



ÅBO AKADEMI UNIVERSITY

CLOUD COMPUTING

Assignment 5



LUIS ARAÚJO(2004624)

MAY 21, 2021

Contents

1	Introduction	3
2	Problem 1: Word counting	4

Chapter 1

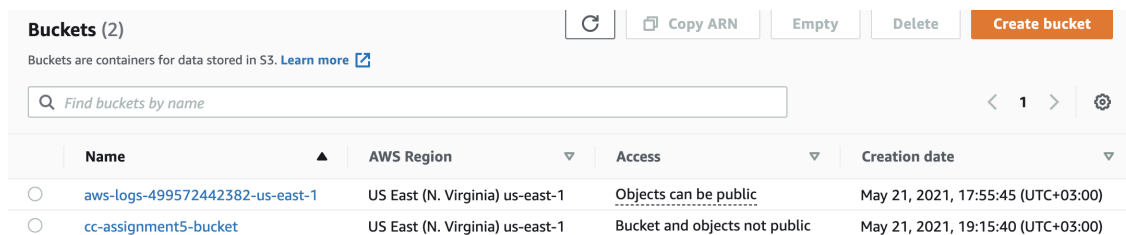
Introduction

Github repository: <https://github.com/it-teaching-abo-akademi/assignment-5-LAraujo7>

Chapter 2

Problem 1: Word counting

First, I started by creating a bucket named "cc-assignment5-bucket".

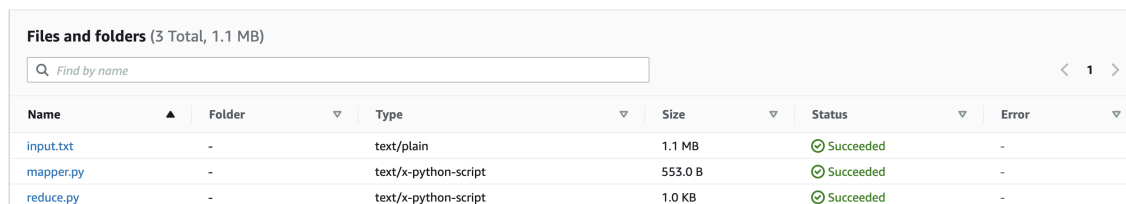


The screenshot shows the AWS S3 Buckets console. At the top, there are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. Below these is a search bar labeled 'Find buckets by name'. The main table lists two buckets:

	Name	AWS Region	Access	Creation date
<input type="radio"/>	aws-logs-499572442382-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	May 21, 2021, 17:55:45 (UTC+03:00)
<input type="radio"/>	cc-assignment5-bucket	US East (N. Virginia) us-east-1	Bucket and objects not public	May 21, 2021, 19:15:40 (UTC+03:00)

Figure 2.1: Creating a bucket

Then, the reduce, the map and the input files were uploaded to the bucket



The screenshot shows the AWS S3 Files and folders console. At the top, there is a search bar labeled 'Find by name'. The main table lists three files:

Name	Folder	Type	Size	Status	Error
input.txt	-	text/plain	1.1 MB	✔ Succeeded	-
mapper.py	-	text/x-python-script	553.0 B	✔ Succeeded	-
reduce.py	-	text/x-python-script	1.0 KB	✔ Succeeded	-

Figure 2.2: Files uploaded

The Mapper file:

```
#!/usr/bin/env python
"""mapper.py"""

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        print '%s\t%s' % (word, 1)
```

Figure 2.3: Mapper file

Reduce file of the 100 most frequent words:

```
#!/usr/bin/python

import sys
import collections

counter = collections.Counter()

for line in sys.stdin:
    k, v = line.strip().split("\t", 2)

    counter[k] += int(v)

print counter.most_common(100)
```

Figure 2.4: 100 Most Frequent Words

Reduce file for the words of length to 3 and 5:

```
#!/usr/bin/env python
"""reducer_length.py"""

import sys
import collections

counter = collections.Counter()

for line in sys.stdin:
    k, v = line.strip().split("\t", 2)

    if len(k) == 3 or len(k) == 5 :
        counter[k] += int(v)

print counter.most_common(100)
```

Figure 2.5: Words of length to 3 and 5

Once uploaded the files, it was the time to create the cluster

Network and hardware

Availability zone: us-east-1f

Subnet ID: [subnet-37517339](#) 

Master: Running 1 m5.xlarge

Core: Running 4 m5.xlarge

Task: --

Cluster scaling: Not enabled

Figure 2.6: Cluster

Finally, I added the step.

	s-16BQ3E2LV4W42	Streaming program	Completed	2021-05-21 20:03 (UTC+3)	38 seconds	View logs
---	-----------------	-------------------	-----------	--------------------------	------------	---------------------------

Figure 2.7: Step completed