This project tries to fulfil the requirements of the AWS Well-Architected Framework.

Operational excellence is achieved through discussions about how to implement the architecture needed for our application. As an organization, we have chat channels for ease of communication. Additionally, we hold meetings to handle difficult subjects and formulate plans how to tackle them.

On AWS console, several metrics and logs can be utilized to help manage operations and development.

Security. We plan to implement AWS Cognito to authenticate application users. Identity and access management service can be used to implement principle of least privilege. This ensures only those who need access to each part of the architecture, have access. Additionally, the first phase of the project will not be scaled to a global application. This helps stagger any compliance issues we might face.

Reliability is improved by running the application code on multiple availability zones. This ensures that a single data center failing does not prevent our application from running. Our environment also makes use of load balancer that ensures one server is not flooded with more requests than it can handle. Data is also duplicated on two separate database instances to make sure fault in one does not cause loss of data.

Performance efficiency is at the heart of the application and architecture. For computation Elastic Beanstalk was selected. The auto-scaling features, in conjunction with the load balancer, ensure the application can handle spikes in traffic.

Another great service for performance is the AWS CloudFront. CloudFront allows the application to serve content from the closest edge location to the user. CloudFront can also be used to serve cached static files. The edge cache reduces the number of requests that need to be made to the backend services, further lightening the load for the BeanStalk. These features ensure we can provide a low latency user experience.

Cost optimization is something that needs to be looked at before the application is released into the wild. We will need to make full use of the free trial periods. This time allows us to look at metrics and data, and make changes where they are needed to optimize cost savings. Some things to look at could be BeanStalk rightsizing and make sure the automatic scaler works as intended.

Some services, like Route 53 are free. Designing the final architecture emphasizing free services can be beneficial. Several other services are pay-as-you-go, which also allows us to save costs. In theory, pay-as-you-go should generate more revenue than costs.