Deployment of Order & Products microservices example with Docker Compose + Gateway Endpoints: GET /users?email= -> Get user through api-gateway: email depends on: GET /users/{id} -> Get a user through id product-service: GET /users/all -> Get all users condition: service started POST /users -> Create a new user order-service: GET /users/{id}/products -> Get the products condition: service started collected by the user user-service: PUT /users/{id} -> Update user info condition: service started POST /users/{id}/products/{skuCode} -> Add image: matthivr1987/api-gateway product to a user ports: GET /products?skuCode= -> Get a product - 8084:8084 based on skuCode environment: GET /products/all -> Get all products PRODUCT SERVICE BASEURL: product-service:8080 external port: auto-generated GET /orders -> Get all orders ORDER SERVICE BASEURL: order-service:8081 ports: 8084:8084 POST /orders -> Create a new order USER SERVICE BASEURL: user-service:8083 api-gateway api-gateway ports: 8081:8081 order-service service volume ports: 8083:8083 POD container user-service order-service user-service ports: ports: 8082:8082 8080:8080 ports: product-service inventory-service 3308:3306 ports: mysql-user inventory-3306:3306 productservice service mysql-order mysql:latest ports: ports: 3307:3306 27017:27017 mysql:latest mysql-inventory mongo-products /var/lib/mysql ─/var/lib/mysql /var/lib/mysql -/data/db mongo:latest mysql:latest mysql_inventory data mongodb product data mysql user data mysql order data