

Migrate to AWS S3, Cloudfront, Route53

Currently if someone visits universityofbath.cn, they get a 404, because it does not redirect to the www subdomain.

Marcaria, our registrar's DNS is not complex enough to allow both this redirect and domain verification for Google Cloud Platform on the apex domain.

This proposes to move all of the infrastructure to AWS and use

- S3 for filestorage
- Cloudfront for CDN distribution
- Route53 for DNS lookup
- [s3_website gem](#) for deployment and Cloudfront cache invalidation
- optionally use Hugo or Jekyll to build the site for deployment

Considerations

- Cost
- Downtime during migration

Cost

AWS pricing is pretty close to GCP, under \$3 per month.

Downtime

There is a danger of downtime while migrating, both hosting and DNS/delivery infrastructure. If we were to immediately change DNS provider and drop the content from GCP, users who have old cached nameserver records cached would be directed to an empty GCP bucket. We need to run both hosts simultaneously for a short while.

To avoid the downtime, before changing DNS provider we should:

1. Deploy all content to the S3 bucket
2. Deploy all content to the CDN
3. Connect Route53's DNS to the CDN

Once that is done, we should tell Marcaria to use Route53 DNS instead of its own.

Users requesting the resource at this point will either be using the old records from Marcaria pointing to GCP, or the new records on Route53 pointing to Cloudfront, until the Marcaria records expire.

The GCP hosted content can be removed after a few days, when `dig` consistently returns the Rout53 nameservers.

Ongoing maintenance of the site

Once the site is on AWS, deployment of updates become easy, using the `s3_website` gem and Rake.

Once the site files are ready to be updated, the `s3_website` gem will push the files to the S3 bucket, and invalidate the Cloudfront cache so we can be sure visitors are receiving the most up to date content.

This will require us to set up a service account with AWS IAM and store the credentials locally in environment variables, so that `s3_website` can authenticate. These credentials should have read/write permissions on the S3 bucket, and permissions to update Cloudfront.

Static site generator

Moving the site into a static site generator would allow for greater automation of the build process as the site becomes more complex and we extract content into partials. In the short term, using Jekyll without a theme will streamline deployment because `s3_website` is preconfigured to deploy Jekyll by default.