

Name: Juhi Checker
SCU ID: 1605378

Q1. Provide a screenshot of invoking the figlet function

- A.** Below is a screenshot of invoking the figlet function using the faas-cli invoke command. Before invoking the command the figlet function needs to be deployed using faas-cli store deploy function.

[illegible]

Q2 & Q3. Complete slack-request/handler.py and slack-interactive/handler.py.

- A. Please find the changed code in the Github repository.
Github repository: <https://github.com/checkerjuhi21/CloudComputing/tree/main/HW2>

Q4. Provide a screenshot of running the following command

```
sudo journalctl -u faasd -lines 40
```

- A. This command shows the logs of faasd upto 40 lines.

```
jchecker@jchecker:~/functions$ sudo journalctl -u faasd --lines 40
-- Logs begin at Fri 2022-10-14 15:09:55 UTC, end at Thu 2022-11-03 20:23:30 UTC. --
Nov 03 20:20:41 jchecker faasd[1018]: 2022/11/03 20:20:41 - gateway
Nov 03 20:20:41 jchecker faasd[1018]: 2022/11/03 20:20:41 - queue-worker
Nov 03 20:20:41 jchecker faasd[1018]: Starting: basic-auth-plugin
Nov 03 20:20:43 jchecker faasd[1018]: 2022/11/03 20:20:43 Created container: basic-auth-plugin
Nov 03 20:21:05 jchecker faasd[1018]: 2022/11/03 20:21:05 basic-auth-plugin has IP: 10.62.0.2
Nov 03 20:21:05 jchecker faasd[1018]: 2022/11/03 20:21:05 Task: basic-auth-plugin Container: basic-auth-plugin
Nov 03 20:21:06 jchecker faasd[1018]: Starting: nats
Nov 03 20:21:06 jchecker faasd[1018]: Creating local directory: /var/lib/faasd/nats
Nov 03 20:21:06 jchecker faasd[1018]: 2022/11/03 20:21:06 Running nats with user: "65534"
Nov 03 20:21:08 jchecker faasd[1018]: 2022/11/03 20:21:08 Created container: nats
Nov 03 20:21:18 jchecker faasd[1018]: 2022/11/03 20:21:18 nats has IP: 10.62.0.3
Nov 03 20:21:18 jchecker faasd[1018]: 2022/11/03 20:21:18 Task: nats Container: nats
Nov 03 20:21:19 jchecker faasd[1018]: Starting: prometheus
Nov 03 20:21:19 jchecker faasd[1018]: Creating local directory: /var/lib/faasd/prometheus
Nov 03 20:21:19 jchecker faasd[1018]: 2022/11/03 20:21:19 Running prometheus with user: "65534"
Nov 03 20:21:21 jchecker faasd[1018]: 2022/11/03 20:21:21 Created container: prometheus
Nov 03 20:21:29 jchecker faasd[1018]: 2022/11/03 20:21:29 prometheus has IP: 10.62.0.4
Nov 03 20:21:29 jchecker faasd[1018]: 2022/11/03 20:21:29 Task: prometheus Container: prometheus
Nov 03 20:21:29 jchecker faasd[1018]: Starting: gateway
Nov 03 20:21:31 jchecker faasd[1018]: 2022/11/03 20:21:31 Created container: gateway
Nov 03 20:21:42 jchecker faasd[1018]: 2022/11/03 20:21:42 gateway has IP: 10.62.0.5
Nov 03 20:21:42 jchecker faasd[1018]: 2022/11/03 20:21:42 Task: gateway Container: gateway
Nov 03 20:21:45 jchecker faasd[1018]: Starting: queue-worker
Nov 03 20:21:46 jchecker faasd[1018]: 2022/11/03 20:21:46 Created container: queue-worker
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 queue-worker has IP: 10.62.0.6
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Task: queue-worker Container: queue-worker
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Supervisor init done in: About a minute
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Looking up IP for: "gateway"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver rebuilding map
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Looking up IP for: "prometheus"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "localhost"="127.0.0.1"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "faasd-provider"="10.62.0.1"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "basic-auth-plugin"="10.62.0.2"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "nats"="10.62.0.3"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "prometheus"="10.62.0.4"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "gateway"="10.62.0.5"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "queue-worker"="10.62.0.6"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Proxy from: 127.0.0.1:9090, to: prometheus:9090 (10.62.0.4)
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 faasd: waiting for SIGTERM or SIGINT
lines 1-40...skipping...
-- Logs begin at Fri 2022-10-14 15:09:55 UTC, end at Thu 2022-11-03 20:23:30 UTC. --
Nov 03 20:20:41 jchecker faasd[1018]: 2022/11/03 20:20:41 - gateway
Nov 03 20:20:41 jchecker faasd[1018]: 2022/11/03 20:20:41 - queue-worker
Nov 03 20:20:41 jchecker faasd[1018]: Starting: basic-auth-plugin
Nov 03 20:20:43 jchecker faasd[1018]: 2022/11/03 20:20:43 Created container: basic-auth-plugin
Nov 03 20:21:05 jchecker faasd[1018]: 2022/11/03 20:21:05 basic-auth-plugin has IP: 10.62.0.2
Nov 03 20:21:05 jchecker faasd[1018]: 2022/11/03 20:21:05 Task: basic-auth-plugin Container: basic-auth-plugin
Nov 03 20:21:06 jchecker faasd[1018]: Starting: nats
Nov 03 20:21:06 jchecker faasd[1018]: Creating local directory: /var/lib/faasd/nats
```

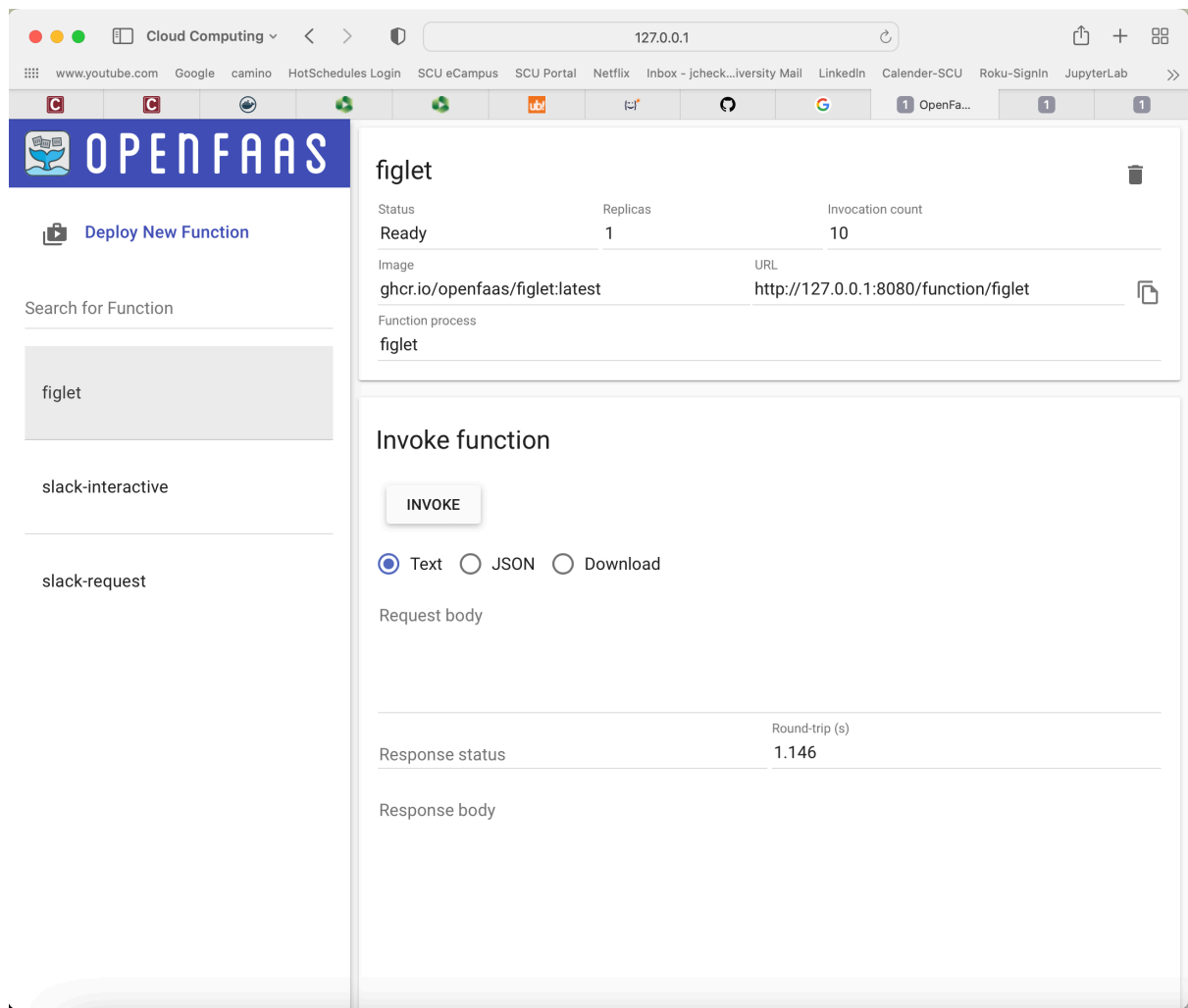
```

lines 1-40...skipping...
-- Logs begin at Fri 2022-10-14 15:09:55 UTC, end at Thu 2022-11-03 20:23:30 UTC. --
Nov 03 20:20:41 jchecker faasd[1018]: 2022/11/03 20:20:41 - gateway
Nov 03 20:20:41 jchecker faasd[1018]: 2022/11/03 20:20:41 - queue-worker
Nov 03 20:20:41 jchecker faasd[1018]: Starting: basic-auth-plugin
Nov 03 20:20:43 jchecker faasd[1018]: 2022/11/03 20:20:43 Created container: basic-auth-plugin
Nov 03 20:21:05 jchecker faasd[1018]: 2022/11/03 20:21:05 basic-auth-plugin has IP: 10.62.0.2
Nov 03 20:21:05 jchecker faasd[1018]: 2022/11/03 20:21:05 Task: basic-auth-plugin Container: basic-auth-plugin
Nov 03 20:21:06 jchecker faasd[1018]: Starting: nats
Nov 03 20:21:06 jchecker faasd[1018]: Creating local directory: /var/lib/faasd/nats
Nov 03 20:21:06 jchecker faasd[1018]: 2022/11/03 20:21:06 Running nats with user: "65534"
Nov 03 20:21:08 jchecker faasd[1018]: 2022/11/03 20:21:08 Created container: nats
Nov 03 20:21:18 jchecker faasd[1018]: 2022/11/03 20:21:18 nats has IP: 10.62.0.3
Nov 03 20:21:18 jchecker faasd[1018]: 2022/11/03 20:21:18 Task: nats Container: nats
Nov 03 20:21:19 jchecker faasd[1018]: Starting: prometheus
Nov 03 20:21:19 jchecker faasd[1018]: Creating local directory: /var/lib/faasd/prometheus
Nov 03 20:21:19 jchecker faasd[1018]: 2022/11/03 20:21:19 Running prometheus with user: "65534"
Nov 03 20:21:21 jchecker faasd[1018]: 2022/11/03 20:21:21 Created container: prometheus
Nov 03 20:21:29 jchecker faasd[1018]: 2022/11/03 20:21:29 prometheus has IP: 10.62.0.4
Nov 03 20:21:29 jchecker faasd[1018]: 2022/11/03 20:21:29 Task: prometheus Container: prometheus
Nov 03 20:21:29 jchecker faasd[1018]: Starting: gateway
Nov 03 20:21:31 jchecker faasd[1018]: 2022/11/03 20:21:31 Created container: gateway
Nov 03 20:21:42 jchecker faasd[1018]: 2022/11/03 20:21:42 gateway has IP: 10.62.0.5
Nov 03 20:21:42 jchecker faasd[1018]: 2022/11/03 20:21:42 Task: gateway Container: gateway
Nov 03 20:21:45 jchecker faasd[1018]: Starting: queue-worker
Nov 03 20:21:46 jchecker faasd[1018]: 2022/11/03 20:21:46 Created container: queue-worker
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 queue-worker has IP: 10.62.0.6
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Task: queue-worker Container: queue-worker
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Supervisor init done in: About a minute
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Looking up IP for: "gateway"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver rebuilding map
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Looking up IP for: "prometheus"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "localhost"="127.0.0.1"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "faasd-provider"="10.62.0.1"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "basic-auth-plugin"="10.62.0.2"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "nats"="10.62.0.3"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "prometheus"="10.62.0.4"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "gateway"="10.62.0.5"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Resolver: "queue-worker"="10.62.0.6"
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Proxy from: 127.0.0.1:9090, to: prometheus:9090 (10.62.0.4)
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 faasd: waiting for SIGTERM or SIGINT
Nov 03 20:21:55 jchecker faasd[1018]: 2022/11/03 20:21:55 Proxy from: 0.0.0.0:8080, to: gateway:8080 (10.62.0.5)
~
jchecker@jchecker:~/functions$

```

Q5. Provide a screenshot of your OpenFaaS gateway AFTER deploying figlet, slack-handler and slack-interactive functions

- A. Deploying the figlet, slack-react and slack-interactive functions and checking if they are deployed using the OpenFaaS UI.




Faas-cli list also shows the deployed functions


```
jchecker@jchecker:~/functions$ faas-cli list
Function              Invocations    Replicas
figlet                 10             1
slack-interactive      9              1
slack-request          9              1
jchecker@jchecker:~/functions$
```

Q6. Provide a screenshot of invoking slack-request and slack-interactive functions

A. OpenFaaS gateway after invoking slack-react handler

On local host:


OPENFAAS

 **Deploy New Function**

Search for Function

slack-interactive

slack-request

figlet

slack-request

Status

Ready

Replicas

1

Invocation count

8

Image

docker.io/jchecker/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

```
{ "Name": "Juhi Checker" }
```

Response status

200

Round-trip (s)

2.324

Response body

```
{
  "text": "Serverless Message",
  "attachments": [
    {
      "fields": [
        {
          "short": true,
          "value": "100"
        }
      ]
    }
  ]
}
```

On terminal using curl:


```
jchecker@jchecker:~/functions$ curl -d '{"Demo1": "Trying Demo 1 to check if it works!"}' http://127.0.0.1:8080/function/slack-request
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "", "image_url": "https://avatars.githubusercontent.com/u/36622318?v=4", "author_name": "checkerjuhi21", "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"}]}
```


On terminal using invoke:

```
jchecker@jchecker:~/functions$ sudo faas-cli invoke slack-request
Reading from STDIN - hit (Control + D) to stop.
Demo using faas-cli
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "", "image_url": "https://avatars.githubusercontent.com/u/36622318?v=4", "author_name": "checkerjuhi21", "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"}]}
```

OpenFaaS gateway after invoking slack-interactive

On local host:


OPENFAAS


[Deploy New Function](#)

slack-interactive

figlet

slack-request

slack-interactive

Status	Replicas	Invocation count
Ready	1	8

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

Function process
python index.py

Invoke function

INVOKE

☐ Text
 ☒ JSON
 ☐ Download

Request body

{ "Name": "Juhi Checker" }

Response status	Round-trip (s)
200	2.563

Response body

```
{
  "attachments": [
    {
      "footer": "Slack Apps built on OpenFaas",
      "author_link": "https://github.com/checkerjuhi21/CloudComputing",
      "color": "#36a64f",
      "text": "Head over to COEN 241",
      "title": "COEN 241",

```

On terminal using curl:

```
jchecker@jchecker:~/functions$ curl -d '{"Demo1": "Trying Demo 1 to check if it works!"}' http://127.0.0.1:8080/function/slack-interactive
{"attachments": [{"footer": "Slack Apps built on OpenFaas", "author_link": "https://github.com/checkerjuhi21/CloudComputing", "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-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/36622318?v=4"}]}
jchecker@jchecker:~/functions$ curl -d '{"Demo1": "Trying Demo 1 to check if it works!"}' http://127.0.0.1:8080/function/slack-interactive
```

On terminal using invoke:

```
jchecker@jchecker:~/functions$ sudo faas-cli invoke slack-interactive
Reading from STDIN - hit (Control + D) to stop.
{"Demo2": "using invoke"}
{"attachments": [{"footer": "Slack Apps built on OpenFaas", "author_link": "https://github.com/checkerjuhi21/CloudComputing", "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-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/36622318?v=4"}]}
jchecker@jchecker:~/functions$
```

Q7. What is the command to invoke the slack-request function

A.

Via Curl

```
curl -d '{"Demo1": "Trying Demo 1 to check if it works!"}'  
http://127.0.0.1:8080/function/slack-request
```

Via faas-cli

```
sudo faas-cli invoke slack-request
```

Q8. What is the output you see when you invoke the slack-request function?

A.

Via curl

The output we see is the json file that we had created inside the handler.py file. On invoking the function we see the output as json format with my changes.

```
jchecker@jchecker:~/functions$ curl -d '{"Demo1": "Trying Demo 1 to check if it works!"}' http://127.0.0.1:8080/function/slack-request  
{  
  "text": "Serverless Message",  
  "attachments": [  
    {  
      "fields": [  
        {  
          "short": true,  
          "value": "100",  
          "title": "Amazing Level"  
        }  
      ],  
      "author_icon": "",  
      "image_url": "https://avatars.githubusercontent.com/u/36622318?v=4",  
      "author_name": "checkerjuhi21",  
      "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"  
    }  
  ]  
}
```

Via faas-cli

The output we see is the json file that we had created inside the handler.py file. On invoking the function we see the output as json format with my changes.

```
jchecker@jchecker:~/functions$ sudo faas-cli invoke slack-request  
Reading from STDIN - hit (Control + D) to stop.  
Demo using faas-cli  
{  
  "text": "Serverless Message",  
  "attachments": [  
    {  
      "fields": [  
        {  
          "short": true,  
          "value": "100",  
          "title": "Amazing Level"  
        }  
      ],  
      "author_icon": "",  
      "image_url": "https://avatars.githubusercontent.com/u/36622318?v=4",  
      "author_name": "checkerjuhi21",  
      "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"  
    }  
  ]  
}
```

Q9. What is the command to invoke the slack-interactive function?

A.

Via curl

```
curl -d '{"Demo1": "Trying Demo 1 to check if it works!"}'  
http://127.0.0.1:8080/function/slack-interactive
```

Via faas-cli

```
sudo faas-cli invoke slack-interactive
```

Q10. What is the output you see when you invoke the slack-interactive function?

A.

Via curl:

The output we see is the json file that we had created inside the handler.py file. On invoking the function we see the output as json format with my changes. Here the input has to be given in json format.


```
jchecker@jchecker:~/functions$ curl -d '{"Demo1": "Trying Demo 1 to check if it works!"}' http://127.0.0.1:8080/function/
/slack-interactive
{"attachments": [{"footer": "Slack Apps built on OpenFaaS", "author_link": "https://github.com/checkerjuhi21/CloudComputing", "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-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/36622318?v=4"}]}
jchecker@jchecker:~/functions$ curl -d '{"Demo1": "Trying Demo 1 to check if it works!"}' http://127.0.0.1:8080/function/
```

Via faas-cli:

The output we see is the json file that we had created inside the handler.py file. On invoking the function we see the output as json format with my changes. Here the input has to be given in json format.

```
jchecker@jchecker:~/functions$ sudo faas-cli invoke slack-interactive
Reading from STDIN - hit (Control + D) to stop.
{"Demo2": "using invoke"}
{"attachments": [{"footer": "Slack Apps built on OpenFaaS", "author_link": "https://github.com/checkerjuhi21/CloudComputing", "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-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/36622318?v=4"}]}
jchecker@jchecker:~/functions$
```

Q11. How would you pass different arguments to the functions?

A. There are 4 methods to pass arguments:

1. Via OpenFaaS UI: You can go to the UI, click on the function you want to call. Pass the arguments below the invoke button and click invoke. The function then runs and generates the output.
2. Via curl: You can pass your input using the curl command as shown above.
3. Via faas-cli invoke: when you use this command, the function asks for input from the user and then you write your input and press CNT+D on keyboard to execute.
4. Via echo: You can use echo command like done in figlet to pass arguments to the function.

Q12. How would you change the slack-interactive function to react to different inputs?

A. The current function of handler.py is taking the input in a form of a string value and converting into a URL using the urllib library. Later, we use json.load function to convert the url into a json format via python dictionary and store it in response. By doing this we have the input and from here we can either change it or use it anywhere. In this script, the response can be used in the data which is given. Therefore making the script to react differently for each input. In the end we use json.dump which responds to various inputs.