

DEVOPS Challenge:

The Challenge

Your challenge, should you choose to accept it, is to develop an AWS S3 storage analysis tool. To test your tool, you will have to create a free Amazon account (if you don't already have one).

Specifications

The tool returns information over all S3 buckets in an Amazon account. In particular, it must return the following info:

- Bucket name
- Creation date (of the bucket)
- Number of files
- Total size of files
- Last modified date (most recent file of a bucket)

The tools should provide the following methods/features:

- Ability to get the size results in bytes, KB, MB, ...
- Organize the information by storage type (Standard, IA, RR)
- Ability to group information by regions

Nice to have:

- Support prefix in the bucket filter (e.g.: s3://mybucket/Folder/SubFolder/log*).
- Organize the results according to the encryption type, get additional buckets information (life cycle, cross-region replication, etc.)
- Diff by taking into account the previous file versions in the count + size calculation.
- Some statistics to check the percentage of space used by a bucket, or any other good ideas/features you could have, are more than welcome.

Rules

Code:

- Use Python as the language to implement this.
- Your code must be made available as a public git repo

Expose:

- Provide your tool via a CLI
- Provide your tool via a REST interface (*bonus*)

Test

- Provide a way to test your implementation
(both Integration test as well as Unit Test with mocked S3 API)

Deploy

- Deploy the solution as a service in an AWS free instance using your favourite tools for deployment, provisioning, and packaging (ansible, salt, aws cli, docker, linux service, rpm, python packages, etc)
- Show how you would test the deployment/packaging/provisioning of your tool.
- Expose the REST API (*bonus*)

Performance:

We will test your work over our environment (which contains millions of files).

The overall performance of your tool will be evaluated.

Advices

Try to design and implement your solution as you would do for real production code. Show us how you create clean, maintainable code that does awesome stuff. Build something that we'd be happy to contribute to. This is not a programming contest where dirty hacks win the game. Feel free to add more features! Really, we're curious about what you can think of. We'd expect the same if you worked with us.

Documentation and maintainability is a plus.

Extended testing, Automation, Monitoring is a plus