

Your own Serverless Infrastructure

Rohnie Baskar

Student ID: 1629049

- Screenshot of invoking the figlet function: Invoking via OpenFAas UI and faas-cli. Prints the text in ascii form

The screenshot displays the OpenFAas UI interface. On the left, a sidebar contains the 'OPENFAAS' logo, a 'Deploy New Function' button, and a search bar with 'figlet' entered. The main panel shows the details for the 'figlet' function, including its status ('Not ready'), number of replicas (1), and invocation count (3). Below this, the 'Invoke function' section features an 'INVOKE' button and radio buttons for 'Text' (selected), 'JSON', and 'Download'. The request body is 'Hello, Faas, world'. The response section shows a status of 200, a round-trip time of 0.309s, and a response body containing the ASCII art for 'Hello, Faas, world'.

Status	Replicas	Invocation count
Not ready	1	3

Image: ghcr.io/openfaas/figlet:latest
URL: http://127.0.0.1:8080/function/figlet

Function process: figlet

Invoke function

INVOKE

☒ Text ☐ JSON ☐ Download

Request body
Hello, Faas, world

Response status: 200
Round-trip (s): 0.309

Response body
Hello, Faas, world


```
QEMU
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": "Rohnie Baskar",
            "author_icon": "",
            "image_url": "https://github.com/rohnie.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": "#3AA3E9",
            "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)
```

- slack-interactive/handler.py

```
QEMU - (Press ctrl + alt + g to release Mouse)

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": "Rohnie Baskar",
                "author_link": "https://github.com/rohnrie",
                "author_icon": "https://github.com/rohnrie.png",
                "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)

"handler.py" 28L, 1404C 28,27 All
```

- Screenshot of running `sudo journalctl -u faasd --lines 40`

```
QEMU - (Press ctrl + alt + g to release Mouse)
-- Logs begin at Thu 2023-02-02 04:01:53 UTC, end at Sat 2023-02-18 19:58:20 UTC. --
Feb 18 19:27:38 ubuntu-serv faasd[1026]: 2023/02/18 19:27:38 - gateway
Feb 18 19:27:38 ubuntu-serv faasd[1026]: 2023/02/18 19:27:38 - queue-worker
Feb 18 19:27:38 ubuntu-serv faasd[1026]: Starting: prometheus
Feb 18 19:27:39 ubuntu-serv faasd[1026]: Creating local directory: /var/lib/faasd/prometheus
Feb 18 19:27:39 ubuntu-serv faasd[1026]: 2023/02/18 19:27:39 Running prometheus with user: "65534"
Feb 18 19:27:42 ubuntu-serv faasd[1026]: 2023/02/18 19:27:42 Created container: prometheus
Feb 18 19:28:08 ubuntu-serv faasd[1026]: 2023/02/18 19:28:08 prometheus has IP: 10.62.0.2
Feb 18 19:28:08 ubuntu-serv faasd[1026]: 2023/02/18 19:28:08 Task: prometheus Container: prom
Feb 18 19:28:09 ubuntu-serv faasd[1026]: Starting: basic-auth-plugin
Feb 18 19:28:10 ubuntu-serv faasd[1026]: 2023/02/18 19:28:10 Created container: basic-auth-plugin
Feb 18 19:28:17 ubuntu-serv faasd[1026]: 2023/02/18 19:28:17 basic-auth-plugin has IP: 10.62.0.3
Feb 18 19:28:17 ubuntu-serv faasd[1026]: 2023/02/18 19:28:17 Task: basic-auth-plugin Contain
Feb 18 19:28:18 ubuntu-serv faasd[1026]: Starting: nats
Feb 18 19:28:18 ubuntu-serv faasd[1026]: Creating local directory: /var/lib/faasd/nats
Feb 18 19:28:18 ubuntu-serv faasd[1026]: 2023/02/18 19:28:18 Running nats with user: "65534"
Feb 18 19:28:19 ubuntu-serv faasd[1026]: 2023/02/18 19:28:19 Created container: nats
Feb 18 19:28:24 ubuntu-serv faasd[1026]: 2023/02/18 19:28:24 nats has IP: 10.62.0.4
Feb 18 19:28:24 ubuntu-serv faasd[1026]: 2023/02/18 19:28:24 Task: nats Container: nats
Feb 18 19:28:25 ubuntu-serv faasd[1026]: Starting: gateway
Feb 18 19:28:26 ubuntu-serv faasd[1026]: 2023/02/18 19:28:26 Created container: gateway
Feb 18 19:28:33 ubuntu-serv faasd[1026]: 2023/02/18 19:28:33 gateway has IP: 10.62.0.5
Feb 18 19:28:33 ubuntu-serv faasd[1026]: 2023/02/18 19:28:33 Task: gateway Container: gateway
Feb 18 19:28:33 ubuntu-serv faasd[1026]: Starting: queue-worker
Feb 18 19:28:34 ubuntu-serv faasd[1026]: 2023/02/18 19:28:34 Created container: queue-worker
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 queue-worker has IP: 10.62.0.6
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Task: queue-worker Container: q
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Supervisor init done in: About a minute
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 faasd: waiting for SIGTERM or SIGINT
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Resolver rebuilding map
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Resolver: "localhost"="127.0.0.1"
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Resolver: "faasd-provider"="10.62.0.1"
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Resolver: "prometheus"="10.62.0.2"
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Resolver: "basic-auth-plugin"="10.62.0.3"
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Resolver: "nats"="10.62.0.4"
Feb 18 19:28:40 ubuntu-serv faasd[1026]: 2023/02/18 19:28:40 Resolver: "gateway"="10.62.0.5"
lines 1-36
```

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

a. **Via Curl**

```
curl localhost:8080/function/slack-request
```

Here, input is actually not required, but faas-cli will look for an input.

To replicate the command similar to faas-cli, it would be

```
curl localhost:8080/function/slack-request -d "hi"
```

b. **Via faas-cli**

As this function doesn't require any request body, we can invoke it two ways

faas-cli invoke slack-request and press ctrl+D or echo "hi" | faas-cli invoke slack-request

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

The output we see is a json. It's a python object converted into a json string. This will be parsed as json as we have set content-type to application/json in the slack-request.yml file. The output can be seen in the screenshot above. It has three attachments

- Title: The awesome world of cloud computing! COEN 241 with subtitle amazing level and the picture of my github profile.
- About COEN 241 and text of it's the most amazing class ever
- Two buttons of recommendation - Of course and Most definitely

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

a. Via curl

```
curl -H "Content-Type: application/json" -X POST -d '{"name": "jay"}' localhost:8080/function/slack-interactive
```

b. Via faas-cli

Echo '{"name": "jay"}' | faas-cli invoke slack-interactive

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

The output is a json. (Python object converted to json string and parsed as json due to content-type setup). The output is the same regardless of any button chosen from slack-request. It consists of text saying, yes, recommending COEN 241 is a great choice and COEN 241 is amazing. It provides a link to go to scu course descriptions under the hyperlink COEN 241. It also provides a thumbnail picture of SCU mission logo and clicking on that takes to scu course descriptions page.

The python handler is changed to react to different inputs of button pressed. This can be seen in the summary text saying which button was pressed. The rest of the output remains the same.

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

- -d to provide input in curl. In Faas-cli, echo can be piped. Different input parameters can be passed to the function by manipulating this json. If username and password has to be passed, the json can be modified to have that. In our slang request, we are sending which recommendation button was clicked to slack-interactive function
- Via the YAML file: Some arguments not exactly to the function but to determine how the function would behave and its environment can be set in the yml file like:

Gateway: To specify which host and port will the service come up on

Provider: here it would be faas

Functions: to specify the function specs

Lang: The language of the function can be specified

Handler: which function will be invoked and what it would do
Image: docker hub image

Environment: env can be set here

Content-type can be set in this yml

Build-option of dev/debug can be set here

Additional packages required can be set here to passed

- Instead of yml file, --lang can be used to set language and --build-arg for other additional arguments like ADDITIONAL_PACKAGE
- The other arguments could be --filter, --parallel or -X/--header for curl commands

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

- Parse the request to get the payload
- Get the action from it and the value of button pressed
- Setting value to the button pressed based on equality condition
- Appending value to pretext
- Changes shown in the below screenshot

```
QEMU - (Press ctrl + alt + g to release Mouse)
import json
import urllib
def handle(req):
    urlstring=urllib.unquote(req).decode('utf8').strip('payload=')
    response=json.loads(urlstring)
    buttonSelected = response["actions"][0]["value"]
    value=(buttonPressed == 'recommended' and 'Of Course was pressed') or 'Definitely was pressed!'
    data={
        "attachments": [
            {
                "replace_original": True,
                "response_type": "ephemeral",
                "fallback": "Required plain-text summary of the attachment.",
                "color": "#36a64f",
                "pretext": value+"Ahh yeah! Great choice, COEN 241 is absolutely amazing!",
                "author_name": "Rohnie Baskar",
                "author_link": "https://github.com/rohnies",
                "author_icon": "https://github.com/rohnies.png",
                "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)

"handler.py" 30L, 1562C 6,51 A11
```

