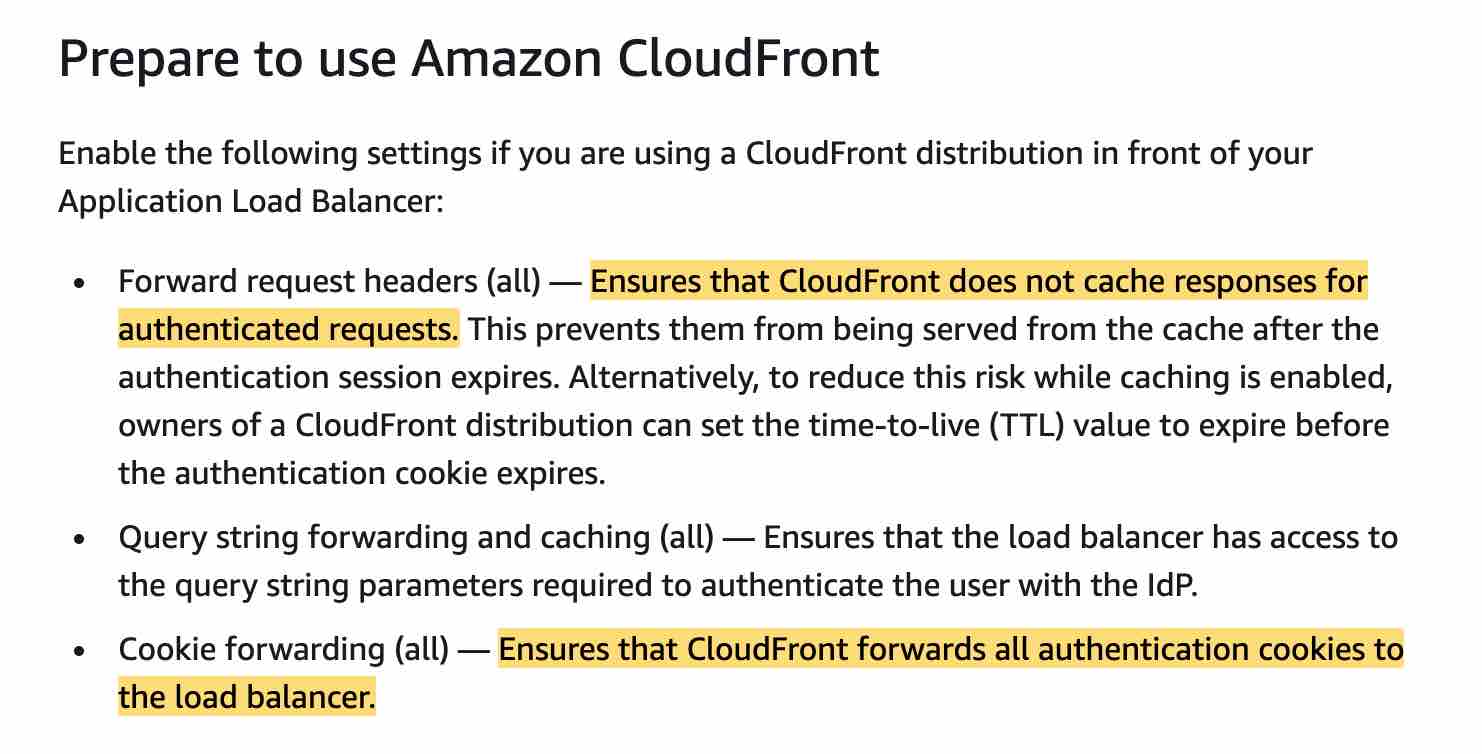
CLOUDFRONT

Please make sure that you adhere to the following configurations while using CloudFront distribution in front of your Application Load Balancer:



**CloudFront Key Pairs** - IAM users can't create CloudFront key pairs. You must log in using root credentials to create key pairs.

Q. What can I do with Amazon CloudFront?

Amazon CloudFront provides a simple API that lets you:

* Distribute content with low latency and high data transfer rates by serving requests using a network of edge locations around the world.
* Get started without negotiating contracts and minimum commitments.

Q. How do I use Amazon CloudFront?

To use Amazon CloudFront, you:

* For static files, store the definitive versions of your files in one or more origin servers. These could be Amazon S3 buckets. For your dynamically generated content that is personalized or customized, you can use Amazon EC2 – or any other web server – as the origin server. These origin servers will store or generate your content that will be distributed through Amazon CloudFront.
* Register your origin servers with Amazon CloudFront through a simple API call. This call will return a CloudFront.net domain name that you can use to distribute content from your origin servers via the Amazon CloudFront service. For instance, you can register the Amazon S3 bucket “bucketname.s3.amazonaws.com” as the origin for all your static content and an Amazon EC2 instance “dynamic.myoriginserver.com” for all your dynamic content. Then, using the API or the AWS Management Console, you can create an Amazon CloudFront distribution that might return “abc123.cloudfront.net” as the distribution domain name.
* Include the cloudfront.net domain name, or a CNAME alias that you create, in your web application, media player, or website. Each request made using the cloudfront.net domain name (or the CNAME you set-up) is routed to the edge location best suited to deliver the content with the highest performance. The edge location will attempt to serve the request with a local copy of the file. If a local copy is not available, Amazon CloudFront will get a copy from the origin. This copy is then available at that edge location for future requests.

Q. How does Amazon CloudFront provide higher performance?

Amazon CloudFront employs a global network of edge locations and regional edge caches that cache copies of your content close to your viewers. Amazon CloudFront ensures that end-user requests are served by the closest edge location. As a result, viewer requests travel a short distance, improving performance for your viewers. For files not cached at the edge locations and the regional edge caches, Amazon CloudFront keeps persistent connections with your origin servers so that those files can be fetched from the origin servers as quickly as possible. Finally, Amazon CloudFront uses additional optimizations – e.g. wider TCP initial congestion window – to provide higher performance while delivering your content to viewers.

Q. How does Amazon CloudFront speed up my entire website?

Amazon CloudFront uses standard cache control headers you set on your files to identify static and dynamic content. Delivering all your content using a single Amazon CloudFront distribution helps you make sure that performance optimizations are applied to your entire website or web application. When using AWS origins, you benefit from improved performance, reliability, and ease of use as a result of AWS’s ability to track and adjust origin routes, monitor system health, respond quickly when any issues occur, and the integration of Amazon CloudFront with other AWS services. You also benefit from using different origins for different types of content on a single site – e.g. Amazon S3 for static objects, Amazon EC2 for dynamic content, and custom origins for third-party content – paying only for what you use.

Q. How is Amazon CloudFront different from Amazon S3?

Amazon CloudFront is a good choice for distribution of frequently accessed static content that benefits from edge delivery—like popular website images, videos, media files or software downloads.

Q. What types of content does Amazon CloudFront support?

Amazon CloudFront supports content that can be sent using the HTTP or WebSocket protocols. This includes dynamic web pages and applications, such as HTML or PHP pages or WebSocket-based applications, and any popular static files that are a part of your web application, such as website images, audio, video, media files or software downloads. Amazon CloudFront also supports delivery of live or on-demand media streaming over HTTP.

Q. Does Amazon CloudFront work with non-AWS origin servers?

Yes. Amazon CloudFront works with any origin server that holds the original, definitive versions of your content, both static and dynamic. There is no additional charge to use a custom origin

Q. What is CloudFront Regional Edge Cache?

CloudFront delivers your content through a worldwide network of data centers called edge locations. The regional edge caches are located between your origin web server and the global edge locations that serve content directly to your viewers. This helps improve performance for your viewers while lowering the operational burden and cost of scaling your origin resources.

Q. Is regional edge cache feature enabled by default?

Yes. You do not need to make any changes to your CloudFront distributions; this feature is enabled by default for all new and existing CloudFront distributions. There are no additional charges to use this feature.

Q. Can I choose to serve content (or not serve content) to specified countries?

Yes, the Geo Restriction feature lets you specify a list of countries in which your users can access your content. Alternatively, you can specify the countries in which your users cannot access your content. In both cases, CloudFront responds to a request from a viewer in a restricted country with an HTTP status code 403 (Forbidden).

Q. Can I serve a custom error message to my end users?

Yes, you can create custom error messages (for example, an HTML file or a .jpg graphic) with your own branding and content for a variety of HTTP 4xx and 5xx error responses. Then you can configure Amazon CloudFront to return your custom error messages to the viewer when your origin returns one of the specified errors to CloudFront.

Q. How do I remove an item from Amazon CloudFront edge locations?

There are multiple options for removing a file from the edge locations. You can simply delete the file from your origin and as content in the edge locations reaches the expiration period defined in each object’s HTTP header, it will be removed. In the event that offensive or potentially harmful material needs to be removed before the specified expiration time, you can use the Invalidation API to remove the object from all Amazon CloudFront edge locations. You can see the charge for making invalidation requests [here](https://aws.amazon.com/cloudfront/pricing/).

Q. Is there a limit to the number of invalidation requests I can make?

If you're invalidating objects individually, you can have invalidation requests for up to 3,000 objects per distribution in progress at one time. This can be one invalidation request for up to 3,000 objects, up to 3,000 requests for one object each, or any other combination that doesn't exceed 3,000 objects.

If you're using the \* wildcard, you can have requests for up to 15 invalidation paths in progress at one time. You can also have invalidation requests for up to 3,000 individual objects per distribution in progress at the same time; the limit on wildcard invalidation requests is independent of the limit on invalidating objects individually. If you exceed this limit, further invalidation requests will receive an error response until one of the earlier request completes.

You should use invalidation only in unexpected circumstances; if you know beforehand that your files will need to be removed from cache frequently, it is recommended that you either implement a versioning system for your files and/or set a short expiration period.

Q. What types of HTTP requests are supported by Amazon CloudFront?

Amazon CloudFront currently supports GET, HEAD, POST, PUT, PATCH, DELETE and OPTIONS requests.

Q. Does Amazon CloudFront cache POST responses?

Amazon CloudFront does not cache the responses to POST, PUT, DELETE, and PATCH requests – these requests are proxied back to the origin server. You may enable [caching](https://aws.amazon.com/caching/cdn/) for the responses to OPTIONS requests.

Q. What are WebSockets?

WebSocket is a real-time communication protocol that provides bidirectional communication between a client and a server over a long-held TCP connection. By using a persistent open connection, the client and the server can send real-time data to each other without the client having to frequently reinitiate connections checking for new data to exchange. WebSocket connections are often used in chat applications, collaboration platforms, multiplayer games, and financial trading platforms. Refer to our documentation to learn more about [using the WebSocket protocol](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/distribution-working-with.websockets.html) with Amazon CloudFront.

Q. How do I enable my Amazon CloudFront distribution to support the WebSocket protocol?

You can use WebSockets globally, and no additional configuration is needed to enable the WebSocket protocol within your CloudFront resource as it is now supported by default.

Q. When is a WebSocket connection established through Amazon CloudFront?

Amazon CloudFront establishes WebSocket connections only when the client includes the 'Upgrade: websocket' header and the server responds with the HTTP status code 101 confirming that it can switch to the WebSocket protocol.

Q. Does Amazon CloudFront support secured WebSockets over TLS?

Yes. Amazon CloudFront supports encrypted WebSocket connections (WSS) using the SSL/TLS protocol.

Q: What is the maximum size of a file that can be delivered through Amazon CloudFront?

The maximum size of a single file that can be delivered through Amazon CloudFront is 20 GB. This limit applies to all Amazon CloudFront distributions.

Q: What logging capabilities are available with Amazon CloudFront?

When you create or modify a CloudFront distribution, you can enable access logging. CloudFront provides two ways to log the requests that are delivered from your distributions: Standard logs and Real-time logs.  
  
CloudFront standard logs are delivered to the Amazon S3 bucket of your choice (log records are delivered within minutes of a viewer request). When enabled, CloudFront will automatically publish detailed log information in a W3C extended format into an Amazon S3 bucket that you specify. Access logs contain detailed information about each request for your content, including the object requested, the date and time of the request, the edge location serving the request, the client IP address, the referrer, the user agent, the cookie header, and the result type (for example, cache hit, or miss, or error). CloudFront doesn’t charge for standard logs, though you incur Amazon S3 charges for storing and accessing the log files.  
  
CloudFront real-time logs are delivered to the data stream of your choice in Amazon Kinesis Data Streams (log records are delivered within seconds of a viewer request). You can choose the sampling rate for your real-time logs—that is, the percentage of requests for which you want to receive real-time log records. You can also choose the specific fields that you want to receive in the log records. CloudFront real-time logs contain all the same data points as the standard logs and also contain certain additional information about each request such as viewer request headers, and country code, in a W3C extended format. CloudFront charges for real-time logs, in addition to the charges you incur for using Kinesis Data Streams.

Q: What are the different log destination options available?  
CloudFront standard logs are delivered to your S3 bucket. You can also use the integration build by third party solutions such as DataDog and Sumologic to create dashboards from these logs.  
  
The real-time logs are delivered to your Kinesis Data Stream. From Kinesis Data Streams, the logs can be published to Amazon Kinesis Data Firehose. Amazon Kinesis Data Firehose supports easy data delivery to Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and service providers like Datadog, New Relic, and Splunk. Kinesis Firehose also supports data delivery to a generic HTTP endpoint.

Q: How many Kinesis shards do I need in Kinesis Data Stream?  
Use the following steps to estimate the number of shards you need:

1. Calculate (or estimate) the number of requests per second that your CloudFront distribution receives. You can use the CloudFront usage reports or the CloudFront metrics to help you calculate your requests per second.
2. Determine the typical size of a single real-time log record. A typical record that includes all available fields is around 1 KB. If you’re not sure what your log record size is, you can enable real-time logs with a low sampling rate (for example, 1%), and then calculate the average record size using monitoring data in Kinesis Data Streams (total number of records divided by total incoming bytes).
3. Multiply the number of requests per second (from step 1) by the size of a typical real-time log record (from step 2) to determine the amount of data per second that your real-time log configuration is likely to send to the Kinesis data stream.
4. Using the data per second, calculate the number of shards that you need. A single shard can handle no more than 1 MB per second and 1,000 requests (log records) per second. When calculating the number of shards that you need, we recommend adding up to 25% as a buffer.

For example, assume your distribution receives 10,000 requests per second, and that your real-time log records size is typically 1 KB. This means that your real-time log configuration could generate 10,000,000 bytes (10,000 multiplied by 1,000), or 9.53 MB, per second. In this scenario you would need just 10 Kinesis shards. You should consider creating at least 12 shards to have some buffer.

Q: What is Lambda@Edge?

[Lambda@Edge](https://aws.amazon.com/lambda/edge/) allows you to run code at global AWS edge locations without provisioning or managing servers, responding to end users at the lowest network latency. You just upload your Node.js/Python code to [AWS Lambda](https://aws.amazon.com/lambda/) and configure your function to be triggered in response to Amazon CloudFront requests (i.e., when a viewer request lands, when a request is forwarded to or received back from the origin, and right before responding back to the end user). The code is then ready to execute at every AWS edge location when a request for content is received, and scales with the volume of requests across CloudFront edge locations. Learn more in our [documentation](http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/lambda-at-the-edge.html).

Q: What events can be triggered with Amazon CloudFront?

Your functions will automatically trigger in response to the following Amazon CloudFront events:

* Viewer Request - This event occurs when an end user or a device on the Internet makes an HTTP(S) request to CloudFront, and the request arrives at the edge location closest to that user.
* Viewer Response - This event occurs when the CloudFront server at the edge is ready to respond to the end user or the device that made the request.
* Origin Request - This event occurs when the CloudFront edge server does not already have the requested object in its cache, and the viewer request is ready to be sent to your backend origin webserver (e.g. Amazon EC2, or Application Load Balancer, or Amazon S3).
* Origin Response - This event occurs when the CloudFront server at the edge receives a response from your backend origin webserver.