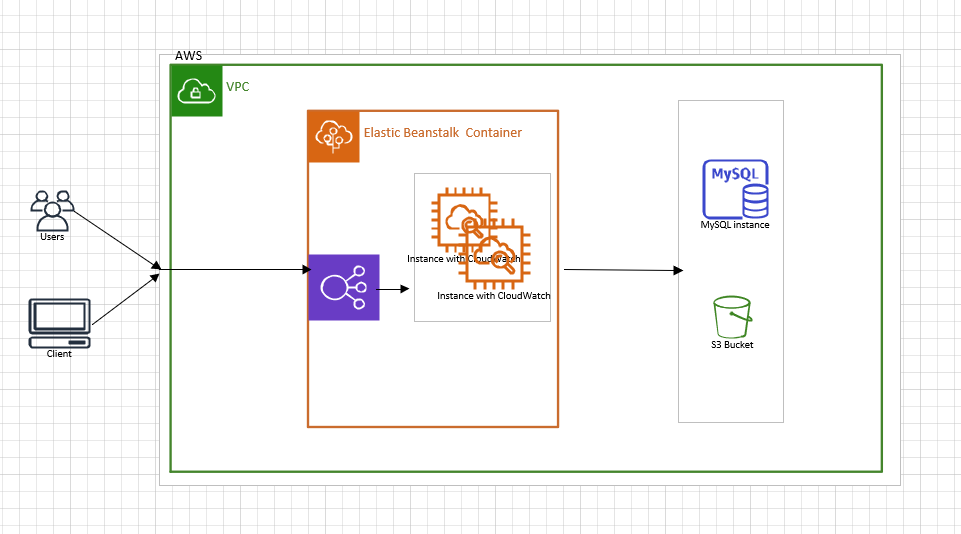
1. Project Overview

Patent files should be able to process and store the information. This goes by two steps, as a first step need to download the patent files from USPTO website and do OCR process on downloaded patents. Store the metadata and processed data.

OCR ([Optical Character Recognition](https://nanonets.com/blog/what-is-ocr/)) is a technology that identifies and recognizes text within scanned documents, photos, or images. OCR software leverages this technology to [extract data from PDFs](https://nanonets.com/blog/extract-data-from-pdf/) or scanned documents by converting it into machine-readable text data that can be edited & stored more conveniently for further processing.

Using the Opensource solution **Tesseract** to process the image.

1. Technical Design Diagram



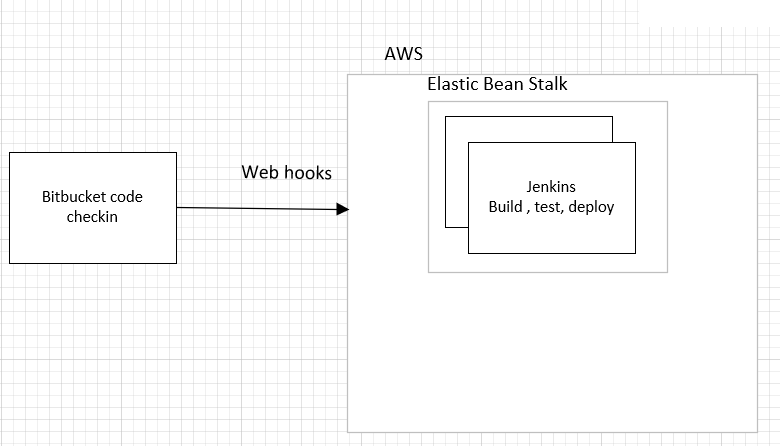
**3. Technical Specifications**

* OpenJdk11
* Spring Boot 2.5.3
* Embedded tomcat of the Spring Boot
* Docker
* S3
* LDAP
* MySQL database
* Splunk
* Grafana
* AWS
* Jenkins/ CodeBuild for CI/CD
* Elastic Beanstalk
* Cloud Watch

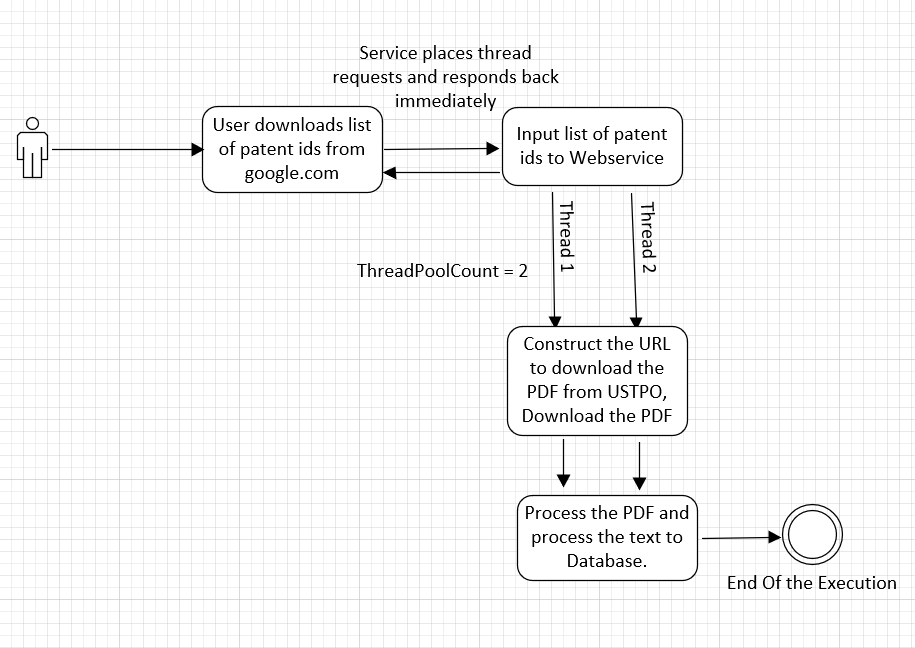
**4****. Assumptions**

1. USPTO website should allow us to download the info lawfully from code.
2. USPTO should have webservices for metadata to consume.
3. USPTO website should not be hammered so as of now taking thread count as 2. Can be changed later.

**5. CI/CD process**

****

**5. Use case Diagram**

****

**6. Security**

Before accessing any API, user/system should request the **JWT** token by providing the username password by using “/authenticate” service. Once token is available then input the token to other services as a header.

**Token generation**: UserDetails are authenticated against **LDAP**. Once details are verified, JWT token would be generated.

**Token Verification**: Read the JWT token from header, parse it and validate the timestamp and user details.

**7. Cost**

Based on the cost aspect. AWS services can be used if necessary. One of the costs reducing factor is to scale services based on the load. So that resources would be provisioned based on the usage.

**8. Monitor**:

* All Application logs would be forwarded to Splunk. Dashboards/Alerts can be created from the logs source.
* All Memory and CPU usage metrics also can be forwarded to Splunk to monitor the service loads.
* Health Checks should be enabled to alert team if any service is down.
* Metrics can be created using Grafana.

End of the Document