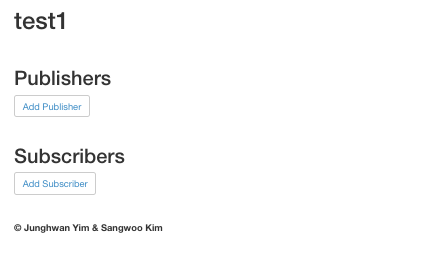
CSE 586/486 Project 2

Junghwan Yim

50173046

Publisher-Subscriber Model Implementation

1. Version 2 Centralized Publisher-Subscriber Model Implementation

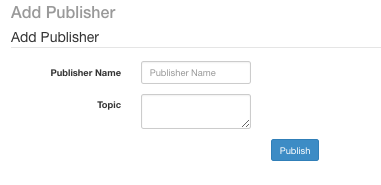
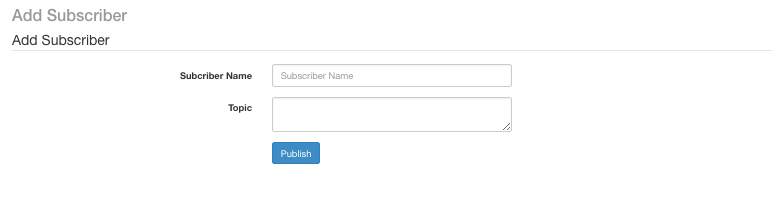
First, made the web server which run on the Docker containers with Python and Flask. The app.py is the python file which contains the web server source code and publisher-subscriber model implementation source code. By Flask, it direct to HTML files in the templates. When run the file with the command,

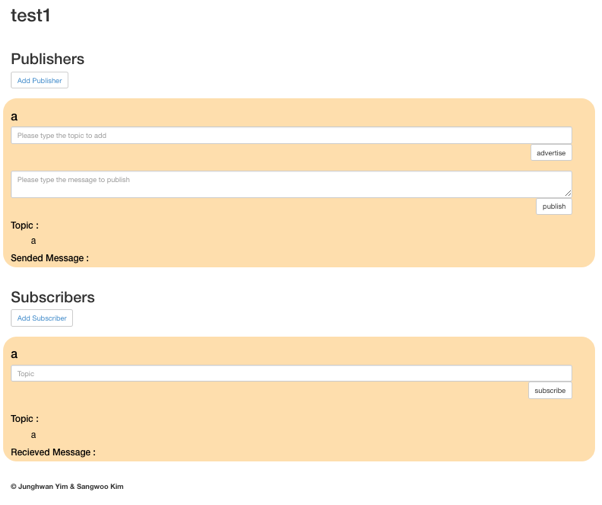
“”

python app.py

“”

The screen is shown like the left. When press the press the “Add Publishers” and “Add Subscribers”. It is connected the ADD page like the below.

When you filling this form and press the button the publisher or the subscriber will be added. If fill this form with any blank, the process is terminated without executing. After adding the publisher or subscriber, the functions “subscriber”, “publisher”, and “subscriber” button are appeared. “Advertise” and “Subscribe” functions works for adding topics for subscriber and publisher. “Publish” function makes the message and push it to message queue in the server. Then, server executes the “notify” function to distribute the messages to subscriber depends on the topic to publish.

This application is able to run through Docker with the Dockerfile which indicates configuration of the container. It reference the requirement.txt to install to run this application. To deploy this application to Docker, use the code listed below at the terminal.

“”

docker build -t phase2\_junghwanyim\_sangwookim .

docker run -p 4000:80 phase2\_junghwanyim\_sangwookim

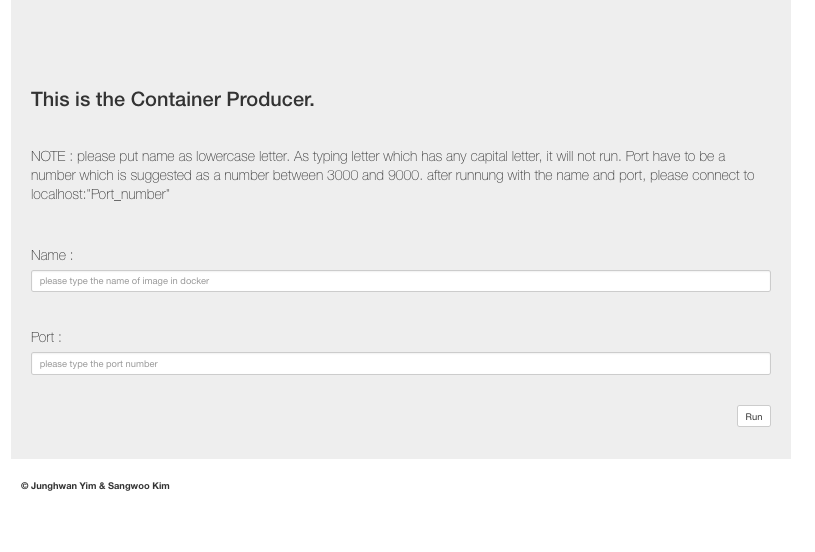
“”

When you go to localhost:4000, use this application.

1. Version 1 Interface to Utilize the Docker

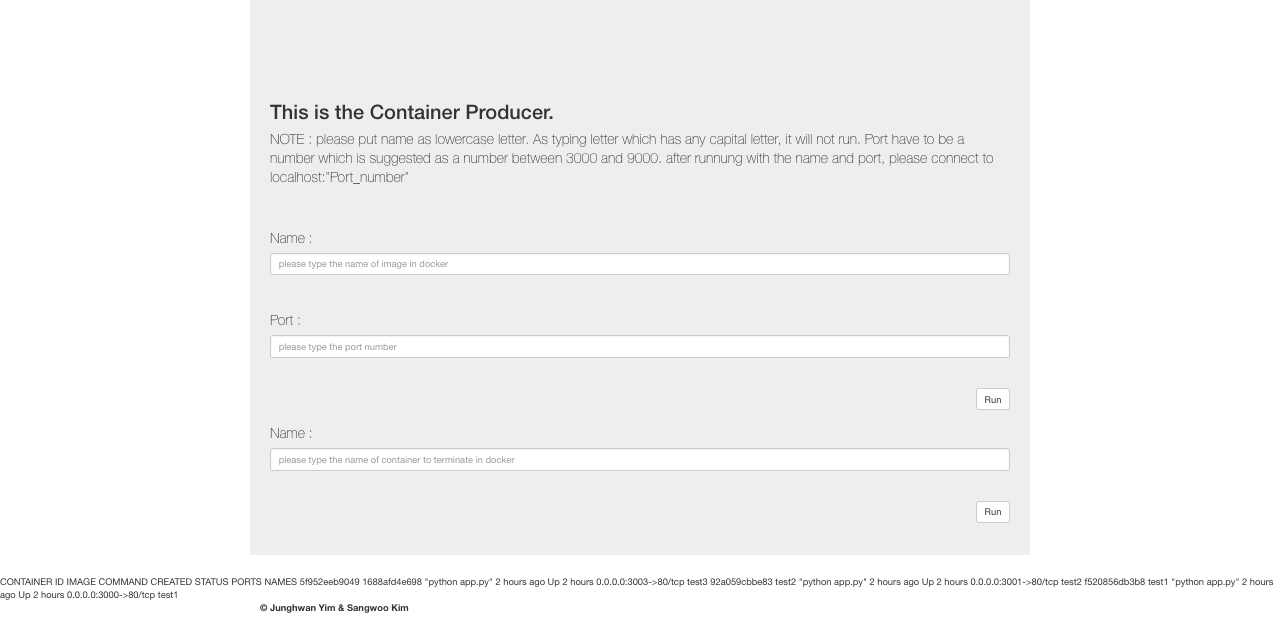
In the web interface made by PHP and HTML, there are text input to take a string to name the container generated by Docker Command and the button to submit the input to PHP server. The PHP server executes the command to run the Centralized Publisher-Subscriber application which made at the Version 2.

When press the Interface button, the result of running at the terminal is prompt at the below of interface.



1. Version 3 Distributed Publisher-Subscriber Model Implementation

To connect several centralized publisher-subscriber models, use data volume of Docker. By connecting these several the centralized publisher subscriber model, compose distributed publisher subscriber model. The interface is composed by the interface used at the Version 1, and add the deconstruction part that destruct the container which has name typed.



Through this interface, construct several containers using publisher-subscriber application. Even though it runs on the several port, it connected with the network sharing the data. Therefore, it is available to send the message from one node publisher to all node subscribers who subscribe the topic. Running method is same with the Version 1 and 2.

