Here’s a detailed explanation you can use in an interview to confidently talk through your project:

----------------------

✅ **Project Overview**

*"I deployed a Java Gradle-based web application on AWS using Kubernetes (EKS) and exposed it over a custom domain using an Application Load Balancer (ALB) via Ingress. I also configured HTTPS using AWS Certificate Manager and Route 53."*

----------------------

🔧 **Steps I Performed**

**Built & Containerized the App**

*"I used Docker to containerize the Gradle application, then pushed it to Docker Hub for deployment in Kubernetes."*

**Created Kubernetes Resources**

*"I wrote and applied YAML manifests for Deployment, Service, and Ingress. The Deployment managed pod replicas; the Service exposed them internally using NodePort; and the Ingress exposed the application externally via ALB."*

**Configured ALB Ingress Controller**

*"I annotated the Ingress to use the AWS ALB Ingress Controller, set the scheme to internet-facing, and added ports 80 and 443 for HTTP and HTTPS traffic."*

**Handled HTTPS with ACM and Route 53**

*"I issued an SSL certificate using AWS ACM and attached it to the Ingress. Then, I created an A record in Route 53, pointing my custom domain to the ALB endpoint."*

**Debugged 502 Errors**

*"Initially, I faced 502 Bad Gateway errors. I debugged this by inspecting the Ingress, Service, and target group health checks in AWS. I discovered that the service and container ports were mismatched — the container listened on port 8080, but the service exposed port 80. Once I aligned these correctly, the health checks passed and the app loaded successfully."*

----------------------

🎯 **What I Learned / Achieved**

*"This project deepened my understanding of AWS networking — especially how Ingress, ALB target groups, and Route 53 interact."*

*"It also taught me how to troubleshoot real-world Kubernetes deployment issues, like port mismatches and missing target registration."*

*"The final result was a fully working, HTTPS-enabled web app accessible via a custom domain, backed by a Kubernetes cluster on AWS."*