

CSEN-241 Cloud Computing

HW2: Your Own Serverless Infrastructure

Submitted by: Rishabh Agrawal
(W1651177)

Tasks:

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

[illegible]

- Provide a screenshot of running the following command (5 pts)

sudo journalctl -u faasd --lines 40

```

rishabh@rishabh:~/faasd$ sudo journalctl -u faasd --lines 40
Feb 27 05:18:35 rishabh faasd[5960]: Removing old container for: prometheus
Feb 27 05:18:35 rishabh faasd[5960]: Removing old container for: gateway
Feb 27 05:18:35 rishabh faasd[5960]: 2024/02/27 05:18:35 Start-up order:
Feb 27 05:18:35 rishabh faasd[5960]: 2024/02/27 05:18:35 - nats
Feb 27 05:18:35 rishabh faasd[5960]: 2024/02/27 05:18:35 - queue-worker
Feb 27 05:18:35 rishabh faasd[5960]: 2024/02/27 05:18:35 - prometheus
Feb 27 05:18:35 rishabh faasd[5960]: 2024/02/27 05:18:35 - gateway
Feb 27 05:18:35 rishabh faasd[5960]: Starting: nats
Feb 27 05:18:35 rishabh faasd[5960]: Creating local directory: /var/lib/faasd/nats
Feb 27 05:18:35 rishabh faasd[5960]: 2024/02/27 05:18:35 Running nats with user: "65534"
Feb 27 05:18:35 rishabh faasd[5960]: 2024/02/27 05:18:35 Created container: nats
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 nats has IP: 10.62.0.2
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Task: nats Container: nats
Feb 27 05:18:36 rishabh faasd[5960]: Starting: queue-worker
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Created container: queue-worker
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 queue-worker has IP: 10.62.0.3
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Task: queue-worker Container: queue-worker
Feb 27 05:18:36 rishabh faasd[5960]: Starting: prometheus
Feb 27 05:18:36 rishabh faasd[5960]: Creating local directory: /var/lib/faasd/prometheus
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Running prometheus with user: "65534"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Created container: prometheus
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 prometheus has IP: 10.62.0.4
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Task: prometheus Container: prometheus
Feb 27 05:18:36 rishabh faasd[5960]: Starting: gateway
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Created container: gateway
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 gateway has IP: 10.62.0.5
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Task: gateway Container: gateway
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Supervisor init done in: 14 seconds
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Looking up IP for: "gateway"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Resolver rebuilding map
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Resolver: "localhost"="127.0.0.1"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Resolver: "faasd-provider"="10.62.0.1"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Resolver: "nats"="10.62.0.2"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Resolver: "queue-worker"="10.62.0.3"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Resolver: "prometheus"="10.62.0.4"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Resolver: "gateway"="10.62.0.5"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Looking up IP for: "prometheus"
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Proxy from: 127.0.0.1:9090, to: prometheus:9090 (10.62.0.4)
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 faasd: waiting for SIGTERM or SIGINT
Feb 27 05:18:36 rishabh faasd[5960]: 2024/02/27 05:18:36 Proxy from: 0.0.0.0:8080, to: gateway:8080 (10.62.0.5)

```

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

```

rishabh@rishabh:~/functions/slack-request$ 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": "Rishabh Agrawal",
            "author_icon": "https://github.com/agrawal-rishabh-manoj.png",
            "image_url": "https://github.com/agrawal-rishabh-manoj.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)

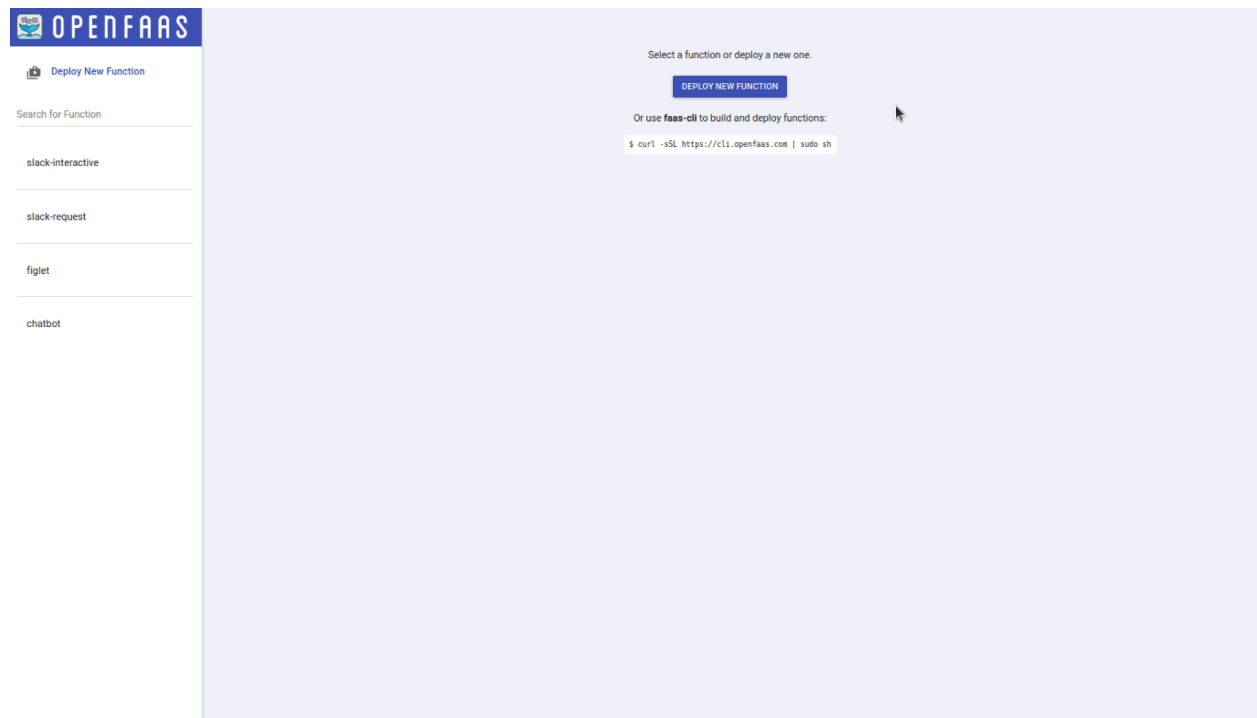
```

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

```
rishabh@rishabh:~/functions/slack-interactive$ cat handler.py
#!/usr/bin/env python3
import json
import urlparse

def handle(req):
    urlstring = urlparse.unquote(req).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": "",
                "author_link": "https://github.com/agrawal-rishabh-nanoj",
                "author_icon": "https://github.com/agrawal-rishabh-nanoj",
                "title": "COEN 241",
                "title_link": "https://www.scu.edu/engineering/academic-programs/department-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/department-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)
```

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



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

slack-request:

OPENFAAS

Deploy New Function

Search for Function

slack-interactive

slack-request

figlet

chatbot

slack-request

Status: Ready, Replicas: 1, Invocation count: 10

Image: docker.io/rmagrawal/slack-request:latest, URL: http://127.0.0.1:8080/function/slack-request

Function process: python index.py

Invoke function

INVOKE

☒ Text ☐ JSON ☐ Download

Request body

Response status: 200, Response time (s): 0.09

Response body

```
{
  "text": "👋👋👋 Message.",
  "attachments": [
    {
      "fields": [
        {
          "short": true,
          "value": "100",
          "title": "Amazing Level"
        }
      ],
      "author_icon": "https://github.com/agrawal-rishabh-manoj.png",
      "image_url": "https://github.com/agrawal-rishabh-manoj.png",
      "author_name": "👋👋👋👋👋👋👋👋👋",
      "title": "The Awesome world of Cloud Computing! 🚀🚀🚀 241"
    },
    {
      "text": "🚀🚀🚀 241 is the most awesome class ever!.",
      "title": "About 🚀🚀🚀 241"
    },
    {
      "text": "Would you recommend 🚀🚀🚀 241 to your friends?",
      "color": "#3A83E3",
      "actions": [
        {
          "text": "Of Course!",
          "type": "button",
          "name": "recommend",
          "value": "recommend"
        },
        {
          "text": "Most Definitely!",
          "type": "button",
          "name": "definitely",
          "value": "definitely"
        }
      ]
    }
  ]
}
```

slack-interactive:

OPENFAAS

Deploy New Function

Search for Function

figlet

slack-interactive

slack-request

slack-interactive

Status: Ready, Replicas: 11, Invocation count: 11

Image: docker.io/rmagrawal/slack-interactive:latest, URL: http://127.0.0.1:8080/function/slack-interactive

Function process: python index.py

Invoke function

INVOKE

☒ Text ☐ JSON ☐ Download

Request body

"hello"

Response status: 200, Response time (s): 0.15

Response body

```
{
  "attachments": [
    {
      "footer": "Slack Apps built on OpenFaaS",
      "author_link": "https://github.com/agrawal-rishabh-manoj",
      "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-amp-photography/visual-identity-toolkit/logos-amp-seals/Mission-Dent3.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://github.com/agrawal-rishabh-manoj"
    }
  ]
}
```

- Complete the chatbot with a yml file (25pt)

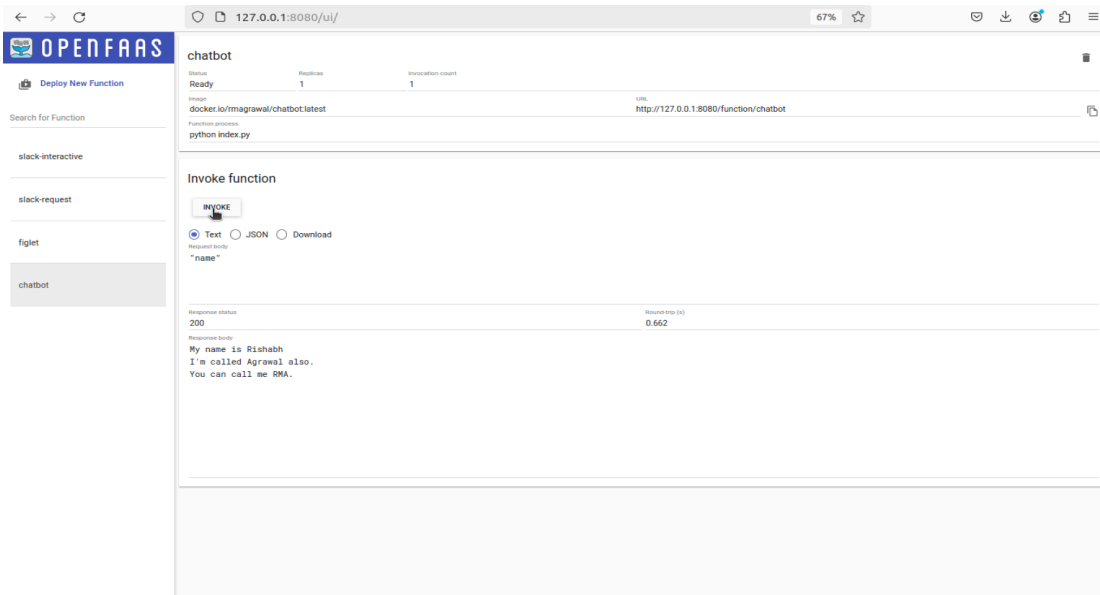
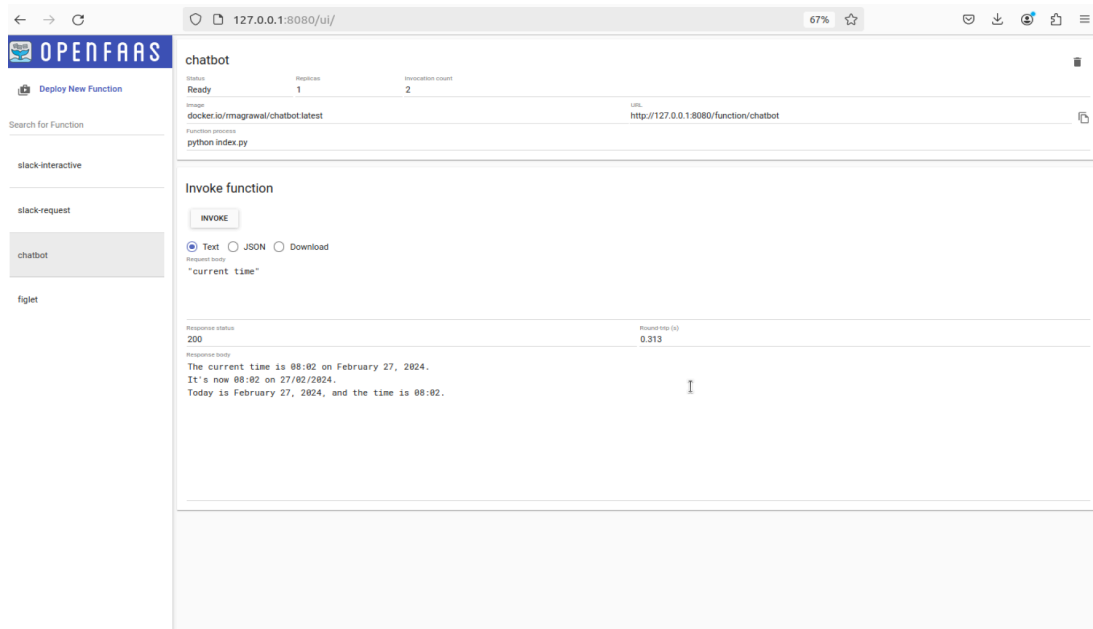
```
shabhb@rishabh:~/chatbot/chatbot$ cat handler.py
import datetime
from pyfiglet import Figlet
import sys

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
        response = [
            "My name is Rishabh",
            "I'm called Agrawal also.",
            "You can call me RMA."
        ]
        return "\n".join(response)
    elif "current time" in req.lower() or "current date" in req.lower():
        # Respond with the current date and time in 3 different ways
        current_time = datetime.datetime.now()
        responses = [
            current_time.strftime("The current time is %H:%M on %B %d, %Y."),
            current_time.strftime("It's now %H:%M on %d/%m/%Y."),
            current_time.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
        extract_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(extract_text)
    else:
        return "I'm not sure how to process that request."

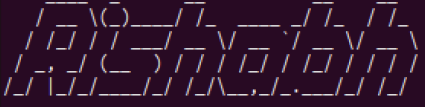
if __name__ == "__main__":
    # For local testing, input can be sent directly through command line
    req = sys.argv[1] if len(sys.argv) > 1 else ""
    print(handle(req))
```

```
rishabh@rishabh:~/chatbot$ cat chatbot.yml
version: 1.0
provider:
  name: openfaas
  gateway: http://127.0.0.1:8080
functions:
  chatbot:
    lang: python
    handler: ./chatbot
    image: rmagrawal/chatbot:latest
```

- Provide a screenshot of invoking three different cases of the chatbot (5 pts)



```
rishabh@rishabh:~/chatbot$ echo "generate a figlet for Rishabh" | faas-cli invoke chatbot
```



Questions:

1. What is the command to invoke the slack-request function?

- via curl:
`curl -d '{"Hello":"COEN 241"}' http://127.0.0.1:8080/function/slack-request`
- via faas-cli:
`sudo faas-cli invoke slack-request`

2. What is the output you see when you invoke the slack-request function?

```
rishabh@rishabh:~/functions$ sudo faas-cli invoke slack-request
Reading from STDIN - hit (Control + D) to stop.
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "https://github.com/agrawal-rishabh-manoj.png", "image_url": "https://github.com/agrawal-rishabh-manoj.png", "author_name": "Rishabh Agrawal", "title": "The Awesome world of Cloud Computing! COEN 241", {"text": "COEN 241 is the most awesome class ever I.", "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"}]}
rishabh@rishabh:~/functions$ curl -d '{"Hello":"COEN 241"}' http://127.0.0.1:8080/function/slack-request
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "https://github.com/agrawal-rishabh-manoj.png", "image_url": "https://github.com/agrawal-rishabh-manoj.png", "author_name": "Rishabh Agrawal", "title": "The Awesome world of Cloud Computing! COEN 241", {"text": "COEN 241 is the most awesome class ever I.", "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"}]}
```

3. What is the command to invoke the slack-interactive function?

- via curl
`curl -d '{"Hello":"COEN 241"}' http://127.0.0.1:8080/function/slack-interactive`
- via faas-cli
`sudo faas-cli invoke slack-interactive`

4. What is the output you see when you invoke the slack-interactive function?

```
rishabh@rishabh:~/functions$ sudo faas-cli invoke slack-interactive
Reading from STDIN - hit (Control + D) to stop.
{"text": "Hi", "attachments": [{"footer": "Slack Apps built on OpenFaas", "author_link": "https://github.com/agrawal-rishabh-manoj", "color": "#36a64f", "text": "Head over to COEN 241", "title": "COEN 241", "ts": 123456789, "author_name": "", "title_link": "https://www.scu.edu/engineering/academic-programs/departments-of-computer-engineering/graduate/course-descriptions/", "image_url": "https://www.scu.edu/media/offices/unc/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 plain-text summary of the attachment.", "thumb_url": "https://www.scu.edu/engineering/academic-programs/departments-of-computer-engineering/graduate/course-descriptions/", "author_icon": "https://github.com/agrawal-rishabh-manoj"}]}
rishabh@rishabh:~/functions$ curl -d '{"Hello":"COEN 241"}' http://127.0.0.1:8080/function/slack-interactive
{"text": "Hi", "attachments": [{"footer": "Slack Apps built on OpenFaas", "author_link": "https://github.com/agrawal-rishabh-manoj", "color": "#36a64f", "text": "Head over to COEN 241", "title": "COEN 241", "ts": 123456789, "author_name": "", "title_link": "https://www.scu.edu/engineering/academic-programs/departments-of-computer-engineering/graduate/course-descriptions/", "image_url": "https://www.scu.edu/media/offices/unc/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 plain-text summary of the attachment.", "thumb_url": "https://www.scu.edu/engineering/academic-programs/departments-of-computer-engineering/graduate/course-descriptions/", "author_icon": "https://github.com/agrawal-rishabh-manoj"}]}
```

5. How would you pass different arguments to the functions?

There are multiple methods for providing arguments to functions, each suitable for different contexts or environments.

a. For invoking the figlet function through the FaaS CLI, one approach involves utilizing the echo command to supply the argument "Hello, FaaS World" directly to the function as follows:

```
echo "Hello, FaaS World" | faas-cli invoke figlet
```

b. When making a Slack request via a curl command, arguments can be sent in a POST request to the server by specifying the data as follows:

```
curl -d '{"Hello":"COEN 241"}' https://127.1.0.1:8080/function/slack-request
```

c. Arguments can also be submitted through the OpenFaaS API Gateway's user interface, providing a graphical method for function interaction.

6. How would you change the slack-interactive function to react to different inputs?

To modify the slack-interactive function to respond to diverse inputs, within the `handler.py` file, we leverage a function from the `urllib` package named `unquote` to decode strings into URLs, defaulting to UTF-8 encoding. Subsequently, the `json.load()` method is employed to convert a valid JSON string, such as `urlstring`, into a Python dictionary, which is then assigned to a variable named `response`. Initially, this `response` variable is not actively used within the function. To enhance the function's capability to react to varying inputs, the `response` dictionary can be integrated into the existing `data` dictionary. By doing so and returning `json.dumps(data)`, the function is then enabled to dynamically respond to a range of inputs.

7. How long does it take for the chat response to come back?

Please refer screenshots below for answer

```
ishabh@rishabh:~/chatbot$ cat chatbot_response_time.py
import requests
import time

SERVER_URL = "http://127.0.0.1:8080/function/chatbot"

def calculate_response_time(data, repeat=1):
    total_time = 0
    for _ in range(repeat):
        start_time = time.time()
        response = requests.post(SERVER_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 = calculate_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 = calculate_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 = calculate_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 = calculate_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 = calculate_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)
    calculate_response_time("What is your name?")
    # Measure the second request (with figlet)
    response_time_f = calculate_response_time("Generate a figlet for Hello")
    print(f"f. Response time for the second request (with figlet, after no figlet): {response_time_f:.4f} seconds")

    # g. Average over 10 requests that do call figlet
    average_response_time_g = calculate_response_time("Generate a figlet for Hello", repeat=10)
    print(f"g. Average response time over 10 requests (with figlet): {average_response_time_g:.4f} seconds")

if __name__ == "__main__":
    main()
```

```
ishabh@rishabh:~/chatbot$ python3 chatbot_response_time.py
a. Response time for the first request (no figlet): 0.2657 seconds
b. Response time for the second request (no figlet): 0.1737 seconds
c. Average response time over 10 requests (no figlet): 0.1550 seconds
d. Response time for the first request (with figlet): 0.1666 seconds
e. Response time for the second request (with figlet): 0.1723 seconds
f. Response time for the second request (with figlet, after no figlet): 0.1774 seconds
g. Average response time over 10 requests (with figlet): 0.1544 seconds
```

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?

Please refer screenshots below for answer

```
rishabh@rishabh:~/chatbot$ python3 parallel_requests.py
Attempted requests: 50
Successful replies: 50
Efficiency: 100.00%
rishabh@rishabh:~/chatbot$ cat parallel_requests.py
import concurrent.futures
import requests
import time

# Endpoint for the chatbot function
CHAT_FUNCTION_URL = "http://127.0.0.1:8080/function/chatbot"

def post_chatbot_data(payload):
    """Post data to the chatbot function."""
    try:
        result = requests.post(CHAT_FUNCTION_URL, data=payload)
        return result.status_code
    except Exception as err:
        return str(err)

def launch_requests(rate_of_requests, test_duration=10):
    """Launch requests concurrently to achieve a certain rate."""
    with concurrent.futures.ThreadPoolExecutor() as pool:
        tasks = []
        start = time.time()

        while time.time() - start < test_duration:
            for _ in range(rate_of_requests):
                # Customize your request payload as necessary
                task = pool.submit(post_chatbot_data, "Can you tell me a joke?")
                tasks.append(task)
            time.sleep(1) # Interval before the next group of requests

        outcomes = [task.result() for task in tasks]

    successful_outcomes = [outcome for outcome in outcomes if outcome == 200]
    print(f"Attempted requests: {len(outcomes)}")
    print(f"Successful replies: {len(successful_outcomes)}")
    print(f"Efficiency: {(len(successful_outcomes) / len(outcomes)) * 100:.2f}%")

# Usage example:
requests_each_second = 5 # Modify this to test different intensities
launch_requests(requests_each_second)
```

Extra Credit:

Creating Incoming webhook

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
https://hooks.slack.com/services/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 Copy	#incoming-webhook	Rishabh Feb 27, 2024 
		Rishabh


Creating new command

Edit Command

Command

/coen241

Request URL

https://6b14-24-23-244-181.ngrok-... 

Short Description

Extra Credit - Rishabh

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"
CSEN241
/coen241 Extra Credit - Rishabh
+ /coen241

Cancel

Save

Setting request url for interactivity

Interactivity & Shortcuts

Interactivity

On

Any interactions with shortcuts, modals, or interactive components (such as buttons, select menus, and datepickers) will be sent to a URL you specify. [Learn more.](#)

Request URL

`https://6b14-24-23-244-181.ngrok-free.app/function/slack-interactive`

Slack will send an HTTP POST request with information to this URL when users interact with a shortcut or interactive component.

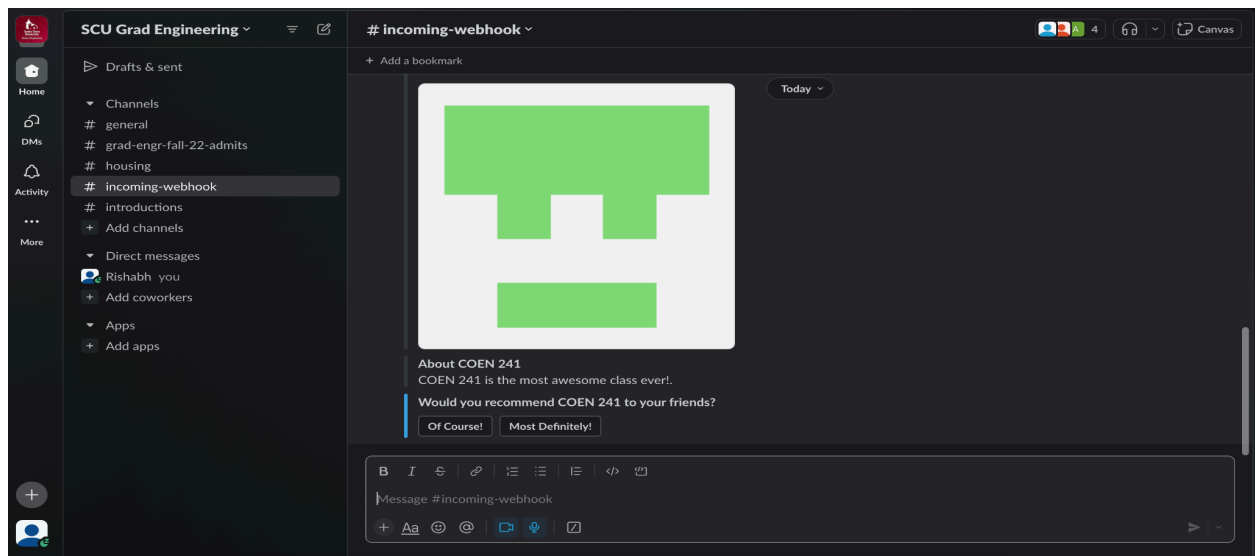
Shortcuts

Let people take actions — like filing a bug or adding a new sales lead — while using your app in Slack. [Learn more.](#)

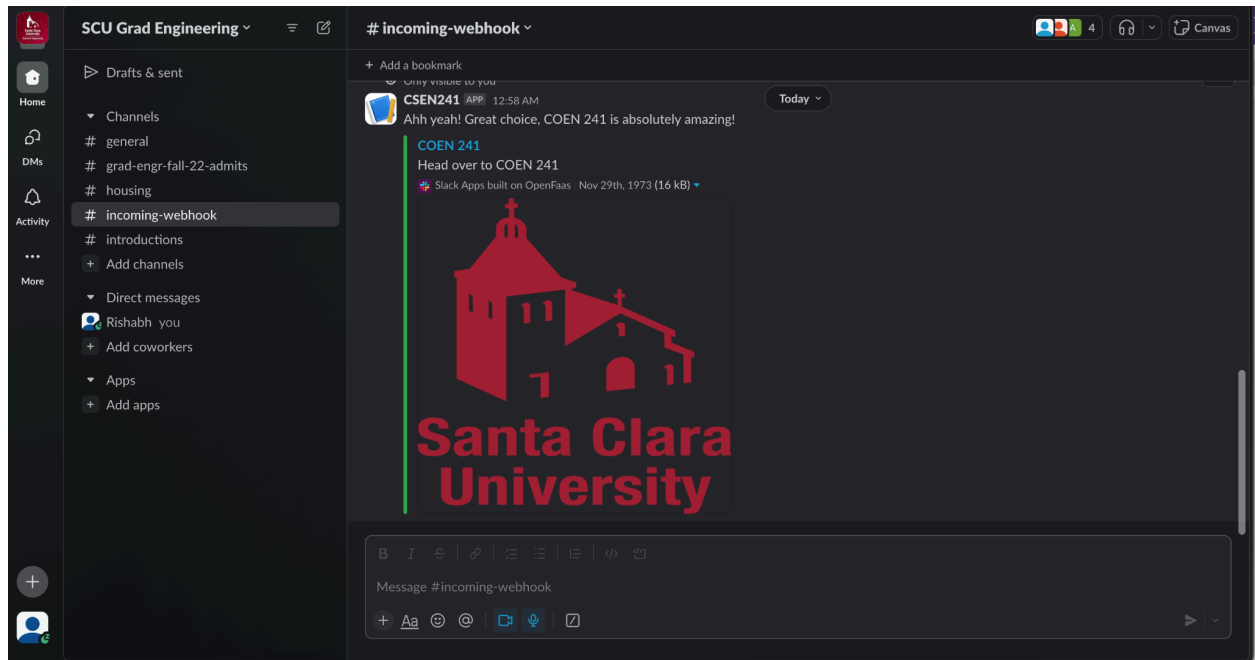
Name	Location	Callback ID
------	----------	-------------

Create New Shortcut

After using slash command in the channel : /coen241



After pressing “Of Course”



ngrok:

```
ngrok
Build better APIs with ngrok. Early access: ngrok.com/early-access

Session Status      online
Account             Rishabh Agrawal (Plan: Free)
Version             3.6.0
Region              United States (California) (us-cal-1)
Latency             22ms
Web Interface       https://127.0.0.1:4040
Forwarding           https://6b14-24-23-244-181.ngrok-free.app -> http://localhost:8080

Connections
  ttl    opn    rt1    rt5    p50    p90
  1       1    0.01   0.00   60.17  60.17

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

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-2dlb3ghf6-pZCdTnornOsI9cIyV42Dwg

Slask command url:

<https://6b14-24-23-244-181.ngrok-free.app/function/slack-request>

<https://6b14-24-23-244-181.ngrok-free.app/function/slack-interactive>

Git Repository Information:

Account name	agrawal-rishabh-manoj
Repository name	COEN-241-Cloud-Computing
Folder which contains HW2	HW2
Link to repository	https://github.com/agrawal-rishabh-manoj/CSEN-241-Cloud-Computing