### 1. Relation between k8 Pod vs k8 Container?

**Variants:**

* 1:1
* 1:many
* many:1
* **Right answer:**  1:many

### 2. [Open question] Does Kafka guarantee out-of-the-box «exactly once» delivery strategy?

**Right answer:**  no , not out of the box  
**Follow-up question:** How to perform that type of delivery?  
**Right answer:**  mulipule ways 1 idempotent producer (enable.idemptence=true)  
 transactional producer to avoid dublicate right

### 3. [Open question] What are some ways to manage dependencies in a project using Docker?

**Good answers:**

**Dockerfile to define the base path and commands like run copy add   
we can use dcokerCompose file as well , in this file we can define all the depndicer   
port and dependent as wll**

### 4. What is PostgreSQL's default index type?

**Variants:**

* Hashtable
* Binary tree
* There is no default
* GIST

**Right answer:**  Binary tree

### 5. Where would you avoid using a microservice architecture?

**Variants:**

* One-page site
* CRM system
* Script
* Marketplace
* **Right answers:**  one page site ,script

### 6. [Part 1] If object A equals object B, will their hashcodes also be equal?

**Variants:**

* Yes
* No
* Yes, but not in 100% of cases

**Right answer: yes**

[Part 2] If two objects A and B have the same hash code, does it mean they are always equal?  
**Variants:**

* Yes
* No
* Yes, but not in 100% of cases
* **Right answer:**  yes , but not 100% cases

### 7. Which Java object type is a hashcode?

**Variants:**

* String
* Long
* int
* Object

**Right answer:**  int

### 8. Which of the following statements about volatile in Java are correct? (Select all that apply)

**Options:**  
A) volatile guarantees atomicity of operations.  
B) volatile prevents caching of the variable in thread registers.  
C) volatile ensures visibility of variable changes across all threads.  
D) volatile can replace synchronized in any case.

**Answer:** C,B

### 9. What ORM-related problems are you aware of?

**Good answers:**  N+1 problem , Lazy loading, difficult optimised queries , tight coupling scehma

### 10. What is the N+1 problem in the context of ORM (e.g., Hibernate)? How can it be avoided?

**Options:**  
A) It is a problem where one query is executed to fetch N entities and N additional queries are executed to fetch related data. Solution: Use JOIN FETCH or @EntityGraph.  
B) It is a problem where N queries are executed to fetch a single entity. Solution: Use caching.  
C) It is a problem where too many insert queries are executed. Solution: Use batch inserts.

**Answer:**  A