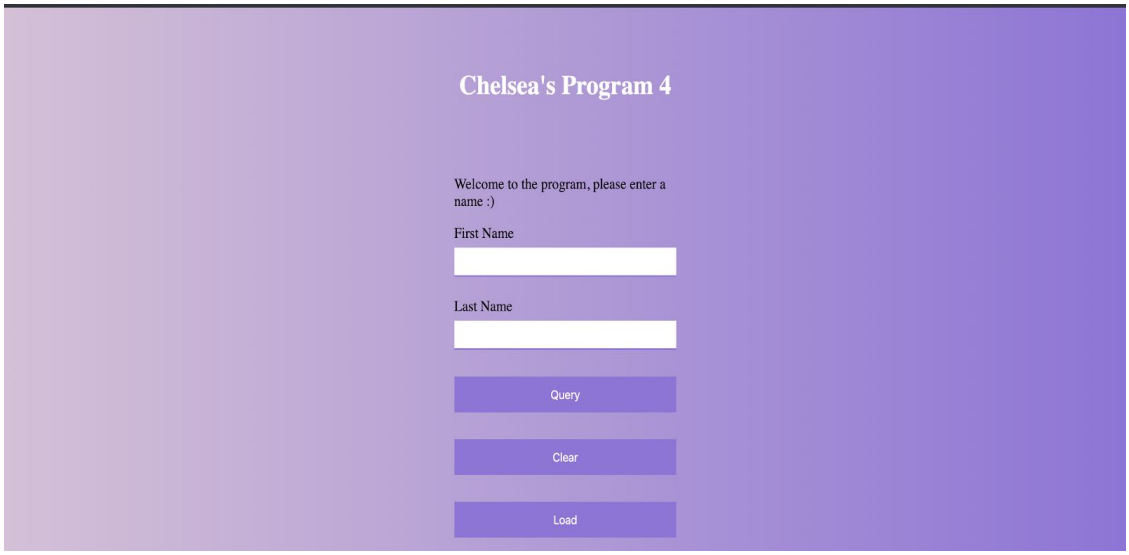
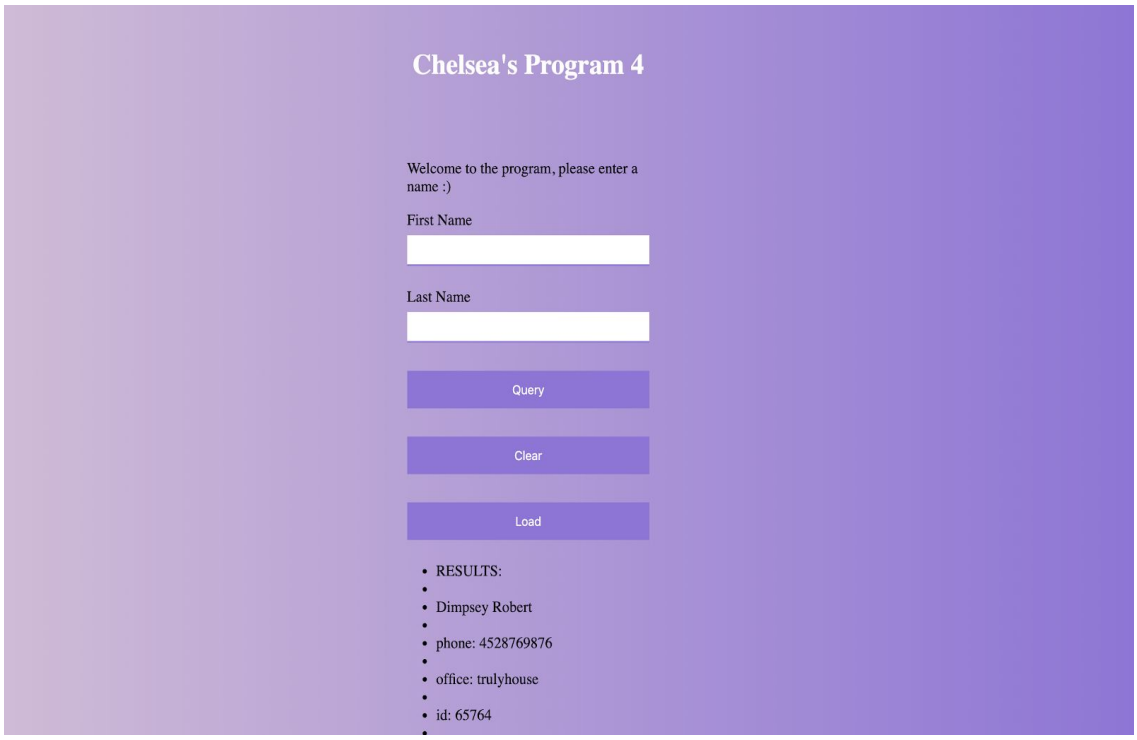
CSS490 Cloud Computing
Chelsea Ip Tze Hwan
Program 4 Official Report

The program is Python 2.7 running on 64bit Amazon Linux/2.9.4. To run it locally, you may need to install all the requirements with this command : pip **install** -r **requirements.txt. Followed by, running python application.py.** It might sometimes attempt to throw an import module error, you could just simply do a pip install (the missing import) to get rid of it.

Website URL : http://newflaskenv.jnaxirau7y.us-east-1.elasticbeanstalk.com/
Object URL : https://chelsfriedchickenloveryas637.s3.amazonaws.com/input.txt





CSS490 Cloud Computing
Chelsea Ip Tze Hwan
Program 4 Official Report

Use Cases

- When the **LOAD** data is hit, the website will load data from an object stored at a given url.
- Required to parse and load into a <key, value> NoSQL store
- Note that the **LOAD** button can be hit multiple times
- When **CLEAR** data is hit, the blob will be removed and the table will be deleted as well.
- After the data is loaded, users can type in the first name and last name
- When a **QUERY** button is hit, results will be loaded on the website

Files

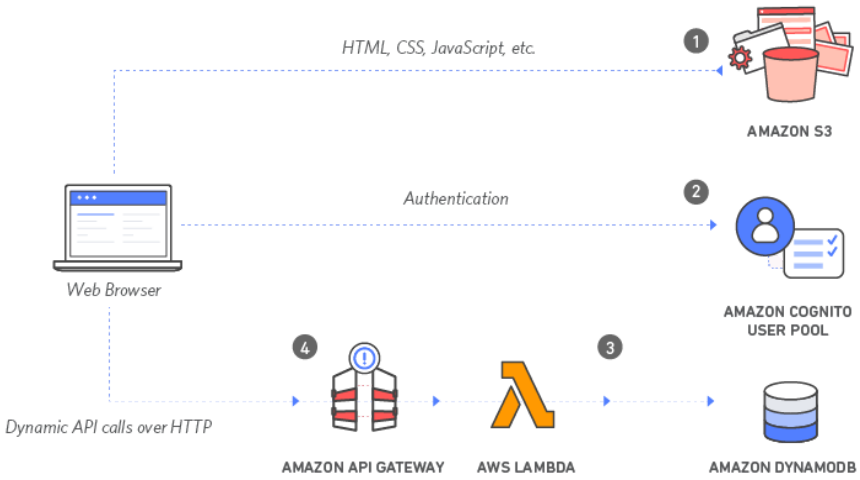
- Requirements.txt
    - pip **install** -r **requirements.txt** (Flask==1.0.2/urllib3/botocore..etc)

- Templates.py
    - Base.html
    - Error.html
    - Query,html
    - Clear.html
    - Load.html
    - Flask.html

- Application.py
    - query(first, last)
        - Looks up the DynamoDB database created for the first and last names for the person and simply display their attributes. The program is made to be user friendly as it will flash messages if the person's name is incorrect or entered incomplete.
    - clear()
        - Clears all the first and last names in the database, clear_s3() deletes the bucket itself. If user attempts to clear a name from the database that does not exist, then an exception will be thrown.
    - load()
        - A database will be created then the awsCall() function will search for the target_url before setting the object. The parseData function assumes that the attributes will come after the equal sign. When a person is successfully loaded into the database, update_DB will show a success message.

CSS490 Cloud Computing
Chelsea Ip Tze Hwan
Program 4 Official Report

Summary Architecture



- A bucket will be created in Amazon S3 where Professor Dimpsey's input.txt will be loaded. Press load on the website to insert first and last names to the database.
- I used AWS CLI/EB CLI to deploy my FLASK application -- I ran eb configure to fill in my secret key, and such. Followed some helpful tutorials, was stuck on adding wsgi.path to application.py. For some kind of reason, ebextension was giving me some minor problems in the beginning.
- Once the first, last names and attributes are loaded into the database -- the user can easily query information from the website.It scans for the first/last name given and returns the attributes belonging to the specific person.
- User can press clear which removes the bucket from S3 and subsequently deletes the entire database by scanning for each key.
- Finally, the application will be deployed on Elastic Beanstalk where the environment is successfully launched. A cloud watch alarm and instance is also created. Here we can download error logs as well as monitor the application's health respectively.

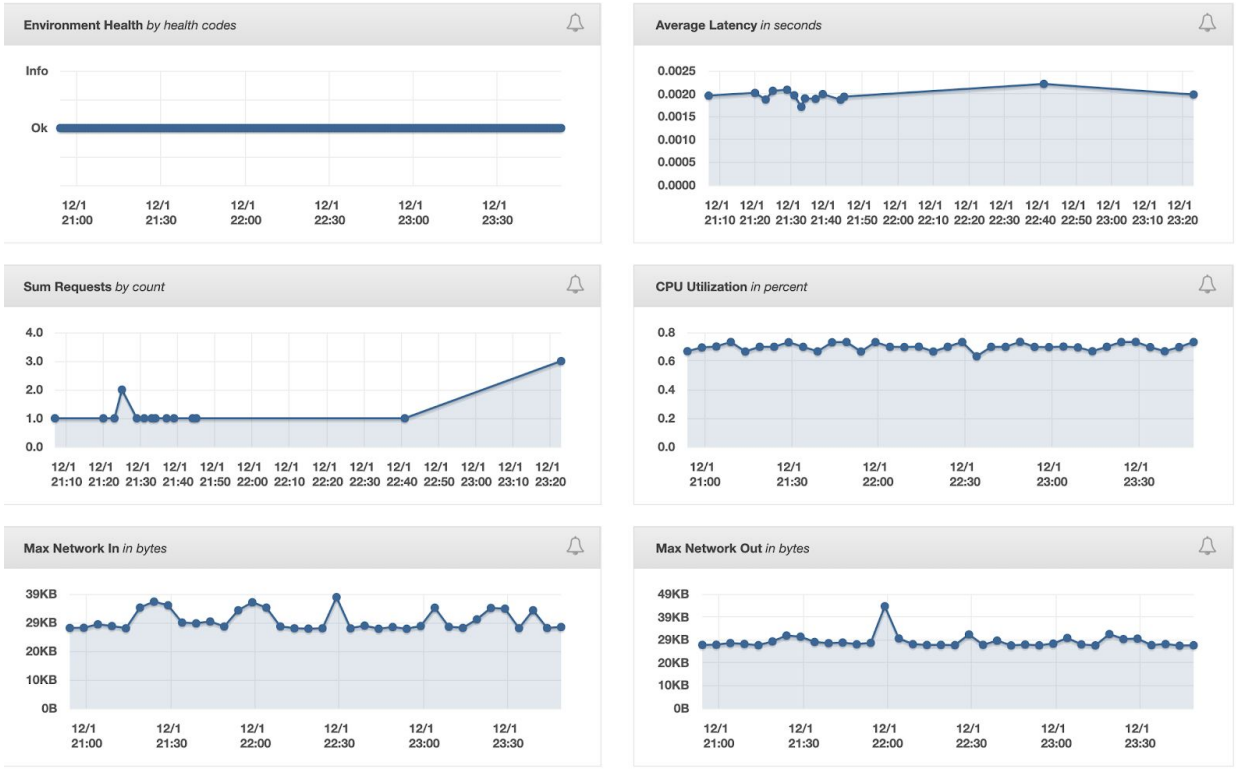
How does it scale with load?

The load balancer that I am currently using is using a specified port and protocol for my instances. AWS automatically distributes incoming traffic across multiple targets, such as Amazon EC2 instances. It can handle varying load of my application traffic in a single availability zone or across multiple availability zones.I am also able to add and remove compute resources

from your load balancer as my needs change, without disrupting the overall flow of requests to my applications. ELB can route to other healthy routes, if some become unavailable.

How to monitor?

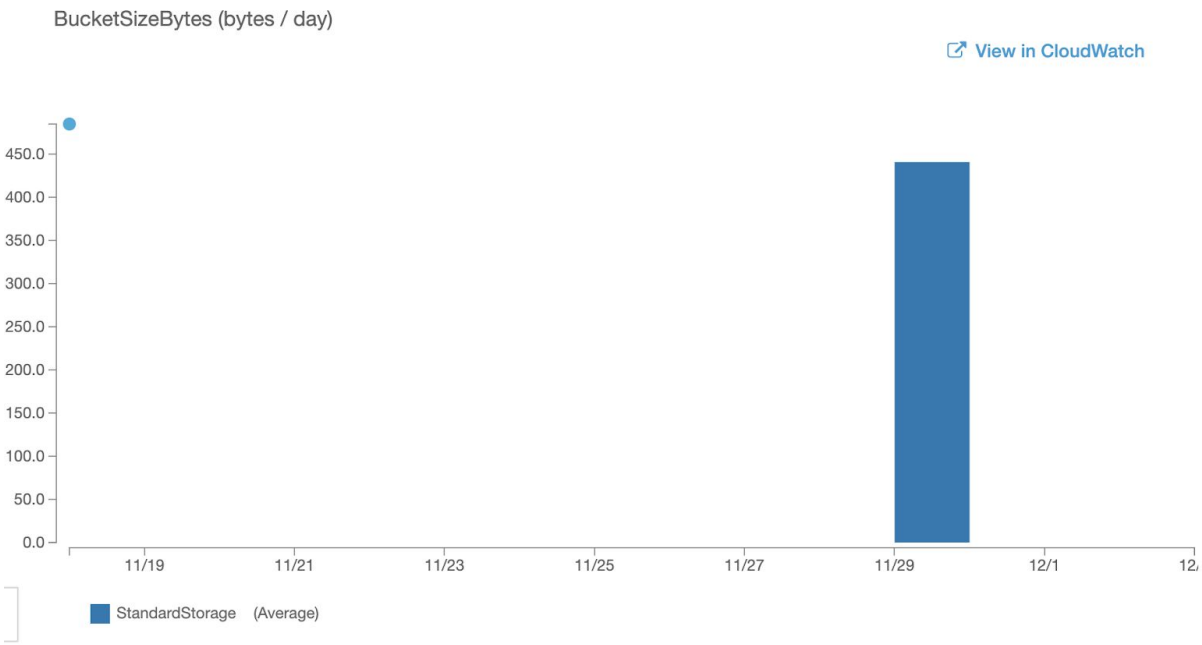
Using Elastic Beanstalk Metrics



Using DynamoDB metrics

CSS490 Cloud Computing
Chelsea Ip Tze Hwan
Program 4 Official Report

Using Amazon S3 metrics



Estimating SLA

DynamoDB

- Durability : 100%
- Availability :99.99%

EC2

- Durability : 99.999999999%
- Availability : 99.99%

Amazon S3

- Durability : 99.999999999%
- Availability : 99.95%

Overall

- Durability : 99.999999999 * 99.999999999 * 100= 99.9999998%
- Availability : 99.99*99.99*99.95 = 99.93%