Using Cloudfront For CDN

Basic Setup

To setup a CDN using Cloudfront you first need to create an S3 bucket and make it public. In this example we will use pslampdemo.s3.amazonaws.com

Note that when setting up a cloudfront distribution you will need to assign an SSL certificate. See: https://impressto.net/aws-setup-ssl-certificates

Once your public S3 bucket has been created to to the Cloudfront console and create a new distribution. Go to https://console.aws.amazon.com/cloudfront/home?region=us-east-2#create-distribution

1. Select one of the S3 buckets we are using for the CDN.
2. For the origin path we will leave it empty so we can use the root folder of the S3 bucket.
3. Select the HTTP > HTTPS redirect as a precatution to prevent accidental use of assets on HTTP
4. For alternative domain names add the domain name we will be using for the CDN. This will be added to route53 as a CNAME record.
5. Select the ssl certificate (this is one we upload ourselves)
6. Click the Create Distribution button. It takes several minutes for a distribution to generate but that is ok as we have work to do now with route53.
7. Click on the new distribution to get the url . You can not look for the new domain name for the distribution. It will look something like: dr8thfc1fd2g.cloudfront.net
8. copy the domain and head over to Route53 – https://console.aws.amazon.com/route53/home?region=us-east-2
9. Add the CNAME record linking pdlampdemo.com to the cloudfront distribution domain (e.g. dr8thfc1fd2g.cloudfront.net)
10. That’s it.

Select one of the S3 buckets we are using for the CDN. For the origin path we will leave it empty so we can use the root folder of the S3 bucket. Set a HTTP > HTTPS redirect as a precatution to prevent accidental use of assets on HTTP.

For alternative domain names add the domain name we will be using for the CDN. This will be added to route53 as a CNAME record. Select the ssl certificate (this is one we uploaded ourselves earlier).

Click on the new distribution to get the url . You can not look for the new domain name for the distribution. It will look something like: dr8thfc1fd2g.cloudfront.net.

Copy the domain and head over to Route53 –

https://console.aws.amazon.com/route53/home?region=us-east-2#resource-record-sets:Z34JVZ4MBJ5FQW

Add the CNAME record linking pslampdemo.com to the cloudfront distribution domain (e.g. dr8thfc1fd2g.cloudfront.net). After saving that you will be able to access the S3 assets with the CDN domain.

Enabling CORS for CDN

> https://docs.aws.amazon.com/AmazonS3/latest/user-guide/add-cors-configuration.html
>
> https://aws.amazon.com/premiumsupport/knowledge-center/no-access-control-allow-origin-error/

You will needed to enable CORS for the s3 bucket. Navigate to the S3 bucket on AWS and click the Cors Configuration button.

Add the following XML and save

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
</CORSRule>
</CORSConfiguration>
```

Enable Header Whitelisting in the Cloudfront Distribution

To forward the headers from your CloudFront distribution, follow these steps:

1. Open your distribution from the CloudFront console.
2. Choose the **Behaviors** tab.
3. Choose **Create Behavior**, or choose an existing behavior, and then choose **Edit**.

1. For **Cache Based on Selected Request Headers**, choose **Whitelist**.
2. Under **Whitelist Headers**, choose headers from the menu on the left, and then choose **Add**.

- Access-Control-Request-Headers
- Access-Control-Request-Method
- Origin

Enable GZip Compression

By default gzip compression is off. To turn it on you will need to edit the behavoir setting for the Cloudfront distribution.

1. Select the distribution and click the “Distribution Settings” button
2. Select the “Behaviors” tab then click the “Edit” button
3. Set the cache policy to “Managed-CacheOptimization”

Enabling Gzip Compresson

Open the cloudfront page then select the distribution.

Click the behavoir tag.

Select the behavior and click Edit

Set the Cache Policy to Managed-CachingOptimized and turn on Compress Object Automatically.

Invalidating Cloudfront Files

To clear the files from cache (ssl update of emergency fixes after a deployment) follow these steps:

1. Go to the cloudfront distribution page, select the distribution page.
2. Select the distribution for which you want to invalidate files.
3. Choose Distribution Settings.
4. Choose the Invalidations tab.
5. Choose Create Invalidation.
6. For the files that you want to invalidate, enter one invalidation path per line. For information about specifying invalidation paths, see Specifying the Files to Invalidate.
7. Important
   Specify file paths carefully. You can’t cancel an invalidation request after you start it.
8. Choose Invalidate.

How S3 paths become CDN paths

This s3 path

> https://pslampdemo.s3.us-east-2.amazonaws.com/website/images/kitten.png

now works as:

https://dr8thfc1fd2g.cloudfront.net/website/images/broadcast-marketplace.png

which in turn with a cname record in route53 becomes:

> https://pslampdemo.com/website/images/broadcast-marketplace.png

Additional info:

> https://console.aws.amazon.com/cloudfront/home?region=us-east-2#create-distribution:
>
> https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-values-alias.html#rrsets-values-alias-alias-target
>
> https://aws.amazon.com/blogs/aws/new-gzip-compression-support-for-amazon-cloudfront/

Using CDN for WordPress

We use the wp-offload plugin to host wordpress media files on S3. This allows us to upload images to a standard folder which is then accesible via CDN.

Make sure to read the instruction and set your CDN url accordingly.

Revision #1
Created 2024-06-14 01:11:19 UTC by peterd
Updated 2024-06-14 01:12:07 UTC by peterd