

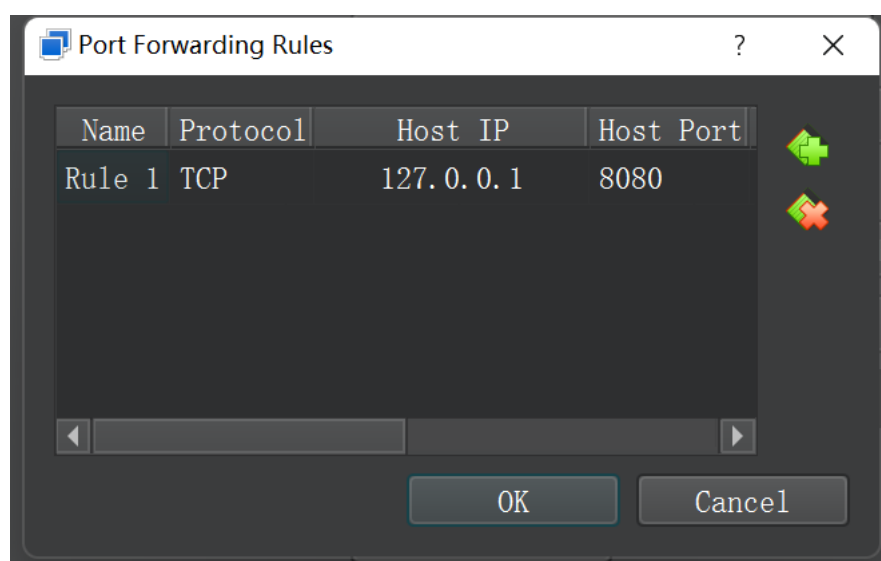
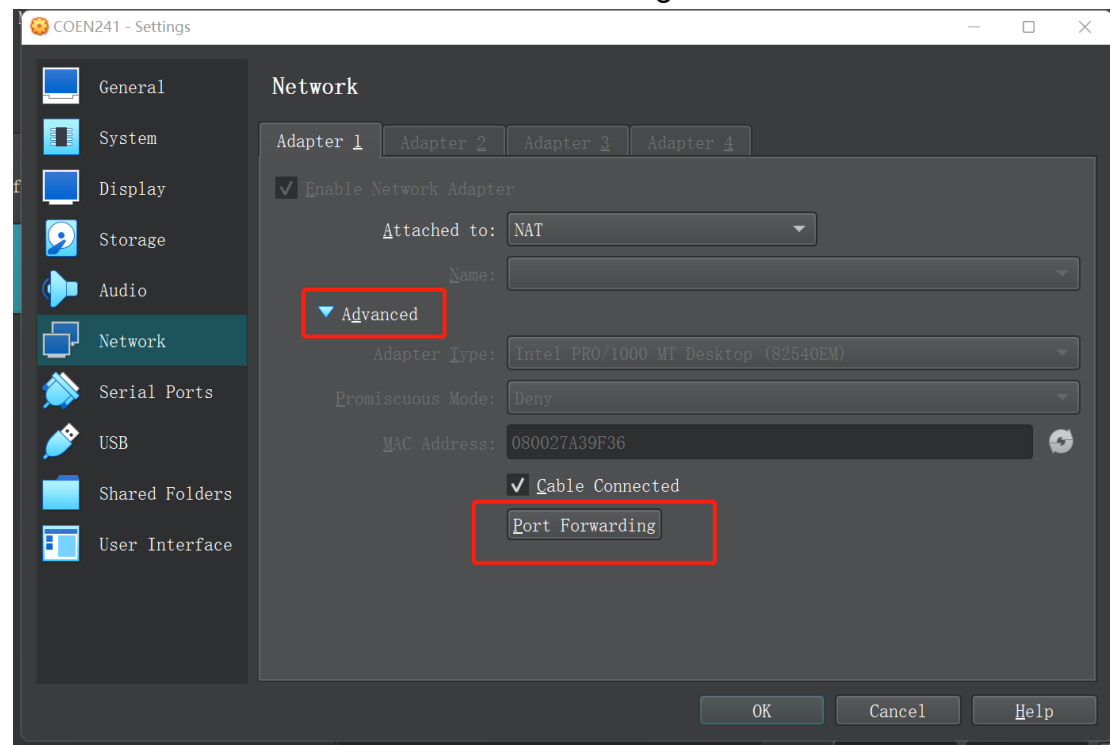
COEN 241 HW 2 Your Own Serverless Infrastructure

Student Name: Mingyang Wu

Student ID: 00001628984

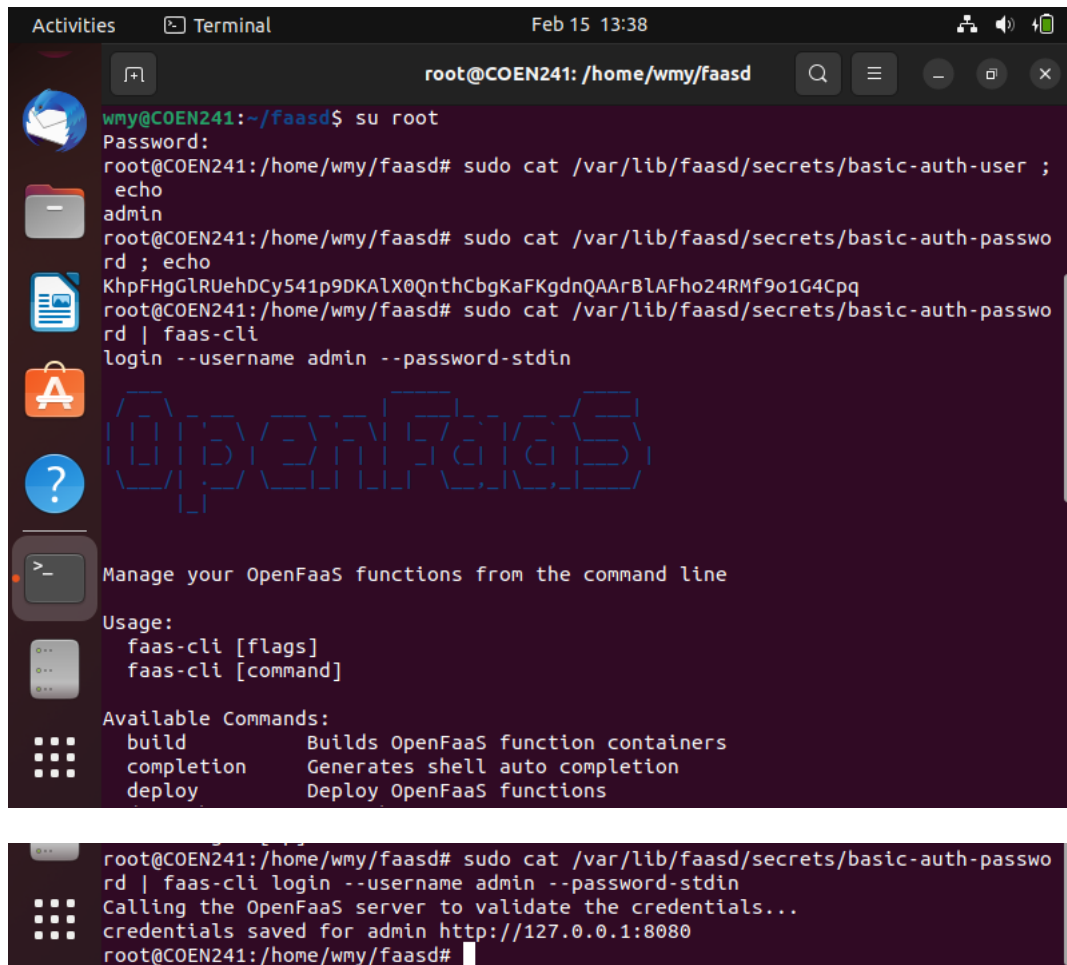
In this homework, I used VirtualBox on Windows Host OS.

Before this homework, we should adjust some configurations in VirtualBox: Port Forwarding. We need to forward the port from the host to the guest VM's port 8080 before we check the OpenFaaS gateway UI is running correctly on port 8080 in the localhost of the VM we are running:



After this, we can get into 127.0.0.1:8080 with the password (a long characters) and username (admin) we got by using commands:

```
$ sudo cat /var/lib/faasd/secrets/basic-auth-password
$ sudo cat /var/lib/faasd/secrets/basic-auth-user
$ sudo cat /var/lib/faasd/secrets/basic-auth-password | faas-cli login --
username admin --password-stdin
```



The screenshot shows a terminal window titled 'Terminal' with the date 'Feb 15 13:38'. The user is logged in as 'wmy' at 'COEN241' and is in the directory '/home/wmy/faasd'. The user runs 'su root' to become root. Then, they run 'sudo cat /var/lib/faasd/secrets/basic-auth-user ; echo' which outputs 'admin'. Next, they run 'sudo cat /var/lib/faasd/secrets/basic-auth-password ; echo' which outputs a long alphanumeric string. Finally, they run 'sudo cat /var/lib/faasd/secrets/basic-auth-password | faas-cli login --username admin --password-stdin'. This triggers the OpenFaaS CLI login process, displaying the 'OpenFaaS' logo and usage information. The usage shows 'faas-cli [flags]' and 'faas-cli [command]'. Available commands listed are 'build' (Builds OpenFaaS function containers), 'completion' (Generates shell auto completion), and 'deploy' (Deploy OpenFaaS functions). The login process then shows 'Calling the OpenFaaS server to validate the credentials...' and 'credentials saved for admin http://127.0.0.1:8080'.

```
root@COEN241: /home/wmy/faasd
wmy@COEN241:~/faasd$ su root
Password:
root@COEN241:/home/wmy/faasd# sudo cat /var/lib/faasd/secrets/basic-auth-user ;
echo
admin
root@COEN241:/home/wmy/faasd# sudo cat /var/lib/faasd/secrets/basic-auth-passwo
rd ; echo
KhpFHgGlrUehDCy541p9DKAlX0QnthCbgKaFKgdnQAARBlAFho24RMf9o1G4Cpq
root@COEN241:/home/wmy/faasd# sudo cat /var/lib/faasd/secrets/basic-auth-passwo
rd | faas-cli
login --username admin --password-stdin

OpenFaaS

Manage your OpenFaaS functions from the command line

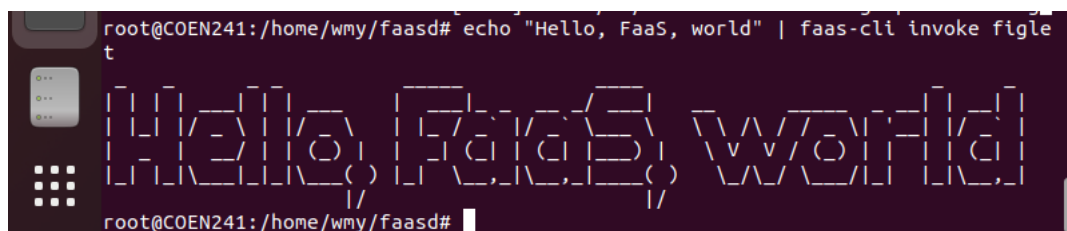
Usage:
  faas-cli [flags]
  faas-cli [command]

Available Commands:
  build      Builds OpenFaaS function containers
  completion Generates shell auto completion
  deploy     Deploy OpenFaaS functions

root@COEN241:/home/wmy/faasd# sudo cat /var/lib/faasd/secrets/basic-auth-passwo
rd | faas-cli login --username admin --password-stdin
Calling the OpenFaaS server to validate the credentials...
credentials saved for admin http://127.0.0.1:8080
root@COEN241:/home/wmy/faasd#
```

Tasks

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



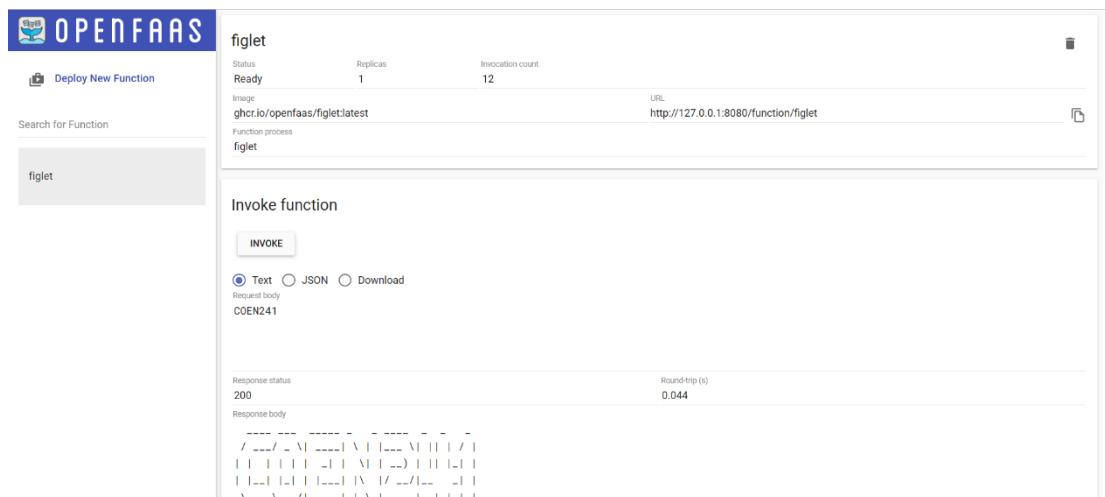
The screenshot shows a terminal window with the user 'root' at 'COEN241' in the directory '/home/wmy/faasd'. The user runs the command 'echo "Hello, FaaS, world" | faas-cli invoke figlet'. The output of the command is 'Hello, FaaS, world' displayed in a large, stylized font.

```
root@COEN241:/home/wmy/faasd# echo "Hello, FaaS, world" | faas-cli invoke figle
t
Hello, FaaS, world
root@COEN241:/home/wmy/faasd#
```

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

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

```
root@COEN241:/home/wmy/faasd# sudo journalctl -u faasd --lines 40
Feb 16 10:05:22 COEN241 faasd[1239]: 2023/02/16 10:05:22 - gateway
Feb 16 10:05:22 COEN241 faasd[1239]: 2023/02/16 10:05:22 - queue-worker
Feb 16 10:05:22 COEN241 faasd[1239]: Starting: prometheus
Feb 16 10:05:22 COEN241 faasd[1239]: Creating local directory: /var/lib/faasd/>
Feb 16 10:05:22 COEN241 faasd[1239]: 2023/02/16 10:05:22 Running prometheus wi>
Feb 16 10:05:23 COEN241 faasd[1239]: 2023/02/16 10:05:23 Created container: pr>
Feb 16 10:05:30 COEN241 faasd[1239]: 2023/02/16 10:05:30 prometheus has IP: 10>
Feb 16 10:05:30 COEN241 faasd[1239]: 2023/02/16 10:05:30 Task: prometheus >
Feb 16 10:05:31 COEN241 faasd[1239]: Starting: basic-auth-plugin
Feb 16 10:05:32 COEN241 faasd[1239]: 2023/02/16 10:05:32 Created container: ba>
Feb 16 10:05:33 COEN241 faasd[1239]: 2023/02/16 10:05:33 basic-auth-plugin has>
Feb 16 10:05:33 COEN241 faasd[1239]: 2023/02/16 10:05:33 Task: basic-auth-plug>
Feb 16 10:05:33 COEN241 faasd[1239]: Starting: nats
Feb 16 10:05:33 COEN241 faasd[1239]: Creating local directory: /var/lib/faasd/>
Feb 16 10:05:33 COEN241 faasd[1239]: 2023/02/16 10:05:33 Running nats with use>
Feb 16 10:05:33 COEN241 faasd[1239]: 2023/02/16 10:05:33 Created container: na>
Feb 16 10:05:34 COEN241 faasd[1239]: 2023/02/16 10:05:34 nats has IP: 10.62.0.4>
Feb 16 10:05:34 COEN241 faasd[1239]: 2023/02/16 10:05:34 Task: nats Con>
Feb 16 10:05:34 COEN241 faasd[1239]: Starting: gateway
Feb 16 10:05:35 COEN241 faasd[1239]: 2023/02/16 10:05:35 Created container: ga>
Feb 16 10:05:35 COEN241 faasd[1239]: 2023/02/16 10:05:35 gateway has IP: 10.62>
Feb 16 10:05:35 COEN241 faasd[1239]: 2023/02/16 10:05:35 Task: gateway >
Feb 16 10:05:36 COEN241 faasd[1239]: Starting: queue-worker
Feb 16 10:05:36 COEN241 faasd[1239]: 2023/02/16 10:05:36 Created container: qu>
Feb 16 10:05:37 COEN241 faasd[1239]: 2023/02/16 10:05:37 queue-worker has IP: >
Feb 16 10:05:37 COEN241 faasd[1239]: 2023/02/16 10:05:37 Task: queue-worker >
```





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

```
root@COEN241:/home/wmy/faasd# faas-cli store deploy figlet
Deployed. 200 OK.
URL: http://127.0.0.1:8080/function/figlet

root@COEN241:/home/wmy/faasd# faas-cli store inspect figlet
Info for: Figlet

Name      figlet
Description Generate ASCII logos with the figlet CLI
Image     ghcr.io/openfaas/figlet:latest
Process   figlet
Repo URL  https://github.com/openfaas/store-functions
```

**OPENFAAS**

 **Deploy New Function**

Search for Function

slack-interactive

slack-request

figlet



Select a function or deploy a new one.

DEPLOY NEW FUNCTION

Or use **faas-cli** to build and deploy functions:

```
$ curl -sSL https://cli.openfaas.com | sudo sh
```

Wasm is a fast, light alternative to Linux containers – try it out today with the Docker+Wasm Beta. ×

 **docker hub** [Explore](#) [Repositories](#) [Organizations](#) [Help](#) Upgrade  mingyang1998 ▼

mingyang1998 ▼ ▼ **Create repository**

mingyang1998 / **slack-request**
Contains: image | Last pushed: 4 minutes ago

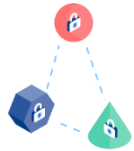
Not Scanned ☆ 0 ⬇ 1 Public


mingyang1998 / **slack-interactive**
Contains: image | Last pushed: 9 minutes ago

Not Scanned ☆ 0 ⬇ 2 Public

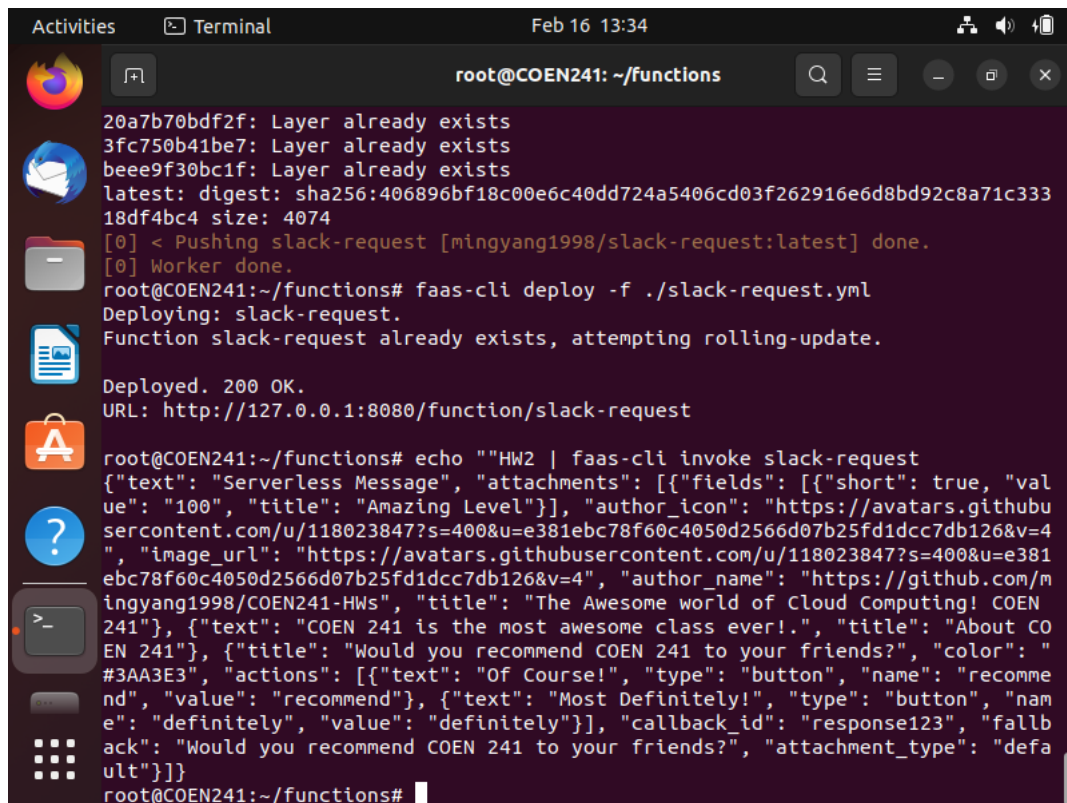
mingyang1998 / **docker101tutorial**
Contains: image | Last pushed: 23 days ago

Not Scanned ☆ 0 ⬇ 2 Public


Create an Organization
Manage Docker Hub repositories
with your team


community

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

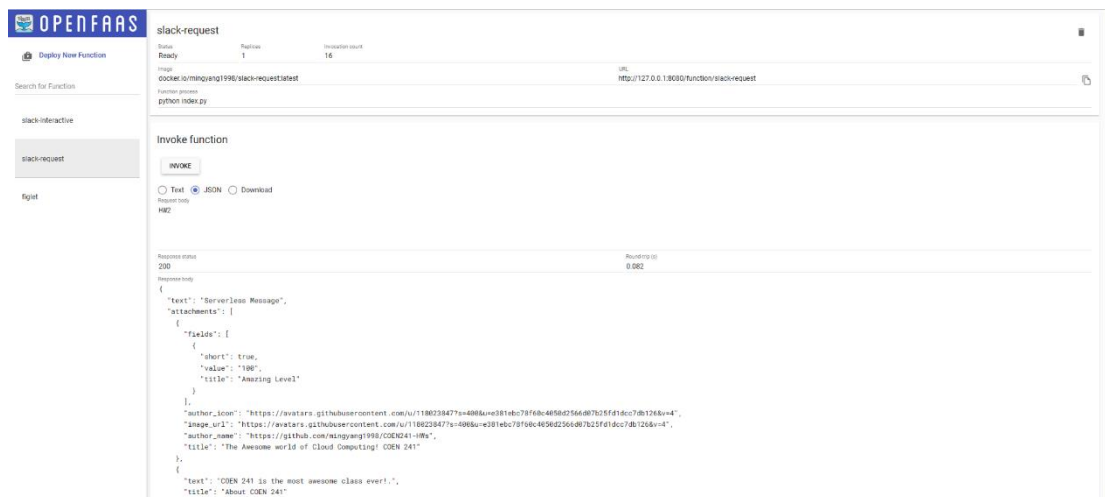


```
Activities Terminal Feb 16 13:34
root@COEN241: ~/functions

20a7b70bdf2f: Layer already exists
3fc750b41be7: Layer already exists
beee9f30bc1f: Layer already exists
latest: digest: sha256:406896bf18c00e6c40dd724a5406cd03f262916e6d8bd92c8a71c33318df4bc4 size: 4074
[0] < Pushing slack-request [mingyang1998/slack-request:latest] done.
[0] Worker done.
root@COEN241:~/functions# faas-cli deploy -f ./slack-request.yml
Deploying: slack-request.
Function slack-request already exists, attempting rolling-update.

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

root@COEN241:~/functions# echo '"HW2 | faas-cli invoke slack-request
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "https://avatars.githubusercontent.com/u/118023847?s=400&u=e381ebc78f60c4050d2566d07b25fd1dcc7db126&v=4", "image_url": "https://avatars.githubusercontent.com/u/118023847?s=400&u=e381ebc78f60c4050d2566d07b25fd1dcc7db126&v=4", "author_name": "https://github.com/mingyang1998/COEN241-HWs", "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"}, {"text": "Most Definitely!", "type": "button", "name": "definitely", "value": "definitely"}], "callback_id": "response123", "fallback": "Would you recommend COEN 241 to your friends?", "attachment_type": "default"}]}'
root@COEN241:~/functions#
```



OPENFAAS

Deploy New Function

Search for Functions

slack-interactive

slack-request

slack-request

Invoke function

INVOKE

☐ Text ☒ JSON ☐ Download

Response only


HW2


Function status: 200 Response time: 0.002s

Response body:

```
{
  "text": "Serverless Message",
  "attachments": [
    {
      "fields": [
        {
          "short": true,
          "value": "100",
          "title": "Amazing Level"
        }
      ],
      "author_icon": "https://avatars.githubusercontent.com/u/118023847?s=400&u=e381ebc78f60c4050d2566d07b25fd1dcc7db126&v=4",
      "image_url": "https://avatars.githubusercontent.com/u/118023847?s=400&u=e381ebc78f60c4050d2566d07b25fd1dcc7db126&v=4",
      "author_name": "https://github.com/mingyang1998/COEN241-HWs",
      "title": "The Awesome world of Cloud Computing! COEN 241"
    },
    {
      "text": "COEN 241 is the most awesome class ever!.",
      "title": "About COEN 241"
    }
  ]
}
```

Slack interactive:

**OPENFAAS**

 [Deploy New Function](#)

Search for Function

slack-request

slack-interactive

figlet

slack-interactive

Status: Ready Replicas: 1 Invocation count: 15

Image: docker.io/mingyang1998/slack-interactive:latest URL: http://127.0.0.1:8080/function/slack-interactive

Function process: python index.py

Invoke function

☐ Text ☒ JSON ☐ Download

Request body: "HW2"

Response status: 200 Round trip (s): 0.168

Response body: { "attachments": [{ "footer": "Slack Apps built on OpenFaaS", "author_link": "https://github.com/mingyang1998/COEN241-HWs", "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/118023847?s=400&u=e381ebc78f60c4050d2566d07b25fd1dc07db126&v=4" }] }

Questions:

1. What is the command to invoke the slack-request function (4 pts)?

a. Via Curl:

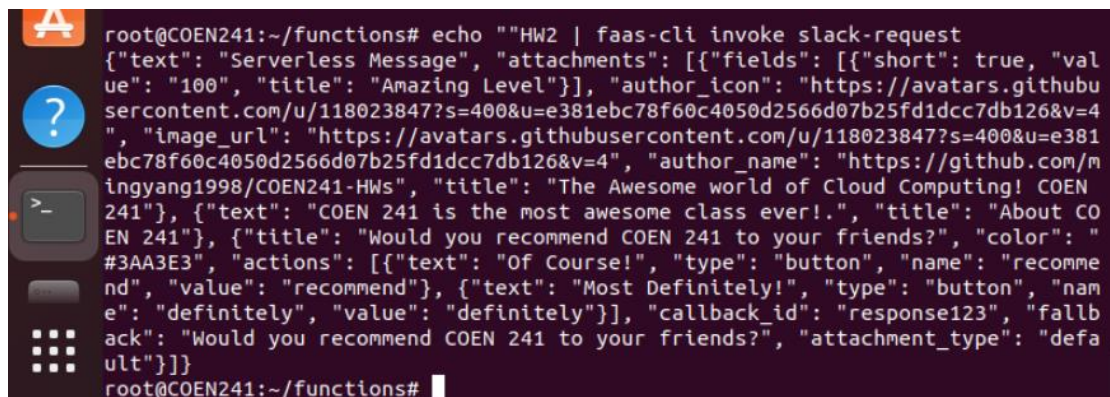
```
$ curl http://127.0.0.1:8080/function/slack-request
```

b. Via faas-cli:

```
$ echo ""HW2 | faas-cli invoke slack-request
```

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

I see JSON responses as returned by the slack-request/handler.py file.



```
root@COEN241:~/functions# echo ""HW2 | faas-cli invoke slack-request
{"text": "Serverless Message", "attachments": [{"fields": [{"short": true, "value": "100", "title": "Amazing Level"}], "author_icon": "https://avatars.githubusercontent.com/u/118023847?s=400&u=e381ebc78f60c4050d2566d07b25fd1dcc7db126&v=4", "image_url": "https://avatars.githubusercontent.com/u/118023847?s=400&u=e381ebc78f60c4050d2566d07b25fd1dcc7db126&v=4", "author_name": "https://github.com/mingyang1998/COEN241-HWs", "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"}]}
root@COEN241:~/functions#
```

3. What is the command to invoke the slack-interactive function? (4 pts)

a. Via curl:

```
$ curl http://127.0.0.1:8080/function/slack-interactive
```

b. Via faas-cli:

```
$ echo "HW2" | faas-cli invoke slack-interactive
```

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

I see JSON responses as returned by the slack-request/handler.py file, the format is similar to slack-request/handler.py file.

5. How would you pass different arguments to the functions? (4 pts)

I may try to invoke the functions by pass different arguments:

```
$ echo <type what you want here> | faas-cli invoke function
```

6. How would you change the slack-interactive function to react to different inputs? (4 pts)

Parsing JSON data:

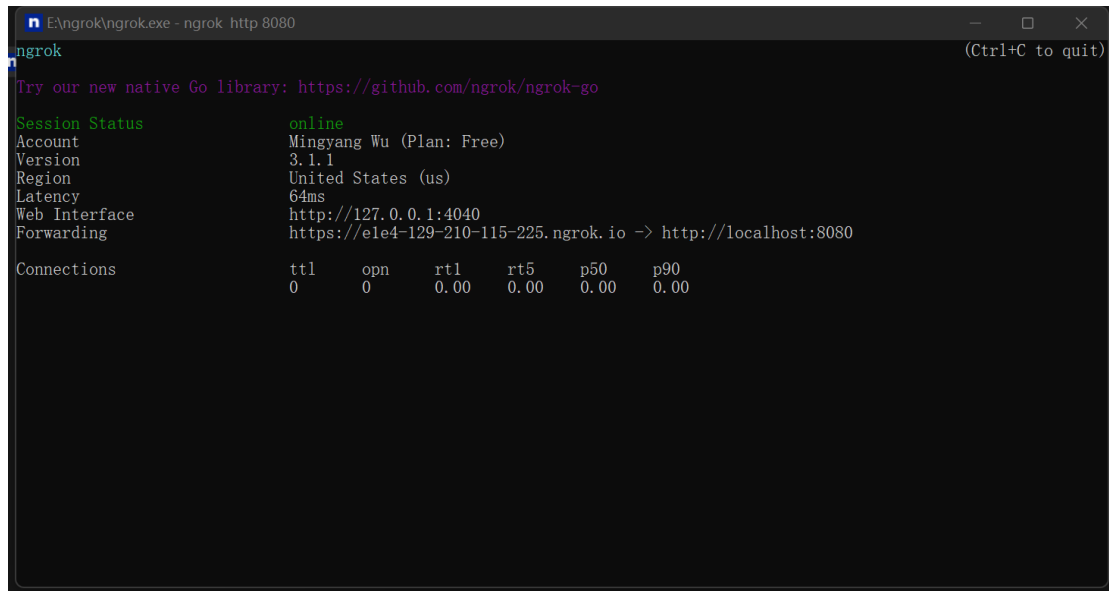
The Slack-interactive function can receive JSON data that contains information about the user's interaction. We can parse this data in our code to extract the relevant information, such as the value of a selected option or the text entered into an input field. Based on this information, we can trigger different actions.

Extra Credit

In this part, I do not have a public routable address, so I download ngrok and use the command:

ngrok http 8080

to get a public routable address. The following is the screenshot:

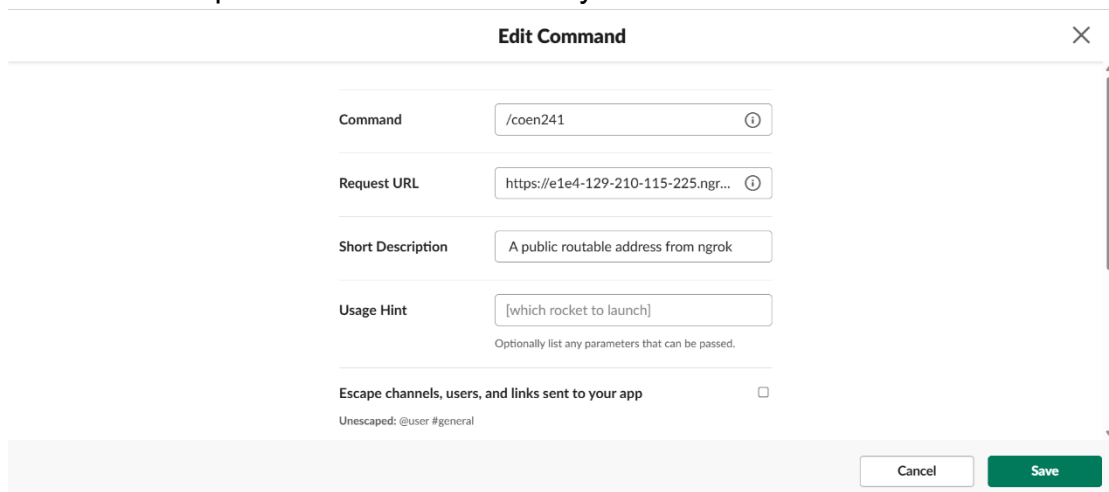


```
E:\ngrok\ngrok.exe - ngrok http 8080
ngrok
Try our new native Go library: https://github.com/ngrok/ngrok-go

Session Status      online
Account             Mingyang Wu (Plan: Free)
Version             3.1.1
Region              United States (us)
Latency              64ms
Web Interface        http://127.0.0.1:4040
Forwarding           https://e1e4-129-210-115-225.ngrok.io -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p90
                   0      0      0.00   0.00   0.00   0.00
```

After I got the public IP, I set the request url to the public endpoint of my function and set the request url for the interactivity:



Edit Command [X]

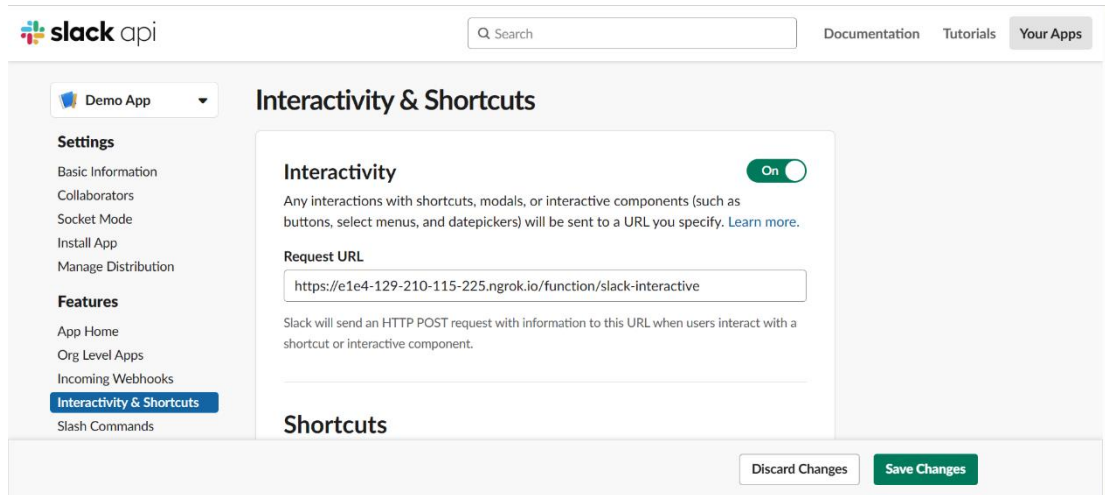
Command:

Request URL:

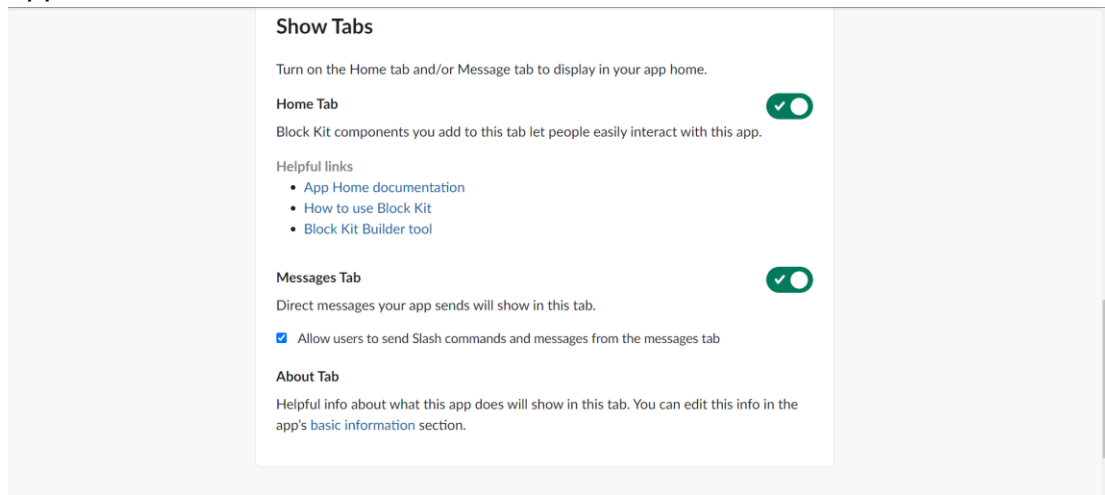
Short Description:

Usage Hint:
Optionally list any parameters that can be passed.

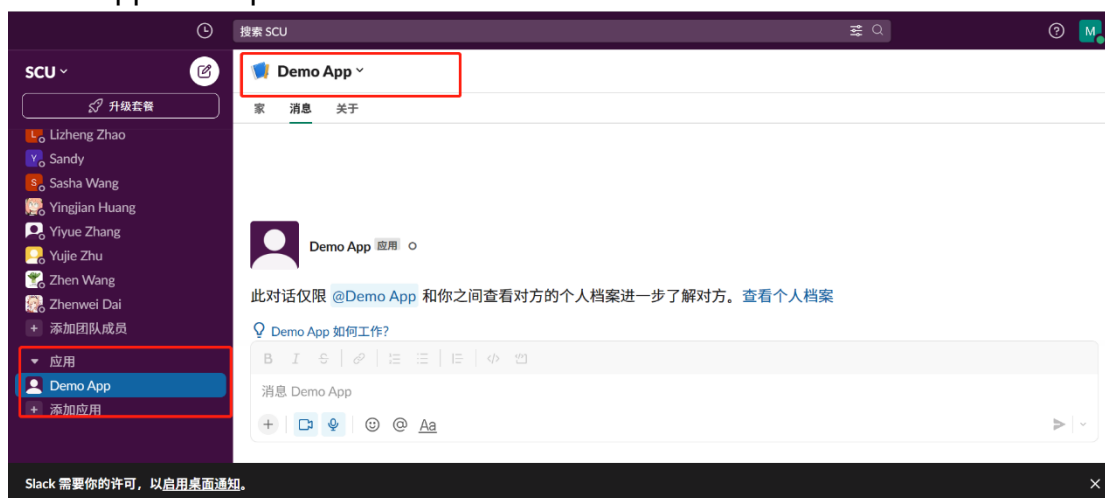
Escape channels, users, and links sent to your app ☐
Unescaped: @user #general



Before we chat with the bot in slack, we should change one configuration in the app we created:



Slack App Workspace screenshot:



Slack App URL: <https://app.slack.com/client/T034EHXG87M/D04Q6S2B8MQ>

