**Description of Services Used:**

1. **Api Gateway:** Used as it can handle up to 10,000 requests per second. This quota can be increased allowing for scalability. Api Gateway allows for the easy creation of the REST Apis, allowing for developers to easily publish, maintain and secure APIs. Api Gateway handles all the essential tasks such as throttling, monitoring, authorisation, traffic management etc. It is also very cost effective due to payment only being applicable for the API calls made. Additionally, It also has easy connectivity to AWS Lambda
2. **AWS Lambda:** Allows us to run the typescript code without managing or setting up the servers. It is easily scalable and allows for asynchronous data processing. Additionally, it is very cost effective due to the payment being application for the compute time used i.e. no need for payment when an event is not present. Lastly, it allows for the easy communication between decoupled services – essential for event driven applications
3. **MySql on AWS RDS:** MySQL has been chosen as it is an open-source relational database that uses SQL to query – satisfying the requirements. Being a very popular database, it has benefits such as security and support. Implementing MySQL on AWS allows us to leverage the benefits of both MySQL and AWS. MySQL on AWS allows for automated backup, easy deployments, high availability, automatic security patches, monitoring and fast performance.
4. **S3 Bucket:** AWS S3 was chosen to store the events – possibly for audit and compliance requirements. S3 has excellent scalability, availability, security and performance. The event data can be archived to S3 Glacier and can be further backed up by S3 to ensure availability. Furthermore, bucket policies can be implemented on S3 to prevent the unauthorised access of data.
5. **AWS IAM:** Used to specify what resources can access services and also limit the level of access. For e.g. IAM role attached to the lambda can ensure that the lambda is only able to implement the GET, PUT and POST Commands. But it is unable call the DELETE command due to the restrictions placed. This enables increased security for the application. In this instance, the security would be set towards that of the ‘least privilege’.
6. **Cloudwatch:** Can be used for logging and tracking application metrics. Logging allows us to monitor the system and resolve issues that may arise. Cloudwatch Metrics allow us to monitor the performance of our systems. For e.g. Cloudwatch metrics allow us to analyse the number of lambdas spun up and is there was any throttling for the lambdas.