**Fundamentals of Microservices**

**Post Test**

1. Which of the following is the benefit of microservices based application?
2. Simple to develop
3. Simple to scale horizontally
4. Easy to deploy
5. **Easily adopt new technologies**
6. A single-page application is an app that works inside a browser and does not require page reloading during use.
7. **True**
8. False
9. ...... is a collection of small functional modules which can be independently deployable, scalable, target specific business goals, and communicate with each other over standard protocols.
10. Monolithic Architecture
11. Service Oriented Architecture
12. **Microservices Architecture**
13. The domain logic, subdomains, bounded contexts, context maps, domain models, and ubiquitous language terminologies are associated with .....
14. Object oriented design
15. **Domain driven design**
16. Procedural Programming
17. None of the above
18. A developer needs to fire multiple queries that join data. Which of the following approach to choose?
19. Database per service
20. **Shared database**
21. Command Query Responsibility Segregation
22. None of the above
23. ..... as an abstraction over REST-based calls, by which microservices can communicate with each other, but developers don't have to bother about REST internal details.
24. RestTemplate
25. **Feign Client**
26. Both of the above
27. None of the above
28. When making a request to a service, the client makes a request via a router that runs at a well known location. The router queries a service registry, which might be built into the router, and forwards the request to an available service instance. Which of the following are you using?
29. **Server side discovery**
30. Client side discovery
31. Self registration
32. None of the above
33. While using microservice, it might lead to resource exhaustion so calling the service unable to handle other requests. This may cause the failure of one service which can potentially cascade to other services throughout the application. Which of the following will can be used to prevent this issue?
    1. Domain driven design
    2. Creating a service per host
    3. Using API Gateway
    4. **Using circuit breaker**
34. Choose the annotation to enable registry with the Discovery server
    1. @EnableDiscovery
    2. @EnableDiscoveryServer
    3. **@EnableDiscoveryClient**
    4. None of the above
35. Choose the property to be configured in properties file to set active profile,
36. **spring.profiles.active**
37. spring.active
38. spring.profiles
39. None of the above
40. The Load Balancer supports ………….
41. Server side discovery
42. **Client side discovery**
43. Both of the above
44. None of the above
45. If you do not want the gateway to be enabled due to any reason, which of the below property in your application.properties file you will configure?
46. spring.gateway.cloud.enabled=false
47. spring.gateway.enabled=false
48. **spring.cloud.gateway.enabled=false**
49. It’s not possible to disable gatway api.
50. Choose that applies so that a service must be provided with configuration data that tells it how to connect to the external or 3rd party services.
51. Configure Gateway API
52. Configure Hystrics
53. **Externalized the configuration with central repository**
54. None of the above
55. RestTemplate and FeignClient supports,
56. **Synchronous communication**
57. Asynchronous communication
58. Location independency is enabled by which of the following,
59. Service per host
60. Using centralized repository
61. **Eureka naming server**
62. None of the above