

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  import sys
6  import json
7  import datetime
8
9  import boto3
10
11  DEBUG = False # True
12
13  # Storage(MinIO)にアクセスするライブラリ
14
15  class Storage:
16      # コンストラクタ
17      def __init__(self, url, key_id, access_key):
18          print("Constructor of Storage")
19          self.url = url
20          self.access_key_id = key_id
21          self.secret_access_key = access_key
22          # S3 への接続
23          self.c = boto3.resource(
24              service_name = 's3',
25              use_ssl=False,
26              endpoint_url=url,
27              aws_access_key_id=key_id,
28              aws_secret_access_key=access_key)
29
30      # デストラクタ
31      def __del__(self):
32          print("Destructor of Storage")
33
34
35      # バケット名補正
36      def bucket_name(self, bucket_name):
37          # 小文字で 3 文字以上
38          if len(bucket_name) < 3:
39              print("at least 3 charactors.")
40              return None
```

```
41
42     return bucket_name.lower()
43
44
45 # バケット作成
46 def create_bucket(self, bucket_name):
47     print("create_bucket: " + bucket_name)
48     name = self.bucket_name(bucket_name)
49     if name == None:
50         print("Invalid name")
51         return None
52     bkt = self.c.Bucket(name)
53     res = bkt.create()
54     return res
55
56
57 # バケット削除
58 def delete_bucket(self, bucket_name):
59     print("delete_bucket")
60     name = self.bucket_name(bucket_name)
61     if name == None:
62         print("Invalid name")
63         return None
64     bkt = self.c.Bucket(name)
65     res = bkt.delete()
66     return res
67
68
69 # ファイル保存
70 def put_file(self, bucket_name, file_name, file):
71     print("put_file", bucket_name, file_name)
72
73     obj = self.c.Object(bucket_name, file_name)
74     res = obj.put(Body=file)
75     return res
76
77
78 # ファイル取り出し
79 def get_file(self, bucket_name, file_name, path=None):
80     print("get_file", bucket_name, file_name)
```

```
81     try:
82         obj = self.c.Object(bucket_name, file_name)
83         if path:
84             res = obj.download_file(path)
85             return res
86         else:
87             res = obj.get()
88             f = res['Body'].read()
89             return f
90     except:
91         return None
92
93     # ファイル削除
94     def delete_file(self, bucket_name, file_name):
95         print("delete_file", bucket_name, file_name)
96         obj = self.c.Object(bucket_name, file_name)
97         res = obj.delete()
98         print(res)
99         return res
100
101
102     # ファイル一覧
103     def list_file(self, bucket_name):
104         print("list_file")
105         bkt = self.c.Bucket(bucket_name)
106         res = []
107         for obj in bkt.objects.all():
108             res.append(obj.key)
109
110         return res
```