LESSONS LEARNED AND COST ANALYSIS OF HOSTING A FULL STACK OPEN DATA CUBE (ODC) APPLICATION ON THE AMAZON WEB SERVICES (AWS)

[1]Syed R Rizvi, [2]Brian Killough, [1]Andrew Cherry, [1]Sanjay Gowda

[1]Analytical Mechanics Associates, Hampton, VA
[2]NASA Langley Research Center, Hampton, VA

ABSTRACT

The Open Data Cube (ODC) initiative, with support from the Committee on Earth Observation Satellites (CEOS) System Engineering Office (SEO) has developed a state-of-the-art suite of software tools and products to facilitate the analysis of Earth Observation data. This paper presents a short summary and cost analysis of our experience using Amazon Web Services (AWS) to host one such software product, the CEOS Data Cube (CDC) web-based User Interface (UI). In order to provide adaptability, flexibility, scalability, and robustness, we leverage widely-adopted and well-supported technologies such as the Django web framework and the AWS Cloud platform. The UI has empowered users by providing features that assist with streamlining data preparation, data processing, data visualization, and the sub-setting of Analysis Ready Data (ARD) products in order to achieve a wide variety of Earth imaging objectives.

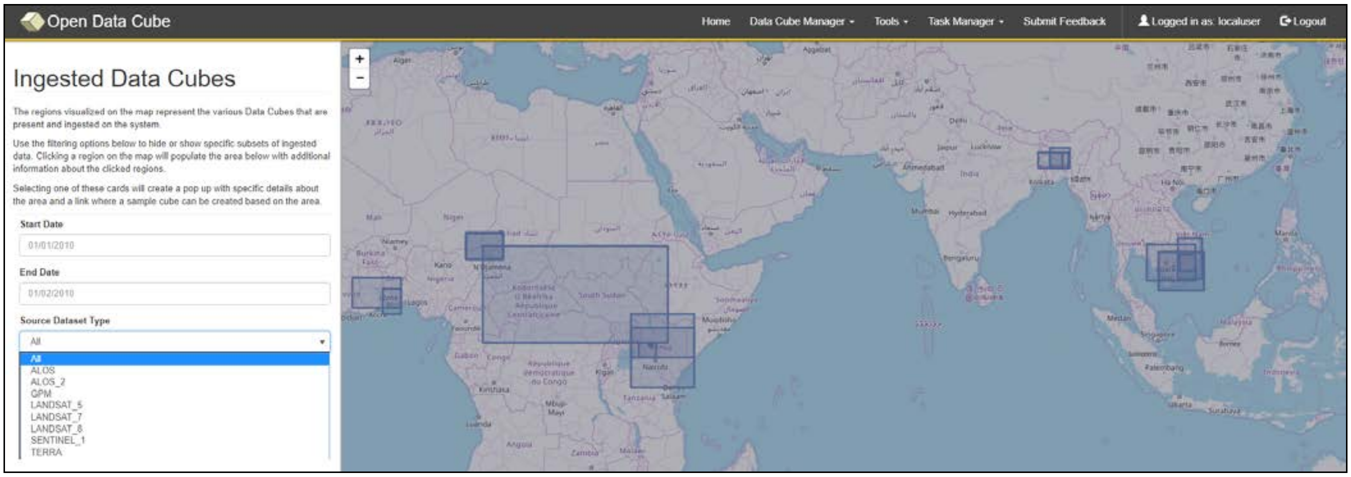
Index Terms— Open Data Cube, ODC, CEOS, Remote Sensing, Earth Observation, Satellite, Amazon Web Services

1. INTRODUCTION

The Committee on Earth Observation Satellites (CEOS) System Engineering Office (SEO) has supported the Open Data Cube (ODC) initiative to provide a data architecture solution that has value to its global users and increases the impact of EO satellite data [1-2]. The Open Data Cube (ODC) is an open-source platform for managing satellite data. We have developed software products and tools around the core ODC. The CEOS Data Cube (CDC) web-based User Interface (UI) is one such well-known tool [3-4]. The UI has empowered users by providing features that assist with streamlining data preparation, data processing, data visualization, and exporting ingested data in order to achieve a wide variety of Earth imaging objectives. In a nutshell, the UI allows analyses to be run from a web interface (Figure 1). Due to the efforts put into developing the UI, CEOS SEO is uniquely able to provide substantial contributions to the ODC initiative and to support global implementations. The web interface, available to the public at http://ec2-52-201-154-0.compute-1.amazonaws.com/, has been used by members of the remote sensing community around the world, and has also been presented at multiple conferences, tutorials, training sessions, and international presentations [5-7].

The UI (along with the ODC core) utilizes a number of different software frameworks, including Python, JavaScript, PostgreSQL, and the Django web framework. It is hosted on an Ubuntu operating system and the source code is publicly available under the Apache License, Version 2.0. The Python programming language is greatly suited for research in scientific computing, remote sensing, Earth science, and machine learning due to its extensive standard library and selection of add-on packages, its readability, and its ease of programming compared to other languages, and the great number of help resources easily found online. The ODC utilizes PostgreSQL to meet security and performance requirements by organizing the data into stacks of consistent, time-stamped geographic “tiles” which can be rapidly manipulated in an HPC environment. The database not only organizes the data and metadata for the ODC core and Django framework, but can also be used to track every observation back to the point of collection, thus providing data provenance. AWS has been used as a one-stop solution for web hosting, parallel and distributed processing, and data storage, distribution, and analysis.

The bulk of our usage has been Amazon Elastic Compute Cloud (Amazon EC2) instances, which we are using for both analysis of remote sensing data and the hosting of the UI. EC2, in general, makes web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows us to obtain and configure capacity with minimal friction. With EC2, we created an Amazon Machine Image (AMI) containing an operating system, application programs, and configuration settings.



Ingested Data Cubes

The regions visualized on the map represent the various Data Cubes that are present and ingested on the system.

Use the filtering options below to hide or show specific subsets of ingested data. Clicking a region on the map will populate the area below with additional information about the clicked regions.

Selecting one of these cards will create a pop up with specific details about the area and a link where a sample cube can be created based on the area.

Start Date: 01/01/2010

End Date: 01/02/2010

Source Dataset Type: All

Figure 1. Web User Interface (UI) of CEOS Data Cube (CDC).

We are currently running two instances that are used together as a clustered computing system for both our analysis cases and the UI operations. The two instances subscribe to a single job queue and the main process divides large tasks into smaller tasks in order to take advantage of all CPU cores and memory available to us. This gives us the option of adding additional instances in the future, scaling horizontally to handle periods of heavy demand.

S3 has been used mostly for distribution of sample datasets to interested parties and the long-term storage of such datasets. We have developed an interface that includes descriptions of our datasets, the datasets themselves, and instructions for the use of the data, as well as an administrative interface to manage the UI itself. The fully-customizable source code of the UI is available at our public repository [3]. Interested parties can download the source, and build their own UIs. In the future, we may keep a larger amount of data on S3 and put links to the relevant data on our UI for users to download.

2. COST BURDEN

We began using AWS for our hosting and storage needs in April of 2016. A sample report of our costs grouped by service from April 2017 to January 2018 can be seen in Figure 2. This paper will describe the cost during this duration in order to illustrate some insights obtained from our recent experience. Additionally, Table 1 and 2 show the monthly AWS calculator for the *Amazon EC2 Instances* and the *Amazon EBS Volumes* respectively [8].

The bulk of our cost has been the EC2 instances. We are currently running two c4.8xlarge instances for use in our parallel processing cluster for a combined 72 virtual CPU cores and 120 GB of RAM. The EC2 instances have a predictable and constant cost as they have 100% uptime and are used to host our Data Cube UI. Note that the actual utilization of this 100% uptime is low. Since many of the analyses involve loading and processing multiple gigabytes of data per region, we have been able to optimize our systems to use all available resources for each task.

Secondary costs to the EC2 instances are in the EC2-Other category and include snapshots, storage, and elastic IP addresses.

This cost is driven mostly by the amount of storage we are using at any given time. The raw data (mostly GeoTIFF scene data) is *ingested*, i.e. pre-processed into aligned, compressed blocks which are 7-8 times smaller. For example, in one of our case studies related to determining historical trends in the water quality of Lake Chad in Cameroon, we created a small data cube (0.25 degrees square) for the southern portion of the lake. The raw data in this case study was around 920GB (unzipped) but the pre-processed NetCDF files amounted to around 117GB. After pre-processing, the raw data is not needed for any later processing so we are only hosting the pre-processed data, totaling roughly 500GB per server. We are currently replicating data between the servers, but plan to move to shared Elastic File Systems for dataset storage in the future.

S3 was our lowest cost, showing only small spikes during times of large data transfer. Note that large data transfer occurs when moving the raw data to the cloud for ingestion. CEOS SEO aims to reach operational Data Cubes in 20 countries by 2020. As of early 2018, there are three operational Data Cubes (Australia, Colombia, and Switzerland) [6], seven in development (Georgia, Moldova, Taiwan, Uganda, United States, United Kingdom, and Vietnam) [7] and 29 other countries with expressed interest. As the interest and involvement from these counties grow in the future, the S3 cost will go up in when we move to make more of our datasets available to additional UI users.

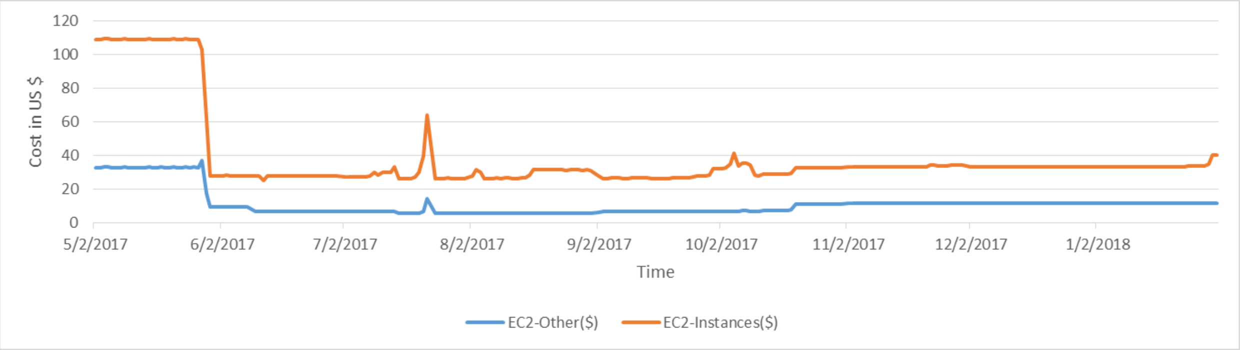


Figure 2. Costs grouped by service.

Table 1. Monthly Calculator for Amazon EC2 Instances (Compute) [8].

Description	Instances	Usage	Type	Billing Option	Monthly Cost
Worker	3	100% utilized per month	Linux on c4.2xlarge	On-demand	$874.02
Notebook Server	1	100% utilized per month	Linux on m4.xlarge	On-demand	$146.40
CEOS Main (Burstable)	1	1% utilized per month	Linux on t2.2xlarge	On-demand	$2.97
Worker Image	1	0% utilized per month	Linux on m4.2xlarge	On-demand	$0.00

Table 2. Monthly Calculator for Amazon EBS Volumes (Storage) [8].

Description	Volumes	Volume Type	Storage	IOPS	Baseline Throughput (MBs/sec)
CEOS Main	1	General Purpose SSD (gp2)	300GB	900	160
CEOS Main Data	1	Throughput Optimized HDD (st1)	8192GB	0	320
Misc. (attached)	6	General Purpose SSD (gp2)	75GB	225	128
Misc. (unattached)	3	General Purpose SSD (gp2)	75GB	225	128

3. JUPYTER NOTEBOOKS

Recall that the bulk of our usage has been EC2 instances, which are used for both analysis of remote sensing data and hosting the UI. We also host an ODC Jupyter Notebook server on EC2. These notebooks act as interactive Python development environments which allow developers to divide their code into blocks which can be run independently of each other, with variables stored in the background and the environment persisted between blocks. The notebooks were instrumental in providing hands-on training to many international users in the remote sensing community and have been presented at multiple conferences, tutorials, training sessions, and international presentations [5-7].

4. TESTING APPROACH

Testing a web application such as the UI component of ODC is a complex task because it is made of several layers of logic – from HTTP(S) request handling, to form validation and processing, to template rendering. We heavily utilize Django's automated test-execution framework and assorted utilities. It simulates requests, inserts test data, inspects the application's output and generally verifies the source code for correctness. We have utilized the combination Unittest/Nose2 testing framework for automated unit tests, code coverage, etc. The Selenium and Locust web testing frameworks have also been explored for additional UI testing.

5. WORK-IN-PROGRESS

Apart from the plans for AWS usage that have been described in the previous sections, the main features we are currently targeting for near-term development are Elastic File System, SPOT Processing, and increasing the utilization of our current resources.

The current plan is to set up an EFS system to cut back on our data duplication and to allow for greater scalability as we add more EC2 instances. Some added benefits of this approach include using the same system for passing data and intermediate products back and forth between EC2 instances during parallel processing, and removing the need to transfer large amounts of data when we create a new instance.

Although this will slightly increase our storage costs per month with our current number of instances, it allows for greater scalability and will reduce costs when we have many more instances. Figure 5 illustrates the cost of storage with and without EFS as additional nodes are added. Currently, we are using EBS ST1 volumes which are $.045 per GB per month. Our parallel processing requires keeping redundant data on each server. Therefore, we pay 2 × $.045 = $0.09 per GB per month. On the other hand, EFS storage costs $0.30 per GB month, so if we were running 6+ instances then using EFS would become cost-effective.

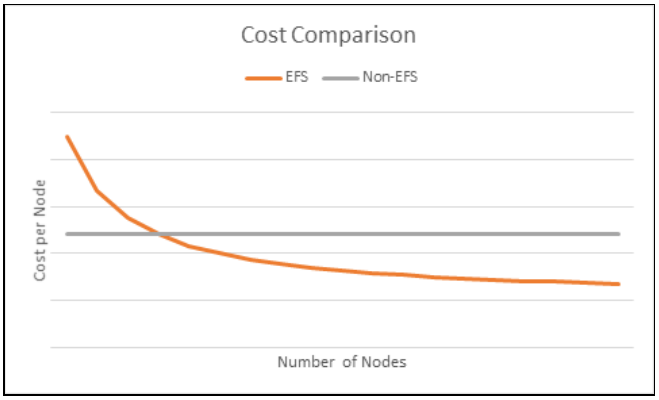


Figure 5. The theoretical growth of cost with EFS and without EFS, as nodes are added.

6. FUTURE PLANS

For the coming year, we intend to investigate the following ODC concepts with AWS:

1. Develop a "Data Cube on Demand" function using hosted AWS datasets.
2. Test the use of "spot" on-demand processing to support global data cube deployments.
3. Test the use of *Lambda* functions for finding new datasets to ingest into data cubes.
4. Test how EC2 instance performance scales with multiple data cube users.
5. Test elastic load balancing for horizontal scaling of EC2 instances for data cubes.
6. Test AWS "Workspaces" to host QGIS and Jupyter Notebooks for cloud analysis.
7. Test the use of *Docker Containers* for on-demand computing instances
8. Explore the use of QGIS to read data cube content directly from S3

7. CONCLUSION

Amazon AWS has served as a unified solution for all of our CDC storage and analysis needs. We have both expanded the use of our currently employed features and branched out to several new services offered by Amazon. The services AWS provides have allowed us to create an internationally accessible interface where users in the remote sensing community can see our progress, access our data, and understand the impact of open satellite data and its application. The AWS-hosted CDC UI and Jupyter notebooks play a critical role in demonstrating how the Open Data Cube can take advantage of the AWS infrastructure and exploit open datasets in order to achieve the CEOS SEO goal of having operational Data Cubes in 20 countries by 2020.

8. ACKNOWLEDGMENT

The authors would like to acknowledge the efforts of Jesse Harrison and Alfredo Delos Santos on the early development of the CEOS ODC web-based UI.

9. DISCLAIMER

Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

10. REFERENCES

[1] Open Data Cube Website: https://www.opendatacube.org.
[2] Open Data Cube GitHub Repository: https://github.com/opendatacube.
[3] CEOS Data Cube User Interface GitHub Repository: https://github.com/ceos-seo/data_cube_ui.
[4] CEOS Data Cube web-based User Interface: http://ec2-52-201-154-0.compute-1.amazonaws.com/.
[5] The 1st CEOS Open Data Cube Workshop: http://ceos.org/home-2/1st-ceos-open-data-cube-workshop/.
[6] G. Giuliani *et al.*, “Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD),” *Big Earth Data*, vol. 1, no. 1, pp. 1–18, Nov. 2017.
[7] A. Singh, “New satellite data sharing system, Vietnam Data Cube, introduced,” GeoSpatialWorld, Mar-2018. [Online]. Available: https://www.geospatialworld.net/news/new-satellite-data-sharing-system-viet-nam-data-cube-introduced. [Accessed: 15-Mar-2018].
[8] AWS Calculator for CEOS Data Cube systems. [Online]. Available: https://calculator.s3.amazonaws.com/index.html#r=IAD&s=EC2&key=calc-FAF98858-1C94-4FDF-AC1E-CEE1F33EDDE6. [Accessed: 03-Jan-2018].