Task:
Design the backend architecture for an online marketplace platform. This platform should allow users to browse products, make purchases, and manage their orders. Your job is to outline a microservices-based architecture that supports scalability and fault tolerance.

Services:
User Service (handles user accounts, authentication, etc.).
- Database: SQL - to allow inter-user relatability, ACID transactions (account updates), and more straightforward user base management.
- Scaling: Ideally, the user DB would be a single instance (with backups and replication), so horizontal scaling is likely unnecessary. If this includes session data, however, autoscaling may be required for high traffic spikes.

Product Service (manages the product catalog).
- Database: SQL - to simplify product categorization and promote inventory consistency.
- Scaling: The product catalog should not need to be scaled horizontally, but would be replicated and potentially region-based.

Order Service (handles order creation, updates, and tracking).
- Database: noSQL - to allow high availability, large volumes, and simplified data flow.
- Scaling: This service could be scaled horizontally by increasing (autoscaling) the number of servers running the service. It may also require a message queue based on the volume of data.

CI/CD:
During the standard development cycle, a package can be regularly built and tested in a Dev/QA environment (e.g. using Helm). Once ready for promotion to Production, the Docker package can be staged and deployed with a Kubernetes CD tool (e.g. ArgoCD). Using a manager allows tiered rollouts, one-click rollbacks, and integrated cluster monitoring.
The infrastructure can be built on any cloud platform, but AWS likely has the most support currently. EC2s can be used to host the service applications with RDS and DynamoDB databases for long-term data storage. If required, Kafka can be used as a message broker for the order service.

Architecture: