URL Shortening Service

Daratha Galkissa

Requirement

The requirement is to create a URL shorting service.

- Provide the shorten form, when the URL is given.
- Provide the original URL when the short form is given.

How the Problem Solved

- Generate incremental number for each URL: this is the unique number represents URL.

- Save the unique number and the URL in a key value store.

- Encode the unique number
  For encoding 62 characters were used. So the unique decimal number converted to Base 62.

  Following is the Alphabet:
  abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

this alphabet can be changed according to the requirement. For an example: following I have provided two other alphabets.

```
Base 36 - "0123456789abcdefghijklmnopqrstuvwxyz";
Base 58 - "abcdefghijkmnopqrstuvwxyzABCDEFGHJKLMNPQRSTUVWXYZ123456789";
```

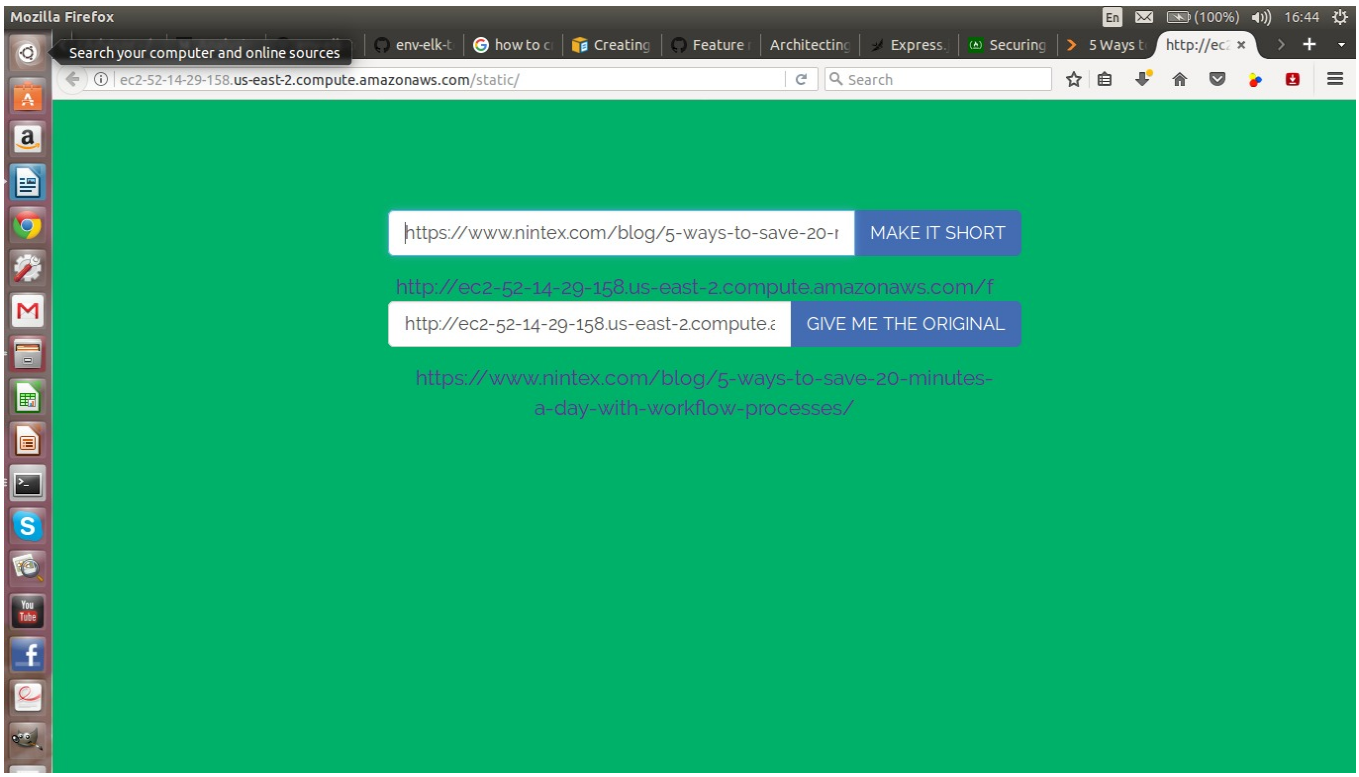
- Returns the encoded number appended to the base URL

- When the request comes to the above generated short URL, extract the unique key from the short URL,
  get the long URL by passing the unique key.

- Redirect the request to the particular long URL.

Original URL :
https://www.nintex.com/blog/5-ways-to-save-20-minutes-a-day-with-workflow-processes/

Generated URL :
http://ec2-52-14-29-158.us-east-2.compute.amazonaws.com/f



Links

App

http://ec2-52-14-29-158.us-east-2.compute.amazonaws.com/static/index.html

Monitoring

http://ec2-52-14-29-158.us-east-2.compute.amazonaws.com:8080/containers/

Github

https://github.com/daratha/shorturl

Technologies

Front End
html, css, Axios library

Back End
Node js, Express js

Unit Testing
Mocha, Chai(), Istanbul(Code Coverage)

Database
Redis

Web Server / Load Balancer
Nginx

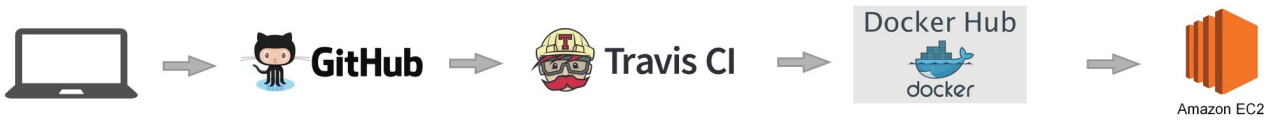
Deployment
Amazon AWS EC2, Docker, Docker Compose

Git Repository
Github

Continues Integration
Travis CI

Monitoring
Google cAdvisor

CI/CD Pipeline



CI CD Pipeline

Developer commits to github,

Gihub notifies Travis,

Travis pulls the code: build, run unit test, make docker images, push to docker hub.

EC2 instance pull the images and start them as defined in the docker-compose.yml. Running the docker compose is a manual process.

Architecture

Nginx does two jobs here,
servers static web pages and act as a load balancer to the node js application.

Nodejs Application saves data to Redis key value store.

Nginx, Nodejs, Redis all are in docker containers.