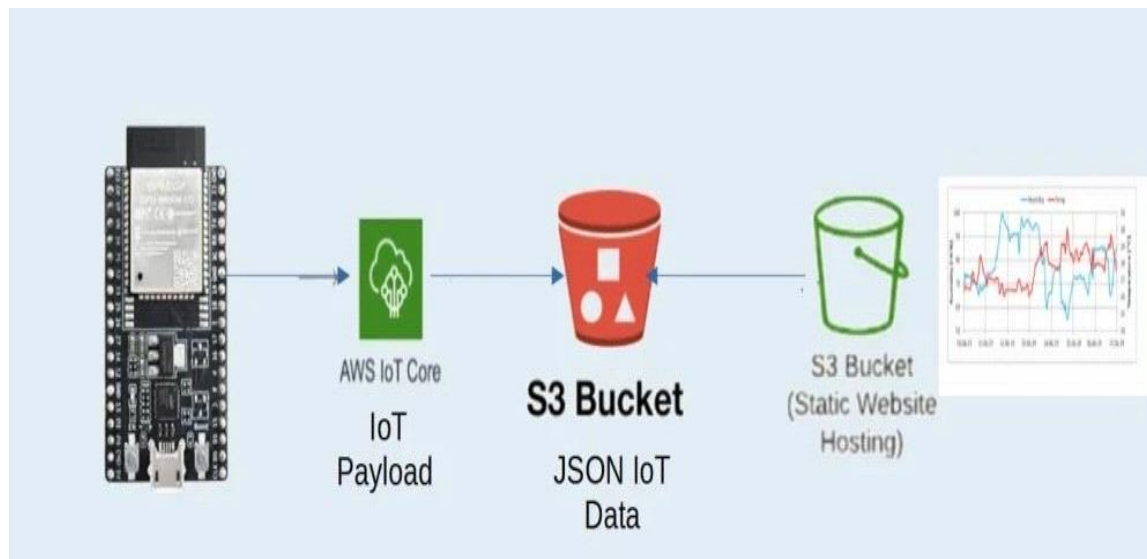


SERVERLESS IOT DATA PROCESSING DESING

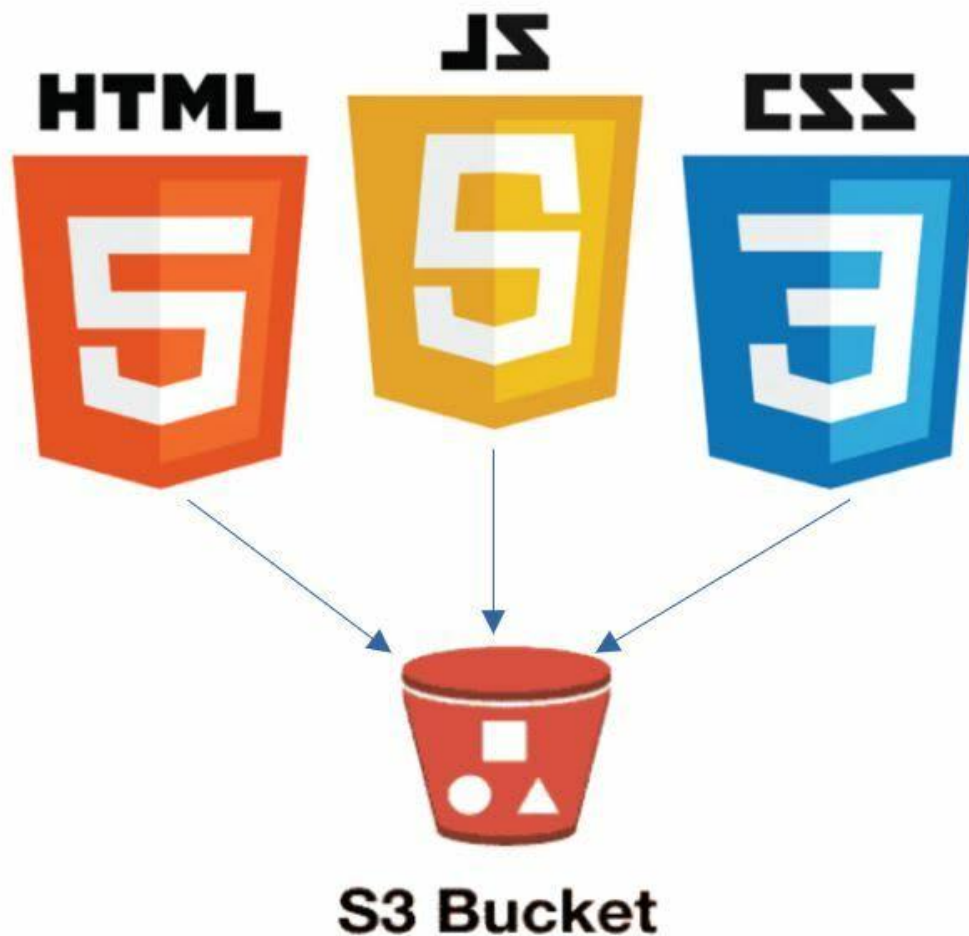


When working with IoT devices, which transmit data to AWS, serverless IoT workflows can save the customer a tremendous amount of money. Instead of setting up an “always on” EC2 instance the client can engage individual AWS services only as needed. This multi-part IoT series will cover a variety of methods, with increasing levels of sophistication and functionality, to visualize IoT data on a static web host using various IoT centric services on AWS. The overall cost of using these AWS serverless services, even assuming you are off the free tier, will be

pennies for normal use.

Creating a public bucket in S3

Serverless IoT data



Making a Public S3 Bucket

The process of creating a public S3 bucket for website hosting

Go to AWS S3 and then select “Create bucket”

FOLLOW STEPS

A) Give your bucket a globally unique name, here I call mine a catchy name: mybucket034975

B) Keep your S3 bucket in the same region as the rest of your AWS services for this lab.

C) Switch “Object Ownership” to “ACL’s enabled”, this is new for late 2021! We now must first enable our Access Control Lists to make them public.



Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

mybucket034975

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and granted using access control lists (ACLs). Object ownership determines who can specify access to objects.

☐ ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☒ ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

☒ Bucket owner preferred

If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☐ Object writer

The object writer remains the object owner.

*** Unblock your S3 bucket and acknowledge you really want to do this.**

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

***Finally, select the “Create bucket” button at the bottom of the screen. That's all you have to do for this page, but don't worry, we are going to have more opportunity to make sure we really, really, and truly want to create a public bucket soon.**

Create bucket

Now go back into your newly created bucket and click on the “Permissions” tab.

Go to Bucket Policy and choose “Edit.” We will paste and save a basic read-only policy.

```
[
  {
    "AllowedHeaders": [
      "Authorization"
    ],
    "AllowedMethods": [
      "GET"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "ExposeHeaders": [],
    "MaxAgeSeconds": 6000
  }
]
```

You must paste the name of your bucket into the policy then follow it by ‘/*’ to allow access to all Get/Read partitions within the bucket. Also it's a good idea to change the “Sid” to something unique within your account.

***Now we get a chance to visit that ACL we enabled earlier in this process. Click “Edit” then**

make the changes as shown below:

The screenshot shows the Amazon S3 console interface for configuring bucket permissions. On the left, the 'Buckets' menu is expanded. The main panel displays the 'Permissions' tab for a specific bucket. The 'Bucket owner (your AWS account)' section shows the canonical ID and the 'List' and 'Write' permissions are checked. The 'Everyone (public access)' section shows the 'List' and 'Read' permissions are checked, with blue checkmarks and arrows indicating the changes. The 'Authenticated users group' and 'S3 log delivery group' sections show no permissions are checked. A warning message states: 'When you grant access to the Everyone or Authenticated users group grantees, anyone in the world can access the objects in this bucket.' Below this, there is a checkbox 'I understand the effects of these changes on my objects and buckets.' which is checked. At the bottom, there is a 'Save changes' button circled in blue.

We are giving “Everyone,” or at least those know or can discover our unique bucket URL, permission to read our bucket info. Click on the 'List' and 'Read' buttons where shown and then acknowledge again that you are extra special certain that you want to do this . Then click “Save changes.”

*** Wow, we are at our last step in creating a**

public bucket. Now we should set the CORS policy so we don't get any pesky "mixed use" access-control non-allowed origin issues for cross domain access – I hate those CORS rules used to be in XML only format and then AWS decided to keep everything consistent and switch the CORS format to JSON. Even though this change caused some legacy conflict issues with existing XML CORS rules it was the right choice as JSON is clearly better than XML despite what the SOAP fans on social media will tell you . Below is a generic CORS JSON document you can use in your own S3 bucket:


```
[
  {
    "AllowedHeaders": [
      "Authorization"
    ],
    "AllowedMethods": [
      "GET"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "ExposeHeaders": [],
    "MaxAgeSeconds": 6000
  }
]
```

COMPLETE..