

Module 2.8 Assignment

By Chua Lai Chwang of NTU CE7 on 15 Aug 2024

About the Project

- To develop a cloud-native, B2C e-commerce software platform for an online retail business selling men/women clothings, accessories, shoes, etc.
 - Located in Singapore with markets in East Asia and Latin Americas.
 - With a current website traffic 340 mil monthly visitors.
- The implementation goals of the software platform are as follows:
 - Provide both 1-Click Buying and Same-Day Delivery services for greater customer satisfaction.
 - Implement an AI-based customer product recommendation system with closed-loop feedback that offers customers the right products at the right time, right place with the right deals.
 - Offer a loyalty-based, rebate system that encourages first-time and ongoing customer purchases.
 - Support high resilience, availability and scalability to ensure the best-in-class uptime of '24x7 shopping'.

The Right Methodology for the Project

- **Agile software development** ('Asd') is best-suited for delivering a highly responsive, customer-focused, and resilient e-commerce platform because:
 - **Rapid Iteration and Flexibility:** Asd's iterative approach enables the quick development and refinement of features like 1-Click Buying and Same-Day Delivery. The flexibility of Asd ensures that these services can be continually optimized based on customer feedback and evolving market demands, enhancing overall customer satisfaction.
 - **Customer-Centric Development:** Asd emphasizes continuous stakeholder feedback, which is crucial for implementing an AI-based customer product recommendation system. This system, with its closed-loop feedback mechanism, can be iteratively improved to ensure customers receive the right products at the right time, in the right place, with the right deals, thereby increasing relevance and customer engagement.

The Right Methodology for the Project

- **Incremental Delivery:** Asd's focus on delivering small, functional increments allows for the early rollout of essential features, such as a loyalty-based rebate system. This approach encourages both first-time and repeat purchases by gradually enhancing the platform's capabilities while ensuring a swift time-to-market.
- **Collaboration and Communication:** The Asd framework fosters close collaboration among developers, AI specialists, and business stakeholders, which is critical for the seamless integration of features like high-resilience and scalable infrastructure. This ensures the platform can support 24x7 shopping with best-in-class uptime, providing a reliable and consistent shopping experience.

The Right Methodology for the Project

- **Continuous Improvement:** Asd's emphasis on retrospectives and iterative refinement helps in continuously enhancing the platform's resilience, availability, and scalability. This ensures the platform remains robust, capable of handling high traffic, and providing uninterrupted services.

The Right Software Deployment for the Project

- Basing on the implementation goals, **Continuous Deployment (CD)** combined with **Blue-Green Deployment** is the most suitable software deployment methodology because:

1. **Continuous Deployment (CD):**

- **Rapid Feature Releases:** CD allows for the automatic deployment of code changes to production as soon as they pass automated testing. This is critical for maintaining and enhancing features like 1-Click Buying, Same-Day Delivery, and the AI-based recommendation system, enabling swift updates and continual improvements.
- **Frequent Updates:** CD supports the iterative nature of Agile, ensuring that updates to the loyalty-based rebate system or AI algorithms can be deployed frequently without manual intervention, keeping the platform up-to-date and responsive to customer needs.
- **Reduced Risk:** By deploying small, incremental changes continuously, CD reduces the risk associated with large-scale deployments, ensuring that any issues can be quickly identified and resolved with minimal impact on the platform's availability.

The Right Software Deployment for the Project

2. Blue-Green Deployment:

- **Zero Downtime:** Blue-Green Deployment ensures that your platform can achieve the best-in-class uptime required for 24x7 shopping. By maintaining two identical production environments (blue and green), traffic can be switched seamlessly between them during updates, allowing for new features or updates to be deployed without affecting the live system.
- **Easy Rollback:** If an issue arises during deployment, Blue-Green Deployment allows for an instant rollback to the previous stable environment, ensuring high resilience and maintaining customer trust.
- **Testing in Production:** This methodology allows for the testing of new features in a production-like environment (the blue or green instance) before they are fully rolled out, ensuring that the AI recommendations, 1-Click Buying, and other critical features work as intended under real-world conditions.

CI/CD Pipeline Tools to use for the Project

- It is recommended as follows:
 - **Source Control:** AWS CodeCommit or GitHub.
 - **CI/Build:** Jenkins or AWS CodeBuild.
 - **Testing:** Integrated with Jenkins pipelines or using AWS CodeBuild.
 - **CD/Deployment:** AWS CodeDeploy with Blue-Green deployment strategy.
 - **Infrastructure Management:** Terraform for IaC.
 - **Containerization & Orchestration:** Docker and Kubernetes.
 - **Monitoring & Observability:** Prometheus for metrics and alerting, Grafana for visualization.
 - **Application & Infrastructure Monitoring:** Amazon CloudWatch, integrated with Prometheus and Grafana for a comprehensive monitoring solution.

CI/CD Pipeline Tools to use for the Project

- It is recommended as follows:

1. **AWS CodePipeline:**

- **Purpose:** Manages the complete CI/CD process, automating the build, test, and deployment phases.
- **Integration:** Seamlessly integrates with other AWS services, making it ideal for deploying applications on AWS infrastructure.

2. **AWS CodeBuild:**

- **Purpose:** Builds and tests code continuously as part of the CI/CD pipeline.
- **Scalability:** Provides scalable and managed build service, ensuring that build resources scale according to the workload, which is crucial for maintaining 24x7 shopping uptime.

CI/CD Pipeline Tools to use for the Project

3. **AWS CodeDeploy:**

- **Purpose:** Automates code deployment to a variety of compute services like EC2, Lambda, or on-premises servers.
- **Deployment Strategies:** Supports Blue-Green and Rolling deployments, ensuring zero downtime during updates, which aligns with your high resilience and availability goals.

4. **AWS CodeCommit:**

- **Purpose:** A fully managed source control service that hosts Git repositories.
- **Security:** Provides encryption and fine-grained access control, ensuring secure code management, which is critical for a platform handling financial transactions and customer data.

CI/CD Pipeline Tools to use for the Project

5. Amazon CloudWatch:

- **Purpose:** Provides monitoring and observability, which is essential for tracking the performance of the CI/CD pipeline, as well as the deployed applications.
 - **Integration:** Integrates with AWS CodePipeline for setting up automated alarms and notifications.
-
- And to ensure better cloud-native interoperability and avoid lock-in as much as possible, it is recommended to use some of these open-source, baseline technologies:

CI/CD Pipeline Tools to use for the Project

1. **Jenkins** where possible to supplement and integrate with AWS CodePipeline:

- **Purpose:** A widely-used open-source automation server that supports building, deploying, and automating any project.
- **Plugins:** Offers extensive plugin support, including integration with AWS services, enabling complex workflows and customizations for your CI/CD pipeline.
- **Flexibility:** Ideal if you require more customization and control over the CI/CD pipeline processes.

CI/CD Pipeline Tools to use for the Project

2. Terraform:

- **Purpose:** Used for infrastructure as code (IaC) to manage cloud infrastructure through code.
- **Flexibility:** Enables the definition, provisioning, and scaling of infrastructure, ensuring the platform's scalability and resilience are maintained.

3. Docker & Kubernetes:

- **Docker:** For containerizing applications, ensuring consistent deployment across different environments.
- **Kubernetes:** For orchestrating containerized applications, managing scaling, load balancing, and self-healing, which is crucial for maintaining uptime and scalability.

CI/CD Pipeline Tools to use for the Project

4. Prometheus:

- **Purpose:** Prometheus is an open-source monitoring and alerting tool designed to collect and store metrics as time series data.
- **Relevance to Goals:**
 - **Performance Monitoring:** Prometheus can continuously monitor the health and performance of key features such as 1-Click Buying, Same-Day Delivery, and the AI-based recommendation system, ensuring that these services meet customer satisfaction expectations.
 - **Custom Metrics:** It enables the collection of custom metrics like response times, error rates, and user interactions with loyalty and rebate systems, helping to fine-tune the customer experience.
 - **Alerting for High Availability:** With integrated alerting capabilities, Prometheus ensures that any deviation from desired uptime or performance levels triggers immediate alerts, supporting your goal of 24x7 shopping availability.

CI/CD Pipeline Tools to use for the Project

4. Grafana:

- **Purpose:** Grafana is an open-source platform for visualizing monitoring data.
- **Relevance to Goals:**
 - **Real-Time Visualization:** Grafana provides real-time dashboards that display metrics collected by Prometheus, offering clear visibility into the performance and reliability of the platform's critical components.
 - **Custom Dashboards:** Custom dashboards can be created to monitor specific areas like the AI recommendation system's impact on customer satisfaction, the success of loyalty-based purchases, and the scalability of infrastructure.
 - **Multi-Source Data Integration:** Grafana's ability to aggregate and visualize data from various sources, including Prometheus and AWS CloudWatch, ensures a unified view of the platform's health, directly supporting high resilience and uptime.