

ECE1779 Assignment 2 Dynamic Resource Management

Zichuan Wang
1000474300

Zhaohui Qu
1005783127

Huan Qi
1006437214

1 User Documentation

1.1 Functionality

This web application we designed for manager allows for easy worker instances management. The login username and password for the manager app are both "admin". After logging in to the home page as shown in figure 1, the main functionality of this page is to list worker information including instance id, open port number, instance status, CPU utilization chart, http requests chart and total worker number chart. Link to load balanced user-app entry URL is provided as well.

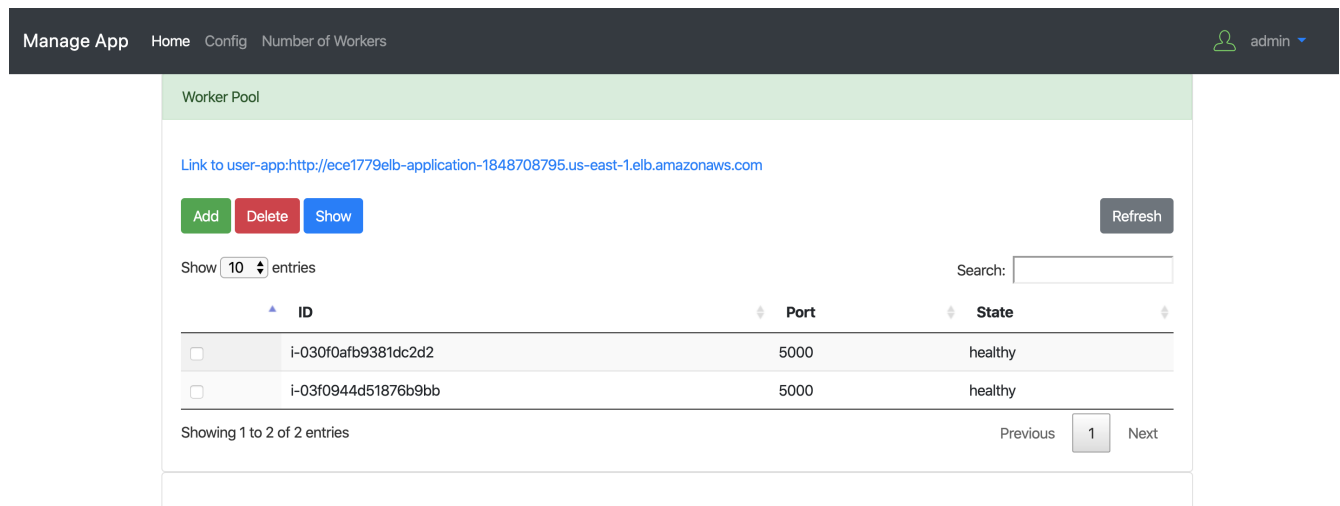


Figure 1: Home Page

The main functionality of the config page is to setup auto scaling policy parameters as shown in figure 7. CPU growing and shrinking thresholds are based on average percentage CPU utilization of the worker pool for the past 2 minutes. Ratio to expand is the multiplication factor to expand the worker pool with. Ratio to shrink is the percentage of the worker pool to be stopped. Function for clearing all S3 storage and RDS database entries is provided under section clear all application data. Function for stop manager and terminate workers is also provided under section stop manager and terminate workers.

A dedicated page is provided for displaying the number of workers for the past 2 hours as shown in figure 3.

Manager App
Home
Config
Number of Workers
admin

Auto scaling configure

Cpu threshold to grow30Modify

Cpu threshold to shrink15.0Modify

Ratio of expand2.0Modify

Ratio of shrink0.5Modify

Submit

Clear all application data

Clear

Stop Manager and Terminate Workers

Stop

Figure 2: Config Page

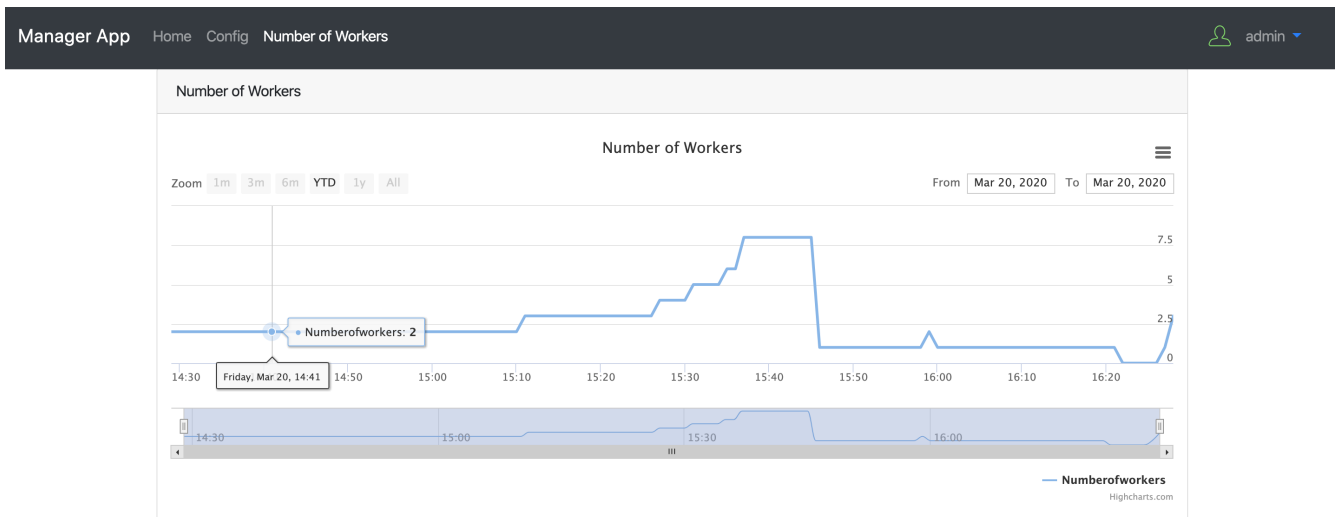


Figure 3: Number of Workers Page

1.2 How to Run

Start Manager Instance, navigate to desktop and run `./run_manage.sh`.

2 Developer's Documentation

As shown in the block diagram in figure 4, home page has three buttons. Add button would create and register one EC2 instance to the target group of our Elastic Load Balancer. Delete button

would suspend one EC2 instance for later use. Show button would display the CPU utilization chart fetched from Cloud Watch and http requests count chart fetched from RDS for selected instance.

Config page has three sections. Auto scaling section provides parameter adjustment for scaling threshold and scaling factor. Confirmed parameters would be saved on RDS for future access. Clear application data sections provides ability to clear all application data including S3 storage and RDS entires for http requests history, auto scaling parameter history, user information and image reference. Stop manager and all workers section provides ability to terminate all instances in target group of ELB and manager instance itself.

Number of workers page displays worker count variation over time within target group of ELB. Number of workers is determined by the health instances metric provided by cloud watch.

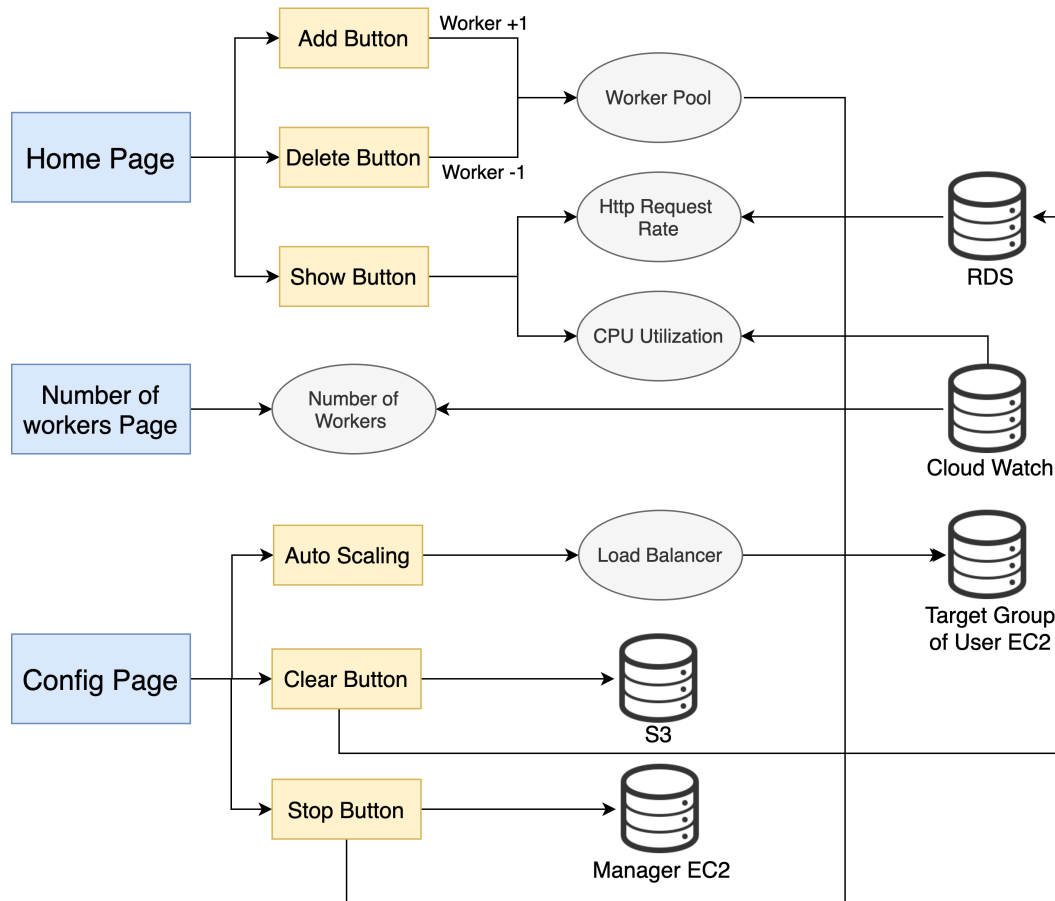


Figure 4: Block Diagram for Manager App

2.1 Project Structure

Project tree structure of manager app is shown below.

```

/website
├── __init__.py    #flask app and rds database initialization
└── /aws
    ├── __init__.py

```

```

├── auto_scaling.py    #auto scaling policy based on average CPU utilization
├── aws.py             #aws client handling class
├── Userdata.txt       #aws UserData for user app initialization at launch
├── config.py          #flask app global configuration
├── configure.py       #manager app configuration routes
├── error.py           #manager app error routes
├── forms.py           #auto scaling config form and manager login form
├── home.py            #manager app home routes displaying user instances info
├── login.py           #manager app login routes
├── models.py          #aws rds database models
├── /static
│   ├── /css           #cascading style sheets for manager app
│   │   ├── error.css
│   │   ├── home.css
│   │   └── login.css
│   ├── /js            #JQuery front-end asynchronous request handling
│   │   ├── home.js
│   │   ├── configure.js
│   │   └── Numberofworkers.js
├── /templates         #html templates for manager app
│   ├── base.html      #base layout to inherit
│   ├── configure.html  #manager app configure page
│   ├── error.html     #manager app error page
│   ├── login.html     #manager app login page
│   └── numberofworkers.html #manager app worker number plot page

```

3 Database

3.1 Database Setup

Database is setup using RDS for storing user information, storage reference in S3, auto scaling configuration data, http requests history. The MySQL code for initialization is shown below.

```

create schema if not exists testtable;

use testtable;

drop table if exists testtable;
create table if not exists testtable (
    Email varchar(100) not null unique,
    Username varchar(100) not null unique,
    Password varchar(300) not null,
    primary key (Username)
);

drop table if exists photos;

```

```

create table if not exists photos(
  Username varchar(100) not null,
  PhotoURL varchar(300) not null,
  primary key (PhotoURL)
);

drop table if exists autoscalingconfig;
create table if not exists autoscalingconfig (
  ascid bigint(32) not null auto_increment,
  cpu_grow float not null,
  cpu_shrink float not null,
  ratio_expand float not null,
  ratio_shrink float not null,
  timestamp datetime not null,
  primary key (ascid)
)AUTO_INCREMENT=200 DEFAULT CHARSET=utf8;

drop table if exists requestperminute;
create table if not exists requestperminute (
  requestid bigint(32) not null auto_increment,
  instance_id varchar(50) not null,
  timestamp DATETIME not null,
  primary key (requestid)
)DEFAULT CHARSET=utf8;

```

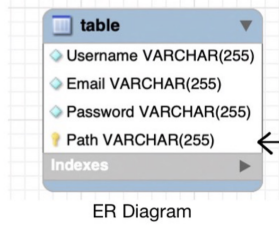
3.2 Database Schema

Database schema before and after normalization are shown in figure [5a](#) and figure [5b](#) respectively. Before normalization, user information is duplicated for each image under that specific user. After normalization, unique user id is used to represent each user in order to identify the owner of each image.

4 Results

As shown in figure [6](#), when first start the manager app, one worker instance will be created automatically. The configuration for auto scaling is shown in figure [7](#).

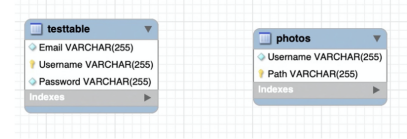
Then we start to use load generate to upload a 8KB picture with an uploading rate of 0.5 per second. After the average CPU utilization surpasses 40%, a new instance was created. The load was shifted to the newly created instance as shown in figure [8](#). Re-distribution of HTTP requests is shown in figure [9](#)



Primary key

Username	Email	Password	Path
qihuan1	qihuan1@q.com	\$5\$rounds=535000\$qCrXMH7NiSe187Tq\$W	1581303411.799204Screenshot_1580614928.png
qihuan2	qihuan2@q.com	\$5\$rounds=535000\$mmgqwfKvgfKJbcX\$R	1581303646.692148屏幕快照 2020-01-23 下午5.20.56.png
qihuan5	qihuan5@q.com	\$5\$rounds=535000\$FW5UyXvMjGgl8Egy\$ix	1581366438.765706屏幕快照 2020-02-04 下午8.59.15.png
qihuan	qihuan@q.com	\$5\$rounds=535000\$SIvOKDyLNVA8rPvq\$sv	1581366461.352361屏幕快照 2020-02-07 下午2.56.24.png
qihuan6	qihuan6@q.com	\$5\$rounds=535000\$mbI8zxVCqB0BSswh\$B	1581370585.7371611.png
qihuan6	qihuan6@q.com	\$5\$rounds=535000\$mbI8zxVCqB0BSswh\$B	1581370672.8697871.png
qihuan6	qihuan6@q.com	\$5\$rounds=535000\$mbI8zxVCqB0BSswh\$B	1581370681.7087442.png
qihuan6	qihuan6@q.com	\$5\$rounds=535000\$mbI8zxVCqB0BSswh\$B	1581370786.2224932.png
qihuan6	qihuan6@q.com	\$5\$rounds=535000\$mbI8zxVCqB0BSswh\$B	1581371100.8579412.png
huanqi	huanqi@q.com	\$5\$rounds=535000\$AVX.OvX2zqEGjwnK\$et	1581385318.222844_3.png
huanqi	huanqi@q.com	\$5\$rounds=535000\$AVX.OvX2zqEGjwnK\$et	1581388221.775812_raptors.jpg
huanqi	huanqi@q.com	\$5\$rounds=535000\$AVX.OvX2zqEGjwnK\$et	1581388645.351313_raptors.jpg
qihuan5	qihuan5@q.com	\$5\$rounds=535000\$FW5UyXvMjGgl8Egy\$ix	1581388874.338753_street1.jpg

(a) Before Normalization



Primary key

Email	Username	Password
huanqi@q.com	huanqi	\$5\$rounds=535000\$AVX.OvX2;
qihuan2@q.com	qihuan	\$5\$rounds=535000\$SIvOKDyLI
qihuan1@q.com	qihuan1	\$5\$rounds=535000\$qCrXMH7?
qihuan2@q.com	qihuan2	\$5\$rounds=535000\$mmgqwfK?
qihuan5@q.com	qihuan5	\$5\$rounds=535000\$FW5UyXv?
qihuan6@q.com	qihuan6	\$5\$rounds=535000\$mbI8zxVC;

Primary Key

Username	Path
huanqi	1581385318.222844_3.png
huanqi	1581388221.775812_raptors.jpg
huanqi	1581388645.351313_raptors.jpg
qihuan	1581366461.352361屏幕快照 2020-02-07 下午2.56.24.png
qihuan1	1581303411.799204Screenshot_1580614928.png
qihuan2	1581303646.692148屏幕快照 2020-01-23 下午5.20.56.png
qihuan5	1581366438.765706屏幕快照 2020-02-04 下午8.59.15.png
qihuan6	1581366874.338753_street1.jpg
qihuan6	1581370585.7371611.png
qihuan6	1581370672.8697871.png
qihuan6	1581370681.7087442.png
qihuan6	1581370786.2224932.png
qihuan6	1581371100.8579412.png

(b) After Normalization

Figure 5: Database Schema

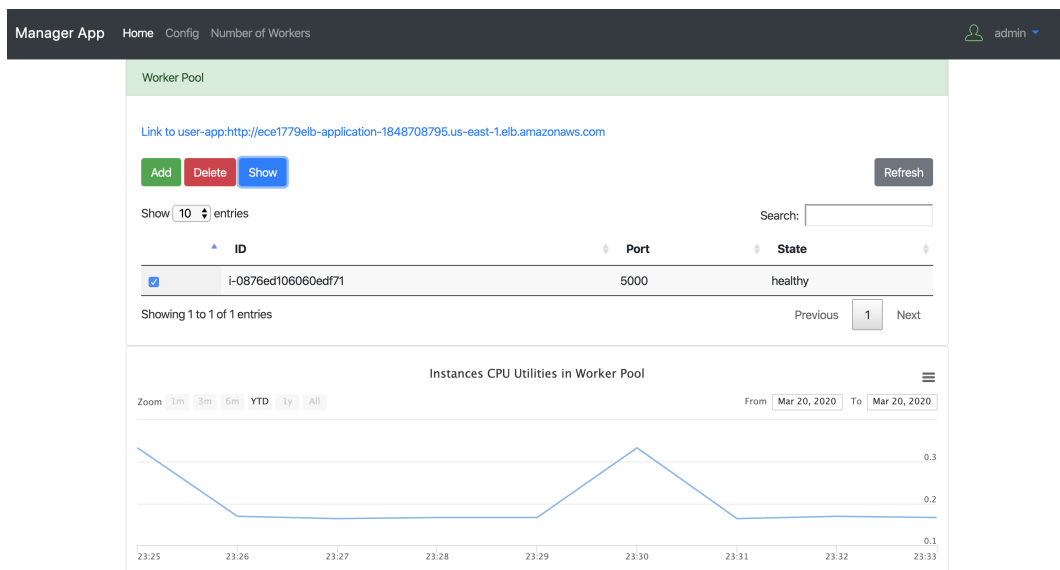


Figure 6: Start

Manager App
Home
Config
Number of Workers
admin

Auto scaling configure

CPU threshold to grow
40
Modify

CPU threshold to shrink
5.0
Modify

Ratio of expand
2.0
Modify

Ratio of shrink
0.5
Modify

Submit

Clear all application data

Clear

Stop Manager and Terminate Workers

Stop

Figure 7: Auto Scaling Configuration

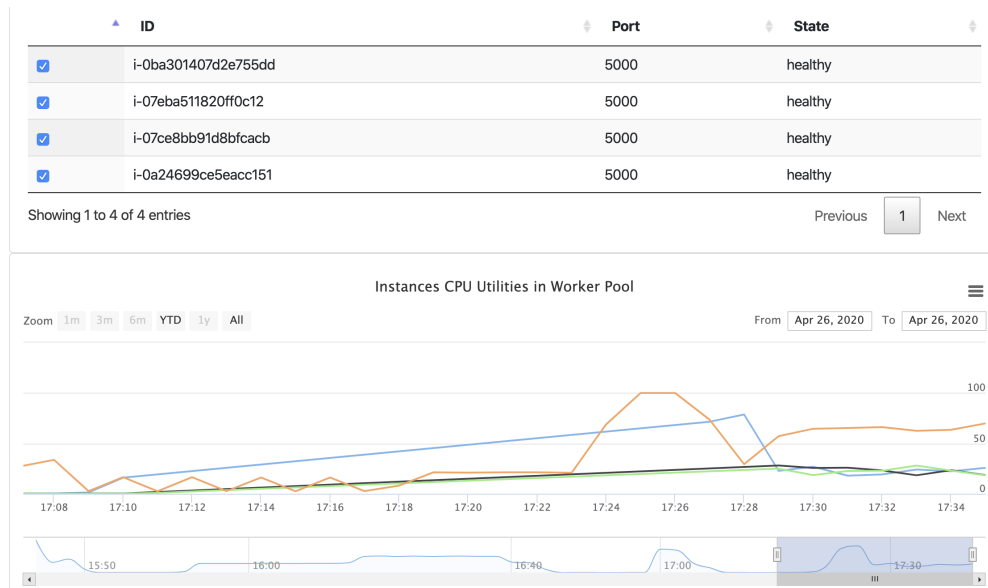


Figure 8: CPU utilization

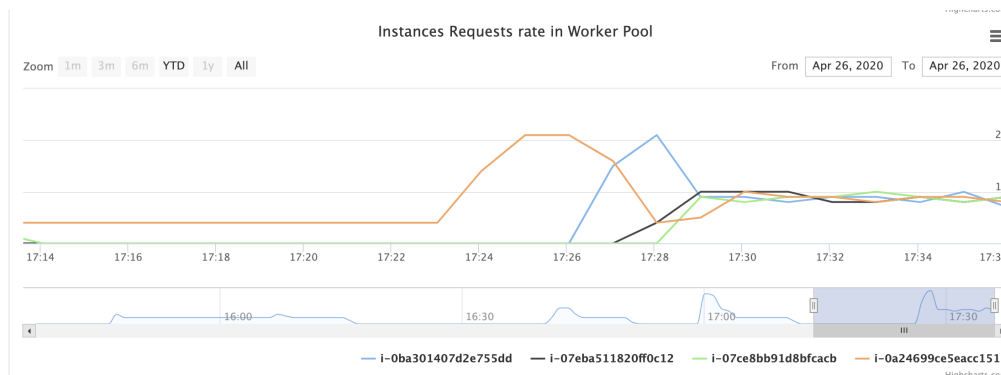


Figure 9: HTTP requests