Making a Secure Static Web Site in Amazon S3

Info Used adding a DNS record

You need to add a record for your subdomain via your DNS provider.

CNAME record LHS

```
blog
```

RHS You wantthe cloudfront distribution url here.

```
gobledeegook.cloudfront.net
```

Info Used Setting Up our bucket site

Bucket Name

It must match the domain you will use.

```
blog.bernatchez.net
```

Bucket Label

```
https-blog-site
```

Landing page containing language selector implementation

```
index.html
```

Landing page content

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Select Language</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      padding: 50px;
      background: #f4f4f4;
    }

    h1 {
      margin-bottom: 20px;
    }

    .language-options {
      display: flex;
      justify-content: center;
```

```
      gap: 20px;
      margin-top: 30px;
    }

    .language-options a {
      text-decoration: none;
      border: 1px solid #ccc;
      border-radius: 8px;
      padding: 10px 20px;
      background: white;
      font-size: 18px;
      transition: background 0.3s;
    }

    .language-options a:hover {
      background: #eaeaea;
    }
  </style>
</head>
<body>
  <h1>blog.bernatchez.net language selector</h1>
  <p>You will be redirected shortly based on your browser language...</p>
  <p>Or select a language</p>

blog.bernatchez.net/.en/
  <div class="language-options">
    <a href="./lang-version.en/index.html">■■ English</a>
    <a href="./lang-version.fr/index.html">■■ Français</a>
    <a href="./lang-version.es/index.html">■■ Español</a>
  </div>

  <script>
    // Define supported language codes and redirect paths
    const languageRedirects = {
      'en': './lang-version.en/index.html',
      'fr': './lang-version.fr/index.html',
      'es': './lang-version.es/index.html'
    };

    // Get the browser's primary language (e.g., 'en', 'fr')
    const userLang = navigator.language.slice(0, 2).toLowerCase();

    // Check if we support the detected language
    const redirectTo = languageRedirects[userLang] || './lang-version.en/index.html';

    // Wait 5 seconds before redirecting
    setTimeout(() => {
      window.location.href = redirectTo;
    }, 60000);
  </script>
</body>
</html>
```

404 page

```
oops.html
```

404 page content

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Page Not Found</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            color: #333;
            text-align: center;
            padding: 50px;
        }
        .container {
            background-color: #fff;
            padding: 30px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
            display: inline-block;
        }
        h1 {
            color: #d9534f; /* Red for error */
        }
        a {
            color: #007bff;
            text-decoration: none;
        }
        a:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Oops: Page Not Found</h1>
        <p>Sorry, but the page you're looking was not found.</p>
        <p>It may have been moved, deleted, or there might be a typo in the URL.</p>
        <p>
            <a href="/">Go back to the homepage</a>
        </p>
    </div>
</body>
</html>
```

Bucket policy

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

The bucket policy original resource

```
https://docs.aws.amazon.com/AmazonS3/latest/userguide/HostingWebsiteOnS3Setup.html

Note: replaced  "Bucket-Name" in the resource to make it "blog.bernatchez.net"
```

Aws Region

```
US East (N. Virginia) us-east-1
```

Info Used For Cloudfront distribution

alternate domain name

```
blog.bernatchez.net
```

super domain

This is where the validation email goes

```
bernatchez.net
```

original domain

```
blog.bernatchez.net.s3-website-us-east-1.amazonaws.com
```

default route object

```
index.html
```

Walk through

This is a walk through exercise of creating a rudimentary static web site.

- Sign in to the AWS console.
- Go to S3

- Click on create bucket.
- Supply *Bucket Name*
- Supply *Bucket label*
- Supply *Aws Region*
- Uncheck Block all public access, and acknowledge that.
- Press Create Bucket.
- Upload the *Landing page* to the bucket.
- Upload the *404 page* to the bucket.
- **Enable static website**
    - Press **Properties**
    - Go down to static website hosting and press **Edit**
    - press **enable**
    - Supply *Landing page* for index document value
    - Supply *404 page* for error page value
    - Press **Save Changes**
- **Attach a Bucket Policy**
    - Press **Permissions**
    - Press **Edit** bucket policy. and Paste *Bucket policy*.
    - Press **Save**
- Get an SSL certificate
    - Search for certificate manager in our console
    - Click on Certificat Manager
    - Click on Request Certificate
    - Public certificate. Next.
    - For fully qualified domain name put in: *Bucket Name*
    - Use email validation.
    - Validation domain is where the email will go. Use : *super domain*
    - Add a tag pair such as : "certify", *bucket name*
    - Press request You get a successfully requested certificate, status is pending validation.
    - You will receive an email you need to follow instructions to validate. Your certificate validation will change to issued.

- Login to you DNS server and add a record:
    - *LHS* CNAME *RHS*
- Try it out in browser to make sure it is working (without https) *Bucket Name* should yield our hello page *Bucket Name*/foo.html should yield our error page
- Cloudfront distribution
    - Go to cloud front and press create cloudfront distribution
    - Choose previous create distribution page. I cant seem to find how to add an alternate with the current one.
    - Again following the advice, use the amazon url instead of ours as the original domain.
    - Enable redirect http to https
    - Add alternate domain name:
    - Choose the ssl certificate
    - Default route object
    - Create distribution

  The cloudfront distribution set up was not working for me. I spent days trying to find other ways of achieving the same thing, using other services, looking for reverse proxy as a service to use instead. Until I realized that I had supplied to wrong value for RHS above. I game the domain of the http bucket, what I need to give it was the domain of the cloudfront distribution. DOH! When I fixed that it all worked like a charm.