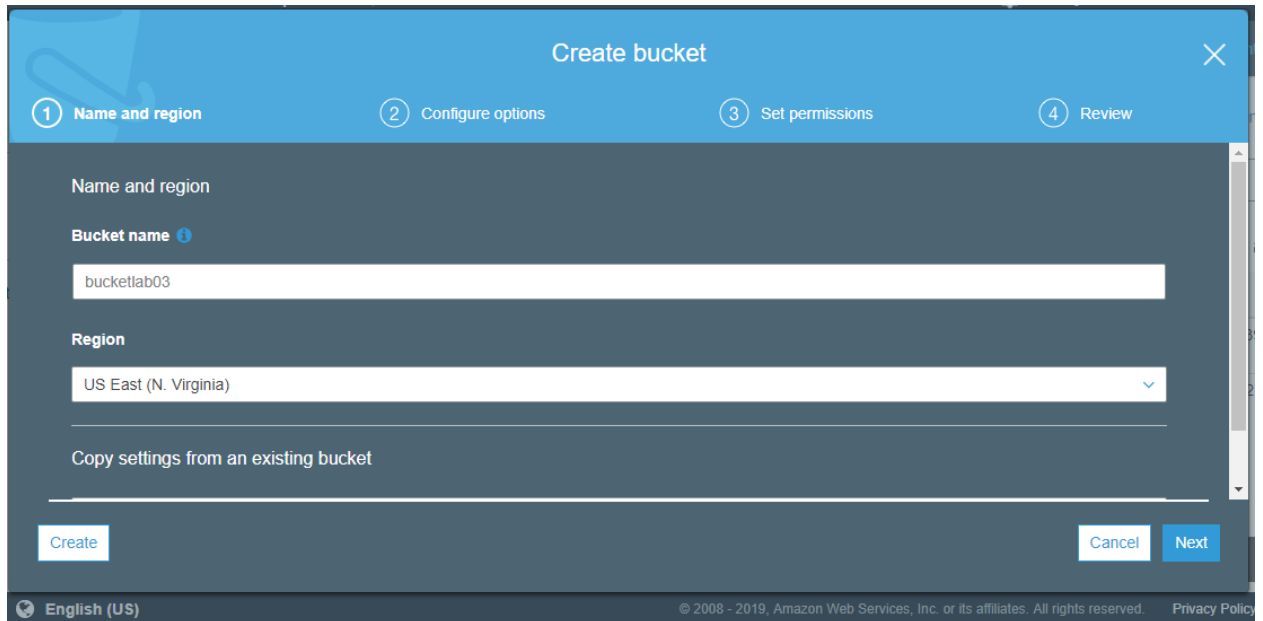


Lab Assignment 4

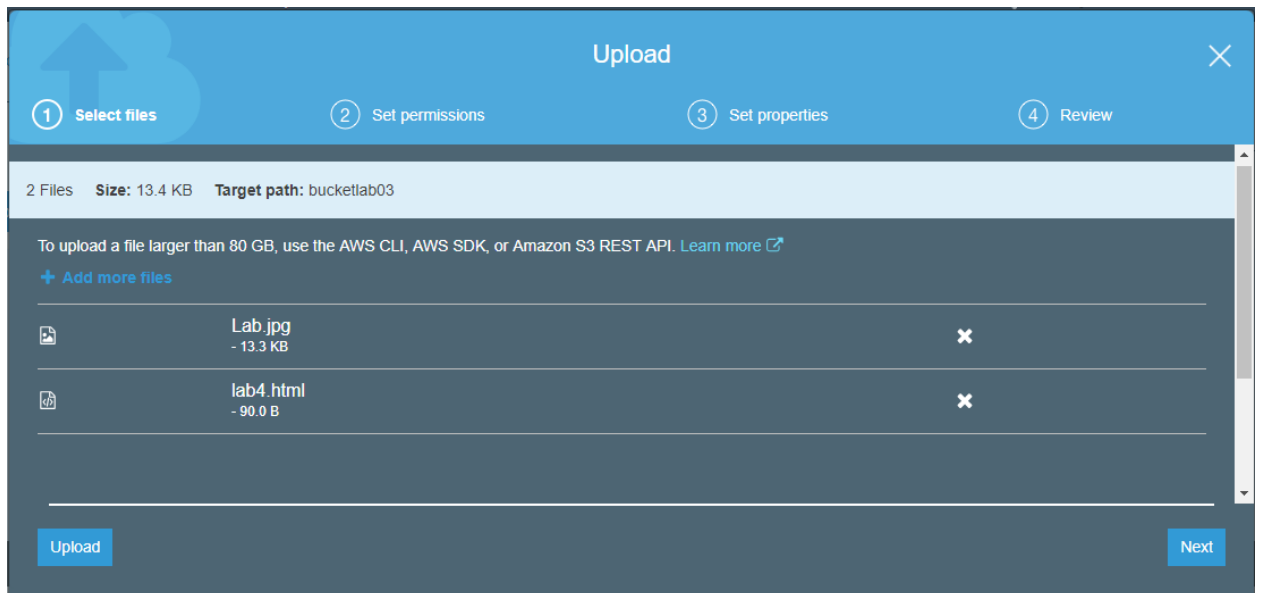
According to the assignment these are the basic steps followed:

1. Creating an S3 bucket:



The screenshot shows the 'Create bucket' wizard in the AWS Management Console. The interface has a blue header with the title 'Create bucket' and a close button. Below the header is a progress bar with four steps: 1. Name and region, 2. Configure options, 3. Set permissions, and 4. Review. The first step is active. The main area is titled 'Name and region' and contains a 'Bucket name' input field with the text 'bucketlab03', a 'Region' dropdown menu set to 'US East (N. Virginia)', and a 'Copy settings from an existing bucket' checkbox. At the bottom, there are 'Create', 'Cancel', and 'Next' buttons. The footer shows 'English (US)' and copyright information.

After creating the bucket and uploading the image. I got the following URL which showed me desired image: <https://s3.amazonaws.com/bucketlab04/Lab.jpg>



The screenshot shows the 'Upload' wizard in the AWS Management Console. The interface has a blue header with the title 'Upload' and a close button. Below the header is a progress bar with four steps: 1. Select files, 2. Set permissions, 3. Set properties, and 4. Review. The first step is active. The main area shows '2 Files' with a total size of '13.4 KB' and a 'Target path' of 'bucketlab03'. It includes a link to 'Learn more' about uploading large files. Below this is a table of files being uploaded:

File Name	Size	Actions
Lab.jpg	13.3 KB	✕
lab4.html	90.0 B	✕

At the bottom, there are 'Upload' and 'Next' buttons.

2. Creating a Web distribution in Cloud Front:

The screenshot shows the AWS CloudFront Distributions console. The left sidebar contains navigation links: Distributions, What's New, Reports & Analytics, Cache Statistics, Monitoring and Alarms, Popular Objects, Top Referrers, Usage, Viewers, and Security. The main panel is titled 'CloudFront Distributions' and includes buttons for 'Create Distribution', 'Distribution Settings', 'Delete', 'Enable', and 'Disable'. Below these are filters for 'Viewing: Any Delivery Method' and 'Any State'. A table lists the distribution details:

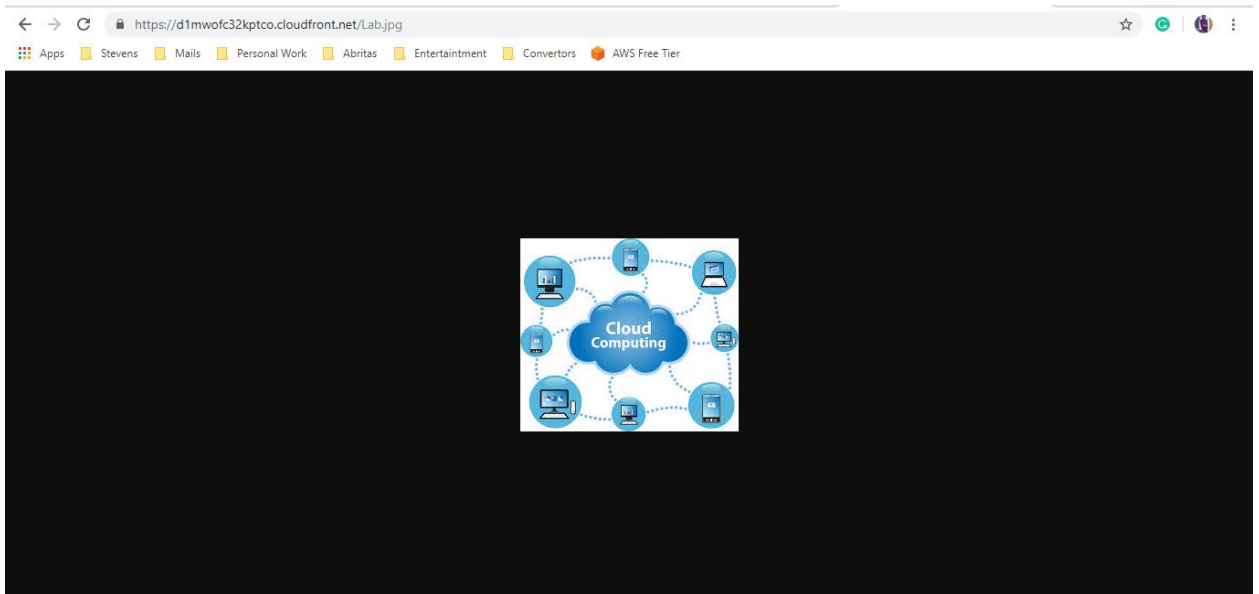
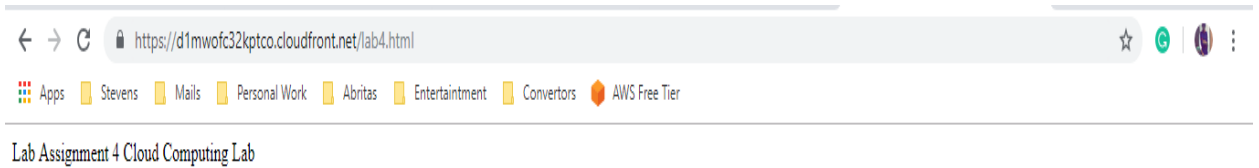
Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last Modified
Web	E3Q7MUKZ2IDVD8	d1mwofc32kptco.cl	-	bucketlab03.s3.ama	-	Deployed	Enabled	2019-04-28 22

Unless, we are using Adobe Flash Media Server with RTMP, this value is always **Web**. One can't change the delivery method for an existing distribution.

Parameters:

- Origin domain name:** As we are using the S3 bucket object I used Origin name as bucketlab03.s3.amazonaws.com
- Origin path:** As we are using from default folder we can specify it in the URL(I have used an html and jpg file for demonstration so no specific path is given over here.)
- View protocol policy:** I have changed it to "Redirect HTTP to HTTPS" so that it would be always secured.
- Restrict Bucket Access :** Yes, because we want access to the bucket objects.
- Origin Access Identity:** If you chose Yes for Restrict Bucket Access, choose whether to create a new origin access identity or use an existing one that is associated with your AWS account.
- Comment for New Identity:** If you chose Create a New Identity for Origin Access Identity, enter a comment that identifies the new origin access identity. CloudFront will create the origin access identity when you create this distribution.
- Your Identities:** If you want CloudFront to automatically grant the origin access identity the permission to read objects in your Amazon S3 bucket, choose Yes, Update Bucket Policy.
- HTTP Port:** The HTTP port that the custom origin listens on. Valid values include ports 80, 443, and 1024 to 65535. The default value is port 80.
- Cache Behaviour:** A cache behavior lets you configure a variety of CloudFront functionality for a given URL path pattern for files on your website.
- Cache Based on Selected Request Headers:** Specify whether you want CloudFront to cache objects based on the values of specified headers:
- Query String Forwarding and Caching:** CloudFront can cache different versions of your content based on the values of query string parameters.

The domain name for distribution is as follows:



After disabling the public access I get the following error:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>9E23ADB2D5F3D1EF</RequestId>
  <HostId>
    m66ihGUjnp3nW6zUw8PaozZdTBeJOa1umacTxFRCHS109ImQ4uJwhzNfahtm+qj0sMk/7xCIF3A=
  </HostId>
</Error>
```

Amazon S3 Block Public Access provides four settings. You can apply these settings in any combination to individual buckets or to entire AWS accounts. If you apply a setting to an account, it applies to all buckets that are owned by that account. The following table contains the available settings.

Name	Description
BlockPublicAcls	<p>Setting this option to TRUE causes the following behavior:</p> <ul style="list-style-type: none"> • PUT Bucket acl and PUT Object acl calls fail if the specified access control list (ACL) is public. • PUT Object calls fail if the request includes a public ACL. • If this setting is applied to an account, then PUT Bucket calls fail if the request includes a public ACL. <p>When this setting is set to TRUE, the specified operations fail (whether made through the REST API, AWS CLI, or AWS SDKs). However, existing policies and ACLs for buckets and objects are not modified. This setting enables you to protect against public access while allowing you to audit, refine, or otherwise alter the existing policies and ACLs for your buckets and objects.</p>
IgnorePublicAcls	<p>Setting this option to TRUE causes Amazon S3 to ignore all public ACLs on a bucket and any objects that it contains. This setting enables you to safely block public access granted by ACLs while still allowing PUT Object calls that include a public ACL (as opposed to BlockPublicAcls, which rejects PUT Object calls that include a public ACL). Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.</p>
BlockPublicPolicy	<p>Setting this option to TRUE causes Amazon S3 to reject calls to PUT Bucket policy if the specified bucket policy allows public access. This setting enables you to allow users to manage bucket policies without allowing them to publicly share the bucket or the objects it contains. Enabling this setting doesn't affect existing bucket policies.</p> <p>Important</p> <p>To use this setting effectively, you should apply it at the <i>account</i> level. Because a bucket policy can allow users to alter a bucket's Block Public Access settings, users who have permission to change a bucket policy could insert a policy that allows them to disable the Block Public Access settings for the bucket. If this setting is enabled for the entire account, rather than for a specific bucket, then Amazon S3 blocks public policies even if a user alters the bucket policy to disable this setting.</p>

RestrictPublicBuckets	<p>Setting this option to TRUE restricts access to a bucket with a public policy to only AWS services and authorized users within the bucket owner's account. This setting blocks all cross-account access to the bucket (except by AWS services), while still allowing users within the account to manage the bucket.</p> <p>Enabling this setting doesn't affect existing bucket policies, except that Amazon S3 blocks public and cross-account access derived from any public bucket policy, including non-public delegation to specific accounts.</p>
-----------------------	--

There are six points which dignifies my observations with the benefits and the differences using the image before CDN and after CDN:

1. **Security to the content** : AWS CloudFront is a highly-secure Content Delivery Network (CDN) that has each network and application level protection
2. **Integrating networks**: The AWS CloudFront content delivery network constructs on the increasing international AWS infrastructure that presently includes fifty-five accessible zones at intervals eighteen geographic regions nowadays.
3. **Perfromance**: The AWS CloudFront content delivery network optimizes for low latency and high information transfer speeds
4. **Programmable CDN**: With Lambda@Edge the user will simply run the code across AWS locations worldwide, permitting the user to retort to the finish users with very cheap latency.
5. **Economical**: Amazon CloudFront's evaluation is easy – the user pays just for the information transfer and requests accustomed deliver content to the customers.
6. **Deep Integration with Key AWS Services**: Amazon CloudFront is deeply integrated with and optimized to figure with fashionable AWS services together with Amazon straightforward Storage Service (**Amazon S3**), Amazon Elastic work out Cloud (**Amazon EC2**), Elastic Load equalization, and Amazon Route fifty-three to assist speed up DNS resolution of applications delivered by CloudFront

Conclusion:

Hence, we studied there is no need of semi-permanent contracts, as support for CloudFront encloses in your existing AWS Support subscription. One can also become the member of AWS CloudFront and participate in the AWS Service Delivery Program. The service delivery program verifies the APN partners with a known and verified path of delivering specific services. Also workloads to AWS customers, including Amazon CloudFront. It also gives static quality caching, live and on demand streaming , dynamic and customized content with security and DDos protection, API acceleration and efficient software distribution.