

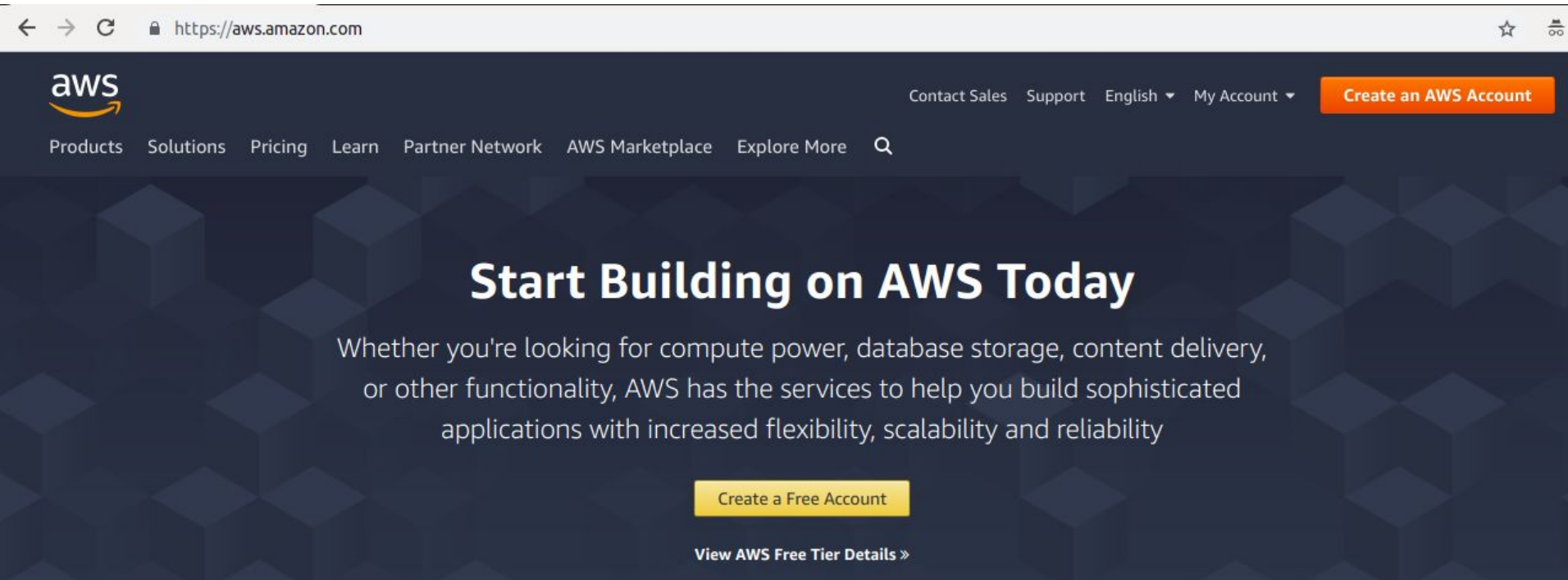
# PyWren

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.



A screenshot of the AWS website homepage. The browser address bar shows 'https://aws.amazon.com'. The AWS logo is in the top left. The top navigation bar includes links for 'Contact Sales', 'Support', 'English', and 'My Account', along with a 'Create an AWS Account' button. Below this is a secondary navigation bar with links for 'Products', 'Solutions', 'Pricing', 'Learn', 'Partner Network', 'AWS Marketplace', 'Explore More', and a search icon. The main content area has a dark blue background with a geometric pattern of cubes. It features the heading 'Start Building on AWS Today' in large white text, followed by a paragraph: 'Whether you're looking for compute power, database storage, content delivery, or other functionality, AWS has the services to help you build sophisticated applications with increased flexibility, scalability and reliability'. At the bottom, there is a yellow 'Create a Free Account' button and a link to 'View AWS Free Tier Details »'.

aws

Products Solutions Pricing Learn Partner Network AWS Marketplace Explore More

Contact Sales Support English My Account Create an AWS Account

## Start Building on AWS Today

Whether you're looking for compute power, database storage, content delivery, or other functionality, AWS has the services to help you build sophisticated applications with increased flexibility, scalability and reliability

Create a Free Account

View AWS Free Tier Details »

# Check your credits

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

## Credits

Please enter your code below to redeem your credits.

Promo Code

Security Check ⓘ



[Refresh Image](#)

Please type the characters as shown above

By clicking "Redeem" you indicate that you have read and agree to the terms of the AWS Promotional Credit Terms & Conditions located [here](#).

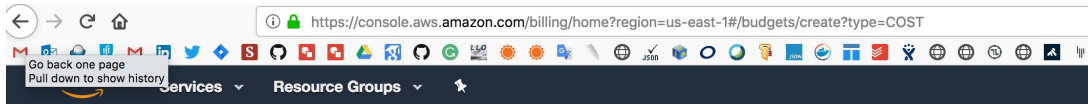
Redeem

The table below displays all AWS credits redeemed by your account. Credits are automatically applied to charges associated with qualifying AWS service usage. Please note that the values for used and remaining credit amounts are updated each month when your invoice is finalized.

Expiration Date	Credit Name	Amount Used	Amount Remaining	Applicable Products
2020-02-29	EDU_ENG_FY2018_IC-Q3_3_100USD	\$0.00	\$100.00	<a href="#">See complete list</a>

Total Credit Amount Remaining (as of 2019-12-01): \$100.00

# Create budgets



Dashboard

Bills

Cost Explorer

**Budgets**

Reports

Cost Allocation Tags

Payment Methods

Payment History

Consolidated Billing

Preferences

Credits

Tax Settings

## Create budget

Set budgets that automatically alert you when your costs, usage, RI utilization, and RI coverage exceed the thresholds you define.

### 1 Budget details

Budget Type

☒ Cost

☐ Usage

☐ RI Utilization ⓘ

☐ RI Coverage

Name\*

Monthly budget

Period

Monthly ▾

Start date

10/01/18 ▾

End date

03/31/20 ▾

Budgeted Amount\*

\$100.00

1 Name your budget, specify be active.

2 Refine your budget criteria

3 Send notifications via email

### 2 Refine your budget

Include costs related to

☒ Service

✕ AWS Lambda

✕ Amazon Simple Queue Service

✕ Amazon Simple Storage Service

✕ AmazonCloudWatch

☐ Linked Account

- 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 type	Budget name	Current	Forecasted	Budgeted	Current vs. budgeted	Forecasted vs. budgeted
<div><div></div><div></div></div>	<div>▼</div>	Cost	Monthly budget	\$0.01	\$0.50	\$100.00	0% <div></div> 1% <div></div>	

### 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 Amortized unblended costs**

Resource Groups ▾



pradeeban

Global v

Support ▾

## Dashboard

## Groups

## Users

## Roles

## Policies

## Identity providers

## Account settings

## Credential report

## Encryption keys

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access M

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#) in AWS General

▲ Password

- Multi-factor authentication (MFA)

- ▼ Access keys (access key ID and secret access key)

You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the [signing documentation](#). For your protection, store your access keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90 days.

Note: You can have a maximum of two access keys (active or inactive) at a time.

Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Oct 11th 2018		AKIAJWXH5AKCNZQXC2GQ	N/A	N/A	N/A	Active	<a href="#">Make Inactive</a>   <a href="#">Delete</a>
Oct 11th 2018		AKIAJ72U52MPJKWR5DDQ	2018-10-11 17:36 EDT	us-east-1	s3	Active	<a href="#">Make Inactive</a>   <a href="#">Delete</a>

### Create New Access Key



## Important Change - Managing Your AWS Secret Access Keys

As described in a [previous announcement](#), you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a



# Install AWS CLI tools

\$ pip install awscli

```
pradeeban@llovizna: ~ 145x38
pradeeban@llovizna:~$ pip install awscli
Collecting awscli
  Downloading https://files.pythonhosted.org/packages/ac/12/38a00649e6d56d80aeee31dd42343d0795da92a7862b1af7b275f8979613/awscli-1.16.32-py2.py3-none-any.whl (1.4MB)
    100% |#####| 1.4MB 374kB/s
Collecting botocore==1.12.22 (from awscli)
  Downloading https://files.pythonhosted.org/packages/61/e4/c1d729886b5d586ee1a43c21e6e1f0ac60dd51946401048d7c41d288cd54/botocore-1.12.22-py2.py3-none-any.whl (4.7MB)
    100% |#####| 4.7MB 163kB/s
Collecting docutils>=0.10 (from awscli)
  Downloading https://files.pythonhosted.org/packages/50/09/c53398e0005b11f7fffb27b7aa720c617aba53be4fb4f4f3f06b9b5c60f28/docutils-0.14-py2-none-any.whl (543kB)
    100% |#####| 552kB 811kB/s
Collecting PyYAML<=3.13,>=3.10 (from awscli)
  Downloading https://files.pythonhosted.org/packages/9e/a3/1d13970c3f36777c583f136c136f804d70f500168edc1edea6daa7200769/PyYAML-3.13.tar.gz (270kB)
    100% |#####| 276kB 1.3MB/s
Collecting rsa<=3.5.0,>=3.1.2 (from awscli)
  Downloading https://files.pythonhosted.org/packages/e1/ae/baedc9cb175552e95f3395c43055a6a5e125ae4d48a1d7a924baca83e92e/rsa-3.4.2-py2.py3-none-any.whl (46kB)
    100% |#####| 51kB 1.7MB/s
Collecting colorama<=0.3.9,>=0.2.5 (from awscli)
  Downloading https://files.pythonhosted.org/packages/db/c8/7dcf9dbcb22429512708fe3a547f8b6101c0d02137acbd892505aee57adf/colorama-0.3.9-py2.py3-none-any.whl
  Downloading https://files.pythonhosted.org/packages/d7/14/2a0004d487464d120c9fb85313a75cd3d71a7506955be458eebfe19a6b1d/s3transfer-0.1.13-py2.py3-none-any.whl (59kB)
    100% |#####| 61kB 2.2MB/s
Collecting jmespath<1.0.0,>=0.7.1 (from botocore==1.12.22->awscli)
  Downloading https://files.pythonhosted.org/packages/b7/31/05c8d001f7f87f0f07289a5f0fc3832e9a57f2dbd4d3b0fee70e0d51365/jmespath-0.9.3-py2.py3-none-any.whl
Collecting python-dateutil<3.0.0,>=2.1; python_version >= "2.7" (from botocore==1.12.22->awscli)
  Downloading https://files.pythonhosted.org/packages/cf/f5/af2b09c957ace60dcfac112b669c45c8c97e32f94aa8b56da4c6d1682825/python_dateutil-2.7.3-py2.py3-none-any.whl (211kB)
    100% |#####| 215kB 1.1MB/s
Collecting urllib3<1.24,>=1.20 (from botocore==1.12.22->awscli)
  Downloading https://files.pythonhosted.org/packages/bd/c9/6fdd990019071a4a32a5e7cb78a1d92c53851ef4f56f62a3486e6a7d8fffb/urllib3-1.23-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.6MB/s
```

# 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 particular IAM role. What is the name of the role you would 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.

The screenshot shows the AWS S3 console interface. The browser address bar displays the URL: `https://s3.console.aws.amazon.com/s3/buckets/llovizna-pywren-872/?region=us-east-1`. The console header includes the AWS logo, navigation menus for 'Services' and 'Resource Groups', and user information for 'pradeeban' in the 'Global' region. The breadcrumb trail indicates the current location: 'Amazon S3 > llovizna-pywren-872'. Below the breadcrumb, there are four tabs: 'Overview', 'Properties' (which is selected), 'Permissions', and 'Management'. A search bar is present with the placeholder text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar, there are three buttons: 'Upload', 'Create folder', and 'Actions'. On the right side, the region is set to 'US East (N. Virginia)'. The main content area displays a table with the following columns: 'Name', 'Last modified', 'Size', and 'Storage class'. The table contains one entry: a folder named 'pywren.jobs'. At the bottom right, it says 'Viewing 1 to 1'.

Name	Last modified	Size	Storage class
pywren.jobs	--	--	--

# AWS Simple Queue Service (SQS)

Browser address bar: <https://console.aws.amazon.com/sqs/home?region=us-east-1#>

Navigation: **aws** | **Services** | **Resource Groups**

Buttons: **Create New Queue** | **Queue Actions**

Filter by Prefix:

Name	Queue Type	Content-Based Deduplication	Messages Available	Messages in Flight	Created
pywren-jobs-1	Standard	N/A	0	0	2018-10-11 14:21:55 GMT-04:00

1 SQS Queue selected

Details

Permissions

Redrive Policy

Monitoring

Tags

Encryption

Lambda Triggers

**Name:** pywren-jobs-1

**URL:** <https://sqs.us-east-1.amazonaws.com/104572863027/pywren-jobs-1>

**ARN:** arn:aws:sqs:us-east-1:104572863027:pywren-jobs-1

**Created:** 2018-10-11 14:21:55 GMT-04:00

**Last Updated:** 2018-10-11 14:21:55 GMT-04:00

**Delivery Delay:** 0 seconds

**Queue Type:** Standard

**Content-Based Deduplication:** N/A

**Default Visibility Timeout:** 20 seconds

**Message Retention Period:** 4 days

**Maximum Message Size:** 256 KB

**Receive Message Wait Time:** 0 seconds

**Messages Available (Visible):** 0

**Messages in Flight (Not Visible):** 0

**Messages Delayed:** 0

# 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 21.70611810684204
```

```
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 16.890866604118347

228.84975565937071 GFLOPS



Some points

# Find the code deployed by PyWren

The screenshot shows the AWS Lambda console interface for a function named `pywren_1`. The browser address bar displays the URL `https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions/pywren_1?tab=graph`. The console header includes the AWS logo, navigation tabs for Services and Resource Groups, and user information for 'pradeeban' in the 'N. Virginia' region. The function configuration section at the top shows 'Code entry type' set to 'Edit code inline', 'Runtime' set to 'Python 2.7', and 'Handler' set to 'wrenhandler.aws\_lambda\_handler'. Below this, a code editor displays the source code for `wrenhandler.py`. The code includes imports for `base64`, `json`, `logging`, `os`, `shutil`, `signal`, `subprocess`, `sys`, `tarfile`, `time`, `traceback`, and `Thread` from the `threading` module. It also imports `boto3` and `botocore`. A conditional block checks `sys.version_info > (3, 0)` to import `Queue` and `Empty` from the `queue` module, or imports them from `collections` for Python 2.7. The code sets environment variables for `PYTHON_MODULE_PATH`, `CONDA_RUNTIME_DIR`, and `RUNTIME_LOC`, and initializes a logger. The bottom right corner of the editor shows '1:1 Python Spaces: 4'.

Code entry type: Edit code inline

Runtime: Python 2.7

Handler: `wrenhandler.aws_lambda_handler`

```
1 import base64
2 import json
3 import logging
4 import os
5 import shutil
6 import signal
7 import subprocess
8 import sys
9 import tarfile
10 import time
11 import traceback
12 from threading import Thread
13
14 import boto3
15 import botocore
16
17 if sys.version_info > (3, 0):
18     from queue import Queue, Empty # pylint: disable=import-error
19     from . import wrenutil # pylint: disable=relative-import
20     from . import version # pylint: disable=relative-import
21 else:
22     from Queue import Queue, Empty # pylint: disable=import-error
23     import wrenutil # pylint: disable=relative-import
24     import version # pylint: disable=relative-import
25
26 PYTHON_MODULE_PATH = "/tmp/pymodules"
27 CONDA_RUNTIME_DIR = "/tmp/condaruntime"
28 RUNTIME_LOC = "/tmp/runtimes"
29
30 logger = logging.getLogger(__name__)
31
32 PROCESS_STDOUT_SLEEP_SECS = 2
```

[illegible]

## Tax Settings

## Download CSV Print

\$0.00

## ▼ Credits

Simple Storage Service - Note: \$0.02 of credits have been applied across products on your bill

Your invoiced total will be displayed once an invoice is issued.

## + Expand All

**\$0.00**

\$0.00

**\$0.00**

## AmazonCloudWatch PutLogEvents

First 5GB per month of log data ingested is free.

0.004 GB

\$0.00

\$0.00

**\$0.00**

### AWS Data Transfer USE1-EU-AWS-In-Bytes

\$0.00 per GB - US East (Northern Virginia) data transfer from EU (Ireland)

0.000001 GB

\$0.00

### AWS Data Transfer USE1-EU-AWS-Out-Bytes

\$0.000 per GB - data transfer out under the monthly global free tier

0.000001 GB

\$0.00

### AWS Data Transfer USE1-USW1-AWS-In-Bytes

\$0.00 per GB - US East (Northern Virginia) data transfer from US West (Northern California)

0.000338 GB

\$0.00

\$0.00

—

—

—

Reserved Instance (RI) savings information now available in the RI Utilization & Coverage Reports. [Learn more.](#)

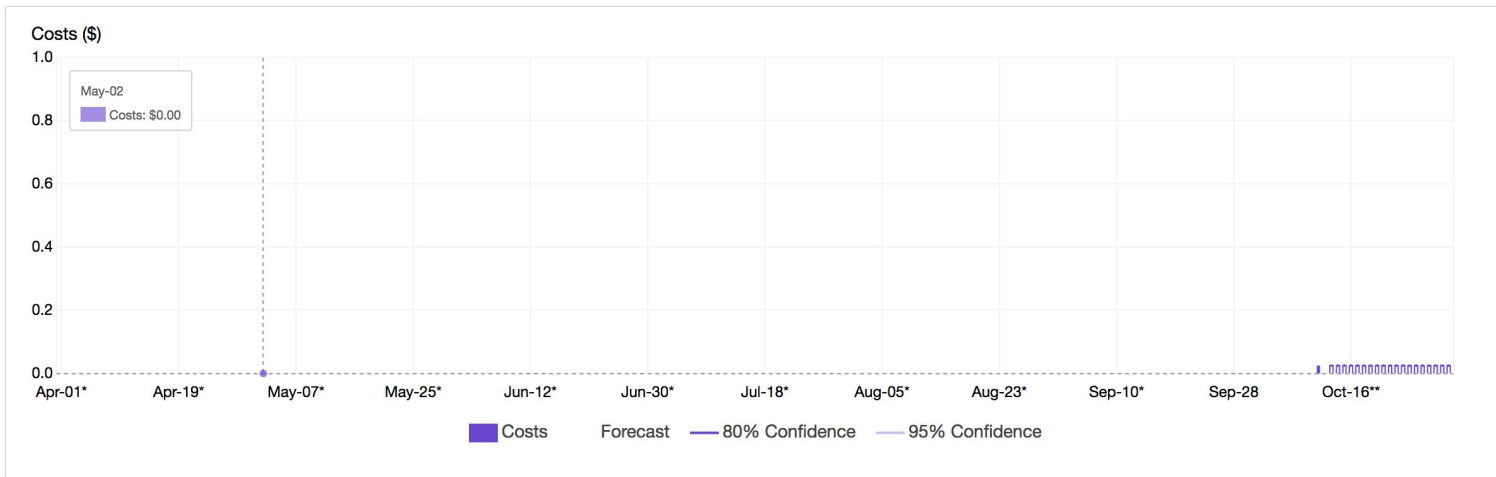
Reports ▼

**+ New report**

### Daily costs

Daily ▼

Bar ▼



# Terminate your *AWS* resources once you are done

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