Documentation

1.0 Web Application Introduction

Welcome to the documentation of the AWS Management application. This manager app allows the manager to control the Cloud Server for object detection applications. This manager app enables users to:

- **Listing workers**: users can see:
  - A list of workers in real time and browse the details of these workers.
  - Total utilization of workers for the past 30 minutes with the resolution of 1 minute.
  - The rate of HTTP requests received by the worker in each minute for the past 30 minutes.
  - The chart of number of workers for the past 30 minutes.
- **Link to user-app**: link to user-app via the link to load balanced user-app entry URL.
- **Manually changing worker pool size**: manually grow the worker pool size by 1 and shrink the worker pool size by 1 by two buttons.
- **Auto-scaling**: users will enjoy the service of autoscaling by setting the following parameters (CPU threshold for growing, CPU threshold for shrinking, Ratio to expand the worker pool, Ratio to shrink the worker pool), this manager app will automatically resize the worker pool based on the load.
- **Stop the manager:** terminates all the workers and stops the manager itself.
- **Deleting application data**: delete application data stored on the RDS and images stored on S3.

2.0 Agenda of the documentation

The structure of the documentation is basically in three parts: users guide, developers' reference and results.

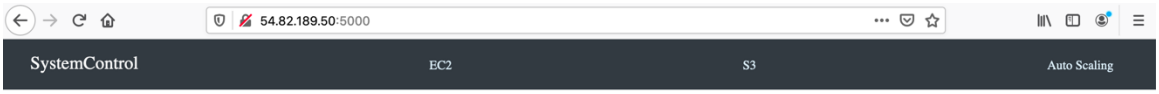
- **Users guide:** instruction for users to use the manager app with its functionality of resizing the worker pool on demand.
- **Developers reference:** instruction for developers to understand the architecture of the application.
- **Results:** Test cases and scenarios that demonstrate the functionality of auto-scaler component works.

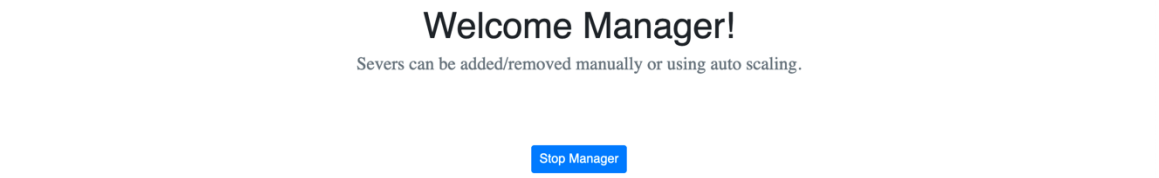
3.0 Users guide

The guide will give the user a tutorial on how to use this application.

3.1 Main page

Open the browser and enter the dynamic URL Server's IPv4 address + :5000 to access the main page. The upside navigation menu contains **EC2**, **S3,** and **Auto-Scaling**, each can be visited by clicking its link. EC2 button can redirect to the instance page; the S3 button can redirect to the S3 and database clear page. Auto Scaling button can redirect to the Auto Scaling page to control the autoscaling process. The Stop Manager in the middle can terminate all the workers and stops the manager itself.

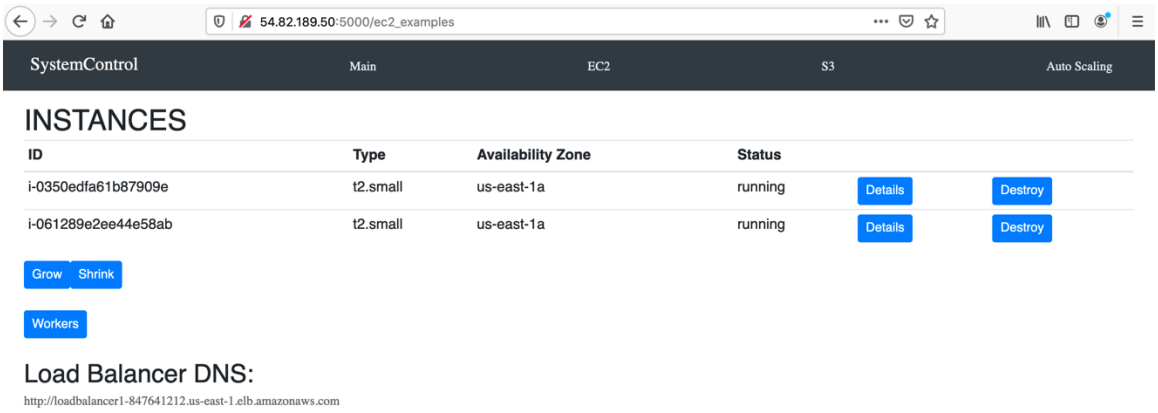




Main Page

3.2 EC2 page

Click the link in the navigation menu to access the EC2 page. EC2 page lists information of all the instances, including **ID**, **Type**, **Availability Zone,** and **Status**. Each instance comes with two buttons: **Details** and **Destroy**. If you click on the Details button, you will link to the EC2 view page of this specific instance (introduce later in the documentation). If you click on the Destroy button, you will terminate this instance. In the left bottom of the instance list, there are three buttons: **Grow**, **Shrink,** and **Workers**. The first two buttons are for users to manually change the worker pool size by 1. If you click on the Grow button, the worker pool will grow by 1. If you click on the Shrink button, the worker pool will shrink by 1. If you click on the button Workers, you can see the chart of the number of workers for the past 30 minutes. Below the Workers button, there is a link named **Load Balancer DNS** link to load balanced user-app. You can redirect to user-app by click on this link.

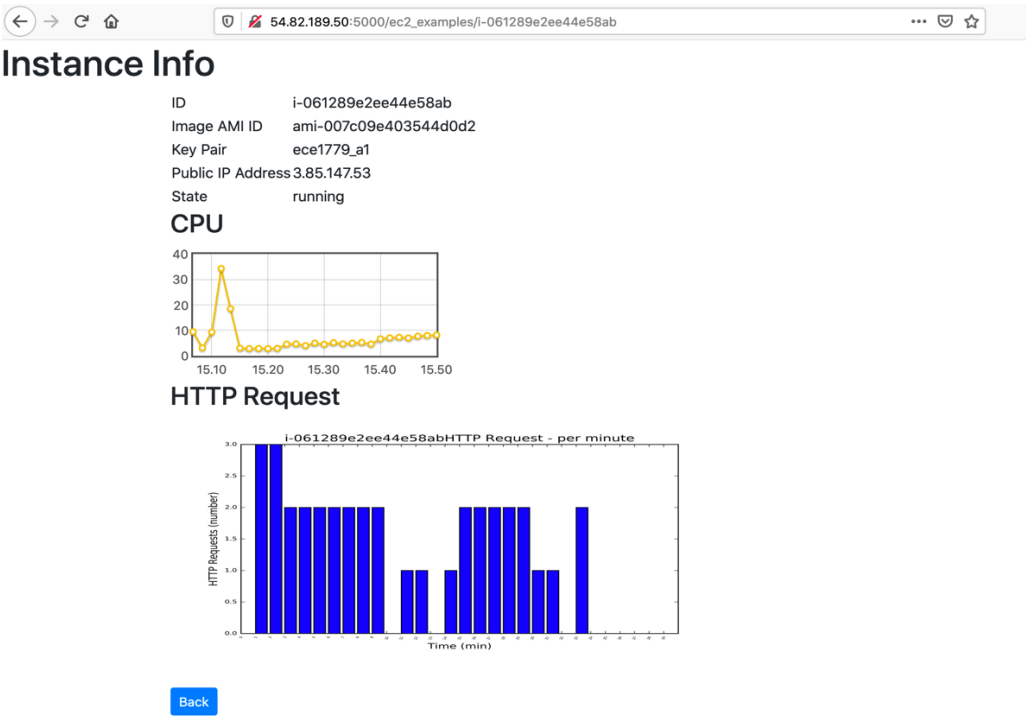


EC2 Page

3.3 View page

Click the link in the navigation menu to access the EC2 view page. EC2 view page is basically made of two charts. The **basic information** (ID, Image AMI ID, Key Pair, Public IP Address and State) of the specific instance is on the up of the page.

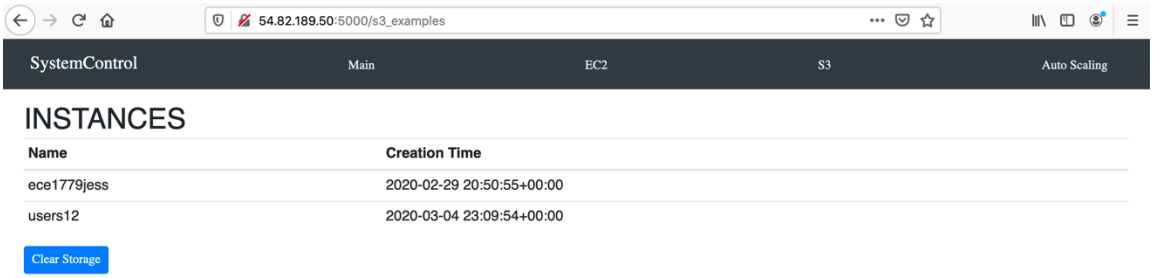
- The **CPU utilization chart:** showing the total CPU utilization of the worker of the past 30 minutes.
- The **HTTP request chart:** showing the rate of HTTP requests received by the worker in each minute for the past 30 minutes.



View Page

3.4 S3 page

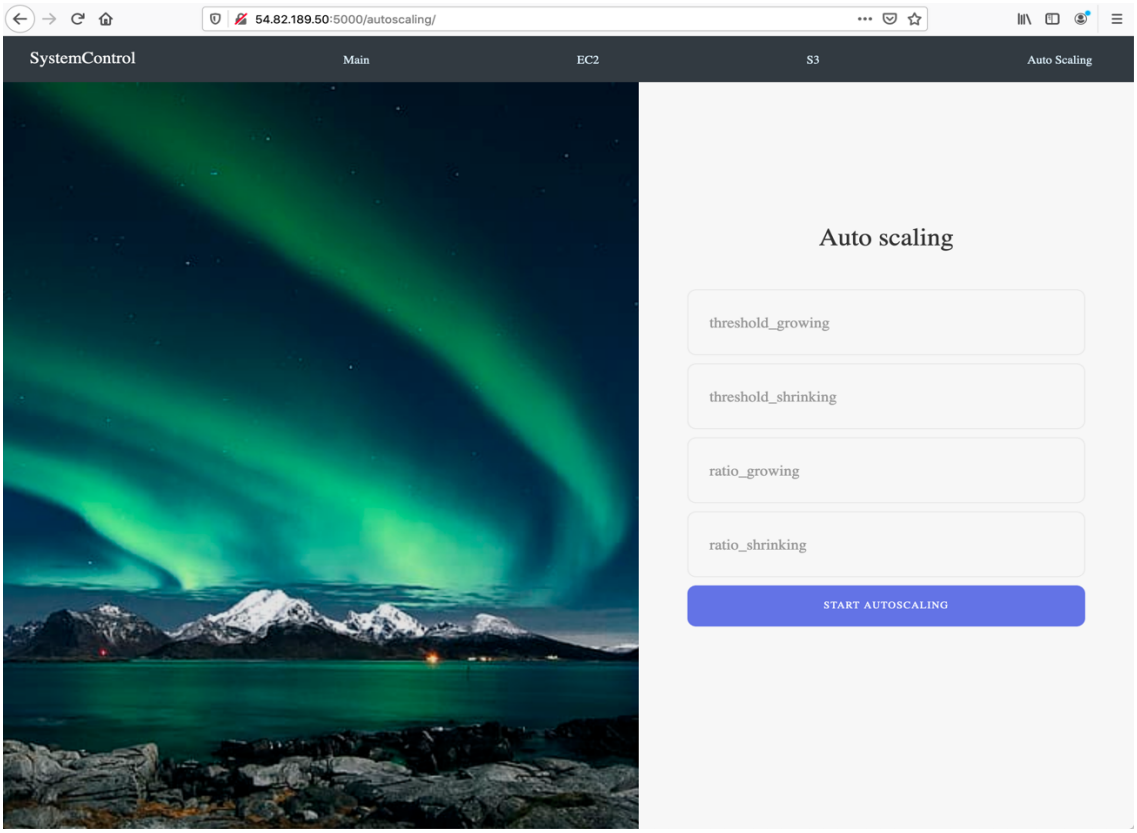
Click the link in the navigation menu to access the S3 page. S3 page lists the basic information (Name and Creation time) of instances in the S3 bucket. **Clear Storage** button is below the information chart, this button is used to delete all application data. If you click on the Clear Storage button, application data stored on the RDS and images stored on S3 will be deleted.



S3 Page

3.5 Auto-scaling

Click the link in the navigation menu to access the Auto Scaling page. This page allows managers to control the number of works automatically. Threshold_growing, threshold_shrinking, ratio_growing, and ratio_shrinking can be adjusted by its form. Click Start Auto Scaling to install all your settings. Then our auto-scaling service will apply the parameters provided by the manager to resize the worker pool automatically.



Auto-scaling Page

4.0 Developer's Reference

4.1 Programming environment

Language: Python 3.7
Framework: Flask
IDE: Pycharm
Database: Sqlalchemy

4.2.1 Developer program architecture

```
app                                  # Directory for the whole developer
program
├ services
│ ├ Autoscaling.py                   # File for autoscalling functionality
│ ├ EC2.py                           # EC2, ELB,S3 services function
│ ├ model.py                         # Autoscaling model
│ └ Manager_app                       # Manager Functionality
│
├ templates                          # Webpage file directory, for Manager UI
│ ├ auto_scaling.html
│ ├ base.html
│ ├ EC2_example.html
│ ├ EC2_view.html
│ ├ main_page.html
│ ├ S3_example
│ └ numofwokers.html
```

4.2.2 Worker program architecture

```
user                                   # Directory for the whole worker program
├ static
│
├ templates                                        # Webpage file directory
│ ├ detail_page.html
│ ├ login.html
│ ├ upload.html
│ ├ register.html
│ ├ upload_page.html
│ └ user_page.html
│
├ venv                                        # Virtual environment directory
├ app.py                                                  # Main function
```

```
├ model.py                                          # Create class for app.py
├ object_detection.py                              # The function for the object
detection
├ savephoto.py           # Functions for the operation of the table user_photo
├ thumbnail.py                                      # Functions to create the
thumbnail
├ create_database.py                                          # Init database
├ coco.names
├ yolo.py
├ yolov3.cfg
├ yolov3.weights
```

4.3 Main program

4.3.1 Open Module

flask_sqlalchemy	# Connector that connects the flask and the Mysql
flask	# a micro web framework based on the Werkzeug and Jinja2
os	# Operator that is used to obtain operating addresses in the system
jinja2	# Operator that writes the server's output to the web page
boto3	# Operator that connect to aws service
matplotlib	# Operator that plot the http request graph

4.3.2 Customized Class

EC2	# Define the interfaces for aws EC2 operations such as creating instances, deleting instances and describing instances
Autoscaling_Services	# Define all the implementation of the autoscaling
Autoscaling	# Define a class to get the user input autoscaling policy and save the policy in the database

4.3.3 Customized Function

grow_one_worker()	# Add a new instance to the worker pool. If has stopted instance, then restart one. Otherwise, create a new instance.
shrink_one_worker()	# Stop one running instance in the worker pool.
stop_manager()	# stop the manager instance and terminate all the worker instances.
delete_app_data_rds()	# Delete all the database in the RDS
get_available_target()	# Get all the instances registed in the target group, except the draining instances
get_running_instances	# Get all the instances that are currently running

get_cpu_utility()	# Calculate the current CPU utilization, number of instances that has CPU utility, and the time
auto_scaling()	# Autoscaling functionality implementation, grow or shrink number of workers when CPU utilization reach the threshold
grow_worker_by_ratio()	# Grow the number of running instances by a given ratio
shrink_worker_by_ratio()	# Shrink the number of running instances by a given ratio

4.3.4 Templates

auto_scaling	# Display the auto-scaling page
base	# A helper HTML template to plot chart
EC2_example	# Display all the instances
EC2_view	# Display two charts: CPU utilization and HTTP request chart
main_page	# Display the main page
Numberofworkers	# Frontend that plot the chart of number of workers for the past 30 minutes
S3_example	# Frontend to display the S3 buckets

4.4 Database

Service name: ece1779db@ece1779db.cj2g85prhcmw.us-east-1.rds.amazonaws.com

RDS: Database is stored in Amazon RDS. We built two different databases (1779db and 1779dbA2) for user-app and manager-app.

4.4.1 Database name:1779db (The database for user-app)

Table name: users

Column Name	Data type	Characteristic	Default
id	int(11)	Primary key, Not Null, Auto increment	NULL
name	varchar(255)	Not Null	NULL
email	varchar(255)	Not Null	NULL
password	varchar(255)	Not Null	NULL
salt	varchar(255)	Not Null	NULL
photo_id	varchar(255)	Not Null	NULL

Table name: user_photo

Column Name	Data type	Characteristic	Default
id	int(11)	Primary key, Not Null, Auto increment	NULL

name	varchar(255)	Not Null	NULL
org_photo	varchar(255)	Not Null	NULL
thumbnail	varchar(255)	Not Null	NULL
det_photo	varchar(255)	Not Null	NULL

4.4.2 **Database name:1779dbA2** (The database for manager-app)

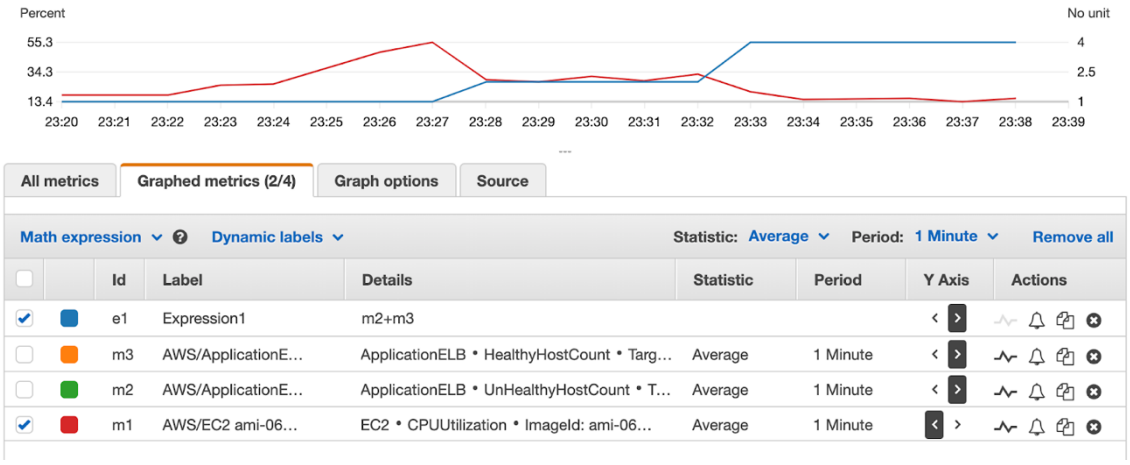
Table name: autoscaling

Column Name	Data type	Characteristic	Default
id	int(11)	Primary key, Not Null, Auto increment	NULL
threshold_growing	float	Not Null	NULL
threshold_shrinking	float	Not Null	NULL
ratio_growing	float	Not Null	NULL
ratio_shrinking	float	Not Null	NULL

5.0 Results

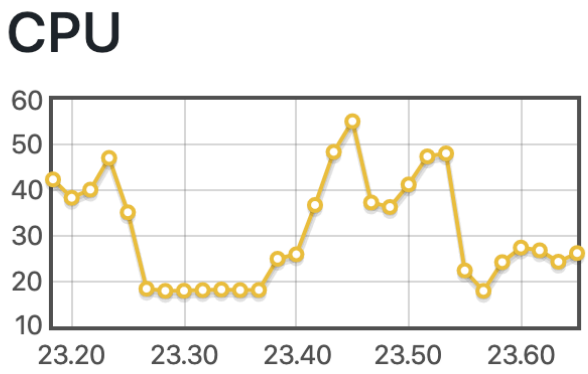
5.1 Test case 1: CPU utilization rising up.

We use the load generator to upload 200 images at a speed of 5s per image. The policy are as follows: threshold_growing: 30.0, threshold_shrinking: 13.0, ratio growing: 2.0, ratio shrinking: 0.5.



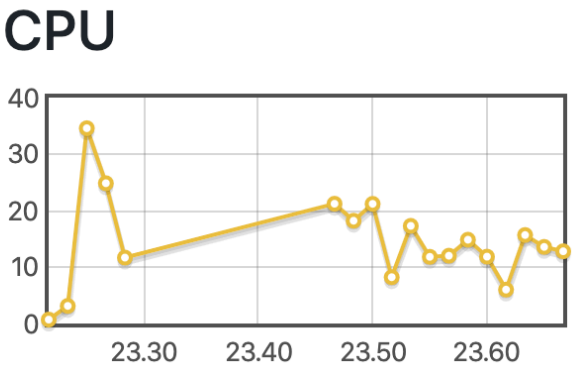
AWS CloudWatch plot

- **Stage 1:** 1 running instance (i-0977f8d59d1b0abc2), CPU utilization grows.



CPU utility plot for instance i-0977f8d59d1b0abc2

- **Stage 2:** At around 23:41 in the CPU plot (23:24 in the AWS CloudWatch plot), CPU utilization grows to exceed the growing threshold, one more instance needed. Then instance i-0fc6f2fbbaf844d03 created. We can observe that the CPU utilization drops after the new instance running. From the AWS CloudWatch plot, we can find that the CPU utilization and the number of instances are stable and converge for 4 minutes.

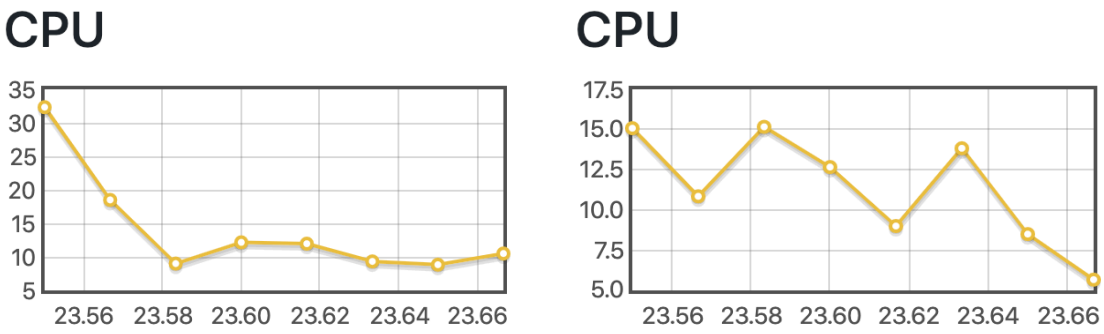


CPU utility plot for instance i-0fc6f2fbbaf844d03

- **Stage 3:** At around 23:53 (23:32 in the AWS CloudWatch plot), the average CPU utilization grows again. The utilization is: (49+18) / 2 = 33.5 > 30, which exceed the growing threshold. Two more instances needed since the growing ratio is 2.0. The two new instances are i-0e9dde8da46167825 and i-043c157fd42865122. Now we have 4 running instances in the worker pool, and the average CPU utilization is less than the threshold. Therefore, we do not need to create any new instance. From the AWS CloudWatch plot, we can observe that the average CPU utilization and number of instances in the worker pool can keep converging for 5 minutes.

ID	i-0e9dde8da46167925
Image AMI ID	ami-06be66856b59a7aaa
Key Pair	ece1779_a1
Public IP Address	35.153.168.102
State	running

ID	i-043c157fd42865122
Image AMI ID	ami-06be66856b59a7aaa
Key Pair	ece1779_a1
Public IP Address	3.92.57.10
State	stopping

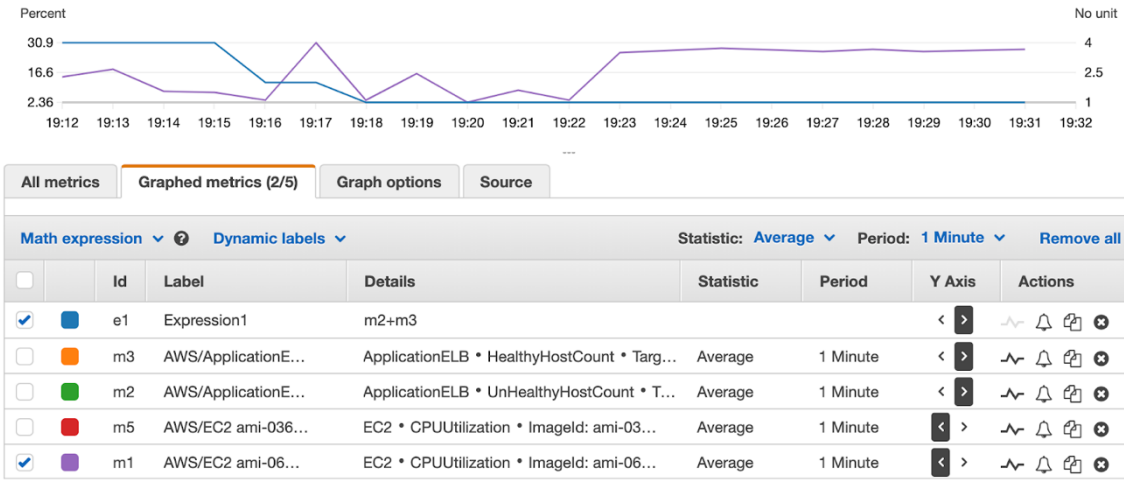


CPU utility plot for instances i-0e9dde8da46167825 and i-043c157fd42865122

5.2 Test case 2: CPU utilization dropping down.

After using a load generator to upload 100 images at a speed of 5s per image, there are 4 instances in the worker pool. The policy are as follows: threshold_growing: 32.0, shrinking: 14.0, ratio growing: 2.0, ratio shrinking: 0.5. Below is the AWS CloudWatch plot with a blue line represents the number of instances and a pulper line represents the CPU utilization.

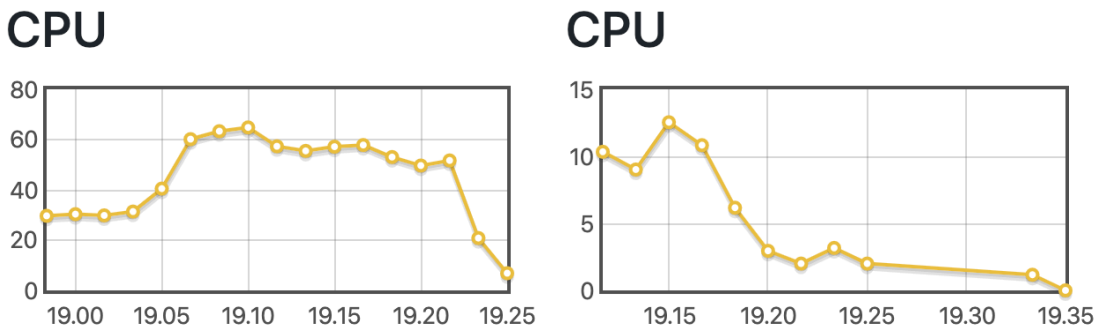
- i-0977f8d59d1b0abc2
- i-0ca9a157e57f51c09
- i-0e9dde8da46167925
- i-0fc6f2fbbaf844d03



AWS CloudWatch plot

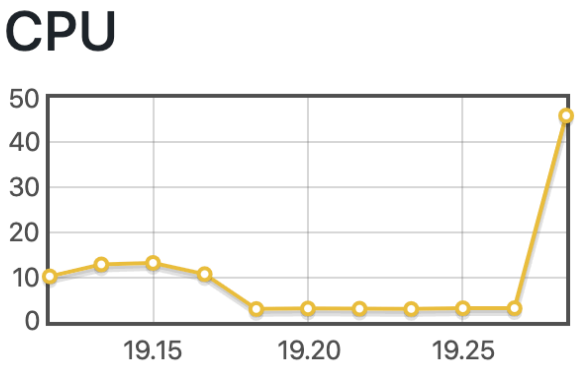
- **Stage 1:** Now we stop loading images. Then the CPU utilization drops, and the average CPU utilization is less than the threshold. By our policy, two instances need to stop. At 19:25 (19:15 in the AWS CloudWatch plot), instance i-0ca9a157e57f51c09 and i-0e9dde8da46167925 are stopped. We can observe that after the stopping, the CPU utility has a short but sharp increase.

ID	i-0ca9a157e57f51c09	ID	i-0e9dde8da46167925
Image AMI ID	ami-06be66856b59a7aaa	Image AMI ID	ami-06be66856b59a7aaa
Key Pair	ece1779_a1	Key Pair	ece1779_a1
Public IP Address	None	Public IP Address	None
State	stopped	State	stopped



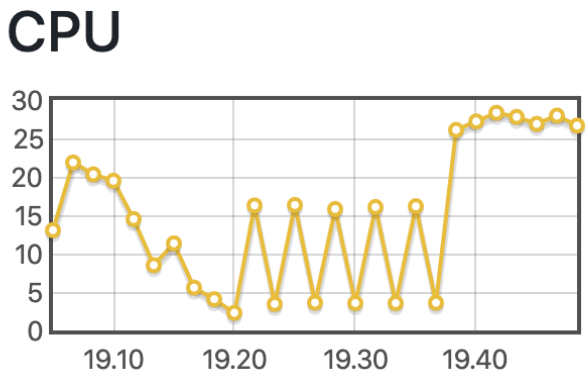
CPU utility plot for instances i-0ca9a157e57f51c09 and i-0e9dde8da46167925

- **Stage 2:** Although the CPU utilization goes up after the first two instances stopping, the autoscaling keeps shrinking since the CPU utilization drops down again. At 19:29(19:17 in the AWS CloudWatch plot), the instance i-0fc6f2fbbaf844d03 was stopped.



CPU utility plot for instance i-0fc6f2fbbaf844d03

- **Stage 3:** Now there are only on worker running, which is i-0977f8d59d1b0abc2. Its CPU utility plot is shown below. During this time (from 19:38 to 19:49), the CPU utilization stays stable in the CPU plot below. Moreover, in the AWS CloudWatch plot, we can also find that the CPU utilization and the number of instances converge for about 8 minutes.



CPU utility plot for instance i-0977f8d59d1b0abc2

6.0 Work Distribution

Qiuchen Wang	CPU and HTTP request chart User Interface Save photo to S3 and delete S3 storage
Ziyue Zhang	Number of workers chart Manually changing worker pool size Get CPU utility and set up autoscaling policy (in Autoscaling) Build databases in RDS and related interaction operations Documentation
Jun Deng	Autoscaling services, automatically grow or shrink size of worker pool with changes of CPU utilization AWS services set up and configure: EC2, S3, ELB Autoscaling functionality test and report Documentation