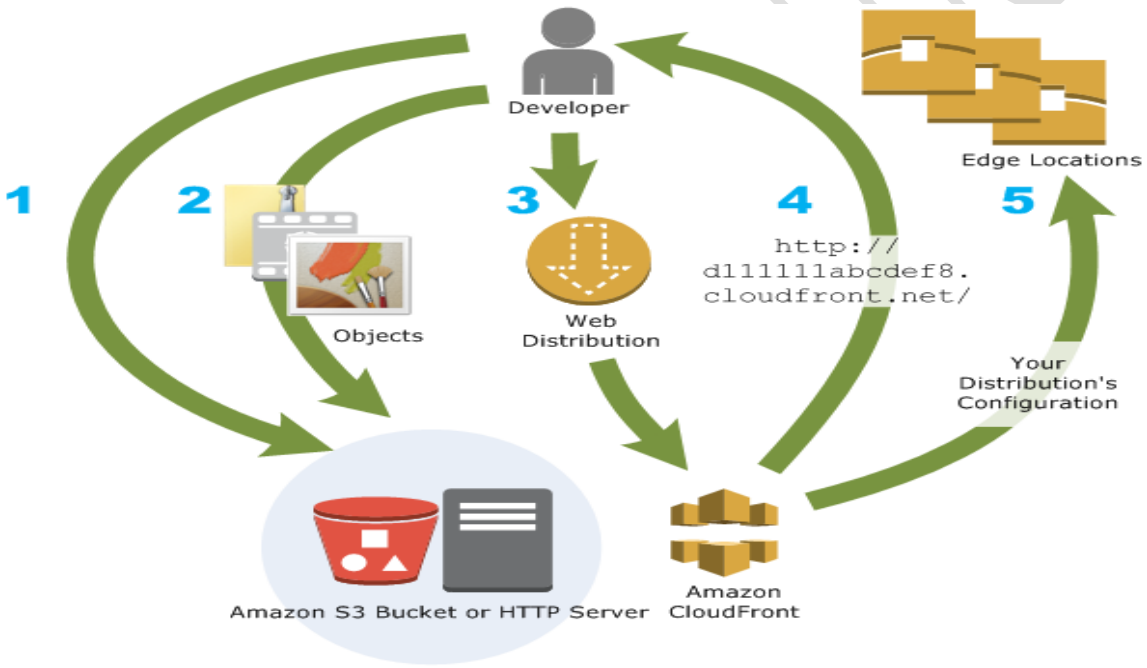
CLOUDFRONT

1. What is the purpose of Cloud Front?

Amazon Cloud Front is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. Cloud Front delivers your content through a worldwide network of data centers called edge locations.



- If the content is already in the edge location with the lowest latency, Cloud Front delivers it immediately.
- If the content is not in that edge location, Cloud Front retrieves it from an origin that you've defined—such as an Amazon S3 bucket, a Media Package channel, or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.

As an example, suppose that you're serving an image from a traditional web server, not from Cloud Front. For example, you might serve an image, sunsetphoto.png, using the URL https://example.com/sunsetphoto.png.

2.Creating a cloud front distribution

Create a distribution

This topic explains how to use the Cloud Front console to create a distribution. Other helpful topics include the following:

- To learn how to create a distribution that uses an Amazon Simple Storage Service (Amazon S3) bucket origin with origin access control (OAC), see Getting started with a basic CloudFront distribution.
- For information about using the Cloud Front APIs to create a distribution, see Create Distribution in the *Amazon Cloud Front API Reference*.
- For information about updating a distribution (for example, to add or change cache behaviors), see Update a distribution.
- To see the current maximum number of distributions that you can create for each AWS account, or to request a higher quota (formerly known as limit), see General quotas on distributions**.**

Create a Cloud Front distribution in the console

To create a distribution (console)

1. Sign in to the AWS Management Console and open the Cloud Front console at https://console.aws.amazon.com/cloudfront/v4/home.
2. In the navigation pane, choose **Distributions**, then choose **Create distribution**.
3. Specify settings for the distribution. For more information, see Distribution settings reference.
4. Save your changes.
5. After Cloud Front creates your distribution, the value of the **Status** column for your distribution will change from **Deploying** to the date and time that the distribution is deployed. If you chose to enable the distribution, it will be ready to process requests at this time.

   The domain name that Cloud Front assigns to your distribution appears in the list of distributions. (It also appears on the **General** tab for a selected distribution.)

6. When your distribution is deployed, confirm that you can access your content using your new Cloud Front URL or CNAME. For more information, see Test a distribution.

3. Hosting a website of cloud front distribution

To host a website using Amazon Cloud Front, you need to follow a series of steps that involve setting up your origin server, configuring Cloud Front, and ensuring your content is distributed efficiently and securely. Here's a detailed guide to help you through the process:

Prerequisites

AWS Account: Make sure you have an AWS account. If not, sign up at AWS.

Domain Name: You should have a domain name registered and set up to use Route 53 or another DNS service.

Content to Host: Your website's content, which could be on an S3 bucket, an EC2 instance, or another web server.

Steps to Host a Website with Cloud Front

Step 1: Prepare Your Origin Server

Step 2: Create a Cloud Front Distribution

Step 3: Update DNS Settings.

Step 4: Test Your Setup

Access Your Website: Visit your domain and check if your website is loading correctly.

Verify SSL and Performance: Ensure the SSL certificate is working and check the performance improvements due to Cloud Front's caching.

4.Implementation restrictions

Implementations of Scheme are not required to implement the whole tower of subtypes given in section Numerical types, but they must implement a coherent subset consistent with both the purposes of the implementation and the spirit of the Scheme language. For example, an implementation in which all numbers are real may still be quite useful.

Implementations may also support only a limited range of numbers of any type, subject to the requirements of this section. The supported range for exact numbers of any type may be different from the supported range for inexact numbers of that type. For example, an implementation that uses flonums to represent all its inexact real numbers may support a practically unbounded range of exact integers and rationals while limiting the range of inexact reals (and therefore the range of inexact integers and rationals) to the dynamic range of the flonum format. Furthermore the gaps between the representable inexact integers and rationals are likely to be very large in such an implementation as the limits of this range are approached.

An implementation of Scheme must support exact integers throughout the range of numbers that may be used for indexes of lists, vectors, and strings or that may result from computing the length of a list, vector, or string. The length, vector-length, and string-length procedures must return an exact integer, and it is an error to use anything but an exact integer as an index. Furthermore any integer constant within the index range, if expressed by an exact integer syntax, will indeed be read as an exact integer, regardless of any implementation restrictions that may apply outside this range.

5. Origins and behaviors

An origin is a location where Cloud Front sends requests for content that it distributes through the edge locations. Depending on your implementation you can have one or two origins. One for dynamic content (the Light sail instance in the single-server deployment option, or the Application Load Balancer in the elastic deployment option) using a custom origin. You may have a second origin to direct Cloud Front to for your static content. In the preceding reference architecture, this is an S3 bucket. When you use Amazon S3 as an origin for your distribution, you need to use a bucket policy to make the content publicly accessible.

Behaviors allow you to set rules that govern how Cloud Front caches your content, and, in turn, determine how effective the cache is. Behaviors allow you to control the protocol and HTTP methods your website is accessible by. They also allow you to control whether to pass HTTP headers, cookies, or query strings to your backend (and, if so, which ones). Behaviors apply to specific URL path patterns.