

WEEK 3 ASSIGNMENT

Large-Scale Data Storage Systems – DATA-5400 | Spring 2020

Christina Morgenstern

In this week's assignment, I am using the AWS Web and CLI (Command Line Interface) interfaces to view and modify the contents of the cloud storage.

The first step involves the creation of the Secret and Access keys in my AWS account. Access keys are used to make programmatic calls to AWS from the AWS CLI. The keys are created in the Identity and Access Management (IAM) section through pressing the "Create New Access Key" button. The status of the key must be enabled active when using and disabled after that to prevent unauthorized access.

Next, I am starting the CentOS Virtual Machine through opening Virtual Box, selecting the CentOS instance and logging into the VM using my credentials. I am also retrieving the IP address using the `ip a` command.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1062.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Wed Feb  5 20:52:16 on tty1
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:c6:e2:02 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.34/24 brd 10.0.0.255 scope global noprefixroute dynamic enp0s3
        valid_lft 86377sec preferred_lft 86377sec
    inet6 fe80::14c2:ada8:f9a2:40e3/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost ~]#
```

Connecting to the VM was done using the Terminal on my Mac and the following command:

```
ssh root@10.0.0.34
```

```

Last login: Wed Feb  5 20:51:02 on ttys000
(base) Christinas-MacBook-Pro:~ Christina$ ssh root@10.0.0.34
The authenticity of host '10.0.0.34 (10.0.0.34)' can't be established.
ECDSA key fingerprint is SHA256:CCXATBnWDb5kGYueybnWaC3b0ycaflTGUE6ZdDYtMFE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.34' (ECDSA) to the list of known hosts.
root@10.0.0.34's password:
Last login: Wed Feb  5 20:58:01 2020

```

Checking for the version of Python revealed Python 2.7.5, which comes standard with CentOS. Also, there is no pip preinstalled on CentOS.

```

[root@localhost ~]# python --version
Python 2.7.5
[root@localhost ~]# pip --version
-bash: pip: Kommando nicht gefunden.
[root@localhost ~]#

```

I followed this tutorial (<https://linuxize.com/post/how-to-install-python-3-on-centos-7/>) in order to install Python 3 using the following commands:

```

sudo yum install centos-release-scl
sudo yum install rh-python36

```

When now checking for the Python version, still Python 2.7 is the default.

```
python --version
```

After enabling Python 3.6 version, Python 3.6.9 is now active. However, this will only last for this session.

```
scl enable rh-python36 bash
```

```

[root@localhost ~]# python --version
Python 2.7.5
[root@localhost ~]# scl enable rh-python36 bash
[root@localhost ~]# python --version
Python 3.6.9

```

I also installed pip on CentOS using the tutorial (<https://linuxize.com/post/how-to-install-pip-on-centos-7/>)

```

sudo yum install epel-release
sudo yum install python-pip

```

```

[root@localhost ~]# pip --version
pip 9.0.1 from /opt/rh/rh-python36/root/usr/lib/python3.6/site-packages (python 3.6)

```

Installing the AWS CLI (<https://aws.amazon.com/cli/>)

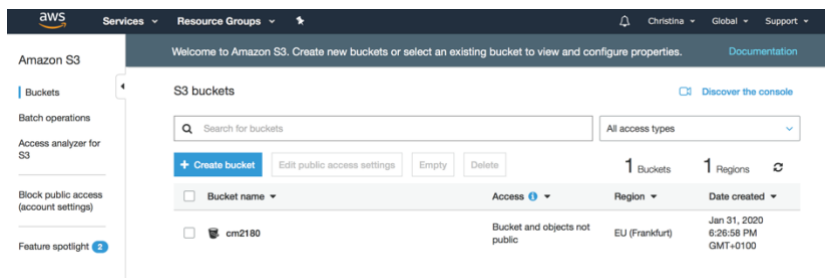
```
pip install awscli
```

```
Successfully installed PyYAML-5.2 awscli-1.17.11 botocore-1.14.11 colorama-0.4.1
docutils-0.15.2 jmespath-0.9.4 pyasn1-0.4.8 python-dateutil-2.8.1 rsa-3.4.2 s3t
ransfer-0.3.2 six-1.14.0 urllib3-1.25.8
You are using pip version 9.0.1, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

And upgrading pip to version 20.0.2

```
[root@localhost ~]# pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/54/0c/d01aa759fdc501a58f43
1eb594a17495f15b88da142ce14b5845662c13f3/pip-20.0.2-py2.py3-none-any.whl (1.4MB)
    100% |#####| 1.4MB 801kB/s
Installing collected packages: pip
  Found existing installation: pip 9.0.1
  Uninstalling pip-9.0.1:
    Successfully uninstalled pip-9.0.1
  Successfully installed pip-20.0.2
```

Creating a S3 bucket using the browser by just clicking on the +Create bucket button and specifying a unique name



For configuring the AWS CLI, I used the `aws configure` command and entered the previously generated AWS Access Key ID, the AWS Secret Access Key, eu-central-1 for the Default region name and json for the default output format (<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html>).

Creating a bucket via the CLI wasn't that straight-forward. First, it was not possible to create a bucket in my chosen region eu-central-1. The following note was given on the AWS CLI Command Reference tutorial website (<https://docs.aws.amazon.com/cli/latest/reference/s3api/create-bucket.html>).

Note

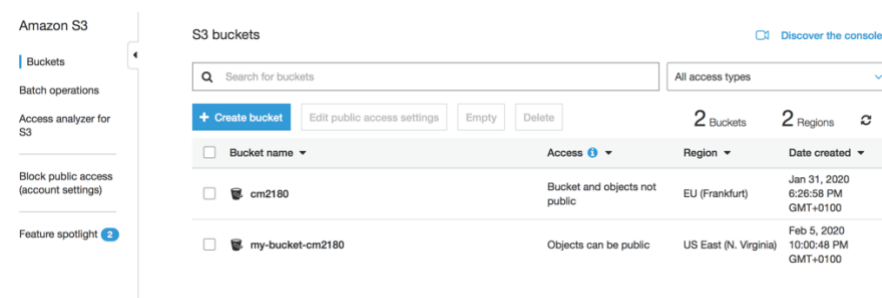
If you send your create bucket request to the `s3.amazonaws.com` endpoint, the request goes to the us-east-1 Region. Accordingly, the signature calculations in Signature Version 4 must use us-east-1 as the Region, even if the location constraint in the request specifies another Region where the bucket is to be created. If you create a bucket in a Region other than US East (N. Virginia), your application must be able to handle 307 redirect. For more information, see [Virtual Hosting of Buckets](#).

So, I created a bucket in the us-east-1 region using the following command
`aws s3api create-bucket --bucket my-bucket-cm2180 --region us-east-1`

It is important to use a bucket name that is unique otherwise you'll get an error message, as I did before.

```
[root@localhost ~]# aws s3api create-bucket --bucket my-bucket_cm2180
An error occurred (IllegalLocationConstraintException) when calling the CreateBucket operation: The unspecified location constraint is incompatible for the region on specific endpoint this request was sent to.
[root@localhost ~]# aws s3api create-bucket --bucket my-bucket --region us-east-1
An error occurred (BucketAlreadyExists) when calling the CreateBucket operation: The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.
[root@localhost ~]# aws s3api create-bucket --bucket my-bucket_cm2180 --region us-east-1
An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.
[root@localhost ~]# aws s3api create-bucket --bucket my-bucket-cm2180 --region us-east-1
{
  "Location": "/my-bucket-cm2180"
}
[root@localhost ~]#
```

Now both buckets show up in the Console window.



The screenshot shows the Amazon S3 console interface. On the left is a navigation menu with options like Buckets, Batch operations, and Access analyzer for S3. The main area displays a table of S3 buckets. There are two buckets listed: 'cm2180' and 'my-bucket-cm2180'. The 'cm2180' bucket is in the EU (Frankfurt) region and has public access blocked. The 'my-bucket-cm2180' bucket is in the US East (N. Virginia) region and has public access allowed for objects.

Bucket name	Access	Region	Date created
cm2180	Bucket and objects not public	EU (Frankfurt)	Jan 31, 2020 6:26:58 PM GMT+0100
my-bucket-cm2180	Objects can be public	US East (N. Virginia)	Feb 5, 2020 10:00:48 PM GMT+0100

Creating two folders:

```
[root@localhost ~]# mkdir folder1
[root@localhost ~]# mkdir folder2
```

Navigating to folder1 using the `cd folder1` command and adding three files to folder1:

```
[root@localhost folder1]# echo "This is a test file 1" > testfile1
[root@localhost folder1]# echo "This is a test file 2" > testfile2
[root@localhost folder1]# echo "This is a test file 3" > testfile3
[root@localhost folder1]# ls
testfile1 testfile2 testfile3
```

Navigating to folder2 using the `cd ..` and `cd folder2` commands and adding five files to folder2:

```
[root@localhost folder1]# cd ..
[root@localhost ~]# cd folder2
[root@localhost folder2]# echo "This is a test file a" > testfileA
[root@localhost folder2]# echo "This is a test file b" > testfileB
[root@localhost folder2]# echo "This is a test file c" > testfileC
[root@localhost folder2]# echo "This is a test file d" > testfileD
[root@localhost folder2]# echo "This is a test file e" > testfileE
[root@localhost folder2]# ls
testfileA testfileB testfileC testfileD testfileE
```

Uploading the files from the "folder1" folder into the S3 bucket and listing of bucket content:

```
[root@localhost folder1]# cd ..
[root@localhost ~]# cd folder2
[root@localhost folder2]# echo "This is a test file a" > testfileA
[root@localhost folder2]# echo "This is a test file b" > testfileB
[root@localhost folder2]# echo "This is a test file c" > testfileC
[root@localhost folder2]# echo "This is a test file d" > testfileD
[root@localhost folder2]# echo "This is a test file e" > testfileE
[root@localhost folder2]# ls
testfileA testfileB testfileC testfileD testfileE
[root@localhost folder2]# cd ..
[root@localhost ~]# aws s3 ls
2020-01-31 18:26:58 cm2180
2020-02-05 22:00:48 my-bucket-cm2180
[root@localhost ~]# aws s3 cp folder1/testfile1 s3://cm2180
upload: folder1/testfile1 to s3://cm2180/testfile1
[root@localhost ~]# aws s3 cp folder1/testfile2 s3://cm2180
upload: folder1/testfile2 to s3://cm2180/testfile2
[root@localhost ~]# aws s3 cp folder1/testfile3 s3://cm2180
upload: folder1/testfile3 to s3://cm2180/testfile3
[root@localhost ~]# aws s3 ls
2020-01-31 18:26:58 cm2180
2020-02-05 22:00:48 my-bucket-cm2180
[root@localhost ~]# aws s3 ls s3://cm2180
2020-02-05 22:21:31      22 testfile1
2020-02-05 22:21:52      22 testfile2
2020-02-05 22:22:04      22 testfile3
```

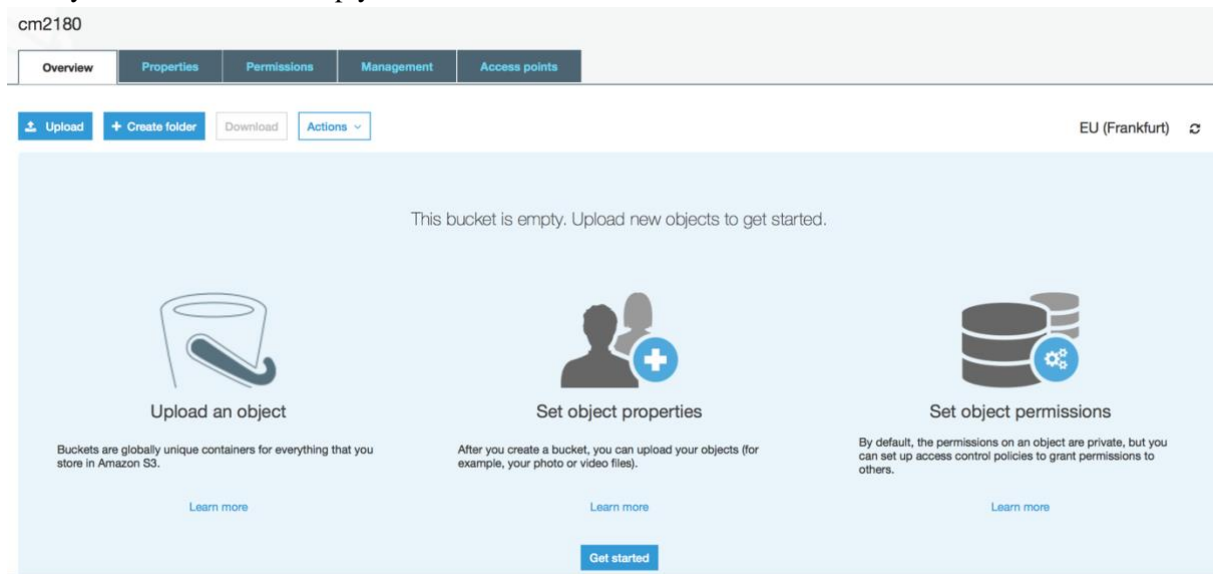
I successfully transferred the three testfiles from folder1 into the S3 bucket cm2180 on S3. This is verified in the browser:

cm2180			
Overview	Properties	Permissions	Management
Access points			
Q Type a prefix and press Enter to search. Press ESC to clear.			
Upload	Create folder	Download	Actions
EU (Frankfurt)			
Viewing 1 to 3			
<input type="checkbox"/> Name	Last modified	Size	Storage class
<input type="checkbox"/> testfile1	Feb 5, 2020 10:21:31 PM GMT+0100	22.0 B	Standard
<input type="checkbox"/> testfile2	Feb 5, 2020 10:21:52 PM GMT+0100	22.0 B	Standard
<input type="checkbox"/> testfile3	Feb 5, 2020 10:22:04 PM GMT+0100	22.0 B	Standard

Removing the files from the S3 bucket and listing the contents:

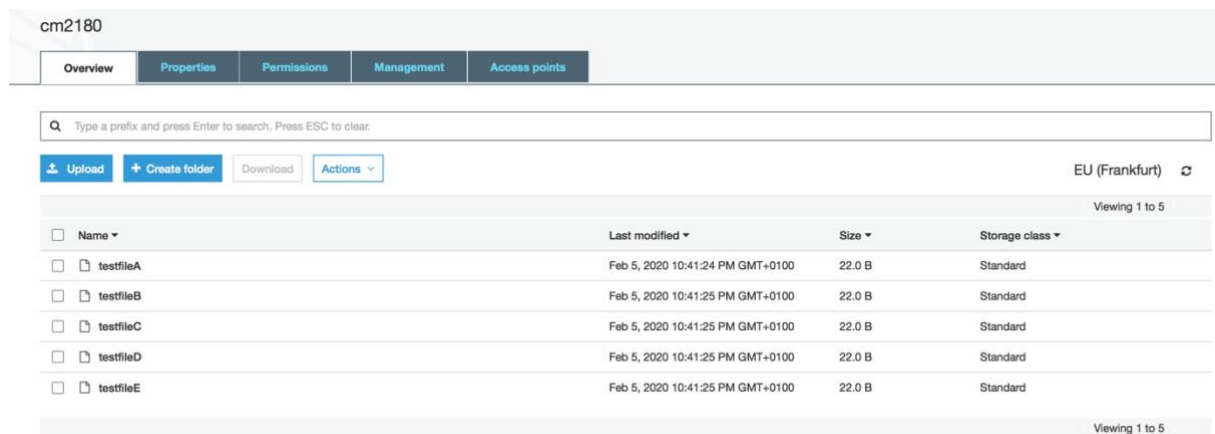

```
[root@localhost ~]# aws s3 rm s3://cm2180/testfile1
delete: s3://cm2180/testfile1
[root@localhost ~]# aws s3 rm s3://cm2180/testfile2
delete: s3://cm2180/testfile2
[root@localhost ~]# aws s3 rm s3://cm2180/testfile3
delete: s3://cm2180/testfile3
[root@localhost ~]# aws s3 ls s3://cm2180
[root@localhost ~]#
```

Verify that the bucket is empty via the console:



Synchronizing the contents of folder2 with my bucket and verification of the contents:

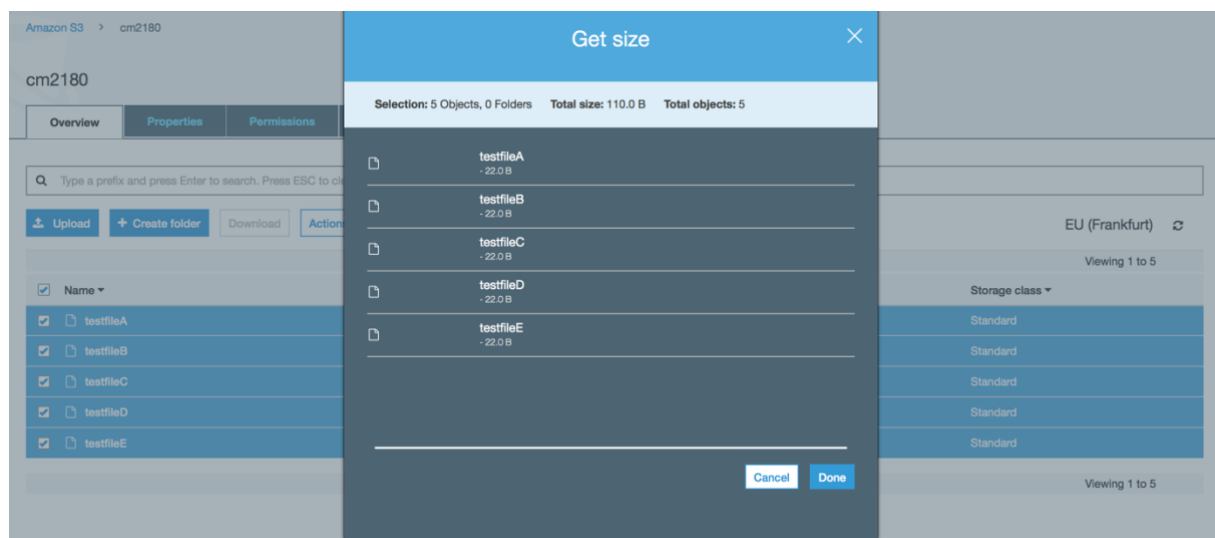
```
[root@localhost folder2]# aws s3 sync . s3://cm2180
upload: ./testfileA to s3://cm2180/testfileA
upload: ./testfileB to s3://cm2180/testfileB
upload: ./testfileC to s3://cm2180/testfileC
upload: ./testfileD to s3://cm2180/testfileD
upload: ./testfileE to s3://cm2180/testfileE
[root@localhost folder2]# aws s3 ls s3://cm2180
2020-02-05 22:41:24      22 testfileA
2020-02-05 22:41:25      22 testfileB
2020-02-05 22:41:25      22 testfileC
2020-02-05 22:41:25      22 testfileD
2020-02-05 22:41:25      22 testfileE
[root@localhost folder2]#
```



Downloading all five files from the bucket cm2180 to folder2 using the recursive option, and excluding .jpg formats.

```
[root@localhost folder2]# cd ..
[root@localhost ~]# aws s3 cp folder2 s3://cm2180/ --recursive --exclude "*.jpg"
upload: folder2/testfileA to s3://cm2180/testfileA
upload: folder2/testfileB to s3://cm2180/testfileB
upload: folder2/testfileC to s3://cm2180/testfileC
upload: folder2/testfileD to s3://cm2180/testfileD
upload: folder2/testfileE to s3://cm2180/testfileE
[root@localhost ~]#
```

The total size of the bucket was calculated using the console window and is 110 bytes.



Checking the total size of the folder yielded 20K but there seem to be some “hidden” files.

```
[root@localhost folder2]# ls -lah
insgesamt 20K
drwxr-xr-x. 2 root root 91  5. Feb 22:16 .
dr-xr-x---. 7 root root 203 5. Feb 22:08 ..
-rw-r--r--. 1 root root 22  5. Feb 22:16 testfileA
-rw-r--r--. 1 root root 22  5. Feb 22:16 testfileB
-rw-r--r--. 1 root root 22  5. Feb 22:16 testfileC
-rw-r--r--. 1 root root 22  5. Feb 22:16 testfileD
-rw-r--r--. 1 root root 22  5. Feb 22:16 testfileE
```

The buckets were deleted and the access keys disabled.