Create a CloudFront Distribution

Amazon CloudFront is a content delivery network (CDN), which is a system of distributed servers (network) that deliver webpages and other web content to a user based on the geographic location of the user, the origin of the webpage, and a content delivery server.

Diagram 1) This diagram does **Not** have a CDN

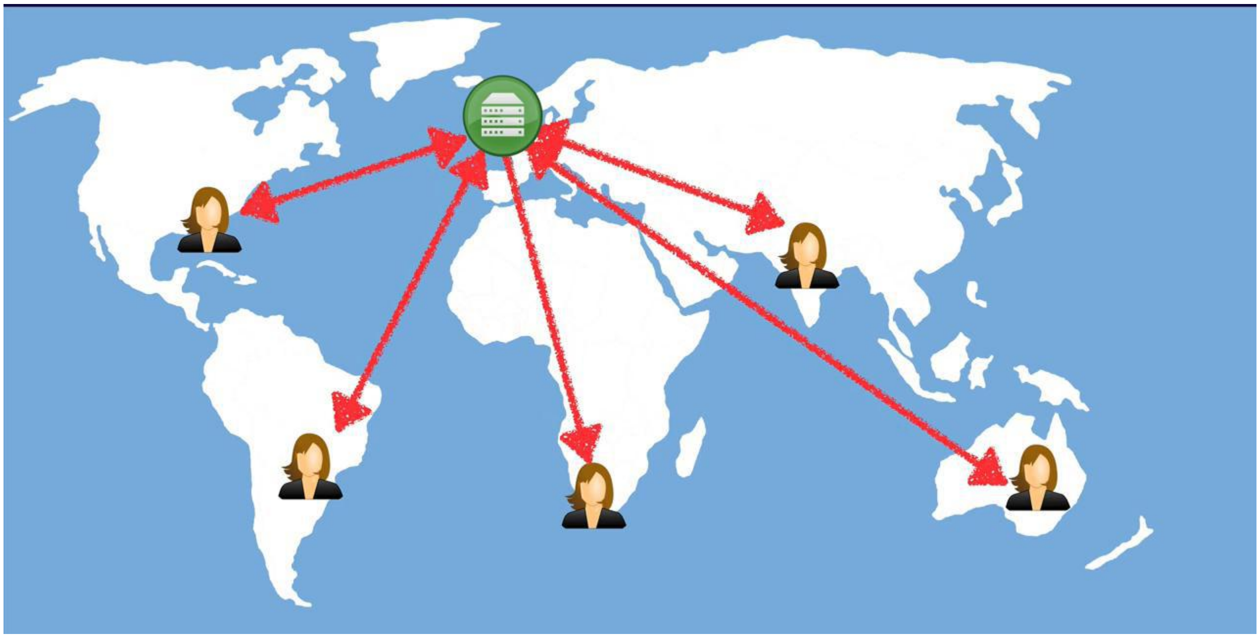
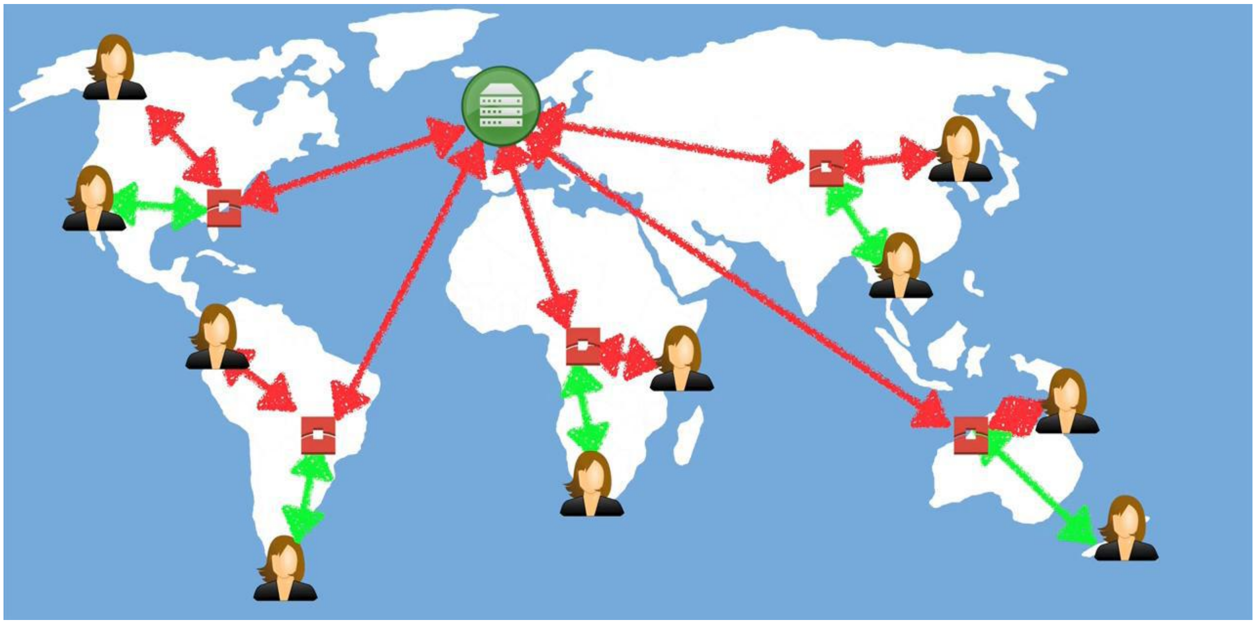


Diagram 2) This diagram has a CDN



In Diagram 1, there is **not** a CDN enabled. So if we are hosting our website in London, our users all around the world will need to pull the website data all the way from London regardless of the user location. This can take some time if the user is very far away.

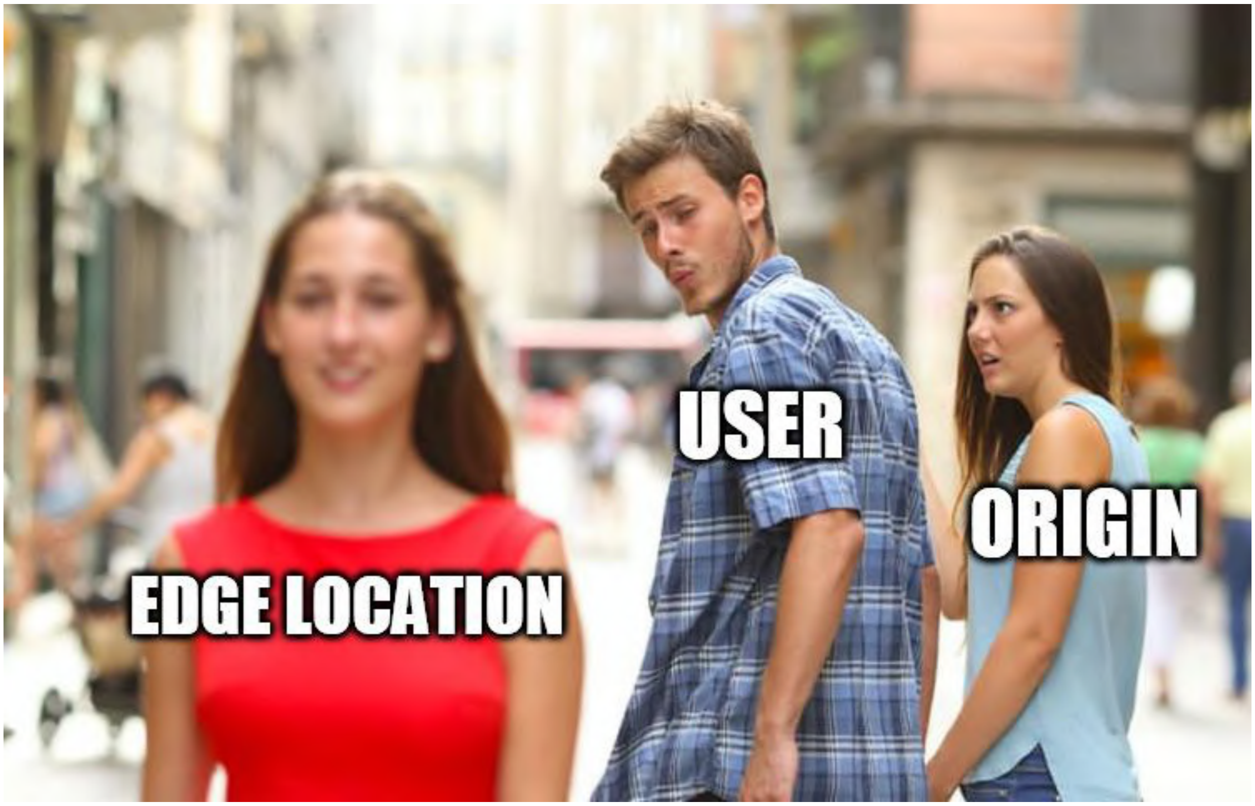
In Diagram 2, there is a CDN enabled. If we are hosting our website in London, our users will first query an edge location nearby. If the edge location has a copy of the website data then the edge location will send the requested data to our user (without having to pull the data from London). If the edge location does not yet have a copy of the website data, then the edge location will download the data from London and return it to the user, and will also cache the data (for hypothetically between 48 hours and 72 hrs). Therefore, when another user queries the edge location for our website data, they will get it a lot quicker because they will not have to pull it all the way from London.

Key Terms:

Edge Location – This is the location where content will be cached. This is separate to an AWS Region/AZ.

0rigin – This is the origin of all the files that the CDN will distribute. This can be either an S3 bucket, an EC2 instance, an elastic load balancer or Route 53.

Amazon CloudFront can be used to deliver your entire website, including dynamic, static, streaming, and interactive content using a global network of edge locations. Requests for your content are automatically routed to the nearest edge location, so content is delivered with the best possible performance.



Section 1) Let’s create an S3 bucket. This S3 bucket will be our origin. So our CloudFront Distribution will distribute the files from our origin (the S3 bucket we create).

Go to the AWS Console and click **Services** and then click **S3**

Click **Create bucket**

Bucket name: ourfirstcloudfronttesbucket ← name your bucket something like this

Region: US West (N. California) ←choose a region for your bucket

Click **Create**

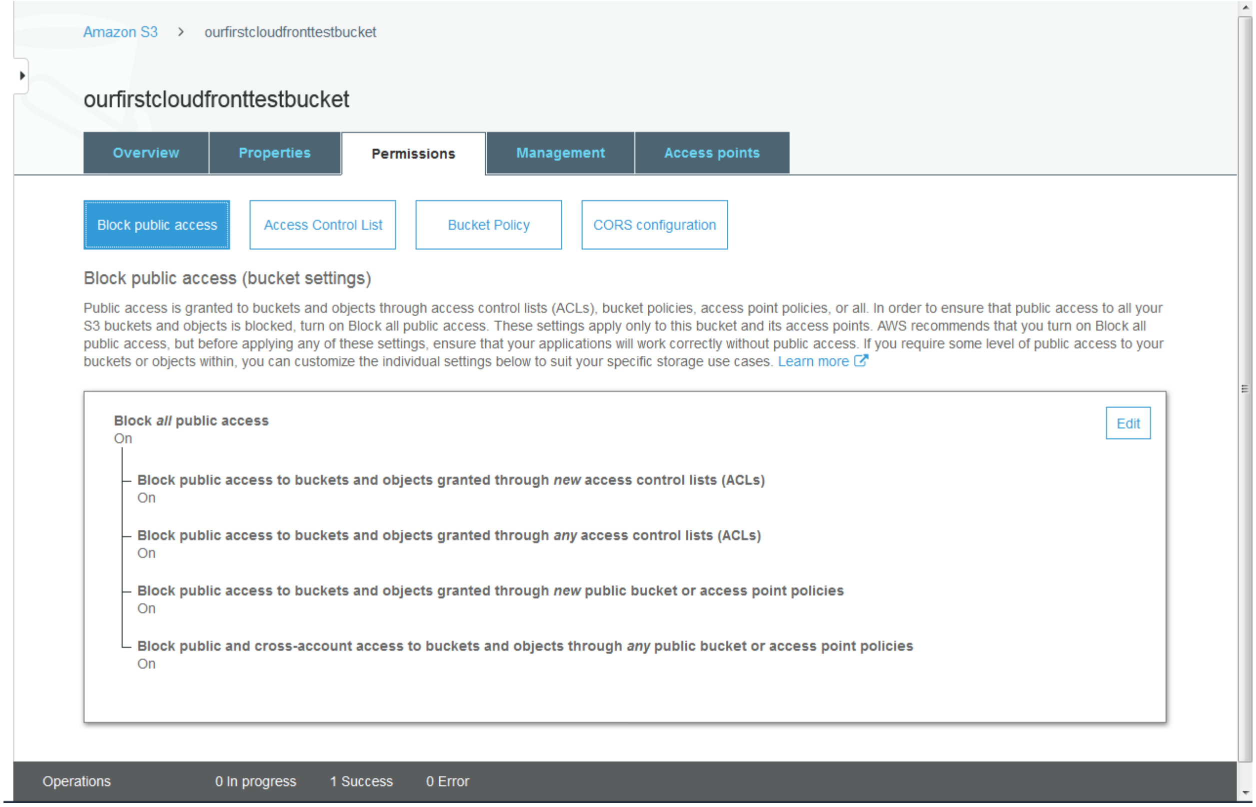
Once the bucket is created, go ahead and click on it.

Click the **Permissions** Tab

Click the **Block Public Access** Tab

And click **Edit**

(See the screenshot below for clarification)



ourfirstcloudfronttestbucket

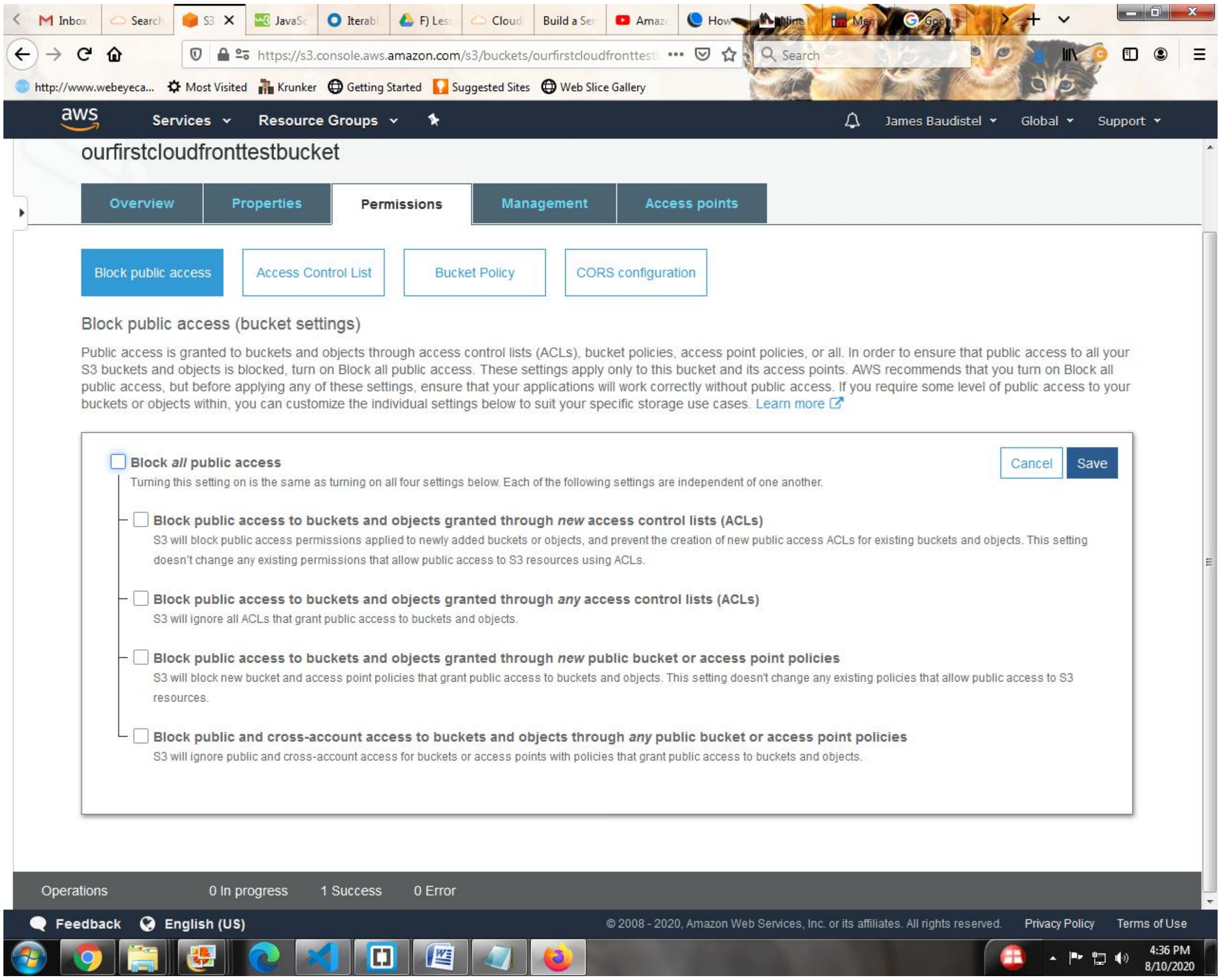
Block public access (bucket settings)

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 all your S3 buckets and 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 your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

Block *all* public access
On

- Block public access to buckets and objects granted through *new* access control lists (ACLs)
  On
- Block public access to buckets and objects granted through *any* access control lists (ACLs)
  On
- Block public access to buckets and objects granted through *new* public bucket or access point policies
  On
- Block public and cross-account access to buckets and objects through *any* public bucket or access point policies
  On

Then **uncheck** where it says Block all public access and then click **Save** (because we want our bucket to be publicly accessible) (See the screenshot below for clarification)



ourfirstcloudfronttestbucket

Block public access (bucket settings)

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 all your S3 buckets and 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 your buckets 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.

Operations 0 In progress 1 Success 0 Error

Then type confirm and click **Confirm**

Our Public access settings have now been updated successfully.

We want any object that we upload to the bucket to be made public automatically without manually having to make each object public. Therefore, let’s create a bucket policy to accomplish this.

Click the **Bucket Policy** Tab

Now paste in the code below:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "PUT ARN HERE/*"
    }
  ]
}
```

★★★On Line 9 of the code make sure to replace **PUT ARN HERE** with the ARN of the bucket that you created. The ARN of the bucket is located under the **Block Public Access** tab next to where it says **Bucket Policy Editor.**

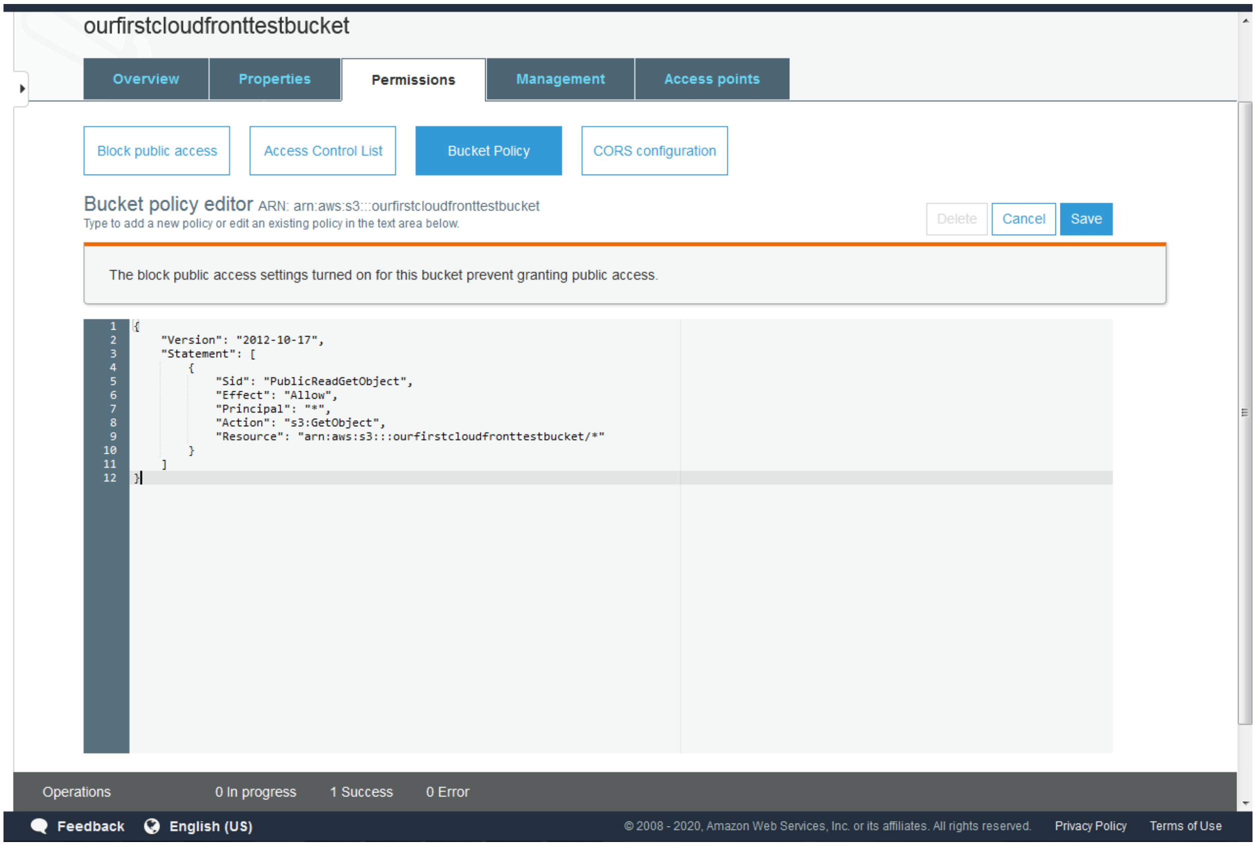
Line 9 will look something like this:

"Resource": "arn:aws:s3:::ourfirstcloudfronttestbucket/*"

★Your ARN will be different because your bucket name will be different

★Note that the ARN is followed by a forward slash and an asterisk and is surrounded by double quotes.

(See the screenshot below for clarification)



ourfirstcloudfronttestbucket

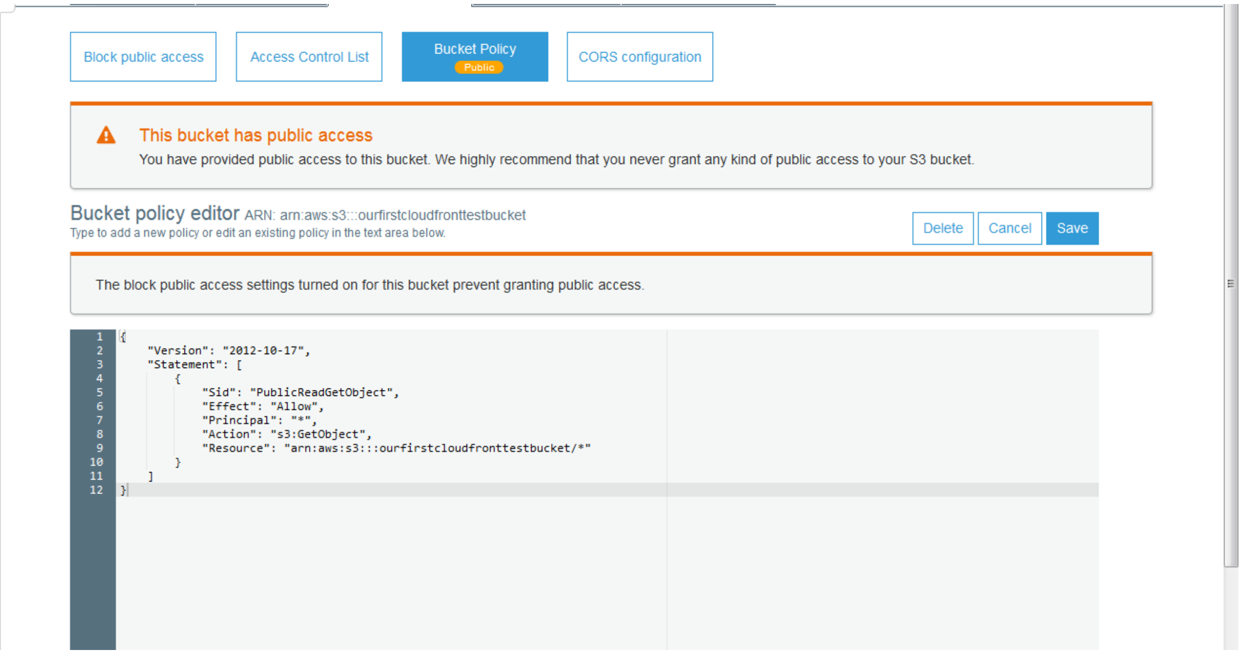
Bucket policy editor ARN: arn:aws:s3:::ourfirstcloudfronttestbucket
Type to add a new policy or edit an existing policy in the text area below.

The block public access settings turned on for this bucket prevent granting public access.

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

Click **Save**

The bucket policy tab will now be marked as Public (See the screenshot below for clarification)



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.

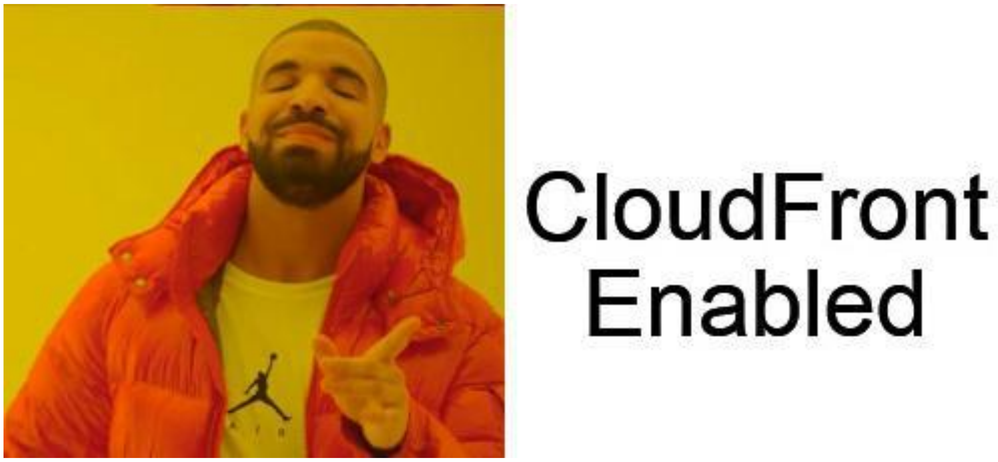
Bucket policy editor ARN: arn:aws:s3:::ourfirstcloudfronttestbucket
Type to add a new policy or edit an existing policy in the text area below.

The block public access settings turned on for this bucket prevent granting public access.

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

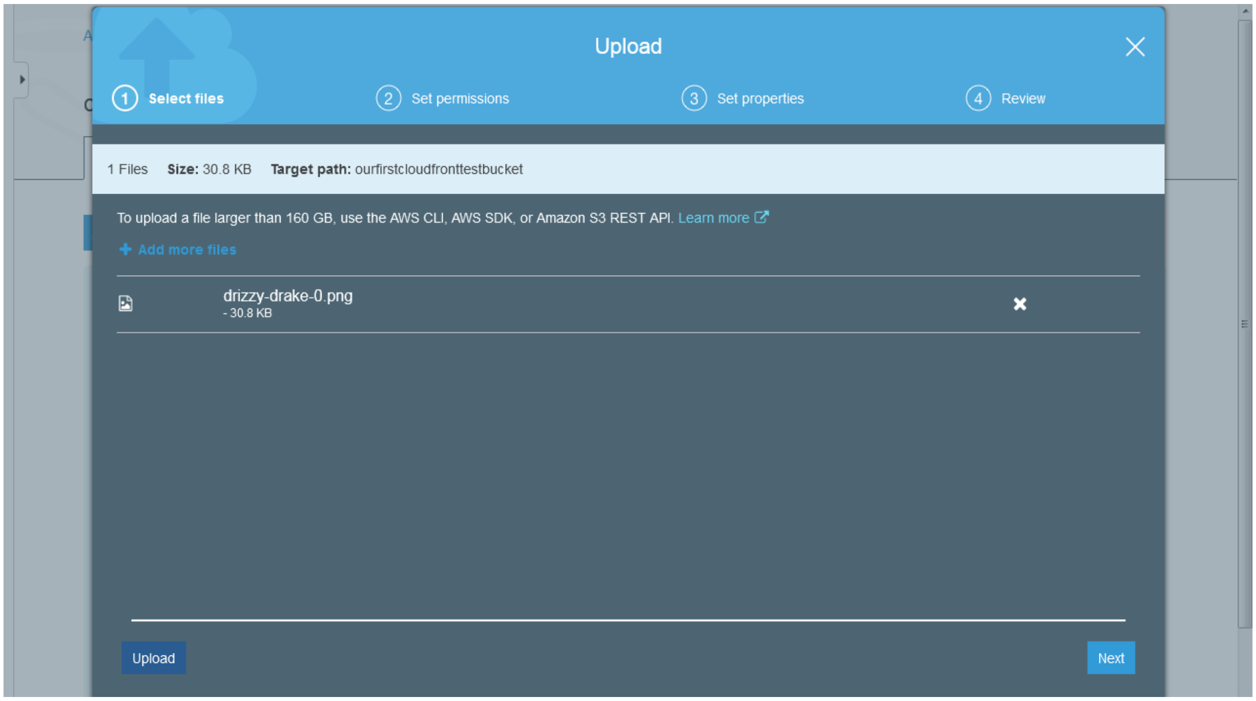
Now click the **Overview** tab

Screenshot and Save the image/meme below on your computer



Once you have saved the image click the **Upload** button

Choose the file and then click **Upload** located towards the bottom left of the screen (See the screenshot below)

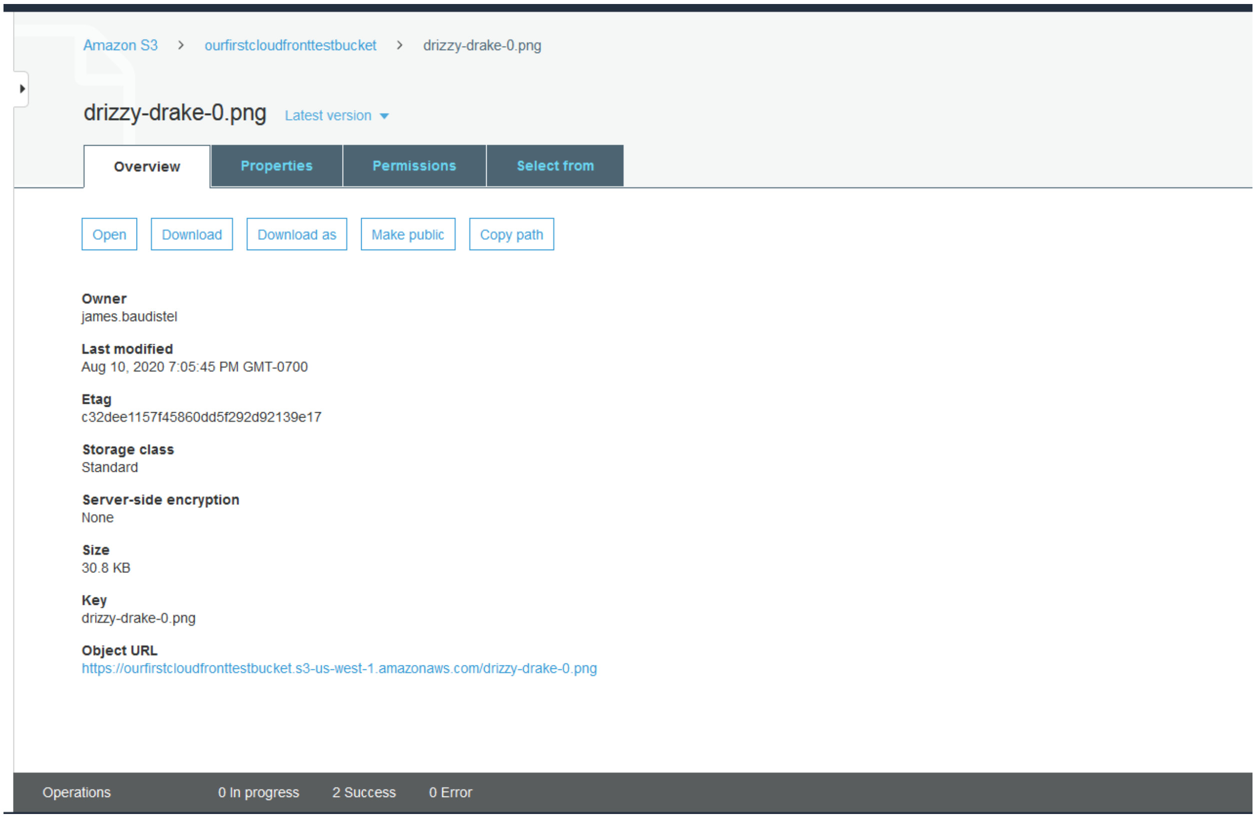


1 Files **Size:** 30.8 KB **Target path:** ourfirstcloudfronttestbucket

To upload a file larger than 160 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. Learn more

drizzy-drake-0.png
- 30.8 KB

Once the file has been uploaded to your bucket. Click on the file. Then click on the **Object URL** link. (See the screenshot below for clarification)



Owner
james.baudistel

Last modified
Aug 10, 2020 7:05:45 PM GMT-0700

Etag
c32dee1157f45860dd5f292d92139e17

Storage class
Standard

Server-side encryption
None

Size
30.8 KB

Key
drizzy-drake-0.png

Object URL
https://ourfirstcloudfronttestbucket.s3-us-west-1.amazonaws.com/drizzy-drake-0.png

When you click on the Object URL link, it is pulling the data from the bucket (which is our origin). However, we want to enable our CloudFront Distribution so that the data gets cached at an edge location for quicker access.

Section 2) Let's enable CloudFront

Click **Services** then click **CloudFront** which is located under Networking

Click **Create Distribution**

★Click the **Get Started** button that is associated with creating a Web Distribution. (We do **NOT** want a RTMP distribution which is used for media files using adobe flash)

Origin Domain Name: ✓ourfirstcloudfronttestbucket.s3.amazonaws.com ← select the bucket you created from the list of all of your buckets

Origin ID: S3-ourfirstcloudfronttestbucket ← this Origin ID will be automatically generated once you select the Origin Domain Name.

(There is an option to restrict access using signed urls or signed cookies but for this lesson we are not going to do that)

(You can also put web application firewalls in front of your CloudFront Distribution, however we will not do that either in this lesson)

Simply scroll down and click **Create Distribution**

(It can take 15-20 minutes to deploy the distribution so be patient)

Once the status says deployed and the State says Enabled

You will see the domain name for your CloudFront distribution

It will look something like the following:

d3kmhmbwzgjqtb.cloudfront.net ← it will look similar to this

Copy the CloudFront Domain name into your clipboard and then paste it into a new URL tab. Then add a forward slash and then add the name of your image file to the end of it as the path. It will looking similar to the following:

d3kmhmbwzgjqtb.cloudfront.net/drizzy-drake-0.png ← make sure to add the name of the image that you put in your bucket as the path at the end of your CloudFront URL

Once you have entered the CloudFront domain with the image file name as the path, press **enter.**

Mission Complete!!!

Now in the AWS console click CloudFront and then click on the CloudFront Distribution that you created.

Then click the **invalidations** tab

(Here you can invalidate individual objects or directories or subdirectories by clicking **create invalidation**. If you delete the examples and just put /* in the box and clicked **invalidate …** this would invalidate everything so that it would no longer be on the edge locations) ← you do not need to do this step. This is just to show you where you can create an invalidation.

Clean up: Let's now delete our distribution.

★★★You need to disable the distribution before you delete it.

Click **Distributions** in the left margin.

Check the box next to the distribution that we created and click the **Disable** button. (Disabling the distribution can take about 15 minutes)

Once your distribution is disabled, check the box next to your distribution once again and click **Delete** and then click **Yes, delete.**

CloudFront Cheat Sheet

-CloudFront is a CDN (Content Distribution Network) It makes website load fast by serving cached content that is nearby

-CloudFront distributes cached copy at Edge Locations

-Edge Locations aren't just read-only, you can write to them as well (for example u can PUT object)

-TTL (Time to live) defines how long until the cache expires (refreshes cache)

-When you invalidate your cache, you are forcing it to immediately expire (refreshes cached data)

-Refreshing the cache costs money **because of transfer costs** to update Edge Locations

-**Origin** is the address of where the original copies of your files reside (example S3, EC2, ELB, Route53)

-Distribution defines a collection of Edge Locations and behavior on how it should handle your cached content

-Distributions has 2 Types: **Web Distribution** (static website content) **RTMP** (streaming media)

-Origin Identity Access (OAI) is used to access private S3 buckets

-Access to cached content can be protected via **Signed Urls** or **Signed Cookies**

-Lambda@Edge allows you to pass each request through a Lambda to change the behavior of the response

Question

A multinational corporation needs an AWS service that delivers its web content on a global level. Which one of the following AWS services will accomplish that?

A) AWS CloudTrail
B) Amazon CloudWatch
C) AWS Config
D) Amazon CloudFront

Explanation:
CloudFront is the AWS service used for delivering web content to users. The other three choices are for AWS management and governance, rather than content delivery like CloudFront. CloudWatch monitors AWS resources and applications, Config records configuration changes, and CloudTrail increases visibility into user and resource activity.
Resources

Answer: D) Amazon CloudFront