
Introduction

This is a tool that tries to discover all AWS resources created in an account. AWS has many products (a.k.a. services) with new ones constantly being added and existing ones expanded with new features. The ecosystem allows users to piece together many different services to form a customized cloud experience. The ability to instantly spin up services at scale comes with a manageability cost. It can quickly become difficult to audit an AWS account for the resources being used. It is not only important for billing purposes, but also for security. Dormant resources and unknown resources are more prone to security configuration weaknesses. Additionally, resources with unexpected dependencies pose availability, access control, and authorization issues.

It uses `botocore` to discover AWS services and what regions they run in. It is also used in invoking the service APIs. The APIs that are invoked are those which should list or describe resources. The results can be printed to `stdout` in JSON format. They can also be written across several files:

- Raw responses from API endpoints can be written to a file specified on the commandline. The file format is Python pickle.
- Exceptions raised during tool execution can be written to a file specified on the commandline. The file format is Python pickle.
- `gui/aws_inventory_data-<environment_name>.json` - JSON format. Parsed responses structured for input to the GUI.

Installation

First, install Python2.7.

There is a small GUI for displaying progress which uses the standard Python *Tkinter* module. However, the underlying native library code for Tcl/Tk may need extra steps to install. Then,

```
pip install -r requirements.txt
```

Windows

Use the Python installer to install Tkinter/Tcl/Tk.

Linux

Use your OS package manager:

Ubuntu / Debian

```
sudo apt-get install python-tk
```

Usage

You can run the script without any parameters. It will search for your AWS creds in your shell environment, instance metadata, config file, then credentials file. You can also provide a CSV file, containing your creds, on the commandline. You will want a user that has permissions like the AWS managed policy `ViewOnlyAccess`. If you are feeling lucky, you could just pipe the output of the tool to a JSON parser like `jq`.

The tool could take a long time (dozens of minutes) to complete if no restrictions are placed on which operations to invoke for each service across each region. Filtering by service and region can be done on the commandline while filtering by service operation can be done via configuration file. A pre-configured file was created and checked into the repository. It will be used by default.

Aside from the commandline output, you can view the results locally in a React single-page app. No web server needed. Just open the HTML file in a browser and select the generated JSON file when prompted.

The app uses `jsTree` to display the data in a hierarchical, tree-like structure. There is also a search feature.

NOTE: When invoking APIs, those that raise an exception are not used again regardless of region. Known causes of exceptions are:

- required API parameter not specified in service model (or the tool is not properly reading model?)
- insufficient authorization for the selected credentials
- network error

Examples

- Run with defaults.

```
$ python aws_inventory.py
```

- List AWS services known to *botocore*. This is all done locally by reading service model files.

```
1 $ python aws_inventory.py --list-svcs
2 acm
3 apigateway
4 application-autoscaling
5 appstream
6 autoscaling
7 batch
8 budgets
9 clouddirectory
10 cloudformation
11 cloudfront
12 .
13 .
14 .
```

- List service operations known to *botocore*. This is all done locally by reading service model files.

```
1 $ python aws_inventory.py --list-operations
2 [shield]
3 DescribeSubscription
4 ListAttacks
5 ListProtections
6
7 [datapipeline]
8 ListPipelines
9
10 [firehose]
11 ListDeliveryStreams
12 .
13 .
14 .
15 [glacier]
16 # NONE
17
18 [stepfunctions]
19 ListActivities
20 ListStateMachines
21
22 Total operations to invoke: 4045
```

- Print what APIs would be called for a service. This is all done locally.

```
$ python aws_inventory.py --debug --dry-run
```

Screenshots

