hostess: Lightweight Distributed Resource Management and Data Processing in Python

Michael St. Clair, Chase Million, Sierra V. Brown

Million Concepts LLC — (mstclair@millionconcepts.com)

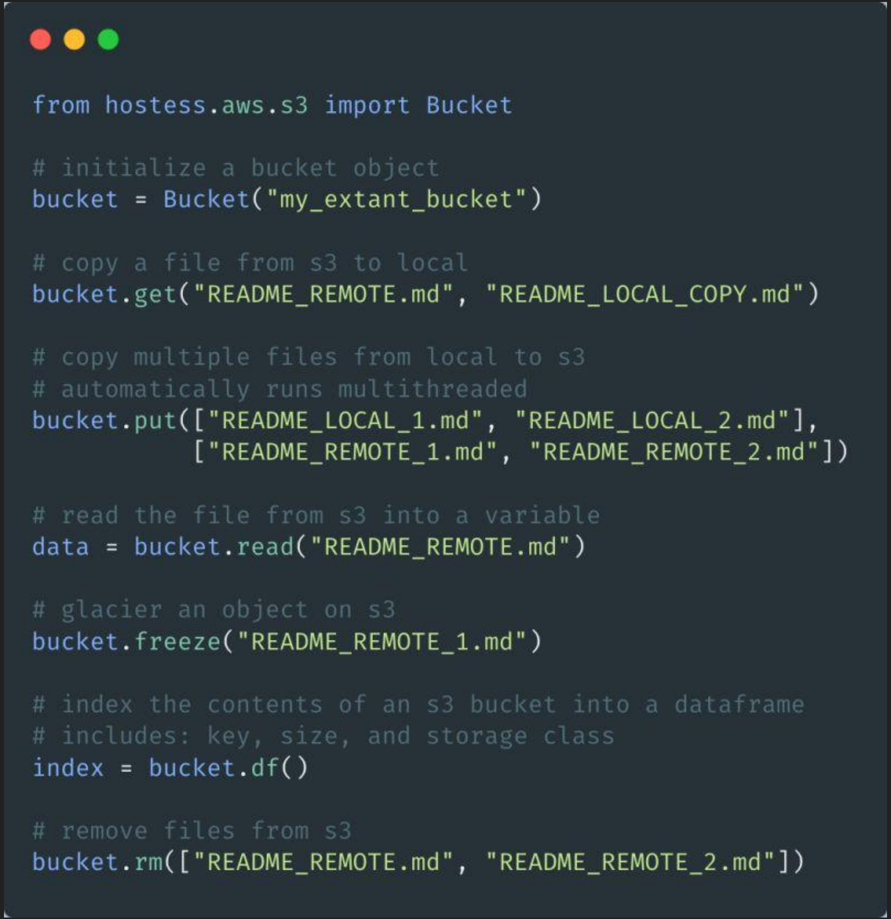
```
Placeholder
for QR code
to Zenodo
repository
```

```
https://github.com/MillionConcepts/hostess/
```

10^6C Million Concepts

hostess is a general-purpose utility and administration library that provides lightweight, Pythonic interfaces to distributed resources, system processes, and Python internals. It includes high-level modules with special functionality for working with AWS EC2 instances and S3 buckets, and a framework (called station) for workflow coordination / orchestration.

hostess reduces syntactic headaches and boilerplate while still allowing low-level manipulation, with an emphasis on fitting intuitively and idiomatically into common data science and scientific analysis workflows.



```
from hostess.aws.s3 import Bucket

# initialize a bucket object
bucket = Bucket("my_extant_bucket")

# copy a file from s3 to local
bucket.get("README_REMOTE.md", "README_LOCAL_COPY.md")

# copy multiple files from local to s3
# automatically runs multithreaded
bucket.put(["README_LOCAL_1.md", "README_LOCAL_2.md"],
           ["README_REMOTE_1.md", "README_REMOTE_2.md"])

# read the file from s3 into a variable
data = bucket.read("README_REMOTE.md")

# glacier an object on s3
bucket.freeze("README_REMOTE_1.md")

# index the contents of an s3 bucket into a dataframe
# includes: key, size, and storage class
index = bucket.df()

# remove files from s3
bucket.rm(["README_REMOTE.md", "README_REMOTE_2.md"])
```