Setting Up and Maintaining Containerized Environments

Part 1: Office Roleplay Dialogue

Scenario: A DevOps Engineer, Kenji, is working with his colleague, Priya, to set up and maintain containerized environments using Docker and Kubernetes.

Priya: Kenji, I see you're setting up the new application environment. Are you using **Docker** for this?

Kenji: Yes, I'm containerizing the app with **Docker** and managing deployments with **Kubernetes** for better scalability.

Priya: That makes sense. Are you handling **orchestration** manually, or is Kubernetes automating it?

Kenji: Kubernetes is taking care of the **orchestration**, so we don't have to worry about manually managing containers.

Priya: That's great! Have you pushed the images to the **container registry** yet?

Kenji: Not yet. I'm testing them first, then I'll upload them to our private **container registry** for security.

Priya: Good idea. We should also define the resource limits to optimize the **containers** for performance.

Kenji: Definitely. I'll set CPU and memory limits in the YAML configurations.

Priya: Nice. And do we have an automatic restart policy in case a container fails?

Kenji: Yes, Kubernetes will restart failed **containers** automatically based on our configurations.

Priya: Perfect. That should help keep downtime minimal. Let's do one final test before pushing everything live.

Kenji: Agreed! I'll trigger a deployment and monitor the logs.

Part 2: Comprehension Questions

1. What tool is Kenji using to containerize the application?
(A) VirtualBox
(B) Docker
(C) Apache
(D) Jenkins

2. Why is Kubernetes being used in the setup?
(A) To manually control every container
(B) To manage and automate orchestration
(C) To replace all containerized environments
(D) To increase the number of developers

3. Where will Kenji upload the tested container images?
(A) A cloud storage folder
(B) The company's website
(C) A container registry
(D) A Git repository

4. What will Kubernetes do if a container fails?
(A) Delete the application permanently
(B) Restart the container automatically

(C) Stop all running containers
(D) Notify users through email

Part 3: Key Vocabulary Definitions in Japanese

1. **Docker (ドッカー)** – アプリケーションをコンテナ化するためのプラットフォーム。
2. **Kubernetes (クバネティス)** – コンテナ化されたアプリケーションを管理・自動化するオープンソースのシステム。
3. **Containers (コンテナ)** – ソフトウェアを実行するための軽量な仮想環境。
4. **Orchestration (オーケストレーション)** – コンテナの配置やスケーリングを自動化するプロセス。
5. **Container registry (コンテナレジストリ)** – コンテナイメージを保存・管理するリポジトリ。

Part 4: Questions & Correct Answers

1. **What tool is Kenji using to containerize the application?**
   ✅ (B) Docker
2. **Why is Kubernetes being used in the setup?**
   ✅ (B) To manage and automate orchestration

3. **Where will Kenji upload the tested container images?**

   ✅ (C) A container registry

4. **What will Kubernetes do if a container fails?**

   ✅ (B) Restart the container automatically