

# **Project Documentation**

**BeatDrift-CLI Automation Tool Implementation**

**COSC595**

**Trimester 1 2022**

Jason O. Aboh

Paldeep Kaur

Sumit Khokhar

## **Table of Contents**

**1.0 Project Description**   

**1.1 Client**   

**1.2 Project Team**

**2.0 Stakeholders**

**3.0 Installation Manual**

## Project Description

The project is oriented towards creating a CLI tool which will be used to build a pipeline to address drift in an Infrastructure as Code (IaC) environment.

### Client

Client: Mark Wallis, Director of Hunter Orbit Pty Ltd.

Email: [mark@hunterorbit.com.au](mailto:mark@hunterorbit.com.au)

### 1.2. Project Team

The team members are:

Name	Email id (s)
Jason Aboh	<a href="mailto:jaboh@myune.edu.au">jaboh@myune.edu.au</a> , <a href="mailto:bigbossw107@gmail.com">bigbossw107@gmail.com</a>
Paldeep Kaur	<a href="mailto:pkaur21@myune.edu.au">pkaur21@myune.edu.au</a> , <a href="mailto:paldeepkaur1995@gmail.com">paldeepkaur1995@gmail.com</a>
Sumit Khokhar	<a href="mailto:skhokha2@myune.edu.au">skhokha2@myune.edu.au</a> , <a href="mailto:sumitkhokhar359@gmail.com">sumitkhokhar359@gmail.com</a>

## **2.0. Stakeholders**

The stakeholders of the project are as follows:

Client: **Mark Wallis**

Unit Coordinator (COSC 595): **Dr. Edmund Sadgrove**

Team Member (Project Lead): **Jason O. Aboh**

Team Member: **Sumit Khokhar**

Team Member: **Paldeep Kaur**

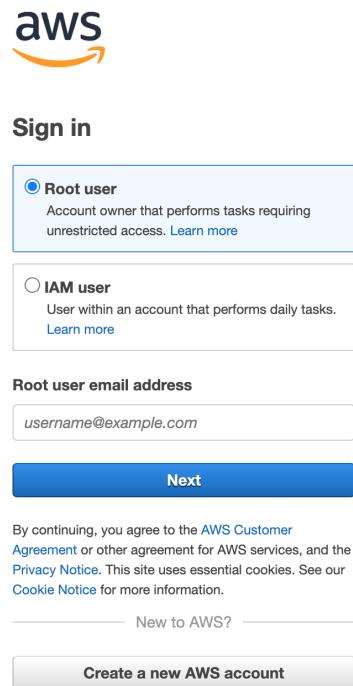
## **3.0. Installation Manual and User manual:**

### **3.1 Creating Resources in AWS for testing (crawling)**

After creation of an aws account and set up of credentials, to access the given aws; navigate to the aws Iam console website with the designated region at the beginning of the url link, the below link takes us to the “us-east-1” region:

<https://us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/home>

The user shall arrive at the page below; which they will be required to enter the credentials for the AWS account they have access to:



The image shows the AWS Sign In page. At the top is the AWS logo. Below it is the title "Sign in". There are two radio button options: "Root user" (selected) and "IAM user". The "Root user" option is described as an "Account owner that performs tasks requiring unrestricted access". The "IAM user" option is described as a "User within an account that performs daily tasks". Below these options is a field for "Root user email address" containing "username@example.com". A large blue "Next" button is centered below the email field. Below the "Next" button is a small text block about AWS Customer Agreement and Privacy Notice, followed by a "Cookie Notice". At the bottom of the page are two links: "New to AWS?" and "Create a new AWS account".

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The user will need to input credentials for a “Root” user account (performs tasks with unrestricted access), or as a “IAM” user account ( a user within an account that performs daily tasks ).

If the user supplies an alias (account name) or account ID number rather than the email address of a “Root” user account, they are taken to the IAM user page to sign in using an alias or account ID as shown below:



## Sign in as IAM user

Account ID (12 digits) or account alias

637333041330

IAM user name

jasona

Password

.....

Remember this account

Sign in

[Sign in using root user email](#)

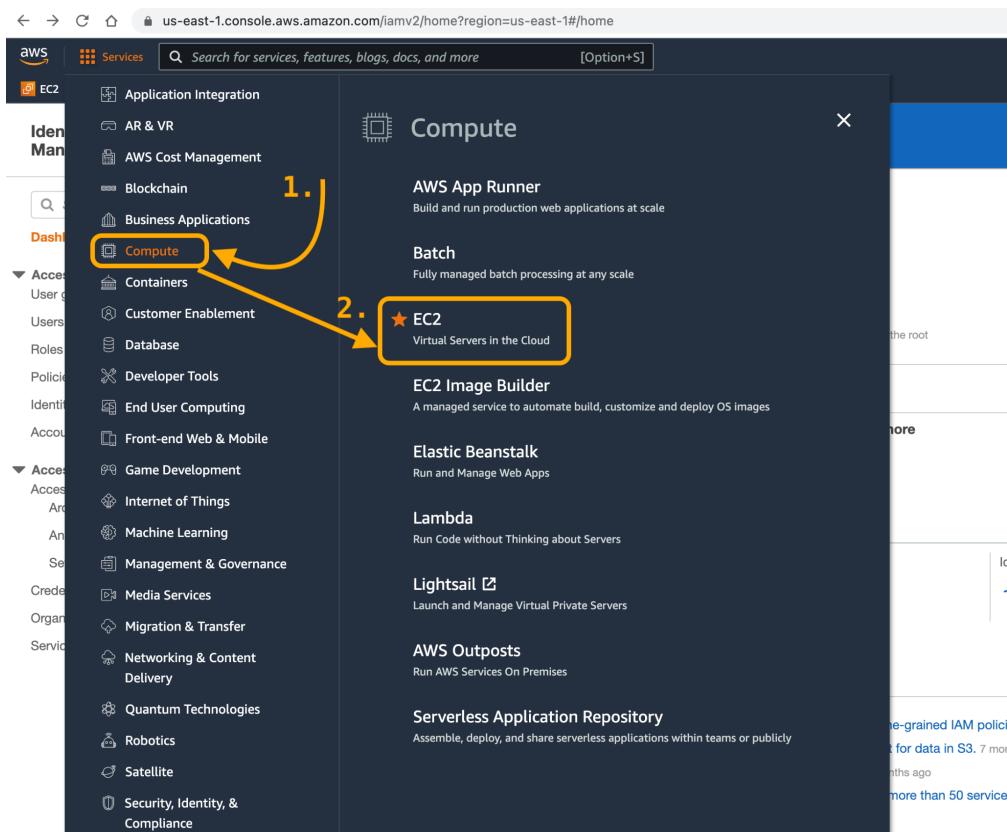
[Forgot password?](#)

On successful sign in, the user arrives at the IAM dashboard as shown below:

The screenshot shows the AWS IAM dashboard. At the top, there is a navigation bar with the AWS logo, a search bar, and various service icons. The IAM logo is highlighted with a blue box. The main content area has a blue header bar with the text "Introducing the new IAM dashboard experience" and a message about the redesign. Below this, the "IAM dashboard" section is visible, featuring "Security recommendations" with two items: "Add MFA for root user" and "Add MFA for yourself". A green checkmark indicates that "Your user, jasona, does not have any active access keys that have been unused for more than a year". The "AWS Account" section shows the account ID (637333041330) and a "Create" button. The "Quick Links" section includes links for "My security credentials", "Policy simulator", and "Tools". The "IAM resources" section displays statistics: 0 User groups, 3 Users, 10 Roles, 1 Policies, and 1 Identity providers. The "What's new" section lists recent updates, including "Right-size permissions for more roles in your account using IAM Access Analyzer to generate 50 fine-grained IAM policies per day." and "Amazon S3 Object Ownership can now disable access control lists to simplify access management for data in S3.".

Please note that the IAM access management settings are located to the left within the blue rectangle, access to the AWS services is located to the top left within the yellow rectangle, access to IAM Users, Roles, Policies and Identity providers are located within the orange rectangle in the middle of the page; then finally, the current account username and ID , along with the region it is signed into, is located at the top right of the screen within the light green rectangle.

AWS resources can be created by either navigating to the services tab at the top right and clicking on the service sections within the scrollable options on the left, or searching for the service of interest to create:



Or by searching for “EC2” in the search bar located at the top left of the page:

The screenshot shows the AWS IAM service search results for 'EC2'. The search bar at the top contains 'EC2'. Below the search bar, there is a sidebar with 'Identity and Access Management (IAM)' and a 'Services' section. The 'Services' section lists 'Features 40', 'Blogs 1,751', 'Documentation 127,314', 'Knowledge Articles 30', and 'Tutorials 18'. The main content area shows a search result for 'EC2' with a star icon, titled 'Virtual Servers in the Cloud'. It includes a 'Top features' section with links to 'Dashboard', 'Launch templates', 'Instances', 'Spot Instance requests', and 'Savings plans'. A 'See all 9 results' link is also present.

After selecting the “EC2” service (which is used to create virtual servers in the cloud); the user is then able to create an EC2 instance by clicking on the “Launch instance” button as shown below:

The screenshot shows the AWS EC2 service 'Resources' page. The top navigation bar includes 'Services', a search bar, and links for 'DynamoDB', 'Console Home', 'CloudFront', 'RDS', 'Resource Groups & Tag Editor', and 'CloudFormation'. On the left, a sidebar shows 'New EC2 Experience' (with a 'Tell us what you think' link), 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances' (with 'Instances New', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances New', 'Dedicated Hosts', 'Scheduled Instances', and 'Capacity Reservations' listed), and 'Images' (with 'AMIs New' and 'AMI Catalog' listed). The main content area is titled 'Resources' and shows a summary of EC2 resources: 4 Instances (running), 0 Dedicated Hosts, 0 Elastic IPs, 4 Instances, 1 Key pairs, 0 Load balancers, 0 Placement groups, 2 Security groups, 5 Volumes, and 0 Snapshots. A callout box with an arrow points to the 'Launch instance' button in the 'Launch instance' section, which is highlighted with an orange box. The 'Launch instance' section also includes a 'Migrate a server' button. To the right, there is a 'Service health' section showing 'Region: US East (N. Virginia)' and 'Status: This service is operating normally'.

After clicking on the “Launch Instance” button, the user is directed to the form page below to configure the resource they intend to create, for EC2 instances the below is the form page the user must use to configure the instance:

At the top of the form, the user is able to assign a Name /Tag to the instance resource; A resource tag is a label that the user assigns to an AWS resource. Each tag consists of a key and an optional value, both of which the user defines - Tags allow the user to locate resources in the account via the use of the Tag Editor (which is used to list out the available tagged resources within a region - if specified).

The user is also able to select an application and Operating system image, and on the right of the screen, the user can input the number of instances they intend to create:

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstances:

**Launch an instance** Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** Info

Name  Add additional tags

**Application and OS Images (Amazon Machine Image)** Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux Ubuntu Windows Red Hat SUSE Linux > Search Browse more AMIs

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-0022f774911c1d690 (64-bit (x86)) / ami-0e449176ecc3e577 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

**Summary**

Number of instances Info

Software Image (AMI)  
Amazon Linux 2 Kernel 5.10 AMI... read more  
ami-0022f774911c1d690

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance

The user has the ability to configure instance types and key pairs:

us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstances:

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-0022f774911c1d690 (64-bit (x86)) / ami-0e449176ecc3e577 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Description  
Amazon Linux 2 Kernel 5.10 AMI 2.0.20220426.0 x86\_64 HVM gp2

Architecture  AMI ID

**Instance type** Info

t2.micro Free tier eligible Compare instance types

Family: t2 1 vCPU 1 GiB Memory  
On-Demand Linux pricing: 0.0116 USD per Hour  
On-Demand Windows pricing: 0.0162 USD per Hour

**Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required  Create new key pair

**Network settings** Edit

**Summary**

Number of instances Info

Software Image (AMI)  
Amazon Linux 2 Kernel 5.10 AMI... read more  
ami-0022f774911c1d690

Virtual server type (instance type)  
t2.micro

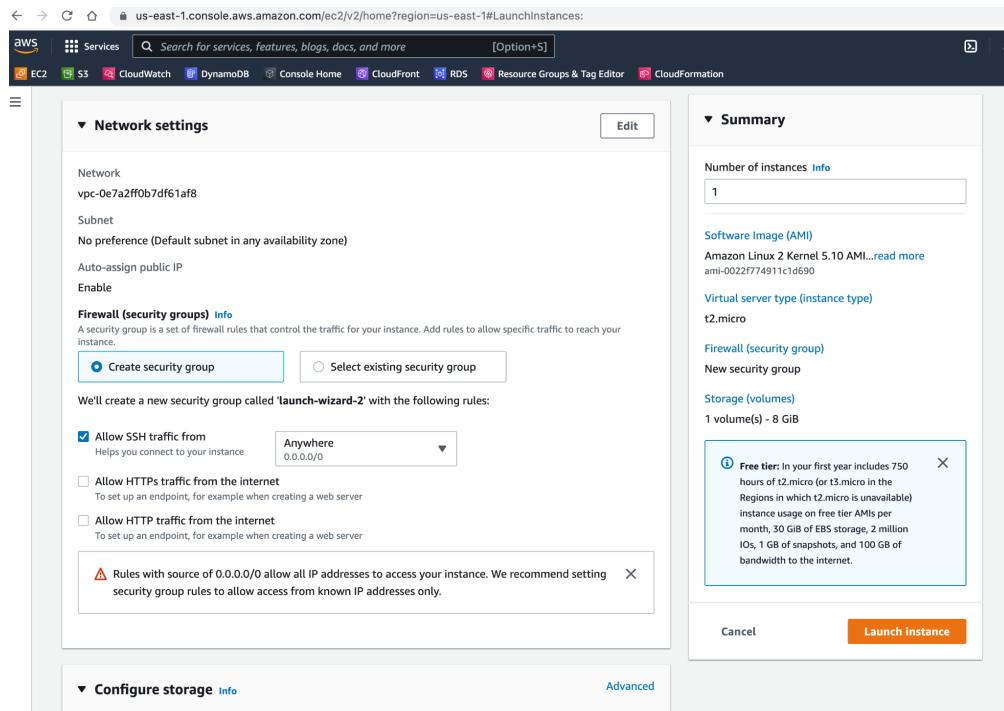
Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance

And also can configure Network settings for the EC2 instance being created:



The screenshot shows the AWS EC2 Launch Wizard interface. On the left, the 'Network settings' section is open, showing a VPC (vpc-0e7a2ff0b7df61af8) and a subnet (No preference (Default subnet in any availability zone)). It includes an 'Auto-assign public IP' checkbox and an 'Enable' checkbox. Below these are 'Firewall (security groups)' and 'Create security group' buttons. A note says: 'We'll create a new security group called "launch-wizard-2" with the following rules:'. Under 'Allow SSH traffic from' (checked), the 'Anywhere' dropdown shows '0.0.0.0/0'. Other options like 'Allow HTTPS traffic from the internet' and 'Allow HTTP traffic from the internet' are shown with their descriptions. A warning message states: '⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' On the right, the 'Summary' section shows 'Number of instances' (1), 'Software Image (AMI)' (Amazon Linux 2 Kernel 5.10 AMI...), 'Virtual server type (instance type)' (t2.micro), and 'Storage (volumes)' (1 volume(s) - 8 GiB). A 'Free tier' callout box provides details: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom are 'Cancel' and 'Launch instance' buttons.

And set other configurations such as the below:

▼ **Configure storage** [Info](#)

1x  GiB  [▼](#) Root volume

**Info** Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X](#)

[Add new volume](#)

0 x File systems [Edit](#)

▼ **Instance type** [Info](#)

Instance type

<b>t2.micro</b>	Free tier eligible	<a href="#">Compare instance types</a>
Family: t2	1 vCPU	1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour		
On-Demand Windows pricing: 0.0162 USD per Hour		

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

[▼](#) [Create new key pair](#)

## ▼ Advanced details [Info](#)

### Purchasing option [Info](#)

Request Spot Instances

Request Spot Instances at the Spot price, capped at the On-Demand price

### IAM instance profile [Info](#)

Select

 [Create new IAM profile](#)  


### Hostname type [Info](#)

IP name

### DNS Hostname [Info](#)

- Enable IP name IPV4 (A record) DNS requests
- Enable resource-based IPV4 (A record) DNS requests
- Enable resource-based IPV6 (AAAA record) DNS requests

### Instance auto-recovery [Info](#)

Select

### Shutdown behavior [Info](#)

Select

### Stop - Hibernate behavior [Info](#)

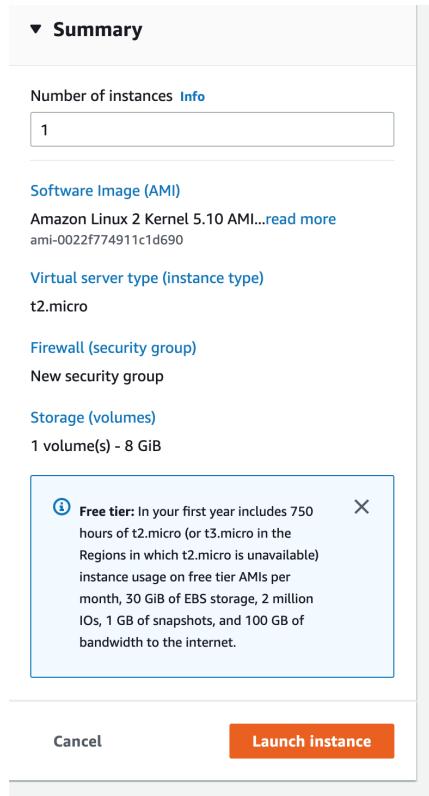
Select

### Termination protection [Info](#)

Select

### Stop protection [Info](#)

When done with the settings; the user can click the “Launch Instance” button to create the resource:



The new instance will then appear with its status in the list on the EC2 instance page, along with the other previously created instances:

Instances (4) <small>Info</small>								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	ExampleAppS...	i-0858f25e706de0459	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	+ us-east-1a	ec2-100-26-109-145.co...
<input type="checkbox"/>	example	i-0befaf04e1889a745	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	+ us-east-1a	ec2-34-201-244-77.co...
<input type="checkbox"/>	example1	i-0d70512caa9596gebe	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	+ us-east-1a	ec2-44-204-67-52.com...
<input type="checkbox"/>	My Web Server	i-064fb49804d805f6	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	+ us-east-1b	ec2-52-204-214-254.co...

The user can also navigate to the AWS Management console where there is a dedicated section with links to build some common AWS resources (including the estimated time it would take to build on average).

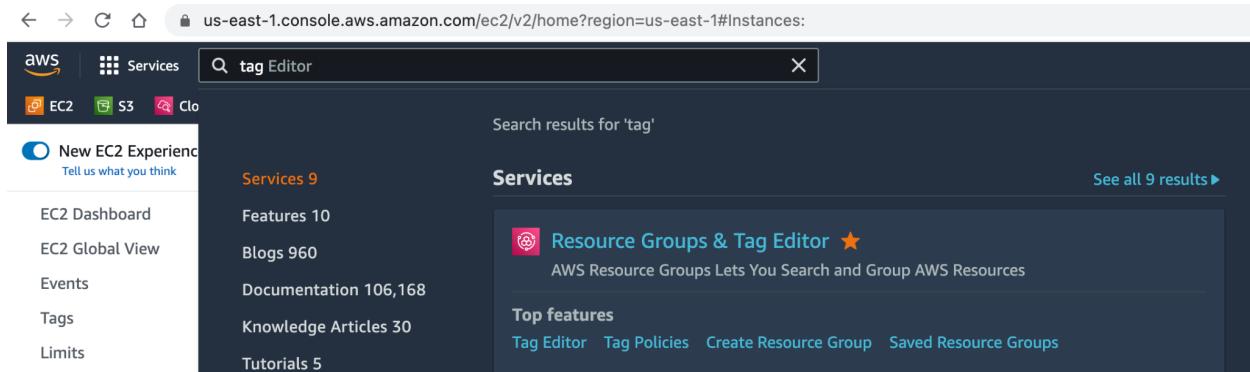
This is accomplished by simply searching for “Console home” at the top left search bar of the page

Then clicking on “Console Home” in the services results, to navigate to the “AWS Management Console”. On the “AWS Management Console” home page, the user can scroll down to utilize the provided links to begin the creation of some common AWS resources located at the bottom left of the screen in the orange rectangle highlight:

The user should note that the costs for the current month’s use of AWS resources and services are displayed at the right of the page within the green rectangle highlight.

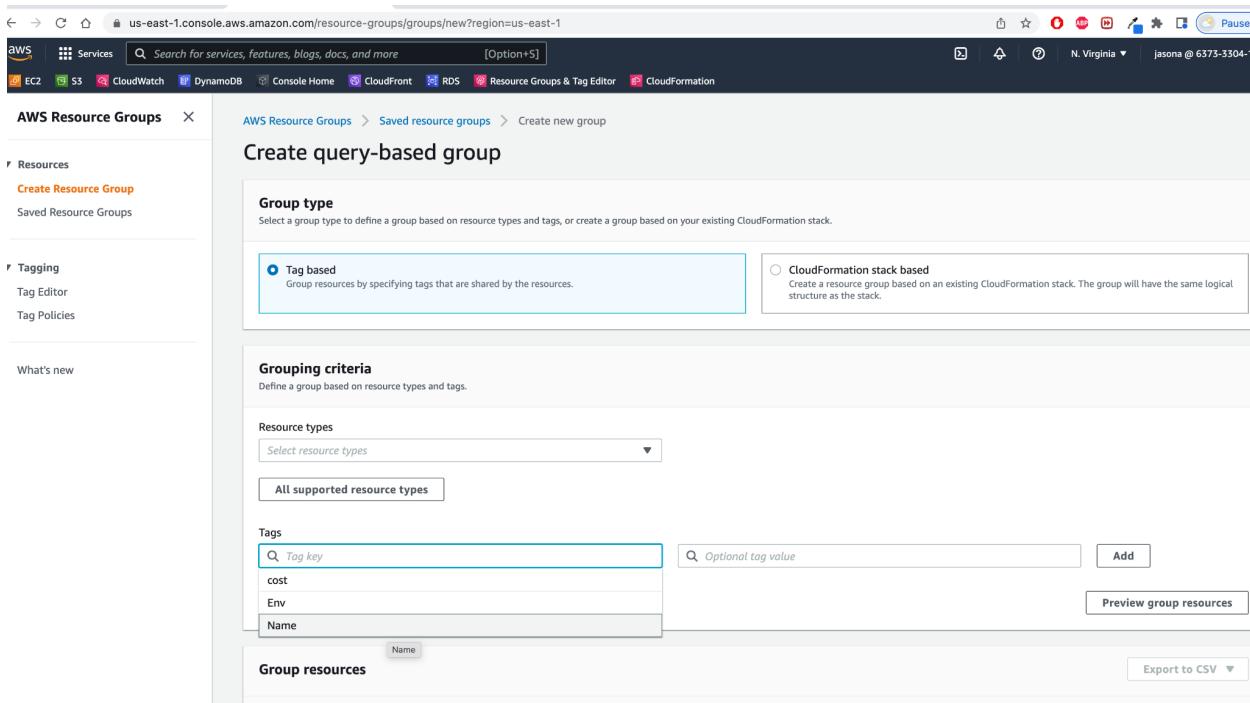
The user can utilise the tag manager to list out created resources which has been tagged and add it to a group of resources, which they can monitor:

Search for the tag manager in the top bar:



The screenshot shows the AWS EC2 console with a search bar at the top containing 'tag'. The search results are displayed under the heading 'Search results for 'tag''. There are two main sections: 'Services' and 'Top features'. The 'Services' section lists 'Resource Groups & Tag Editor' as a featured service with a star icon. Below it are links to 'AWS Resource Groups Lets You Search and Group AWS Resources', 'Top features', and 'Tag Editor', 'Tag Policies', 'Create Resource Group', and 'Saved Resource Groups'. The 'Top features' section also includes links to 'Tag Editor', 'Tag Policies', 'Create Resource Group', and 'Saved Resource Groups'. On the left, a sidebar shows links to 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', and 'Limits'.

When the user clicks into this option, they are able to create groups of tagged resources:



The screenshot shows the 'Create query-based group' page in the AWS Resource Groups section. The left sidebar shows 'Create Resource Group' selected. The main form is titled 'Create query-based group' and has two main sections: 'Group type' and 'Grouping criteria'. In the 'Group type' section, the 'Tag based' option is selected, with a sub-note: 'Group resources by specifying tags that are shared by the resources.' In the 'Grouping criteria' section, there is a 'Resource types' dropdown set to 'Select resource types' and a 'Tags' section. The 'Tags' section contains a table with columns 'Tag key' and 'Optional tag value'. The table has three rows: 'cost' (with value 'cost'), 'Env' (with value 'Env'), and 'Name' (with value 'Name'). There are 'Add' and 'Preview group resources' buttons at the bottom of the table. A 'Group resources' button is located at the bottom left, and an 'Export to CSV' button is at the bottom right.

After filling in the fields and saving the group, the users will now have a saved group list of tagged resources:

us-east-1.console.aws.amazon.com/resource-groups/group/All\_aws\_resources?region=us-east-1

AWS Resource Groups Services Search for services, features, blogs, docs, and more [Option+S]

EC2 S3 CloudWatch DynamoDB Console Home CloudFront RDS Resource Groups & Tag Editor CloudFormation

AWS Resource Groups All\_aws\_resources

All\_aws\_resources

**Group details**

Group name: All\_aws\_resources

Group description: all resources

Group ARN: arn:aws:resource-groups:us-east-1:637333041330:group/All\_aws\_resources

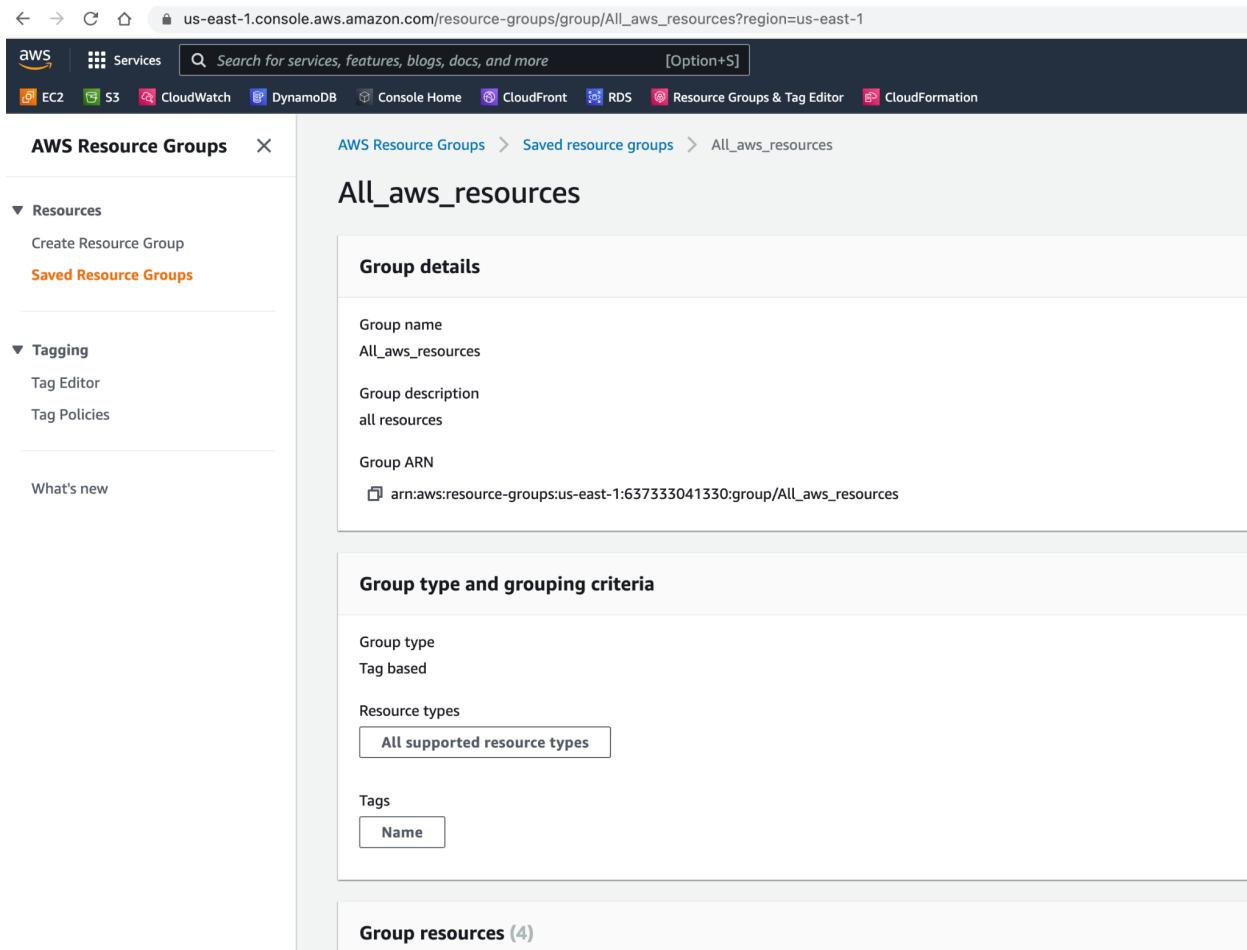
**Group type and grouping criteria**

Group type: Tag based

Resource types: All supported resource types

Tags: Name

**Group resources (4)**



us-east-1.console.aws.amazon.com/resource-groups/groups?region=us-east-1

AWS Resource Groups Services Search for services, features, blogs, docs, and more [Option+S]

EC2 S3 CloudWatch DynamoDB Console Home CloudFront RDS Resource Groups & Tag Editor CloudFormation

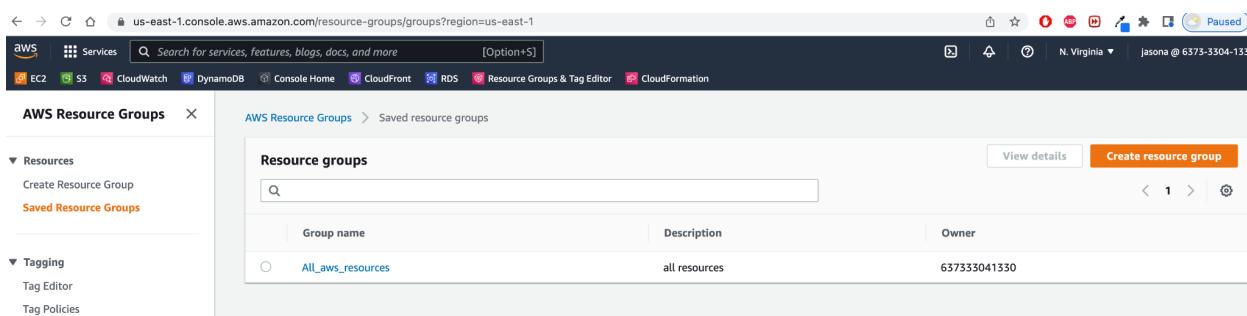
N. Virginia jasona @ 6373-3304-1330

AWS Resource Groups Saved resource groups

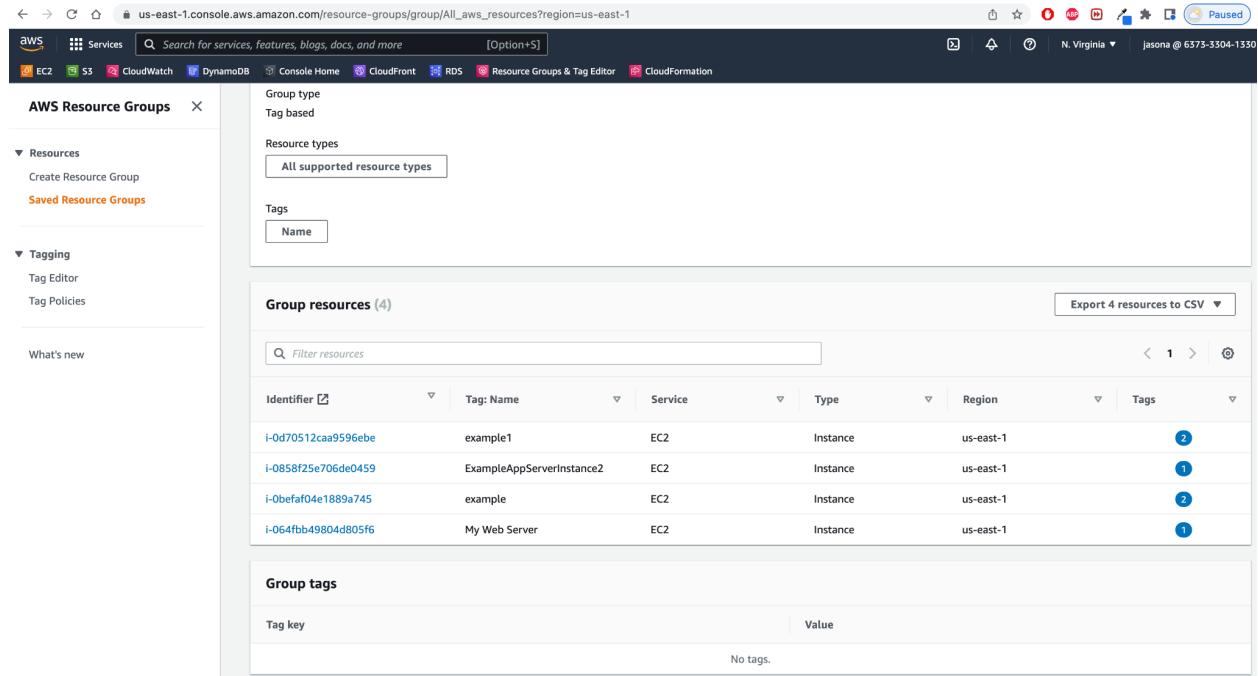
**Resource groups**

Group name	Description	Owner
All_aws_resources	all resources	637333041330

View details Create resource group



which they can click into and monitor on the aws web app while cross-checking with the aws cli crawler:



The screenshot shows the AWS Resource Groups console with the following interface elements:

- Left sidebar:** AWS logo, Services (selected), Search bar, [Option+S] button, EC2, S3, CloudWatch, DynamoDB, Console Home, CloudFront, RDS, Resource Groups & Tag Editor, CloudFormation.
- Top navigation:** Group type (Group type), Tag based (Tag based), Resource types (All supported resource types), Tags (Name).
- Main content:**
  - Group resources (4):** A table showing four EC2 instances. The table has columns: Identifier, Tag: Name, Service, Type, Region, and Tags.
  - Group tags:** A table showing no tags.
- Bottom right:** Export 4 resources to CSV button.

Identifier	Tag: Name	Service	Type	Region	Tags
i-0d70512caa9596ebe	example1	EC2	Instance	us-east-1	2
i-0858f25e706de0459	ExampleAppServerInstance2	EC2	Instance	us-east-1	1
i-0beaff04e1889a745	example	EC2	Instance	us-east-1	2
i-064fb49804db05f6	My Web Server	EC2	Instance	us-east-1	1

## 3.2 Configure AWS Environment in CLI

The configuration of the AWS environment in the Command Line Interface is quite simple. It involves the use of amazon's "aws cli", which needs to be installed on the user's machine and configured with the user's account details and credentials.

After the user accesses the directory of choice with

```
cd /directoryOfChoice
```

Input the following command and hit the return/enter button in the command line interface:

```
aws configure
```

You will be prompted and enter the corresponding credentials supplied from the AWS account you are using :

```
Macs-MBP-2:~ JasonAbeeowhage$ aws configure
AWS Access Key ID [*****XEW4]: ****N4
AWS Secret Access Key [*****s81a]: ****Us
Default region name [us-east-1]: us-east-1
Default output format [json]: json
Macs-MBP-2:~ JasonAbeeowhage$ aws configure set profile jasona
Macs-MBP-2:~ JasonAbeeowhage$ aws configure list
  Name            Value        Type    Location
  ----
  profile        <not set>    None    None
  access_key     *****XEW4  shared-credentials-file
  secret_key     *****s81a  shared-credentials-file
  region         us-east-1    config-file  ~/.aws/config
```

After entering the correct credentials for AWS Access Key ID, AWS Secret Access Key, Default region name, and output format; the user's aws cli environment is configured.

The user is also able to list out the aws accounts that have been configured in the environment by entering :

### **aws configure list**

As shown in the screenshot above.

## **3.3 Crawling AWS for resources with BeatDrift-CLI tool**

Crawling AWS for specific or all resources can be done with the use of the “BeatDrift-CLI” tool.

The user simply needs to download the source code zip file and then unzip this file, Then go into the “beatdrift-cli-master\_copy” directory from the home directory it was downloaded into by typing in:

```
cd ./beatdrift-cli-master_copy”
```

within the directory of their choice (This can also be a virtual environment). The environment in which the python script is run, needs to have the required installed system software environment tools and versions as specified in the ‘setup.py’ file:

**Programming language version:** Python 2’, Python 2.7’, Python 3’, or Python 3.6’

**Boto3 version:** 1.16.57

**App\_json\_file\_cache version:** 0.2.2

To install the latest version of a package enter:

```
pip install 'PackageName'
```

To install a specific version, type the package name followed by the required version enter:

```
pip install 'PackageName ==1.4'
```

The user should note that there are compatibility issues with the latest version of boto3, due to several changes made that would affect the ability to run BeatDrift-cli as intended; until updates are made, users are advised to stick to the requirements stated.

pip CLI python packaging - version (or equivalent python package manager)

Can be used to install the specific versions of these packages, if they are still available; the user can also search the web to install the specific versions of the required libraries, to ensure the script runs as intended.

Tip:

Care must be taken to ensure the user's machine is set to the current date and time to avoid denial of access to list the available resources, due to invalid AWS credentials authentication caused by a non- automatically set time and date on the machine/computer; an example of this authentication error is shown below:

Once the environment is set up correctly; cd into the “BeatDrift-CLI” folder via the command line to execute the python scripts;

Please enter :

```
chmod +x __main__.py
```

Then :

```
cp __main__.py ~/bin/beatdrift-cli-master
```

The user would then be able to run the program using commands such as:

```
beatdrift-cli query --region us-east-1 --service ec2 --directory ./listdataCheck2/
```

However, since full integration was not achieved in the project, if you are having problems with the execution or have prompts about missing modules, you can utilize the ‘aws-list-all’ package incorporated by BeatDrift-cli (with credit given in the notice to the author) to achieve the same results of crawling AWS resources and storing them as json files within the designated directory/folder within the project’s folder:

AWS resources can be crawled by installing and using the “aws-list-all” package

If you are using a virtual environment simply enter the below code:

```
mkvirtualenv -p $(which python3) aws
pip install aws-list-all
```

After installation, The user is able to query specific AWS resources by typing in the following.

```
aws-list-all query --region us-east-1 --service ec2 --directory ./listdataCheck2/demo
```

This command line argument crawls the provided aws account for all resources contained within the specified parameters:

(1) The main command is “query”

(2) Parameter 1 - region : us-east-1: this parameter

(3) Parameter 2 - service : ec2

- This parameter tells the script to only crawl ec2 resources.

(4) Parameter 3 - Directory: ./listdataCheck2/demo

- This parameter tells the script where to store the json files created to store the information about the AWS resources it has detected and listed.

More parameters like “-verbose” can be added and repeated for emphasis on expected outcome.

```

Mac-MBP-2:beatdrift-cli-master_copy JasonAbeowhage$ aws-list-all query --region us-east-1 --service ec2 --directory ./listdataCheck2/demo
Increasing the open connection limit "nofile" from 256 to 6000.
Building set of queries to execute...
...done. Executing queries...
--- ec2 us-east-1 DescribeAddresses None Addresses
--- ec2 us-east-1 DescribeAddressesAttribute None Addresses
--- ec2 us-east-1 DescribeByIpCidrs None ByIpCidrs
--- ec2 us-east-1 DescribeCapacityReservationFleets None CapacityReservationFleets
--- ec2 us-east-1 DescribeCapacityReservations None CapacityReservations
--- ec2 us-east-1 DescribeCarrierGateways None CarrierGateways
--- ec2 us-east-1 DescribeClassicLinkInstances None Instances
--- ec2 us-east-1 DescribeClientVpnEndpoints None ClientVpnEndpoints
--- ec2 us-east-1 DescribeCoipPools None CoipPools
--- ec2 us-east-1 DescribeConversionTasks None ConversionTasks
--- ec2 us-east-1 DescribeCustomerGateways None CustomerGateways
--- ec2 us-east-1 DescribeEgressOnlyInternetGateways None EgressOnlyInternetGateways
--- ec2 us-east-1 DescribeElasticGpus None ElasticGpuSet
--- ec2 us-east-1 DescribeExportImageTasks None ExportImageTasks
--- ec2 us-east-1 DescribeExportTasks None ExportTasks
--- ec2 us-east-1 DescribeFastLaunchImages None FastLaunchImages
--- ec2 us-east-1 DescribeFastSnapshotRestores None FastSnapshotRestores
--- ec2 us-east-1 DescribeFleets None Fleets
--- ec2 us-east-1 DescribeFlowLogs None FlowLogs
--- ec2 us-east-1 DescribeFpgaImages None FpgaImages
--- ec2 us-east-1 DescribeHostReservations None HostReservationSet
--- ec2 us-east-1 DescribeHosts None Hosts
--- ec2 us-east-1 DescribeIamInstanceProfileAssociations None IamInstanceProfileAssociations
--- ec2 us-east-1 DescribeImages None Images
--- ec2 us-east-1 DescribeImportImageTasks None ImportImageTasks
--- ec2 us-east-1 DescribeImportSnapshotTasks None ImportSnapshotTasks
--- ec2 us-east-1 DescribeInstanceCreditSpecifications None InstanceCreditSpecifications
--- ec2 us-east-1 DescribeInstanceEventWindows None InstanceEventWindows
--- ec2 us-east-1 DescribeInternetGateways None InternetGateways
--- ec2 us-east-1 DescribeIpamPools None IpamPools
--- ec2 us-east-1 DescribeIpamScopes None IpamScopes
--- ec2 us-east-1 DescribeIpams None Ipams
--- ec2 us-east-1 DescribeIpv6Pools None Ipv6Pools
--- ec2 us-east-1 DescribeLaunchTemplates None LaunchTemplates
--- ec2 us-east-1 DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations None LocalGatewayRouteTableVirtualInterfaceGroupAssociations
--- ec2 us-east-1 DescribeLocalGatewayRouteTableVpcAssociations None LocalGatewayRouteTableVpcAssociations
--- ec2 us-east-1 DescribeLocalGatewayRouteTables None LocalGatewayRouteTables
--- ec2 us-east-1 DescribeLocalGatewayVirtualInterfaceGroups None LocalGatewayVirtualInterfaceGroups
--- ec2 us-east-1 DescribeLocalGatewayVirtualInterfaces None LocalGatewayVirtualInterfaces
--- ec2 us-east-1 DescribeLocalGateways None LocalGateways
--- ec2 us-east-1 DescribeMovingAddresses None MovingAddressStatuses
--- ec2 us-east-1 DescribeNatGateways None NatGateways
--- ec2 us-east-1 DescribeNetworkAcls None NetworkAcls
--- ec2 us-east-1 DescribeNetworkInsightsAccessScopeAnalyses None NetworkInsightsAccessScopeAnalyses
--- ec2 us-east-1 DescribeNetworkInsightsAccessScopes None NetworkInsightsAccessScopes
--- ec2 us-east-1 DescribeNetworkInsightsAnalyses None NetworkInsightsAnalyses
--- ec2 us-east-1 DescribeNetworkInsightsPaths None NetworkInsightsPaths
--- ec2 us-east-1 DescribeNetworkInterfacePermissions None NetworkInterfacePermissions
--- ec2 us-east-1 DescribePlacementGroups None PlacementGroups
--- ec2 us-east-1 DescribePublicIpv4Pools None PublicIpv4Pools
--- ec2 us-east-1 DescribeReservedInstances None ReservedInstances
--- ec2 us-east-1 DescribeReservedInstancesListings None ClientError("An error occurred (OptInRequired) when calling the DescribeReservedInstancesListings operation: AccountId '63733041330' , You are not authorized to use the requested product. Please complete the seller registration null.")
--- ec2 us-east-1 DescribeReservedInstancesModifications None ReservedInstancesModifications

```

After installation, The user is able to query the following.

aws-list-all query --region us-east-1 --service

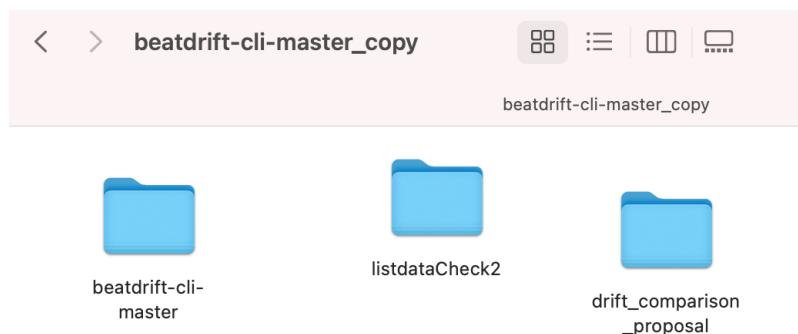
## 4.0 Terraform

Terraform is an open source infrastructure as code tool developed by Hashicorp. Terraform is a coding tool that helps you provision and manage infrastructure for an application. Terraform allows you to define your infrastructure using a simple, declarative language known as HCL (Hashicorp Configuration Language).

### 4.1 Infrastructure as a Code

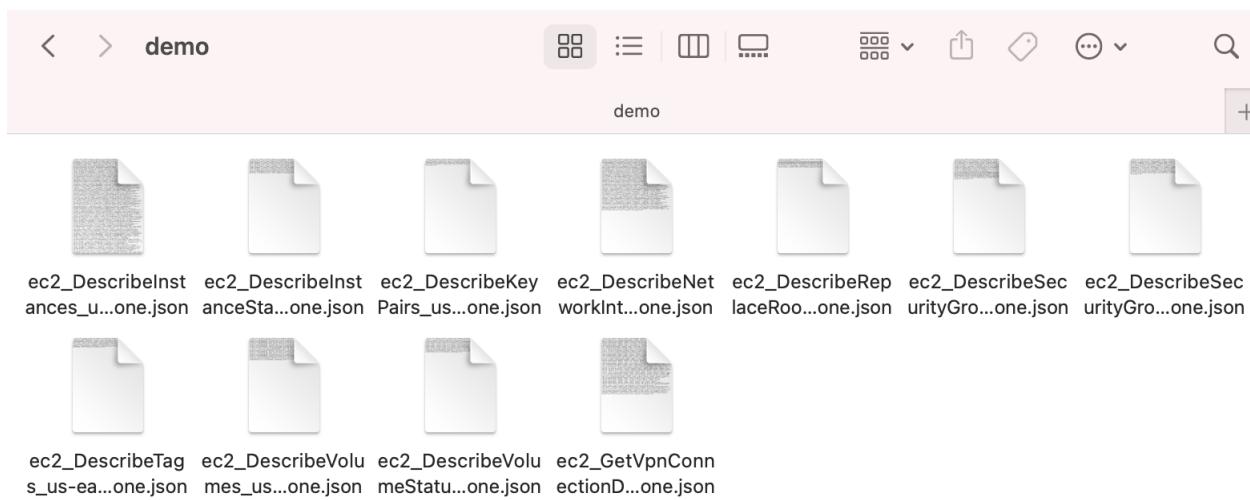
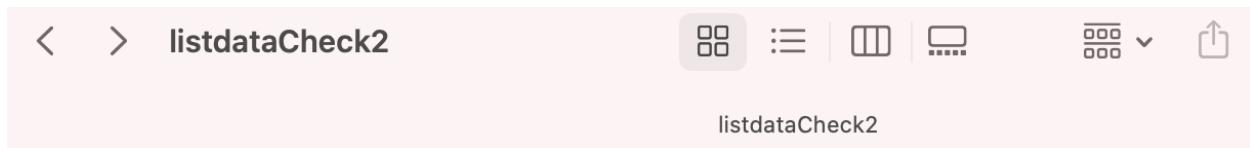
```
ec2 us-east-1 DescribeNetworkInsightsAccessScopes None NetworkInsightsAccessScopes
ec2 us-east-1 DescribeNetworkInsightsAnalyses None NetworkInsightsAnalyses
ec2 us-east-1 DescribeNetworkInsightsPaths None NetworkInsightsPaths
ec2 us-east-1 DescribeNetworkInterfacePermissions None NetworkInterfacePermissions
ec2 us-east-1 DescribePlacementGroups None PlacementGroups
ec2 us-east-1 DescribePublicIpv4Pools None PublicIpv4Pools
ec2 us-east-1 DescribeReservedInstances None ReservedInstances
ec2 us-east-1 DescribeReservedInstancesListings None ClientError("An error occurred (OptInRequired) when calling the DescribeReservedInstancesListings operation: AccountId '63733041330' , You are not authorized to use the requested product. Please complete the seller registration null.") Extensions Help Last edit was seconds ago
ec2 us-east-1 DescribeReservedInstancesModifications None ReservedInstancesModifications
ec2 us-east-1 DescribeRouteTables None RouteTables
ec2 us-east-1 DescribeScheduledInstances None
ec2 us-east-1 DescribeSnapshotTierStatus None SnapshotTierStatuses
ec2 us-east-1 DescribeSnapshots None Snapshots
ec2 us-east-1 DescribeSpotFleetRequests None SpotFleetRequestConfigs
ec2 us-east-1 DescribeSpotInstanceRequests None SpotInstanceRequests
ec2 us-east-1 DescribeStoreImageTasks None StoreImageTaskResults
ec2 us-east-1 DescribeSubnets None Subnets
ec2 us-east-1 DescribeTrafficMirrorFilters None TrafficMirrorFilters
ec2 us-east-1 DescribeTrafficMirrorSessions None TrafficMirrorSessions
ec2 us-east-1 DescribeTrafficMirrorTargets None TrafficMirrorTargets
ec2 us-east-1 DescribeTransitGatewayAttachments None TransitGatewayAttachments
ec2 us-east-1 DescribeTransitGatewayConnectPeers None TransitGatewayConnectPeers
ec2 us-east-1 DescribeTransitGatewayConnects None TransitGatewayConnects
ec2 us-east-1 DescribeTransitGatewayMulticastDomains None TransitGatewayMulticastDomains
ec2 us-east-1 DescribeTransitGatewayPeeringAttachments None TransitGatewayPeeringAttachments
ec2 us-east-1 DescribeTransitGatewayRouteTables None TransitGatewayRouteTables
ec2 us-east-1 DescribeTransitGatewayVpcAttachments None TransitGatewayVpcAttachments
ec2 us-east-1 DescribeTransitGateways None TransitGateways
ec2 us-east-1 DescribeVolumesModifications None VolumesModifications
ec2 us-east-1 DescribeVpcEndpointConnectionNotifications None ConnectionNotificationSet
ec2 us-east-1 DescribeVpcEndpointConnections None VpcEndpointConnections
ec2 us-east-1 DescribeVpcEndpointServiceConfigurations None ServiceConfigurations
ec2 us-east-1 DescribeVpcEndpoints None VpcEndpoints
ec2 us-east-1 DescribeVpcPeerConnections None VpcPeerConnections
ec2 us-east-1 DescribeVpcs None Vpcs
ec2 us-east-1 DescribeVpnConnections None VpnConnections
ec2 us-east-1 DescribeVpnGateways None VpnGateways
ec2 us-east-1 ListImagesInRecycleBin None Images
ec2 us-east-1 ListSnapshotsInRecycleBin None Snapshots
ec2 us-east-1 DescribeInstanceStatus None InstanceStatuses
ec2 us-east-1 DescribeInstances None Reservations
ec2 us-east-1 DescribeKeyPairs None KeyPairs
ec2 us-east-1 DescribeNetworkInterfaces None NetworkInterfaces
ec2 us-east-1 DescribeReplaceRootVolumeTasks None ReplaceRootVolumeTasks, truncated
ec2 us-east-1 DescribeSecurityGroupRules None SecurityGroupRules, truncated
ec2 us-east-1 DescribeSecurityGroups None SecurityGroups
ec2 us-east-1 DescribeTags None Tags
ec2 us-east-1 DescribeVolumeStatus None VolumeStatuses
ec2 us-east-1 DescribeVolumes None Volumes
ec2 us-east-1 GetVpnConnectionDeviceTypes None VpnConnectionDeviceTypes
ec2 us-east-1 DescribeInstanceEventNotificationAttributes None ClientError("No listing: InstanceTagAttribute is no list:", {'InstanceTagAttribute': {'InstanceTagKeys': [], 'IncludeAllTagsOfInstance': False}})
ec2 us-east-1 DescribeTrunkInterfaceAssociations None ClientError("An error occurred (OperationNotPermitted) when calling the DescribeTrunkInterfaceAssociations operation: User 63733041330 is not permitted to perform this operation")
ec2 us-east-1 GetAssociatedEnclaveCertificateIamRoles None ClientError("An error occurred (InvalidCertificateArn.Malformed) when calling the GetAssociatedEnclaveCertificateIamRoles operation: The request must contain a valid certificate arn")
ec2 us-east-1 GetSerialConsoleAccessStatus None ClientError("No listing: SerialConsoleAccessEnabled is no list:", {'SerialConsoleAccessEnabled': False})
ec2 us-east-1 GetTransitGatewayMulticastDomainAssociations None ClientError("An error occurred (MissingParameter) when calling the GetTransitGatewayMulticastDomainAssociations operation: Missing required parameter in request: TransitGatewayMulticastDomainId")
docs.google.com/document/d/1t9hYUdrcI4sPBKh_d4BqT9hOdz0ByXMyazzi_Q6a
```

The data is listed with the “+” symbols, indicating resources that were found in the AWS account, the “-” symbol indicates resources not found in the AWS account and the “!!!” symbol indicates a warning.



The json files then appear within the created “listdataCheck2” folder

In the second created folder directory named “demo”:



The contents of the files created are of “JSON” format and contain the details of each crawled AWS resource; including properties such as “ARN” (Amazon Resource Names) which uniquely identify AWS resources. We require an ARN when you need to specify a resource unambiguously across all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.

When the query is run without a filter on services:

```
aws-list-all query --region us-east-1 --directory ./listdata2/demo2
```

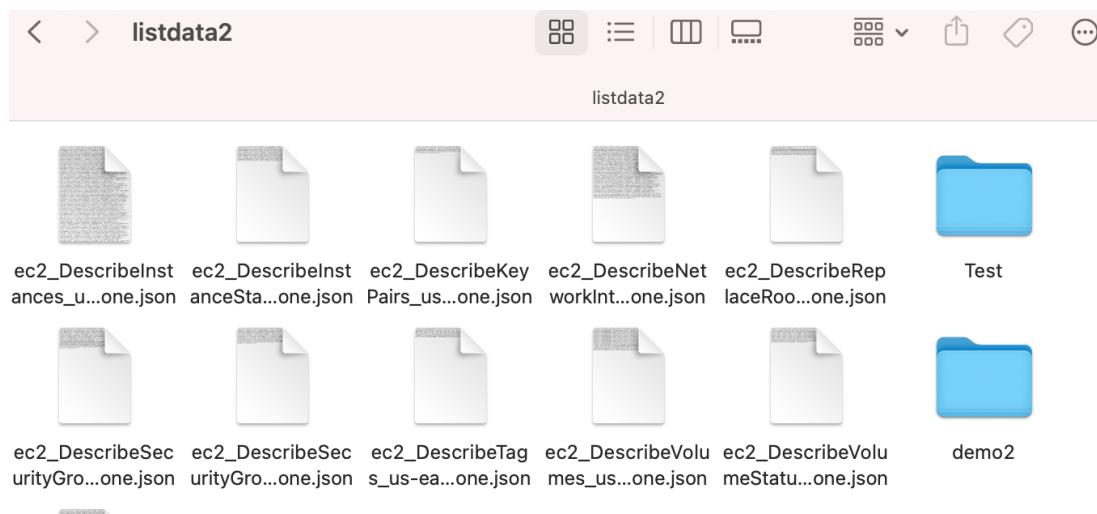
it takes a much longer time to crawl the AWS account for all resources in the specified region the difference in speed is

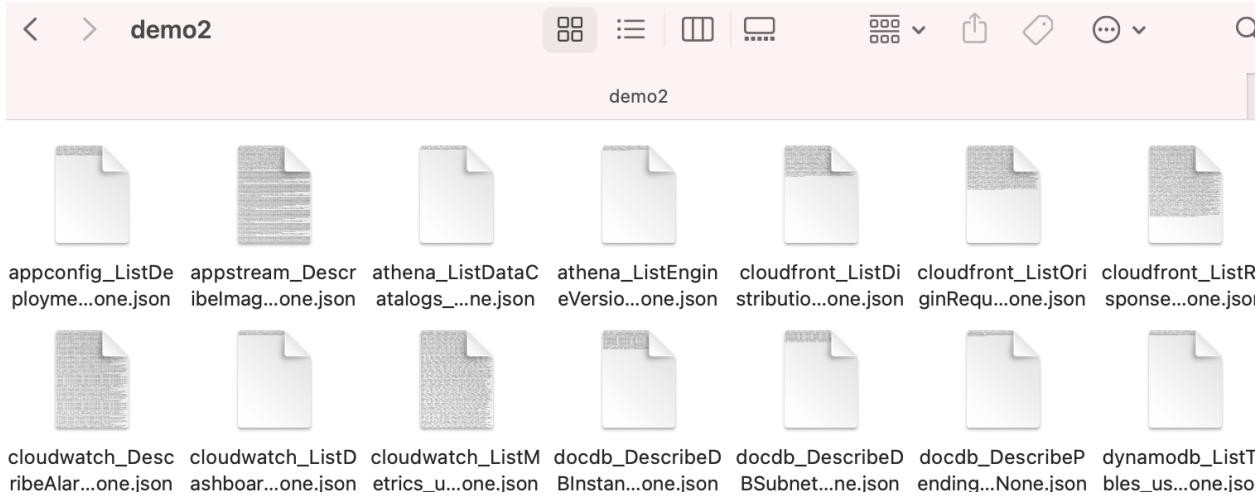
### Benchmarked at

~ 4-6 minutes as opposed to the usual 45 seconds to 1 minute that it usually takes to crawl the AWS account for specific services.

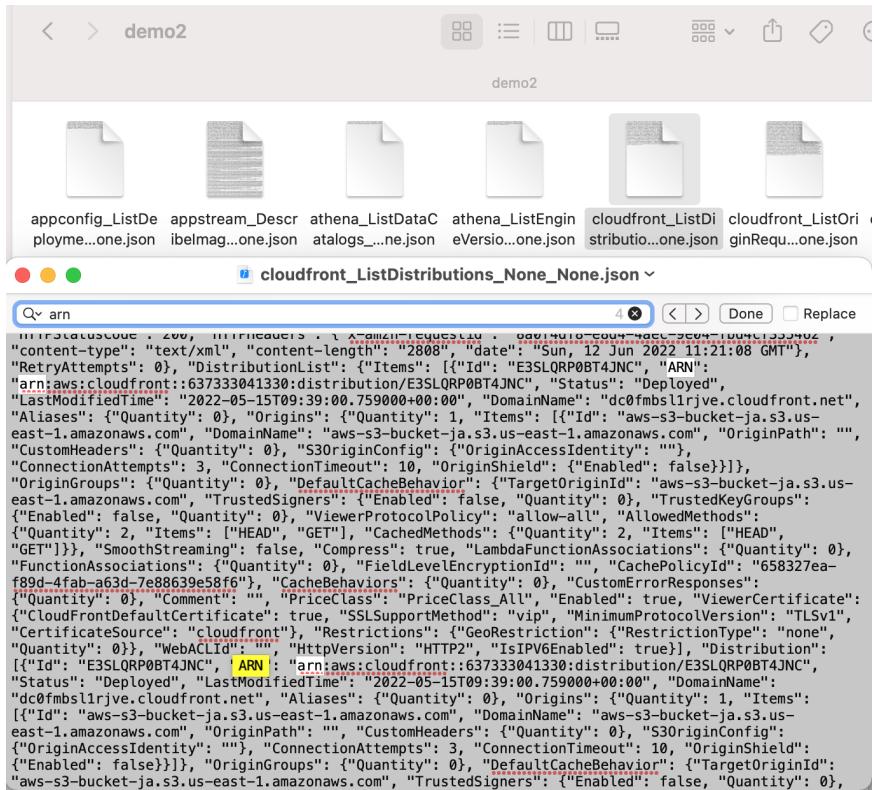
This is due to the scripts utilisation of a list of “service regions” stored in ‘service\_regions.json’ file and “endpoint hosts” stored in the ‘endpoints\_hosts.json’ file for the script to search through, when specific services are not specified in the command line argument.

With the second command , we get more resources listed and stored in json file formats within the demo2 folder located in the “./listdata2/demo2” directory:





And within these json files are the “ARNs” of the crawled AWS resources which can be used to compare against the ARN’s within terraform’s tfstate file to detect infrastructural drift.



**Project github link:** <https://github.com/Jase-The-Ace/beatdrift-cli>