We are looking for a full-time remote DevOps Engineer who has worked with CI/CD automation, big data pipelines and Cloud Infrastructure, to solve complex technical challenges at scale that will reshape the healthcare industry for generations. You will get the opportunity to be involved in the latest tech in big data engineering, novel machine learning pipelines and highly scalable backend development. The successful candidates will be working in a team of highly skilled and experienced developers, data scientists and CTO.

Job Requirements

• Experience deploying, automating, maintaining, and improving complex services and pipelines • Strong understanding of DevOps tools/process/methodologies
• Experience with AWS Cloud Formation and AWS CLI is essential
• The ability to work to project deadlines efficiently and with minimum guidance
• A positive attitude and enjoys working within a global distributed team

Skills

• Highly proficient working with CI/CD and automating infrastructure provisioning
• Deep understanding of AWS Cloud platform and hands on experience setting up and maintaining with large scale implementations
• Experience with JavaScript/TypeScript, Node, Python and Bash/Shell Scripting
• Hands on experience with Docker and container orchestration
• Experience setting up and maintaining big data pipelines, Serverless stacks and containers infrastructure
• An interest in healthcare and medical sectors
• Technical degree with 5 plus years' infrastructure and automation experience

---

DevOps Challenge

Your challenge, should you choose to accept it, is to create 3 buckets on AWS S3. And write a script to list out the content of those buckets. To test your tool, you will have to create a free Amazon account (if you don't already have one).

Specifications

The tool is a shell command line utility or a node script that returns information over all files inside the three S3 buckets in an Amazon account.

- Your tool must work on Ubuntu or OSX.
- It must be easy to install and use.
- Ideally, your tool won't require installation of any other tools / libraries / frameworks to work.
- Time is money, we cannot afford running a tool that takes hours to complete, your solution should return results within seconds (or minutes if you are willing to test our patience :-).

Time allowed

People who have successfully passed the challenge and are now happy members of the Vamstar team usually took from 4 to 10 hours to complete it.

Remember the KISS principle...

The tool must return the following information

For each bucket:

- Name
- Creation date
- Number of files
- Total size of files
- Last modified date of the most recent file
- And the most important of all, **how much does it cost. The following options should be supported.**

Display

Ability to get the size results in bytes, kB, MB
Ability to group buckets by regions

Filters

By bucket name

By storage type (Standard, IA, RR). You may provide stats on the objects in the buckets (how many have which storage type) and/or add a filter on storage type (the bucket data would only reflect the objects that have the chosen storage type).

Some additional features that could be useful (optional)

It would be nice to:

- Support prefixes, glob and/or regexes 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.)
- Take into account the previous file versions in the count + size calculation
- Create the buckets using Cloud Formation

Some statistics to check the percentage of space used by a bucket, or any other good ideas you could have, are more than welcome.

Rules

- Your are free to use the programming language and the SDK of your choice, but remember that installation must not require us to install external tools in order to test the result of your work.
- We will test your work in our environment (which contains **several millions of files**). The overall performance of your tool will be evaluated. Most of the projects we receive take weeks to run in our environment, can you do better?
- Your code must be made available on GitHub or any other public version control software. Your project must be a standalone project (i.e. **do not fork it from our challenge or any other project**).

Advice

- **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.
- Don't you forget those unit tests.
- We don’t want to know if you can do exactly as asked (or everybody would have the same result). We want to know what **you** bring to the table when working on a project, what is your secret sauce. More features? Best solution? Thinking outside the box?