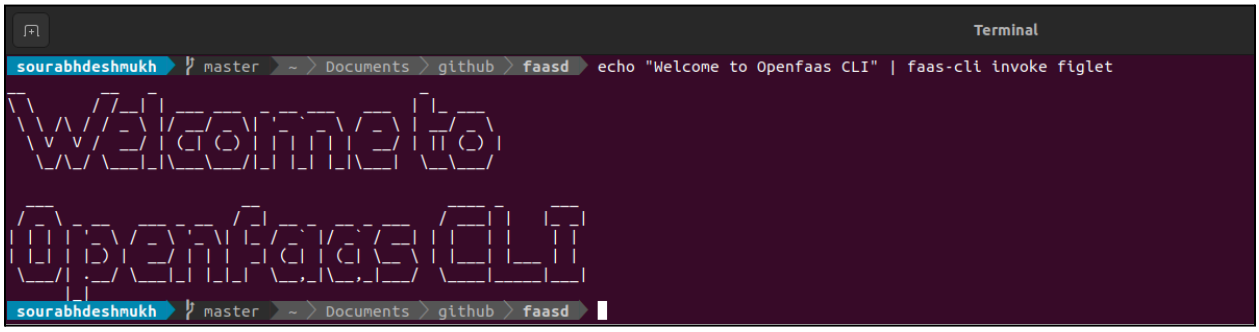
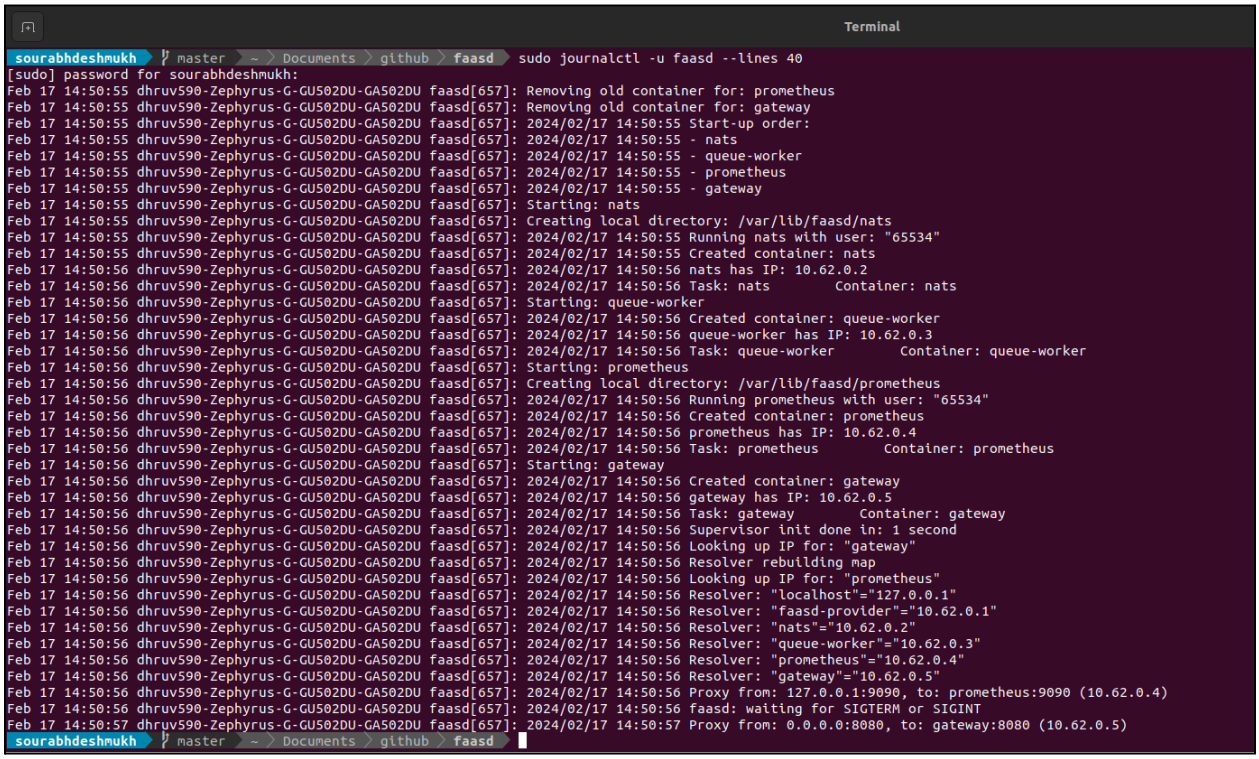
Provide a screenshot of invoking the figlet function.



Provide a screenshot of running the following command.

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



Complete slack-request/handler.py

https://github.com/sourabhdeshmukh/Cloud-CSEN-241/blob/main/Assignments/HW2/functions/slack-request/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": "Sourabh Deshmukh",
           "author_icon": "https://github.com/sourabhdeshmukh",
           "image_url":
"https://hub.docker.com/r/sourabhdeshmukh/slack-request"
       },
       {
           "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)
```

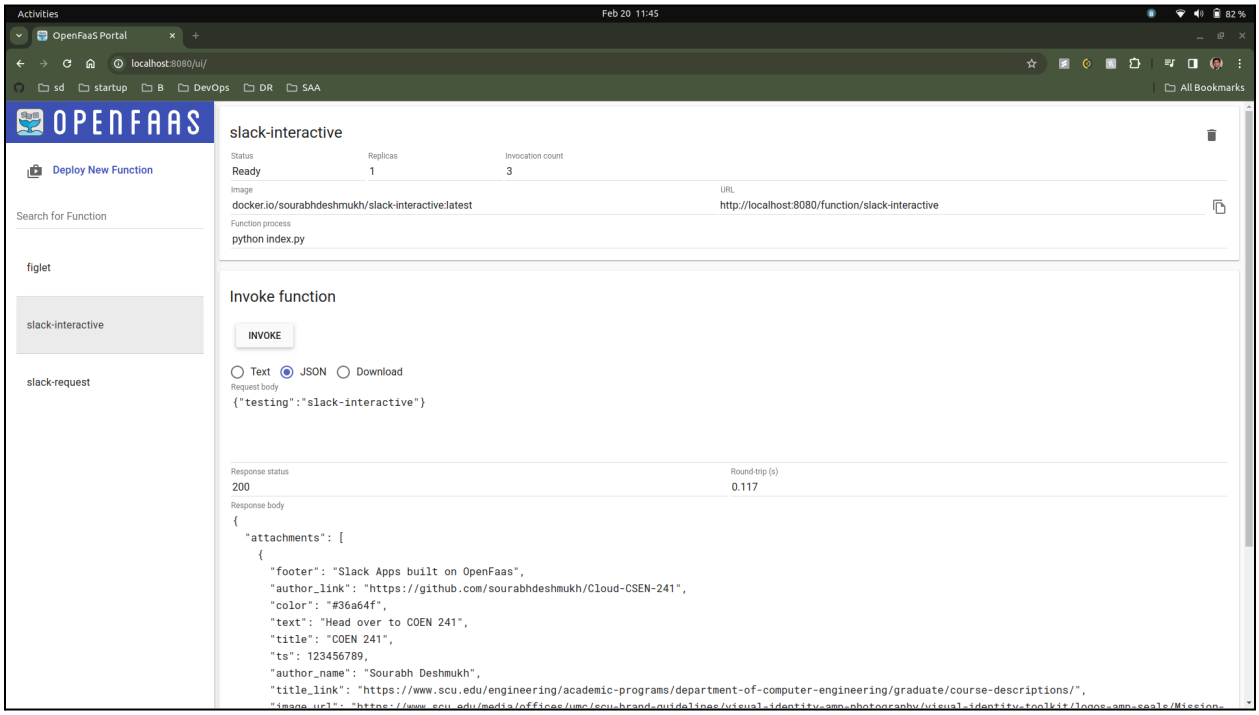
Complete slack-interactive/handler.py

https://github.com/sourabhdeshmukh/Cloud-CSEN-241/blob/main/Assignments/HW2/functions/slack-interactive/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": "Sourabh Deshmukh",
                "author_link":
"https://github.com/sourabhdeshmukh/Cloud-CSEN-241",
                "author_icon": "https://github.com/sourabhdeshmukh",
                "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-identit
y-amp-photography/visual-identity-toolkit/logos-amp-seals/Mission-Dont3.pn
g",
```

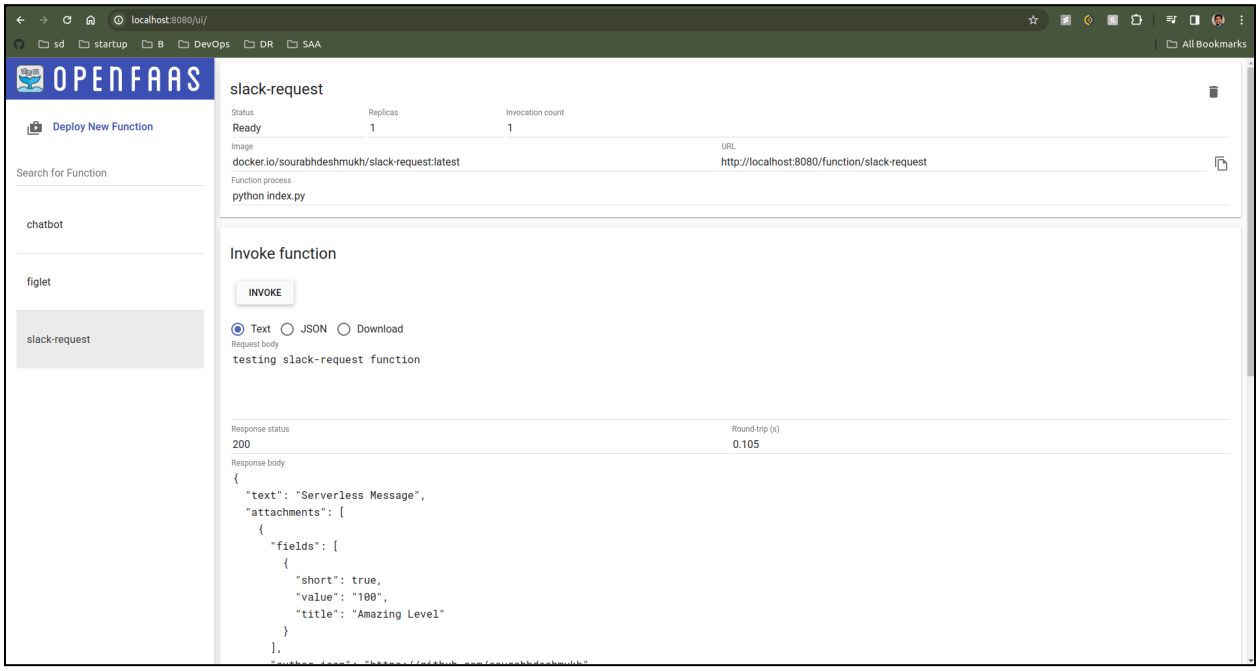
```
                "thumb_url":
"https://www.scu.edu/engineering/academic-programs/department-of-computer-
engineering/graduate/course-descriptions/",
                "footer": "Slack Apps built on OpenFaas",
                "footer_icon":
"https://a.slack-edge.com/45901/marketing/img/_rebrand/meta/slack_hash_256
.png",
                "ts": 123456789
            }
        ]
    }
    return json.dumps(data)
```

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

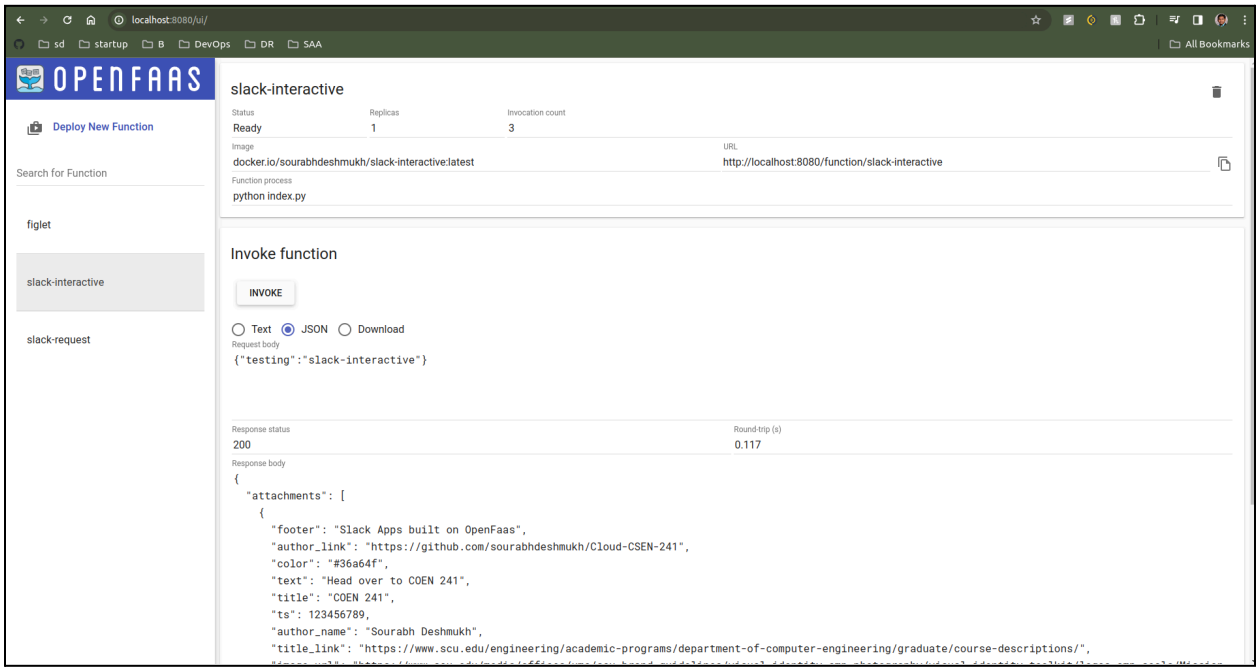


Provide a screenshot of invoking slack-request and slack-interactive functions

Slack request



Slack-interactive

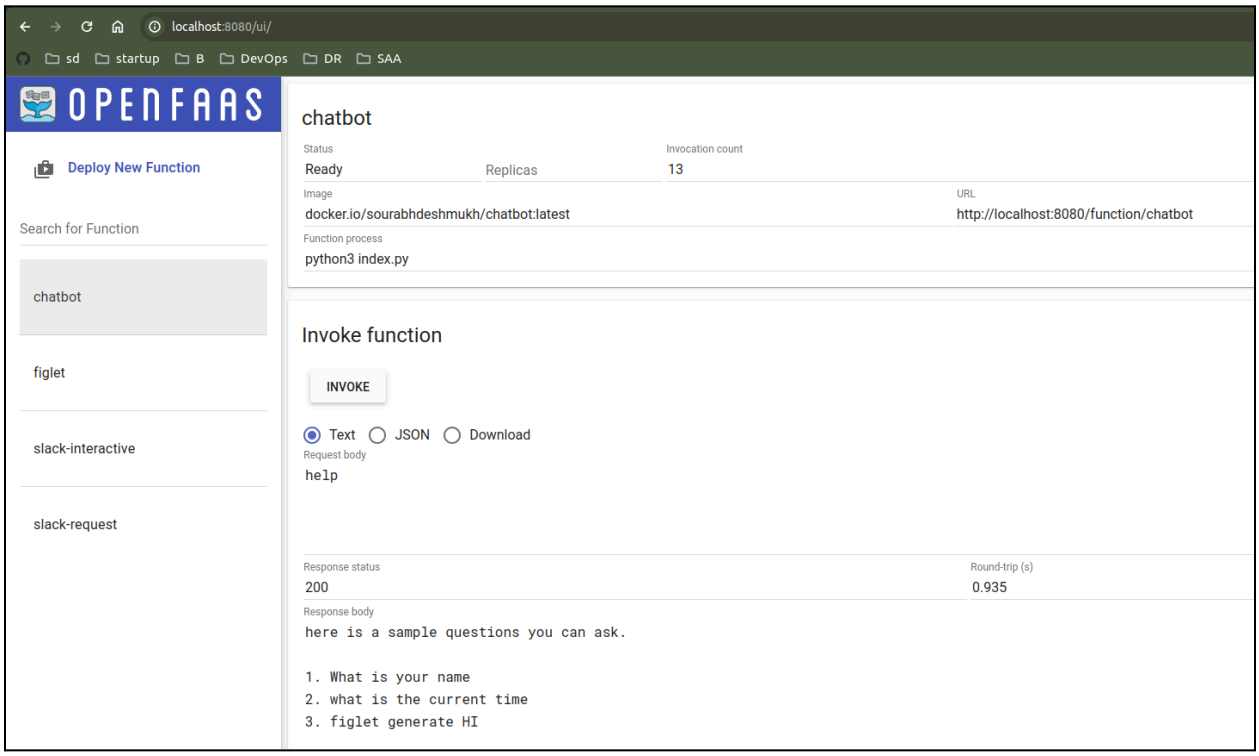


- Complete the chatbot with a yml file

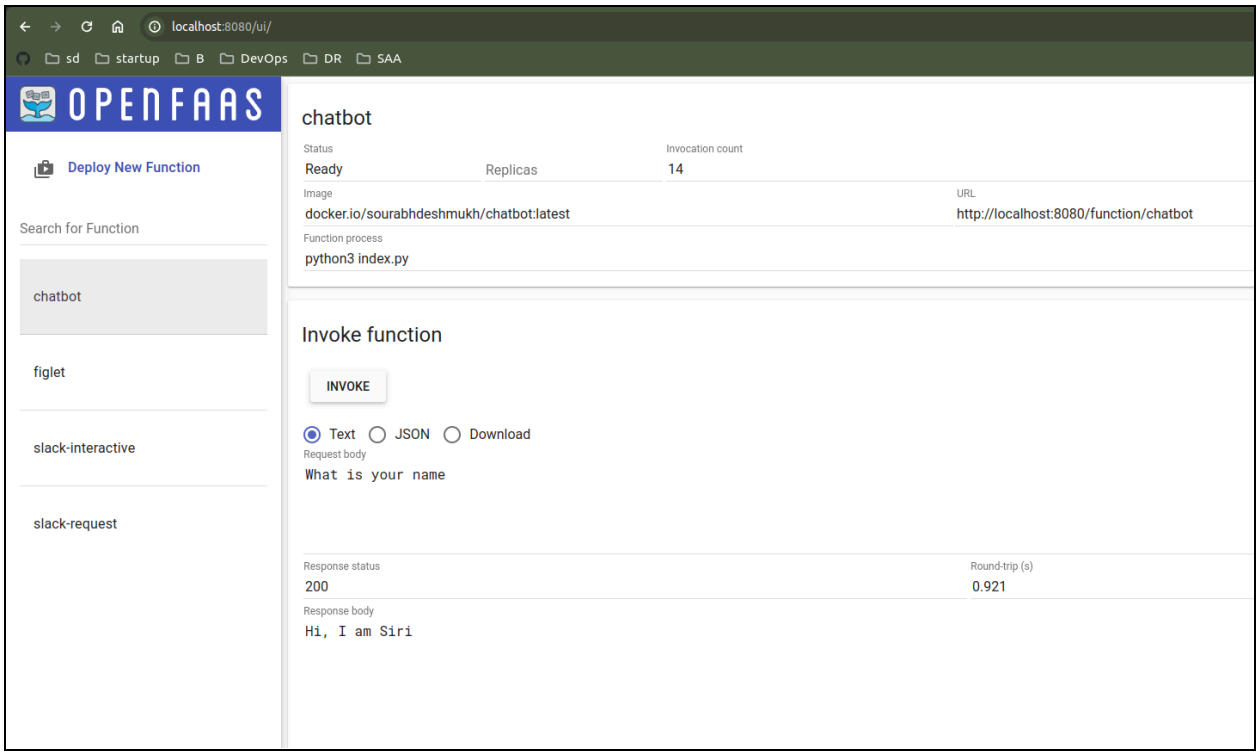
```
import requests
import datetime
import random

# Chatbot Handler Function
def handle(req):
    processedText = req.lower().strip()
    preformat = list(processedText.split())

    if "name" in preformat:
        response = get_name()
    elif "time" in preformat:
        response = get_time()
    elif "figlet" in preformat:
        response = get_figlet(req)
    else:
        return "here is a sample questions you can ask.\n\n1. What is your
name\n2. what is the current time\n3. figlet generate HI\n"
    return response

# Random name generator for chatbot
def get_name():
    names = ["Assistant", "Alexa", "Siri"]
    return "Hi, I am " + random.choice(names)

# Random format for current date and time generator.
def get_time():
    now = datetime.datetime.now()
    responses = [
        "Current time is {}".format(now.strftime('%I:%M %p')),
        "Current Date is {}".format(now.strftime('%A, %B %d, %Y')),
        "Current Date and Time is {}".format(now.strftime('%A, %B %d, %Y,
%I:%M %p')),
    ]
    return random.choice(responses)

# External figlet function calling function.
def get_figlet(req):
```

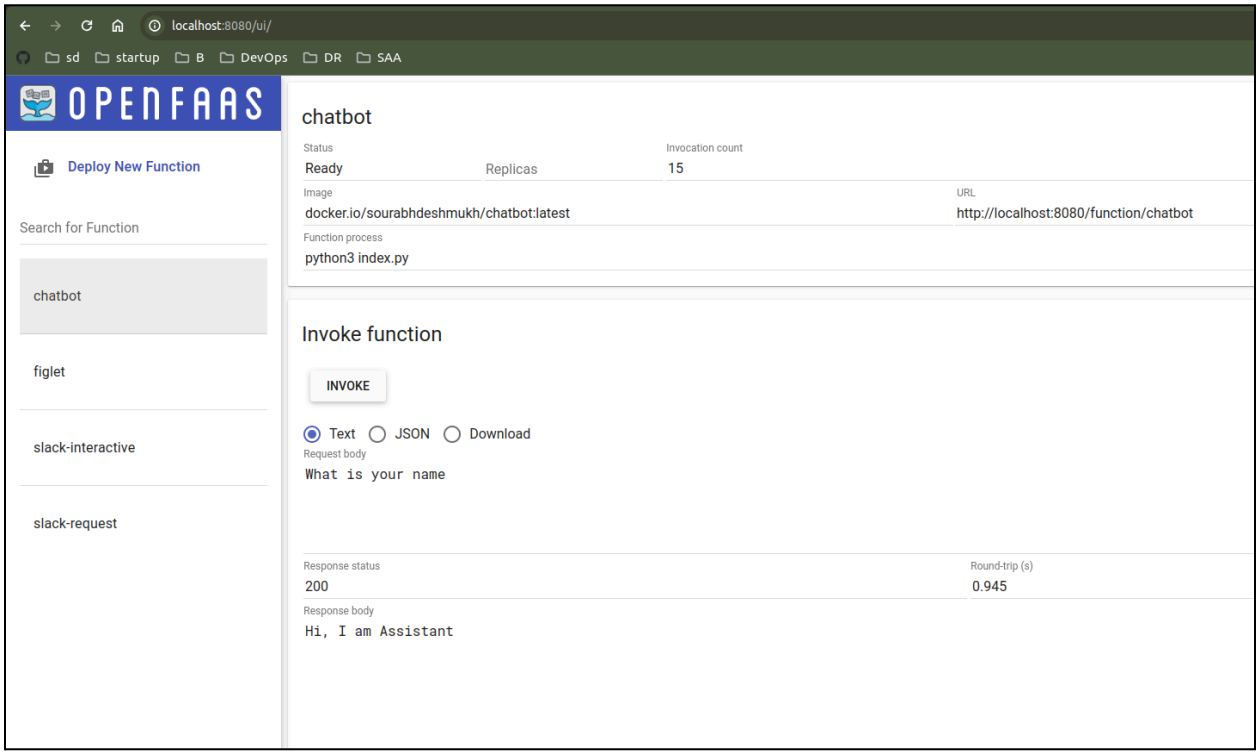
```
    text = req.split("generate", 1)[1].strip()
    url="http://10.62.0.1:8080/function/figlet"
    result = requests.post(url, data=text)
    return result.text
```

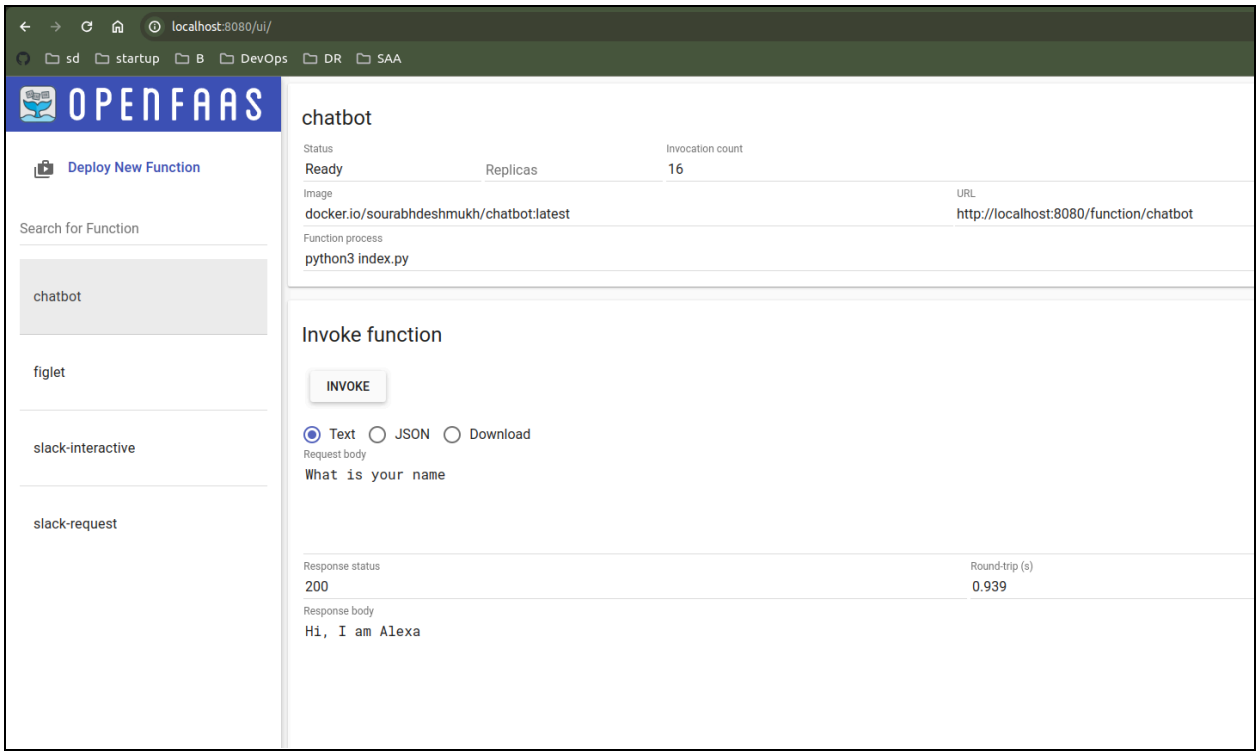
- Provide a screenshot of invoking three different cases of the chatbot
- Chatbot help.



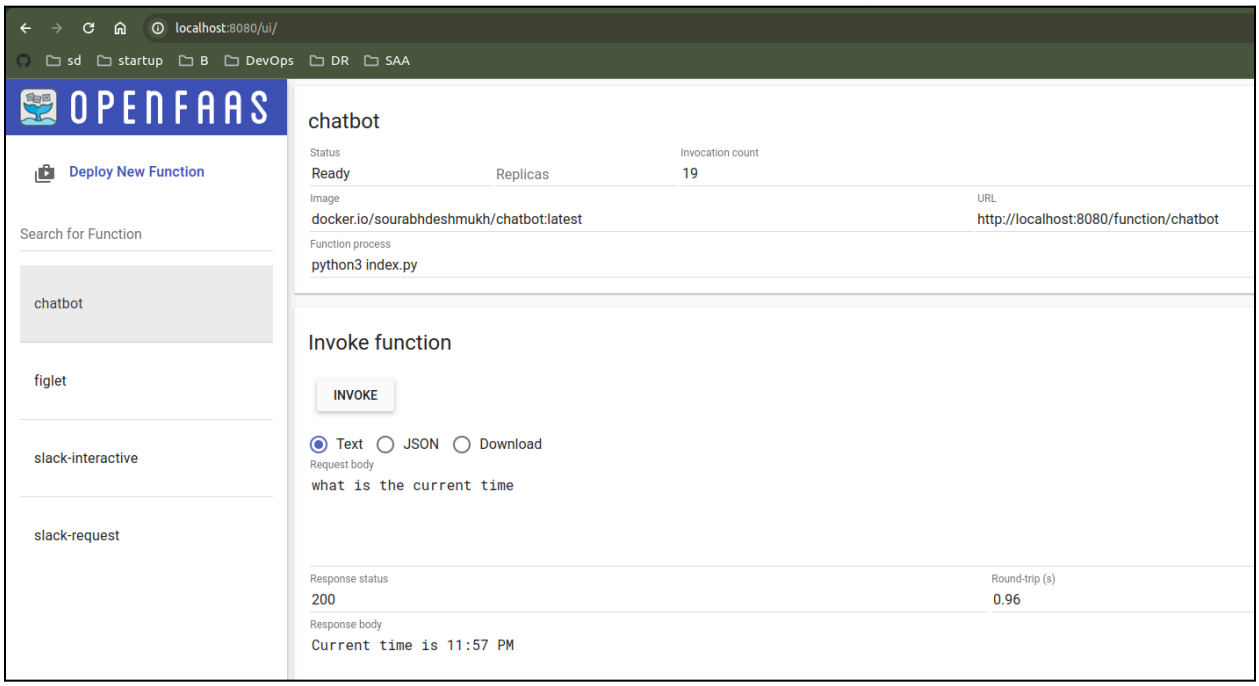
- Generate 3 different names.

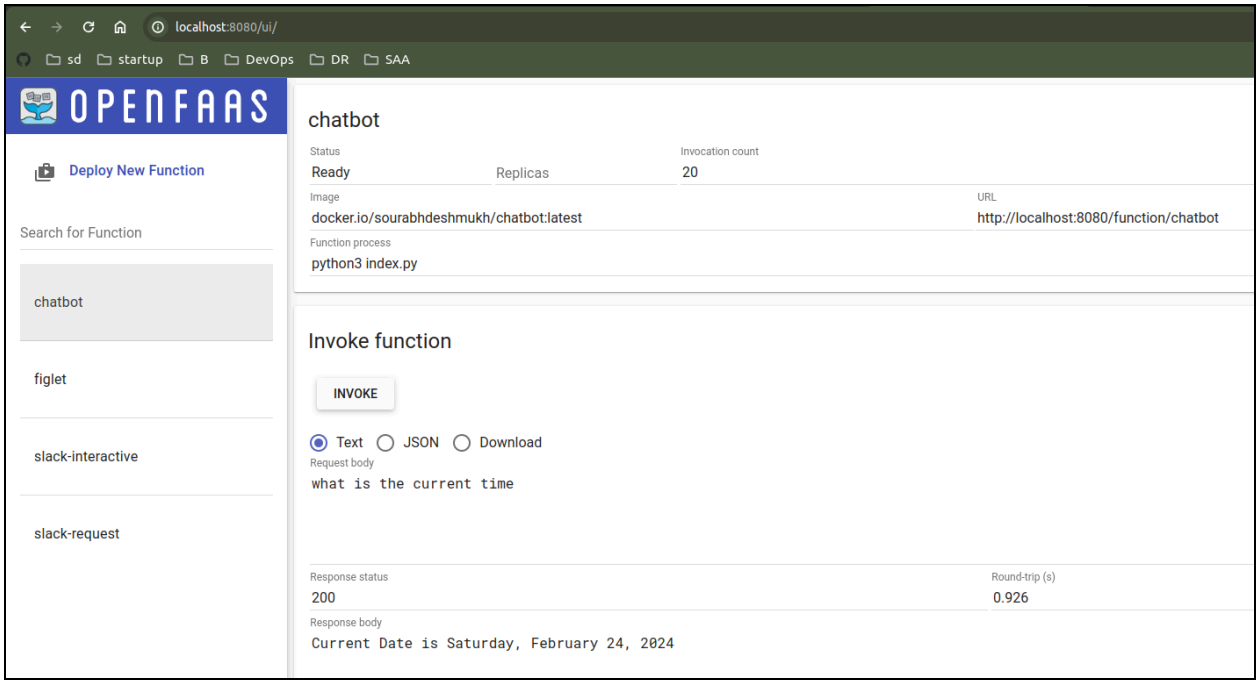

localhost:8080/ui/
sd
startup
B
DevOps
DR
SAA
OPENFAAS
Deploy New Function
Search for Function
chatbot
figlet
slack-interactive
slack-request
chatbot
Status
Ready
Replicas
Invocation count
14
Image
docker.io/sourabhdeshmukh/chatbot:latest
URL
http://localhost:8080/function/chatbot
Function process
python3 index.py
Invoke function
INVOKE
Text
JSON
Download
Request body
What is your name
Response status
200
Round-trip (s)
0.921
Response body
Hi, I am Siri

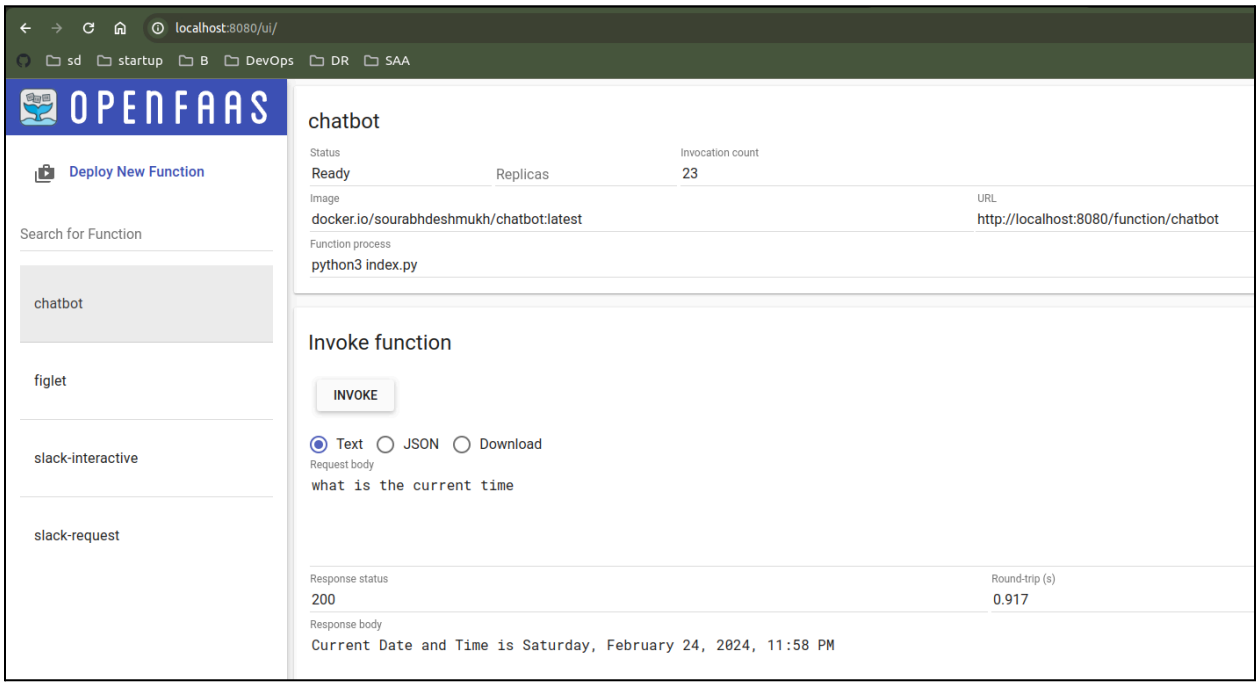

localhost:8080/ui/
sd
startup
B
DevOps
DR
SAA
OPENFAAS
Deploy New Function
Search for Function
chatbot
figlet
slack-interactive
slack-request
chatbot
Status
Ready
Replicas
Invocation count
15
Image
docker.io/sourabhdeshmukh/chatbot:latest
URL
http://localhost:8080/function/chatbot
Function process
python3 index.py
Invoke function
INVOKE
Text
JSON
Download
Request body
What is your name
Response status
200
Round-trip (s)
0.945
Response body
Hi, I am Assistant



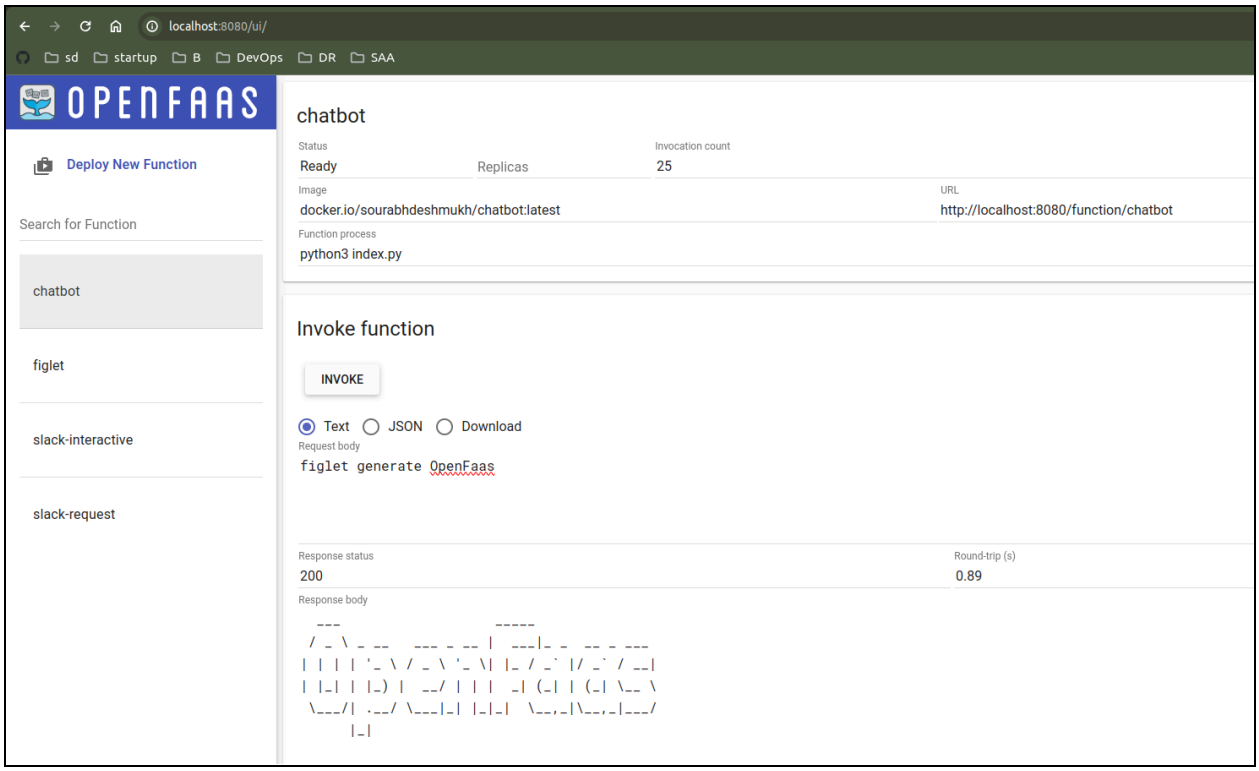
- Generate 3 different formats for current time.





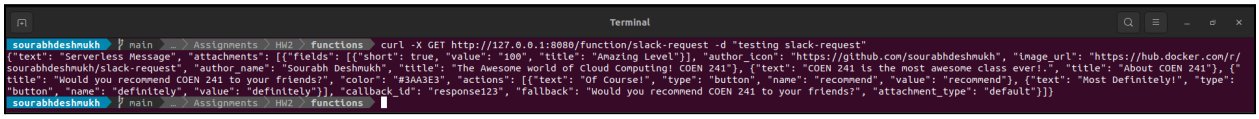


- Invoking figlet function



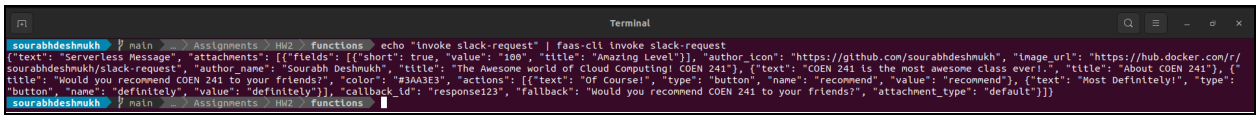
- What is the command to invoke the slack-request function (2 pts)?
    - Via Curl

```
sourabhdeshmukh$ curl -X GET http://127.0.0.1:8080/function/slack-request
-d "testing slack-request"
```



- Via faas-cli

```
sourabhdeshmukh$ echo "invoke slack-request" | faas-cli invoke
slack-request
```



- What is the output you see when you invoke the slack-request function?
- When we provide the text based or json based argument then we get the json data object defined in the handle function of slack-request handler.py

- OUTPUT

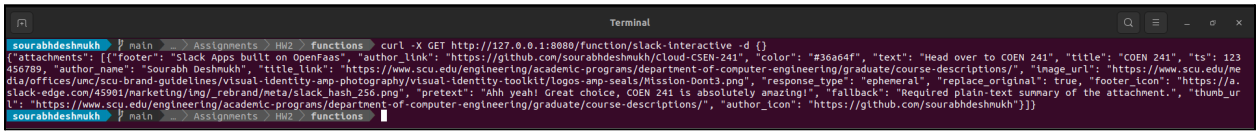
Response body


```
{
  "text": "Serverless Message",
  "attachments": [
    {
      "fields": [
        {
          "short": true,
          "value": "100",
          "title": "Amazing Level"
        }
      ],
      "author_icon": "https://github.com/sourabhdeshmukh",
      "image_url": "https://hub.docker.com/r/sourabhdeshmukh/slack-request",
      "author_name": "Sourabh Deshmukh",
      "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"
    }
  ]
}
```


- What is the command to invoke the slack-interactive function?
    - Via curl

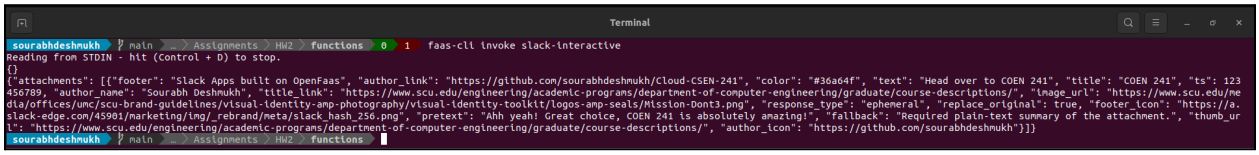
```
sourabhdeshmukh$ curl -X GET
```

```
http://127.0.0.1:8080/function/slack-interactive -d {}
```



- Via faas-cli

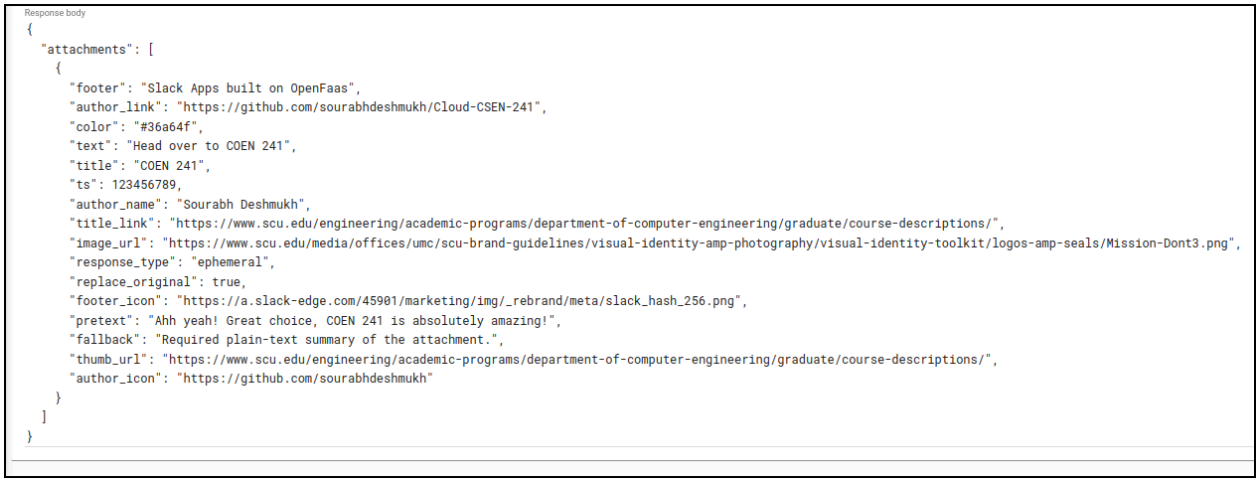
```
sourabhdeshmukh$ faas-cli invoke slack-interactive
```



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

- When we provide the text based or json based argument then we get the json data object defined in the handle function of slack-request handler.py

- OUTPUT

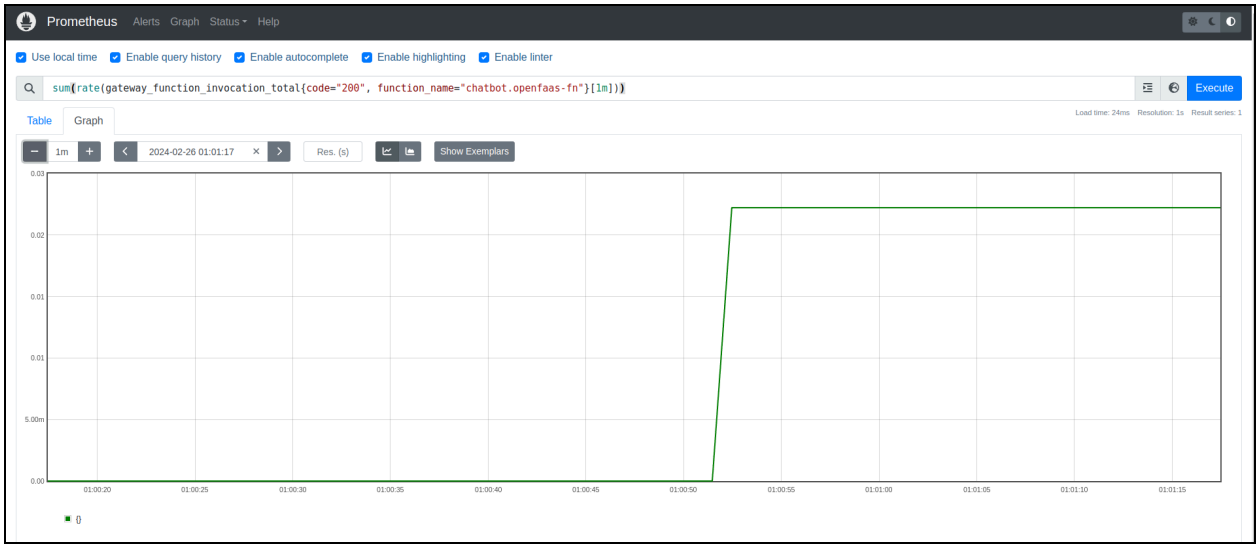


- How would you pass different arguments to the functions?
  - There are two types of arguments we can pass to the function
    - Text
    - Json
  - We can pass the arguments while invoking the openfaas function
    - We can invoke the function on the web UI of the openfaas.

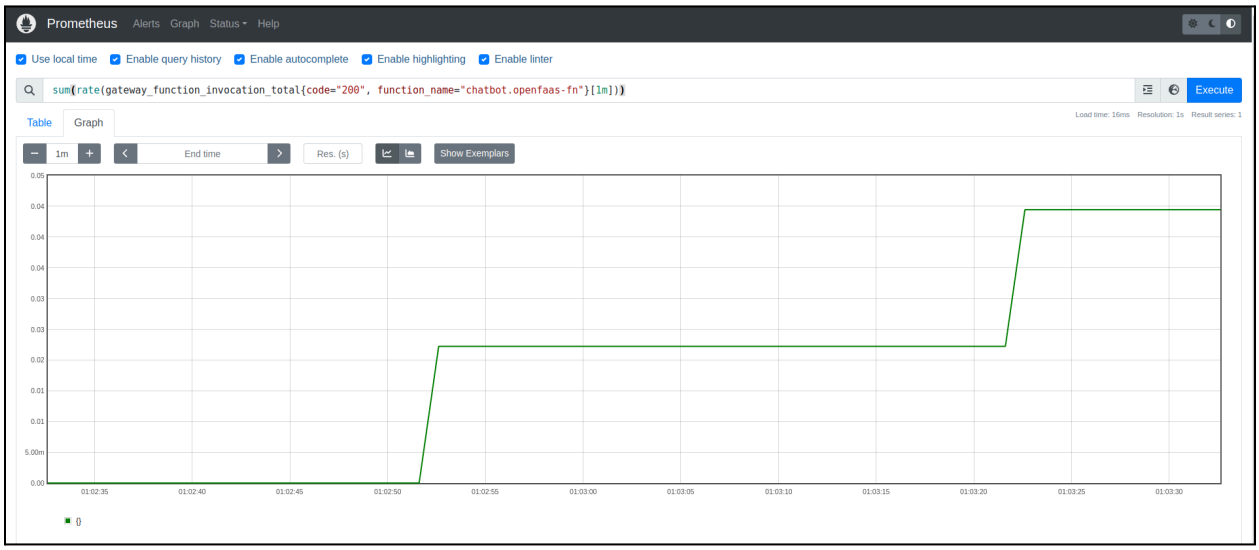
        - We can invoke the function using the curl command and using the -d option we can provide the json or text argument.
        - Another way is by using the faas-cli. We can invoke the function using echo “arguments” | faas-cli invoke <function-name>.
    - We can provide the content type inside the yaml file either we can provide the application/json or text/plain content type.
    - We can also alter the arguments to the function and depending upon the request we can perform the changes inside the handler function to process the text based or the json based request.

- How would you change the slack-interactive function to react to different inputs?
- In order to provide the additional support to function to react to different inputs we can leverage the response object, which serves as a Python dictionary containing the user's input. The function already processes the request as JSON and stores it within the response object, so direct access to the user input becomes feasible.
- The important steps that needs to be performed in the function are:
    1. Accepting the request in either text or JSON format within the handler.py file.
    2. Utilizing the unquote function from the urllib package to decode the request string, typically employing UTF-8 encoding.
    3. Making use of the json.load() method to convert the valid JSON request into a Python dictionary, which is then stored in the response object.
    4. At present the response object remains unused within the function.
    5. To activate the function to respond to varied inputs, extracting the user input data from the response object becomes very important, given its structure as a Python dictionary. While returning json.dumps(data), the function becomes capable of reacting to different inputs based on the data present in the response object.

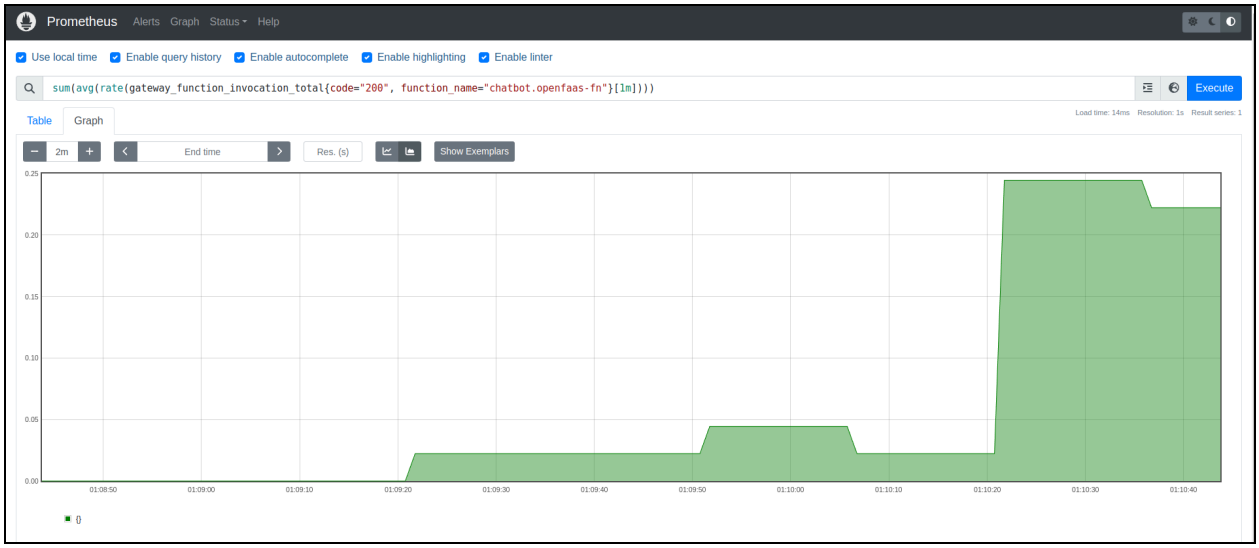
- How long does it take for the chat response to come back?
    - For the first request that does not call figlet
      0.2657 sec



- For the second request that does not call figlet
  0.0069 sec



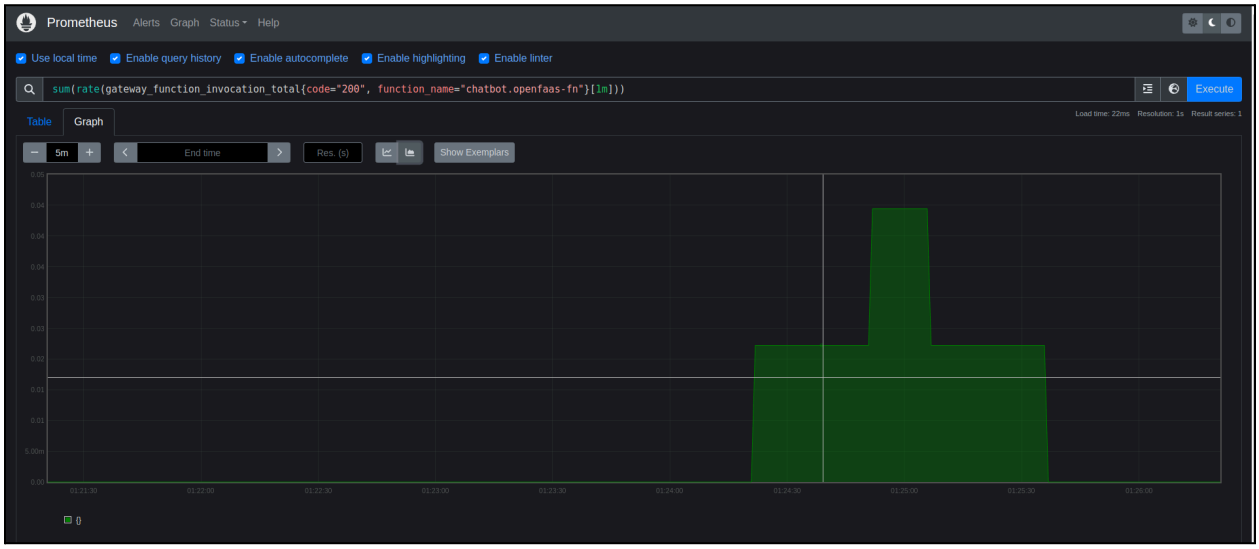
- Average over 10 requests that do not call figlet
  0.6732 sec



- For the first request that calls figlet
  1.0195 sec

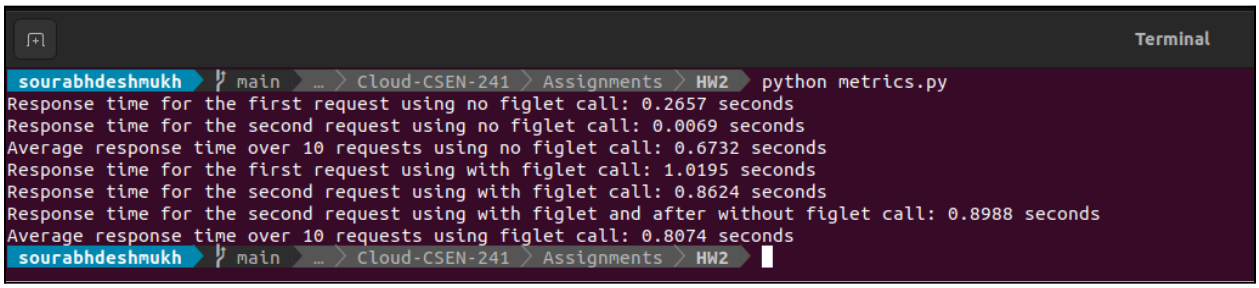
- For the second request that calls figlet
  0.8624 sec

- For the second request that calls figlet that follows the first request that does not call figlet
  0.8988 sec



- Average over 10 requests that do call figlet
  0.8074 sec

Overall metrics result for above questions.



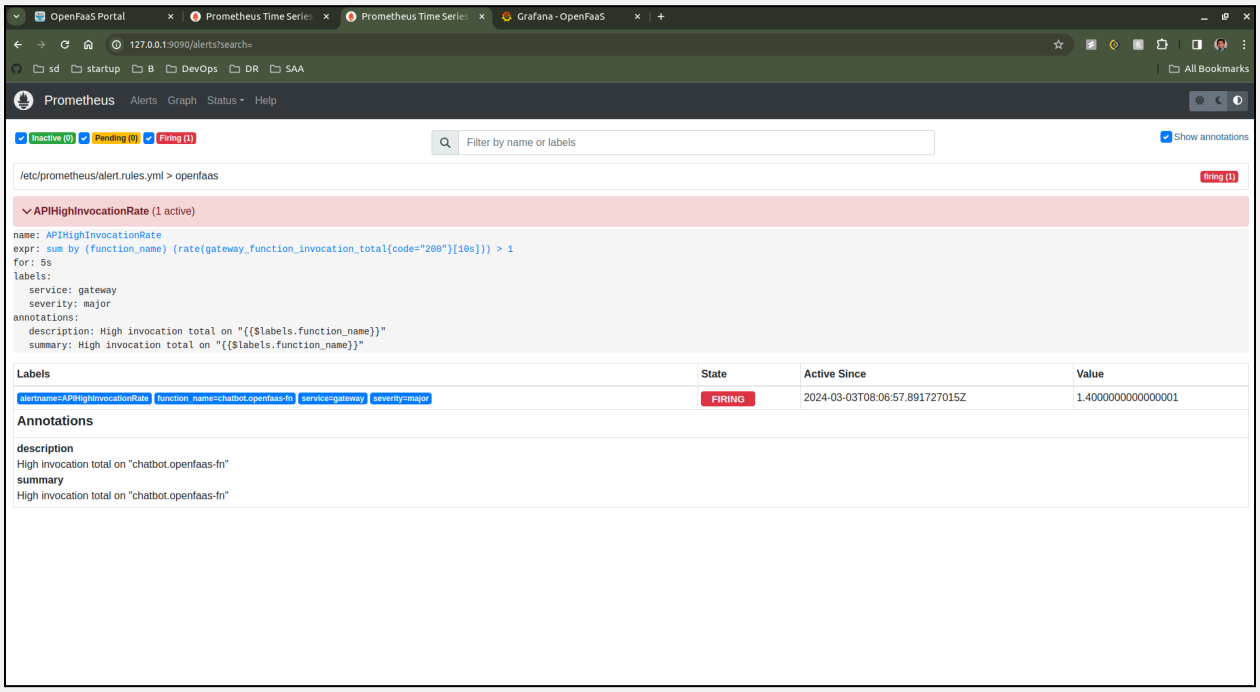
- Now try sending a series of requests to the chatbot in parallel. At what queries per second does OpenFaaS add a new instance of the function?

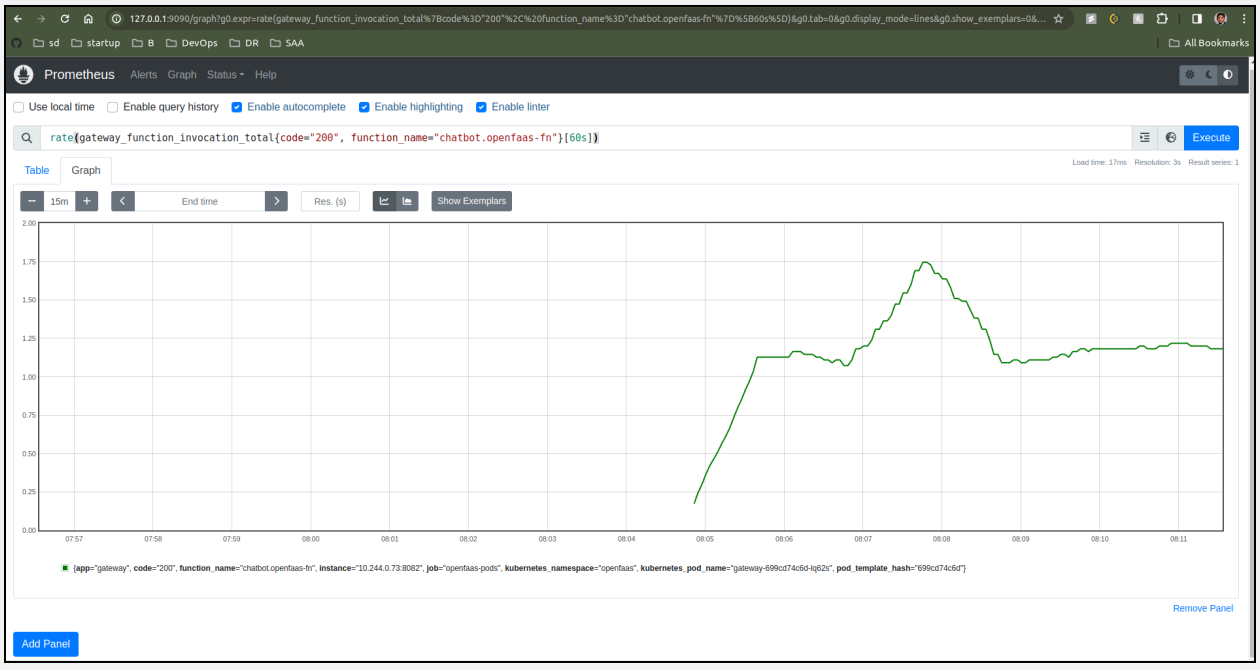
I have used the hey load testing tool to send requests to my function. Below is the command I used to send 10 requests per second for 60 seconds, with a payload of data=time to the /function/chatbot endpoint on localhost port 8080:

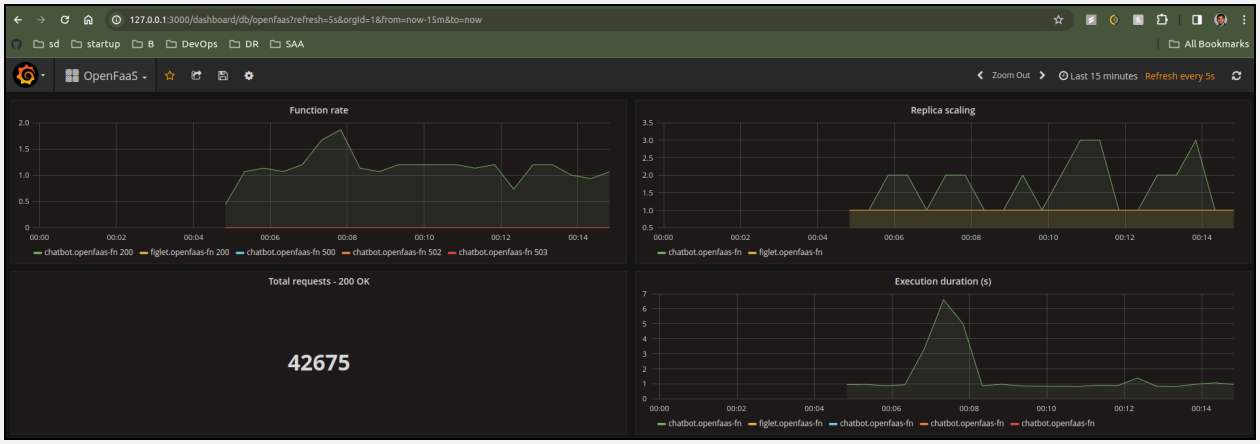
```
sourabhdeshmukh$ hey -z 60s -q 20 -c 10 -m POST -d=time'
http://127.0.0.1:8080/function/chatbot
```

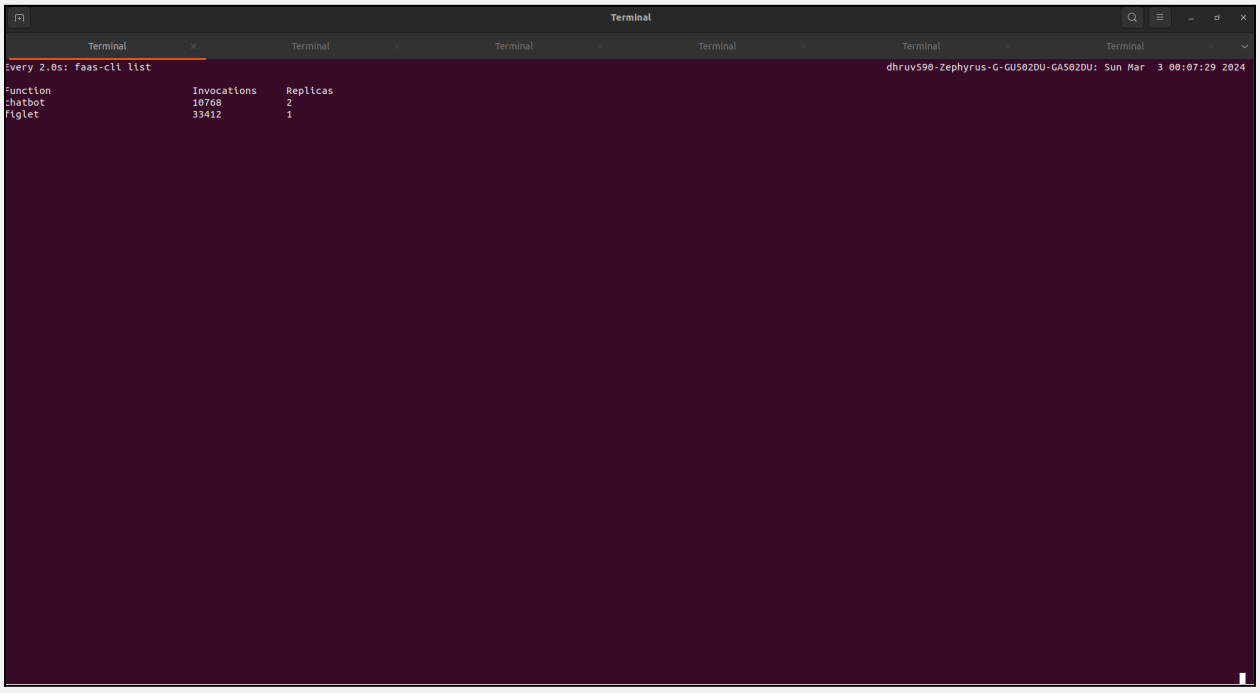
The alert expression in Prometheus Alertmanager is configured to trigger an alert when the function invocation count is greater than 1 over a 5 second period. During my test, the invocation count exceeded 1, which triggered the alert to fire and caused OpenFaas to scale the function by adding another instance. At that point, the queries per second (QPS) was around 1.5425 as the new instance was launching.

Please find below screenshots obtained during testing


OpenFaaS Portal
Prometheus Time Series
Prometheus Time Series
Grafana - OpenFaaS
127.0.0.1:9090/alerts?search=
Prometheus
Alerts
Graph
Status
Help
Inactive (0)
Pending (0)
Firing (1)
Filter by name or labels
Show annotations
/etc/prometheus/alert.rules.yml > openfaas
firing (1)
APIHighInvocationRate (1 active)
name: APIHighInvocationRate
expr: sum by (function_name) (rate(gateway_function_invocation_total{code="200"}[10s])) > 1
for: 5s
labels:
service: gateway
severity: major
annotations:
description: High invocation total on "{{$labels.function_name}}"
summary: High invocation total on "{{$labels.function_name}}"
Labels
State
Active Since
Value
alertname=APIHighInvocationRate
function_name=chatbot.openfaas-fn
service=gateway
severity=major
FIRING
2024-03-03T08:06:57.891727015Z
1.4000000000000001
Annotations
description
High invocation total on "chatbot.openfaas-fn"
summary
High invocation total on "chatbot.openfaas-fn"


127.0.0.1:9090/graph?g0.expr=rate(gateway_function_invocation_total%7Bcode%3D"200"%2C%20function_name%3D"chatbot.openfaas-fn"%7D%5B60s%5D)&g0.tab=0&g0.display_mode=lines&g0.show_exemplars=0&...
Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
rate(gateway_function_invocation_total{code="200", function_name="chatbot.openfaas-fn"}[60s])
Execute
Table
Graph
15m
End time
Res. (s)
Show Exemplars
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
07:57
07:58
07:59
08:00
08:01
08:02
08:03
08:04
08:05
08:06
08:07
08:08
08:09
08:10
08:11
{app="gateway", code="200", function_name="chatbot.openfaas-fn", instance="10.244.0.73:8082", job="openfaas-pods", kubernetes_namespace="openfaas", kubernetes_pod_name="gateway-699cd74c6d-lq62s", pod_template_hash="699cd74c6d"}
Remove Panel
Add Panel


127.0.0.1:3000/dashboard/db/openfaas?refresh=5s&orgId=1&from=now-15m&to=now
sd
startup
B
DevOps
DR
SAA
All Bookmarks
OpenFaaS
Zoom Out
Last 15 minutes
Refresh every 5s
Function rate
chatbot.openfaas-fn 200
figlet.openfaas-fn 200
chatbot.openfaas-fn 500
chatbot.openfaas-fn 502
chatbot.openfaas-fn 503
Replica scaling
chatbot.openfaas-fn
figlet.openfaas-fn
Total requests - 200 OK
42675
Execution duration (s)
chatbot.openfaas-fn
figlet.openfaas-fn
chatbot.openfaas-fn
chatbot.openfaas-fn
chatbot.openfaas-fn


Terminal
Terminal
Every 2.0s: faas-cli list
dhruv590-Zephyrus-G-GU502DU-GA502DU: Sun Mar 3 00:07:29 2024
Function Invocations Replicas
chatbot 10768 2
figlet 33412 1

Extra Credit

Link to the application

https://api.slack.com/apps/A06LA0P690E

Shareable URL for the application

https://slack.com/oauth/v2/authorize?client_id=6709310927441.6690023213014&scope=chat:write,commands,incoming-webhook,mpim:write,im:write&user_scope=

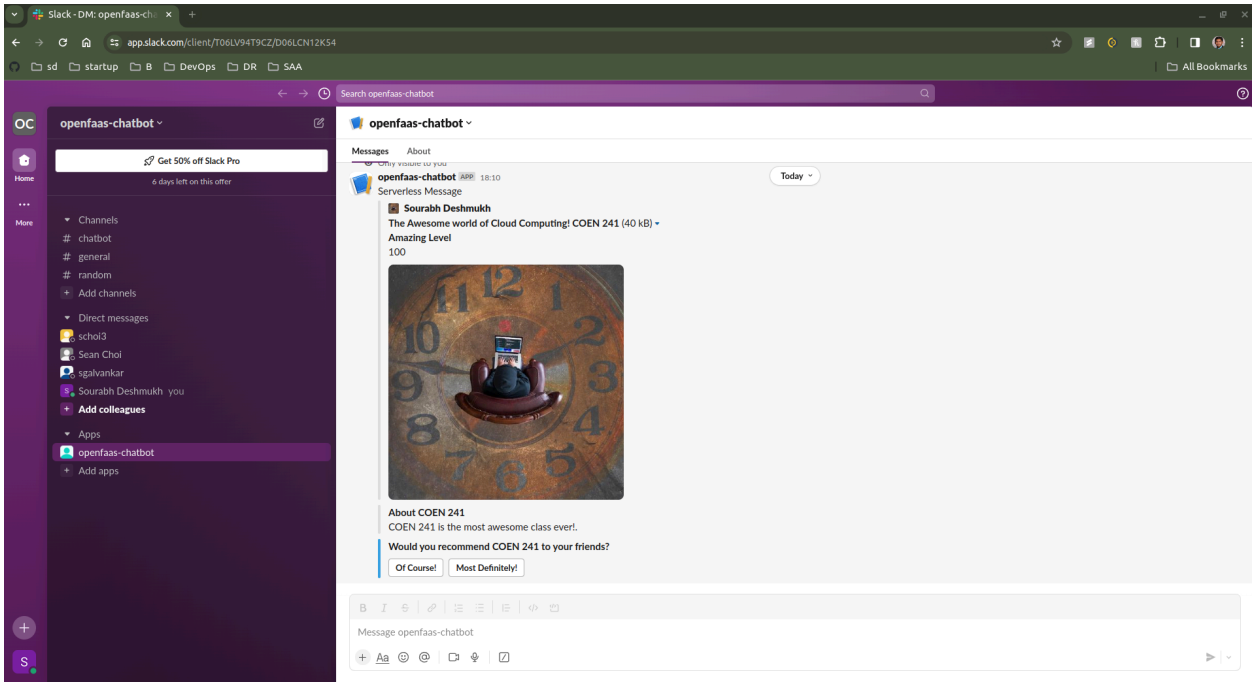
Link to join OpenFaas Chatbot Workspace

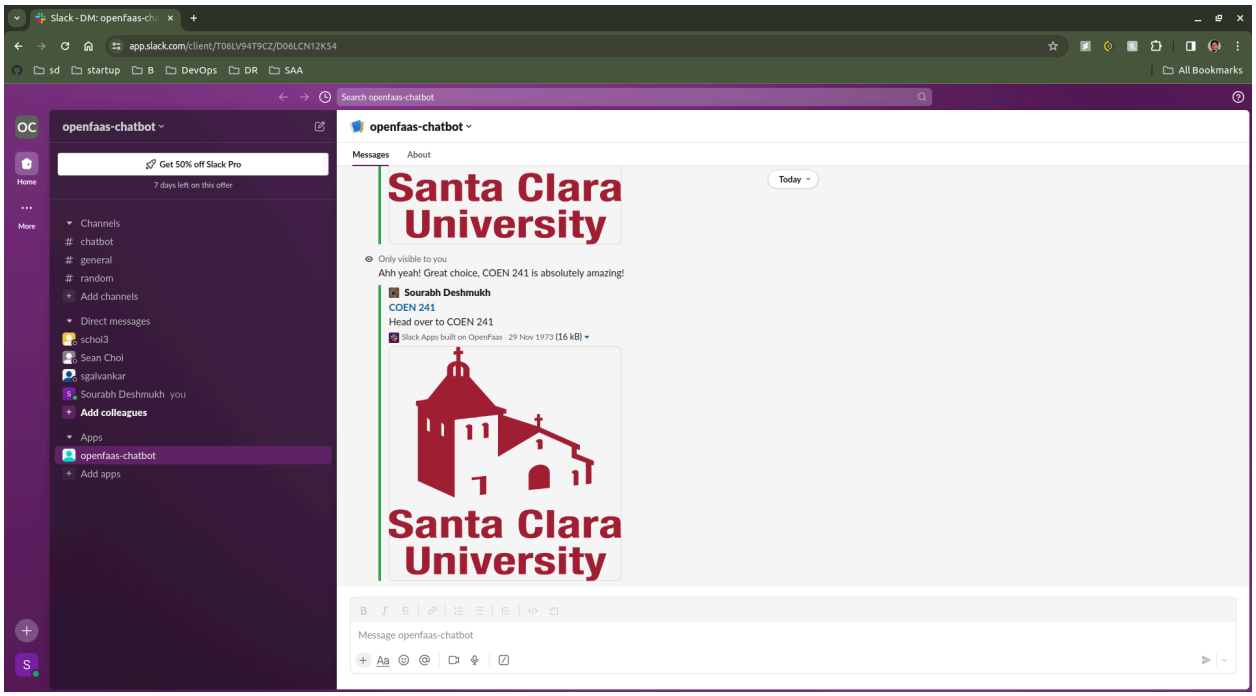
https://join.slack.com/t/openfaas-chatbotgroup/shared_invite/zt-2dm7qis1d-XxjsrZBXDWhgcm~9O2_GAA

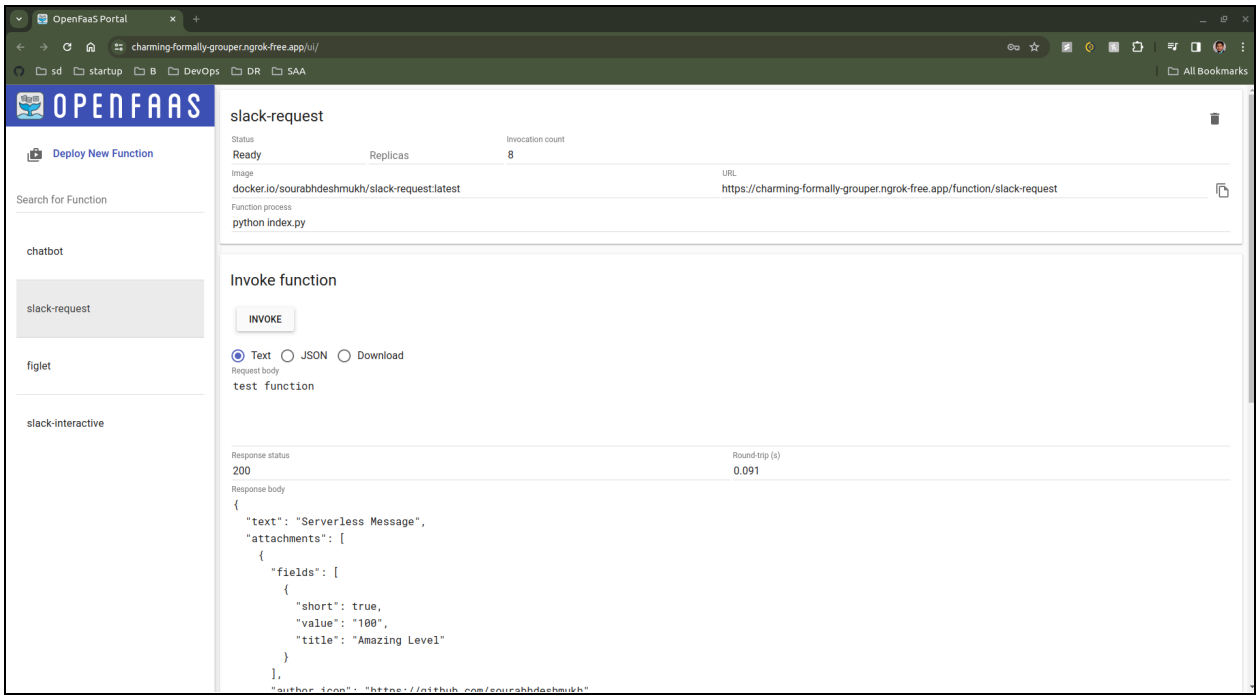
URL for Slash Command (`/coen241`)

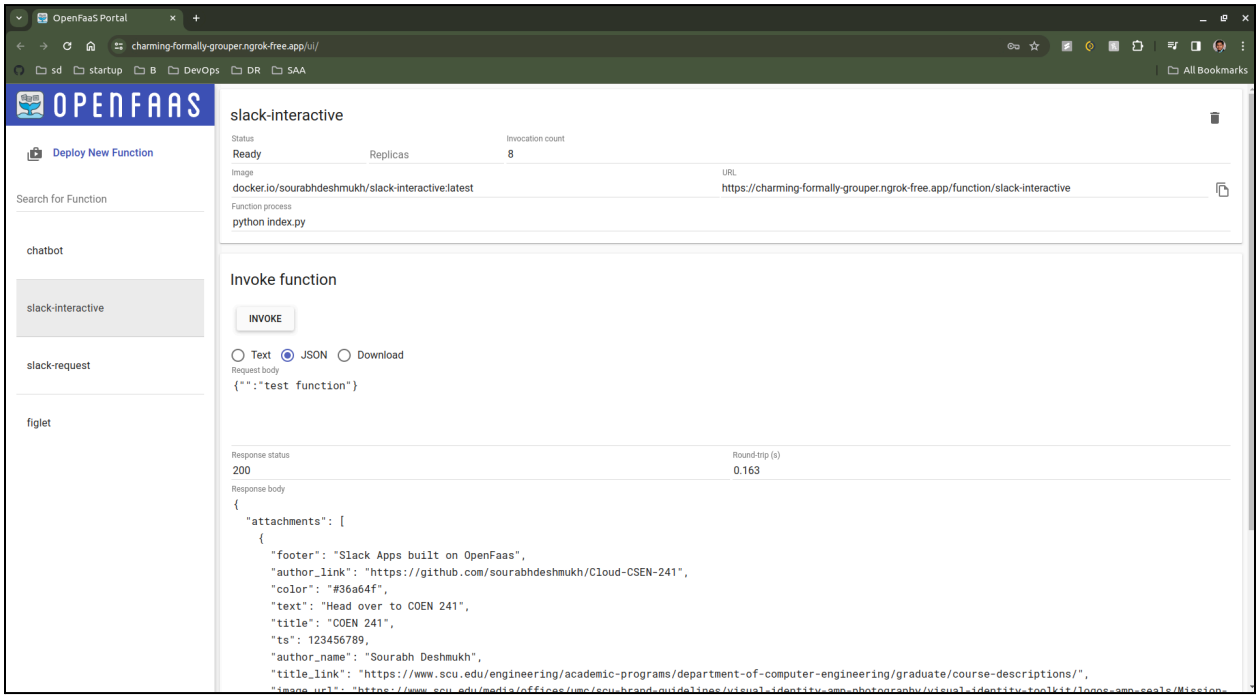
https://charming-formally-grouper.ngrok-free.app/function/slack-request

Slack Application Bot


Slack - DM: openfaas-ch
app.slack.com/client/T06LV94T9CZ/D06LCN12K54
sd
startup
B
DevOps
DR
SAA
All Bookmarks
Search openfaas-chatbot
OC
openfaas-chatbot
Get 50% off Slack Pro
6 days left on this offer
Home
More
Channels
chatbot
general
random
Add channels
Direct messages
schoi3
Sean Choi
sgalvankar
Sourabh Deshmukh you
Add colleagues
Apps
openfaas-chatbot
Add apps
openfaas-chatbot
Messages
About
Today
openfaas-chatbot APP 18:10
Serverless Message
Sourabh Deshmukh
The Awesome world of Cloud Computing! COEN 241 (40 kB)
Amazing Level
100
About COEN 241
COEN 241 is the most awesome class ever!.
Would you recommend COEN 241 to your friends?
Of Course!
Most Definitely!
Message openfaas-chatbot


Slack - DM: openfaas-ch
app.slack.com/client/T06LV94T9CZ/D06LCN12K54
sd
startup
B
DevOps
DR
SAA
All Bookmarks
Search openfaas-chatbot
OC
openfaas-chatbot
Get 50% off Slack Pro
7 days left on this offer
Home
More
Channels
chatbot
general
random
Add channels
Direct messages
schoi3
Sean Choi
sgalvankar
Sourabh Deshmukh you
Add colleagues
Apps
openfaas-chatbot
Add apps
openfaas-chatbot
Messages
About
Today
Santa Clara University
Only visible to you
Ahh yeah! Great choice, COEN 241 is absolutely amazing!
Sourabh Deshmukh
COEN 241
Head over to COEN 241
Slack Apps built on OpenFaas 29 Nov 1973 (16 kB)
Santa Clara University
Message openfaas-chatbot


OpenFaaS Portal
charming-formally-grouper.ngrok-free.app/ui/
sd
startup
B
DevOps
DR
SAA
All Bookmarks
OPENFAAS
Deploy New Function
Search for Function
chatbot
slack-request
figlet
slack-interactive
slack-request
Status
Ready
Replicas
Invocation count
8
Image
docker.io/sourabhdeshmukh/slack-request:latest
URL
https://charming-formally-grouper.ngrok-free.app/function/slack-request
Function process
python index.py
Invoke function
INVOKE
Text
JSON
Download
Request body
test function
Response status
200
Round-trip (s)
0.091
Response body
{
"text": "Serverless Message",
"attachments": [
{
"fields": [
{
"short": true,
"value": "100",
"title": "Amazing Level"
}
],


OpenFaaS Portal
charming-formally-grouper.ngrok-free.app/ui/
sd
startup
B
DevOps
DR
SAA
All Bookmarks
OPENFAAS
Deploy New Function
Search for Function
chatbot
slack-interactive
slack-request
figlet
slack-interactive
Status
Ready
Replicas
Invocation count
8
Image
docker.io/sourabhdeshmukh/slack-interactive:latest
URL
https://charming-formally-grouper.ngrok-free.app/function/slack-interactive
Function process
python index.py
Invoke function
INVOKE
Text
JSON
Download
Request body
{"":"test function"}
Response status
200
Round-trip (s)
0.163
Response body
{
"attachments": [
{
"footer": "Slack Apps built on OpenFaas",
"author_link": "https://github.com/sourabhdeshmukh/Cloud-CSEN-241",
"color": "#36a64f",
"text": "Head over to COEN 241",
"title": "COEN 241",
"ts": 123456789,
"author_name": "Sourabh Deshmukh",
"title_link": "https://www.scu.edu/engineering/academic-programs/department-of-computer-engineering/graduate/course-descriptions/",