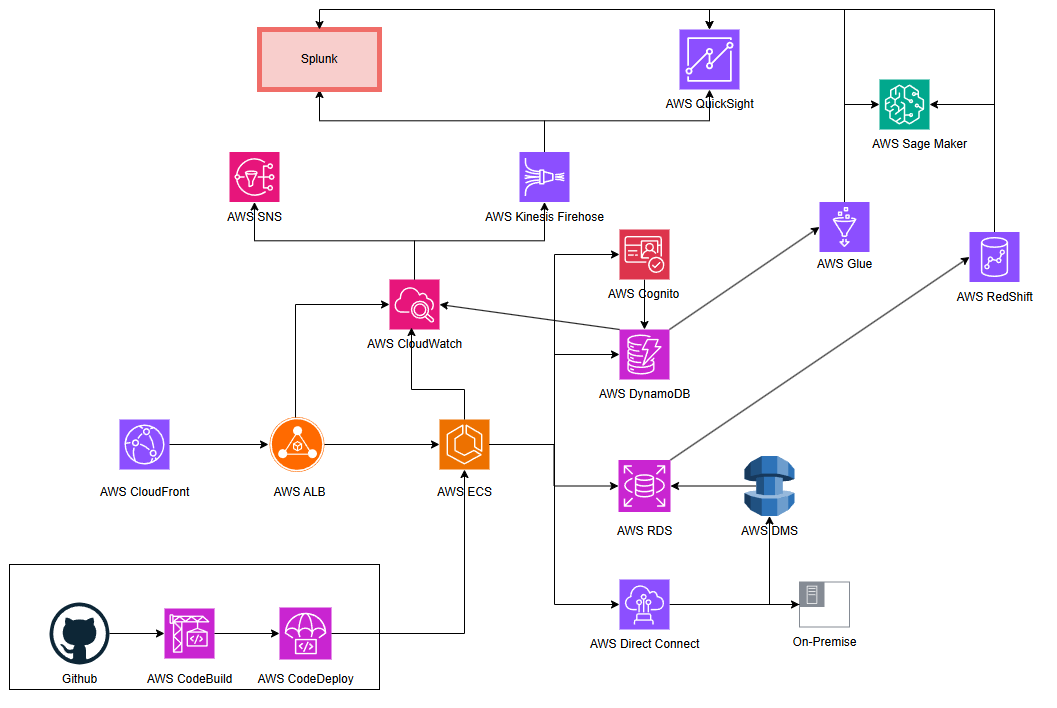
**Telecom Application Modernization and Cloud Migration with AWS**

****

**1. Core Architecture**

**Web Servers Migration**

* **Amazon ECS (Elastic Container Service)** or **Amazon EKS (Elastic Kubernetes Service)** for container orchestration.
* Use **Docker** to containerize existing Java-based web servers and APIs (Web Logic, JDBC APIs).
* Implement **Elastic Load Balancer (ELB)** for distributing traffic across container instances to ensure high availability.

**Databases**

1. **SQL Database**:
   * For optional SQL database migration, use **Amazon RDS (Relational Database Service)** with **Oracle DB**.
   * Utilize **AWS Database Migration Service (DMS)** to replicate the database from on-premise to AWS with zero downtime.
2. **NoSQL Database**:
   * Use **Amazon DynamoDB** for NoSQL data storage.
   * Enable **Global Tables** for multi-region data availability and automatic replication.

**Sensitive Data Handling**

* Retain sensitive data (e.g., customer records) in the on-premise Oracle database.
* Use **AWS Direct Connect** or **AWS VPN** for secure, low-latency connectivity between on-premise and cloud.

**2. Scalability and Performance**

* **Amazon CloudFront**: Content Delivery Network (CDN) to serve static content with low latency globally.
* **AWS Auto Scaling**: Automatically scale ECS/EKS instances based on traffic or CPU usage.

**3. Monitoring and Analytics**

1. **Monitoring**:
   * Use **Amazon CloudWatch** for application and infrastructure monitoring.
   * Set up **CloudWatch Alarms** for alerts and automated responses.
2. **Data Analytics**:
   * Stream data from AWS services to Splunk using **Kinesis Data Firehose** or **AWS Lambda**.
   * Use **Amazon Redshift** for performing advanced analytics on large datasets.

**4. Microservices Architecture**

* Decompose the application into microservices:
  + **Authentication Service**: Manage user access and sensitive data.
  + **Data Processing Service**: Handle customer and product data.
  + **Notification Service**: Manage notifications.

**5. CI/CD and Development Automation**

* **CodePipeline**: Automate the CI/CD pipeline.
* **CodeBuild**: Build and test applications.
* **CodeDeploy**: Deploy to ECS/EKS with no downtime.

**6**. **Amazon SageMaker**:

* Redshift data is exported to Amazon SageMaker for building and training machine learning models (e.g., predictive analytics, churn prediction).

**7**. **AWS Glue:**

* Redshift works with AWS Glue for ETL (Extract, Transform, Load) jobs.
* Glue transforms raw data into an analytics-friendly format before loading it into Redshift.

1. Migrate web servers using ECS/EKS with Docker and secure database handling via RDS, DynamoDB, and on-premise Oracle with Direct Connect or VPN.
2. Ensure scalability with CloudFront and Auto Scaling, and enable analytics with CloudWatch, Redshift, Glue, and Kinesis.
3. Implement microservices, automate CI/CD pipelines, and leverage SageMaker for predictive analytics and machine learning.