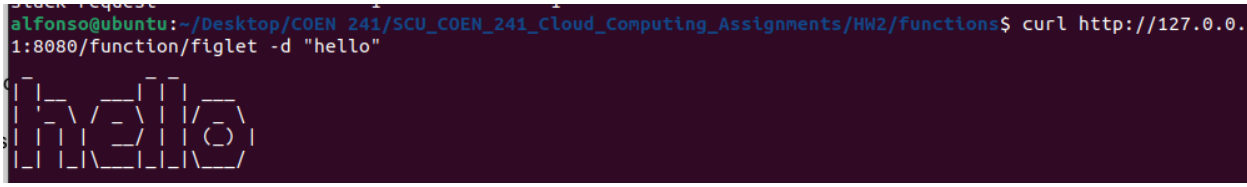
Alfonso De La Rosa
COEN 241 - Cloud Computing
02/12/23
Professor Sean Choi

COEN 241 - Homework 2

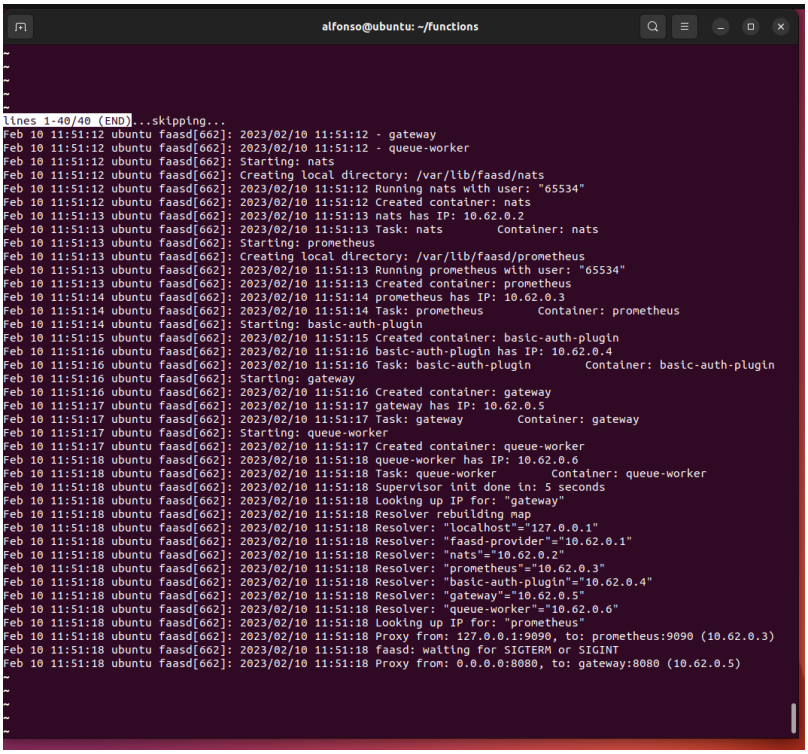
Files of the assignment:

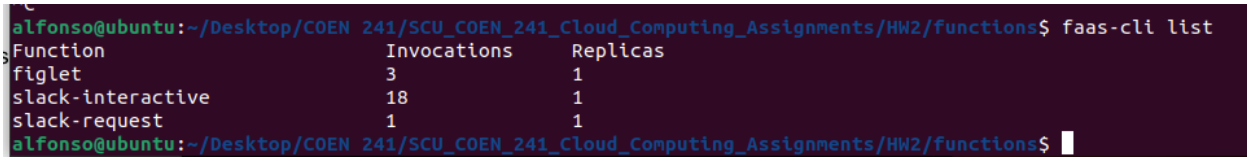
The code of slack-handle & slack-interactive functions including handler.py, requirements.txt and slack-*.yml can be found in the **functions folder**.

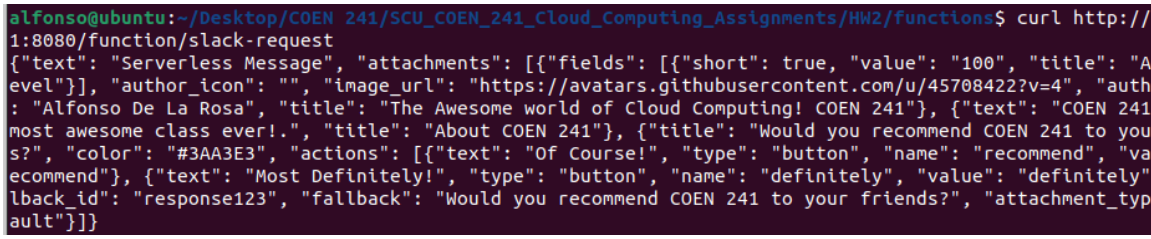
Tasks

- Provide a screenshot of invoking the figlet function (10 pts)
  - 

- ~~Complete slack-request/handler.py~~ (20 pt)
- ~~Complete slack-interactive/handler.py~~ (20 pt)
- Provide a screenshot of running the following command (10 pts)
  - sudo journalctl -u faasd --lines 40

- 

- Provide a screenshot of your OpenFaaS gateway AFTER deploying figlet, slack-handler and slack-interactive functions (10 pts)
  - 

- Provide a screenshot of invoking slack-request and slack-interactive functions (10 pts)
  - Output of slack-request
    - 

  - Output of slack-interactive
    - 

Questions

1. What is the command to invoke the slack-request function (4 pts)?
    a. Via Curl
        i. curl http://127.0.0.1:8080/function/slack-request
    b. Via faas-cli
        i. faas-cli invoke slack-request
2. What is the output you see when you invoke the slack-request function? (2 pts)
    a.


```
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "", "image_url": "https://avatars.githubusercontent.com/u/45708422?v=4", "author_name": "Alfonso De La Rosa", "title": "The Awesome world of Cloud Computing! COEN 241"}, {"text": "COEN 241 is the most awesome class ever!.", "title": "About COEN 241"}, {"title": "Would you recommend COEN 241 to your friends?", "color": "#3AA3E3", "actions": [{"text": "Of Course!", "type": "button", "name": "recommend", "value": "recommend"}, {"text": "Most Definitely!", "type": "button", "name": "definitely", "value": "definitely"}], "callback_id": "response123", "fallback": "Would you recommend COEN 241 to your friends?", "attachment_type": "default"}]}
```


    b.
3. What is the command to invoke the slack-interactive function? (4 pts)
    a. Via curl
        i. sudo curl http://127.0.0.1:8080/function/slack-interactive -d '{"message":"hello world"}'
    b. Via faas-cli
        i. echo '{"message":"hello world"}' | faas-cli invoke <function-name> --filter=req
4. What is the output you see when you invoke the slack-interactive function? (2 pts)
    a.


```
{"attachments": [{"footer": "Slack Apps built on OpenFaas", "author_link": "https://github.com/alfonsodelarosa4/SCU_COEN_241_Cloud_Computing_Assignments", "color": "#36a64f", "text": "Head over to COEN 241", "title": "COEN 241", "ts": 123456789, "author_name": "Alfonso De La Rosa", "title_link": "https://www.scu.edu/engineering/academic-programs/department-of-computer-engineering/graduate/course-descriptions/", "image_url": "https://www.scu.edu/media/offices/umc/scu-brand-guidelines/visual-identity-amp-photography/visual-identity-toolkit/logos-amp-seals/Mission-Dont3.png", "response_type": "ephemeral", "replace_original": true, "footer_icon": "https://a.slack-edge.com/45901/marketing/img/_rebrand/meta/slack_hash_256.png", "pretext": "Ahh yeah! Great choice, COEN 241 is absolutely amazing!", "fallback": "Required plain-text summary of the attachment.", "thumb_url": "https://www.scu.edu/engineering/academic-programs/department-of-computer-engineering/graduate/course-descriptions/", "author_icon": "https://avatars.githubusercontent.com/u/45708422?v=4"}]}
```


    b.

5. How would you pass different arguments to the functions? (4 pts)
    a. There are different ways of passing different arguments to these functions. The two main ways of calling a function are either through curl or faas-cli. And, there are three types of arguments to pass: standard input, query string, and JSON Payload. The first way of passing an argument is the standard input:
        i. For curl:
            1. curl http://127.0.0.1:8080/function/slack-interactive -d **“parameter_value”**
        ii. For faas-cli:
            1. **echo parameter_value |** faas-cli invoke slack-interactive **--filter=parameter_name**
    b. The second way of passing an argument is the query string:
        i. For curl:
            1. curl http://127.0.0.1:8080/function/slack-interactive**?[parameter_name]=[parameter_value]**
        ii. For faas-cli:
            1. faas-cli invoke slack-interactive --gateway=http://127.0.0.1:8080**?[parameter_name]=[parameter_value]**
    c. The third way of passing an argument is the JSON Payload
        i. For curl:
            1. curl -X POST -H **"Content-Type: application/json" -d '{"[parameter_name]": "[parameter_value]"}'** http://127.0.0.1:8080/function/slack-interactive
        ii. For faas-cli:
            1. faas-cli invoke slack-interactive **--data='{"[parameter-name]": "[parameter_value]"}'**
6. How would you change the slack-interactive function to react to different inputs? (4 pts)
    a. As mentioned in the previous response, there are three types of arguments. Therefore, there are different ways a slack-interactive function can process these inputs:
        i. Case: A standard input is passed. The value would not be reformatted.
            1.

```
def handle(req):
        print(“Request received: ” + req)
```

            2.
        ii. Case: query string is passed. The url query string would need to be processed. First the value would need to be split by ‘?’ since the query string is composed of the gateway url + the query parameters. Then the query parameters would be converted as a python dictionary.
            1.

```
def handle(req):
```

```
        # split the value by '?'
        req = req.split('?')
        # convert the query string of parameters to a dictionary
        hash = urllib.parse.parse._qs(req)

        # use the resulting dictionary in the function
```

2.

iii. Case: JSON Payload. The JSON payload would need to be converted to a python dictionary.

1.

```
def handle(req):
        # convert string that is formatted as a JSON payload to a
        python dictionary
        json_req = json.loads(req)

        # use the resulting dictionary in the function
```

2.