

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

CSEN 241 HW 2 - Your Own Serverless Infrastructure

1. Tasks –

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

Following are the screenshots for steps involved in invoking the figlet function via command line interface using faas-cli commands.

With the following command we can deploy the figlet function -

```
[avanicloud@avanicloud:~/faasd$ faas-cli store deploy figlet

Deployed. 200 OK.
URL: http://127.0.0.1:8080/function/figlet
```

With the following command you can find the URLs for the figlet function

```
[avanicloud@avanicloud:~/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
```

With the following command you can invoke the figlet function

```
[avanicloud@avanicloud:~/faasd$ echo "Hello, Faas, World" | faas-cli invoke figlet
```



Now, you can also invoke the figlet function via OpenFaas gateway and following is the screenshot for it

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

The screenshot shows the OpenFaaS UI at 127.0.0.1:8000/ui/. On the left sidebar, there are links for 'Deploy New Function', 'Search for Function', 'slack-interactive', 'slack-request', 'slack-chatbot', and 'figlet'. The 'figlet' function is selected. The main panel displays the function details: Status (Ready), Replicas (1), Invocation count (22), Image (ghcr.io/openfaas/figlet:latest), Function process (figlet), and URL (<http://127.0.0.1:8000/function/figlet>). Below this, there's an 'Invoke function' section with an 'INVOKE' button, options for 'Text' (selected), 'JSON', and 'Download', and a request body input field containing 'Open, Faas, World'. The response status is 200, the response body is a large block of ASCII art representing the word 'Open, Faas, World' in various fonts, and the round-trip time is 0.6 seconds.

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

The above journalctl command collects the kernel space and user space logs of the faasd service. Here we display the 40 lines of the logs. Following is the screenshot for the same

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

```
[avanicloud@avanicloud:~/chatbot$ sudo journalctl -u faasd --lines 40
[[sudo] password for avanicloud:
-- Logs begin at Sun 2024-02-25 08:48:43 UTC, end at Mon 2024-02-26 20:52:13 UTC. --
Feb 26 03:34:49 avanicloud faasd[718]: Removing old container for: nats
Feb 26 03:34:49 avanicloud faasd[718]: 2024/02/26 03:34:49 Start-up order:
Feb 26 03:34:49 avanicloud faasd[718]: 2024/02/26 03:34:49 - prometheus
Feb 26 03:34:49 avanicloud faasd[718]: 2024/02/26 03:34:49 - nats
Feb 26 03:34:49 avanicloud faasd[718]: 2024/02/26 03:34:49 - gateway
Feb 26 03:34:49 avanicloud faasd[718]: 2024/02/26 03:34:49 - queue-worker
Feb 26 03:34:49 avanicloud faasd[718]: Starting: prometheus
Feb 26 03:34:49 avanicloud faasd[718]: Creating local directory: /var/lib/faasd/prometheus
Feb 26 03:34:49 avanicloud faasd[718]: 2024/02/26 03:34:49 Running prometheus with user: "65534"
Feb 26 03:34:50 avanicloud faasd[718]: 2024/02/26 03:34:50 Created container: prometheus
Feb 26 03:34:57 avanicloud faasd[718]: 2024/02/26 03:34:57 prometheus has IP: 10.62.0.2
Feb 26 03:34:57 avanicloud faasd[718]: 2024/02/26 03:34:57 Task: prometheus           Container: prometheus
Feb 26 03:34:58 avanicloud faasd[718]: Starting: nats
Feb 26 03:34:58 avanicloud faasd[718]: Creating local directory: /var/lib/faasd/nats
Feb 26 03:34:58 avanicloud faasd[718]: 2024/02/26 03:34:58 Running nats with user: "65534"
Feb 26 03:34:58 avanicloud faasd[718]: 2024/02/26 03:34:58 Created container: nats
Feb 26 03:35:02 avanicloud faasd[718]: 2024/02/26 03:35:02 nats has IP: 10.62.0.3
Feb 26 03:35:02 avanicloud faasd[718]: 2024/02/26 03:35:02 Task: nats           Container: nats
Feb 26 03:35:03 avanicloud faasd[718]: Starting: gateway
Feb 26 03:35:03 avanicloud faasd[718]: 2024/02/26 03:35:03 Created container: gateway
Feb 26 03:35:06 avanicloud faasd[718]: 2024/02/26 03:35:06 gateway has IP: 10.62.0.4
Feb 26 03:35:06 avanicloud faasd[718]: 2024/02/26 03:35:06 Task: gateway          Container: gateway
Feb 26 03:35:07 avanicloud faasd[718]: Starting: queue-worker
Feb 26 03:35:07 avanicloud faasd[718]: 2024/02/26 03:35:07 Created container: queue-worker
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 queue-worker has IP: 10.62.0.5
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Task: queue-worker      Container: queue-worker
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Supervisor init done in: 22 seconds
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Looking up IP for: "gateway"
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Resolver rebuilding map
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Resolver: "localhost"="127.0.0.1"
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Resolver: "faasd-provider"="10.62.0.1"
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Resolver: "prometheus"="10.62.0.2"
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Resolver: "nats"="10.62.0.3"
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Resolver: "gateway"="10.62.0.4"
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Resolver: "queue-worker"="10.62.0.5"
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Looking up IP for: "prometheus"
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Proxy from: 127.0.0.1:9090, to: prometheus:9090 (10.62.0.2)
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Error: listen tcp 127.0.0.1:9090: bind: address already in use
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 faasd: waiting for SIGTERM or SIGINT
Feb 26 03:35:10 avanicloud faasd[718]: 2024/02/26 03:35:10 Proxy from: 0.0.0.0:8080, to: gateway:8080 (10.62.0.4)
```

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

Following is the screenshot for slack-request/handler.py

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

```
[avanicloud@avanicloud:~/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": "Avani Vaidya",
                "author_icon": "https://avatars.githubusercontent.com/u/52353707?v=4",
                "image_url": "https://hub.docker.com/repository/docker/avaidya2/slack-request/general"
            },
            {
                "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)

Following is the screenshot for slack-interactive/handler.py

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

```
[avanicloud@avanicloud:~/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": "Avani Vaidya",
                "author_link": "https://github.com/AvaniVaidya/cloud-computing",
                "author_icon": "https://avatars.githubusercontent.com/u/52353707?v=4",
                "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-amp-photography/visual-identity-toolkit/logos-amp-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)
```

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

Following is the screenshot of OpenFaas gateway after deploying figlet, slack-handler and slack-interactive functions

The screenshot shows the OpenFaas UI at 127.0.0.1:8000/ui/. On the left sidebar, there are three listed functions: figlet, slack-request, and slack-interactive. The figlet function is currently selected. The main panel displays the figlet function details:

- Status: Ready
- Replicas: 1
- Invocation count: 22
- Image: gcr.io/openfaas/figlet:latest
- Function process: figlet
- URL: http://127.0.0.1:8000/function/figlet

Below this, there is an "Invoke function" section with an "INVOKE" button. Under "Request body", the text "Open, Faas, World" is entered. The "Text" radio button is selected. The response status is 200, and the response body contains the output of the figlet command, which is a large block of ASCII art representing the text "Open, Faas, World".

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

Following is the screenshot of invoking the slack-request function via OpenFaas gateway.

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

The screenshot shows the OpenFaaS UI at <http://127.0.0.1:8000/ui/>. The sidebar on the left lists functions: Deploy New Function, Search for Function, slack-interactive, slack-chatbot, figlet, and slack-request (which is selected). The main panel displays the 'slack-request' function details:

- Status: Ready
- Replicas: 1
- Image: docker.io/avaidya2/slack-request:latest
- Function process: python index.py
- URL: http://127.0.0.1:8000/function/slack-request

The 'Invoke function' section shows an 'INVOKE' button. Below it, under 'Request body', is the text "Avani". The response status is 200, and the response body is a JSON object representing a Slack message payload:

```
{
  "text": "Serverless Message",
  "attachments": [
    {
      "fields": [
        {
          "short": true,
          "value": "100",
          "title": "Amazing Level"
        }
      ],
      "author_icon": "https://avatars.githubusercontent.com/u/52353707?v=4",
      "image_url": "https://hub.docker.com/repository/docker/avaidya2/slack-request/general",
      "author_name": "Avani Vaidya",
      "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"
    }
  ]
}
```

Following is the screenshot of invoking the slack-interactive function via OpenFaaS gateway.

The screenshot shows the OpenFaaS UI at <http://127.0.0.1:8000/ui/>. The sidebar on the left lists functions: Deploy New Function, Search for Function, slack-interactive (which is selected), slack-chatbot, figlet, and slack-request. The main panel displays the 'slack-interactive' function details:

- Status: Not ready
- Replicas: 0
- Image: docker.io/avaidya2/slack-interactive:latest
- Function process: python index.py
- URL: http://127.0.0.1:8000/function/slack-interactive

The 'Invoke function' section shows an 'INVOKE' button. Below it, under 'Request body', is the JSON object {"Text": "Avani"}. The response status is 200, and the response body is a JSON object representing a Slack message payload:

```
{
  "attachments": [
    {
      "footer": "Slack Apps built on OpenFaaS",
      "author_link": "https://github.com/AvaniVaidya/cloud-computing",
      "color": "#3AA3E3",
      "text": "Head over to COEN 241",
      "title": "COEN 241",
      "ts": 1523537078,
      "author_name": "Avani Vaidya",
      "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,
      "pretext": "slack-edge@127.0.0.1:3000/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/52353707?v=4"
    }
  ]
}
```

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

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

Following is the screenshot for chatbot/slack-chatbot/handler.py file.

```
[avanicloud@avanicloud:~/chatbot/slack-chatbot$ cat handler.py
import datetime
import random
from urllib2 import urlopen
from urllib import urlencode, unquote

name_responses = ["Avani is my name", "My name is Avani's chatbot", "You can call me by the name Avani"]
currentdt = datetime.datetime.now()
date_time_responses = ["Date-time now is " + currentdt.strftime("%m-%d-%Y:%H:%M:%S"), "Currently the date and time at your location is " + currentdt.strftime("%m-%d-%Y:%H:%M:%S"), "Today's date and time right now is " + currentdt.strftime("%m-%d-%Y:%H:%M:%S")]
random_ind = random.randint(0,2)

def invoke_figlet(text):
    figletUrl = "http://10.62.0.1:8080/function/figlet"
    try:
        post_data = text.split("for", 1)[1].strip()
        fig_response = urlopen(figletUrl, post_data)
        return fig_response.read()
    except Exception as e:
        return None

def handle(req):
    user_input = req.strip().lower()
    if "name" in user_input:
        response = random.choice(name_responses)
        return response
    elif "time" in user_input or "date" in user_input:
        response = random.choice(date_time_responses)
        return response
    elif "figlet" in user_input:
        figlet_response = invoke_figlet(req.strip())
        if figlet_response:
            response = figlet_response
            return response
        else:
            response = "Cannot generate figlet"
            return response
    else:
        response = "Something went wrong"
        return response
else:
    return "Something went wrong"
```

Following is the screenshot for chatbot/slack-chatbot.yml file.

```
[avanicloud@avanicloud:~/chatbot$ cat slack-chatbot.yml
version: 1.0
provider:
  name: openfaas
  gateway: http://127.0.0.1:8080
functions:
  slack-chatbot:
    lang: python
    handler: ./slack-chatbot
    image: avaidya2/slack-chatbot
    environment:
      content_type: text/plain
```

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

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

Following are the screenshots for 3 different responses received for chatbot's name -

The screenshot shows the OPENFAAS UI at <http://127.0.0.1:8000/ui/>. The sidebar lists functions: slack-chatbot (selected), figlet, slack-interactive, and slack-request. The main panel displays the 'slack-chatbot' function details: Status Ready, Replicas 37, Image docker.io/avaidya2/slack-chatbot:latest, Function process python index.py, and URL <http://127.0.0.1:8000/function/slack-chatbot>. Below this, the 'Invoke function' section shows an 'INVOKE' button, request body 'Give me your name', response status 200, and response body 'You can call me by the name Avani'. A note indicates a round-trip of 6.436 seconds.

The screenshot shows the OPENFAAS UI at <http://127.0.0.1:8000/ui/>. The sidebar lists functions: slack-chatbot (selected), figlet, slack-interactive, and slack-request. The main panel displays the 'slack-chatbot' function details: Status Ready, Replicas 39, Image docker.io/avaidya2/slack-chatbot:latest, Function process python index.py, and URL <http://127.0.0.1:8000/function/slack-chatbot>. Below this, the 'Invoke function' section shows an 'INVOKE' button, request body 'Give me your name', response status 200, and response body 'Avani is my name'. A note indicates a round-trip of 5.951 seconds.

The screenshot shows the OPENFAAS UI at <http://127.0.0.1:8000/ui/>. The sidebar lists functions: slack-chatbot (selected), figlet, slack-interactive, and slack-request. The main panel displays the 'slack-chatbot' function details: Status Ready, Replicas 40, Image docker.io/avaidya2/slack-chatbot:latest, Function process python index.py, and URL <http://127.0.0.1:8000/function/slack-chatbot>. Below this, the 'Invoke function' section shows an 'INVOKE' button, request body 'Give me your name', response status 200, and response body 'My name is Avani's chatbot'. A note indicates a round-trip of 6.017 seconds.

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

Following are the screenshots for 3 different responses received for current date and time-

The screenshot shows the OpenFaaS UI at <http://127.0.0.1:8000/ui/>. The main panel displays the "slack-chatbot" function. It shows a status of "Ready", 41 replicas, and an invocation count of 41. The image is "docker.io/avaidya2/slack-chatbot:latest" and the function process is "python index.py". The URL is <http://127.0.0.1:8000/function/slack-chatbot>. In the "Invoke function" section, the "INVOKE" button is highlighted. A request body "Give me current time" is entered. The response status is 200, and the response body is "Currently the date and time at your location is 02-26-2024:22:08:04". Round trip is 5.804ms.

The screenshot shows the OpenFaaS UI at <http://127.0.0.1:8000/ui/>. The main panel displays the "slack-chatbot" function. It shows a status of "Ready", 43 replicas, and an invocation count of 43. The image is "docker.io/avaidya2/slack-chatbot:latest" and the function process is "python index.py". The URL is <http://127.0.0.1:8000/function/slack-chatbot>. In the "Invoke function" section, the "INVOKE" button is highlighted. A request body "Give me todays date" is entered. The response status is 200, and the response body is "Date-time now is 02-26-2024:22:08:50". Round trip is 5.696ms.

The screenshot shows the OpenFaaS UI at <http://127.0.0.1:8000/ui/>. The main panel displays the "slack-chatbot" function. It shows a status of "Ready", 47 replicas, and an invocation count of 47. The image is "docker.io/avaidya2/slack-chatbot:latest" and the function process is "python index.py". The URL is <http://127.0.0.1:8000/function/slack-chatbot>. In the "Invoke function" section, the "INVOKE" button is highlighted. A request body "Give me date and time for now" is entered. The response status is 200, and the response body is "Today's date and time right now is 02-26-2024:22:09:40". Round trip is 5.573ms.

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

Following is the screenshot for response received for generating figlet for given string –

The screenshot shows the OpenFaaS UI interface. On the left, there's a sidebar with icons for Deploy New Function, Search for Function, and several function names: slack-chatbot, figlet, slack-interactive, and slack-request. The main area displays the 'slack-chatbot' function details:

- Status: Not ready
- Replicas: 52
- Invocation count: 52
- Image: docker.io/avaidya2/slack-chatbot:latest
- Function process: python index.py
- URL: http://127.0.0.1:8000/function/slack-chatbot

Below this, there's an 'Invoke function' section with an 'INVOKE' button. It includes options for Text (selected), JSON, and Download, and a Request body input field containing "Generate figlet for Open,Faas,World".

At the bottom, the response status is 200, and the response body is a large, multi-line string of ASCII art representing the text "Open,Faas,World" in a stylized font.

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

2. Questions –

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

a. Via Curl

Ans – Following is the command to invoke slack-request via curl

```
[avanicloud@avanicloud:~/chatbot/slack-chatbot$ curl -d '{"Hello":"COEN241"}' http://127.0.0.1:8080/function/slack-request ]
```

b. Via faas-cli

Ans – Following is the command to invoke slack-request via faas-cli

```
[avanicloud@avanicloud:~/chatbot/slack-chatbot$ echo '{"Hello":"COEN241"}' | faas-cli invoke slack-request ]
```

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

Ans – Following is the output seen for invoking slack-request via curl and faas-cli respectively

```
[avanicloud@avanicloud:~/chatbot/slack-chatbot$ curl -d '{"Hello":"COEN241"}' http://127.0.0.1:8080/function/slack-request ] {"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "https://avatars.githubusercontent.com/u/52353707?v=4", "image_url": "https://hub.docker.com/repository/docker/avaidya2/slack-request/general", "author_name": "Avani Vaidya", "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"}] [avanicloud@avanicloud:~/chatbot/slack-chatbot$ echo '{"Hello":"COEN241"}' | faas-cli invoke slack-request ] {"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "https://avatars.githubusercontent.com/u/52353707?v=4", "image_url": "https://hub.docker.com/repository/docker/avaidya2/slack-request/general", "author_name": "Avani Vaidya", "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"}]}
```

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

a. Via curl

Ans – Following is the command to invoke slack-interactive via curl

```
[avanicloud@avanicloud:~/chatbot/slack-chatbot$ curl -d '{"Hello":"COEN241"}' http://127.0.0.1:8080/function/slack-interactive ]
```

b. Via faas-cli

```
[avanicloud@avanicloud:~/chatbot/slack-chatbot$ echo '{"Hello":"COEN241"}' | faas-cli invoke slack-interactive ]
```

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

Ans - Following is the output seen for invoking slack-interactive via curl and faas-cli respectively

Student Name – Avani Sanjay Vaidya

Student Id - 07700005517

```
[avanicloud@avanicloud:~/chatbot/slack-chatbot$ curl -d '{"Hello":"COEN241"}' http://127.0.0.1:8080/function/slack-interactive
{"attachments": [{"footer": "Slack Apps built on OpenFaaS", "author_link": "https://github.com/AvaniVaidya/cloud-computing", "color": "#36a64f", "text": "Head over to COEN 241", "title": "COEN 241", "ts": 123456789, "author_name": "Avani Vaidya", "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/52353707?v=4"}]}
[avanicloud@avanicloud:~/chatbot/slack-chatbot$ echo '{"Hello":"COEN241"}' | faas-cli invoke slack-interactive
{"attachments": [{"footer": "Slack Apps built on OpenFaaS", "author_link": "https://github.com/AvaniVaidya/cloud-computing", "color": "#36a64f", "text": "Head over to COEN 241", "title": "COEN 241", "ts": 123456789, "author_name": "Avani Vaidya", "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/52353707?v=4"}]}
```

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

Ans - There are many ways of passing arguments to functions.

a. The way we invoke the figlet function using faas-cli along with echo command and we pass the argument "Hello, FaaS World" to the function:
echo "Hello, FaaS World" | faas-cli invoke figlet

b. In the slack request curl command below, we send the specified data in a POST request to the HTTP server:

```
curl -d '{"Hello":"COEN 241"}' https://127.0.0.1:8080/function/slack-request
```

c. We can also pass arguments via the OpenFaas API Gateway UI

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

Ans – In handler.py, we have defined an unquote function which is part of a package called urllib (slack-interactive function) which can be used to decode any string into a URL. The default encoding is UTF-8. After that, we utilize the json.load() method to transform a valid JSON string, such as urlstring, into a Python dictionary whose information is saved in the response. After storing it in the response, we see it is not getting used anywhere. We can update the slack-interactive function to react to different inputs. Response can be utilized inside the data because it is also a Python dictionary. As a result, when we return json.dumps(data), the function will begin to react to various inputs.

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

Ans – To calculate the response time taken by different scenarios below, we can write a small python code, which will use the time library. We will record the start time, then make a post request to the chatbot url and then record the end time.

Thus, response time = (end time – start time).

Following is the snap of the python code to calculate the response time.

```
for _ in range(repeat):
    start_time = time.time()
    response = requests.post(self.chatbot_url, data=data)
    end_time = time.time()
    total_time += (end_time - start_time)
    if repeat == 1:
        return end_time - start_time
return total_time / repeat
```

Student Name – Avani Sanjay Vaidya

Student Id - 07700005517

The repeat parameter is used to calculate the average time for 10 requests.

Thus, running the script we will get different response as follows –

a. For the first request that does not call figlet

Ans – It takes about 0.3164 seconds for first request that does not call figlet.

b. For the second request that does not call figlet

Ans – Similarly, it takes about 0.1415s for second request that does not call figlet

c. Average over 10 requests that do not call figlet

Ans – Considering 10 such requests that do not call figlet the average time is – 0.1818s

d. For the first request that calls figlet

Ans – It takes about 0.1762s for first request that calls figlet.

e. For the second request that calls figlet

Ans – It takes about 0.2160s for second request that calls figlet.

f. For the second request that calls figlet that follows the first request that does not call figlet.

Ans – For the second request that calls figlet that follows the first request that does not call figlet, it takes about 0.2404s.

g. Average over 10 requests that do call figlet

Ans – Considering 10 requests that calls figlet the average time is – 0.2147s

Following is the screenshot for responses received.

```
[avanicloud@avanicloud:~/chatbot$ python3 metrics.py
Response time for the first request using no figlet call: 0.3164 seconds
Response time for the second request using no figlet call: 0.1415 seconds
Average response time over 10 requests using no figlet call: 0.1818 seconds
Response time for the first request using with figlet call: 0.1762 seconds
Response time for the second request using with figlet call: 0.2160 seconds
Response time for the second request using with figlet and after without figlet call: 0.2404 seconds
Average response time over 10 requests using figlet call: 0.2147 seconds
avanicloud@avanicloud:~/chatbot$ ]
```

Student Name – Avani Sanjay Vaidya
Student Id - 07700005517

3. Extra Credit

Link to application - <https://app.slack.com/client/T06LLP219K8/D06MGBLL8GY>

Sharable URL for the application -

https://slack.com/oauth/v2/authorize?client_id=6700784043654.6720088628497&scope=commands&user_scope=

Link to join Slack chatbot workspace -

https://join.slack.com/t/cloudcomputingavani/shared_invite/zt-2dk07vcat-Rp5Toy97S2JdRJgELS~zkQ

Slash command - /coen241

Slack Application Bot –

