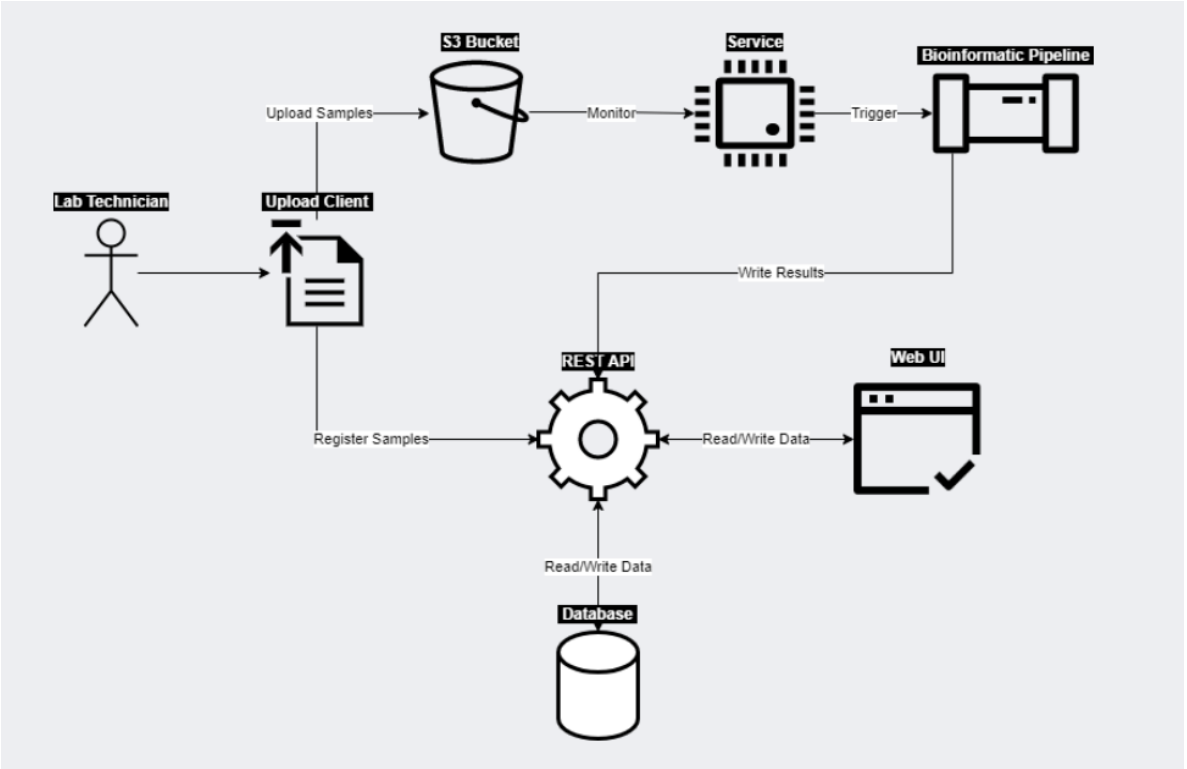
Design Challenge



Overview

Test Objective:
The objective of the test is to validate that the platform can analyse 1,000,000 samples within a 7 day period and then to benchmark the performance at varying levels of load.

Technologies:
In order that this objective can be achieved the following technologies would be used.

- Python: Used to write custom scripts which would automate the process of generating test data and uploading samples using the Upload Client. Could also later automate these tests.
- Upload Client - to upload the test samples to the S3 bucket.
- SQS - trigger the bioinformatic pipeline for the uploaded samples.
- Bioinformatic Pipeline - to analyse the samples and produce the key/value results.
- REST API - to write the results to the database and view the results in the web UI.
- Load Testing Tool - simulate the increasing number of samples and measure performance. Could use either JMeter or Gatling for this.

Test Design

Test Setup:

- Configure the Upload Client with the appropriate deployment URL and token.
- Upload 400 test samples to the S3 bucket using the upload client.

Test Execution:

1. Monitor the backend service to receive the SQS notification for the batch of samples uploaded.
2. Verify that the bioinformatic pipeline is triggered and starts analysing the samples.
3. Wait for the pipeline to complete the analysis of all 400 samples.
4. Verify that the pipeline produces the expected number of key/value results on a per sample basis.
5. Check that the results are written to the database using the REST API.
6. Verify that the results can be viewed in the web UI using the same REST API.

Performance Test Scenarios:

To benchmark the performance at varying levels of load the following test scenarios will be executed.

- Scenario 1: 10,000 samples uploaded concurrently with 10 users using the Upload Client.
- Scenario 2: 100,000 samples uploaded concurrently with 50 users using the Upload Client.
- Scenario 3: 500,000 samples uploaded concurrently with 100 users using the Upload Client.
- Scenario 4: 1,000,000 samples uploaded concurrently with 50 users using the Upload Client.

While running these scenarios the following checks will be carried out.

Test Metrics:

The following metrics will be captured for each scenario:

- Response Time: The tame taken for a sample to be uploaded, processed, and results written to the database.
- Throughput: This metric would show the number of samples processed per unit of time (e.g. samples per minute or samples per hour). It can be used to monitor the performance of the system over time.
- Error Rate: The percentage of failed requests due to errors in the system.
- Network usage: amount of network traffic generated during the test. See if the system is overwhelming the network.
- Memory usage: Amount of memory system used under test, can be used to identify memory leaks.
- CPU usage: percentage used by system under test.
- Total processing time: This metric shows the total time it took for the system to process all the samples in the test.
- Processing time per sample: Metric will show how long it takes for the system to process a single sample. Measured from the time the sample is uploaded to when it is written in the database.
- Total number of samples processed: Metric will show the total number of samples that were processed during the test. Should match the amount that were uploaded.

Outputs:

Test Report:
Document the results and findings from the test execution section.

Performance Report:
A test report will be generated for each scenario, which will include the following details:

- Test environment details, including number of instances, configurations and services used.
- Test objectives and scenario details.
- Test execution details, including the duration, load and metrics captured.
- Analysis of the test results, including graphs and tables.
- Recommendations for improving the performance of the system.

Assumptions:

- The production environment is similar to the test environment in terms of hardware configurations, network connectivity, and services.
- The sample data generated during the test is representative of the production data.
- The bioinformatic pipeline can scale linearly, and there are no bottlenecks or single points of failure in the system.

Questions:

- Are there any constraints on the resources that can be used for performance testing? (e.g., CPU, memory, network bandwidth)
- What are the maximum and minimum file sizes of the samples?
- What is the expected response time for a sample to be processed?
- What is the expected time taken for the bioinformatic pipeline to analyse a batch of 400 samples?
- What are the SLAs (Service Level Agreements) for the REST API and web UI?
- What security measures are in place to protect the data?