

ECE 1779 Assignment 2

Group member

Qin Deng 1006714799 Ziyang Zhao 1007212927 Chenglei Xie 1006118755

Qin Deng is responsible for backend change of userapp and part of the manager app backend.

Ziyang Zhao is responsible for auto-scaler and part of the manager app backend.

Chenglei Xie is responsible for the front-end of user app and manager app.

The test of our application is completed by all of the group members.

How to Start

1. The aws account email is davyq980731@gmail.com
2. Start the EC2 instance with ID: **i-0b2311565889b5d39** and Name: **manager-ece1779_a2**
3. Go to service RDS. Make sure that the database called **database-3** is running.
4. Extract the keypair.pem from a2_submit.rar
5. Put keypair.pem under your ~/.ssh directory. If “~/.ssh” doesn’t exist, mkdir ~/.ssh
6. Once pem file is in your .ssh directory, check the public IPv4 address in your aws console.
7. Run the following command to ssh into the EC2 instance.
ssh -i ~/.ssh/keypair.pem ubuntu@ec2_ip_address
8. Once you ssh into the EC2 instance. We need to cd into code repository directory:
cd Desktop
9. Run the following command in your command line.
bash start.sh
10. The website should be running like the following figure. The website is running on port 5000, so you can access the website by url: 3.90.1562.210(your public IPv4 address):5000



```
ubuntu@ip-172-31-74-213: ~/Desktop
ubuntu@ip-172-31-74-213:~$ cd Desktop
ubuntu@ip-172-31-74-213:~/Desktop$ bash start.sh
Start running the application
Activating the environment
Running program
```

Functionality

This application is designed to manager worker instances in easy steps.

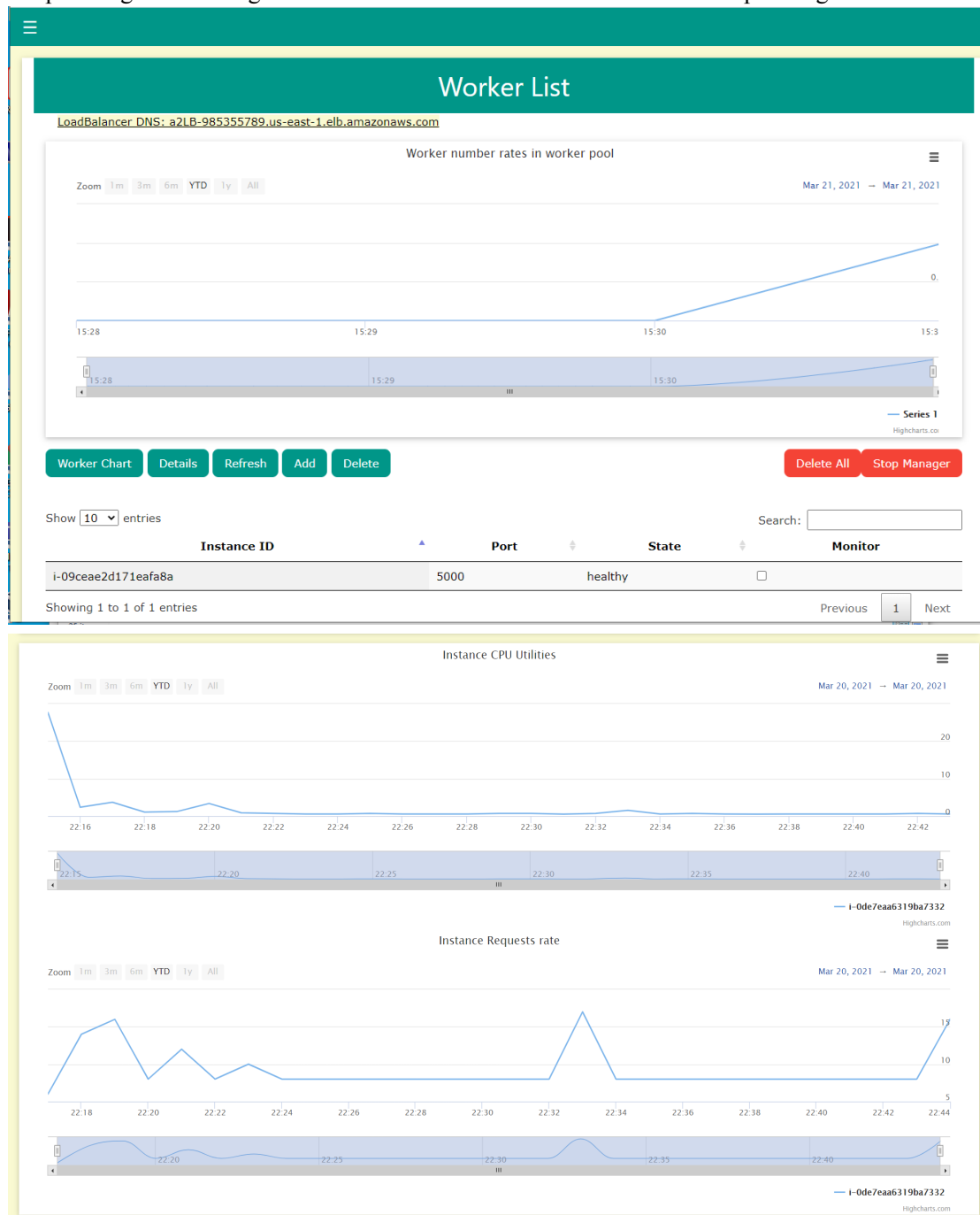
In the home page, a table shows information about all worker instances including instance id, open

port and instance status, by clicking the checkbox and 'Details' button CPU utilization chart and http requests chart for the specific instance will be present on the page. A link to load balanced user-app entry URL is provided on left top as well.

By clicking the 'worker chart' button, a chart contains the number of healthy workers for the past 30 minutes will show up above the buttons.

The 'Add' and 'Shrink' button can add or terminate a work instance manually.

The red button 'Delete All' will delete all the application data in S3 and RDS, and the red button 'Stop Manager' on the right will terminate all the worker instances and stop manager instance.



The bar on left side can guide to auto-scaling configure page. The config page is to setup auto scaling policy parameters as shown in the figure below. CPU growing and shrinking thresholds are based on average CPU utilization percentage (from 1 to 100) 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.

User need to first click 'Configure' and then modify the number. When finish modifying, user can click 'submit' button to renew the parameters.

Auto scaling configure

Cpu threshold for growing

60

Configure

Cpu threshold for shrinking

3.0

Configure

Expand ratio

2.0

Configure

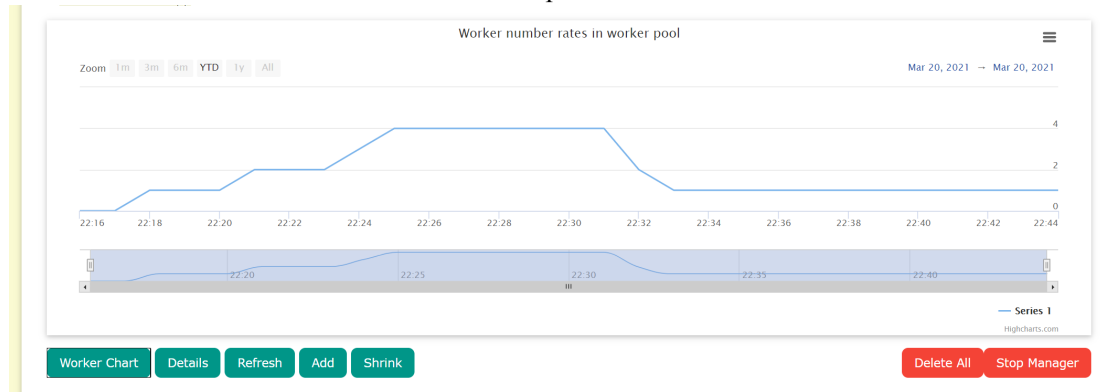
Shrink Ratio

0.5

Configure

Submit

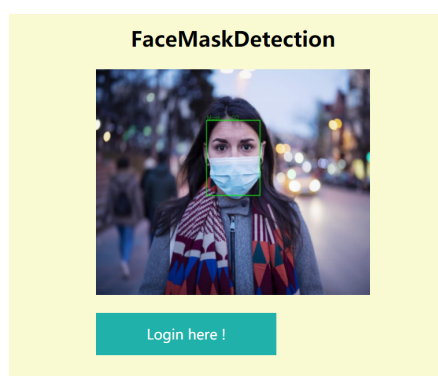
The result of auto-scaling can be seen in the home page, by clicking the 'Worker Chart' button, which shows the number of worker instances in past 30 minutes.



User app:

A link to load balanced user-app entry URL is provided on left top called LoadBalancer DNS. Click that link and you will be directed to user-app. The functionality of the user app is the same as assignment1, but we add UI this time.

After you click the LoadBalancer DNS, you will go to the following page. Click the button login here to go to login page



On login page, input username and password to login. The admin account has Username: admin and password: test.

Login Page

Login

Username

admin

Password

••••

Login

Forgot Your Password? [Click to Reset It](#)

After login page, 'User Menu' page allow users to upload picture and view history.

Users can upload image by url or upload a local picture. The result will be shown in history page.

User Menu

Welcome

username: admin

is_admin: True

[Add a user](#)

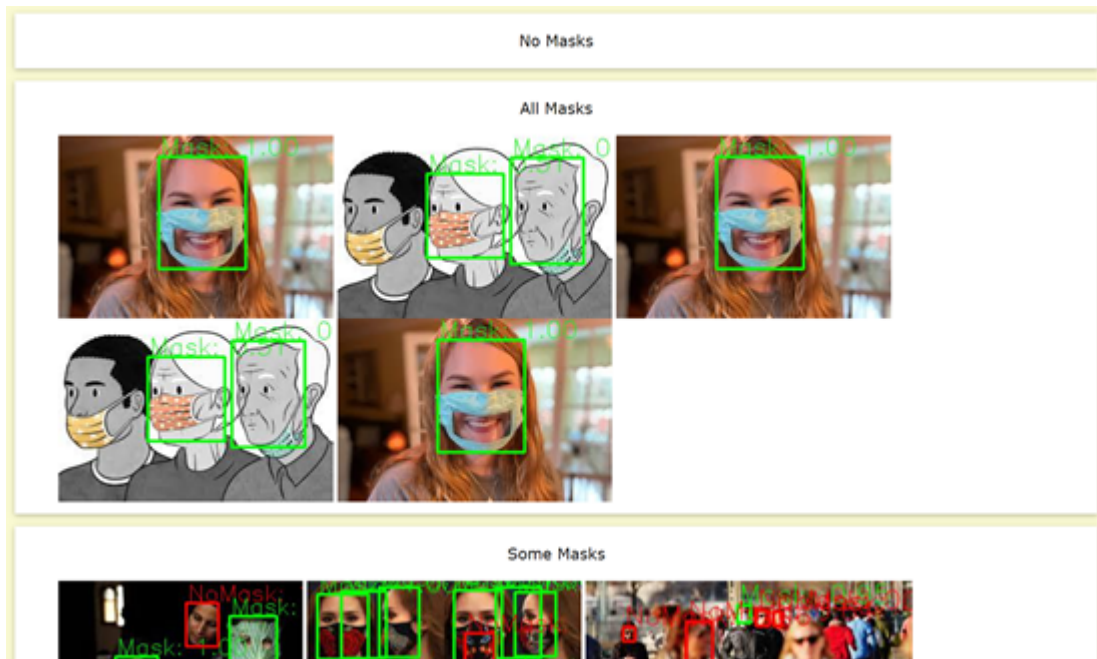
[Delete a user](#)

[Upload an image](#)

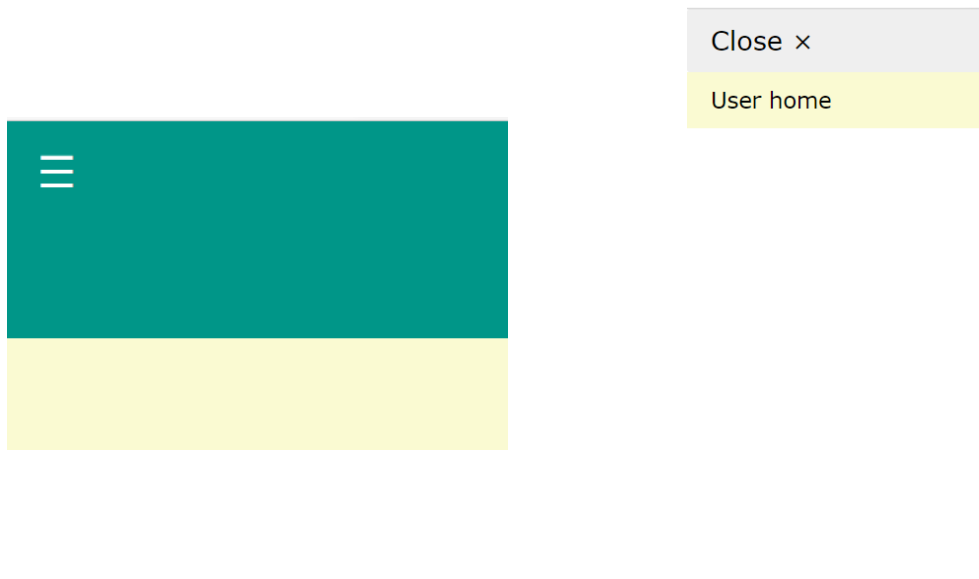
[Upload history](#)

[Change Password](#)

[Log Out](#)

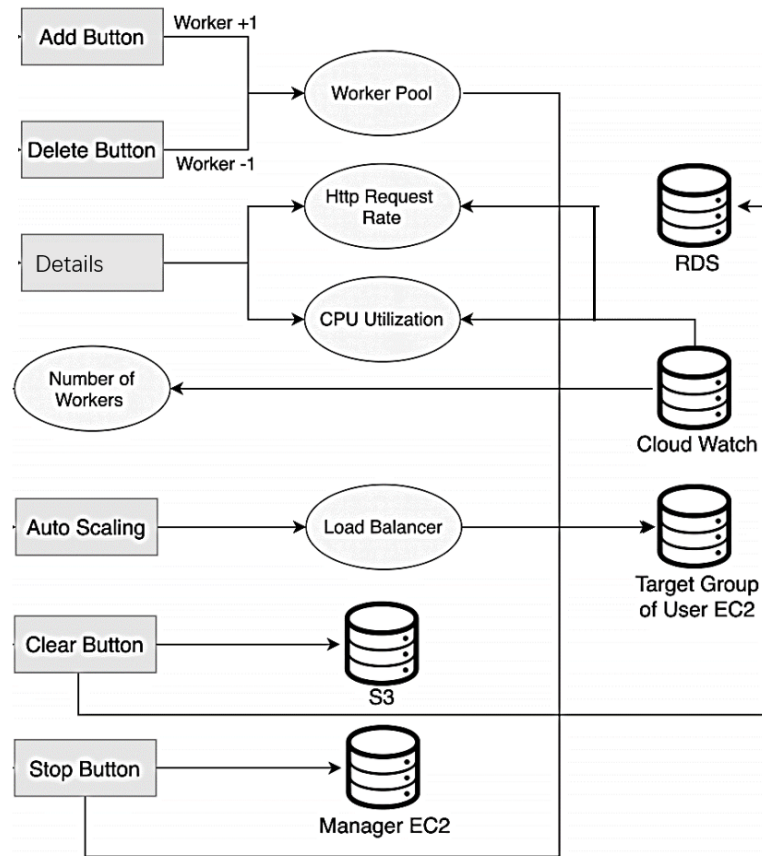


On the right top of upload, history, add_user and delete_user page, there is a button. Click that button, you will get back to home page button.



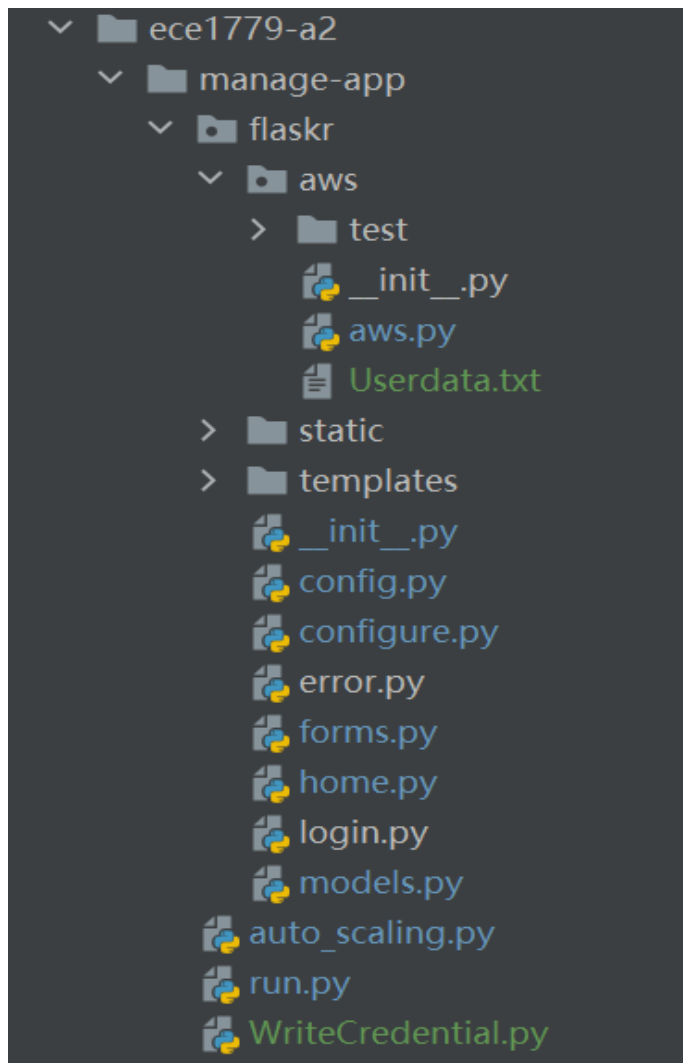
System architecture

The function of every block in manager app works as below.



Code Structure

Tree structure of manager app is shown below.



aws.py: AWS client handling class

Userdata.txt: EC2 userdata use to initialize worker instance

Static folder: contains css, js files

Templates folder: contains html files

__init__.py: flask app, RDS database and manager app initialization

config.py: flask app global configuration, like rds url

configure.py: write manager app auto-scaling parameter

error.py: manager app error routes

forms.py: auto-scaling config form

home.py: home page for presenting worker instances information and clear all data

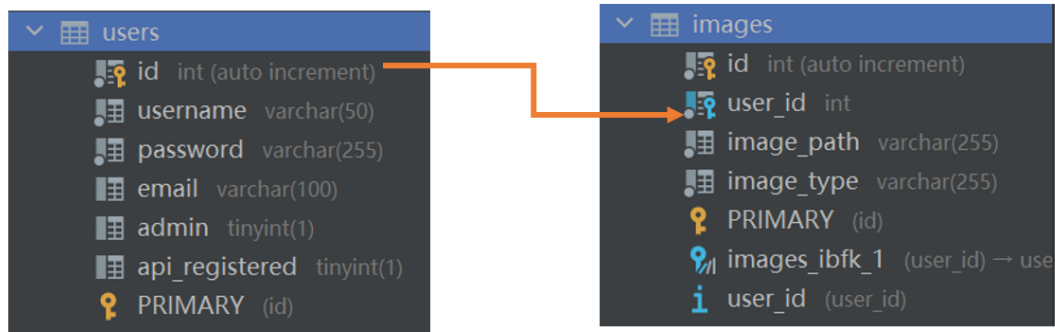
models.py: AWS RDS database models

auto_scaling.py: auto-scaling policy

run.py: start the app

Database

User table and image table is the same as the assignment1.



Users table contain 6 columns, 'id', 'username', 'password', 'email', 'admin'(if the user is administrator this column will be 1 else 0), 'api_registered'(if the user was created by api this will be 1 else 0)

Images table contain 4 columns, 'id', 'user_id'(the foreign key to the user table), 'image_path'(the image url), 'image_type'(no_faces, no_masks, some_masks, all_masks)

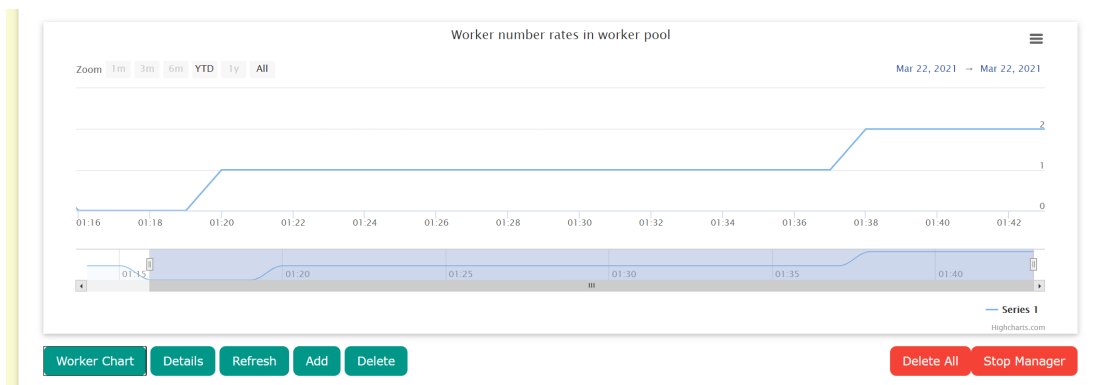
Autoscalingconfig table contains auto-scaling parameters, and will be sorted by time order.

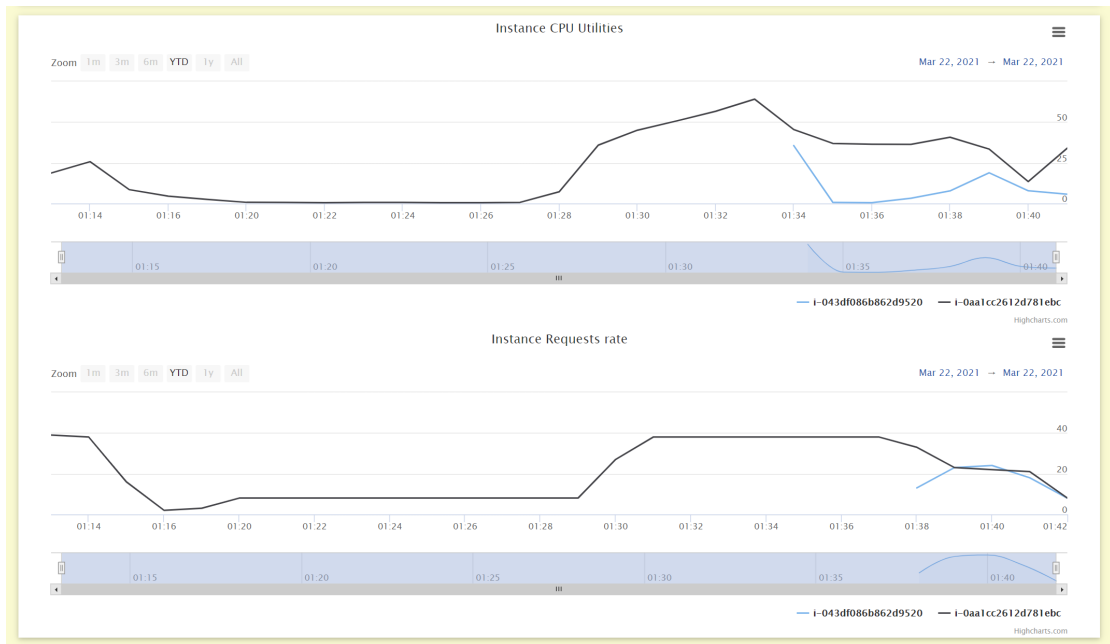
autoscalingconfig	
ascid	bigint (auto increment)
cpu_grow	float
cpu_shrink	float
ratio_expand	float
ratio_shrink	float
timestamp	datetime
PRIMARY	(ascid)

Result

When the average CPU utility is higher than the threshold for growing (here we set threshold for growing as 50%, and expand ratio as 2), the number of workers in work pool will grow by expand ratio multiple the original number of workers in work pool.

The result is shown as following. The healthy worker number line grows a little bit later than average CPU utility exceed 50% is because starting up instances requires certain amount of time.





When the CPU utility is lower than the threshold for shrinking (here we set shrinking threshold as 10%, and the shrink ratio as 0.5), the work pool will shrink by shrink ratio.

