Report

XML to JSON

Parsing XML file to JSON is done with beautiful soup which uses lxml. lxml is based on the C libraries libxml2 and libxslt, so it should be one the fastest solutions. The files as described from the assignment should be around 6-7 MB. Even if each file is a magnitude higher, it doesn't use lots of RAM, while opening and parsing. Thus, streaming solutions, as SAX, have been avoided. In case of expecting files of two or three magnitudes higher, it will be a viable solution to lower down the memory footprint of the service. Also, while streaming it might be beneficial to use JSON lines instead of normal JSON, as they make appending to files easier. Finally, we assume that the XML files are secure. In case the source is not trusted, another XML parser might mitigate or complete stop the attack. For more information visit defusexml.

Amazon Web Services

For this assignment, it is assumed that a single AWS profile has access to both the client's and the server's S3 bucket. Also those buckets pre-exist, along with Simple Queue Service (SQS), at the same region. The SQS is unencrypted at the server side, but the messages are being sent over HTTPS, when a new object is being created at the client's bucket. The SQS is being set up to work with the client's S3 bucket, before the server starts listening to new messages. Another solution would be to list the client's bucket every N minutes, but it might cost more, as SQS first 1 million requests is free per month.

Further Work

The reason for saving locally the JSON files is that failure could happen while uploading the files to the server's S3 bucket. A service could check, every 24-48 hours, if every file is uploaded, then try to add any missing files to the S3 bucket and finally delete them from the local server. Files can be chronologically ordered as each filename has a timestamp.

Tests

Due to time constraints, tests do not cover all the code, but offer an insight of how they could work. Two categories exists, the normal unittests and the moto tests. The latter can mock up a session to AWS, through boto3 and assert the results of each action. Localstack works great to test the infrastructure, while moto offers a more lower level approach that sometimes makes testing code through Python easier.