Homework 1

Amazon S3 Buckets

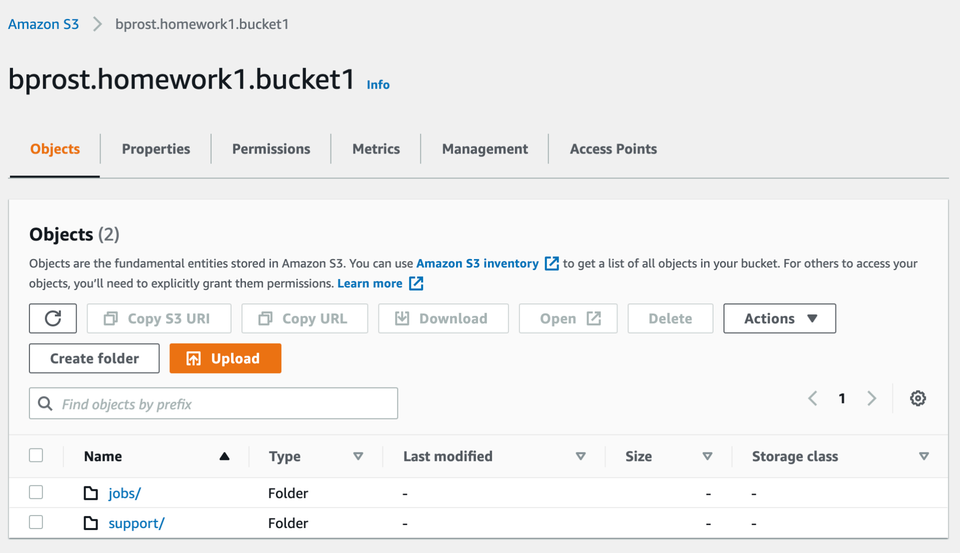
Brian Prost
Errol Waithe

SDEV 400 7380

November 1, 2021

Part One

Create two S3 buckets with DNS-compliant names. In each bucket, create a 'Jobs' and a 'Support' folder.



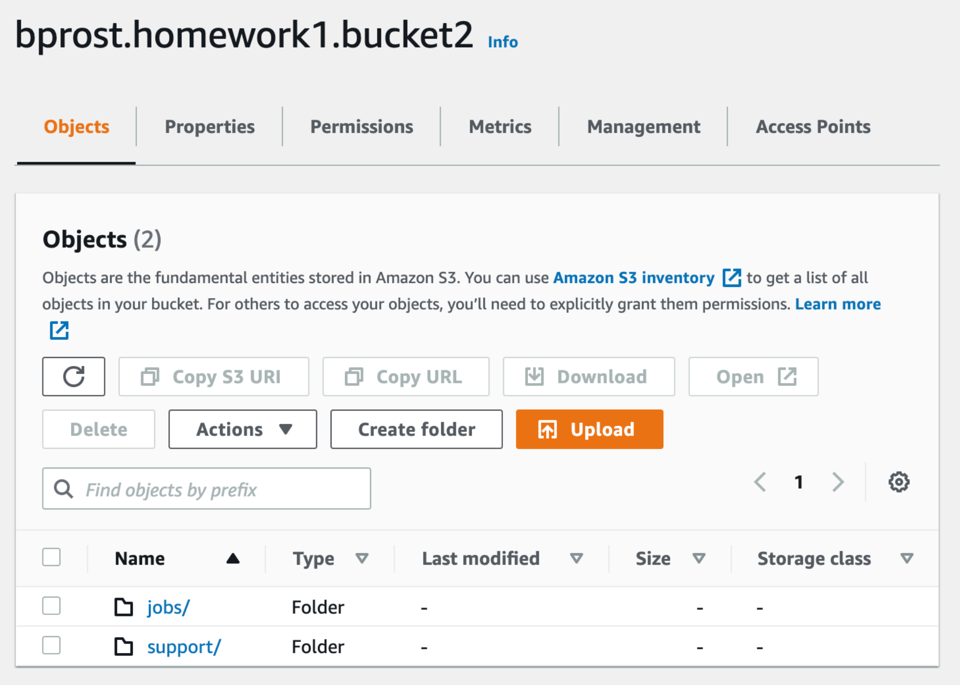
bprost.homework1.bucket1

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	jobs/	Folder	-	-	-
□	support/	Folder	-	-	-

Figure 1: Screenshot of first bucket containing two folders; jobs & support



bprost.homework1.bucket2

Objects (2)

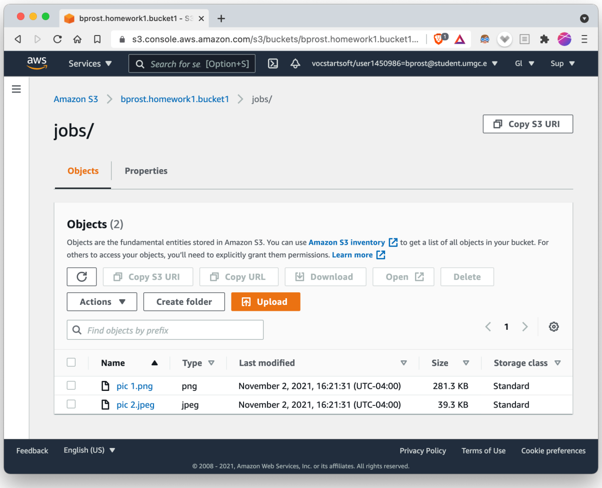
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	jobs/	Folder	-	-	-
□	support/	Folder	-	-	-

Figure 2: Screenshot of second bucket containing two folders; jobs & support

Part Two

Copy 2 files to each of the 2 folders we just created within our S3 buckets. We will be using some pictures, specifically some recent memes from the ProgrammerHumor subreddit.



Amazon S3 > bprost.homework1.bucket1 > jobs/

jobs/

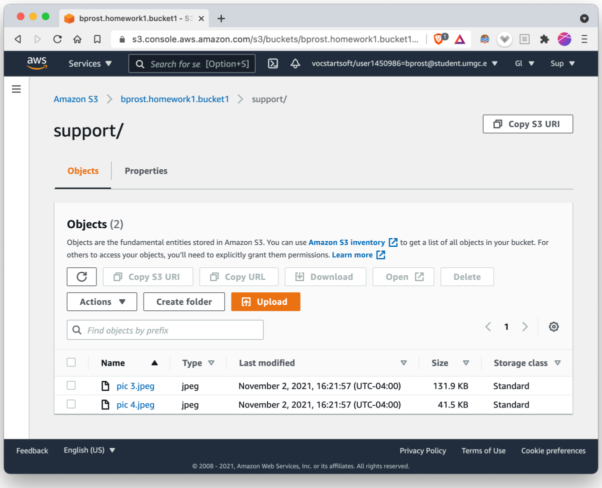
Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use **Amazon S3 inventory** to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. **Learn more**

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	pic 1.png	png	November 2, 2021, 16:21:31 (UTC-04:00)	281.3 KB	Standard
□	pic 2.jpeg	jpeg	November 2, 2021, 16:21:31 (UTC-04:00)	39.3 KB	Standard

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 3: Screenshot of bucket1/jobs



Amazon S3 > bprost.homework1.bucket1 > support/

support/

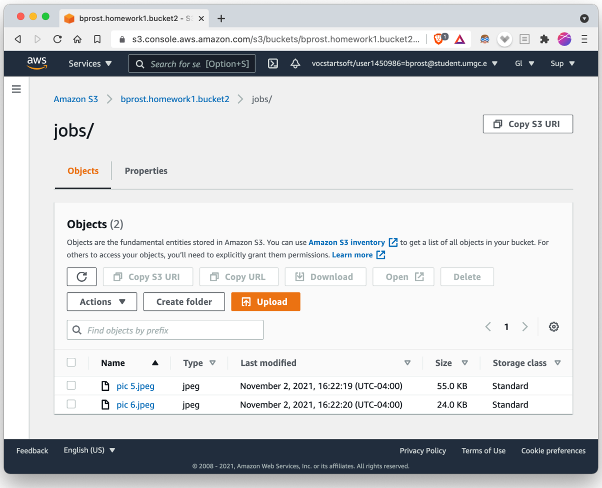
Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use **Amazon S3 inventory** to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. **Learn more**

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	pic 3.jpeg	jpeg	November 2, 2021, 16:21:57 (UTC-04:00)	131.9 KB	Standard
□	pic 4.jpeg	jpeg	November 2, 2021, 16:21:57 (UTC-04:00)	41.5 KB	Standard

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 4: Screenshot of bucket1/support



Amazon S3 > bprost.homework1.bucket2 > jobs/

jobs/

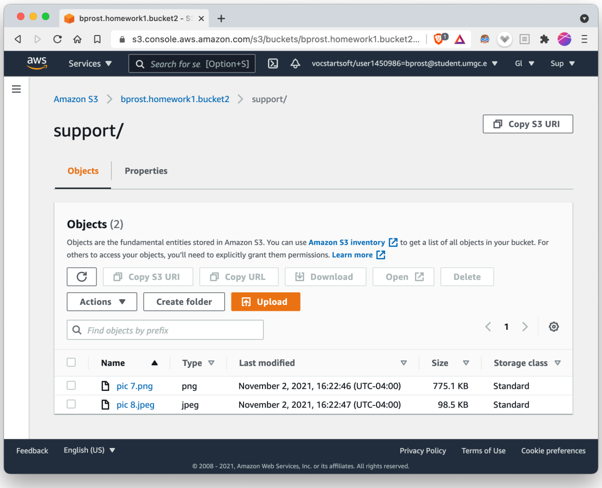
Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use **Amazon S3 inventory** to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. **Learn more**

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	pic 5.jpeg	jpeg	November 2, 2021, 16:22:19 (UTC-04:00)	55.0 KB	Standard
□	pic 6.jpeg	jpeg	November 2, 2021, 16:22:20 (UTC-04:00)	24.0 KB	Standard

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 5: Screenshot of bucket2/jobs



Amazon S3 > bprost.homework1.bucket2 > support/

support/

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use **Amazon S3 inventory** to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. **Learn more**

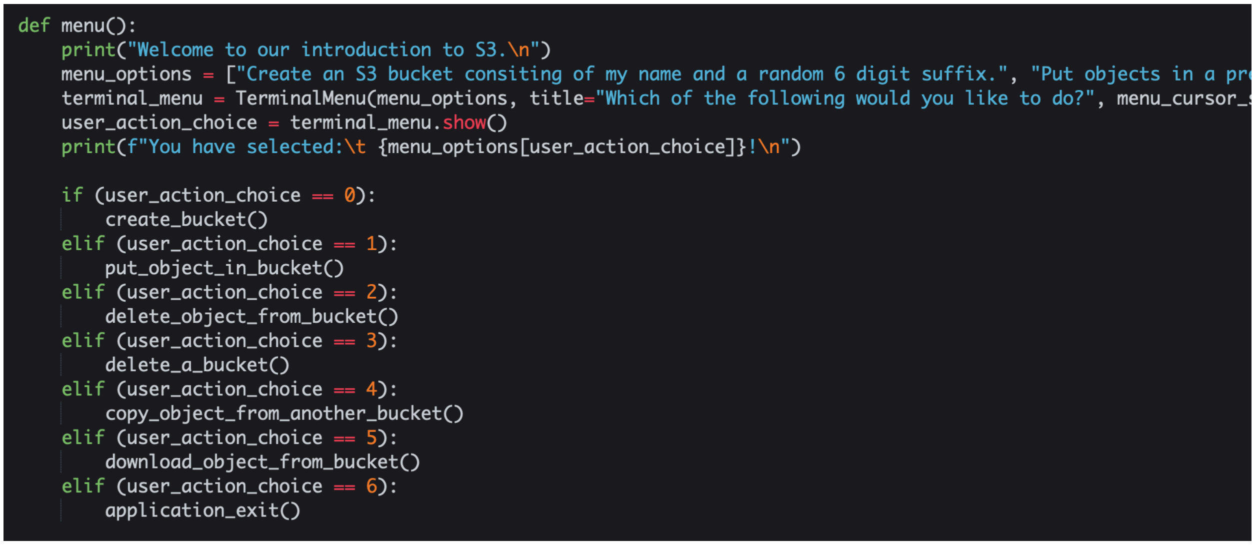
□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	pic 7.png	png	November 2, 2021, 16:22:46 (UTC-04:00)	775.1 KB	Standard
□	pic 8.jpeg	jpeg	November 2, 2021, 16:22:47 (UTC-04:00)	98.5 KB	Standard

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 6: Screenshot of bucket2/support

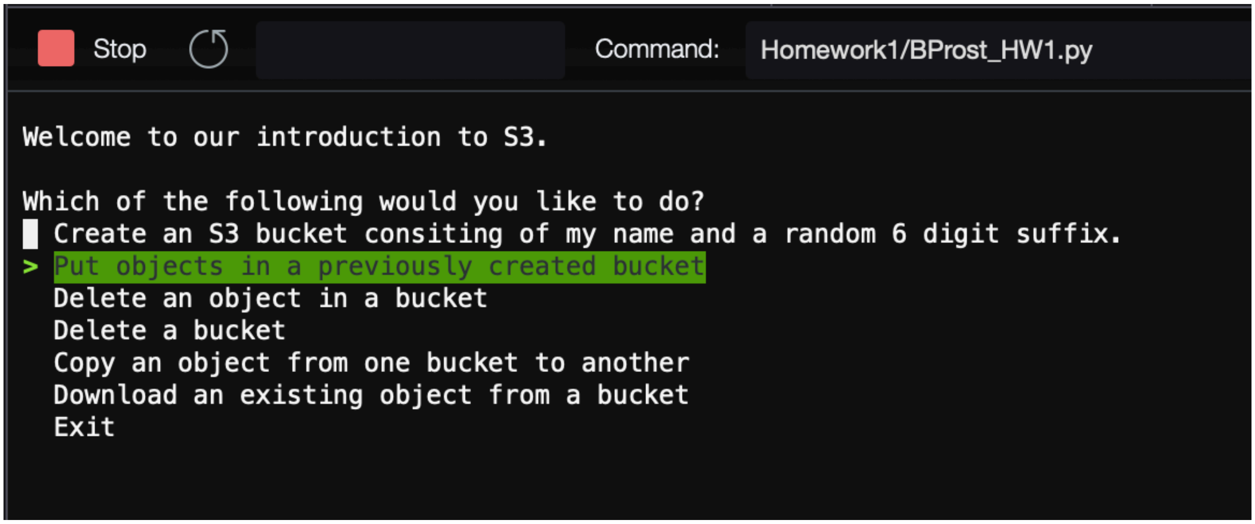
Part Three

To make our Python command line application, we need to make a menu. I will be doing this using simple_term_menu's Terminal Menu, which allows for a classic arrow driven menu interface where you can select your option based by highlighting the choice you want. Here is the code and the menu in action:



```python
def menu():
    print("Welcome to our introduction to S3.\n")
    menu_options = ["Create an S3 bucket consiting of my name and a random 6 digit suffix.", "Put objects in a pre
    terminal_menu = TerminalMenu(menu_options, title="Which of the following would you like to do?", menu_cursor_
    user_action_choice = terminal_menu.show()
    print(f"You have selected:\t {menu_options[user_action_choice]}!\n")

    if (user_action_choice == 0):
        create_bucket()
    elif (user_action_choice == 1):
        put_object_in_bucket()
    elif (user_action_choice == 2):
        delete_object_from_bucket()
    elif (user_action_choice == 3):
        delete_a_bucket()
    elif (user_action_choice == 4):
        copy_object_from_another_bucket()
    elif (user_action_choice == 5):
        download_object_from_bucket()
    elif (user_action_choice == 6):
        application_exit()
```

Figure 7: Python code for menu interface inside Cloud9 IDE



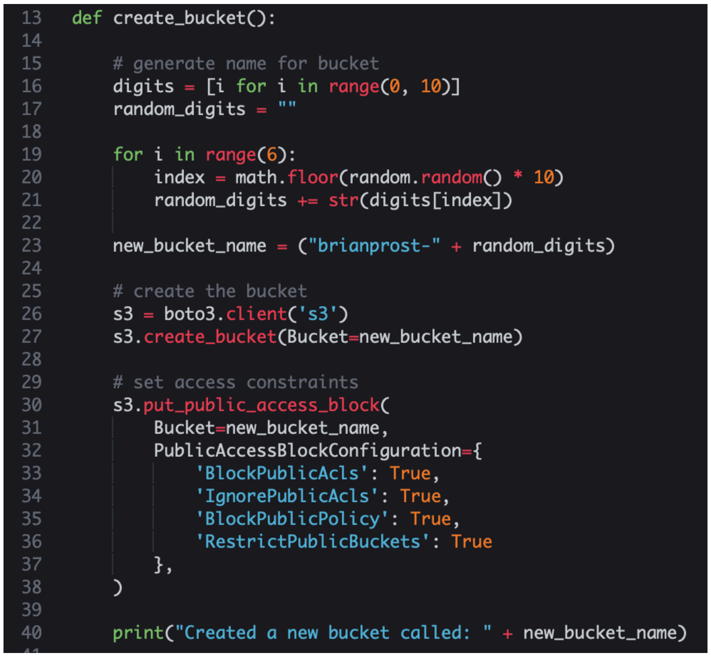
```
Welcome to our introduction to S3.

Which of the following would you like to do?
█ Create an S3 bucket consiting of my name and a random 6 digit suffix.
> Put objects in a previously created bucket
  Delete an object in a bucket
  Delete a bucket
  Copy an object from one bucket to another
  Download an existing object from a bucket
  Exit
```

Figure 8: Our menu in action in the terminal in Cloud9

Once the user makes a selection, the menu returns a corresponding integer, which will filter itself through an if-else loop to launch the correct operation. Let's go through these in order.

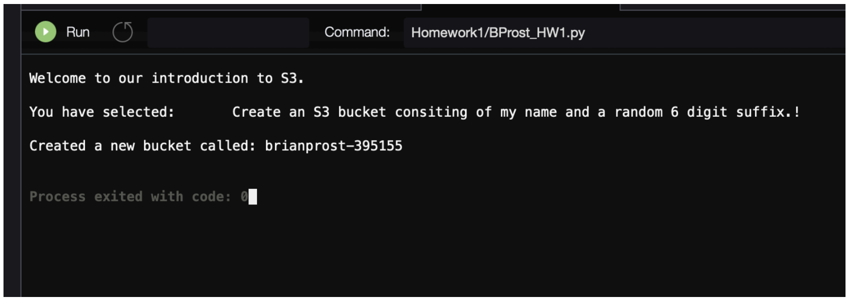
Create S3 Bucket

To create this bucket, we create a random 6 digit code using Python's random and math libraries, and then amend these to a string of 'brianprost' with a dash at the end. We then call the boto3 method to create_bucket, and then set the access constraints for the bucket to be not for all to see.



```python
def create_bucket():

    # generate name for bucket
    digits = [i for i in range(0, 10)]
    random_digits = ""

    for i in range(6):
        index = math.floor(random.random() * 10)
        random_digits += str(digits[index])

    new_bucket_name = ("brianprost-" + random_digits)

    # create the bucket
    s3 = boto3.client('s3')
    s3.create_bucket(Bucket=new_bucket_name)

    # set access constraints
    s3.put_public_access_block(
        Bucket=new_bucket_name,
        PublicAccessBlockConfiguration={
            'BlockPublicAcls': True,
            'IgnorePublicAcls': True,
            'BlockPublicPolicy': True,
            'RestrictPublicBuckets': True
        },
    )

    print("Created a new bucket called: " + new_bucket_name)
```

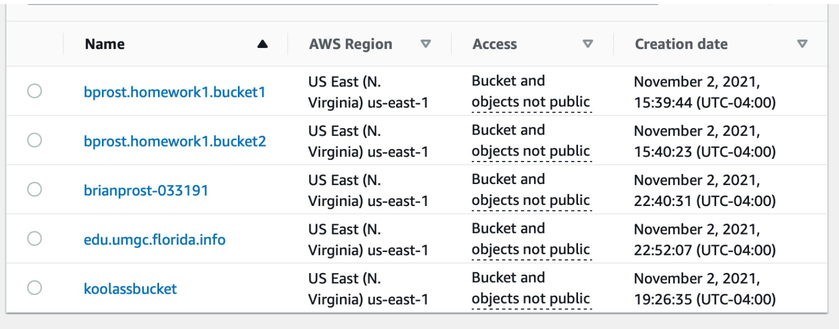
Figure 9: My method for creating a new bucket



Command: Homework1/BProst_HW1.py

```
Welcome to our introduction to S3.

You have selected:      Create an S3 bucket consiting of my name and a random 6 digit suffix.!

Created a new bucket called: brianprost-395155


Process exited with code: 0
```

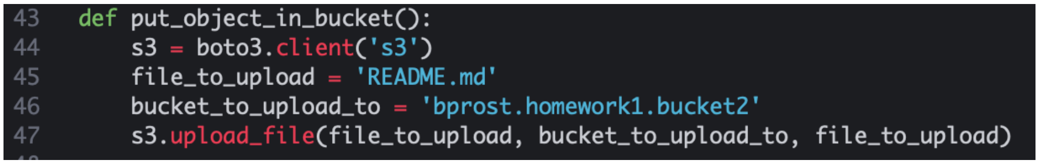
Figure 10: Screenshot of creation of new bucket.



	Name ▲	AWS Region ▽	Access ▽	Creation date ▽
○	bprost.homework1.bucket1	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 15:39:44 (UTC-04:00)
○	bprost.homework1.bucket2	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 15:40:23 (UTC-04:00)
○	brianprost-033191	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 22:40:31 (UTC-04:00)
○	edu.umgc.florida.info	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 22:52:07 (UTC-04:00)
○	koolassbucket	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 19:26:35 (UTC-04:00)

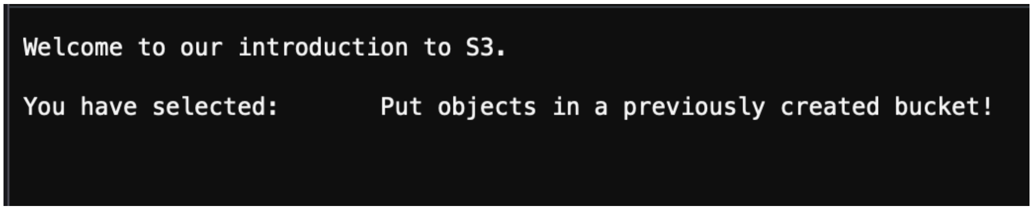
Figure 11: Screenshot of S3 Homepage to verify creation

Put objects in a previously created bucket



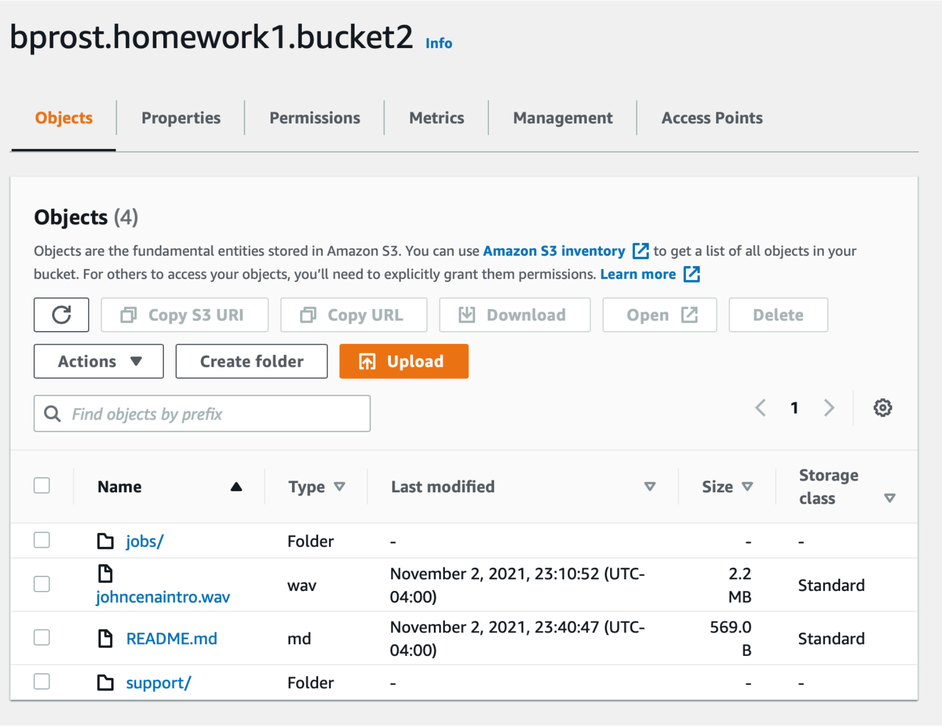
```python
def put_object_in_bucket():
    s3 = boto3.client('s3')
    file_to_upload = 'README.md'
    bucket_to_upload_to = 'bprost.homework1.bucket2'
    s3.upload_file(file_to_upload, bucket_to_upload_to, file_to_upload)
```

Figure 12: Screenshot of method to place an object in a previously created bucket



```
Welcome to our introduction to S3.

You have selected:       Put objects in a previously created bucket!
```

Figure 13: Terminal output from selecting putting objects in a previously created bucket.



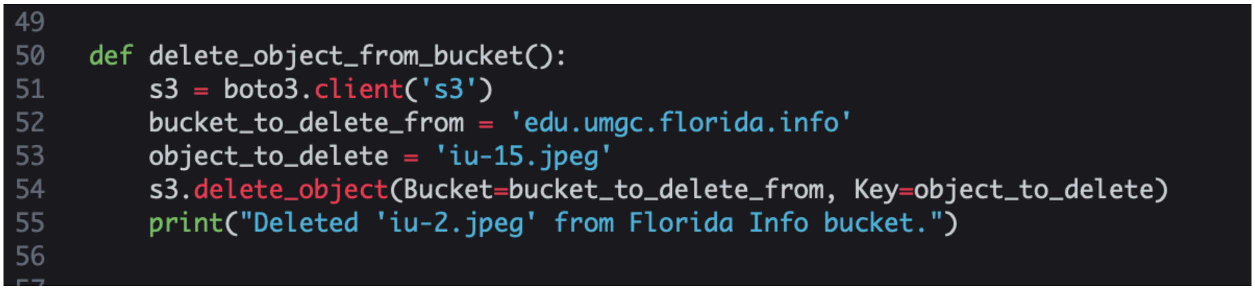
Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	jobs/	Folder	-	-	-
□	johncenaintro.wav	wav	November 2, 2021, 23:10:52 (UTC-04:00)	2.2 MB	Standard
□	README.md	md	November 2, 2021, 23:40:47 (UTC-04:00)	569.0 B	Standard
□	support/	Folder	-	-	-

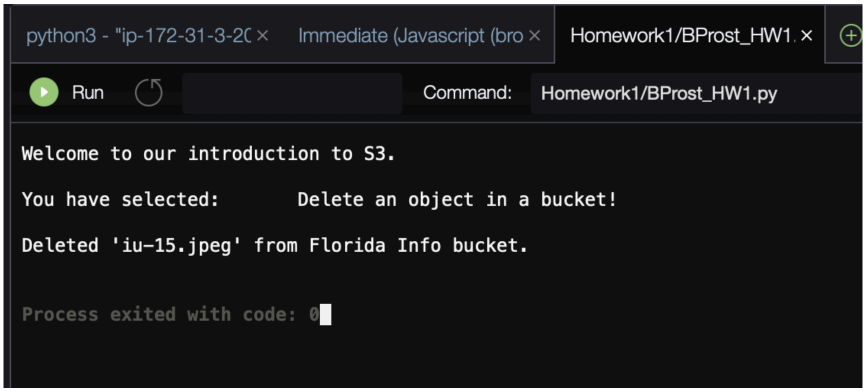
Figure 14: a successful upload of README.md to bucket2

Deleting an object from a a bucket



```python

def delete_object_from_bucket():
    s3 = boto3.client('s3')
    bucket_to_delete_from = 'edu.umgc.florida.info'
    object_to_delete = 'iu-15.jpeg'
    s3.delete_object(Bucket=bucket_to_delete_from, Key=object_to_delete)
    print("Deleted 'iu-2.jpeg' from Florida Info bucket.")

```

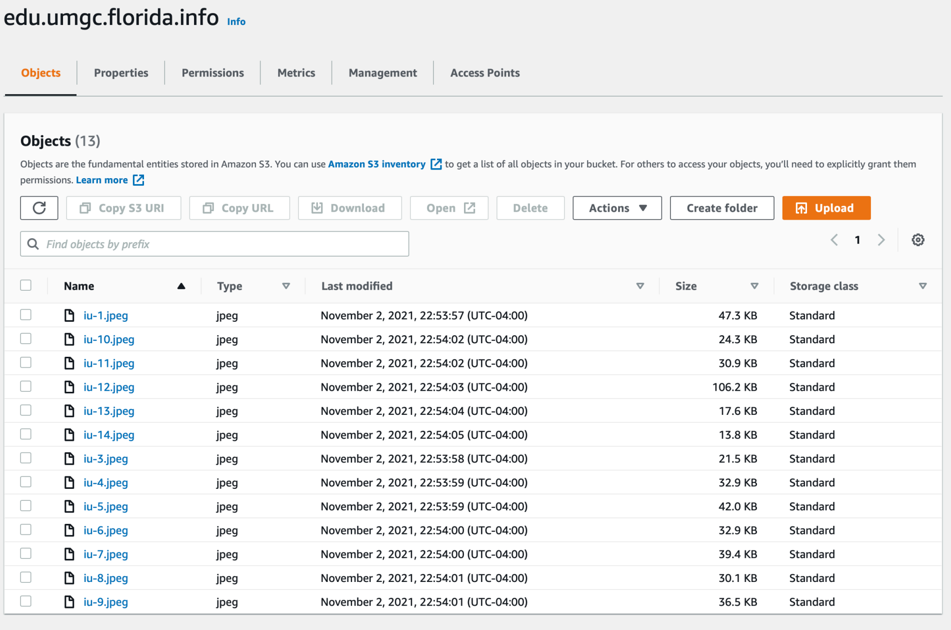
Figure 15: Code for deleting an object from a bucket



Command: Homework1/BProst_HW1.py

```
Welcome to our introduction to S3.

You have selected:       Delete an object in a bucket!

Deleted 'iu-15.jpeg' from Florida Info bucket.


Process exited with code: 0
```

Figure 16: Terminal output of selecting ‘delete an object from a bucket’



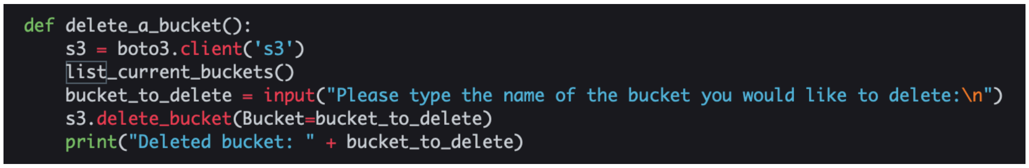
Objects (13)

Objects are the fundamental entities stored in Amazon S3. You can use **Amazon S3 inventory** to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. **Learn more**

□	Name ▲	Type	Last modified	Size	Storage class
□	iu-1.jpeg	jpeg	November 2, 2021, 22:53:57 (UTC-04:00)	47.3 KB	Standard
□	iu-10.jpeg	jpeg	November 2, 2021, 22:54:02 (UTC-04:00)	24.3 KB	Standard
□	iu-11.jpeg	jpeg	November 2, 2021, 22:54:02 (UTC-04:00)	30.9 KB	Standard
□	iu-12.jpeg	jpeg	November 2, 2021, 22:54:03 (UTC-04:00)	106.2 KB	Standard
□	iu-13.jpeg	jpeg	November 2, 2021, 22:54:04 (UTC-04:00)	17.6 KB	Standard
□	iu-14.jpeg	jpeg	November 2, 2021, 22:54:05 (UTC-04:00)	13.8 KB	Standard
□	iu-3.jpeg	jpeg	November 2, 2021, 22:53:58 (UTC-04:00)	21.5 KB	Standard
□	iu-4.jpeg	jpeg	November 2, 2021, 22:53:59 (UTC-04:00)	32.9 KB	Standard
□	iu-5.jpeg	jpeg	November 2, 2021, 22:53:59 (UTC-04:00)	42.0 KB	Standard
□	iu-6.jpeg	jpeg	November 2, 2021, 22:54:00 (UTC-04:00)	32.9 KB	Standard
□	iu-7.jpeg	jpeg	November 2, 2021, 22:54:00 (UTC-04:00)	39.4 KB	Standard
□	iu-8.jpeg	jpeg	November 2, 2021, 22:54:01 (UTC-04:00)	30.1 KB	Standard
□	iu-9.jpeg	jpeg	November 2, 2021, 22:54:01 (UTC-04:00)	36.5 KB	Standard

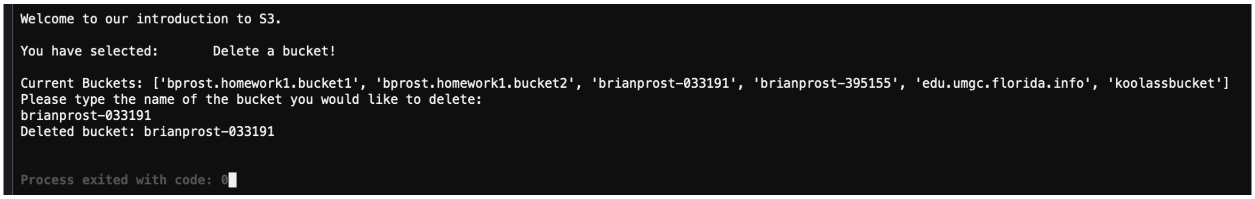
Figure 17: bucket edu.umgc.florida.info has one less picture

Delete a bucket



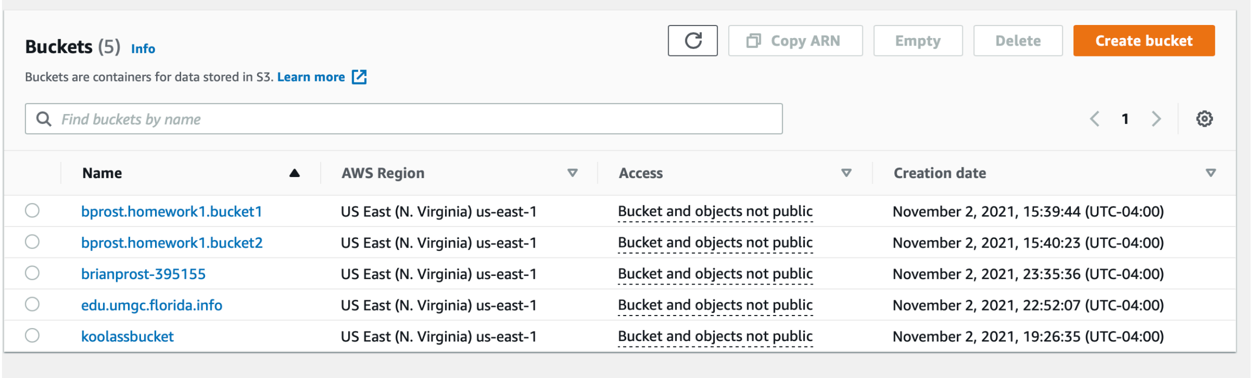
```python
def delete_a_bucket():
    s3 = boto3.client('s3')
    list_current_buckets()
    bucket_to_delete = input("Please type the name of the bucket you would like to delete:\n")
    s3.delete_bucket(Bucket=bucket_to_delete)
    print("Deleted bucket: " + bucket_to_delete)
```

Figure 18: code to delete a bucket



```
Welcome to our introduction to S3.

You have selected:       Delete a bucket!

Current Buckets: ['bprost.homework1.bucket1', 'bprost.homework1.bucket2', 'brianprost-033191', 'brianprost-395155', 'edu.umgc.florida.info', 'koolassbucket']
Please type the name of the bucket you would like to delete:
brianprost-033191
Deleted bucket: brianprost-033191


Process exited with code: 0
```

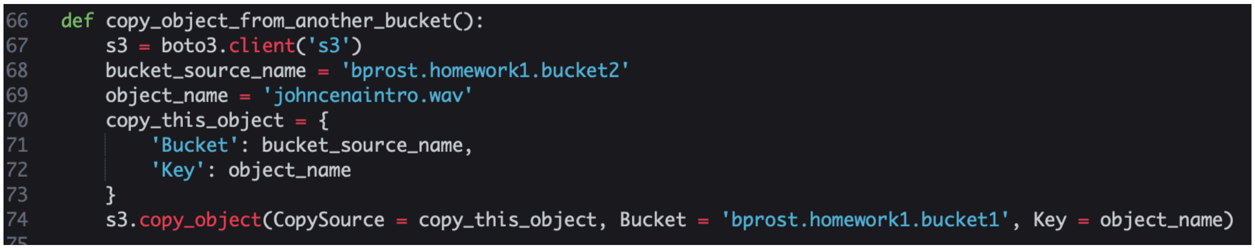
Figure 19: Terminal output from the delete_a_bucket( ) method



	Name	AWS Region	Access	Creation date
○	bprost.homework1.bucket1	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 15:39:44 (UTC-04:00)
○	bprost.homework1.bucket2	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 15:40:23 (UTC-04:00)
○	brianprost-395155	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 23:35:36 (UTC-04:00)
○	edu.umgc.florida.info	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 22:52:07 (UTC-04:00)
○	koolassbucket	US East (N. Virginia) us-east-1	Bucket and objects not public	November 2, 2021, 19:26:35 (UTC-04:00)

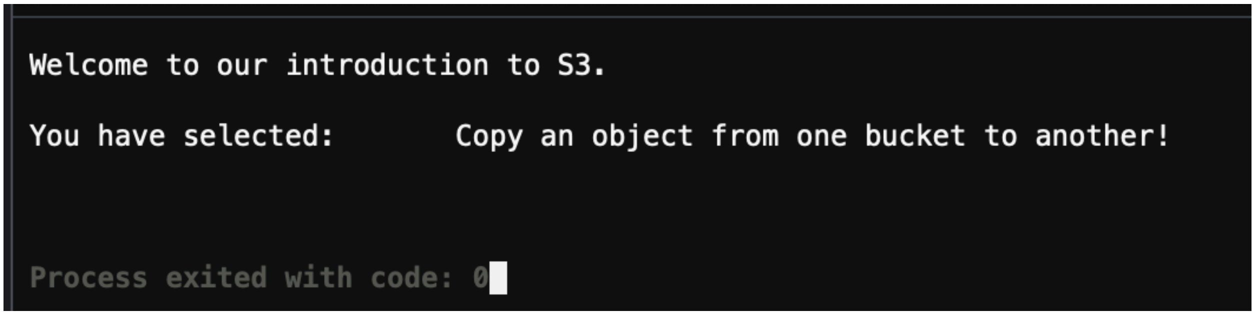
Figure 20: AWS Console list of buckets, with brianprost-033191 no longer showing up

Copy an object from one bucket to another



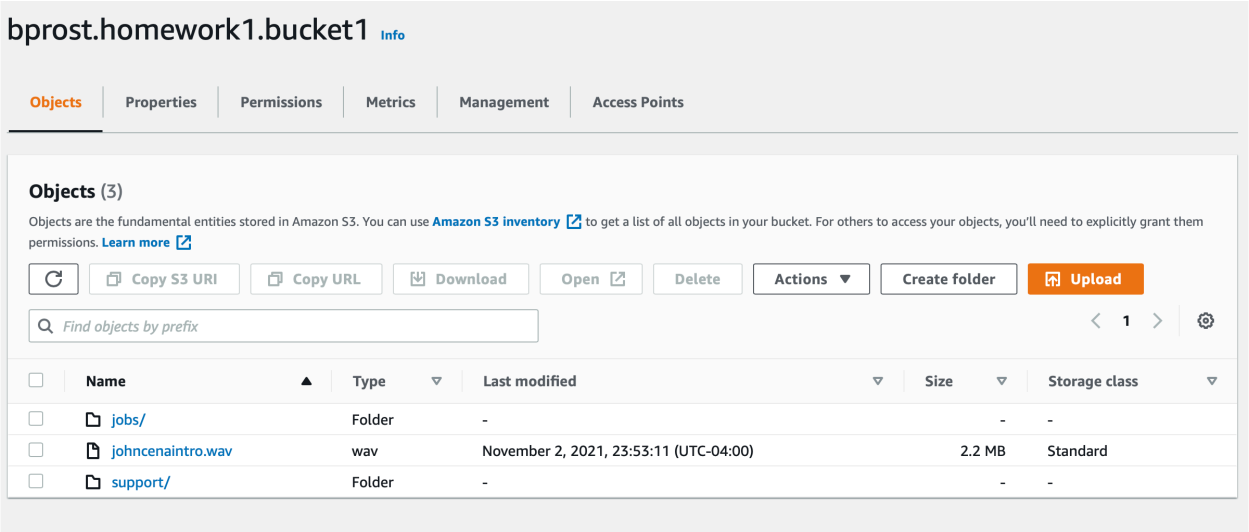
```python
def copy_object_from_another_bucket():
    s3 = boto3.client('s3')
    bucket_source_name = 'bprost.homework1.bucket2'
    object_name = 'johncenaintro.wav'
    copy_this_object = {
        'Bucket': bucket_source_name,
        'Key': object_name
    }
    s3.copy_object(CopySource = copy_this_object, Bucket = 'bprost.homework1.bucket1', Key = object_name)
```

Figure 21: Method for copying one object to another bucket



```
Welcome to our introduction to S3.

You have selected:       Copy an object from one bucket to another!



Process exited with code: 0
```

Figure 22: Terminal output for copying one object to another



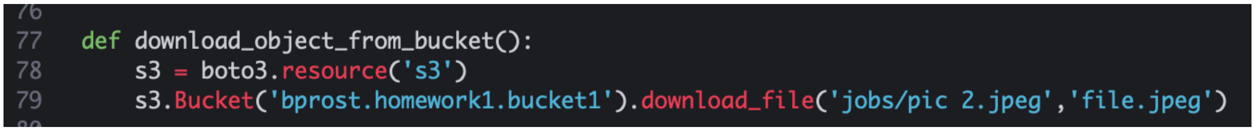
Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use **Amazon S3 inventory** to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. **Learn more**

□	Name ▲	Type	Last modified	Size	Storage class
□	jobs/	Folder	-	-	-
□	johncenaintro.wav	wav	November 2, 2021, 23:53:11 (UTC-04:00)	2.2 MB	Standard
□	support/	Folder	-	-	-

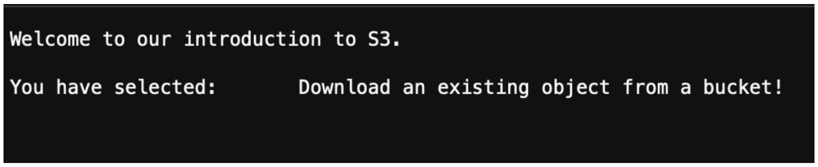
Figure 23: AWS Console showing that ‘johncenaintro.wav’ did in fact make it to the other bucket

Download an existing object from a bucket



```
def download_object_from_bucket():
    s3 = boto3.resource('s3')
    s3.Bucket('bprost.homework1.bucket1').download_file('jobs/pic 2.jpeg','file.jpeg')
```

Figure 24: Method for downloading one object from a bucket



```
Welcome to our introduction to S3.

You have selected:       Download an existing object from a bucket!
```

Figure 25: Terminal Output

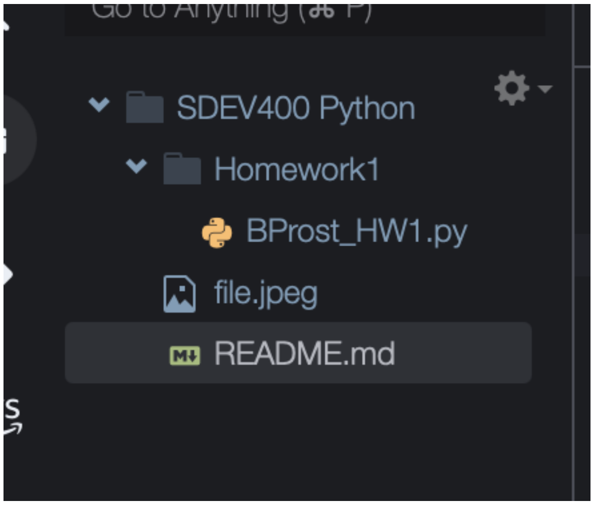
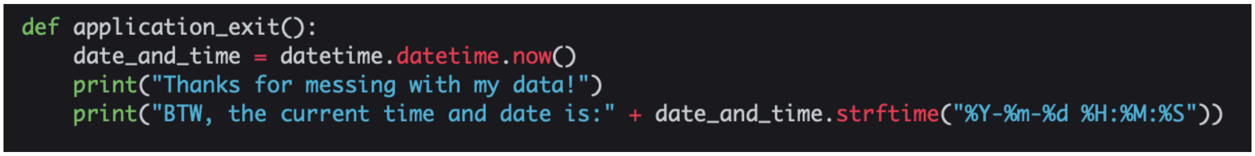


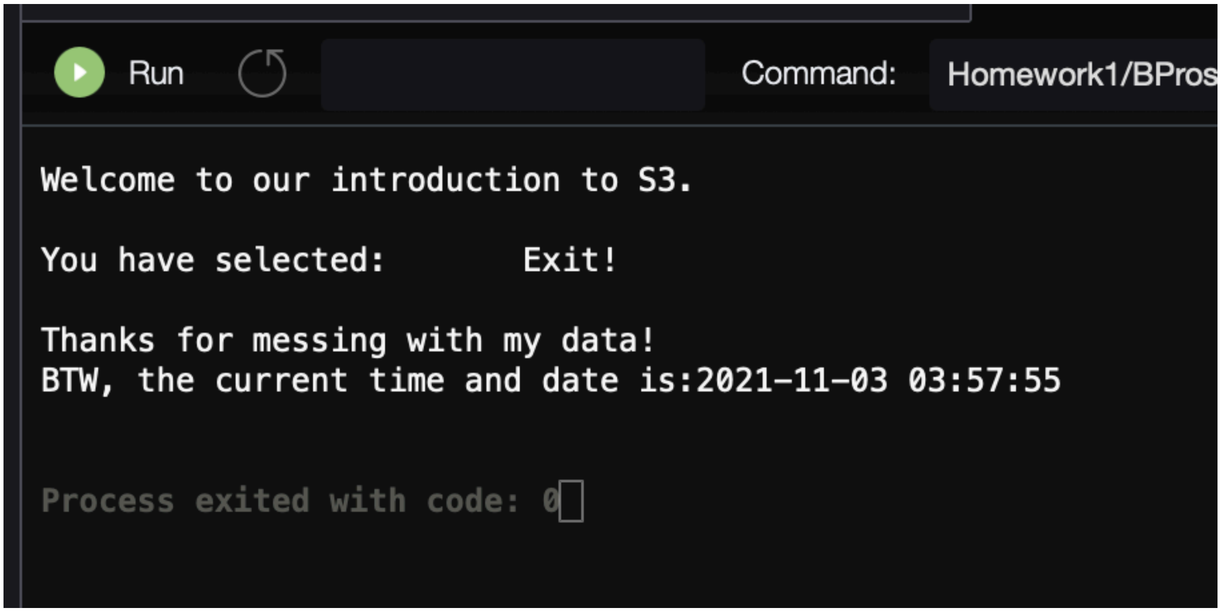
Figure 26: file.jpeg was downloaded into our folder

Exit the application



```
def application_exit():
    date_and_time = datetime.datetime.now()
    print("Thanks for messing with my data!")
    print("BTW, the current time and date is:" + date_and_time.strftime("%Y-%m-%d %H:%M:%S"))
```

Figure 27: Method for closing application with date & time



```
Welcome to our introduction to S3.

You have selected:       Exit!

Thanks for messing with my data!
BTW, the current time and date is:2021-11-03 03:57:55


Process exited with code: 0
```

Figure 28: Terminal output