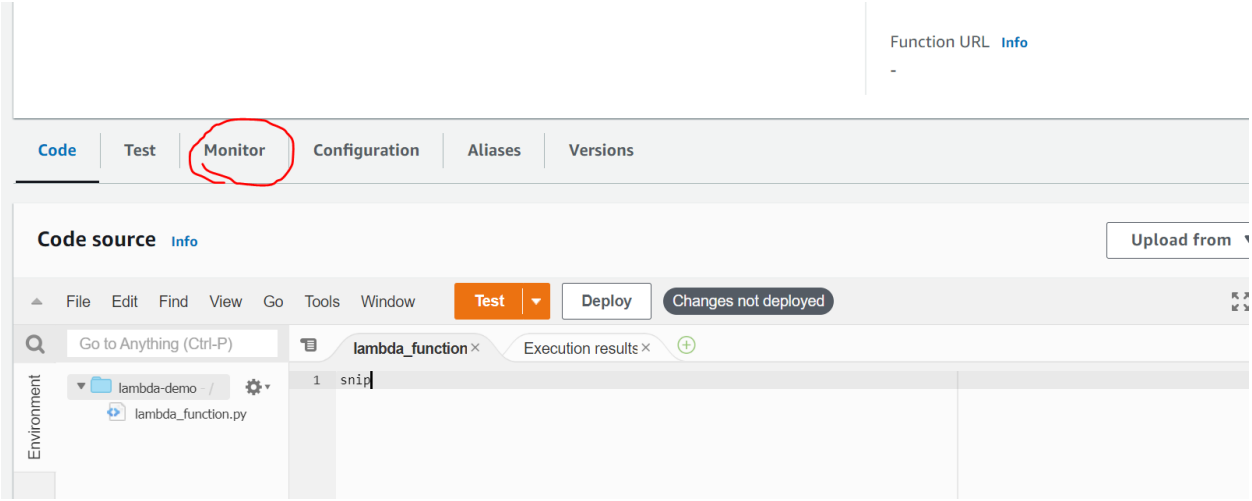



7. Before we go on, it is important to understand what an S3 event actually looks like. Below is the information that gets sent to lambda when a S3 event is processed. The fields that will be important to us are the bucket name and key. The bucket name will be the bucket that has the uploaded file and the key will actually be the name of the uploaded file.

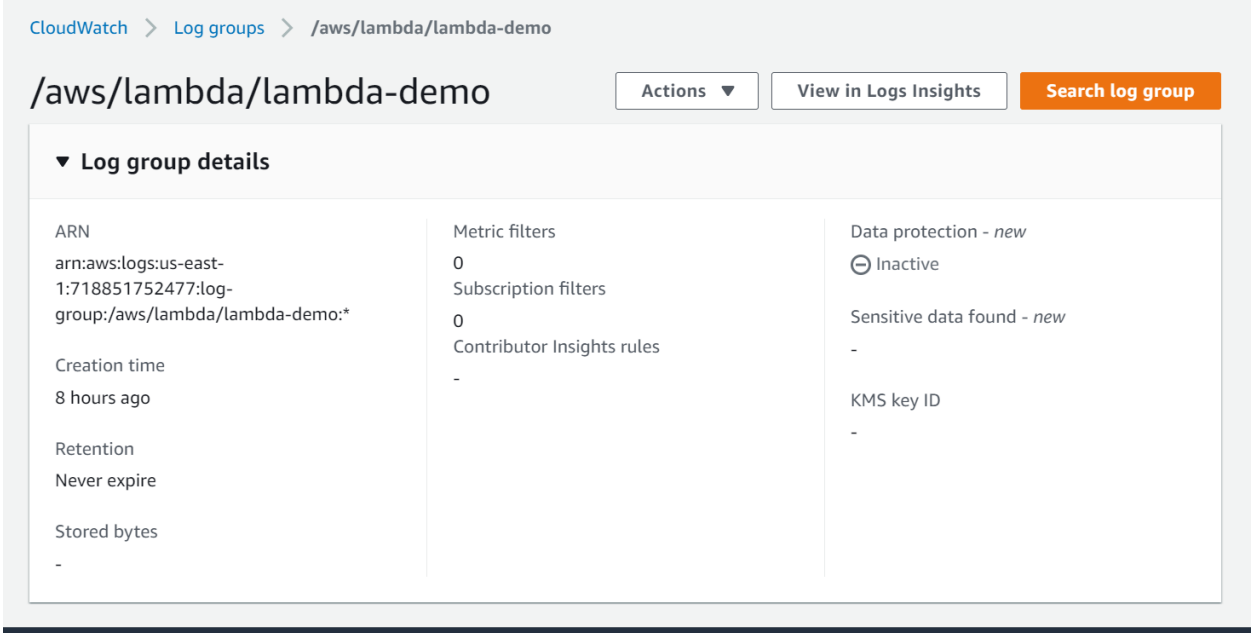
```
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2":
"EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "example-bucket",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::example-bucket"
        },
        "object": {
          "key": "test%2Fkey",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
      }
    }
  ]
```

8. For the code section, here is the Python code for processing the csv file that will be uploaded.

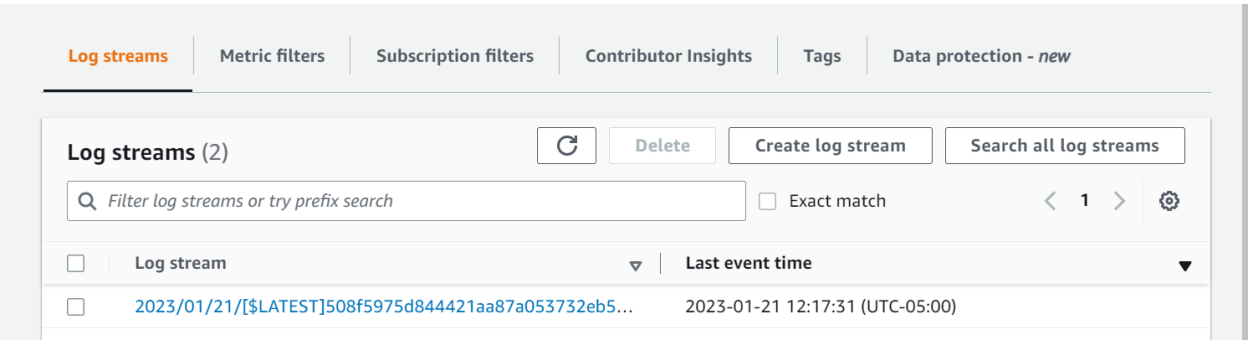
9. Next, here is a link to the csv file to download.
    a. After downloading the link, go to the S3 using the search bar and upload the file into the bucket that you selected for the trigger.
10. Then after you uploaded the file, go back to Lambda, and at the bottom of the page you should see a monitor tab.



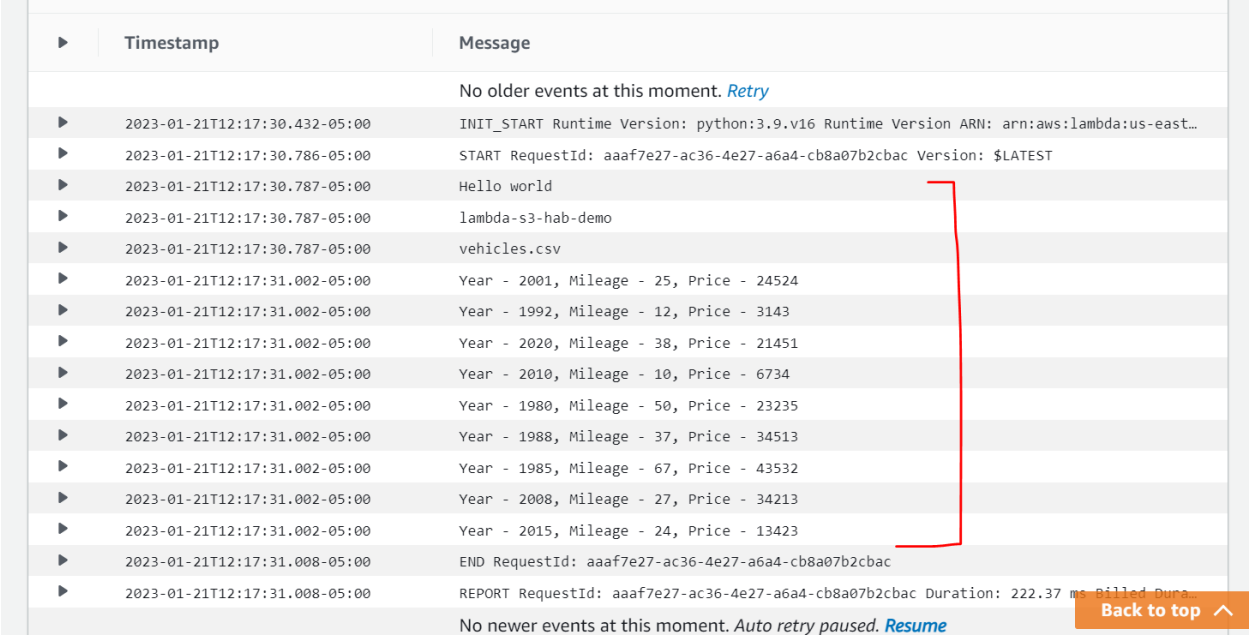

11. Click on this tab and then click on the button that says “View Cloudwatch Logs”.
12. It will take you to a page that looks like this:



13. At the very bottom of the page you will see a log streams tab, click on the most recent log stream which should have been triggered by the S3 event.




14. You should see something like this in the logs, where the output of the data processing is printed in the logs, meaning that the lambda function was triggered and our data has been processed.




**Please reach out to the Hack@Brown team if you have any questions on the content in this starter kit!**