Microservices-based architecture design

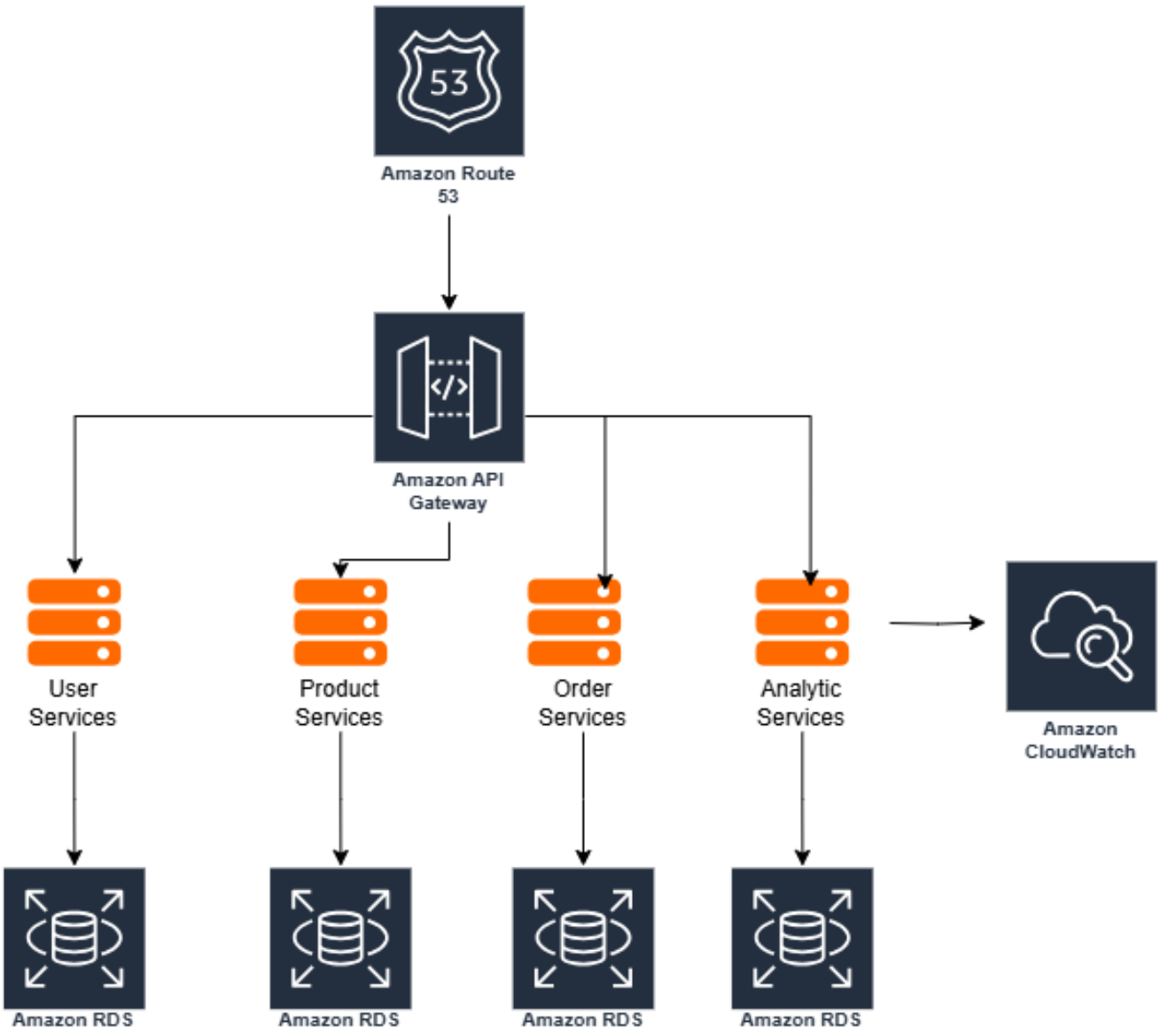
Migrating monolithic architecture to microservices architecture needs careful planning and structured architecture design. At high level, it's essential to understand current monolith architecture including identifying all modules and services, understanding database schema, identifying interdependencies between different services, understanding how loggings data stored, etc.

After that, I will start to define microservice domains and database schema for each microservices. I might need to split monolithic DB into several smaller schemas. Select AWS services like ECS, EKS or Lambda for hosting services.

Next, it's important to set up correct cloud infrastructure, including VPC, subnets, security groups, monitoring and logging, etc.

In terms of migration orders, starting with low impact and low risk services, and creating a new codebase for microservices. Creating a new CI/CD pipeline to deploy code to different environment (Dev, QA, Staging) and setting up API gateway to route traffic to different microservices. Testing API gateway and microservices, until it's stable and reliable then gradually disable monolith endpoints.

Here is the high-level architecture diagram.


Amazon Route 53
Amazon API Gateway
User Services
Product Services
Order Services
Analytic Services
Amazon CloudWatch
Amazon RDS
Amazon RDS
Amazon RDS
Amazon RDS