

5. Minio를 stroge로 사용하기

- 5OpenFaaS에서는 아직 Function 내부에서 hots의 파일이나 디렉터리에 마운트 하는 것을 지원하지 않음
- 따라서 S3, Minio와 같은 Object Storage를 사용해야 함

Minio 설치하기

- 공식 문서 : <https://docs.min.io/minio/k8s/index.html>

? 오픈소스 Object Strage로 셋업이 쉽고 API를 통해 쉽게 데이터를 업로드, 다운로드할 수 있음.

! Linux amd64 버전으로 설치

- MINIO Server 설치

```
$ wget https://dl.min.io/server/minio/release/linux-amd64/minio
$ chmod +x minio

$ tr -dc A-Za-z0-9 </dev/urandom | head -c 32
# V4j7SPWqYQNAdmDyqSVGEbZSStX0osLf

# 디렉터리가 없는 경우
# sudo mkdir /mnt/data
# sudo mkdir /data

# 권한이 없는 경우
# sudo chown -R mgmw /mnt/data && sudo chmod u+rxw /mnt/data
# sudo chown -R mgmw /data && sudo chmod u+rxw /data

# MINIO server 실행
$ MINIO_ROOT_USER=admin MINIO_ROOT_PASSWORD=V4j7SPWqYQNAdmDyqSVGEbZSStX0osLf ./minio serve
r /data --console-address ":9001" &
```

Minio를 사용해 함수 작성하기

- 참고 문서 : <https://blog.alexellis.io/openfaas-storage-for-your-functions/>

- 작성할 내용

🔥 loadimages : URL로부터 이미지를 다운로드받아 incoming 디렉터리에 저장

🔥 processimages : incoming bucket에서 이미지를 다운로드받아 흑백으로 처리하고 processed bucket에 업로드

- Bucket 만들기 (Python API 사용)

```
$ python3
>>> from minio import Minio
>>> mc = Minio("127.0.0.1:9000", access_key="admin", secret_key="V4j7SPWqYqNAdmDyqSVGEbZSS
tX0osLf", secure=False)
>>> mc.make_bucket("incoming")
>>> mc.make_bucket("processed")
>>> buckets = mc.list_buckets()
>>> bucket_list = [bucket.name for bucket in buckets]
>>> bucket_list
```

🔥 UI에서도 생성된 bucket 확인 가능!

- loadimages 함수 만들기

✓ 디렉터리 만들기

```
$ mkdir loadimages && cd loadimages
$ faas-cli new --lang python3 loadimages --prefix="127.0.0.1:5000"
```

✓ Dependency 추가

```
# loadimages/requirements.txt

requests
minio
```

✓ Input & Output

[Input : JSON]

```
{
  "urls": [
    "url1",
    "url2",
    ...
  ]
}
```

[Output: JSON]

```
[
  "filename1",
  "filename2",
  ...
]
```

✓ 함수 작성하기

```
# loadimages/handler.py

from minio import Minio
import requests
import json
import uuid
import os

def handle(st):
    req = json.loads(st)

    mc = Minio(os.environ['minio_hostname'],
               access_key=os.environ['minio_access_key'],
               secret_key=os.environ['minio_secret_key'],
               secure=False)

    names = []
    for url in req["urls"]:
        names.append(download_push(url, mc))
    print(json.dumps(names))

def download_push(url, mc):
    # download file
    r = requests.get(url)

    # write to temporary file
    file_name = get_temp_file()
    f = open("/tmp/" + file_name, "wb")
    f.write(r.content)
```

```

f.close()

# sync to Minio
mc.fput_object("incoming", file_name, "/tmp/"+file_name)
return file_name

def get_temp_file():
    uuid_value = str(uuid.uuid4())
    return uuid_value

```

✓ YAML 수정하기

```

# loadimages.yml

...
functions:
  loadimages:
    ...
    environment:
      minio_hostname: [Your IP]:9000
      minio_access_key: admin
      minio_secret_key: V4j7SPWAYQNAdmDyqSVGEbZSStX0osLf
      write_debug: true

```

✓ 빌드, 푸시, 배포

```

$ faas-cli up -f ./loadimages.yml --gateway http://127.0.0.1:9888
$ kubectl get pods -n openfaas-fn

```

✓ 테스트

```

$ echo '{
  "urls": [
    "https://images.pexels.com/photos/72161/pexels-photo-72161.jpeg?dl&fit=crop&w=640&h=318",
    "https://images.pexels.com/photos/382167/pexels-photo-382167.jpeg?dl&fit=crop&w=640&h=337"]
}' | faas invoke loadimages --gateway http://127.0.0.1:9888

```

```

# ["78a430c7-9295-495f-aa20-7221be5c6fdb", "bc68ca5a-a617-4f2e-9220-f7d9ea6e4740"]

```

- processimages 함수 만들기

✓ 디렉터리 만들기

```
$ mkdir processimages && cd processimages  
$ faas-cli new --lang python3 processimages --prefix="127.0.0.1:5000"
```

✓ Dependency 추가

```
# processimages/requirements.txt  
  
requests  
minio
```

✓ Input & Output

[Input : JSON]

```
[  
  "filename1",  
  "filename2",  
  ...  
]
```

[Output: JSON]

```
[  
  "filename3",  
  "filename4",  
  ...  
]
```

✓ 함수 작성하기

```
# convertbw(흑백으로 변환하는 함수) 배포하기  
  
$ faas-cli deploy --name convertbw --image functions/resizer:latest --fprocess "convert -  
-colorspace Gray fd:1" --gateway http://127.0.0.1:9888
```

```

# processimages/handler.py

from minio import Minio
import requests
import json
import uuid
import os

def handle(st):
    req = json.loads(st)

    mc = Minio(os.environ['minio_hostname'],
               access_key=os.environ['minio_access_key'],
               secret_key=os.environ['minio_secret_key'],
               secure=False)

    names = []
    source_bucket = "incoming"
    dest_bucket = "processed"

    for file_name in req:
        names.append(convert_push(source_bucket, dest_bucket, file_name, mc))

    print(json.dumps(names))

def convert_push(source_bucket, dest_bucket, file_name, mc):
    mc.fget_object(source_bucket, file_name, "/tmp/" + file_name)

    f = open("/tmp/" + file_name, "rb")
    input_image = f.read()

    # download file
    r = requests.post("http://[Your IP]:9888/function/convertbw", input_image)

    # write to temporary file
    dest_file_name = get_temp_file()
    f = open("/tmp/" + dest_file_name, "wb")
    f.write(r.content)
    f.close()

    # sync to Minio
    mc.fput_object(dest_bucket, dest_file_name, "/tmp/"+dest_file_name)

    return dest_file_name

def get_temp_file():
    uuid_value = str(uuid.uuid4())
    return uuid_value

```

✓ YAML 수정하기

```
# processimages.yml

...
functions:
  loadimages:
    ...
    environment:
      minio_hostname: [Your IP]:9000
      minio_access_key: admin
      minio_secret_key: V4j7SPWAYQNAdmDyqSVGEbZSStX0osLf
      write_debug: true
```

✓ 빌드, 푸시, 배포

```
$ faas-cli up -f ./processimages.yml --gateway http://127.0.0.1:9888
$ kubectl get pods -n openfaas-fn
```

✓ 테스트

```
$ echo '["78a430c7-9295-495f-aa20-7221be5c6fdb", "bc68ca5a-a617-4f2e-9220-f7d9ea6e4740"]'
| faas invoke processimages --gateway http://127.0.0.1:9888
```

```
# ["962b1115-90f5-4759-91fb-8fd758be4b87", "33838655-2f68-4ec3-a802-
26747e0ecde2"]
```