Graded Quiz: Introduction to Microservices
Graded Quiz • 30 min

← **Back**

🌐 English ⌄

Due Aug 27,
11:59 PM PKT

✅ **Congratulations! You passed!**

| Grade received **90%** | Latest Submission Grade **90%** | To pass **70%** or higher |

Go to next item

✦ Summarize ⋮ ✕

---

1. Which phase of the Twelve-Factor Methodology contains Port binding?   1 / 1 point

- ◉ Deploy
- ◯ Configure
- ◯ Operate
- ◯ Code

✅ **Correct**
Correct. Deploy phase of the delivery lifecycle contains: Config, Backing services, Processes, and Port binding.

2. What makes an application concurrent?   1 / 1 point

- ◉ It should be stateless
- ◯ It should contain the same backend services across environments
- ◯ It should have a minimal process start time
- ◯ It should store config in environment variables

✅ **Correct**
Correct. If processes are stateless and share nothing, an application can start additional processes to scale horizontally and handle the additional workload without creating interdependencies among processes.

3. In a microservice architecture, you should   0 / 1 point

- ◉ Deploy all services together
- ◯ Use a different technology stack for each service
- ◯ Create interdependency among services
- ◯ Scale all services together

❌ **Incorrect**
Incorrect. Review the What are Microservices video.

4. Select the correct answer that completes this sentence: 1 / 1 point

Monolith application has _____ components.

- ◯ Independent
- ◯ Reusable
- ◯ Scalable
- ◉ Interconnected

✅ **Correct**
Correct. A monolithic application has all or most of its functionality within a single process. Making them tightly connected and dependent on each other.

5. Which component of services design in an SOA architecture defines how the service provider and service consumer should interact?   1 / 1 point

- ◯ Implementation
- ◯ Own storage
- ◉ Contract
- ◯ Interface

✅ **Correct**
Correct. A contract defines how the service provider and service consumer should interact.

6. What is the purpose of Backend For Frontend?

○ Refactoring in stages

○ Loading one interface that never reloads

● Customized user experiences

○ Service discovery

1 / 1 point

✓ **Correct**
Correct. The BFF pattern allows developers to create and support one backend type per user interface using the best options for that interface rather than trying to support a generic backend that works with any interface but may negatively impact frontend performance.

7. What is the purpose of the Strangler pattern?

○ Simplifies the front-end experience

● Supports refactoring in stages

○ Combine functional domains

○ Helps with third-party API integrations

1 / 1 point

✓ **Correct**
Correct. With the Strangler pattern, you use the structure of a web application to split an application into multiple functional domains and replace those domains with a new microservices-based implementation for one domain at a time.

8. Larger services should be broken into smaller services when:

○ Automation is required

○ Refactoring is required

● Common data model becomes overly complex

○ Functional reusability is required

1 / 1 point

✓ **Correct**
Correct. If your data model becomes too broad, it is time to split the service and functionally segregate them.

9. What is the requirement of Microservices?

● Independent storage

○ Its own implementation of auxiliary services like loggings, security, throttling, and so on

○ Manual deployment process

○ Coarse-grained functionality

1 / 1 point

✓ **Correct**
Correct. SOA and microservices are both interested in building reusable individual components that can be consumed by other applications. But each microservice requires independent data storage, such as the bounded context.

10. What is the requirement of a development and production environment?

○ Different dependencies

● Similar backing services across the environments.

○ Different source control

○ Embedded access details in code

1 / 1 point

✓ **Correct**
Correct. If you use a MySQL database in production, you should use the same version of MySQL database in your development environments.