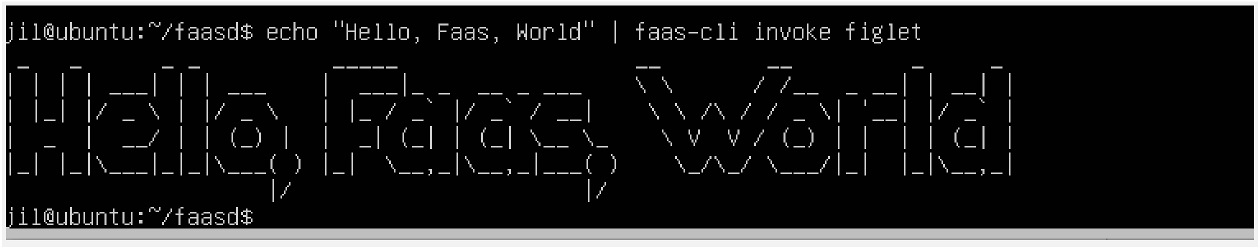
Screenshot explanation

Screenshots

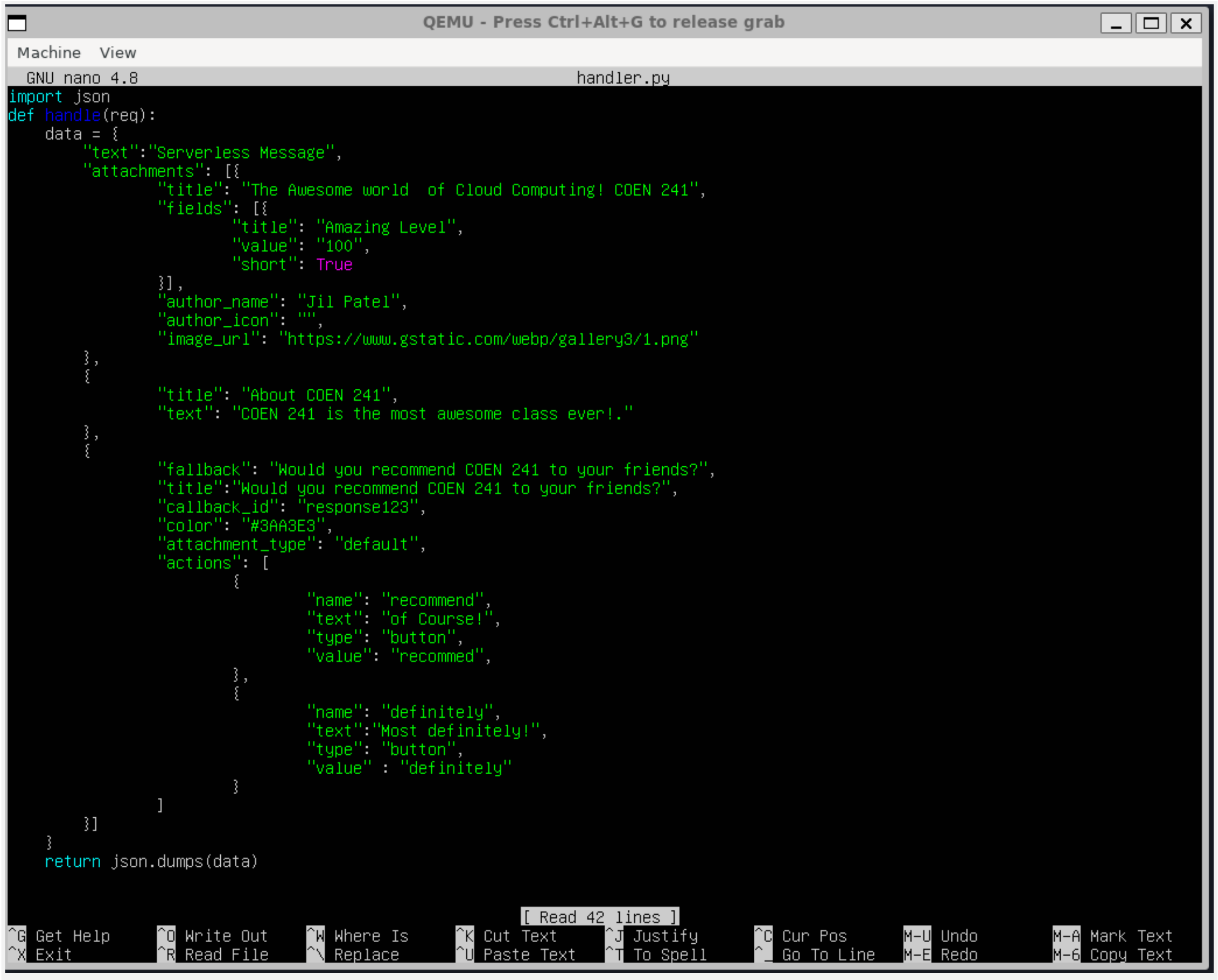
1) Provide a screenshot of invoking the figlet function (10 pts)



```
jil@ubuntu:~/faasd$ echo "Hello, Faas, World" | faas-cli invoke figlet
 _   _      _ _          _____                    __        __         _     _
| | | | ___| | | ___    |  ___|_ _  __ _ ___      \ \      / /__  _ __| | __| |
| |_| |/ _ \ | |/ _ \   | |_ / _` |/ _` / __|      \ \ /\ / / _ \| '__| |/ _` |
|  _  |  __/ | | (_) |  |  _| (_| | (_| \__ \_      \ V  V / (_) | |  | | (_| |
|_| |_|\___|_|_|\___( ) |_|  \__,_|\__,_|___( )      \_/\_/ \___/|_|  |_|\__,_|
                    |/                      |/
jil@ubuntu:~/faasd$
```

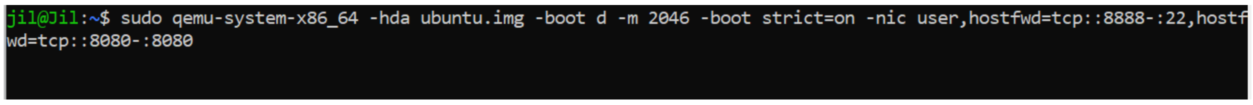
The above screenshot is the output of invoking the Figlet function once it is deployed. We are passing “Hello, Faas, World” as input params to the Figlet function using Pipe. This Figlet function takes this input and prints it on the command shell.

2) Complete slack-request/handler.py (20 pt)



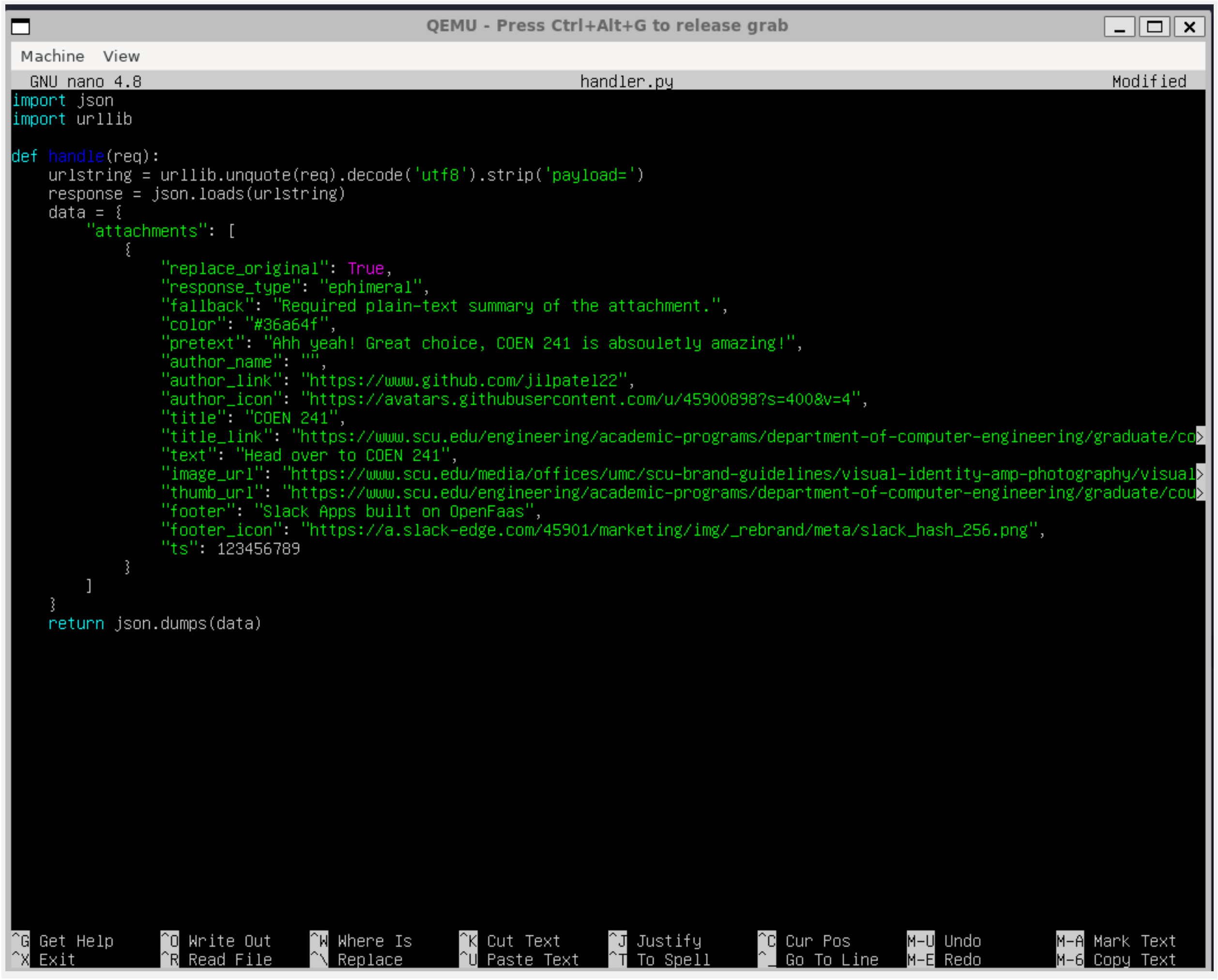
```
  GNU nano 4.8                                        handler.py
import json
def handle(req):
    data = {
        "text":"Serverless Message",
        "attachments": [{
                "title": "The Awesome world  of Cloud Computing! COEN 241",
                "fields": [{
                        "title": "Amazing Level",
                        "value": "100",
                        "short": True
                }],
                "author_name": "Jil Patel",
                "author_icon": "",
                "image_url": "https://www.gstatic.com/webp/gallery3/1.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": "recommed",
                        },
                        {
                                "name": "definitely",
                                "text":"Most definitely!",
                                "type": "button",
                                "value" : "definitely"
                        }
                ]
        }]
    }
    return json.dumps(data)

                                             [ Read 42 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo        M-A Mark Text
^X Exit         ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line   M-E Redo        M-6 Copy Text
```

In the above screenshot we have replaced the author_name and image_url field with its respective value. Once it is done we will build image, push the function to dockerHub and will deploy the function. It will run on the port 8080 which can be accessed from our host machine using network forwarding. Below is the command for the same.



```
jil@Jil:~$ sudo qemu-system-x86_64 -hda ubuntu.img -boot d -m 2046 -boot strict=on -nic user,hostfwd=tcp::8888-:22,hostf
wd=tcp::8080-:8080
```

3) Complete slack-interactive/handler.py (20 pt)

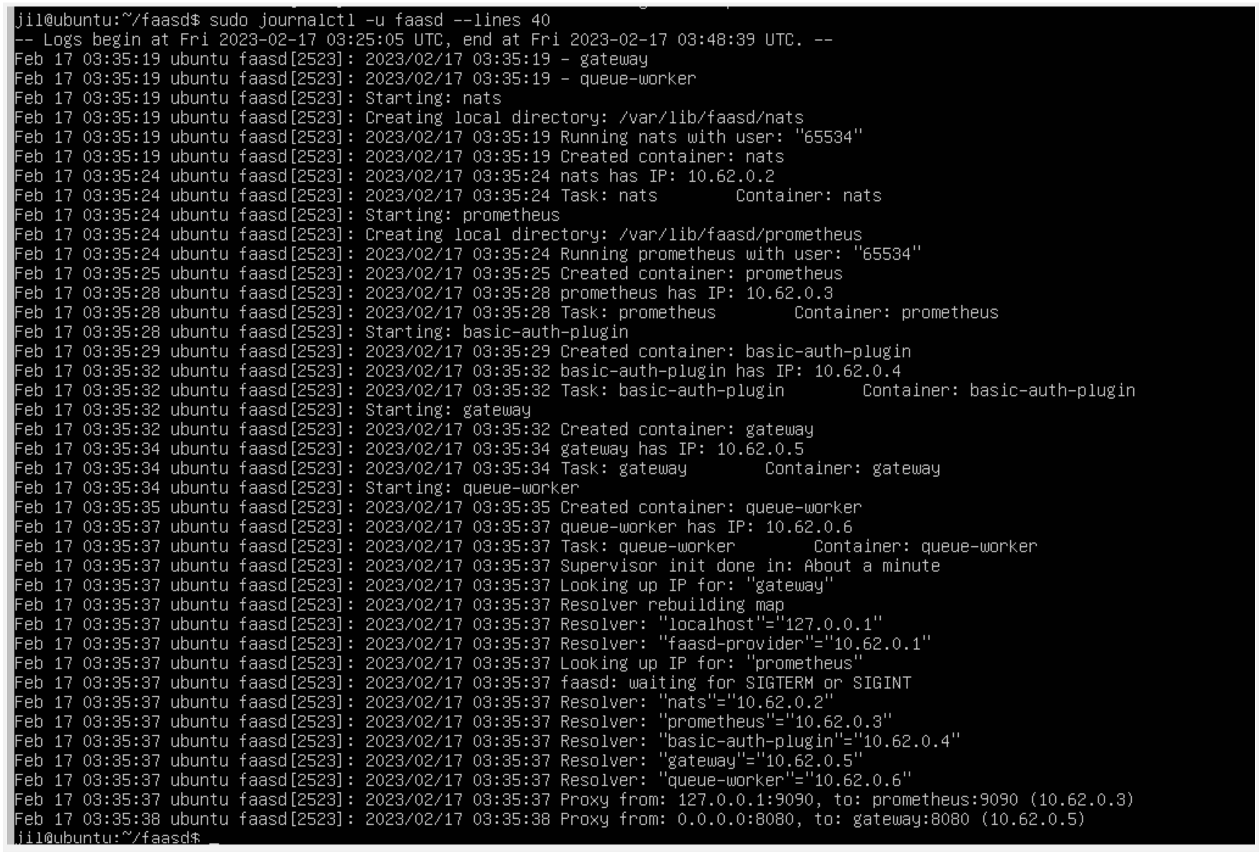


```
  GNU nano 4.8                         handler.py                         Modified
import json
import urllib

def handle(req):
    urlstring = urllib.unquote(req).decode('utf8').strip('payload=')
    response = json.loads(urlstring)
    data = {
        "attachments": [
            {
                "replace_original": True,
                "response_type": "ephimeral",
                "fallback": "Required plain-text summary of the attachment.",
                "color": "#36a64f",
                "pretext": "Ahh yeah! Great choice, COEN 241 is absouletly amazing!",
                "author_name": "",
                "author_link": "https://www.github.com/jilpatel22",
                "author_icon": "https://avatars.githubusercontent.com/u/45900898?s=400&v=4",
                "title": "COEN 241",
                "title_link": "https://www.scu.edu/engineering/academic-programs/department-of-computer-engineering/graduate/co>
                "text": "Head over to COEN 241",
                "image_url": "https://www.scu.edu/media/offices/umc/scu-brand-guidelines/visual-identity-amp-photography/visual>
                "thumb_url": "https://www.scu.edu/engineering/academic-programs/department-of-computer-engineering/graduate/cou>
                "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)
```

In the same way as explained above handler.py for request function, the interactive function will take string as an input and will format it in JSON using json.loads.

4) Provide a screenshot of running the following command (10 pts)

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

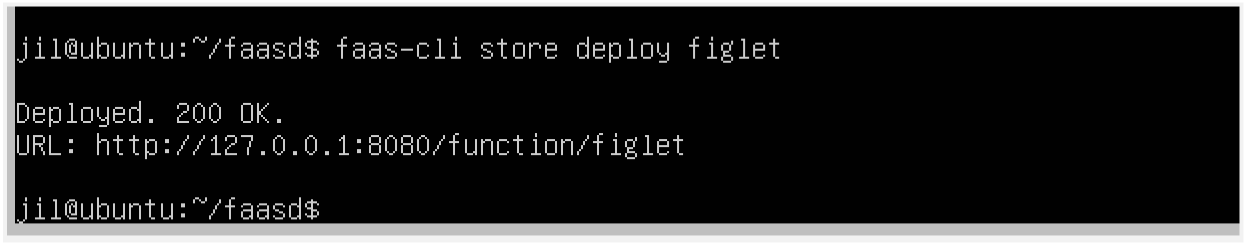


```
jil@ubuntu:~/faasd$ sudo journalctl -u faasd --lines 40
-- Logs begin at Fri 2023-02-17 03:25:05 UTC, end at Fri 2023-02-17 03:48:39 UTC. --
Feb 17 03:35:19 ubuntu faasd[2523]: 2023/02/17 03:35:19 - gateway
Feb 17 03:35:19 ubuntu faasd[2523]: 2023/02/17 03:35:19 - queue-worker
Feb 17 03:35:19 ubuntu faasd[2523]: Starting: nats
Feb 17 03:35:19 ubuntu faasd[2523]: Creating local directory: /var/lib/faasd/nats
Feb 17 03:35:19 ubuntu faasd[2523]: 2023/02/17 03:35:19 Running nats with user: "65534"
Feb 17 03:35:19 ubuntu faasd[2523]: 2023/02/17 03:35:19 Created container: nats
Feb 17 03:35:24 ubuntu faasd[2523]: 2023/02/17 03:35:24 nats has IP: 10.62.0.2
Feb 17 03:35:24 ubuntu faasd[2523]: 2023/02/17 03:35:24 Task: nats        Container: nats
Feb 17 03:35:24 ubuntu faasd[2523]: Starting: prometheus
Feb 17 03:35:24 ubuntu faasd[2523]: Creating local directory: /var/lib/faasd/prometheus
Feb 17 03:35:24 ubuntu faasd[2523]: 2023/02/17 03:35:24 Running prometheus with user: "65534"
Feb 17 03:35:25 ubuntu faasd[2523]: 2023/02/17 03:35:25 Created container: prometheus
Feb 17 03:35:28 ubuntu faasd[2523]: 2023/02/17 03:35:28 prometheus has IP: 10.62.0.3
Feb 17 03:35:28 ubuntu faasd[2523]: 2023/02/17 03:35:28 Task: prometheus        Container: prometheus
Feb 17 03:35:28 ubuntu faasd[2523]: Starting: basic-auth-plugin
Feb 17 03:35:29 ubuntu faasd[2523]: 2023/02/17 03:35:29 Created container: basic-auth-plugin
Feb 17 03:35:32 ubuntu faasd[2523]: 2023/02/17 03:35:32 basic-auth-plugin has IP: 10.62.0.4
Feb 17 03:35:32 ubuntu faasd[2523]: 2023/02/17 03:35:32 Task: basic-auth-plugin        Container: basic-auth-plugin
Feb 17 03:35:32 ubuntu faasd[2523]: Starting: gateway
Feb 17 03:35:32 ubuntu faasd[2523]: 2023/02/17 03:35:32 Created container: gateway
Feb 17 03:35:34 ubuntu faasd[2523]: 2023/02/17 03:35:34 gateway has IP: 10.62.0.5
Feb 17 03:35:34 ubuntu faasd[2523]: 2023/02/17 03:35:34 Task: gateway        Container: gateway
Feb 17 03:35:34 ubuntu faasd[2523]: Starting: queue-worker
Feb 17 03:35:35 ubuntu faasd[2523]: 2023/02/17 03:35:35 Created container: queue-worker
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 queue-worker has IP: 10.62.0.6
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Task: queue-worker        Container: queue-worker
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Supervisor init done in: About a minute
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Looking up IP for: "gateway"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Resolver rebuilding map
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Resolver: "localhost"="127.0.0.1"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Resolver: "faasd-provider"="10.62.0.1"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Looking up IP for: "prometheus"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 faasd: waiting for SIGTERM or SIGINT
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Resolver: "nats"="10.62.0.2"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Resolver: "prometheus"="10.62.0.3"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Resolver: "basic-auth-plugin"="10.62.0.4"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Resolver: "gateway"="10.62.0.5"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Resolver: "queue-worker"="10.62.0.6"
Feb 17 03:35:37 ubuntu faasd[2523]: 2023/02/17 03:35:37 Proxy from: 127.0.0.1:9090, to: prometheus:9090 (10.62.0.3)
Feb 17 03:35:38 ubuntu faasd[2523]: 2023/02/17 03:35:38 Proxy from: 0.0.0.0:8080, to: gateway:8080 (10.62.0.5)
jil@ubuntu:~/faasd$ _
```

The above command prints the first 40 lines of the logs and we can determine the status of the faas whether it is running or not.

5) Provide a screenshot of your OpenFaaS gateway AFTER deploying figlet, slack-handler and slack-interactive functions (10 pts)

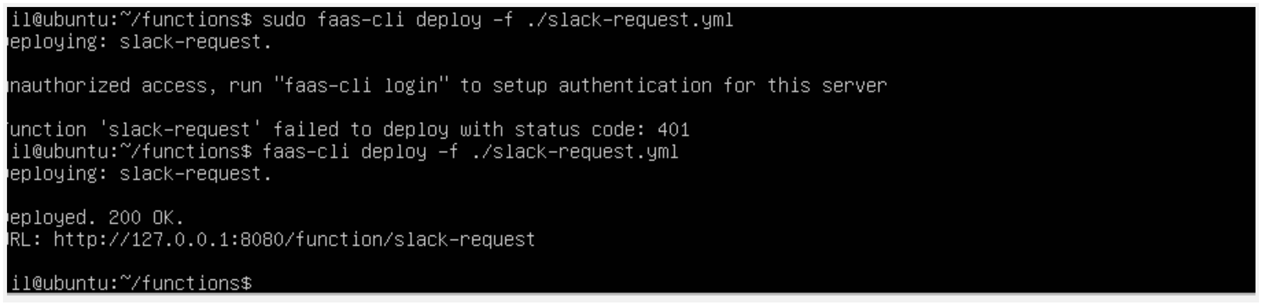
figlet



```
jil@ubuntu:~/faasd$ faas-cli store deploy figlet

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

jil@ubuntu:~/faasd$
```

The above command deploys the Figlet function and prints the url on which it is hosted. We can access the function from our host machine using the same url as we have built a network bridge between host OS and virtual machine.

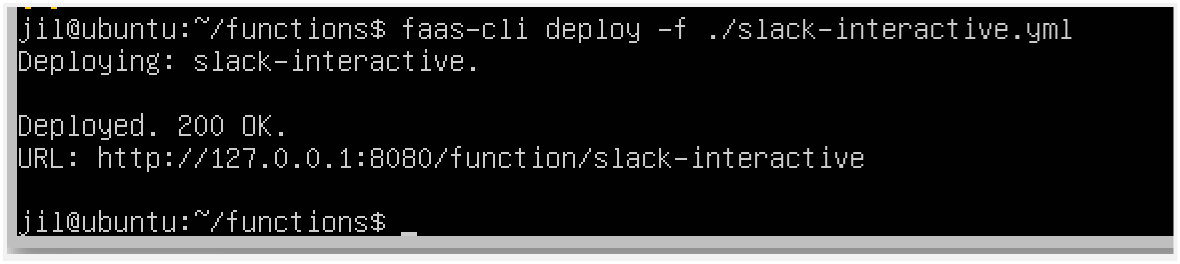
Slack-request



```
il@ubuntu:~/functions$ sudo faas-cli deploy -f ./slack-request.yml
eploying: slack-request.

nauthorized access, run "faas-cli login" to setup authentication for this server

unction 'slack-request' failed to deploy with status code: 401
il@ubuntu:~/functions$ faas-cli deploy -f ./slack-request.yml
eploying: slack-request.

eployed. 200 OK.
RL: http://127.0.0.1:8080/function/slack-request

il@ubuntu:~/functions$
```

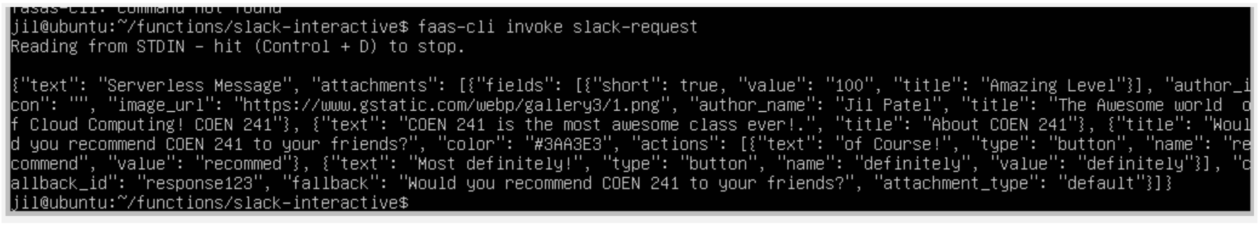
Same goes for the slack-request function as explained above

Slack-interactive



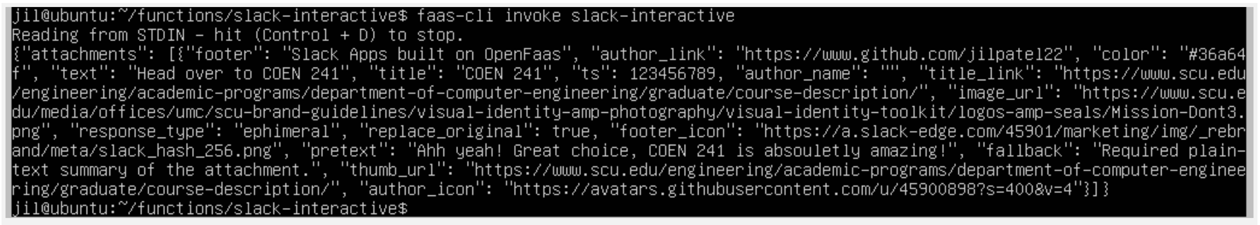
```
jil@ubuntu:~/functions$ faas-cli deploy -f ./slack-interactive.yml
Deploying: slack-interactive.

Deployed. 200 OK.
URL: http://127.0.0.1:8080/function/slack-interactive

jil@ubuntu:~/functions$ _
```

6) Provide a screenshot of invoking slack-request and slack-interactive functions (10 pts)



```
faas-cli: command not found
jil@ubuntu:~/functions/slack-interactive$ 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_i
con": "", "image_url": "https://www.gstatic.com/webp/gallery3/1.png", "author_name": "Jil Patel", "title": "The Awesome world  o
f Cloud Computing! COEN 241"}, {"text": "COEN 241 is the most awesome class ever!.", "title": "About COEN 241"}, {"title": "Woul
d you recommend COEN 241 to your friends?", "color": "#3AA3E3", "actions": [{"text": "of Course!", "type": "button", "name": "re
commend", "value": "recommed"}, {"text": "Most definitely!", "type": "button", "name": "definitely", "value": "definitely"}], "c
allback_id": "response123", "fallback": "Would you recommend COEN 241 to your friends?", "attachment_type": "default"}]}
jil@ubuntu:~/functions/slack-interactive$
```

The above command invokes the slack-request function which first for the command line input. As we don't want to pass it in the request-slack we escape this and then the function prints its output in the command line. The output that we obtained is the json dumped data.



```
jil@ubuntu:~/functions/slack-interactive$ faas-cli invoke slack-interactive
Reading from STDIN - hit (Control + D) to stop.
{"attachments": [{"footer": "Slack Apps built on OpenFaas", "author_link": "https://www.github.com/jilpatel22", "color": "#36a64
f", "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-description/", "image_url": "https://www.scu.e
du/media/offices/umc/scu-brand-guidelines/visual-identity-amp-photography/visual-identity-toolkit/logos-amp-seals/Mission-Dont3.
png", "response_type": "ephimeral", "replace_original": true, "footer_icon": "https://a.slack-edge.com/45901/marketing/img/_rebr
and/meta/slack_hash_256.png", "pretext": "Ahh yeah! Great choice, COEN 241 is absouletly amazing!", "fallback": "Required plain-
text summary of the attachment.", "thumb_url": "https://www.scu.edu/engineering/academic-programs/department-of-computer-enginee
ring/graduate/course-description/", "author_icon": "https://avatars.githubusercontent.com/u/45900898?s=400&v=4"}]}
jil@ubuntu:~/functions/slack-interactive$
```

In the above screenshot we have invoked the slack-interactive function using the input params, and once we enter the params we will press “ctrl + d” to stop. This function will take this input, appends it to existing json and we get the output with our string in json format.This function can also be invoked from the host machine OS using postman or web browser.