1. What is the Service API in Kubernetes primarily designed for?  
— A) Managing container images  
— B) Exposing groups of Pods over a network  
— C) Configuring network security policies  
— D) Orchestrating load balancing mechanisms  
— \*\*Correct Answer: B) Exposing groups of Pods over a network\*\*

2. How does the Service abstraction in Kubernetes help in decoupling applications?  
— A) By defining logical sets of endpoints  
— B) By managing container replicas  
— C) By enforcing security policies  
— D) By providing direct access to Pods  
— \*\*Correct Answer: A) By defining logical sets of endpoints\*\*

3. In Kubernetes, what determines the set of Pods targeted by a Service?  
— A) Replica count  
— B) Selector definition  
— C) Pod IP addresses  
— D) Container image name  
— \*\*Correct Answer: B) Selector definition\*\*

4. What is the purpose of an Ingress in Kubernetes?  
— A) To manage containerized databases  
— B) To consolidate routing rules for web traffic  
— C) To monitor resource usage  
— D) To control access to the Kubernetes API  
— \*\*Correct Answer: B) To consolidate routing rules for web traffic\*\*

5. How does an Ingress differ from a Service in Kubernetes?  
— A) Ingress provides access to internal services while Service handles external traffic.  
— B) Ingress consolidates routing rules, while Service manages endpoint exposure.  
— C) Ingress is used for stateful workloads, while Service is for stateless workloads.  
— D) Ingress manages container lifecycle, while Service manages networking policies.  
— \*\*Correct Answer: B) Ingress consolidates routing rules, while Service manages endpoint exposure.\*\*

6. What additional capabilities does the Gateway API for Kubernetes provide beyond Ingress and Service?  
— A) Enhanced security features  
— B) Load balancing mechanisms  
— C) Custom routing configurations  
— D) Network service access configuration  
— \*\*Correct Answer: D) Network service access configuration\*\*

7. How are Ingress and Gateway related in Kubernetes?  
— A) Ingress is a subset of Gateway functionality.  
— B) Gateway is an alternative to Ingress.  
— C) Ingress and Gateway are complementary for managing network access.  
— D) Ingress is deprecated in favor of Gateway.  
— \*\*Correct Answer: C) Ingress and Gateway are complementary for managing network access.\*\*

8. Which Kubernetes resource acts as the entry point for routing rules consolidation?  
— A) Ingress  
— B) Service  
— C) Gateway  
— D) Pod  
— \*\*Correct Answer: A) Ingress\*\*

9. What type of workload would typically benefit from using Ingress in Kubernetes?  
— A) Stateful applications  
— B) Stateless microservices  
— C) Batch processing jobs  
— D) Database management systems  
— \*\*Correct Answer: B) Stateless microservices\*\*

10. In Kubernetes, how are routing rules consolidated using Ingress?  
— A) By exposing multiple components behind a single listener  
— B) By defining logical sets of endpoints  
— C) By enforcing network security policies  
— D) By managing containerized databases  
— \*\*Correct Answer: A) By exposing multiple components behind a single listener\*\*

11. What role does the Gateway API play in Kubernetes networking?  
— A) It replaces Ingress for routing configurations.  
— B) It provides additional capabilities beyond Ingress and Service.  
— C) It handles container image management.  
— D) It controls access to the Kubernetes API server.  
— \*\*Correct Answer: B) It provides additional capabilities beyond Ingress and Service.\*\*

12. How does the Service API in Kubernetes contribute to application scalability?  
— A) By managing container lifecycles  
— B) By dynamically adjusting replica counts  
— C) By defining logical sets of endpoints  
— D) By enforcing network security policies  
— \*\*Correct Answer: C) By defining logical sets of endpoints\*\*

13. Which Kubernetes resource is responsible for managing access to network services within the cluster?  
— A) Gateway  
— B) Ingress  
— C) Service  
— D) Pod  
— \*\*Correct Answer: C) Service\*\*

14. What is the primary advantage of using a Service in Kubernetes?  
— A) Simplifying container image management  
— B) Ensuring high availability of Pods  
— C) Exposing groups of Pods over a network  
— D) Enforcing security policies within Pods  
— \*\*Correct Answer: C) Exposing groups of Pods over a network\*\*

15. How do frontend clients interact with backend services in Kubernetes, facilitated by Services?  
— A) Clients directly communicate with individual Pods.  
— B) Clients communicate with a logical set of endpoints defined by Services.  
— C) Clients access Pods through a centralized routing system.  
— D) Clients must manually track the set of backend Pods.  
— \*\*Correct Answer: B) Clients communicate with a logical set of endpoints defined by Services.\*\*

16. Which Kubernetes resource helps consolidate routing rules into a single resource for managing web traffic?  
— A) Gateway  
— B) Ingress  
— C) Service  
— D) Pod  
— \*\*Correct Answer: B) Ingress\*\*

17. In Kubernetes, what determines which Pods are targeted by a Service?  
— A) Pod IP addresses  
— B) Replica count  
— C) Selector definition  
— D) Container image name  
— \*\*Correct Answer: C) Selector definition\*\*

18. What is the primary function of an Ingress in Kubernetes?  
— A) Managing container replicas  
— B) Defining routing rules for web traffic  
— C) Exposing groups of Pods over a network  
— D) Enforcing security policies within Pods  
— \*\*Correct Answer: B) Defining routing rules for web traffic\*\*

19. How does the Gateway API in Kubernetes extend networking capabilities beyond Ingress and Service?  
— A) By consolidating routing rules  
— B) By providing additional security features  
— C) By dynamically adjusting Pod replicas  
— D) By configuring access to network services  
— \*\*Correct Answer: D) By configuring access to network services\*\*

20. What is the main benefit of using Ingress for managing web traffic in Kubernetes?  
— A) Simplifying container image management  
— B) Enabling centralized routing rule configuration  
— C) Ensuring high availability of Pods  
— D) Enforcing security policies within Pods  
— \*\*Correct Answer: B) Enabling centralized routing rule configuration\*\*

21. How can you utilize Kubernetes APIs for service discovery in your application?  
— A) By querying the API server for matching Pods  
— B) By querying the API server for matching EndpointSlices  
— C) By directly accessing backend Pods  
— D) By configuring network load balancers manually  
— \*\*Correct Answer: B) By querying the API server for matching EndpointSlices\*\*

22. What does Kubernetes update for a Service whenever the set of Pods changes?  
— A) Service definitions  
— B) Replica counts  
— C) EndpointSlices  
— D) Pod IP addresses  
— \*\*Correct Answer: C) EndpointSlices\*\*

23. For non-native applications in Kubernetes, what options are available to place a network port or load balancer between the application and backend Pods?  
— A) Direct Pod access  
— B) Service discovery mechanisms  
— C) Querying the API server  
— D) Configuring Ingress resources  
— \*\*Correct Answer: D) Configuring Ingress resources\*\*

24. How can Kubernetes assist non-native applications in connecting to backend Pods?  
— A) By exposing Pod IP addresses directly  
— B) By providing EndpointSlices for each Service  
— C) By offering manual configuration of network ports  
— D) By offering service discovery mechanisms  
— \*\*Correct Answer: D) By offering service discovery mechanisms\*\*

25. What is the purpose of EndpointSlices in Kubernetes service discovery?  
— A) To define routing rules  
— B) To manage container replicas  
— C) To track changes in the set of Pods for a Service  
— D) To manage network security policies  
— \*\*Correct Answer: C) To track changes in the set of Pods for a Service\*\*

26. How do non-native applications typically interact with backend Pods in Kubernetes?  
— A) Through direct Pod access  
— B) Through querying the API server for Pods  
— C) Through service discovery mechanisms  
— D) Through manual configuration of load balancers  
— \*\*Correct Answer: C) Through service discovery mechanisms\*\*

27. What enables Kubernetes workloads to find the target they want to connect to?  
— A) Direct querying of Pod IP addresses  
— B) Manual configuration of network ports  
— C) Service discovery mechanisms  
— D) Static routing rules  
— \*\*Correct Answer: C) Service discovery mechanisms\*\*

28. In Kubernetes, what updates when the set of Pods in a Service changes?  
— A) Pod IP addresses  
— B) Service definitions  
— C) Replica counts  
— D) EndpointSlices  
— \*\*Correct Answer: D) EndpointSlices\*\*

29. How does Kubernetes assist non-native applications in connecting to backend Pods?  
— A) By exposing Pod IP addresses directly  
— B) By providing direct access to backend Pods  
— C) By offering service discovery mechanisms  
— D) By configuring network load balancers automatically  
— \*\*Correct Answer: C) By offering service discovery mechanisms\*\*

30. What options does Kubernetes provide for non-native applications to establish connections with backend Pods?  
— A) Direct API server access  
— B) Manual load balancer configuration  
— C) Service discovery mechanisms  
— D) Pod IP address listing  
— \*\*Correct Answer: C) Service discovery mechanisms\*\*

Part 3: — Creating a Service

31. What type of object is a Service in Kubernetes?  
— A) Configuration  
— B) Pod  
— C) Service  
— D) ConfigMap  
— \*\*Correct Answer: C) Service\*\*

32. How can you interact with Service definitions in Kubernetes?  
— A) Directly accessing the API server  
— B) Using kubectl or similar tools to make API calls  
— C) Modifying Pod configurations  
— D) Editing YAML files manually  
— \*\*Correct Answer: B) Using kubectl or similar tools to make API calls\*\*

33. What tool is commonly used to manage Service definitions in Kubernetes?  
— A) Docker  
— B) Minikube  
— C) kubectl  
— D) Helm  
— \*\*Correct Answer: C) kubectl\*\*

34. What type of port can a Service in Kubernetes map to a targetPort?  
— A) HTTP  
— B) UDP  
— C) TCP  
— D) SSH  
— \*\*Correct Answer: C) TCP\*\*

35. When creating a Service, what does Kubernetes assign to it?  
— A) Pod replicas  
— B) Replica counts  
— C) A cluster IP address  
— D) Pod IP addresses  
— \*\*Correct Answer: C) A cluster IP address\*\*

36. How does Kubernetes handle updates to the set of EndpointSlices for a Service?  
— A) Manually by administrators  
— B) Automatically by the controller for the Service  
— C) Through direct API calls  
— D) By restarting Pods  
— \*\*Correct Answer: B) Automatically by the controller for the Service\*\*

37. What is the default type of IP address assigned to a Service in Kubernetes?  
— A) External IP  
— B) Internal IP  
— C) Cluster IP  
— D) Virtual IP  
— \*\*Correct Answer: C) Cluster IP\*\*

38. Which mechanism does Kubernetes use for virtual IP address management in Services?  
— A) Virtual IPs and Service Proxies  
— B) Kubernetes Load Balancer  
— C) NodePort Service  
— D) Ingress Controller  
— \*\*Correct Answer: A) Virtual IPs and Service Proxies\*\*

39. What naming convention must the name of a Service object adhere to?  
— A) RFC 1918  
— B) RFC 3339  
— C) RFC 1035  
— D) RFC 5321  
— \*\*Correct Answer: C) RFC 1035\*\*

40. What is the default value of the targetPort field in a Service definition?  
— A) The same as the port field  
— B) Randomly assigned  
— C) 8080  
— D) 80  
— \*\*Correct Answer: A) The same as the port field\*\*

41. How does Kubernetes continuously update the set of EndpointSlices for a Service?  
— A) Through manual intervention  
— B) By restarting Pods  
— C) Automatically by the controller for the Service  
— D) By reloading configuration files  
— \*\*Correct Answer: C) Automatically by the controller for the Service\*\*

42. What does a Service object in Kubernetes typically target?  
— A) Pod replicas  
— B) Node IP addresses  
— C) Pod IP addresses  
— D) Cluster IP addresses  
— \*\*Correct Answer: C) Pod IP addresses\*\*

43. What is the purpose of a Service controller in Kubernetes?  
— A) To manage Pod lifecycles  
— B) To handle API requests  
— C) To continuously update EndpointSlices for a Service  
— D) To manage networking policies  
— \*\*Correct Answer: C) To continuously update EndpointSlices for a Service\*\*

44. Which type of Kubernetes resource must have a valid RFC 1035 label name?  
— A) Deployment  
— B) StatefulSet  
— C) Service  
— D) PersistentVolume  
— \*\*Correct Answer: C) Service\*\*

45. How does Kubernetes handle incoming traffic for a Service?  
— A) By directly routing traffic to individual Pods  
— B) By using a load balancer  
— C) By applying network policies  
— D) By inspecting Pod configurations  
— \*\*Correct Answer: B) By using a load balancer\*\*

46. What does a Service map incoming ports to on its target Pods by default?  
— A) 443  
— B) 80  
— C) 9376  
— D) Randomly assigned  
— \*\*Correct Answer: B) 80\*\*

47. What does Kubernetes provide to abstract away the complexity of managing network traffic to Pods?  
— A) Virtual IP addresses  
— B) Ingress resources  
— C) Services  
— D) Network policies  
— \*\*Correct Answer: C) Services\*\*

48. How does Kubernetes enable virtual IP address management for Services?  
— A) Through manual configuration  
— B) By using DNS resolution  
— C) Through the Virtual IPs and Service Proxies mechanism  
— D) By assigning static IP addresses  
— \*\*Correct Answer: C) Through the Virtual IPs and Service Proxies mechanism\*\*

49. What is the primary function of a Service object in Kubernetes?  
— A) Managing container replicas  
— B) Exposing Pods over a network  
— C) Defining routing rules  
— D) Configuring network security policies  
— \*\*Correct Answer: B) Exposing Pods over a network\*\*

50. What is the recommended approach to modifying Service definitions in Kubernetes?  
— A) Editing YAML files directly  
— B) Using a graphical user interface  
— C) Making API calls programmatically  
— D) Using kubectl or similar tools  
— \*\*Correct Answer: D) Using kubectl or similar tools\*\*