Department of Computer Science, Electrical and Space Engineering Luleå University of Technology

Cloud Services

Lab 2 Report: Programming Cloud Services - Storage Services

Author:
Hanna Ogbazghi
Elyas Khorasani
Yacine Rabehi

Supervisors:
Karan Mitra

8 November 2021

Contents

1 Exercise A

1.1 Question 1: setup the programming environment to create Cloud services using the APIs provided by AWS

For this lab, we decided to employ python3 to achieve the required tasks. For the setup, we followed the instructions provided in the following link:
`https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html`
The summary of the followed steps is:

1. Installing boto3 using the command: pip install boto3.
2. Creating the AWS folder containing two files, one for the credentials and the other for the default region. This is done using the AWS command line interface.

2 Exercise B

2.1 Question 1:Explain in detail how the Amazon S3 service clients are created by providing details of the packages and classes involved. Create a diagram of the dependencies involved

.

The diagram shown in figure 1 illustrates the dependencies for the employed functions (methods). We note that the access to the dependencies is limited for boto3 (we cannot see the different calls made in other to execute the methods). There might be more access to the details when employing Java instead.

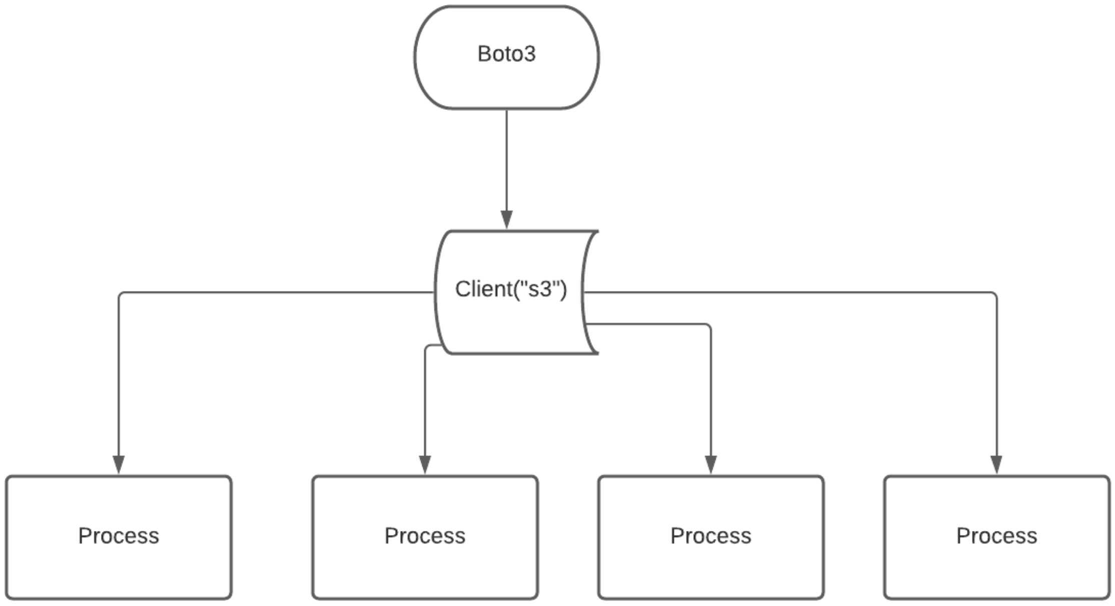
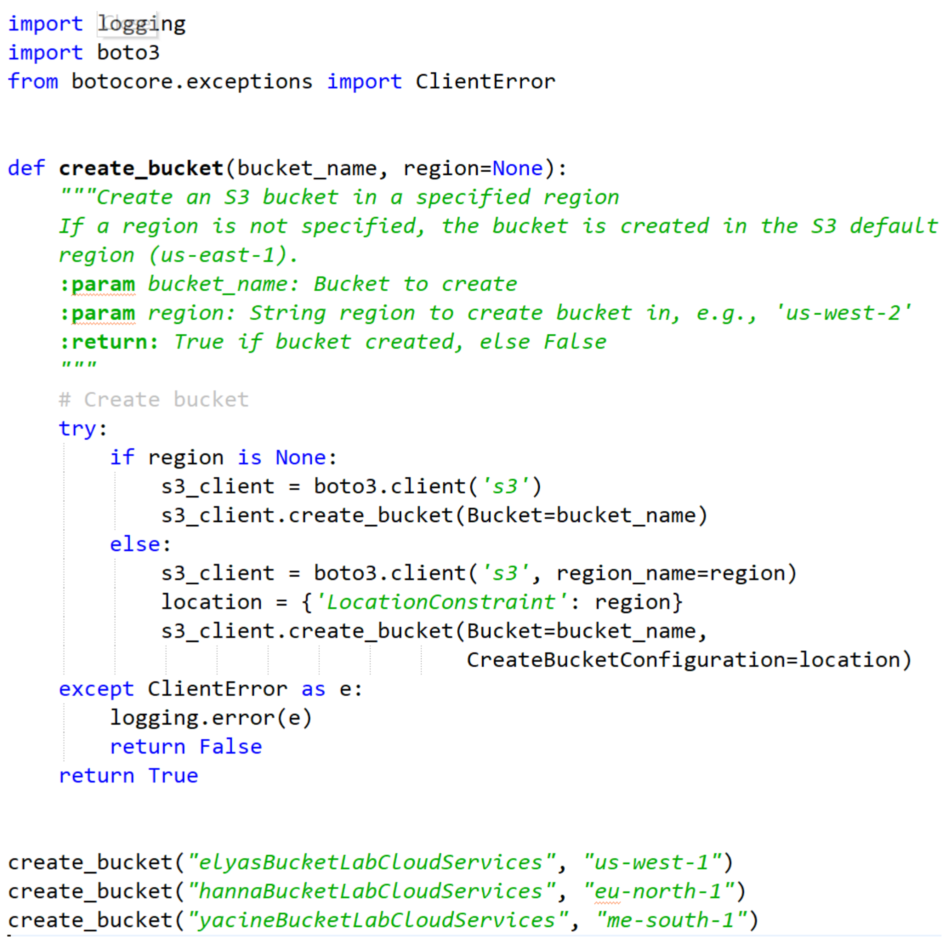


Figure 1: Dependency Diagram for the methods employed

2.2 Question 2: Python program to create a bucket in three regions

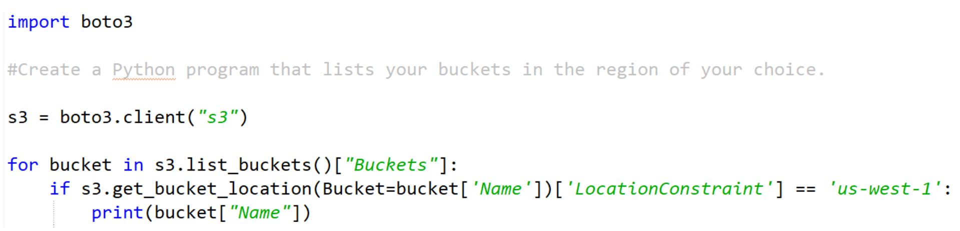
Three different buckets are created in elyasbucketbabcloudservices is created in North California (us-west-1)
hannabucketlabcloudservices is created in Stockholm (eu-north-1)
yacinebucketlabcloudservices is created in Tokyo (ap-northeast-1)



```python
import logging
import boto3
from botocore.exceptions import ClientError


def create_bucket(bucket_name, region=None):
    """Create an S3 bucket in a specified region
    If a region is not specified, the bucket is created in the S3 default
    region (us-east-1).
    :param bucket_name: Bucket to create
    :param region: String region to create bucket in, e.g., 'us-west-2'
    :return: True if bucket created, else False
    """
    # Create bucket
    try:
        if region is None:
            s3_client = boto3.client('s3')
            s3_client.create_bucket(Bucket=bucket_name)
        else:
            s3_client = boto3.client('s3', region_name=region)
            location = {'LocationConstraint': region}
            s3_client.create_bucket(Bucket=bucket_name,
                                    CreateBucketConfiguration=location)
    except ClientError as e:
        logging.error(e)
        return False
    return True


create_bucket("elyasBucketLabCloudServices", "us-west-1")
create_bucket("hannaBucketLabCloudServices", "eu-north-1")
create_bucket("yacineBucketLabCloudServices", "me-south-1")
```

Figure 2: Create a bucket in three regions

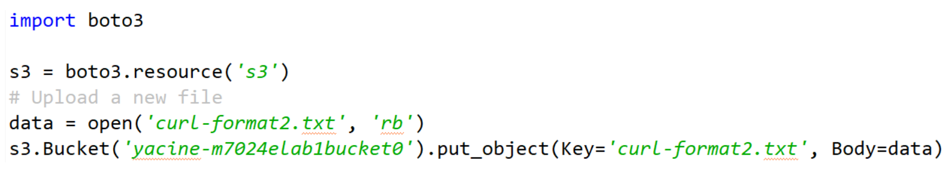
2.3 Question 3: Python program that lists your buckets in the region



```python
import boto3

#Create a Python program that lists your buckets in the region of your choice.

s3 = boto3.client("s3")

for bucket in s3.list_buckets()["Buckets"]:
    if s3.get_bucket_location(Bucket=bucket['Name'])['LocationConstraint'] == 'us-west-1':
        print(bucket["Name"])
```

Figure 3: Listing the existing buckets in the region

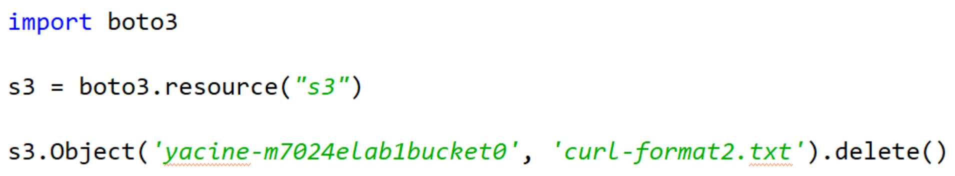
2.4 Question 4: Python program to upload objects to a bucket



```python
import boto3

s3 = boto3.resource('s3')
# Upload a new file
data = open('curl-format2.txt', 'rb')
s3.Bucket('yacine-m7024elab1bucket0').put_object(Key='curl-format2.txt', Body=data)
```

Figure 4: Upload objects into a bucket

2.5 Question 5: Python program to delete a particular object from a bucket



```python
import boto3

s3 = boto3.resource("s3")

s3.Object('yacine-m7024elab1bucket0', 'curl-format2.txt').delete()
```

Figure 5: Delete an object from a bucket

3 Exercise C

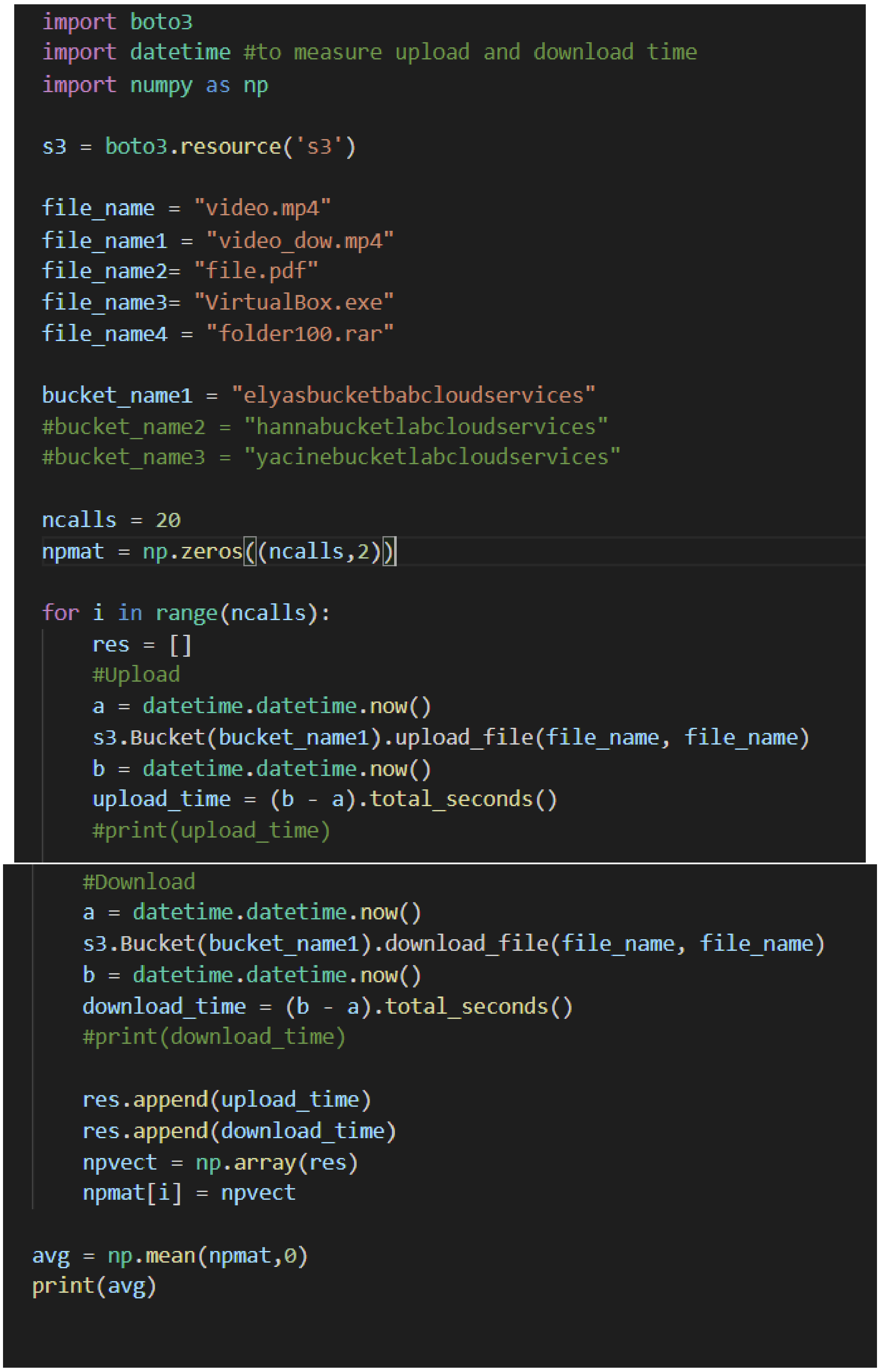
3.1 Question: Measure upload and download latency

Create a Java program to upload and download objects (sizes 1MB, 10 MB, 100 MB and 500MB) from the three regions used in the above exercise. Measure the object upload and download latency from these regions?

In order to have a meaningful analysis, we measured the upload and download latency a number of 20 times and reported the mean of the values. The tests were conducted in three different regions, namely, us-west-1, eu-north-1 and ap-northeast-1. The results can be observed in table 1.

The python code used to execute the task is shown in figure 6. The variant characteristics of the WiFi employed are illustrated in figure 7.

The four different sizes files used are as follows:
1.file.pdf (1.3 Mb)
2.video.mp4 (13.5Mb)
3.VirtualBox.exe (103.2Mb)
4.folder100.rar (500.4Mb)



```python
import boto3
import datetime #to measure upload and download time
import numpy as np

s3 = boto3.resource('s3')

file_name = "video.mp4"
file_name1 = "video_dow.mp4"
file_name2= "file.pdf"
file_name3= "VirtualBox.exe"
file_name4 = "folder100.rar"

bucket_name1 = "elyasbucketbabcloudservices"
#bucket_name2 = "hannabucketlabcloudservices"
#bucket_name3 = "yacinebucketlabcloudservices"

ncalls = 20
npmat = np.zeros((ncalls,2))

for i in range(ncalls):
    res = []
    #Upload
    a = datetime.datetime.now()
    s3.Bucket(bucket_name1).upload_file(file_name, file_name)
    b = datetime.datetime.now()
    upload_time = (b - a).total_seconds()
    #print(upload_time)
    #Download
    a = datetime.datetime.now()
    s3.Bucket(bucket_name1).download_file(file_name, file_name)
    b = datetime.datetime.now()
    download_time = (b - a).total_seconds()
    #print(download_time)

    res.append(upload_time)
    res.append(download_time)
    npvect = np.array(res)
    npmat[i] = npvect

avg = np.mean(npmat,0)
print(avg)
```

Figure 6: Measure upload and download latency

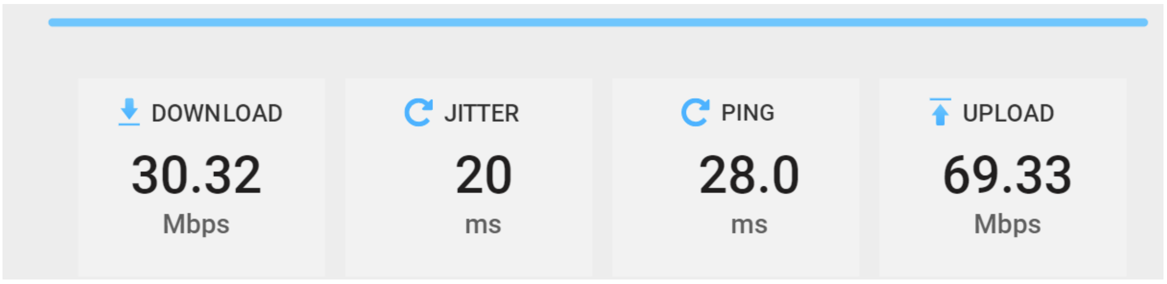

DOWNLOAD 30.32 Mbps
JITTER 20 ms
PING 28.0 ms
UPLOAD 69.33 Mbps

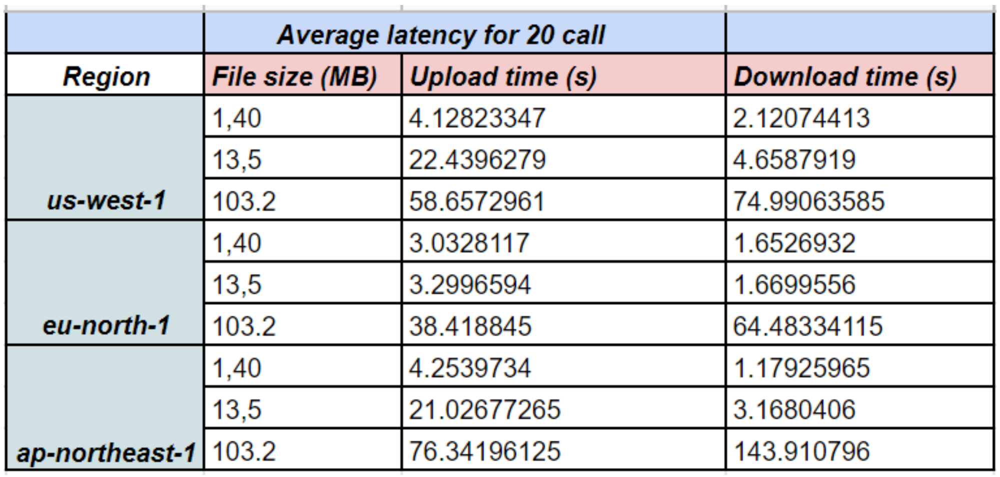
Figure 7: Skeria WIFI Speed

3.2 Analysis

In order to analyse our results, we plot several graphs as shown in figures 8, 9, 10, 11 and 12. Figure 8 shows the comparison of the obtained upload latency between the three different regions and for the specified file sizes. Figure 9 similarly compares the download latency between the regions and for the different file sizes.
In figure 8, we observe that the upload latency is the best in eu-north-1, with us-west-1 coming in the second place for two file sizes (1.4 MB and 13.5MB). However, the upload latency for the file size of 103.2 MB was better for us-west-1 than ap-northeast-1.

In figure 9, we notice that the download latency is relatively similar for the three regions for the two file sizes of 1.4 MB and 13.5MB. For the file size of 103.2 MB, the download latency is better for eu-north-1 than us-west-1.

From the previous comparisons, we can say that the differences between the upload/download latency between the different regions start to show up the larger the files get. Therefore, for small files, no apparent difference would be noticed, while for large files, the differences start to appear clearly.



	Average latency for 20 call		
Region	***File size (MB)***	***Upload time (s)***	***Download time (s)***
	1,40	4.12823347	2.12074413
	13,5	22.4396279	4.6587919
us-west-1	103.2	58.6572961	74.99063585
	1,40	3.0328117	1.6526932
	13,5	3.2996594	1.6699556
eu-north-1	103.2	38.418845	64.48334115
	1,40	4.2539734	1.17925965
	13,5	21.02677265	3.1680406
ap-northeast-1	103.2	76.34196125	143.910796

Table 1: Upload and Download Latency

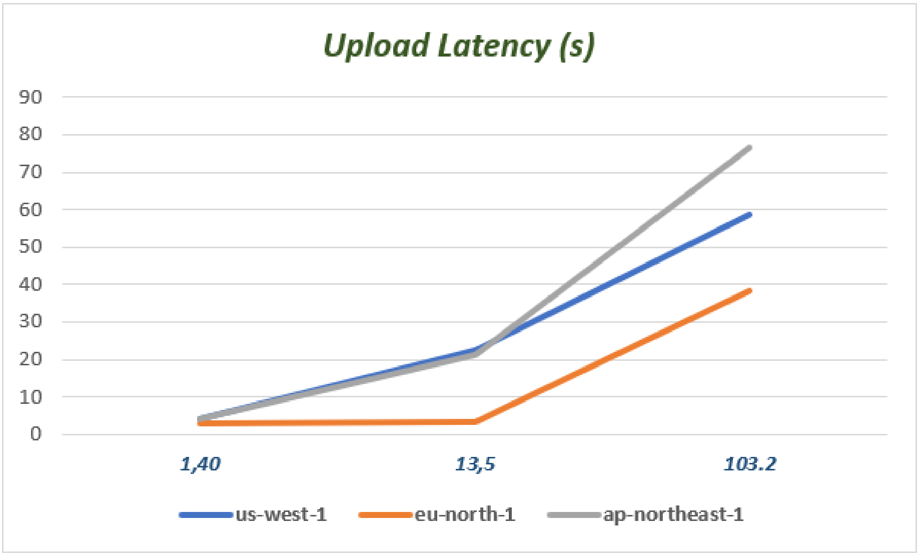


Figure 8: Upload Latency

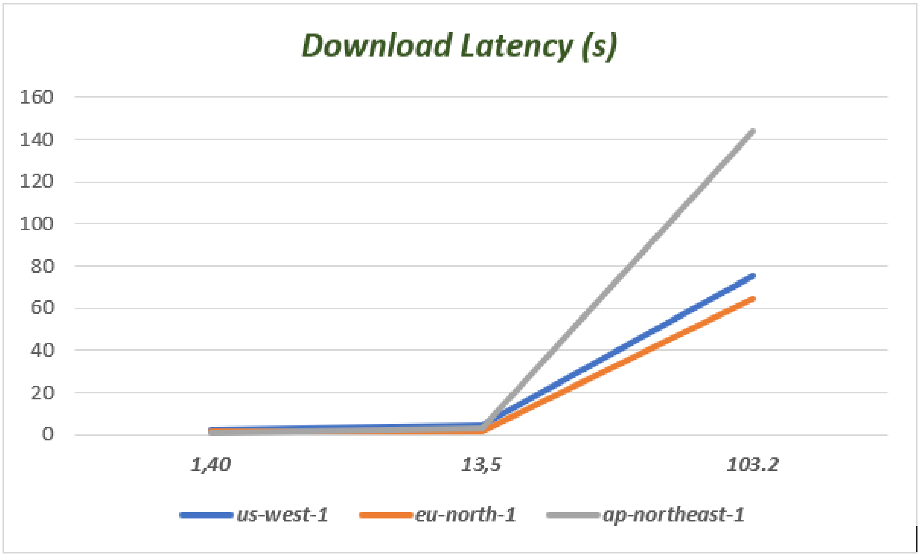


Figure 9: Download Latency

In figures 10, 11 and 12, we compare the download and upload latency. We can observe that for the small file sizes of 1.4 MB and 13.5 MB, the upload latency is considerably lower than the download latency. On the other hand, for the larger file of size 103.4 MB, the download latency is considerably lower than the upload latency. Again, we note that the differences show up when switching from small size files to larger ones.

We note that some inconsistencies with the data were observed due to the unstable "Skeria" (our student dorm) connection.

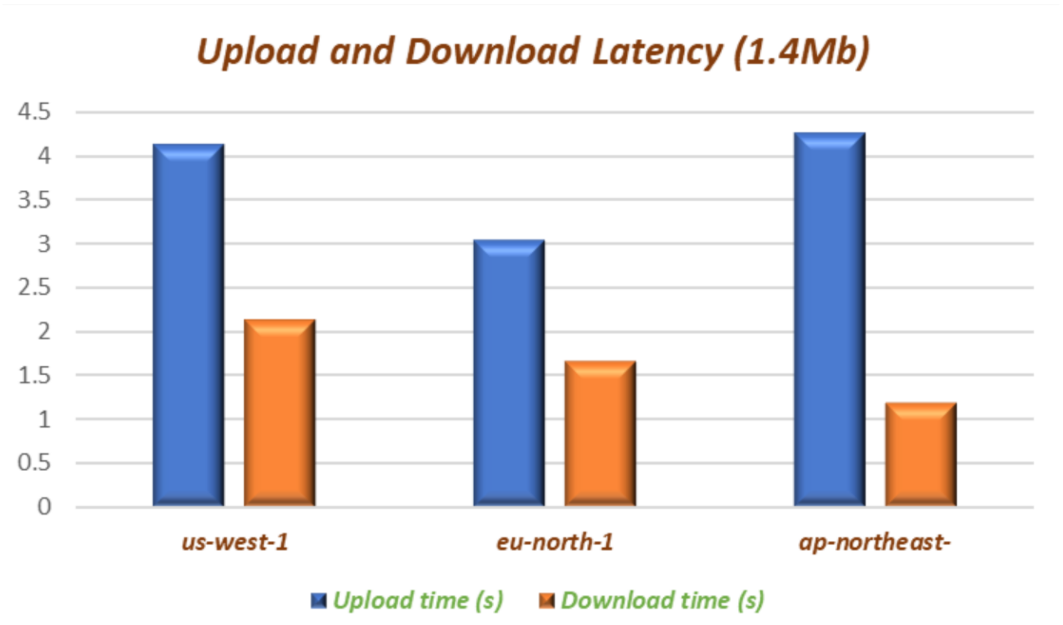


Figure 10: Upload vs Download Latency

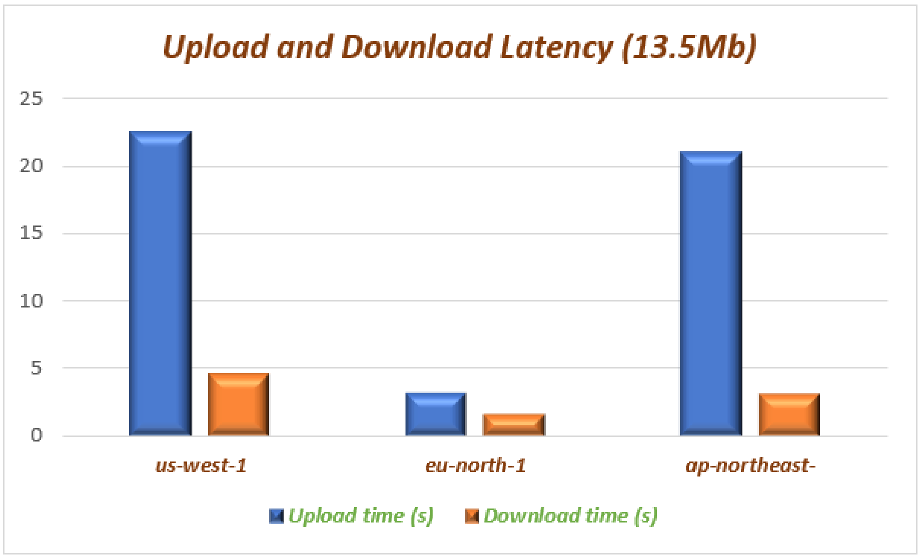


Figure 11: Upload vs Download Latency

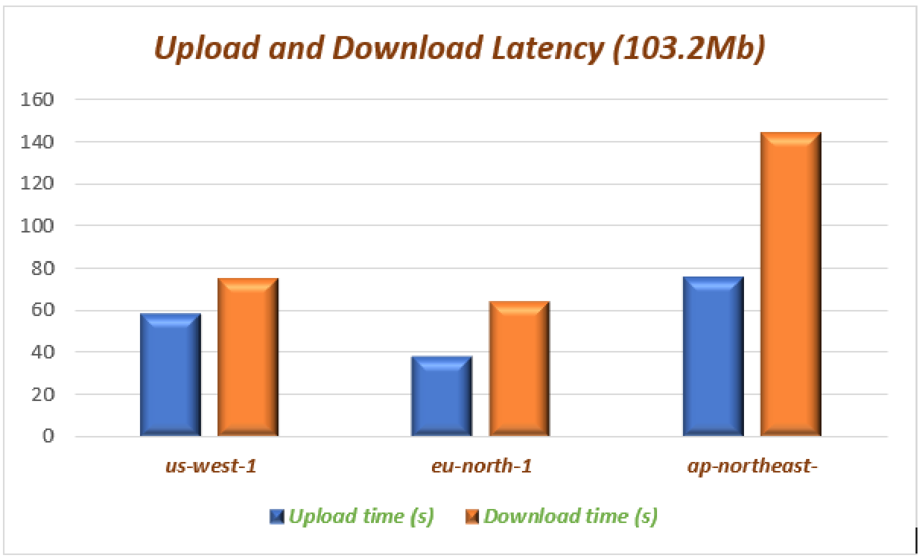


Figure 12: Upload vs Download Latency