Answers to the questions in milestone 4:

1. How is your project architecture related to the theory taught in the lecture?

The software architecture is as follow:

We use python as the backend, connected with the line with http connect, HTTP is a kind of remote invocation way which we mentioned in the lecture 3. In theatrically, http connect is implement above the TCP connect, TCP is a kind of socket.

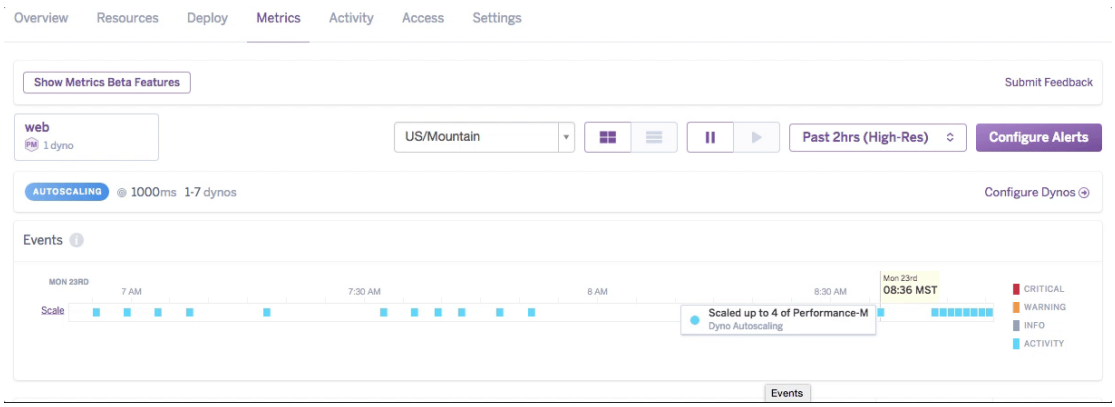
The python backend uses a lot API service which provide by some companies and the backend also use http connect to get and push message between those two part.

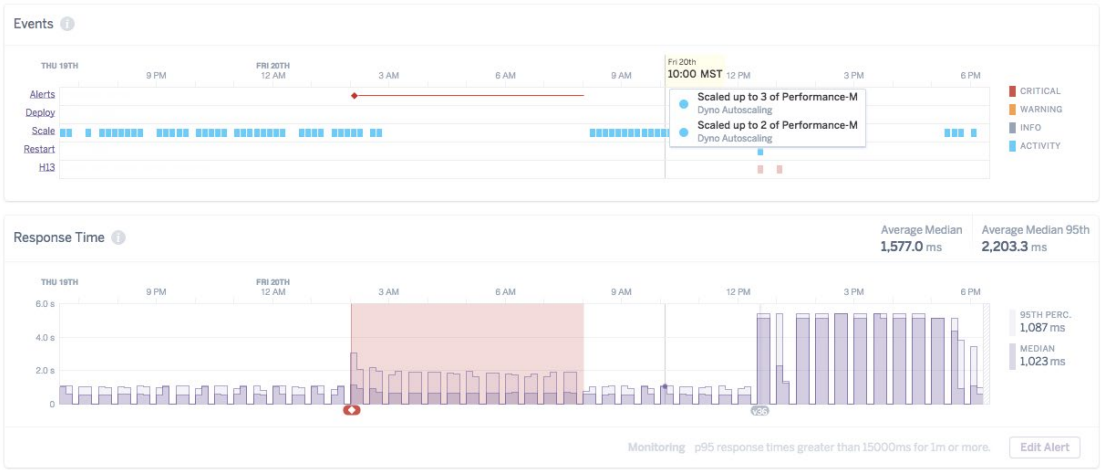
As for the redis, redis have its own connection protocol called" REdis Serialization Protocol" ,it have two connection with our lecture. First of all, it serialization the request which is similar to the technique we mentioned in the object serialization. Secondly, this protocol is implemented above TCP.

2. Can you demonstrate, with some screen cap, how to increase capacity of your chat bot service?

(a)

If the number of people who are accessing the chatbot service at the same time is too large, we can use the auto scaling function provided by heroku. It is the professional package and of course it is charged. But it can do auto scaling according to the access number on the monitor.





(b)

We can also realize the access control on the code aspect.
I wrote lines of simple code to ignore the access randomly.
For example, first to record the history access time, compared with the new access time. For each 15 seconds, we can decrease the access number and update the history access time.
If the access number is too large, 400 (assume 500 is maximum), for example, then it will randomly ignore their access at a certain probability. Thus they are not able to access the line chatbot service.

```
access_number = 0
access_time = 0
compare_time = time.time()

    global access_number
    global access_time
    global compare_time

    if access_number > 400:
        if random.randint(0, 100) < 10:
            return '400'
    elif access_number > 450:
        if random.randint(0, 100) < 30:
            return '400'

    access_number += 1
    if access_time == 0:
        access_time = time.time()
    if access_time - compare_time > 15:
        access_number -= 5 * (access_time - compare_time) // 15
        compare_time = time.time()
        if access_number <= 0:
            access_number = 0
    access_time = time.time()
```

3. Can you identify if you bot is one of the example of PaaS, IaaS, SaaS? Explain your answer.
We can use an analogy to describe these three concepts.
IaaS is the same as a hardware, like network server and hard disk. You can do anything you like with the hardware.
PaaS is the same as a virtual machine with or without an OS. You can install your software or deploy your code on the platform.
SaaS is the same as a software or an app. You can directly use them to solve some particular problems.
In my opinion, the basic heroku resource(like VPS) is an IaaS. Because it provides something like network availability and storage capability to us. We can do anything with these IT resources.
At the same time, those add-on services can be say as a PaaS. Like the redis api in heroku, you can rent the service and then deploy your code on it.
As for our chatbot, it is a SaaS. Because we can directly use it to get information we want to know. Or to chat with it, just like communicating with others.