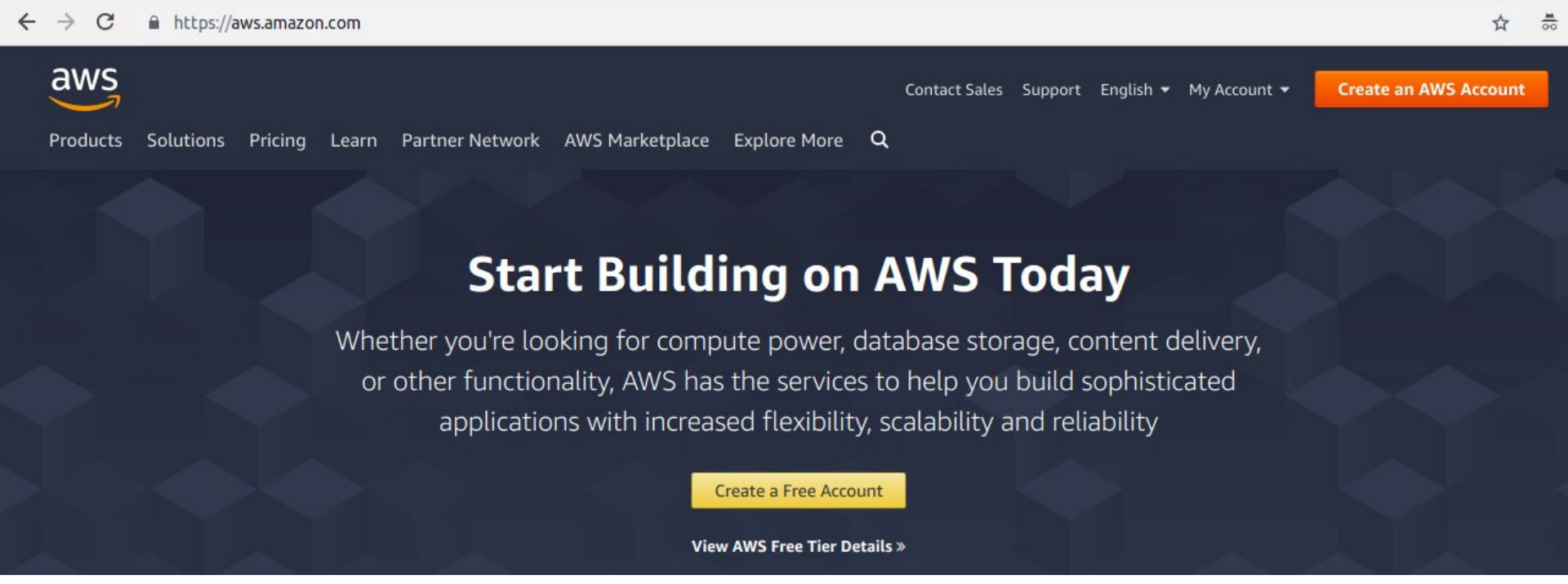
PyWren

Scale your existing code via Amazon Lambda
(Run code without managing cloud servers)

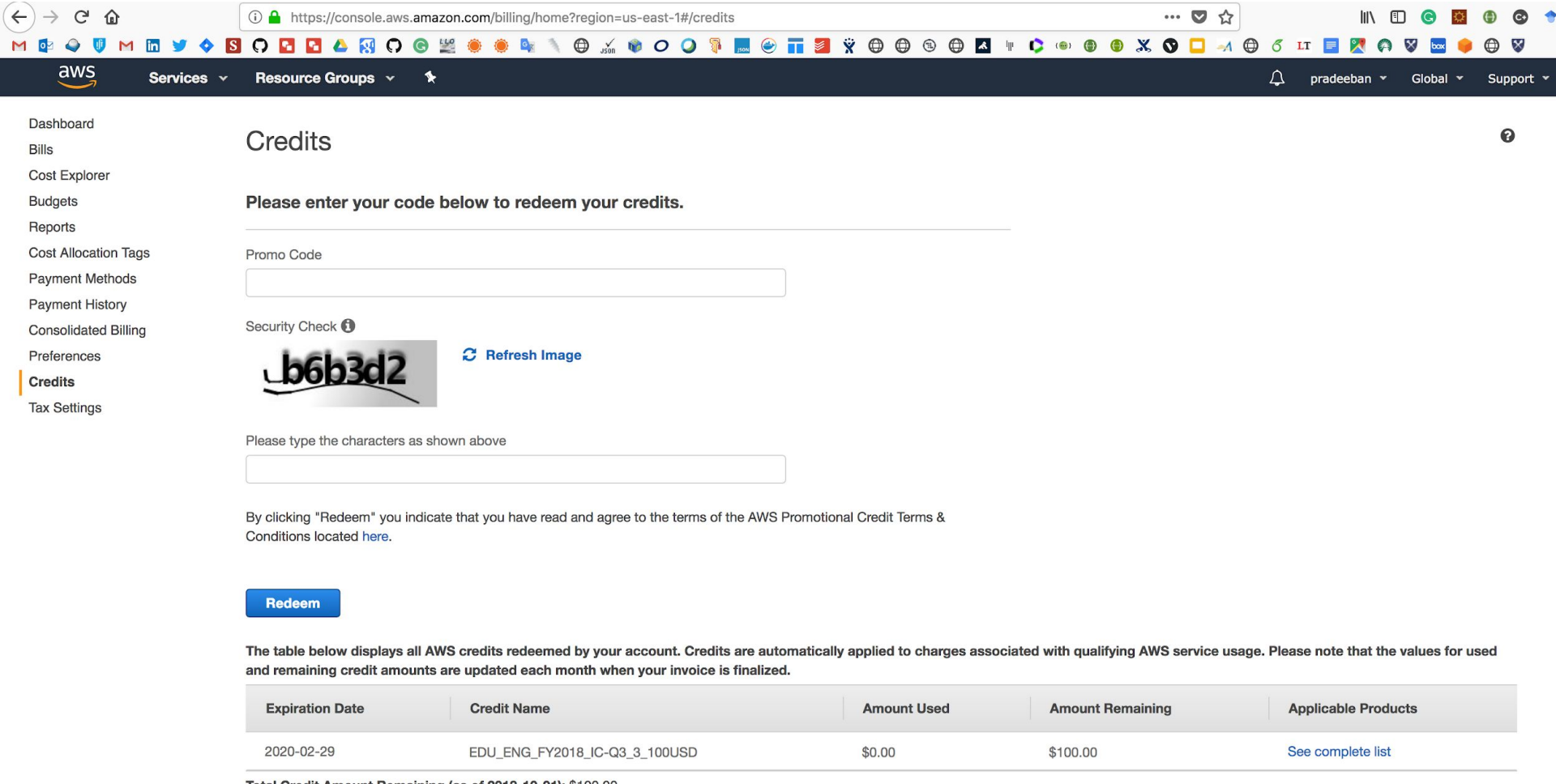
http://pywren.io/

Sign up to Amazon Web Services (AWS)

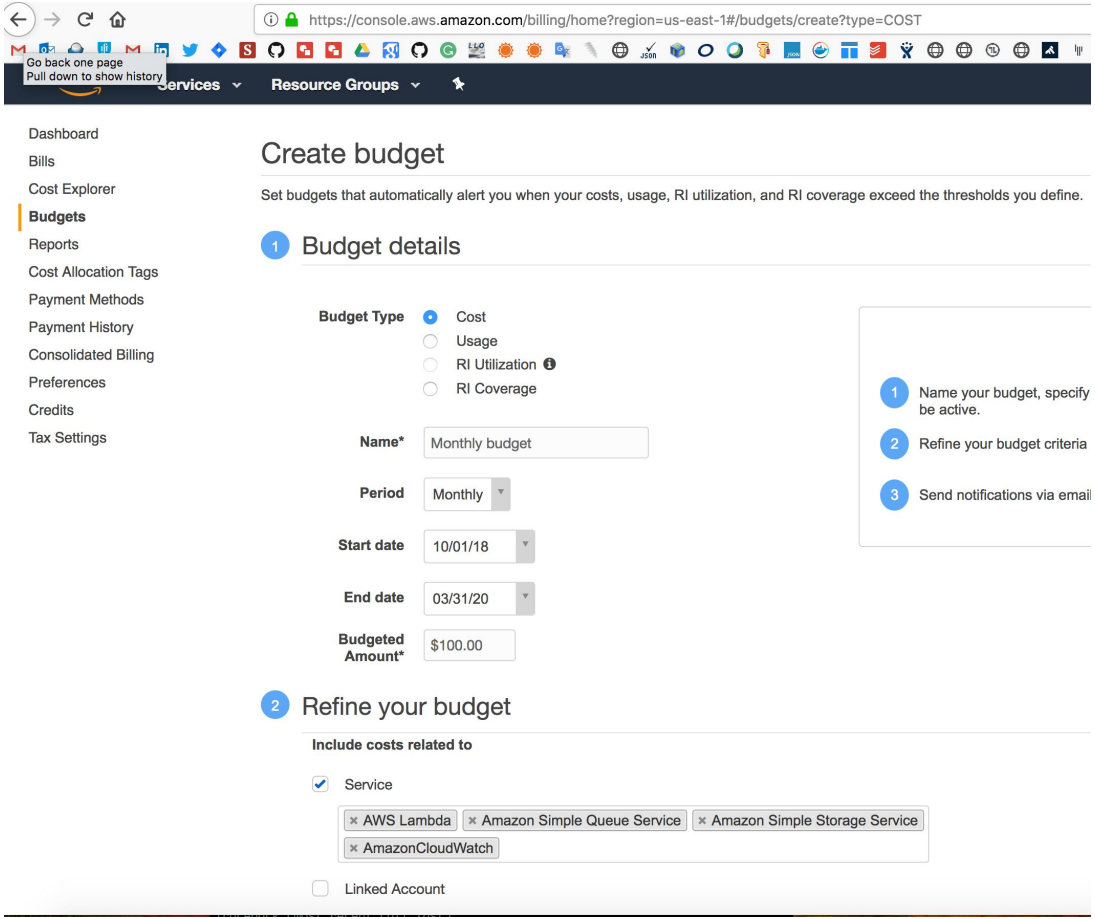
- Use your emory email address → 100 $ credits.

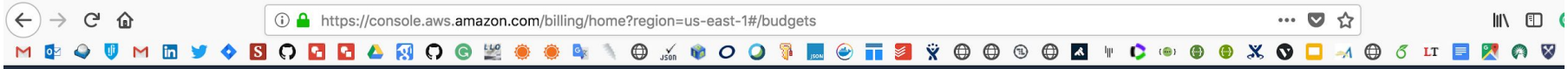


Check your credits



Create budgets





Dashboard
Bills
Cost Explorer
Budgets
Reports
Cost Allocation Tags
Payment Methods
Payment History
Consolidated Billing
Preferences
Credits
Tax Settings

AWS Budgets

Create budget | Copy | Edit | Delete | Download CSV

Filter by budget name

	Budget ty..	Budget name	Current	Forecasted	Budgeted	Current vs. budgeted	Forecasted vs. budgeted
☐	Cost	Monthly budget	$0.01	$0.50	$100.00	0%	1%

Budget details

Start date 10/01/18

End date 03/31/20

Budget period Monthly

Variance analysis

Budget remaining $100.00

% of budget remaining 100%

Forecasted budget remaining $99.50

Forecasted % of budget remaining 99%

Filters

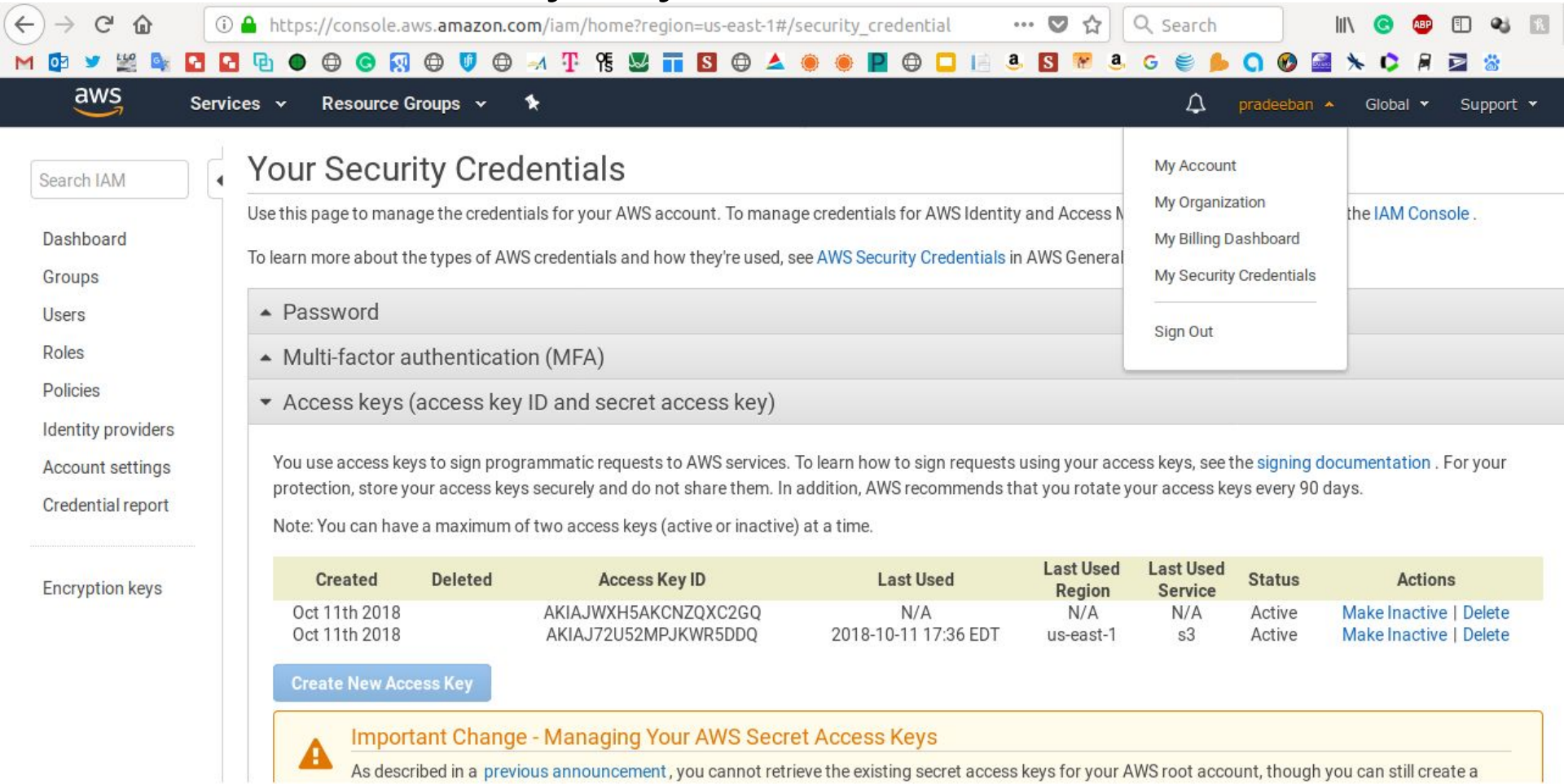
Service AWS Lambda, Amazon Simple Queue Service, Amazon Simple Storage Service, AmazonCloudWatch

Advanced filters

Include Refunds, Recurring reservation charges, Other subscription costs, Support charges, Credits, Taxes, Upfront reservation fees

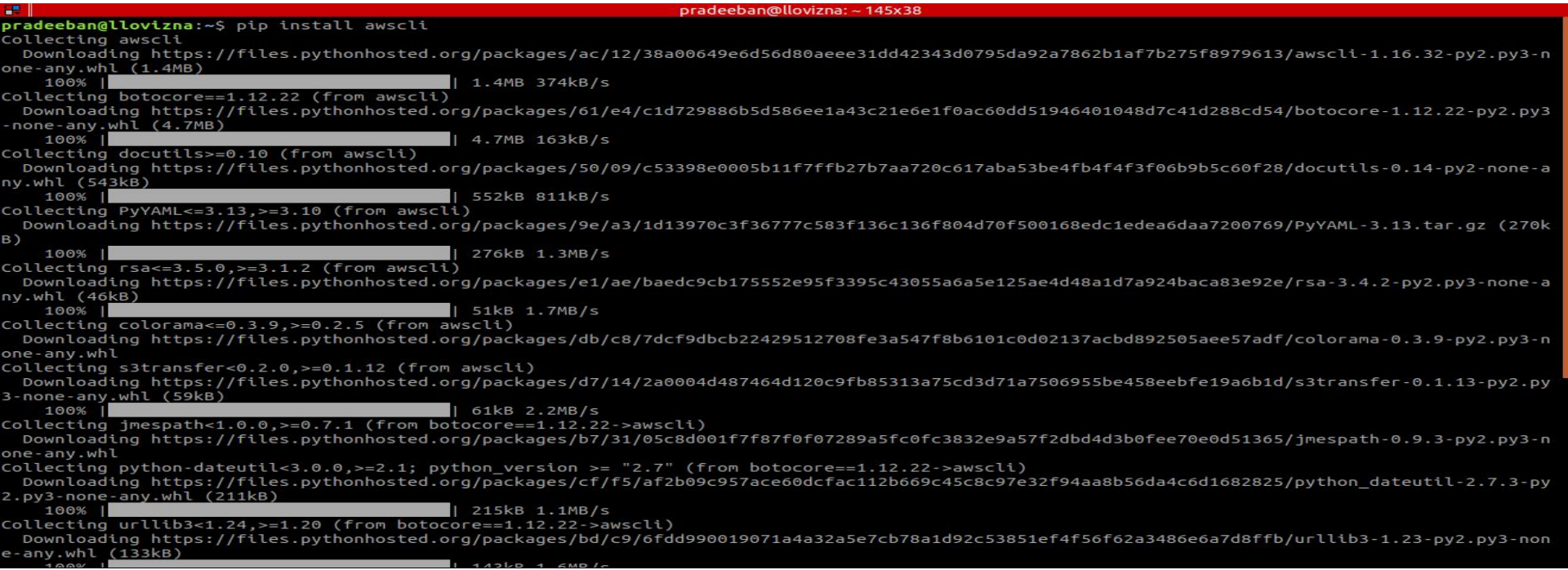
Costs aggregated by Amortizated unblended costs

Create Access key in your AWS Account



Install AWS CLI tools

$ pip install awscli



Configure AWS CLI

```
$ aws configure
```

AWS Access Key ID [None]: **********

AWS Secret Access Key [None]: **********

Default region name [None]: us-east-1

Default output format [None]: json

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html

Install PyWren

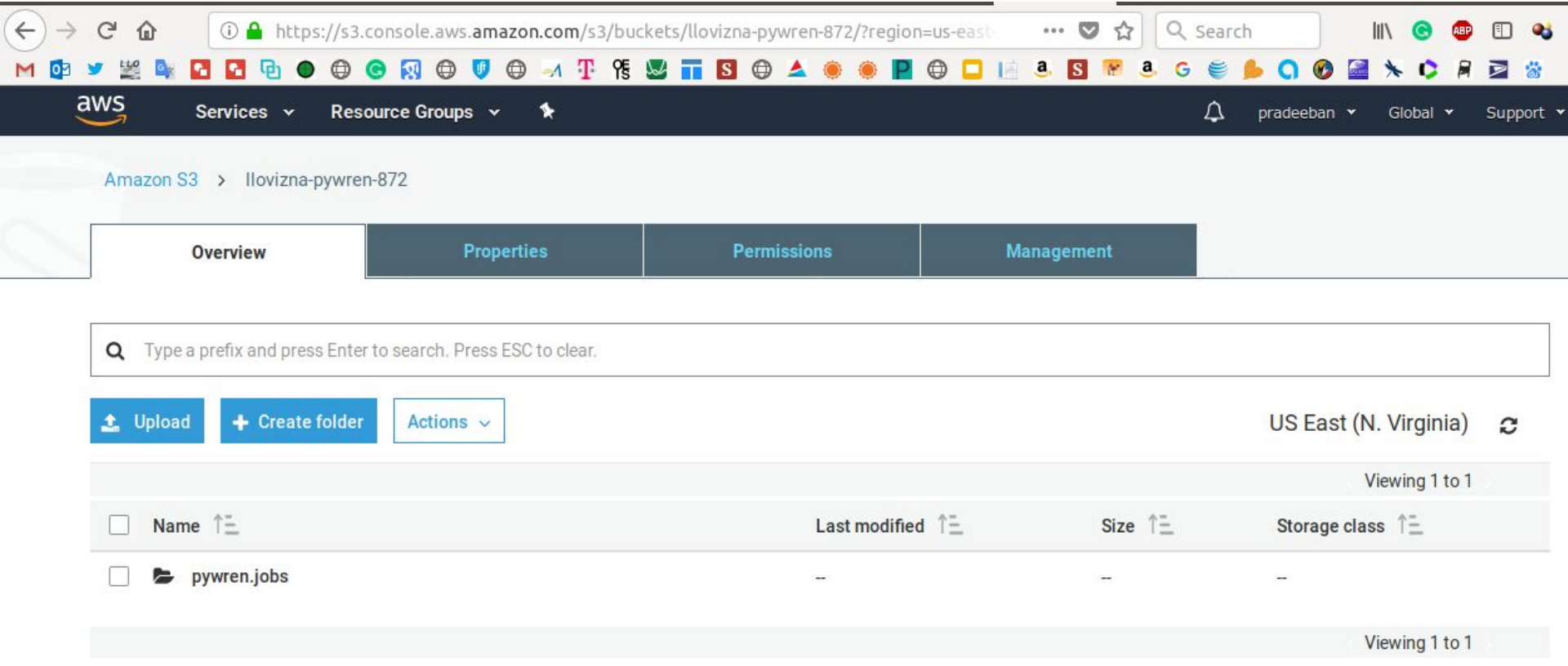
```
$ pip install pywren
```

$ pywren-setup

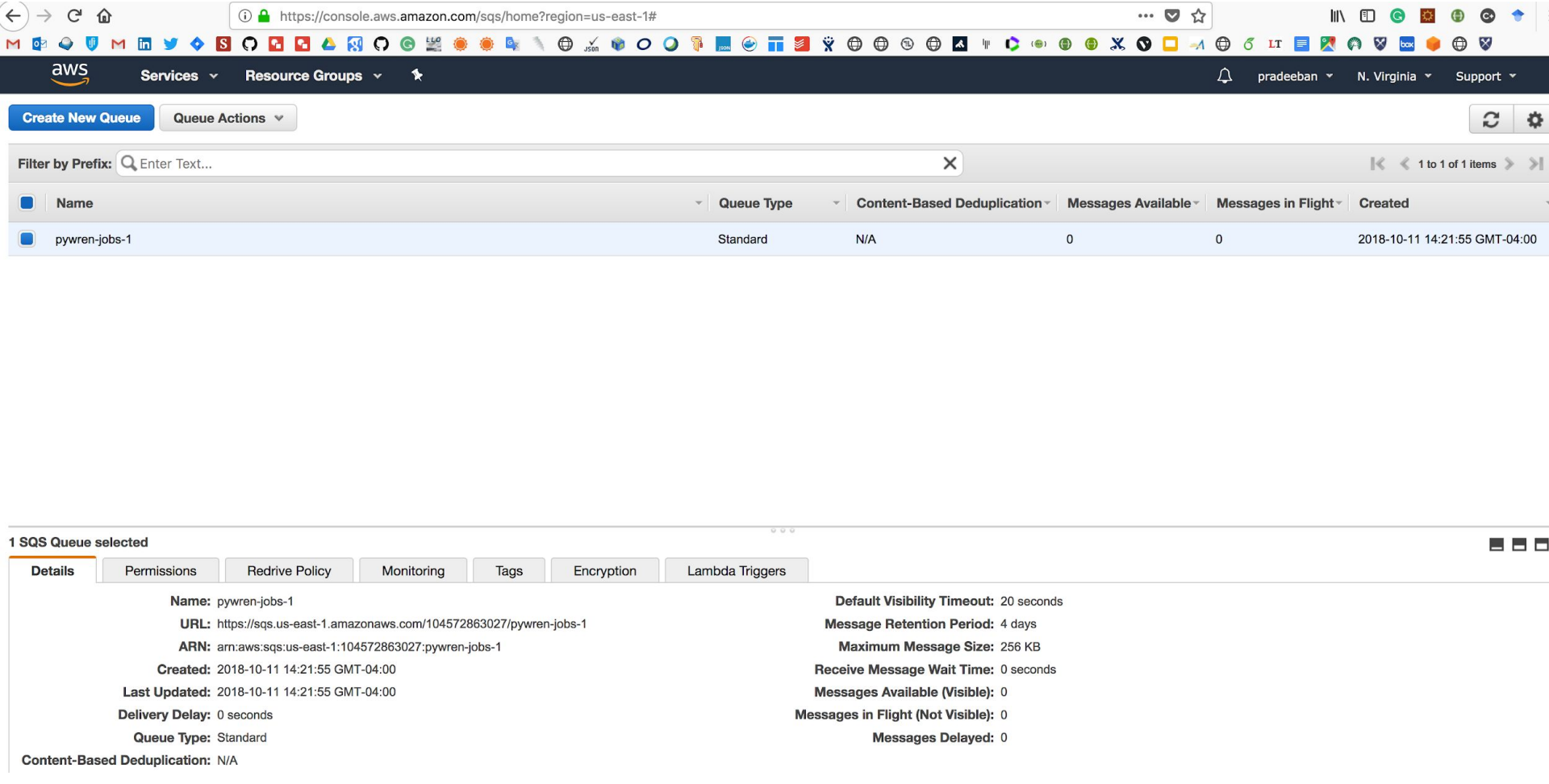
```
What is your default aws region? [us-west-2]: us-east-1
Location for config file:  [/Users/llovizna/.pywren_config]:
PyWren requires an s3 bucket to store intermediate data. What s3 bucket would you like to use?
[llovizna-pywren-872]:
Bucket does not currently exist, would you like to create it? [Y/n]: Y
PyWren prefixes every object it puts in S3 with a particular prefix.
PyWren s3 prefix:  [pywren.jobs]:
Would you like to configure advanced PyWren properties? [y/N]: y
Each lambda function runs as a particularIAM role. What is the name of the role youwould like
created for your lambda [pywren_exec_role_1]:
Each lambda function has a particular function name.What is your function name? [pywren_1]:
PyWren standalone mode uses dedicated AWS instances to run PyWren tasks. This is more flexible, but
more expensive with fewer simultaneous workers.
Would you like to enable PyWren standalone mode? [y/N]: y
Creating config /Users/llovizna/.pywren_config
new default file created in /Users/llovizna/.pywren_config
lambda role is pywren_exec_role_1
Creating bucket llovizna-pywren-872.
Creating role.
…..
```

The S3 Bucket

- PyWren uses this as the intermediate or default storage for its jobs.



AWS Simple Queue Service (SQS)



Enable log level to INFO

```
$ export PYWREN_LOGLEVEL=INFO
```

Confirm PyWren is running

```
$ pywren --help
```

Usage: pywren [OPTIONS] COMMAND [ARGS]...

Options:

--filename TEXT

--help Show this message and exit.

Run Some Examples!

1) Hello World

```
$ pywren test-function
```

2018-10-11 16:27:29,198 [INFO] pywren.executor: using serializer with meta-supplied preinstalls
2018-10-11 16:27:29,961 [INFO] pywren.executor: map 36ba6bca-19eb-4fa0-aa3d-afccf9ffd2ca 00000 apply async
2018-10-11 16:27:29,962 [INFO] pywren.executor: call_async 36ba6bca-19eb-4fa0-aa3d-afccf9ffd2ca 00000 lambda invoke
2018-10-11 16:27:30,149 [INFO] pywren.executor: call_async 36ba6bca-19eb-4fa0-aa3d-afccf9ffd2ca 00000 lambda invoke complete
2018-10-11 16:27:30,229 [INFO] pywren.executor: map invoked 36ba6bca-19eb-4fa0-aa3d-afccf9ffd2ca 00000 pool join
2018-10-11 16:27:46,427 [INFO] pywren.future: ResponseFuture.result() 36ba6bca-19eb-4fa0-aa3d-afccf9ffd2ca 00000 call_success True
function returned: Hello world

2) Simple Adder to an Array

https://github.com/sharmaashish/bmi500/blob/master/Lec7/maptest.py

(Add 7 to an array of [1,2,3,4] asynchronously via Lambda calls)

```
$ python maptest.py
```

8

{'result': 8, 'success': True, 'sys.path': ['/var/task', '/var/task', '/tmp/pymodules', '/tmp/condaruntime/lib/python36.zip', '/tmp/condaruntime/lib/python3.6', '/tmp/condaruntime/lib/python3.6/lib-dynload', '/tmp/condaruntime/lib/python3.6/site-packages', '/tmp/condaruntime/lib/python3.6/site-packages/setuptools-27.2.0-py3.6.egg']}

[8, 9, 10, 11]

```
if __name__ == "__main__":
        def test_add(x):
        return x + 7
        wrenexec = pywren.default_executor()
        x = [1, 2, 3, 4]
        futures = wrenexec.map(test_add, x, invoke_pool_threads=2)
        fs_dones, fs_notdones = pywren.wait(futures)
        result_count = len(fs_dones)
        f = futures[0]
        print (f.result(throw_except=False))
        print (f._call_invoker_result)
        print ([f.result() for f in futures])
```

3) Benchmark FLOPS
(floating point operations per second)

3.1) Get the Pywren examples.

```
$ git clone git@github.com:pywren/examples.git
```

3.2) Install dependencies

1. Pandas Python Data Analysis Library

```
$ pip install pandas
```

3.3) Run the code

```
$ cd examples/benchmark_flops/
$ python flops_benchmark.py
```

```
$ python flops_benchmark.py
```

invocation done, dur= 1.6978318691253662
callset id: 5d030bfb-cdf3-4c7e-9290-3077f27fda09

total time 21.706118106842045

5.936069187764401 GFLOPS

Default number of workers is 10.

Let’s try with more workers!

```
$ python flops_benchmark.py --workers 100
```

invocation done, dur= 2.439581871032715

callset id: fb53e966-d675-4251-a111-dca482cf4a95

total time 32.246346950531006

39.95771027262926 GFLOPS

```
$ python flops_benchmark.py --workers 300
```

invocation done, dur= 3.6147191524505615

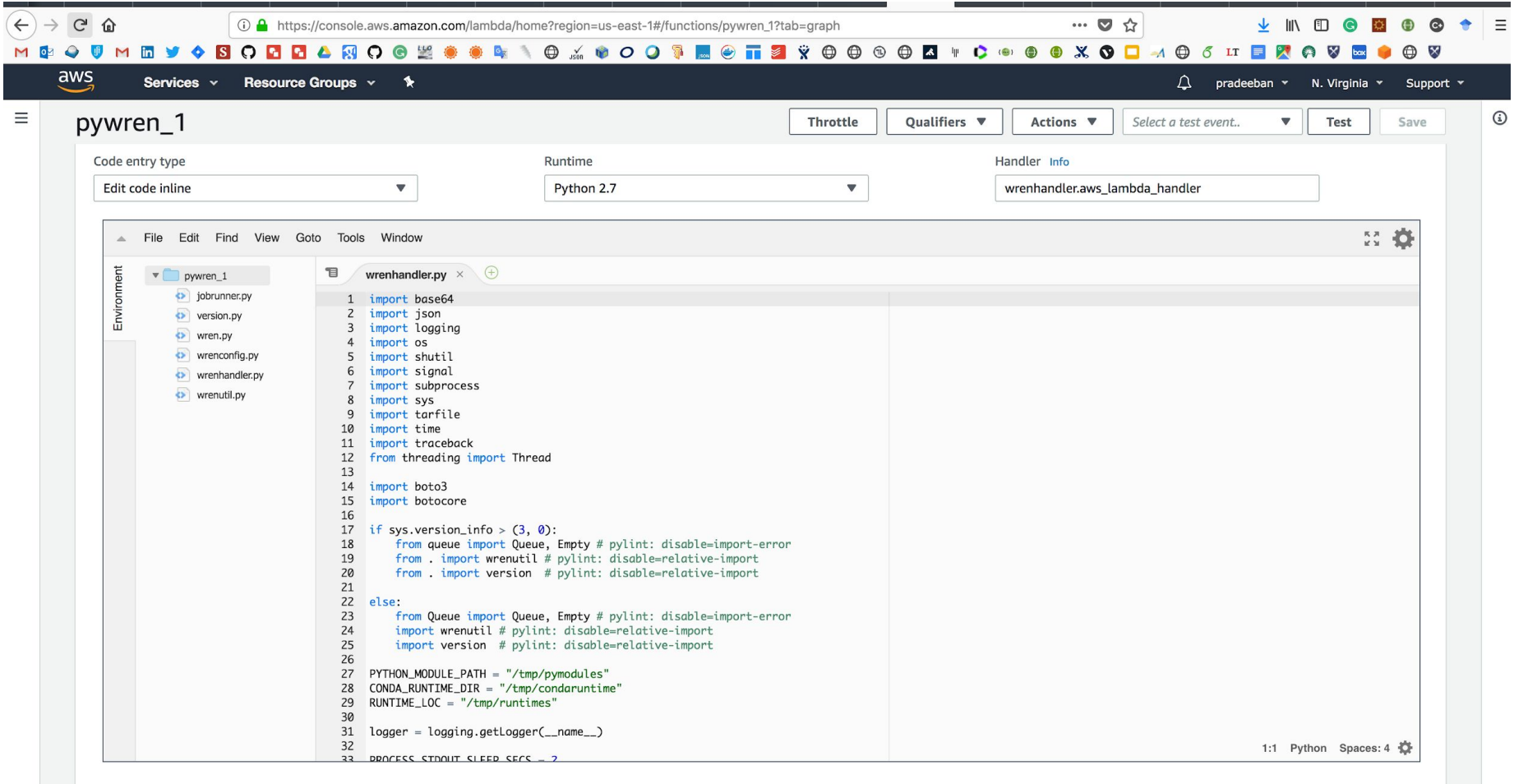
callset id: d72a86a2-98fb-40c2-8a1a-cee31dc0b17a

total time 16.89086604118347

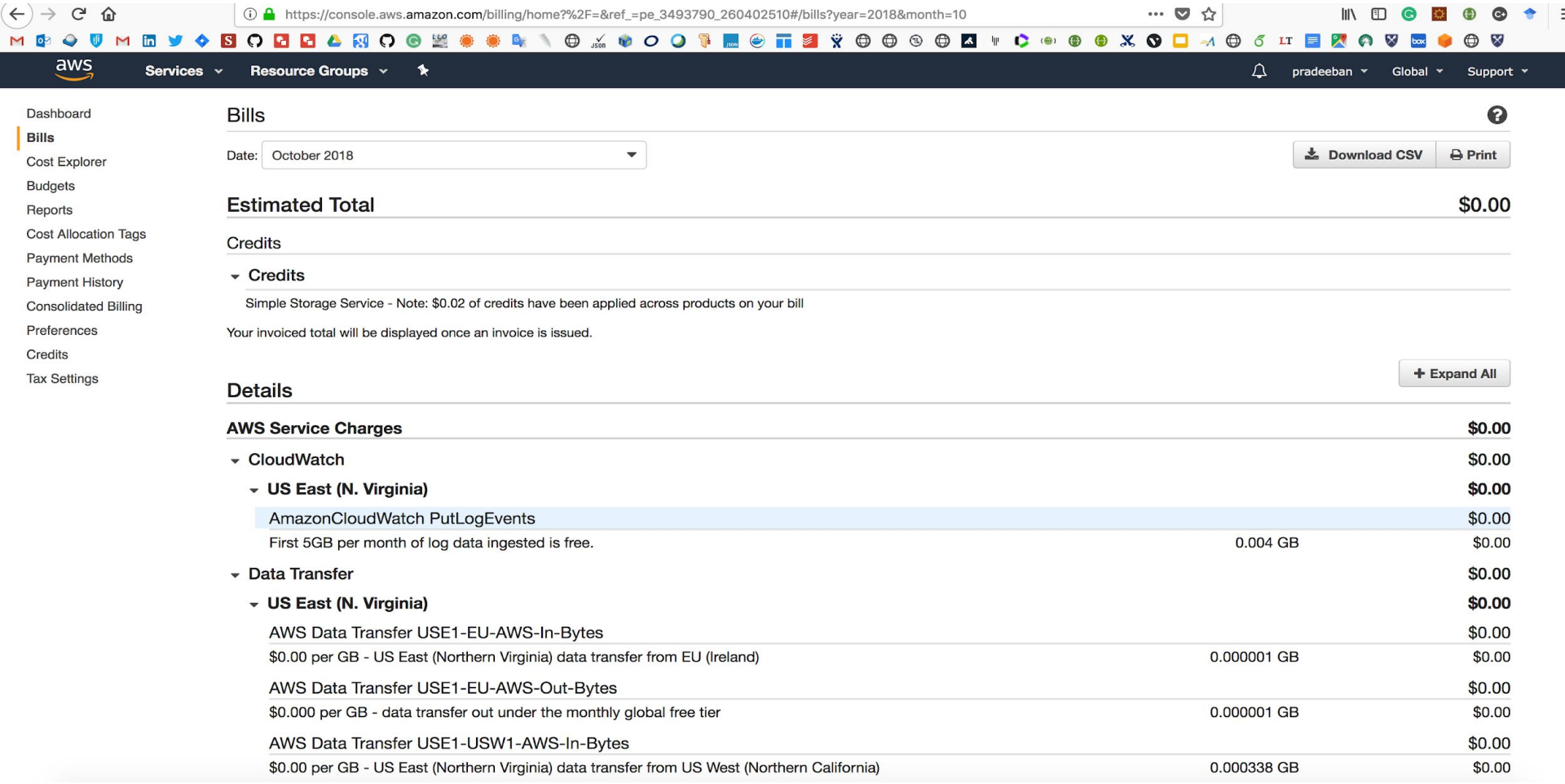
228.84975565937071 GFLOPS

Some points

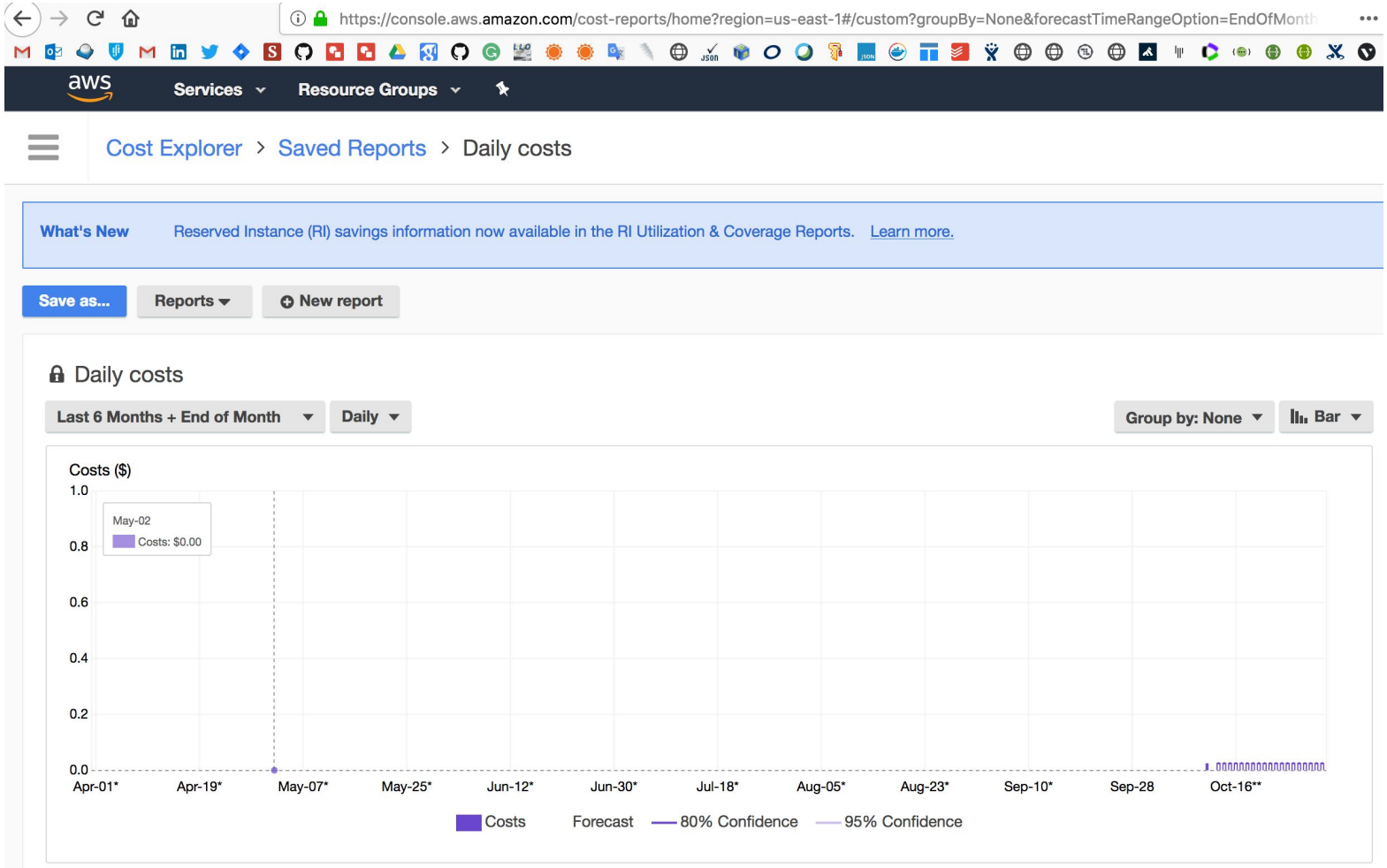
Find the code deployed by PyWren



Keep Track of the Cost



Cost Explorer / Cost Reports



Terminate your AWS resources once you are done

- AWS Lambda Functions and Resources
- S3 Storage Buckets
- SQS Queues