

QUESTIONS & ANSWERS

Kill your exam at first Attempt



KCNA Dumps
KCNA Braindumps
KCNA Real Questions
KCNA Practice Test
KCNA dumps free



Linux-Foundation

KCNA

Kubernetes and Cloud Native Associate (KCNA)

<http://killexams.com/pass4sure/exam-detail/KCNA>



Question: 30

Which project in this list is a leading project in the observability space?

- A. Jaeger
- B. Vitess
- C. Argo
- D. Kubernetes

Answer: A

Explanation:

<https://github.com/cncf/landscape#trail-map>

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape landscape.cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider cncf.io/kcsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

landscape.cncf.io

v20200501



1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer cncf.io/ckd
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application



5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing



7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performance distributed transactional key-value store written in Rust.



9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRI-O.



2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLops



4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger



6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.



8. STREAMING & MESSAGING

When you need higher performance than JSON-RPC, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. CloudEvents is a specification for describing event data in common ways.



10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.



Question: 31

What is scheduling in Kubernetes

- Determining when to execute a cron-job
- Assigning pods to nodes
- Joining a new nodes to the clusters
- Setting a time for automated tasks

Answer: B

Explanation:

<https://kubernetes.io/docs/concepts/scheduling-eviction/>

Scheduling

- [Kubernetes Scheduler](#)
- [Assigning Pods to Nodes](#)
- [Pod Overhead](#)
- [Taints and Tolerations](#)
- [Scheduling Framework](#)
- [Scheduler Performance Tuning](#)
- [Resource Bin Packing for Extended Resources](#)

Graphical user interface, application

Description automatically generated

Question: 32

Which of the following are characteristics of Statefulsets?

- A. Ordered, graceful deployment and scaling
- B. Creates replica sets
- C. Uses headless services

Answer: A

Explanation:

<https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/>

Using StatefulSets

StatefulSets are valuable for applications that require one or more of the following.

- Stable, unique network identifiers.
- Stable, persistent storage.
- Ordered, graceful deployment and scaling.
- Ordered, automated rolling updates.

Graphical user interface, text, application, email

Description automatically generated

Question: 33

Which control plane component is responsible for scheduling pods?

- A. kube-proxy
- B. kube scheduler
- C. kubelet
- D. kube api-server

Answer: B

Explanation:

<https://kubernetes.io/docs/concepts/overview/components/>

kube-scheduler

Control plane component that watches for newly created Pods with no assigned node, and selects a node for them to run on.

Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.

Graphical user interface, text, application

Description automatically generated

Question: 34

What are the benefits of Kubernetes federation?

- A. Avoids scalability limits on pods and nodes
- B. Creates highly available clusters in different regions

C. Low latency

Answer: A,B,C

Question: 35

Which prometheus metric type represents a single number value that can increase and decrease over time?

- A. Gauge
- B. Histogram
- C. Summary
- D. Counter

Answer: A

Explanation:

https://prometheus.io/docs/concepts/metric_types/#gauge

Gauge 

A *gauge* is a metric that represents a single numerical value that can arbitrarily go up and down.

Gauges are typically used for measured values like temperatures or current memory usage, but also "counts" that can go up and down, like the number of concurrent requests.

Graphical user interface, text

Description automatically generated

Question: 36

What is OPA ?

- A. Open Permission Agent
- B. Online Policy Audit
- C. Open Policy Agent
- D. Offline Policy Accessor

Answer: C

Explanation:

<https://www.cncf.io/projects/open-policy-agent-opa/>

Open Policy Agent (OPA)



Open Policy Agent

An open source, general-purpose policy engine.

Open Policy Agent (OPA) was accepted to CNCF on **March 29, 2018** and is at the **Graduated** project maturity level.

Graphical user interface, text, application, email

Description automatically generated

Question: 37

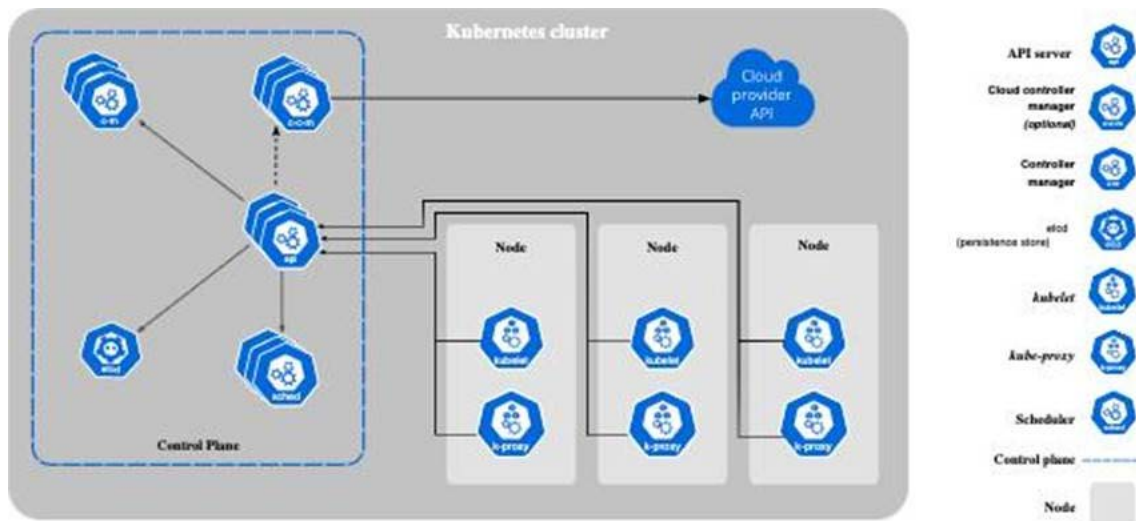
Which of the following is not the part of Kubernetes Control Plane?

- A. kube scheduler
- B. etcd (pronounce: esty-d)
- C. kube api-server
- D. kube-proxy

Answer: D

Explanation:

<https://kubernetes.io/docs/concepts/overview/components/>



Graphical user interface, diagram, application

Description automatically generated

Question: 38

Which access control component of Kubernetes is responsible for authorization and decides what requestor is allowed to do?

- A. Service Account
- B. Role-based access control 'RBAC'
- C. Deployment

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/access-authn-authz/authorization/>

Authorization Modes

The Kubernetes API server may authorize a request using one of several authorization modes:

- **Node** - A special-purpose authorization mode that grants permissions to kubelets based on the pods they are scheduled to run. To learn more about using the Node authorization mode, see [Node Authorization](#).
- **ABAC** - Attribute-based access control (ABAC) defines an access control paradigm whereby access rights are granted to users through the use of policies which combine attributes together. The policies can use any type of attributes (user attributes, resource attributes, object, environment attributes, etc). To learn more about using the ABAC mode, see [ABAC Mode](#).
- **RBAC** - Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise. In this context, access is the ability of an individual user to perform a specific task, such as view, create, or modify a file. To learn more about using the RBAC mode, see [RBAC Mode](#)
 - When specified RBAC (Role-Based Access Control) uses the `rbac.authorization.k8s.io` API group to drive authorization decisions, allowing admins to dynamically configure permission policies through the Kubernetes API.
 - To enable RBAC, start the apiserver with `--authorization-mode=RBAC`.

Text, letter

Description automatically generated

Question: 39

Which role is responsible of creating service level indicator 'SLI', service level objective 'SLO', & Service Level Agreements 'SLA'?

- A. Site reliability engineer 'SRE'
- B. DevOps
- C. GitOps
- D. Security and compliance engineer
- E. Developer

Answer: A

Explanation:

<https://www.atlassian.com/incident-management/kpis/sla-vs-slo-vs-sli>

How does this impact SREs?

For those of you following Google's model and using [Site Reliability Engineering \(SRE\) teams](#) to bridge the gap between development and operations, SLAs, SLOs, and SLIs are foundational to success. SLAs help teams set boundaries and error budgets. SLOs help prioritize work. And SLIs tell SREs when they need to freeze all launches to save an endangered error budget—and when they can loosen up the reins.

Text

Description automatically generated

Question: 40

Which is NOT a use case for the Kubernetes dashboard?

- A. Troubleshooting any issues with applications
- B. Managing running applications
- C. Installing new Kubernetes cluster
- D. Managing the entire Kubernetes cluster

Answer: C

For More exams visit <https://killexams.com/vendors-exam-list>



Kill your exam at First Attempt....Guaranteed!