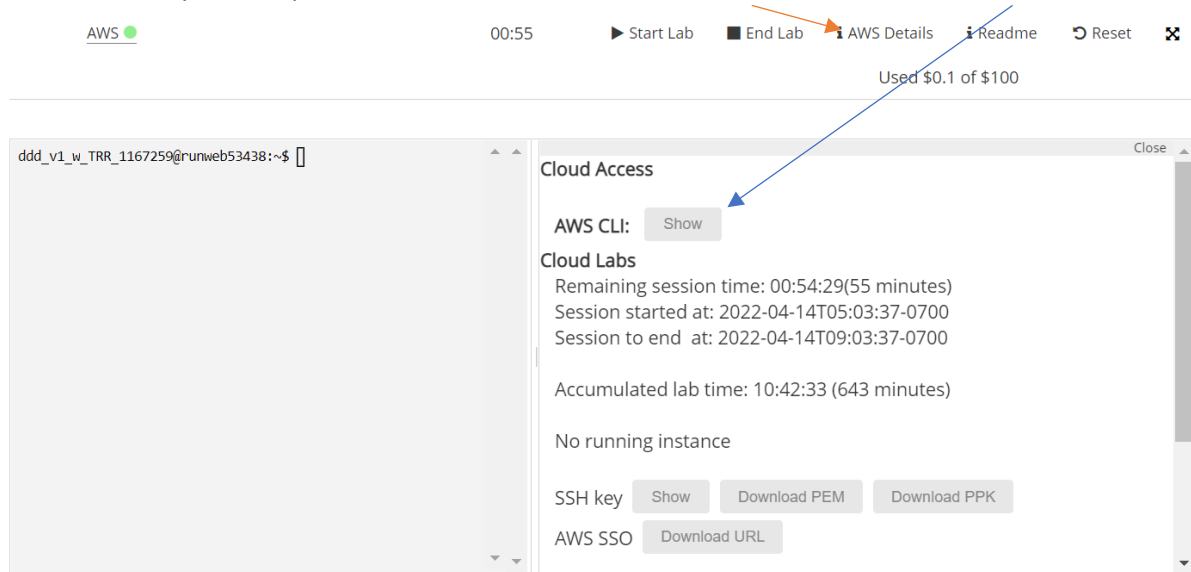


SPARKIFY 3 – Upload logs to S3 using Boto3

<<TASK 0>> Create an instance and add AWS credentials

1. Create an Amazon Linux instance (as done in Sparkify 1)
2. SSH into the instance
3. In AWS academy; where you launched the AWS; click on AWS Details and then click show AWS CLI



4. In SSH; go to home if not already there (`cd ~`)
5. Create new folder:
`mkdir .aws`
6. Create a new file
`vim .aws/credentials`
7. paste entire AWS CLI content – as it is – save and quit
[you **need to modify the credentials file on every launch** since AWS CLI is not constant]

<<TASK 1>> Install Boto3

1. create a virtual python environment (if not already exists) and activate it
[`python3 -m venv venv`
`source venv/bin/activate`]
2. `pip install boto3`
3. to check if installation successful, run python [type python3] then follow these steps:
 - `import boto3`
 - `s3 = boto3.resource('s3')`
 - `for bucket in s3.buckets.all():`
`print(bucket.name)`
 - you should be able to see all the buckets in your s3*

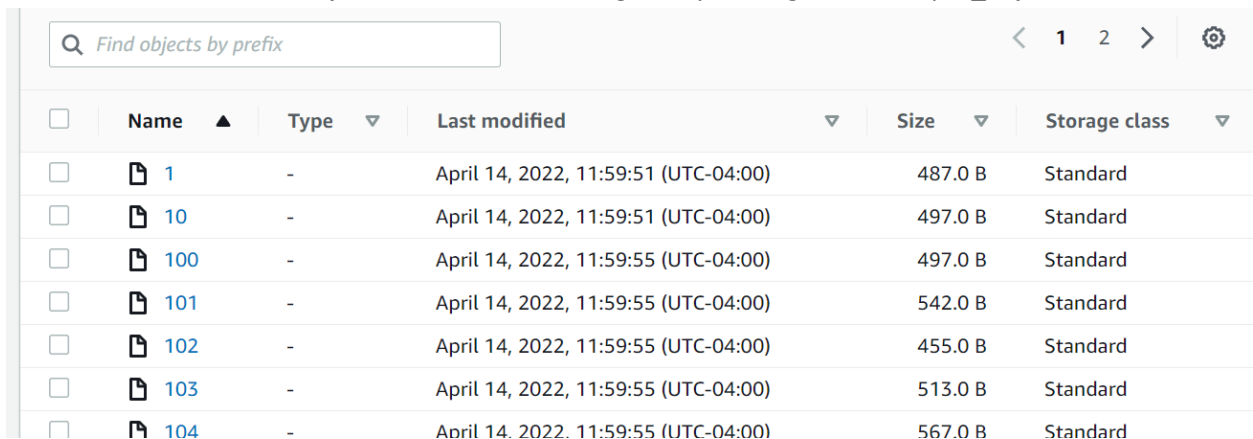
```
>>> import boto3
>>> s3 = boto3.resource('s3')
>>> for bucket in s3.buckets.all():
...     print(bucket.name)
...
aws-logs-352275169258-us-east-1
dsci6007yshah
>>>
```

* If you face any issues (eg - access denied); confirm that your credentials file was created properly

SPARKIFY 3 – Upload logs to S3 using Boto3

<<TASK 2>> Upload logs to S3

0. to be able to upload the log files; make sure the log files actually exist in your instance
[already present from Sparkify 1; if not exists, follow steps from TASK 2 in Sparkify 1 tutorial]
[if you are transferring files here;
make sure to deactivate your python venv – activate again after transferring]
1. Create a .py file to upload log files to S3 bucket
 - Import all required packages
 - [Create bucket](#)
 - [Upload files](#)
2. Run the file – python3 fileForUpload.py
3. Go to S3 [choose the bucket you created]
and confirm that your files have been uploaded
[see screenshot below – object names are according to key value given in the put_object method]



The screenshot shows the AWS S3 console interface. At the top, there is a search bar with the placeholder text "Find objects by prefix". To the right of the search bar are navigation controls: a left arrow, the number "1", the number "2", a right arrow, and a settings gear icon. Below the search bar is a table with the following columns: "Name", "Type", "Last modified", "Size", and "Storage class". The table contains eight rows of data, each representing an uploaded log file. The "Name" column shows files named "1", "10", "100", "101", "102", "103", and "104". The "Last modified" column shows the date and time for each file, all on April 14, 2022, at 11:59:51 or 11:59:55 UTC-04:00. The "Size" column shows the file sizes in bytes, ranging from 487.0 B to 567.0 B. The "Storage class" column shows that all files are stored in the "Standard" storage class.

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	1	-	April 14, 2022, 11:59:51 (UTC-04:00)	487.0 B	Standard
<input type="checkbox"/>	10	-	April 14, 2022, 11:59:51 (UTC-04:00)	497.0 B	Standard
<input type="checkbox"/>	100	-	April 14, 2022, 11:59:55 (UTC-04:00)	497.0 B	Standard
<input type="checkbox"/>	101	-	April 14, 2022, 11:59:55 (UTC-04:00)	542.0 B	Standard
<input type="checkbox"/>	102	-	April 14, 2022, 11:59:55 (UTC-04:00)	455.0 B	Standard
<input type="checkbox"/>	103	-	April 14, 2022, 11:59:55 (UTC-04:00)	513.0 B	Standard
<input type="checkbox"/>	104	-	April 14, 2022, 11:59:55 (UTC-04:00)	567.0 B	Standard

SPARKIFY 3 – Upload logs to S3 using Boto3

uploadToS3.py

```
import os
import boto3
import json
import logging
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

def upload_logData():
    # where the log data resides in the instance
    dire = r"/home/ec2-user/log_data/"
    s3 = boto3.client("s3")
    bucket = "dsci6007yshah1"
    # key is the file name - I have just used numbers
    key = 1
    for filename in os.listdir(dire):
        with open(dire + filename, encoding="utf-8") as f:
            for jsonObj in f:
                dic = json.loads(jsonObj)
                dic = json.dumps(dic)
                s3.put_object(Body=dic, Bucket=bucket, Key=str(key))
                key += 1

# make sure to give a unique (non existing) bucket name
create_bucket("dsci6007yshah1")
upload_logData()
```