Bonus Question

Think about how you would expose this service to the world. What are those considerations you should think about and what topics should be on your launch-list?

I believe the following things should be on my launch-list to increase the likelihood of a successful and secure deployment of my Go and ReactJS app in Docker containers to the world.

- Server Security: Set up a server and ensure the server is properly configured and secure against unauthorized access. (SSH for remote access and configuration)
- HTTPS: Implement SSL/TLS certificates to enable HTTPS for data encryption.
- Load Balancing: Use load balancing to distribute traffic and improve availability.
    - Possible tool to be used: AWS Elastic Load Balancer.
- CI/CD Pipeline: Set up automated testing and deployment for code changes.
    - Possible tool to be used: Jenkins or Github Actions.
- Monitoring and Logging: Implement basic monitoring and logging for app health and troubleshooting.
    - Possible tools to be used: ElasticSearch, Kibana.
- Scalability: Plan for scalability to handle increased traffic.
    - Possible tools to be used: Kubernetes for container orchestration.
- Database Security: Set up a database, secure it and ensure proper backups.
- Content Delivery: Use CDNs to cache and serve static assets.
    - Possible tools to be used: Akamai.
- Error Handling: Implement robust error handling and recovery mechanisms.
- Documentation: Create documentation for deployment and troubleshooting.
    - Possible tools to be used: Confluence.
- Backup and Disaster Recovery: Establish regular backup and recovery plans, using built-in database tools or cloud snapshots for server recovery.

- Testing: Thoroughly test the app in a staging environment.

- User Feedback and Support: Set up mechanisms for user feedback and support.
    - Possible tools to be used: Zendesk, feedback forms or emails.

- Launch Plan: Develop a launch plan with milestones and timelines.
    - Possible tools to be used: Jira for creating and tracking launch tasks and milestones.

- Monitoring and Scaling Strategy: Define how to monitor and scale resources using the cloud provider monitoring services.

- Backup Rollback Plan: Prepare a rollback plan for critical issues.

    - Possible tools to be used: dev’s best friend – Git.