

Linux-Foundation

Exam Questions KCNA

Kubernetes and Cloud Native Associate (KCNA)



NEW QUESTION 1

What tool allows us to build useful visual representations of prometheus data?

- A. Grafana
- B. kubectl
- C. Distributed system tracing
- D. Rook
- E. Kibana

Answer: A

Explanation:

<https://prometheus.io/>
Graphical user interface, text, application Description automatically generated



Great visualization

Prometheus has multiple modes for visualizing data: a built-in expression browser, Grafana integration, and a console template language.

NEW QUESTION 2

What is the primary interface for Kubernetes cluster?

- A. Kubernetes Api
- B. Kubelet
- C. YAML
- D. Control Plane
- E. JSON

Answer: A

Explanation:

<https://kubernetes.io/docs/concepts/overview/components/#kube-apiserver>
A screenshot of a computer Description automatically generated with medium confidence

kube-apiserver

The API server is a component of the Kubernetes control plane that exposes the Kubernetes API. The API server is the front end for the Kubernetes control plane.

The main implementation of a Kubernetes API server is `kube-apiserver`. `kube-apiserver` is designed to scale horizontally—that is, it scales by deploying more instances. You can run several instances of `kube-apiserver` and balance traffic between those instances.

NEW QUESTION 3

Which kubernetes resource type allows defining which pods are isolated when it comes to network-ing?

- A. Network policy
- B. Domain Name System 'DNS'
- C. Role Binding
- D. Service

Answer: A

Explanation:

<https://kubernetes.io/docs/concepts/services-networking/network-policies/#the-two-sorts-of-pod-isolation> Text, letter Description automatically generated

The Two Sorts of Pod Isolation

There are two sorts of isolation for a pod: isolation for egress, and isolation for ingress. They concern what connections may be established. "Isolation" here is not absolute, rather it means "some restrictions apply". The alternative, "non-isolated for \$direction", means that no restrictions apply in the stated direction. The two sorts of isolation (or not) are declared independently, and are both relevant for a connection from one pod to another.

By default, a pod is non-isolated for egress; all outbound connections are allowed. A pod is isolated for egress if there is any NetworkPolicy that both selects the pod and has "Egress" in its `policyTypes`; we say that such a policy applies to the pod for egress. When a pod is isolated for egress, the only allowed connections from the pod are those allowed by the `egress` list of some NetworkPolicy that applies to the pod for egress. The effects of those `egress` lists combine additively.

By default, a pod is non-isolated for ingress; all inbound connections are allowed. A pod is isolated for ingress if there is any NetworkPolicy that both selects the pod and has "Ingress" in its `policyTypes`; we say that such a policy applies to the pod for ingress. When a pod is isolated for ingress, the only allowed connections into the pod are those from the pod's node and those allowed by the `ingress` list of some NetworkPolicy that applies to the pod for ingress. The effects of those `ingress` lists combine additively.

NEW QUESTION 4

What kind of limitation cgroups allows?

- A. Prioritization
- B. Resource limiting
- C. Accounting
- D. None of the options
- E. Control
- F. Server cpu and memory

Answer: ABCE

NEW QUESTION 5

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 (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/kspp

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.

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

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

3. ORCHESTRATION & APPLICATION DEFINITION
 • Kubernetes is the market-leading orchestration solution
 • You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/kubernetes
 • Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

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

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

6. NETWORKING, POLICY, & SECURITY
 To enable more flexible networking, use a CNF-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.

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.

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.

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 CRIO.

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

NEW QUESTION 6

A new Pod is created. Then, the Pod is assigned to a Node. Which Kubernetes component was re-sponsible for determining which Node to assign the Pod to?

- A. kubelet
- B. Scheduler
- C. API Server
- D. Controller manager

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/command-line-tools-reference/kube-scheduler/> Graphical user interface, text, application Description automatically generated

The Kubernetes scheduler is a control plane process which assigns Pods to Nodes. The scheduler determines which Nodes are valid placements for each Pod in the scheduling queue according to constraints and available resources. The scheduler then ranks each valid Node and binds the Pod to a suitable Node. Multiple different schedulers may be used within a cluster; kube-scheduler is the reference implementation. See [scheduling](#) for more information about scheduling and the kube-scheduler component.

```
kube-scheduler [flags]
```

NEW QUESTION 7

Which project is not a dominant CNCF project in the storage landscape?

- A. Envoy

- B. Vitess
- C. Rook
- D. TiKV

Answer: A

Explanation:

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

A picture containing timeline Description automatically generated

CLOUD NATIVE COMPUTING FOUNDATION

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 Solution Provider. cncf.io/csp

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.

l.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 sizable applications and writing future functionality as microservices

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

3. ORCHESTRATION & APPLICATION DEFINITION
 • Kubernetes is the market-leading orchestration solution
 • You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/graduated
 • Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

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

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

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.

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.

8. STREAMING & MESSAGING
 When you need higher performance than JSON-REST, 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.

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 CRIO.

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

NEW QUESTION 8

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 Graphical user interface, text Description automatically generated



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.

NEW QUESTION 9

Which authentication method allows JWTs to authenticate?

- A. OpenId connect
- B. Client 'TLS' certificates
- C. OPA gatekeeper
- D. Anonymous

Answer: A

NEW QUESTION 10

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

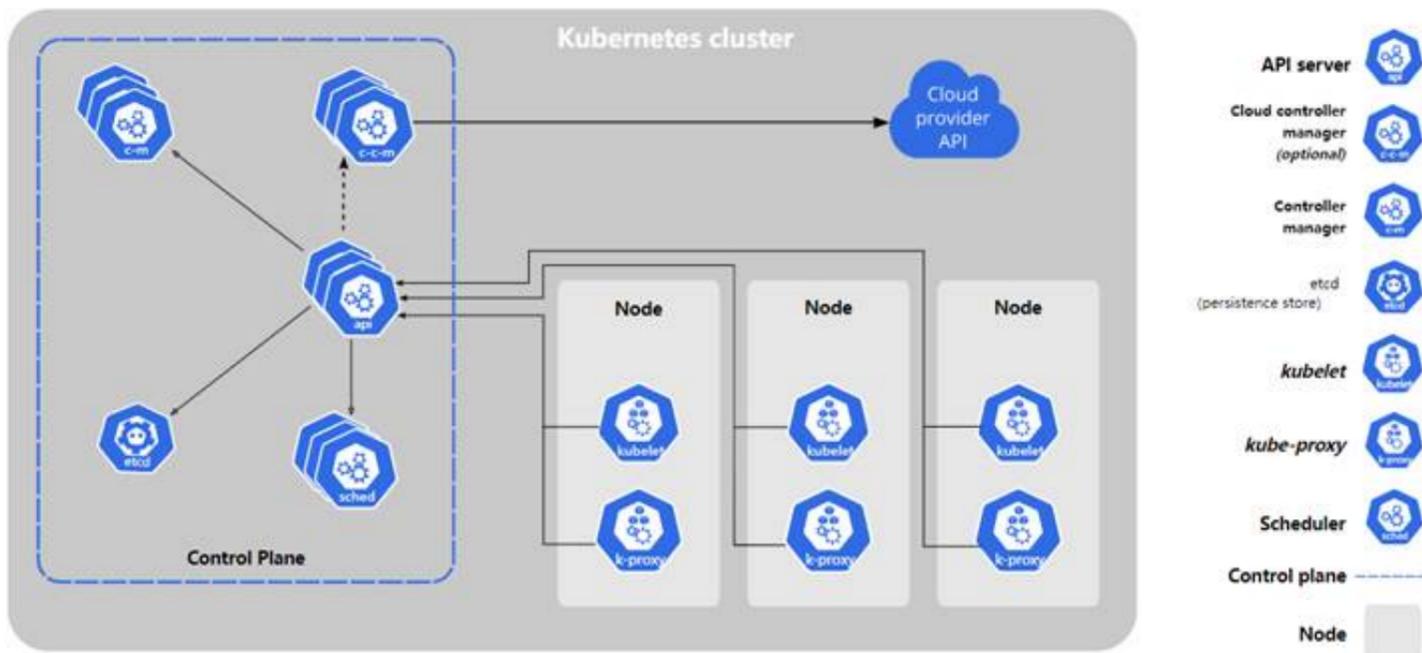
- A. kube scheduler
- B. etcd (pronounce: esty-d)
- C. kube api-server
- D. kube-proxy

Answer: D

Explanation:

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

Graphical user interface, diagram, application Description automatically generated



NEW QUESTION 10

Which style of operations are preferred for kubernetes and cloud-native applications?

- A. Imperative
- B. None of the above
- C. Declarative

Answer: C

Explanation:

<https://kubernetes.io/docs/tasks/manage-kubernetes-objects/declarative-config/#trade-offs>

NEW QUESTION 11

What is Open Container Initiative 'OCI'?

- A. A protocol for communicating with the kubernetes api
- B. The governing body of the Cloud Native Computing Foundation 'CNCF'
- C. An open standard for managing service mesh in kubernetes
- D. An organization that creates open standards for containers

Answer: D

Explanation:

<https://opencontainers.org/>

Text Description automatically generated

Open Container Initiative

The **Open Container Initiative** is an open governance structure for the express purpose of creating open industry standards around container formats and runtimes.

Established in June 2015 by Docker and other leaders in the container industry, the OCI currently contains three specifications: the Runtime Specification (runtime-spec), the Image Specification (image-spec) and the Distribution Specification (distribution-spec). The Runtime Specification outlines how to run a "filesystem bundle" that is unpacked on disk. At a high-level an OCI implementation would download an OCI Image then unpack that image into an OCI Runtime filesystem bundle. At this point the OCI Runtime Bundle would be run by an OCI Runtime.

NEW QUESTION 14

Which of the following command is used to get detailed information about the pod?

- A. kubectl info
- B. kubectl get
- C. kubectl describe
- D. kubectl explain

Answer: C

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#describe> Graphical user interface, application Description automatically generated



The screenshot shows three examples of the 'kubectl describe' command:

- Describe a pod:** `kubectl describe pods/nginx`
- Describe a pod identified by type and name in "pod.json":** `kubectl describe -f pod.json`
- Describe all pods:** `kubectl describe pods`

NEW QUESTION 15

What command can you use to get documentation about a resource type from the command line?

- A. kubectl api-resources
- B. kubectl explain
- C. kubectl get
- D. kubectl get-resource

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#explain> Graphical user interface, text, application, email Description automatically generated

explain

List the fields for supported resources.

This command describes the fields associated with each supported API resource. Fields are identified via a simple JSONPath identifier:

```
<type>.<fieldName>[.<fieldName>]
```

Add the `--recursive` flag to display all of the fields at once without descriptions. Information about each field is retrieved from the server in OpenAPI format.

Use `"kubectl api-resources"` for a complete list of supported resources.

Usage

```
$ kubectl explain RESOURCE
```



NEW QUESTION 16

How would you return all the pod data in the json format using kubectl command?

- A. `kubectl get pods -o json`
- B. `kubectl get pods --all-namespaces`
- C. `kubectl get pods -o wide`
- D. `kubectl get pods -o jsonpath`

Answer: A

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#get>

NEW QUESTION 17

How should folks new to the cloud native ecosystem, go about learning the different aspects of the ecosystem?

- A. by signing up the CNCF slack
- B. by reading the Kubernetes documentation
- C. by looking at the cloud native landscape
- D. by looking at the cloud native trail-map

Answer: D

Explanation:

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

NEW QUESTION 20

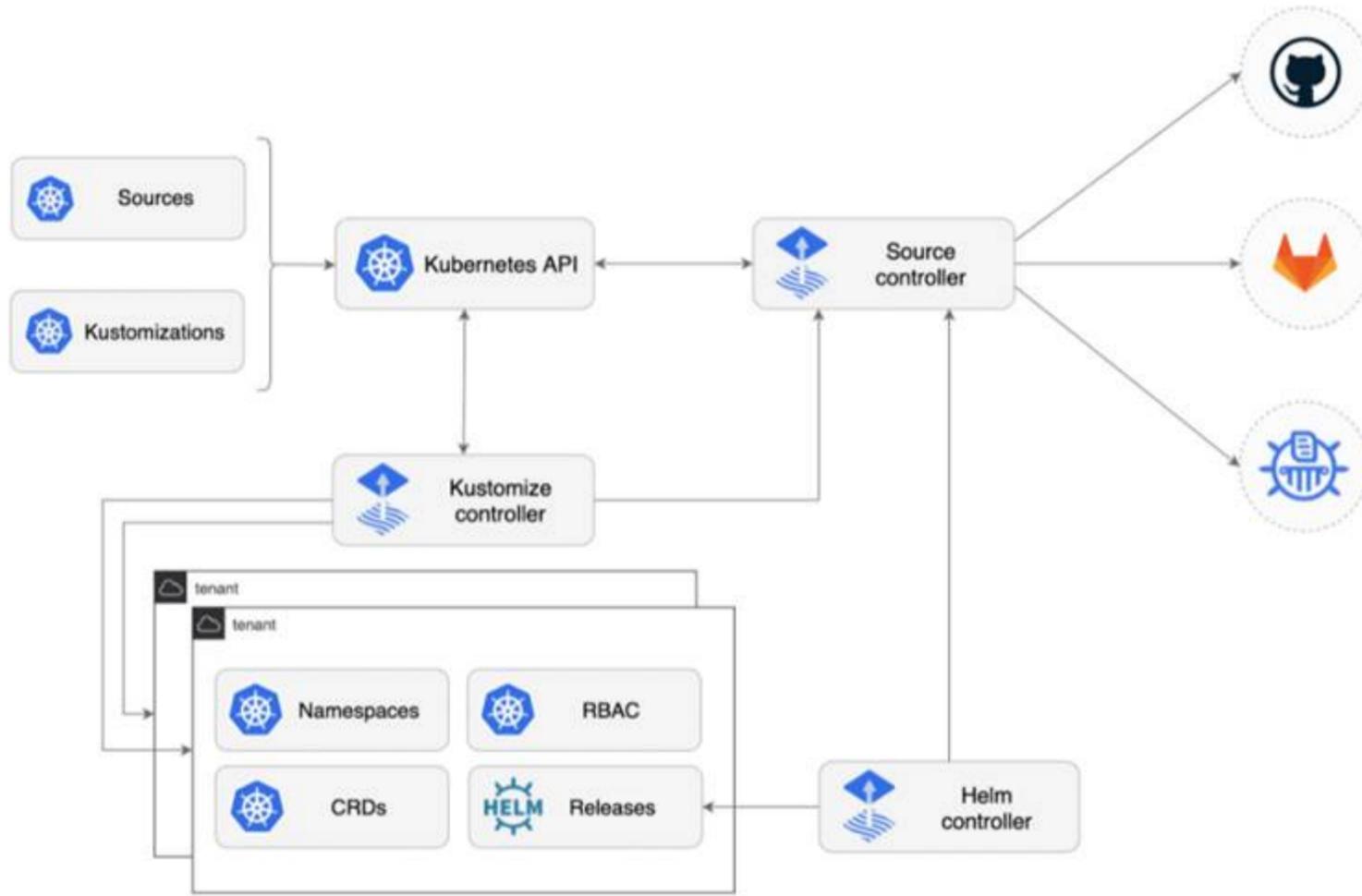
What do GitOps tools do in kubernetes?

- A. They allow us to make changes to a kubernetes cluster using a Git repository
- B. They manage the source code of kubernetes itself
- C. They allow us to store software code in Git
- D. They allows us to store container images in repositories

Answer: A

Explanation:

<https://fluxcd.io/docs/components/>
 Diagram Description automatically generated



NEW QUESTION 22

Flux is built using which toolkit?

- A. CI/CD
- B. DevSecOps
- C. GitOps
- D. DevOps

Answer: C

Explanation:

<https://fluxcd.io/>
 Graphical user interface, text, application Description automatically generated

Flux provides GitOps for both apps and infrastructure

Flux and Flagger deploy apps with canaries, feature flags, and A/B rollouts. Flux can also manage any Kubernetes resource. Infrastructure and workload dependency management is built in.

Just push to Git and Flux does the rest

Flux enables application deployment (CD) and (with the help of Flagger) progressive delivery (PD) through automatic reconciliation. Flux can even push back to Git for you with automated container image updates to Git (image scanning and patching).

NEW QUESTION 25

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> Text Description automatically generated

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.

NEW QUESTION 26

What makes cloud native technology so important?

- A. It makes data centric
- B. It strengthens team
- C. It removes roadblocks to innovation
- D. It helps gather software requirements
- E. It makes operational centric

Answer: C

Explanation:

<https://github.com/cncf/foundation/blob/main/charter.md>

Graphical user interface, text, application Description automatically generated

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.

NEW QUESTION 28

What is the name for the tool that manages communication between pods, injects a sidecar proxy container into each pod and directs network traffic through the proxy container?

- A. namespace
- B. Deployment
- C. Network policy
- D. Service mesh
- E. Service

Answer: D

NEW QUESTION 32

Which is not a service type in Kubernetes?

- A. ClusterIP
- B. NodePort
- C. Ingress
- D. LoadBalancer
- E. ExternalName

Answer: C

Explanation:

<https://kubernetes.io/docs/tutorials/kubernetes-basics/expose/expose-intro/>

without a Service. Services allow your applications to receive traffic. Services can be exposed in different ways by specifying a type in the ServiceSpec:

- *ClusterIP* (default) - Exposes the Service on an internal IP in the cluster. This type makes the Service only reachable from within the cluster.
- *NodePort* - Exposes the Service on the same port of each selected Node in the cluster using NAT. Makes a Service accessible from outside the cluster using `<NodeIP>:<NodePort>` . Superset of ClusterIP.
- *LoadBalancer* - Creates an external load balancer in the current cloud (if supported) and assigns a fixed, external IP to the Service. Superset of NodePort.
- *ExternalName* - Maps the Service to the contents of the `externalName` field (e.g. `foo.bar.example.com`), by returning a CNAME record with its value. No proxying of any kind is set up. This type requires v1.7 or higher of `kube-dns` , or CoreDNS version 0.0.8 or higher.

More information about the different types of Services can be found in the [Using Source IP](#) tutorial. Also see [Connecting Applications with Services](#).

Text Description automatically generated

NEW QUESTION 37

Which of the following factors does scheduling take into account when selecting a Node?

- A. How many replicas there are in a Deployment
- B. Services
- C. Resource requirements
- D. The number of existing Pods on a Node

Answer: C

Explanation:

Scheduling takes resource requirements into account in the form of resource requests.

NEW QUESTION 39

Which of the following is not the required field to describe Kubernetes objects?

- A. metadata
- B. apiVersion
- C. Kind
- D. Container
- E. spec

Answer: D

Explanation:

<https://kubernetes.io/docs/concepts/overview/working-with-objects/kubernetes-objects/> Graphical user interface, text, application Description automatically generated

Required Fields [↔](#)

In the `.yaml` file for the Kubernetes object you want to create, you'll need to set values for the following fields:

- `apiVersion` - Which version of the Kubernetes API you're using to create this object
- `kind` - What kind of object you want to create
- `metadata` - Data that helps uniquely identify the object, including a `name` string, `UID`, and optional `namespace`
- `spec` - What state you desire for the object

The precise format of the object `spec` is different for every Kubernetes object, and contains nested fields specific to that object. The [Kubernetes API Reference](#) can help you find the `spec` format for all of the objects you can create using Kubernetes.

NEW QUESTION 44

What command to view the kube config?

- A. `kubectl view config`
- B. `kubectl config view`
- C. `kubectl get kubeconfig`

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-em-view-em-> Graphical user interface, text, application Description automatically generated

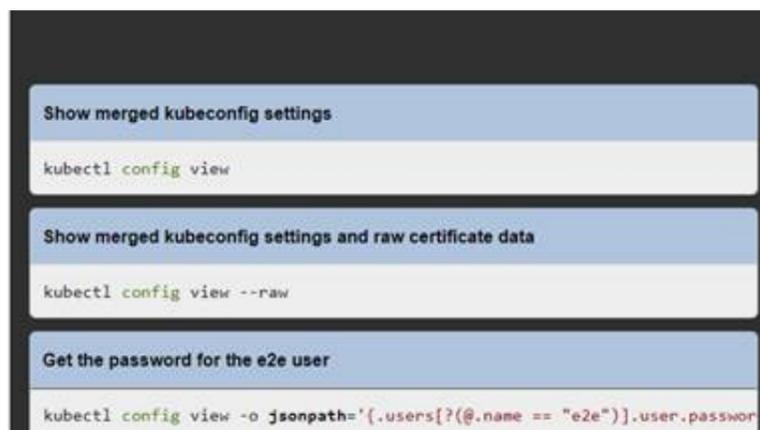
view

Display merged kubeconfig settings or a specified kubeconfig file.

You can use `--output jsonpath={...}` to extract specific values using a jsonpath expression.

Usage

```
$ kubectl config view
```



NEW QUESTION 46

The Kubernetes API provides an interface for storing objects. Which of the following describes the type of objects stored by the Kubernetes API?

- A. Containers
- B. REST
- C. YAML
- D. ETCD

Answer: B

Explanation:

Kubernetes objects are RESTful objects.

NEW QUESTION 49

What kubectl command is used to edit a resource on the server?

- A. `kubectl resource modify`
- B. `kubectl update resource`
- C. `kubectl edit`
- D. `kubectl resource edit`

Answer: C

Explanation:

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#edit> Graphical user interface, text, application, email Description automatically generated

edit

Edit a resource from the default editor.

The edit command allows you to directly edit any API resource you can retrieve via the command-line tools. It will open the editor defined by your KUBE_EDITOR, or EDITOR environment variables, or fall back to 'vi' for Linux or 'notepad' for Windows. You can edit multiple objects, although changes are applied one at a time. The command accepts file names as well as command-line arguments, although the files you point to must be previously saved versions of resources.

Editing is done with the API version used to fetch the resource. To edit using a specific API version, fully-qualify the resource, version, and group.

The default format is YAML. To edit in JSON, specify "-o json".

The flag --windows-line-endings can be used to force Windows line endings, otherwise the default for your operating system will be used.

In the event an error occurs while updating, a temporary file will be created on disk that contains your unapplied changes. The most common error when updating a resource is another editor changing the resource on the server. When this occurs, you will have to apply your changes to the newer version of the resource, or update your temporary saved copy to include the latest resource version.

Edit the service named 'registry'

```
kubectl edit svc/registry
```

Use an alternative editor

```
KUBE_EDITOR="nano" kubectl edit svc/registry
```

Edit the job 'myjob' in JSON using the v1 API format

```
kubectl edit job.v1.batch/myjob -o json
```

Edit the deployment 'mydeployment' in YAML and save the modified config in its annotation

```
kubectl edit deployment/mydeployment -o yaml --save-config
```

Edit the deployment/mydeployment's status subresource

```
kubectl edit deployment mydeployment --subresource='status'
```

NEW QUESTION 51

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

NEW QUESTION 56

Which of the following container runtime is planned to be deprecated in Kubernetes 1.20 and high-er?

- A. cri-o
- B. None of the options
- C. docker
- D. podman
- E. containerd

Answer: C

Explanation:

<https://kubernetes.io/blog/2020/12/02/dont-panic-kubernetes-and-docker/>
 Graphical user interface, text, application, email Description automatically generated

Wednesday, December 02, 2020

Update: *Kubernetes support for Docker via `dockershim` is now removed. For more information, read the [removal FAQ](#). You can also discuss the deprecation via a dedicated [GitHub issue](#).*

Authors: Jorge Castro, Duffie Cooley, Kat Cosgrove, Justin Garrison, Noah Kantrowitz, Bob Killen, Rey Lejano, Dan "POP" Papandrea, Jeffrey Sica, Davanum "Dims" Srinivas

Kubernetes is [deprecating Docker](#) as a container runtime after v1.20.

You do not need to panic. It's not as dramatic as it sounds.

TL;DR Docker as an underlying runtime is being deprecated in favor of runtimes that use the [Container Runtime Interface \(CRI\)](#) created for Kubernetes. Docker-produced images will continue to work in your cluster with all runtimes, as they always have.

NEW QUESTION 61

Which of the following best describes the way kubernetes Role-based access control (RBAC) works?

- A. Kubernetes does not do RBAC

- B. Kubernetes RBAC states which users can perform which actions against which re-source
- C. Kubernetes RBAC lists which operations on which resources are denied to users
- D. Kubernetes RBAC is responsible for authenticating subjects such as users and groups

Answer: B

Explanation:

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

Graphical user interface, text, application, email Description automatically generated

Using RBAC Authorization

Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within your organization.

RBAC authorization uses the `rbac.authorization.k8s.io` API group to drive authorization decisions, allowing you to dynamically configure policies through the Kubernetes API.

To enable RBAC, start the API server with the `--authorization-mode` flag set to a comma-separated list that includes `RBAC`; for example:

```
kube-apiserver --authorization-mode=Example,RBAC --other-options --more-options
```

NEW QUESTION 66

The 4C's of Cloud Native security

- A. Chroot, Compute, Cluster and Container
- B. Cluster, Cloud, Compute, and Containers
- C. Code, Containers, Compute, and Cloud
- D. Cloud, Clusters, Containers, and Code

Answer: D

Explanation:

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

NEW QUESTION 70

Which kubernetes object do deployments use behind the scenes when they need to scale pods?

- A. Horizontal pod autoscaler
- B. ReplicaSets
- C. kubectl
- D. Replication controller

Answer: B

Explanation:

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

Graphical user interface, text, application, email Description automatically generated

ReplicaSet

A ReplicaSet's purpose is to maintain a stable set of replica Pods running at any given time. As such, it is often used to guarantee the availability of a specified number of identical Pods.

NEW QUESTION 71

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/>

Graphical user interface, text, application, email Description automatically generated

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.

NEW QUESTION 72

What is the functionality of the daemon set?

- A. To run a copy of the pod in all the nodes of the cluster
- B. To initialize the pod before starting the main pod
- C. To run a copy of the pod in a single node of the cluster

Answer: A

Explanation:

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

Graphical user interface, text, application Description automatically generated with medium confidence

DaemonSet

A DaemonSet ensures that all (or some) Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created.

Some typical uses of a DaemonSet are:

- running a cluster storage daemon on every node
- running a logs collection daemon on every node
- running a node monitoring daemon on every node

NEW QUESTION 75

Which component of the kubernetes control-plane (master) are all requests to deploy and manage objects posted to?

- A. ETCD
- B. Controller Manager
- C. Kube-proxy
- D. API Server
- E. Kubelet

Answer: D

Explanation:

<https://kubernetes.io/docs/reference/command-line-tools-reference/kube-apiserver/> Graphical user interface, text, application Description automatically generated

Synopsis

The Kubernetes API server validates and configures data for the api objects which include pods, services, replicationcontrollers, and others. The API Server services REST operations and provides the frontend to the cluster's shared state through which all other components interact.

```
kube-apiserver [flags]
```

NEW QUESTION 80

Which command-line tool is used to interact with the Kubernetes cluster?

- A. kube-api
- B. kubectl
- C. kube-scheduler

Answer: B

Explanation:

<https://kubernetes.io/docs/reference/kubectl/>

Graphical user interface, text, application, email Description automatically generated

Command line tool (kubectl)

Kubernetes provides a command line tool for communicating with a Kubernetes cluster's control plane, using the Kubernetes API.

This tool is named `kubectl`.

For configuration, `kubectl` looks for a file named `config` in the `$HOME/.kube` directory. You can specify other `kubeconfig` files by setting the `KUBECONFIG` environment variable or by setting the `--kubeconfig` flag.

This overview covers `kubectl` syntax, describes the command operations, and provides common examples. For details about each command, including all the supported flags and subcommands, see the [kubectl](#) reference documentation.

For installation instructions, see [Installing kubectl](#); for a quick guide, see the [cheat sheet](#). If you're used to using the `docker` command-line tool, [kubectl for Docker Users](#) explains some equivalent commands for Kubernetes.

NEW QUESTION 85

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

KCNA Practice Exam Features:

- * KCNA Questions and Answers Updated Frequently
- * KCNA Practice Questions Verified by Expert Senior Certified Staff
- * KCNA Most Realistic Questions that Guarantee you a Pass on Your First Try
- * KCNA Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

100% Actual & Verified — Instant Download, Please Click
[Order The KCNA Practice Test Here](#)