

Online Shopping Order Processing System (@ MyGreatCompany)

In the digital era, e-commerce platforms are the backbone of modern shopping experiences, enabling customers to browse, purchase, and receive products without visiting physical stores. An **Online Shopping Order Processing System** plays a critical role in this ecosystem by managing the lifecycle of customer orders, ensuring efficient and error-free handling from placement to delivery.

An **Online Shopping Order Processing System** automates and streamlines the workflow of handling customer orders for e-commerce platforms. This system ensures the accurate and timely execution of tasks from order placement to delivery.

This system is essential for businesses operating in competitive markets, where customer satisfaction, operational efficiency, and scalability directly impact revenue and growth.

Below are the key reasons why such a system is indispensable:

Benefits of the System

1. Improved Efficiency

- Automation of order processing reduces human intervention and errors, ensuring faster order fulfillment.

2. Real-Time Tracking

- Customers and administrators can monitor the status of orders, from placement to delivery, enhancing transparency.

3. Scalability

- The system can handle high order volumes during peak seasons (e.g., Black Friday or holiday sales) without degrading performance.

4. Enhanced Customer Experience

- Features like instant notifications, order tracking, and easy returns build trust and satisfaction among customers.

5. Cost Savings

- By automating repetitive tasks (e.g., inventory checks, label generation), businesses can save on operational costs.

6. Data Insights

- Analytics on order patterns, customer preferences, and shipping performance help businesses make data-driven decisions.

7. Seamless Integration

- Integration with third-party payment gateways, logistics providers, and inventory management tools ensures a smooth workflow.

System Components

To support the business needs, the Infrastructure and application architects have designed a potential architecture for the system. This is presented in Figure 1 and the main components are:

1. Frontend:

- User-facing web/mobile application for browsing products, placing orders, and tracking shipments are supported by a stack of **3 web servers** behind a **reverse proxy which also acts as a load balancer**.
- These web servers should be running **Nginx** and the reverse proxy is supported by **Traefik**.
- All of these components run on **Docker on-prem**.

2. Backend Services:

- **Order Management System (OMS)**: Manages order workflows and is based on an event streaming system (**Event Hub**). Because the frequency of orders are of high magnitude, and the need to store/archive the data (for auditing purposes) for at least 10 years, the data is sent to a **Cloud infrastructure**, in this case, in the **Azure Platform**.
- At the same time of being stored in the **Cloud**, the data is streamed back in real-time to a **On-Prem system** where the data is fully processed by an application running in a server using **NodeJS**, and stored in a database for operational purposes and short term activities.

3. Database:

- The data is locally stored **On-prem** in a MySQL database.

Assignment Goals / Deliverables

1. Your task is to prepare the code to deploy the infrastructure needed to implement the **Order Management System (OMS)**, and the automation/orchestration mechanisms to deploy the application code once the servers are up and running.
2. Do so, by using a mix of tools and components such as:
 - **Docker, Docker Compose and Traefik**
 - **Azure Cloud Platform and Azure Event Hubs**
 - **Terraform, Ansible and Azure Cli**
3. Build an **AI Agent** using learned tools in this class to query the **Orders Database (MySQL)** in natural language. The data in the database can be simulated.
4. Deliver all your code in a **Github Repository** and share the link with the Professor of the Class.

Delivery date: January 14th, 2026 18:00h

Groups: Maximum 2 people.

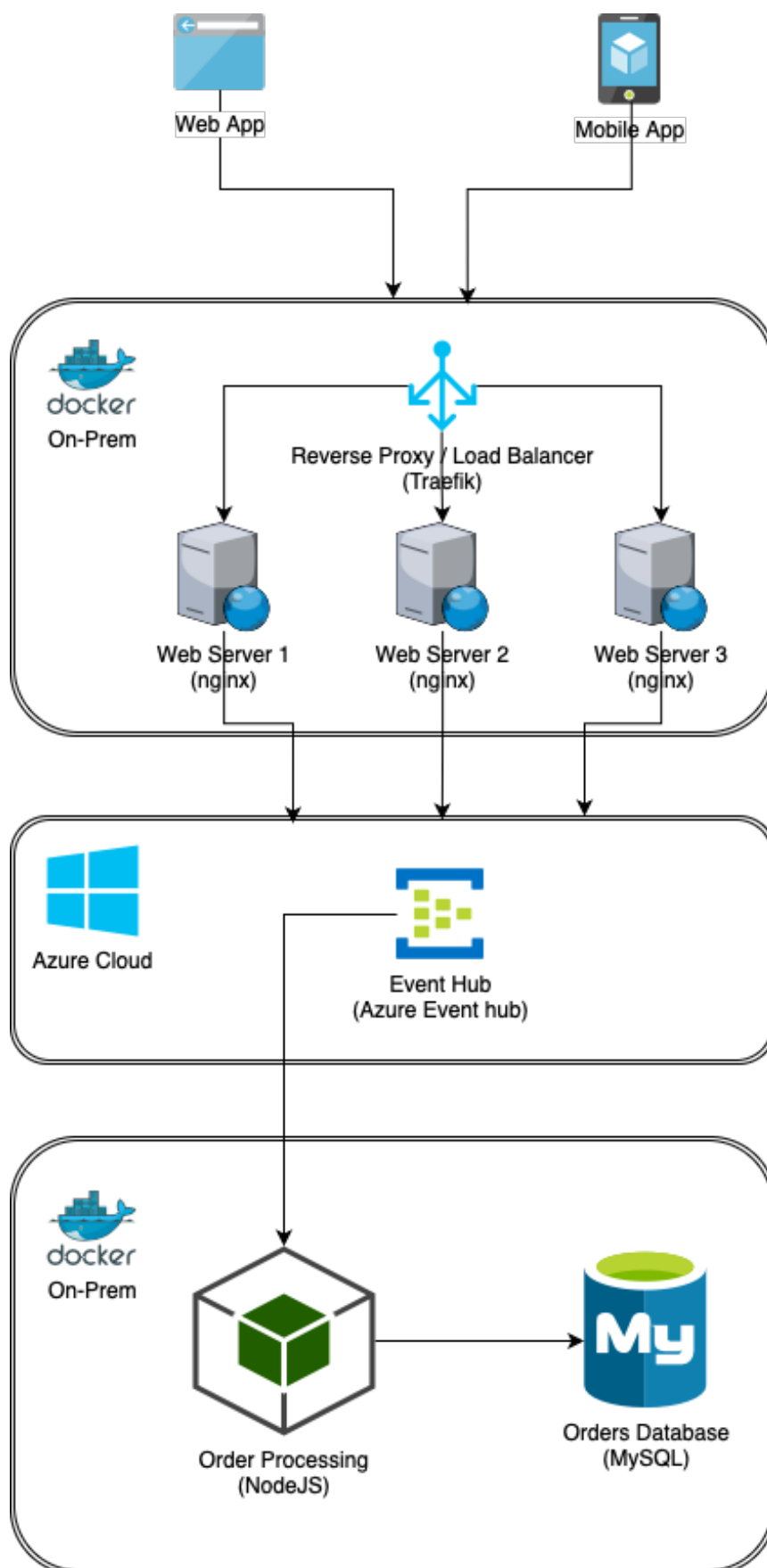


Figure 1. Order Management System Architecture (Example)