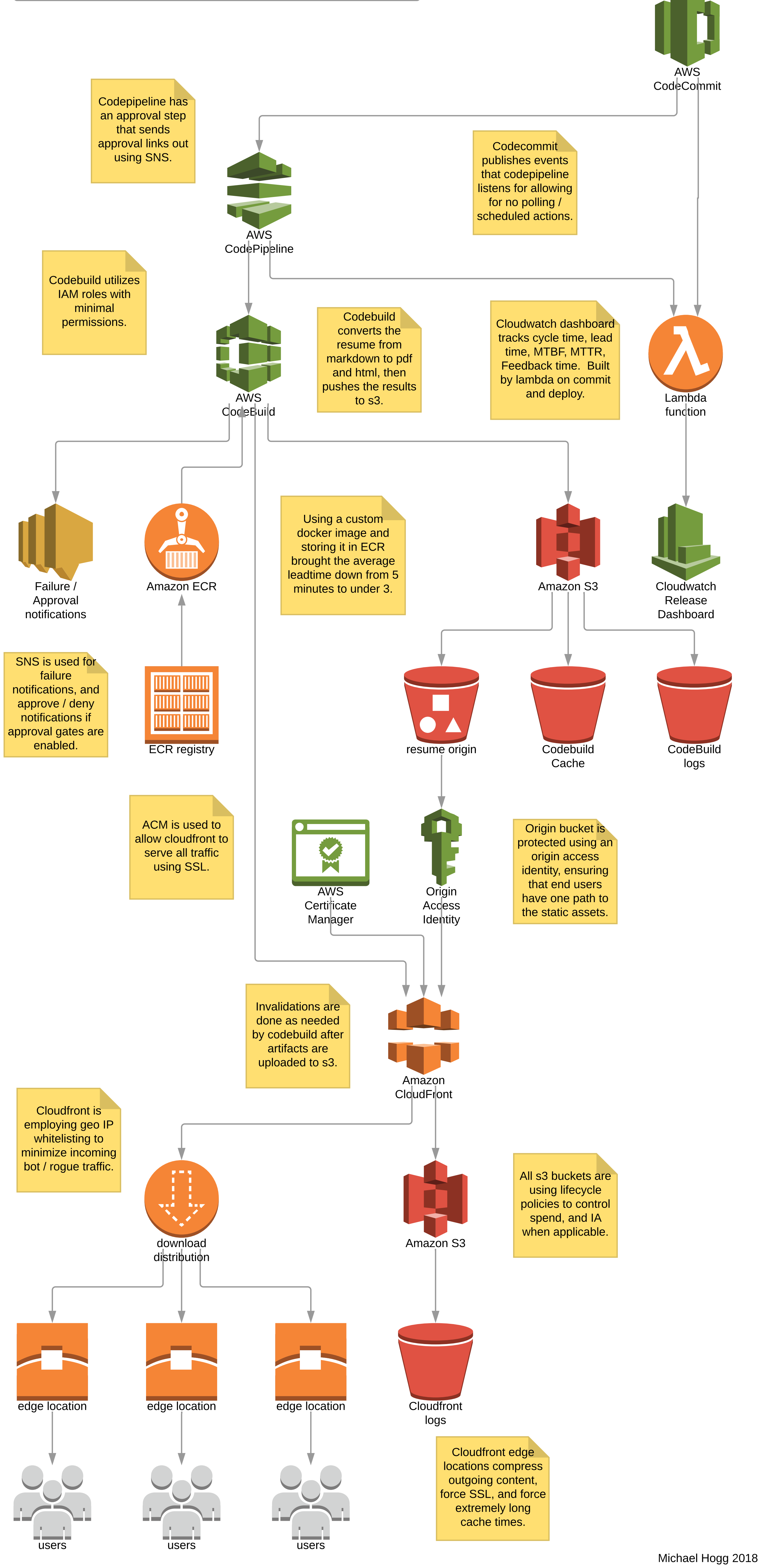
resume.gecko.tc project

Repo Url: https://github.com/jedinerd/resume

Forked from: https://github.com/mwhite/resume

Goals:

- Convert resume into interoperable format.
- Design, build and implement entirely using AWS backed services.
- Maintain serverless design.
- Employ fully automated SDLC proccess.
- Maintain extremely high cache hit ratio using targeted invalidations.
- Utilize metric/data driven approach to drive improvements.
- Utilize minimum access permission model.



Resume project was designed 100% in amazon, including using the Cloud9 IDE

Codepipeline has an approval step that sends approval links out using SNS.

Codecommit publishes events that codepipeline listens for allowing for no polling / scheduled actions.

Codebuild utilizes IAM roles with minimal permissions.

Codebuild converts the resume from markdown to pdf and html, then pushes the results to s3.

Cloudwatch dashboard tracks cycle time, lead time, MTBF, MTTR, Feedback time. Built by lambda on commit and deploy.

Using a custom docker image and storing it in ECR brought the average leadtime down from 5 minutes to under 3.

SNS is used for failure notifications, and approve / deny notifications if approval gates are enabled.

ACM is used to allow cloudfront to serve all traffic using SSL.

Origin bucket is protected using an origin access identity, ensuring that end users have one path to the static assets.

Invalidations are done as needed by codebuild after artifacts are uploaded to s3.

Cloudfront is employing geo IP whitelisting to minimize incoming bot / rogue traffic.

All s3 buckets are using lifecycle policies to control spend, and IA when applicable.

Cloudfront edge locations compress outgoing content, force SSL, and force extremely long cache times.

Michael Hogg 2018