

# CC COEN-241 HW - 2

Nityanand Pujari – W1650422

- 1. Provide a screenshot of invoking the figlet function (5 pts)**

```
ubuntu@faasd:~$ faas-cli store deploy figlet
WARNING! You are not using an encrypted connection to the gateway, consider using HTTPS.

Deployed. 200 OK.
URL: http://10.101.54.43:8080/function/figlet

ubuntu@faasd:~$ faas-cli store inspect figlet
Title:      figlet
Author:     openfaas
Description:
Generate ASCII logos with the figlet CLI

Image:      ghcr.io/openfaas/figlet:latest
Process:    figlet
Repo URL:   https://github.com/openfaas/store-functions
ubuntu@faasd:~$ echo "Hello, FaaS, world" | faas-cli invoke figlet

Hello FaaS, world

ubuntu@faasd:~$
```

2. Provide a screenshot of running the following command (5 pts)
- ```
sudo journalctl -u faasd --lines 40
```

```

ubuntu@faasd:~$ sudo journalctl -u faasd --lines 40
Feb 22 22:15:55 faasd faasd[4869]: Removing old container for: nats
Feb 22 22:15:55 faasd faasd[4869]: Removing old container for: prometheus
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 Start-up order:
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 - nats
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 - prometheus
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 - gateway
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 - queue-worker
Feb 22 22:15:55 faasd faasd[4869]: Starting: nats
Feb 22 22:15:55 faasd faasd[4869]: Creating local directory: /var/lib/faasd/nats
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 Running nats with user: "65534"
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 Created container: nats
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 nats has IP: 10.62.0.2
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 Task: nats Container: nats
Feb 22 22:15:55 faasd faasd[4869]: Starting: prometheus
Feb 22 22:15:55 faasd faasd[4869]: Creating local directory: /var/lib/faasd/prometheus
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 Running prometheus with user: "65534"
Feb 22 22:15:55 faasd faasd[4869]: 2024/02/22 22:15:55 Created container: prometheus
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 prometheus has IP: 10.62.0.3
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Task: prometheus Container: prometheus
Feb 22 22:15:56 faasd faasd[4869]: Starting: gateway
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Created container: gateway
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 gateway has IP: 10.62.0.4
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Task: gateway Container: gateway
Feb 22 22:15:56 faasd faasd[4869]: Starting: queue-worker
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Created container: queue-worker
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 queue-worker has IP: 10.62.0.5
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Task: queue-worker Container: queue-worker
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Supervisor init done in: 12 seconds
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Looking up IP for: "prometheus"
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Resolver rebuilding map
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Resolver: "localhost"="127.0.0.1"
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Resolver: "faasd-provider"="10.62.0.1"
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Resolver: "nats"="10.62.0.2"
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Resolver: "prometheus"="10.62.0.3"
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Resolver: "gateway"="10.62.0.4"
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Resolver: "queue-worker"="10.62.0.5"
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Looking up IP for: "gateway"
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Proxy from: 0.0.0.0:8080, to: gateway:8080 (10.62.0.4)
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 faasd: waiting for SIGTERM or SIGINT
Feb 22 22:15:56 faasd faasd[4869]: 2024/02/22 22:15:56 Proxy from: 127.0.0.1:9090, to: prometheus:9090 (10.62.0.3)
ubuntu@faasd:~$

```

- ### 3. Complete slack-request/handler.py (10 pts)

```

ubuntu@faasd:~/functions/slack-requests$ cat handler.py
import json

def handle(req):
    data = {
        "text": "Serverless Message",
        "attachments": [{
            "title": "The Awesome world of Cloud Computing! COEN 241",
            "fields": [{
                "title": "Amazing Level",
                "value": "100",
                "short": True
            }],
            "author_name": "Nityanand Pujari",
            "author_icon": "https://github.com/Nitai2123.png",
            "image_url": "https://github.com/Nitai2123.png"
        }],
        {
            "title": "About COEN 241",
            "text": "COEN 241 is the most awesome class ever!."
        },
        {
            "fallback": "Would you recommend COEN 241 to your friends?",
            "title": "Would you recommend COEN 241 to your friends?",
            "callback_id": "response123",
            "color": "#3AA3E3",
            "attachment_type": "default",
            "actions": [
                {
                    "name": "recommend",
                    "text": "Of Course!",
                    "type": "button",
                    "value": "recommend"
                },
                {
                    "name": "definitely",
                    "text": "Most Definitely!",
                    "type": "button",
                    "value": "definitely"
                }
            ]
        }
    ]
}
return json.dumps(data)

```

#### 4. Complete slack-interactive/handler.py (10 pts)

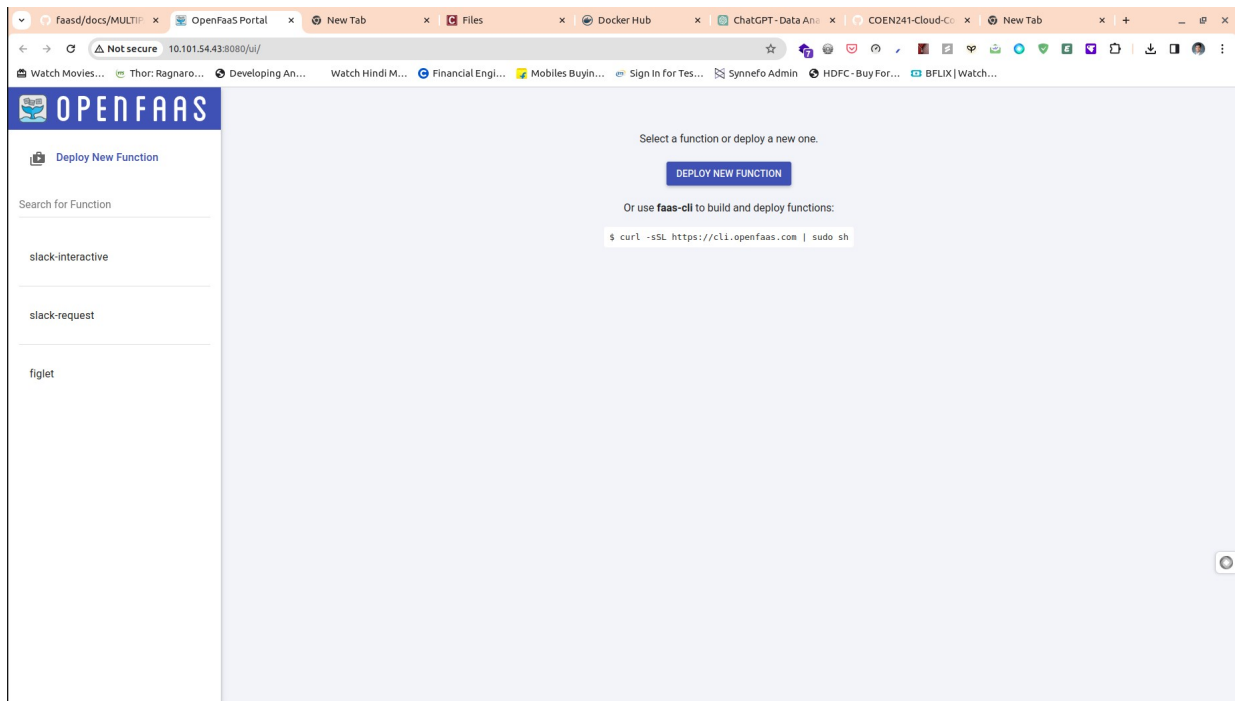
```

ubuntu@faasd:~/functions/slack-interactive$ cat handler.py
import json
import urllib

def handle(req):
    urlstring = urllib.unquote(req.decode('utf8')).strip('payload=')
    response = json.loads(urlstring)
    data = {
        "attachments": [
            {
                "replace_original": True,
                "response_type": "ephemeral",
                "fallback": "Required plain-text summary of the attachment.",
                "color": "#36a64f",
                "pretext": "Ahh yeah! Great choice, COEN 241 is absolutely amazing!",
                "author_name": "Nityanand Pujari",
                "author_link": "https://github.com/Nitai2123",
                "author_icon": "https://github.com/Nitai2123.png",
                "title": "COEN 241",
                "title_link": "https://www.scu.edu/engineering/academic-programs/departments-of-computer-engineering/graduate/course-descriptions/",
                "text": "Head over to COEN 241",
                "image_url": "https://www.scu.edu/media/offices/umc/scu-brand-guidelines/visual-identity-and-photography/visual-identity-toolkit/logos-and-seals/Mission-Dont3.png",
                "thumb_url": "https://www.scu.edu/engineering/academic-programs/departments-of-computer-engineering/graduate/course-descriptions/",
                "footer": "Slack Apps built on OpenFaaS",
                "footer_icon": "https://a.slack-edge.com/45901/marketing/img/_rebrand/meta/slack_hash_256.png",
                "ts": 123456789
            }
        ]
    }
    return json.dumps(data)

```

#### 5. Provide a screenshot of your OpenFaaS gateway AFTER deploying figlet, slack-handler and slack-interactive functions (5 pts)

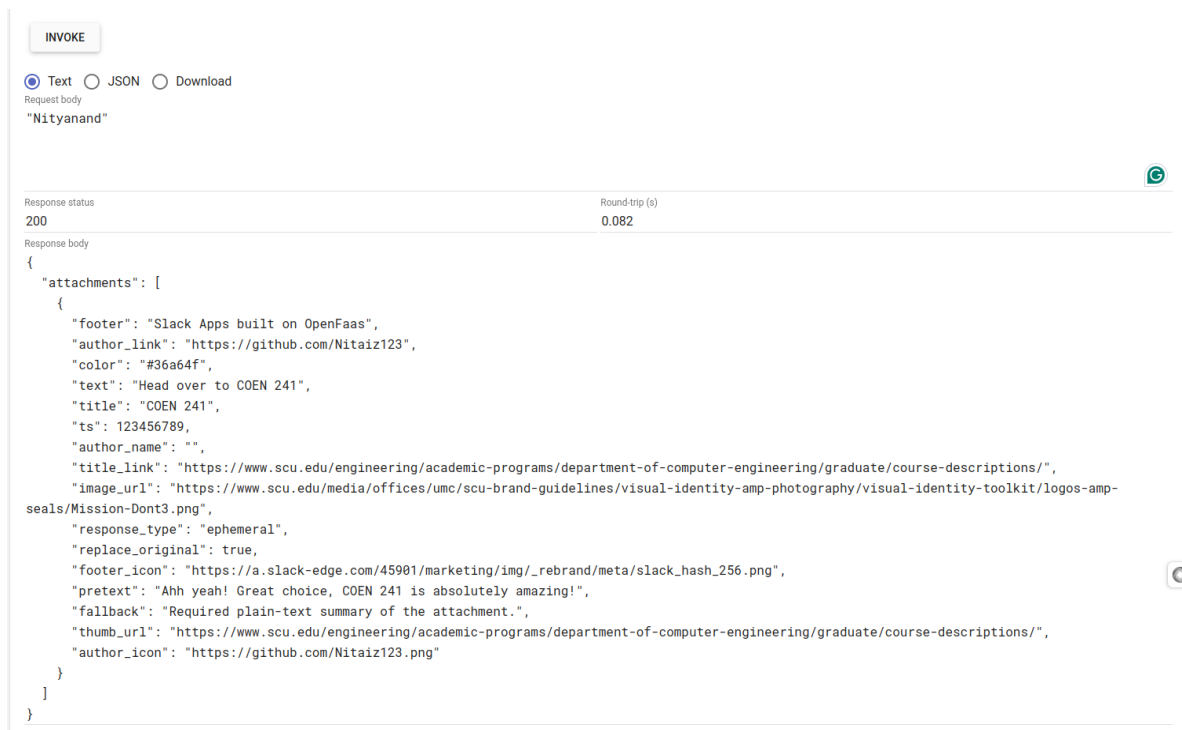


6. Provide a screenshot of invoking slack-request and slack-interactive functions (5 pts)

**Invoking Slack-Request:**

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Response status:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Round-trip (s) |
| 200                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0.055          |
| Response body                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                |
| <pre> {   "text": "Serverless Message",   "attachments": [     {       "fields": [         {           "short": true,           "value": "100",           "title": "Amazing Level"         }       ],       "author_icon": "https://github.com/Nitaiz123.png",       "image_url": "https://github.com/Nitaiz123.png",       "author_name": "Nityanand Pujari",       "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"     }   ] } </pre> |                |

## Invoking Slack-Interactive:



## 7. Complete the chatbot with a yml file (25pt)

```
ubuntu@faasd:~/chatbot$ cat handler.py
def handle(req):
    """Process incoming requests based on the input text"""
    if "name" in req.lower() or "what is your name" in req.lower():
        # Respond with the bot's name in 3 different ways
        responses = [
            "My name is Coen241.",
            "I'm called Coen241.",
            "You can call me COEN241."
        ]
        return "\n".join(responses)
    elif "current time" in req.lower() or "current date" in req.lower():
        # Respond with the current date and time in 3 different ways
        now = datetime.datetime.now()
        responses = [
            now.strftime("The current time is %H:%M on %B %d, %Y."),
            now.strftime("It's now %H:%M on %d/%m/%Y."),
            now.strftime("Today is %B %d, %Y, and the time is %H:%M.")
        ]
        return "\n".join(responses)
    elif req.lower().startswith("generate a figlet for"):
        # Extract the text to generate figlet
        text = req[len("generate a figlet for"):].strip("\n ")
        # For the purpose of this example, we'll simulate figlet output using PyFiglet
        f = Figlet(font='slant')
        return f.renderText(text)
    else:
        return "I'm not sure how to process that request."
ubuntu@faasd:~/chatbot$
```

```
ubuntu@faasd:~$ cat chatbot.yml
version: 1.0
provider:
  name: openfaas
  gateway: http://10.101.54.43:8080
functions:
  chatbot:
    lang: python
    handler: ./chatbot
    image: nitaiz/chatbot:latest
ubuntu@faasd:~$
```

## 8. Provide a screenshot of invoking three different cases of the chatbot (5 pts)

## chatbot



Status  
**Not ready**

Replicas  
**1**

Invocation count  
**26**

Image  
**docker.io/nitaiz/chatbot:latest**

URL  
**http://10.101.54.43:8080/function/chatbot**



Function process  
**python index.py**

### Invoke function

INVOKE

☒ Text ☐ JSON ☐ Download

Request body

**"name"**



Response status  
**200**

Round-trip (s)  
**0.164**

Response body

**My name is Coen241.  
I'm called Coen241.  
You can call me COEN241.**



## chatbot



Status  
**Not ready**

Replicas  
**1**

Invocation count  
**26**

Image  
**docker.io/nitaiz/chatbot:latest**

URL  
**http://10.101.54.43:8080/function/chatbot**



Function process  
**python index.py**

### Invoke function

INVOKE

☒ Text ☐ JSON ☐ Download

Request body

**"current time"**



Response status  
**200**

Round-trip (s)  
**0.202**

Response body

**The current time is 21:28 on February 23, 2024.  
It's now 21:28 on 23/02/2024.  
Today is February 23, 2024, and the time is 21:28.**



```
ubuntu@faasd:~$ echo "generate a figlet for Nityanand" | faas-cli invoke chatbot
```



```
ubuntu@faasd:~$
```

## 1. What is the command to invoke the slack-request function (2 pts)?

### a. Via Curl

→ `curl -d '{"Nityanand":"COEN 241"}' http://10.101.54.43:8080/function/slack-request`

### b. Via faas-cli

→ `echo "Nityanand Pujari" | faas-cli invoke slack-request`

## 2. What is the output you see when you invoke the slack-request function? (2 pts)

```
ubuntu@faasd:~/functions$ curl -d '{"Nityanand":"COEN 241"}' http://10.101.54.43:8080/function/slack-request
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "https://github.com/Nitaiz123.png", "image_url": "https://github.com/Nitaiz123.png", "author_name": "Nityanand Pujari", "title": "The Awesome world of Cloud Computing! COEN 241", {"text": "COEN 241 is the most awesome class ever!.", "title": "About COEN 241", {"text": "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"}]}
ubuntu@faasd:~/functions$
```

```
ubuntu@faasd:~$ echo "Nityanand Pujari" | faas-cli invoke slack-request
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "https://github.com/Nitaiz123.png", "image_url": "https://github.com/Nitaiz123.png", "author_name": "Nityanand Pujari", "title": "The Awesome world of Cloud Computing! COEN 241", {"text": "COEN 241 is the most awesome class ever!.", "title": "About COEN 241", {"text": "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"}]}
ubuntu@faasd:~$
```

## 3. What is the command to invoke the slack-interactive function? (2 pts)

### a. Via curl

→ `curl -d '{"Nityanand":"COEN 241"}' http://10.101.54.43:8080/function/slack-interactive`

### b. Via faas-cli

→ `sudo faas-cli invoke slack-interactive`

## 4. What is the output you see when you invoke the slack-interactive function? (2 pts)

```
ubuntu@faasd:~$ curl -d '{"Nityanand":"COEN 241"}' http://10.101.54.43:8080/function/slack-interactive
{"attachments": [{"footer": "Slack Apps built on OpenFaas", "author_link": "https://github.com/Nitaiz123", "color": "#36a64f", "text": "Head over to COEN 241", "title": "COEN 241", "ts": 123456789, "author_name": "", "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-and-photography/visual-identity-toolkit/logos-and-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 plaintext summary of the attachment.", "thumb_url": "https://www.scu.edu/engineering/academic-programs/department-of-computer-engineering/graduate/course-descriptions/", "author_icon": "https://github.com/Nitaiz123.png"}]}
ubuntu@faasd:~$
```

```
ubuntu@faasd:~/functions$ sudo faas-cli invoke slack-interactive
Reading from STDIN - hit (Control + D) to stop.
"Hi"
{"attachments": [{"footer": "Slack Apps built on OpenFaas", "author_link": "https://github.com/Nitaiz123", "color": "#36a64f", "text": "Head over to COEN 241", "title": "COEN 241", "ts": 123456789, "author_name": "", "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-and-photography/visual-identity-toolkit/logos-and-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 plaintext summary of the attachment.", "thumb_url": "https://www.scu.edu/engineering/academic-programs/department-of-computer-engineering/graduate/course-descriptions/", "author_icon": "https://github.com/Nitaiz123.png"}]}
ubuntu@faasd:~/functions$
```

## 5. How would you pass different arguments to the functions? (3 pts)

Here are two ways of passing arguments to functions:

1. Using faas-cli: We can invoke the figlet function from the command line using faas-cli and pipe in the argument "Hello, FaaS World" like this:  
echo "Hello, FaaS World" | faas-cli invoke figlet  
The text "Hello, FaaS World" gets passed as an argument to the figlet function.
2. Using curl: We can send arguments in a POST request to an HTTP server. For example:
3. curl -d '{"text":"Hello COEN 241"}' <http://example.com/function/slack-request>  
This curl command sends a JSON payload {"text":"Hello COEN 241"} as data in a POST request to the /function/slack-request endpoint. The text "Hello COEN 241" gets passed as an argument to the function.

#### **6. How would you change the slack-interactive function to react to different inputs? (3 pts)**

→ The unquote function from the urllib package is used to decode a string into a URL, using UTF-8 encoding by default. The json.load() method then transforms the urlstring JSON string into a Python dictionary, stored in the response variable.

However, the response variable is currently not used after being assigned. To enable the slack-interactive function to handle different inputs, we can modify it to use the response data.

Since response is a Python dictionary, it can be incorporated into the existing data dictionary. This will allow the function to start responding to varied inputs. By returning json.dumps(data), the slack-interactive function can now handle and respond to diverse inputs, based on the response data.

#### **7. How long does it take for the chat response to come back? (10pts)**

- a. For the first request that does not call figlet
- b. For the second request that does not call figlet
- c. Average over 10 requests that do not call figlet
- d. For the first request that calls figlet
- e. For the second request that calls figlet
- f. For the second request that calls figlet that follows the first request that does not call figlet
- g. Average over 10 requests that do call figlet



```

import requests
import time

# Your chatbot's endpoint
CHATBOT_URL = "http://10.101.54.43:8080/function/chatbot"

def measure_response_time(data, repeat=1):
    """Measure the response time for a single request or the average over multiple requests."""
    total_time = 0
    for _ in range(repeat):
        start_time = time.time()
        response = requests.post(CHATBOT_URL, data=data)
        end_time = time.time()
        total_time += (end_time - start_time)
        if repeat == 1: # If only one request, return its time directly
            return end_time - start_time
    return total_time / repeat # Return average time if multiple requests

def main():
    # Measure response times according to the given scenarios
    # a. For the first request that does not call figlet
    response_time_a = measure_response_time("What is your name?")
    print(f'a. Response time for the first request (no figlet): {response_time_a:.4f} seconds')

    # b. For the second request that does not call figlet
    response_time_b = measure_response_time("What is your name?")
    print(f'b. Response time for the second request (no figlet): {response_time_b:.4f} seconds')

    # c. Average over 10 requests that do not call figlet
    average_response_time_c = measure_response_time("What is your name?", repeat=10)
    print(f'c. Average response time over 10 requests (no figlet): {average_response_time_c:.4f} seconds')

    # d. For the first request that calls figlet
    response_time_d = measure_response_time("Generate a figlet for Hello")
    print(f'd. Response time for the first request (with figlet): {response_time_d:.4f} seconds')

    # e. For the second request that calls figlet
    response_time_e = measure_response_time("Generate a figlet for Hello")
    print(f'e. Response time for the second request (with figlet): {response_time_e:.4f} seconds')

    # f. For the second request that calls figlet following the first request that does not call figlet
    # Measure the first request (no figlet)
    measure_response_time("What is your name?")
    # Measure the second request (with figlet)
    measure_response_time("Generate a figlet for Hello")

-- INSERT --

```

1,1

Top

```

ubuntu@faasd:~$ python3 chatbot_avg.py
a. Response time for the first request (no figlet): 0.1464 seconds
b. Response time for the second request (no figlet): 0.1449 seconds
c. Average response time over 10 requests (no figlet): 0.1483 seconds
d. Response time for the first request (with figlet): 0.1524 seconds
e. Response time for the second request (with figlet): 0.1528 seconds
f. Response time for the second request (with figlet, after no figlet): 0.1526 seconds
g. Average response time over 10 requests (with figlet): 0.1527 seconds
ubuntu@faasd:~$

```

## 8. Now try sending a series of requests to the chatbot in parallel. At what queries per second does OpenFaaS add a new instance of the function? (6 pts)

```

ubuntu@faasd:~$ cat chatbot_parallel.py
import concurrent.futures
import requests
import time

# Your chatbot's endpoint
CHATBOT_URL = "http://10.101.54.43:8080/function/chatbot"

def send_request(data):
    """Function to send a single request to the chatbot."""
    try:
        response = requests.post(CHATBOT_URL, data=data)
        return response.status_code
    except Exception as e:
        return str(e)

def fire_requests(requests_per_second, duration_in_seconds=10):
    """Fire requests in parallel aiming for a target rate of requests per second."""
    with concurrent.futures.ThreadPoolExecutor() as executor:
        futures = []
        start_time = time.time()

        while time.time() - start_time < duration_in_seconds:
            for _ in range(requests_per_second):
                # Adjust the request content as needed
                future = executor.submit(send_request, "What is your name?")
                futures.append(future)
            time.sleep(1) # Wait for a second before firing the next batch of requests

        results = [future.result() for future in futures]

        success_responses = [result for result in results if result == 200]
        print(f'Total requests: {len(results)}')
        print(f'Successful responses: {len(success_responses)}')
        print(f'Success rate: {len(success_responses) / len(results) * 100:.2f}%')



# Example usage:
requests_per_second = 5 # Adjust this to test different load levels
fire_requests(requests_per_second)

```

```

ubuntu@faasd:~$ python3 chatbot_parallel.py
Total requests: 50
Successful responses: 50
Success rate: 100.00%

```

|                                |                                           |                  |                                                                                     |
|--------------------------------|-------------------------------------------|------------------|-------------------------------------------------------------------------------------|
| chatbot                        |                                           |                  |  |
| Status                         | Replicas                                  | Invocation count |                                                                                     |
| Ready                          | 2                                         | 91               |                                                                                     |
| Image                          | URL                                       |                  |  |
| docker.io/nitai/chatbot:latest | http://10.101.54.43:8080/function/chatbot |                  |                                                                                     |
| Function process               |                                           |                  |                                                                                     |
| python index.py                |                                           |                  |                                                                                     |

## Extra Credit

### Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}'  
https://hooks.slack.com/services/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
```

Copy

| Webhook URL                                                                 | Channel           | Added By                                       |
|-----------------------------------------------------------------------------|-------------------|------------------------------------------------|
| <a href="https://hooks.slack.com/">https://hooks.slack.com/</a> <p>Copy</p> | #incoming-webhook | Nityanand Rajendrakumar Pujari<br>Feb 23, 2024 |



Add New Webhook to Workspace

api.slack.com/apps/A06LT3TKCKT/slash-commands?

Watch Movies...Thor: Ragnar...Developing An...Watch Hindi M...Financial Engi...Mobiles Buyin...Sign In for Tes...Synnefo AdminHDFC - Buy For...BFLIX | Watch...

Create New Command

Command/coen241

Request URLhttp://10.101.54.43:8080/function/...

Short DescriptionExtra Credit - Nityanand Pujari

Usage Hint[which rocket to launch]  
Optionally list any parameters that can be passed.

Escape channels, users, and links sent to your app  
Unescaped: @user #general

Preview of Autocomplete Entry  
Commands matching "coen241"  
OpeenFaas-Integration  
/coen241Extra Credit - Nityanand Pu...  
+/coen241

CancelSave

SCU Grad Engineering

Upgrade Plan

Home

Channels

# general

# grad-engr-fall-22-admits

# housing

# incoming-webhook

# Introductions

+ Add channels

Direct messages

Nityanand Rajendrakumar Pujari you

+ Add coworkers

Apps

+ Add apps

Search SCU Grad Engineering

# incoming-webhook

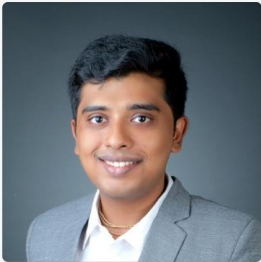
+ Add a bookmark

Nityanand Pujari

The Awesome world of Cloud Computing! COEN 241 (1.6 kB) +

Amazing Level

100



About COEN 241

COEN 241 is the most awesome class ever!.

Would you recommend COEN 241 to your friends?

Of Course!

Most Definitely!

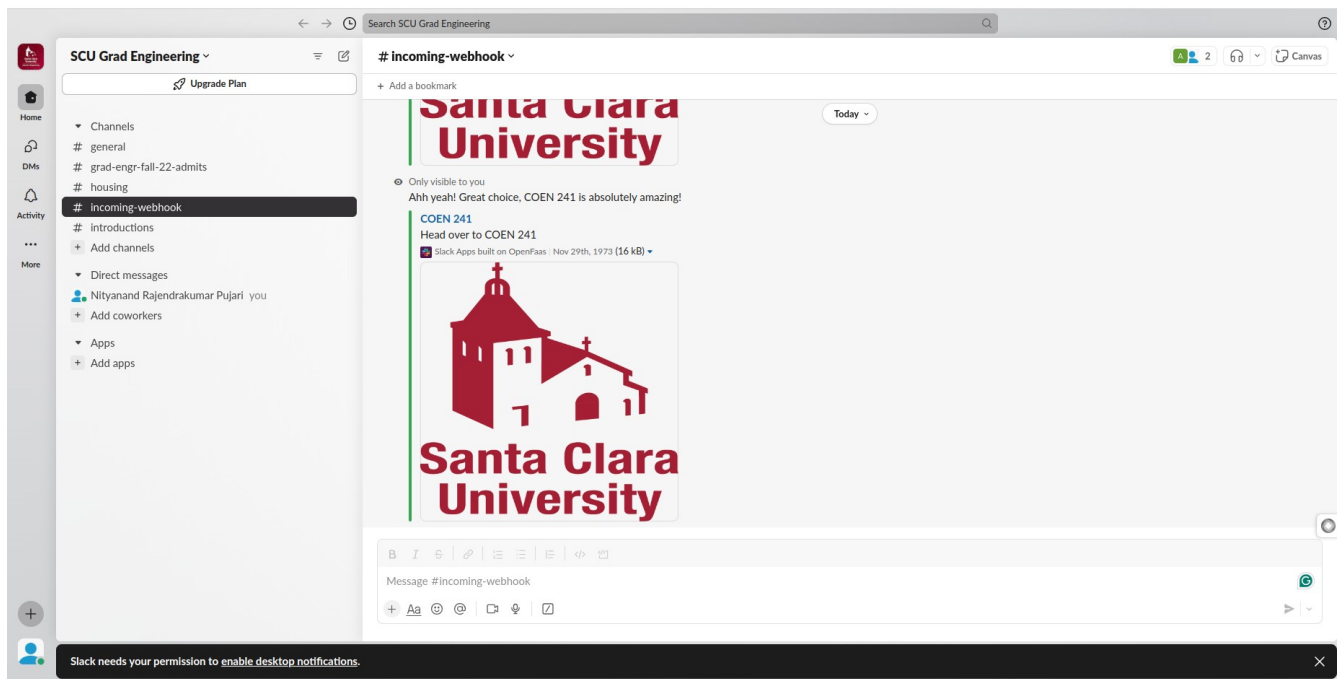
Today

B I [link icon] [list icon] [list icon] [code icon]

Message #incoming-webhook

+ [text icon] [image icon] [video icon] [link icon]

Slack needs your permission to enable desktop notifications.



```
ngrok (Ctrl+C to quit)

Try the new Traffic Inspector dev preview: https://ngrok.com/r/tl

Session Status      online
Account             Nityanand Pujari (Plan: Free)
Version             3.6.0
Region              United States (California) (us-cal-1)
Latency              19ms
Web Interface        http://127.0.0.1:4040
Forwarding            https://9af1-24-23-244-181.ngrok-free.app -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p90
                     4      0      0.01   0.01   60.05  60.05

HTTP Requests
-----
POST /function/slack-interactive 200 OK
POST /function/slack-request     200 OK
POST /function/slack-interactive 200 OK
POST /function/slack-request     200 OK
```

## LINKS:

- Link for application:  
→ <https://app.slack.com/client/T03QJD2DLH0/C048H40RJ5C>
- Invite link to join the workspace:

→ [https://join.slack.com/t/scugradengineering/shared\\_invite/zt-2dl9e5ng3-A9RNJVbwUGuhOimGQUkGTQ](https://join.slack.com/t/scugradengineering/shared_invite/zt-2dl9e5ng3-A9RNJVbwUGuhOimGQUkGTQ)

- Slash-command url:

→ <https://9af1-24-23-244-181.ngrok-free.app/function/slack-request>