Topic 1 - Exam A

Question #1

Topic 1

What native runtime is Open Container Initiative (OCI) compliant?

A. runC

B. runV

C. kata-containers

D. gvisor

Correct Answer: *A*

Community vote distribution

A (100%)

Suya 3 weeks, 3 days ago

Selected Answer: A

runC is the native runtime that is Open Container Initiative (OCI) compliant.

upvoted 1 times

JBangura 6 months, 2 weeks ago

Selected Answer: A

A is correct

upvoted 2 times

vishau 11 months, 3 weeks ago

Selected Answer: A

this is correct

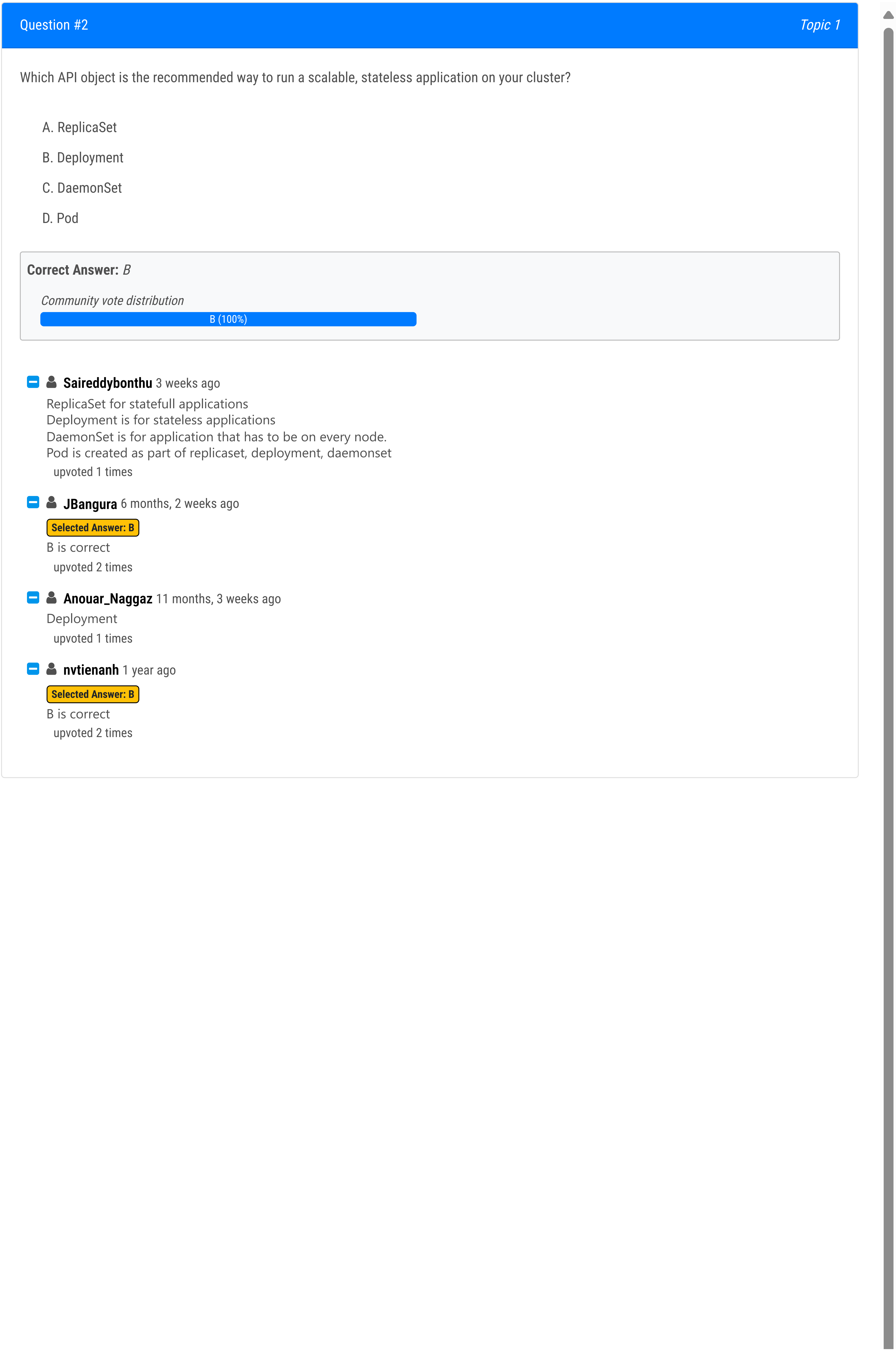
upvoted 2 times

nvtienanh 1 year ago

Selected Answer: A

Answer is A

upvoted 2 times



Question #2

Topic 1

Which API object is the recommended way to run a scalable, stateless application on your cluster?

A. ReplicaSet

B. Deployment

C. DaemonSet

D. Pod

Correct Answer: *B*

Community vote distribution

B (100%)

Saireddybonthu 3 weeks ago

ReplicaSet for statefull applications
Deployment is for stateless applications
DaemonSet is for application that has to be on every node.
Pod is created as part of replicaset, deployment, daemonset

upvoted 1 times

JBangura 6 months, 2 weeks ago

Selected Answer: B

B is correct

upvoted 2 times

Anouar_Naggaz 11 months, 3 weeks ago

Deployment

upvoted 1 times

nvtienanh 1 year ago

Selected Answer: B

B is correct

upvoted 2 times

Question #3 *Topic 1*

A CronJob is scheduled to run by a user every one hour. What happens in the cluster when it's time for this CronJob to run?

A. Kubelet watches API Server for CronJob objects. When it's time for a Job to run, it runs the Pod directly.

B. Kube-scheduler watches API Server for CronJob objects, and this is why it's called kube-scheduler.

C. CronJob controller component creates a Pod and waits until it finishes to run.

D. CronJob controller component creates a Job. Then the Job controller creates a Pod and waits until it finishes to run.

Correct Answer: *D*

Community vote distribution

D (100%)

Saireddybonthu 3 weeks ago

CronJob Controller: The CronJob controller is responsible for managing CronJobs. It schedules Jobs based on the CronJob's schedule (e.g., every hour).

Job Creation: When it's time for a CronJob to run, the CronJob controller creates a Job resource. This Job represents the workload that needs to l executed.

Job Controller: Once the Job is created, the Job controller is responsible for managing this Job. It ensures that the Pods specified by the Job are created and completed successfully.

Pod Execution: The Job controller creates one or more Pods based on the Job specification. These Pods execute the tasks defined by the Job. The Job controller waits until the Pod(s) complete their execution.

upvoted 1 times

JBangura 6 months, 2 weeks ago

Selected Answer: D

Correct Answer: D

upvoted 2 times

Question #4 *Topic 1*

What is the purpose of the kubelet component within a Kubernetes cluster?

A. A dashboard for Kubernetes Clusters that allows management and troubleshooting of applications.

B. A network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept.

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

D. An agent that runs on each node in the cluster. It makes sure that containers are running in a Pod.

Correct Answer: *D*

Community vote distribution

D (100%)

JBangura 6 months, 2 weeks ago

Selected Answer: D

Correct Answer: D

upvoted 2 times

nvtienanh 1 year ago

Selected Answer: D

The given answer is correct

upvoted 2 times

Question #5 *Topic 1*

What is the default value for authorization-mode in Kubernetes API server?

A. --authorization-mode=RBAC

B. --authorization-mode=AlwaysAllow

C. --authorization-mode=AlwaysDeny

D. --authorization-mode=ABAC

Correct Answer: *B*

Community vote distribution

B (100%)

nvtienanh 1 year, 1 month ago

Selected Answer: B

B

upvoted 2 times

Question #6 *Topic 1*

Let's assume that an organization needs to process large amounts of data in bursts, on a cloud-based Kubernetes cluster. For instance: each Monday morning, they need to run a batch of 1000 compute jobs of 1 hour each, and these jobs must be completed by Monday night. What's going to be the most cost-effective method?

A. Run a group of nodes with the exact required size to complete the batch on time, and use a combination of taints, tolerations, and nodeSelectors to reserve these nodes to the batch jobs.

B. Leverage the Kubernetes Cluster Autoscaler to automatically start and stop nodes as they're needed.

C. Commit to a specific level of spending to get discounted prices (with e.g. “reserved instances” or similar mechanisms).

D. Use PriorityClasses so that the weekly batch job gets priority over other workloads running on the cluster, and can be completed on time.

Correct Answer: *B*

pablokoba 4 months, 3 weeks ago

The most cost-effective method would likely be option B, leveraging the Kubernetes Cluster Autoscaler to automatically start and stop nodes as they're needed.

Here's why:

Burst processing workloads, like the one described, benefit from the elasticity provided by cloud-based Kubernetes clusters. With Kubernetes Cluster Autoscaler, you can scale your cluster up when there's a demand for more resources (e.g., Monday mornings when the batch jobs need t run) and scale it down during periods of low demand (e.g., after the batch jobs are completed). This ensures that you're only paying for the resources you actually need, avoiding over-provisioning and reducing costs.

upvoted 4 times

Question #7 *Topic 1*

What is a Kubernetes service with no cluster IP address called?

A. Headless Service

B. Nodeless Service

C. IPLess Service

D. Specless Service

Correct Answer: *A*

EzBL 6 months, 2 weeks ago

A. Headless Service

Here's why:

Headless Service: This type of service doesn't have a cluster IP address assigned. Instead, it maps directly to the pods it manages, allowing communication with individual pods using their unique IP addresses.
Nodeless Service: This term isn't commonly used in Kubernetes. Services can exist and function even without a dedicated node running them.
IPLess Service: While this might seem like a logical term, "Headless Service" is the official designation in Kubernetes.
Specless Service: A service definition requires a specification file (usually YAML). A service without a spec wouldn't be a valid service at all

upvoted 3 times

Question #8 *Topic 1*

CI/CD stands for:

A. Continuous Information / Continuous Development

B. Continuous Integration / Continuous Development

C. Cloud Integration / Cloud Development

D. Continuous Integration / Continuous Deployment

Correct Answer: *D*

Community vote distribution

D (100%)

dadaarce 6 months, 1 week ago

Selected Answer: D

I thought B & D are the same.
Need to take note to answers like this.

Answer is D.

upvoted 2 times

r0xer 7 months ago

Selected Answer: D

The correct answer is D

upvoted 2 times

nvtienanh 1 year ago

Selected Answer: D

D is correct

upvoted 2 times

Question #9 *Topic 1*

What default level of protection is applied to the data in Secrets in the Kubernetes API?

A. The values use AES Symmetric Encryption

B. The values are stored in plain text

C. The values are encoded with SHA256 hashes

D. The values are base64 encoded

Correct Answer: *D*

Community vote distribution

D (100%)

dadaarce 6 months, 1 week ago

Selected Answer: D

Kubernetes Secrets store sensitive information such as passwords, OAuth tokens, and SSH keys.
The data stored in Secrets is base64 encoded by default. However, it's important to note that base64 encoding is not encryption—it's a reversible encoding scheme.
While base64 encoding provides a basic level of obfuscation, it does not provide strong security against unauthorized access.

upvoted 3 times

EzBL 6 months, 2 weeks ago

In Kubernetes, Secrets are stored as base64-encoded strings within etcd, the key-value store used by Kubernetes. Base64 encoding is a method o encoding binary data into ASCII characters, but it is not a form of encryption. Therefore, while base64 encoding obfuscates the data, it does not provide encryption or protection against unauthorized access. It's essential to use additional measures like RBAC (Role-Based Access Control) or encryption mechanisms like encryption at rest to enhance the security of Secrets in Kubernetes.

upvoted 2 times

supersquax 7 months, 2 weeks ago

Selected Answer: D

D is correct

upvoted 2 times

nvtienanh 1 year, 1 month ago

Selected Answer: D

D is correct

upvoted 2 times

eduarte 1 year, 1 month ago

Selected Answer: D

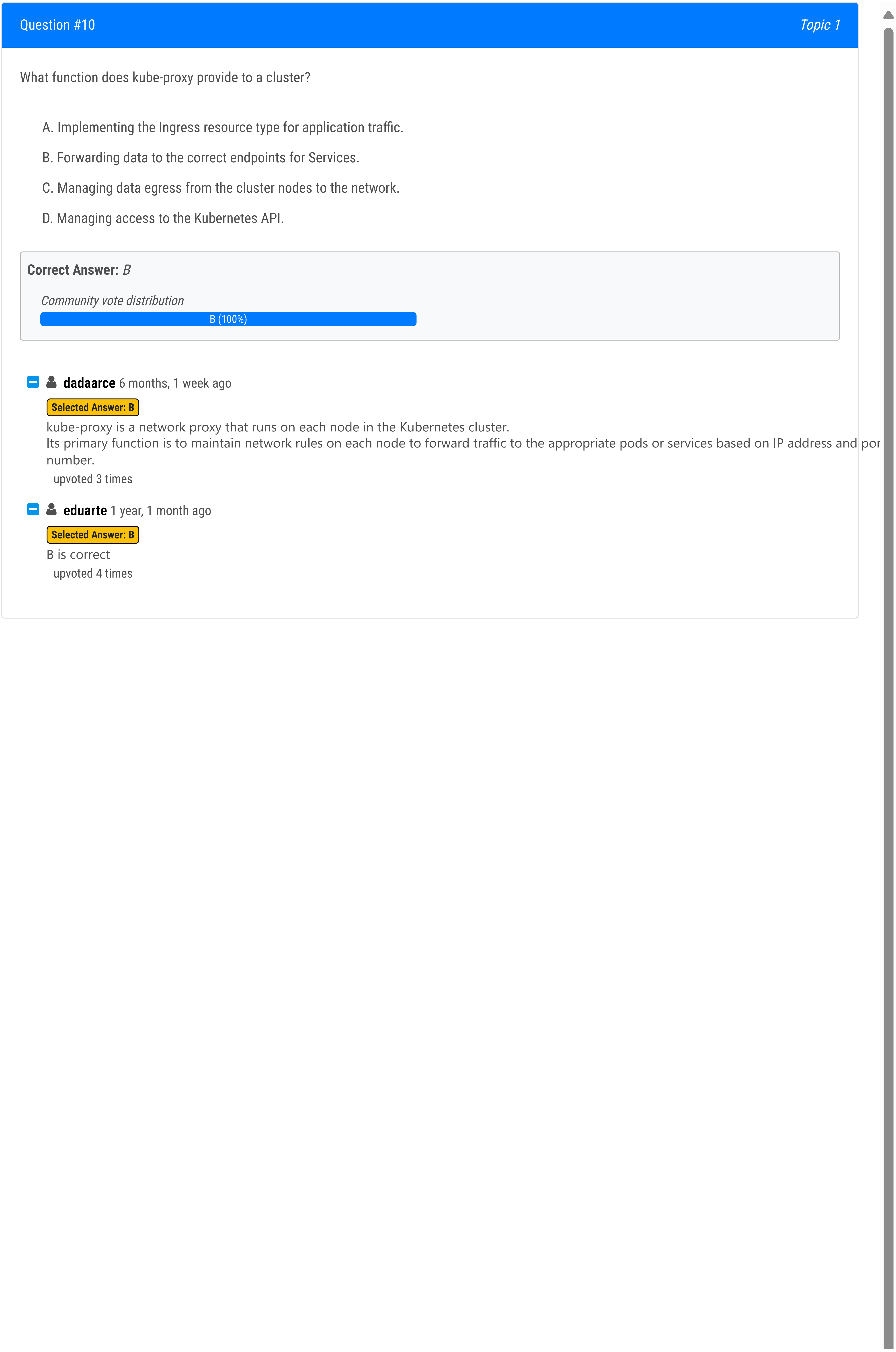
D is correct

upvoted 2 times

mazabel 1 year, 1 month ago

D is correct The secret data is represented as based64-encoded information

upvoted 4 times



Question #10

Topic 1

What function does kube-proxy provide to a cluster?

A. Implementing the Ingress resource type for application traffic.

B. Forwarding data to the correct endpoints for Services.

C. Managing data egress from the cluster nodes to the network.

D. Managing access to the Kubernetes API.

Correct Answer: *B*

Community vote distribution

B (100%)

dadaarce 6 months, 1 week ago

Selected Answer: B

kube-proxy is a network proxy that runs on each node in the Kubernetes cluster.
Its primary function is to maintain network rules on each node to forward traffic to the appropriate pods or services based on IP address and por
number.

upvoted 3 times

eduarte 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 4 times

Question #11 *Topic 1*

How long should a stable API element in Kubernetes be supported (at minimum) after deprecation?

A. 9 months

B. 24 months

C. 12 months

D. 6 months

Correct Answer: *C*

Community vote distribution

C (100%)

pulsefire 6 months, 4 weeks ago

this old question no longer valid:

https://kubernetes.io/docs/reference/using-api/deprecation-policy/#:~:text=Rule%20%231%3A%20API,regardless%20of%20track.

upvoted 1 times

omerco61 7 months, 3 weeks ago

Selected Answer: C

Rule #7: Deprecated behaviors must function for no less than 1 year after their announced deprecation.

https://kubernetes.io/docs/reference/using-api/deprecation-policy/#:~:text=Rule%20%237%3A%20Deprecated%20behaviors%20must%20function%20for%20no%20less%20than%201%20year%20after%20
eir%20announced%20deprecation.

upvoted 2 times

Charliesco 8 months, 2 weeks ago

Rule #11b: Metrics, after their announced deprecation, must function for no less than:

STABLE: 3 releases or 9 months (whichever is longer)
BETA: 1 releases or 4 months (whichever is longer)
ALPHA: 0 releases

Refer to : https://kubernetes.io/docs/reference/using-api/deprecation-policy/#:~:text=STABLE%3A%203%20releases%20or%209%20months%20(whichever%20is%20longer)

upvoted 1 times

eduarte 1 year, 1 month ago

Selected Answer: C

C is correct

upvoted 4 times

Question #12 *Topic 1*

What is the name of the lightweight Kubernetes distribution built for IoT and edge computing?

A. OpenShift

B. k3s

C. RKE

D. k1s

Correct Answer: *B*

Community vote distribution

B (100%)

r0xer 7 months ago

Easy, B is correct

upvoted 2 times

Spam2210561 1 year, 1 month ago

Can someone upload all the 60 KCNA Questions in the discussion forum or mail me?

upvoted 1 times

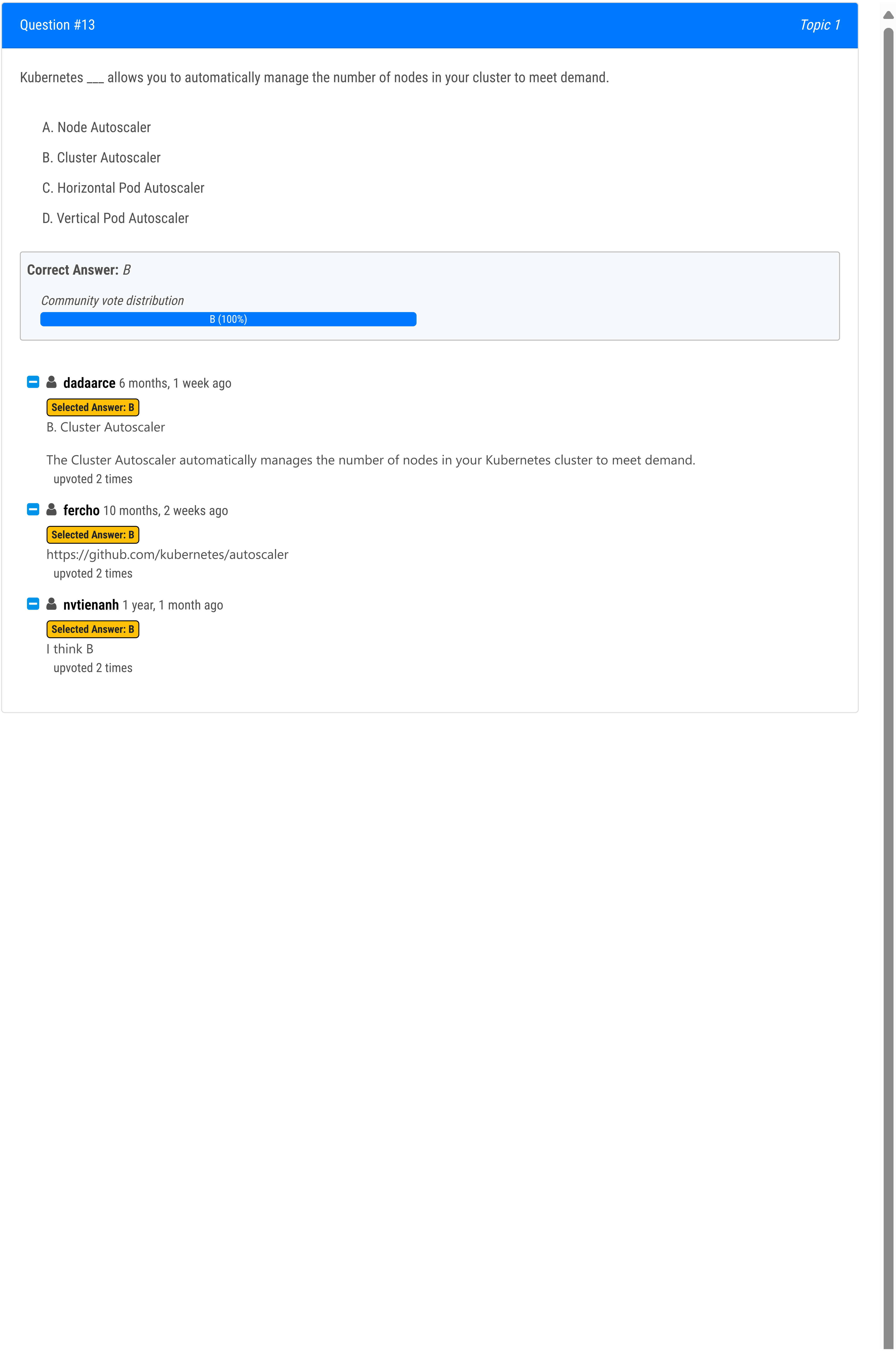
eduarte 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 3 times

> **Spam2210561** 1 year, 1 month ago
>
> Hey can u post all the 60 questions in the discussion forum?
>
> upvoted 2 times



Question #13 *Topic 1*

Kubernetes ___ allows you to automatically manage the number of nodes in your cluster to meet demand.

A. Node Autoscaler

B. Cluster Autoscaler

C. Horizontal Pod Autoscaler

D. Vertical Pod Autoscaler

Correct Answer: *B*

Community vote distribution

B (100%)

dadaarce 6 months, 1 week ago

Selected Answer: B

B. Cluster Autoscaler

The Cluster Autoscaler automatically manages the number of nodes in your Kubernetes cluster to meet demand.

upvoted 2 times

fercho 10 months, 2 weeks ago

Selected Answer: B

https://github.com/kubernetes/autoscaler

upvoted 2 times

nvtienanh 1 year, 1 month ago

Selected Answer: B

I think B

upvoted 2 times

Question #14 *Topic 1*

Which of the following statements is correct concerning Open Policy Agent (OPA)?

A. The policies must be written in Python language.

B. Kubernetes can use it to validate requests and apply policies.

C. Policies can only be tested when published.

D. It cannot be used outside Kubernetes.

Correct Answer: *B*

Community vote distribution

B (100%)

dadaarce 6 months, 1 week ago

Selected Answer: B

OPA is a general-purpose policy engine that can be used to enforce policies across various systems, including Kubernetes.
In Kubernetes, OPA can be integrated to validate admission requests, apply policies for resource creation, and enforce custom policies for securit
compliance, and other requirements.

upvoted 2 times

fercho 10 months, 2 weeks ago

Selected Answer: B

https://www.openpolicyagent.org/docs/latest/#:~:text=The%20Open%20Policy%20Agent%20(OPA,policy%20enforcement%20across%20the%20
ack.

upvoted 2 times

nvtienanh 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 2 times

Question #15 *Topic 1*

In a cloud native world, what does the IaC abbreviation stands for?

A. Infrastructure and Code

B. Infrastructure as Code

C. Infrastructure above Code

D. Infrastructure across Code

Correct Answer: *B*

Community vote distribution

B (100%)

fercho 10 months, 2 weeks ago

Selected Answer: B

B is correct

upvoted 2 times

nvtienanh 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 2 times

Question #16 *Topic 1*

In which framework do the developers no longer have to deal with capacity, deployments, scaling and fault tolerance, and OS?

A. Docker Swam

B. Kubernetes

C. Mesos

D. Serverless

Correct Answer: *D*

Community vote distribution

D (100%)

fercho 10 months, 2 weeks ago

Selected Answer: D

D is correct

upvoted 2 times

nvtienanh 1 year ago

Selected Answer: D

Should be D

upvoted 2 times

Question #17 *Topic 1*

Which of the following characteristics is associated with container orchestration?

A. Application message distribution

B. Dynamic scheduling

C. Deploying application JAR files

D. Virtual Machine distribution

Correct Answer: *B*

Community vote distribution

B (100%)

Train9 10 months ago

Yes B is correct

upvoted 2 times

fercho 10 months, 2 weeks ago

Selected Answer: B

B is correct

upvoted 2 times

Question #18 *Topic 1*

Which of the following workload require a headless service while deploying into the namespace?

A. StatefulSet

B. CronJob

C. Deployment

D. DaemonSet

Correct Answer: *A*

Community vote distribution

A (100%)

dadaarce 6 months, 1 week ago

Selected Answer: A

Headless services are not typically required for the other workload types listed:
CronJob: Executes Jobs at a scheduled time, but doesn't necessarily require a headless service.
Deployment: Manages stateless applications and can use a standard service for load balancing.
DaemonSet: Ensures that a copy of a pod runs on each node in the cluster, but doesn't require a headless service for normal operation.

upvoted 2 times

phcunha 6 months, 2 weeks ago

A. StatefulSet

Explanation:
StatefulSets are used for applications that maintain a persistent state or have a unique identity, such as databases. Each pod in a StatefulSet typically has a unique name, and it's necessary to ensure discovery and communication between these pods consistently, even when they are scaled up or down.

An analogy for StatefulSets could be managing a team of employees in an organization. Each employee has a unique name and a specific role. Even as the team grows or shrinks, it's important to maintain consistent communication among them. Therefore, you need a communication system (service) that can reliably locate each employee, regardless of changes in the team.

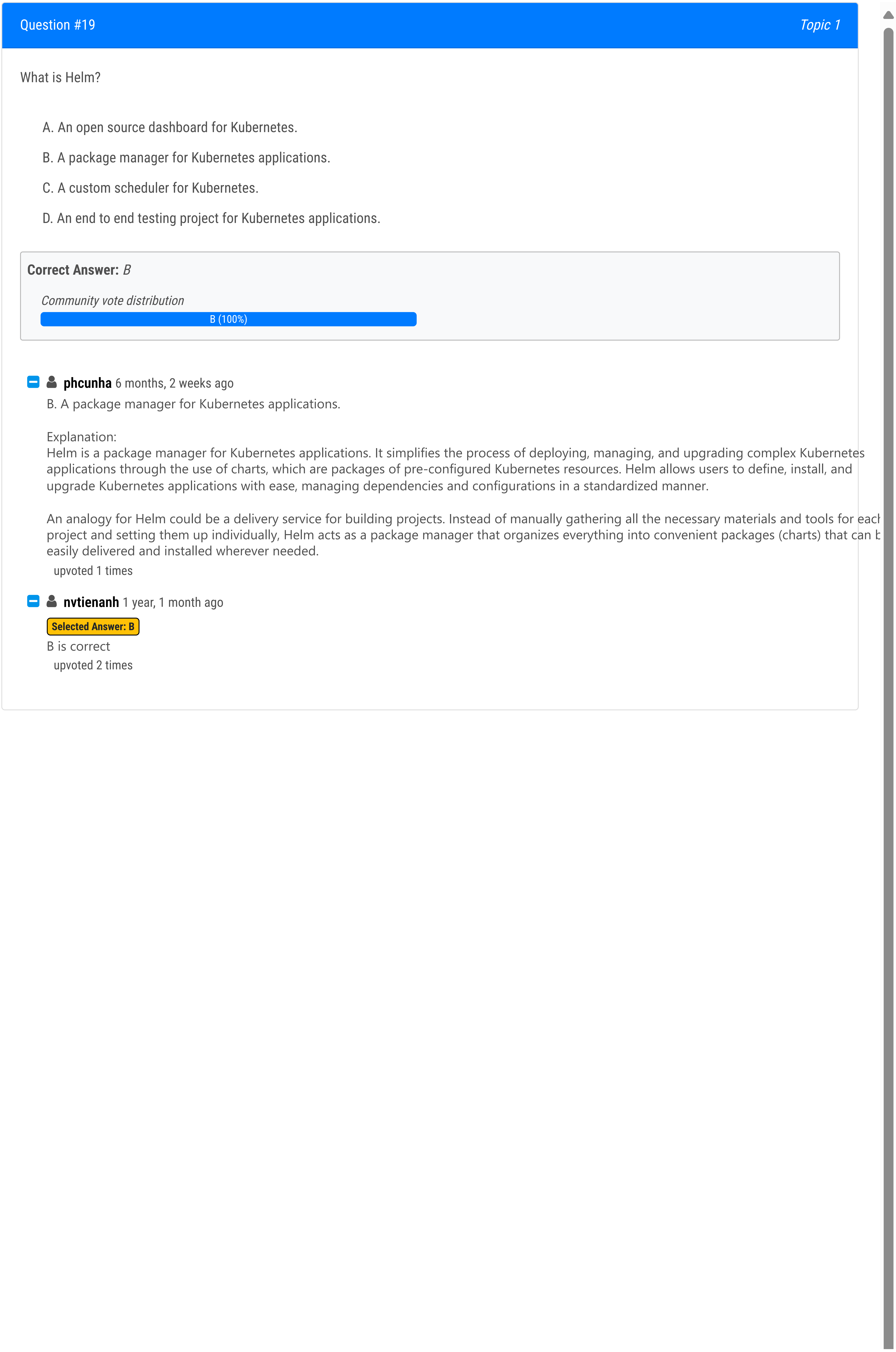
upvoted 2 times

pulsefire 6 months, 4 weeks ago

A.

https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/#:~:text=StatefulSets%20currently%20require%20a%20Headless%20Servi
%20to%20be%20responsible%20for%20the%20network%20identity%20of%20the%20Pods.%20You%20are%20responsible%20for%20creating%
this%20Service.

upvoted 2 times



Question #19 *Topic 1*

What is Helm?

A. An open source dashboard for Kubernetes.

B. A package manager for Kubernetes applications.

C. A custom scheduler for Kubernetes.

D. An end to end testing project for Kubernetes applications.

Correct Answer: *B*

Community vote distribution

B (100%)

phcunha 6 months, 2 weeks ago

B. A package manager for Kubernetes applications.

Explanation:
Helm is a package manager for Kubernetes applications. It simplifies the process of deploying, managing, and upgrading complex Kubernetes applications through the use of charts, which are packages of pre-configured Kubernetes resources. Helm allows users to define, install, and upgrade Kubernetes applications with ease, managing dependencies and configurations in a standardized manner.

An analogy for Helm could be a delivery service for building projects. Instead of manually gathering all the necessary materials and tools for each project and setting them up individually, Helm acts as a package manager that organizes everything into convenient packages (charts) that can b easily delivered and installed wherever needed.

upvoted 1 times

nvtienanh 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 2 times

Question #20 *Topic 1*

Which is the correct kubectl command to display logs in real time?

A. kubectl logs -p test-container-1

B. kubectl logs -c test-container-1

C. kubectl logs -l test-container-1

D. kubectl logs -f test-container-1

Correct Answer: *D*

Community vote distribution

D (100%)

dadaarce 6 months, 1 week ago

Selected Answer: D

The -f or --follow flag is used to stream logs in real time.

upvoted 2 times

phcunha 6 months, 2 weeks ago

D. kubectl logs -f test-container-1

Explanation:
The -f flag in the kubectl logs command stands for "follow," which allows you to continuously stream the logs in real-time as new log entries are added. This is useful for monitoring applications or troubleshooting issues as they occur.

An analogy for this command could be tuning in to a live radio broadcast. When you listen to a live radio show, you want to hear the content as being broadcasted, rather than just a recording of past broadcasts. Similarly, using kubectl logs -f lets you "tune in" to the ongoing activity of a container, receiving log updates in real-time.

upvoted 1 times

nvtienanh 1 year ago

Selected Answer: D

D is correct

upvoted 2 times

Question #21 *Topic 1*

How to load and generate data required before the Pod startup?

A. Use an init container with shared file storage.

B. Use a PVC volume.

C. Use a sidecar container with shared volume.

D. Use another pod with a PVC.

Correct Answer: *A*

Community vote distribution

A (100%)

phcunha 6 months, 2 weeks ago

A. Use an init container with shared file storage.

Explanation:
Init containers are executed before the main containers in a Pod are started. They are designed to perform initialization tasks, such as loading an generating data required before the main containers start. Init containers can share file storage with the main containers in the Pod, allowing the to perform tasks like data loading or generation and then making that data available to the main containers.

An analogy for using an init container could be preparing ingredients before cooking a meal. Just as you might prepare ingredients like chopping vegetables or marinating meat before starting to cook, init containers prepare data or perform tasks necessary for the main containers to functio properly before the main containers start running.

upvoted 4 times

nvtienanh 1 year, 1 month ago

Selected Answer: A

A is correct

upvoted 3 times

Question #22 *Topic 1*

What is the core functionality of GitOps tools like Argo CD and Flux?

A. They track production changes made by a human in a Git repository and generate a human-readable audit trail.

B. They replace human operations with an agent that tracks Git commands.

C. They automatically create pull requests when dependencies are outdated.

D. They continuously compare the desired state in Git with the actual production state and notify or act upon differences.

Correct Answer: *D*

Community vote distribution

D (100%)

phcunha 6 months, 2 weeks ago

An analogy for this functionality could be a thermostat in a room. The desired temperature set on the thermostat represents the configuration defined in Git, while the actual temperature in the room represents the production state. If the actual temperature deviates from the desired temperature, the thermostat (GitOps tool) will automatically adjust the heating or cooling systems to bring the room temperature back to the desired level. Similarly, GitOps tools ensure that the production state of the cluster matches the desired state defined in Git.

upvoted 1 times

phcunha 6 months, 2 weeks ago

D. They continuously compare the desired state in Git with the actual production state and notify or act upon differences.

Explanation:
GitOps tools like Argo CD and Flux are designed to manage Kubernetes clusters by using Git repositories as the source of truth for cluster configuration. The core functionality of these tools involves continuously comparing the desired state of the cluster, as defined in Git, with the actual state of the cluster running in production. If there are differences between the desired and actual states, these tools will automatically reconcile those differences by either updating the production state to match the desired state or notifying operators of any discrepancies.

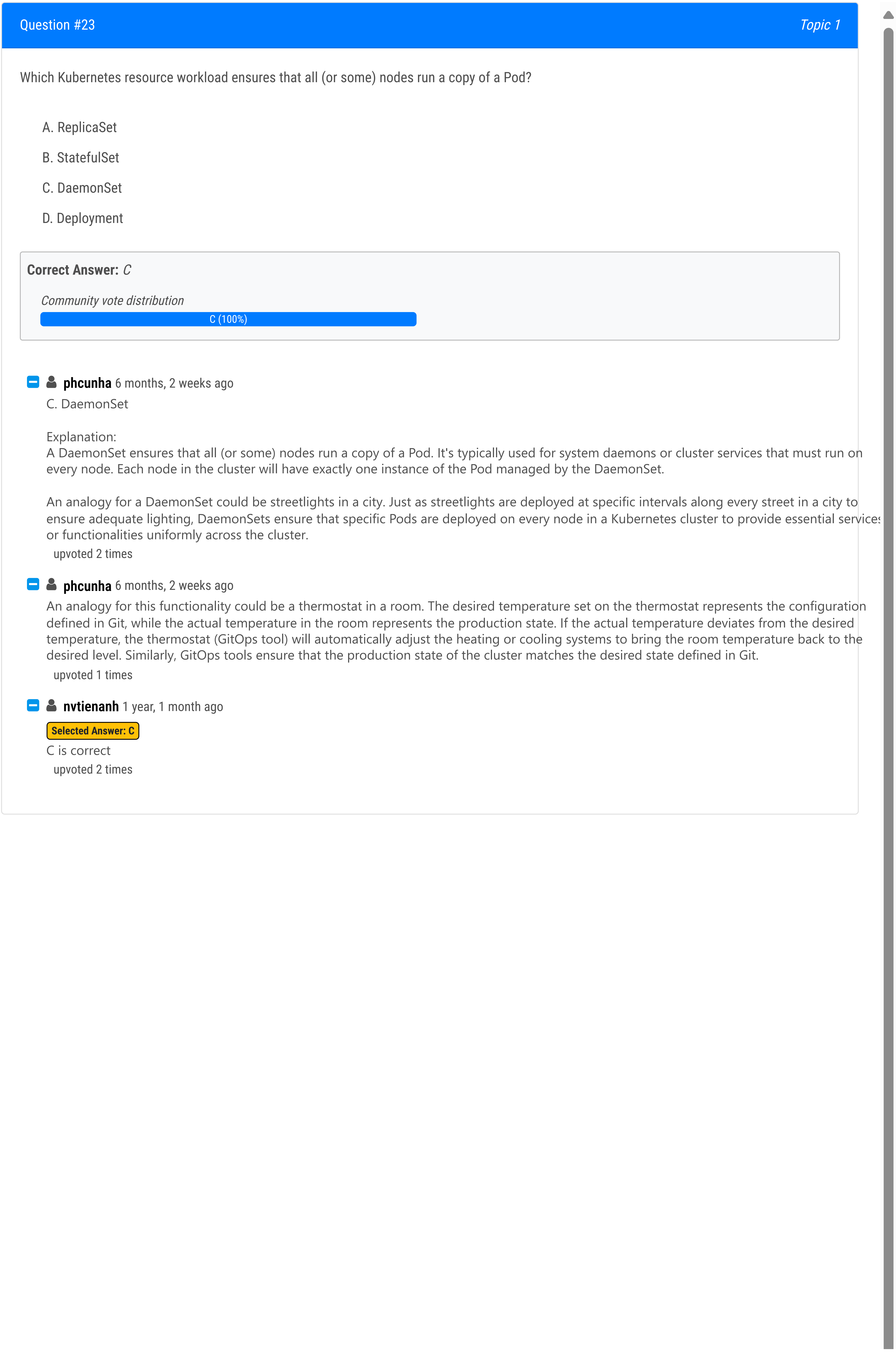
upvoted 2 times

nvtienanh 1 year ago

Selected Answer: D

D look right

upvoted 2 times



Question #23

Topic 1

Which Kubernetes resource workload ensures that all (or some) nodes run a copy of a Pod?

A. ReplicaSet

B. StatefulSet

C. DaemonSet

D. Deployment

Correct Answer: *C*

Community vote distribution

C (100%)

phcunha 6 months, 2 weeks ago

C. DaemonSet

Explanation:
A DaemonSet ensures that all (or some) nodes run a copy of a Pod. It's typically used for system daemons or cluster services that must run on every node. Each node in the cluster will have exactly one instance of the Pod managed by the DaemonSet.

An analogy for a DaemonSet could be streetlights in a city. Just as streetlights are deployed at specific intervals along every street in a city to ensure adequate lighting, DaemonSets ensure that specific Pods are deployed on every node in a Kubernetes cluster to provide essential service or functionalities uniformly across the cluster.

upvoted 2 times

phcunha 6 months, 2 weeks ago

An analogy for this functionality could be a thermostat in a room. The desired temperature set on the thermostat represents the configuration defined in Git, while the actual temperature in the room represents the production state. If the actual temperature deviates from the desired temperature, the thermostat (GitOps tool) will automatically adjust the heating or cooling systems to bring the room temperature back to the desired level. Similarly, GitOps tools ensure that the production state of the cluster matches the desired state defined in Git.

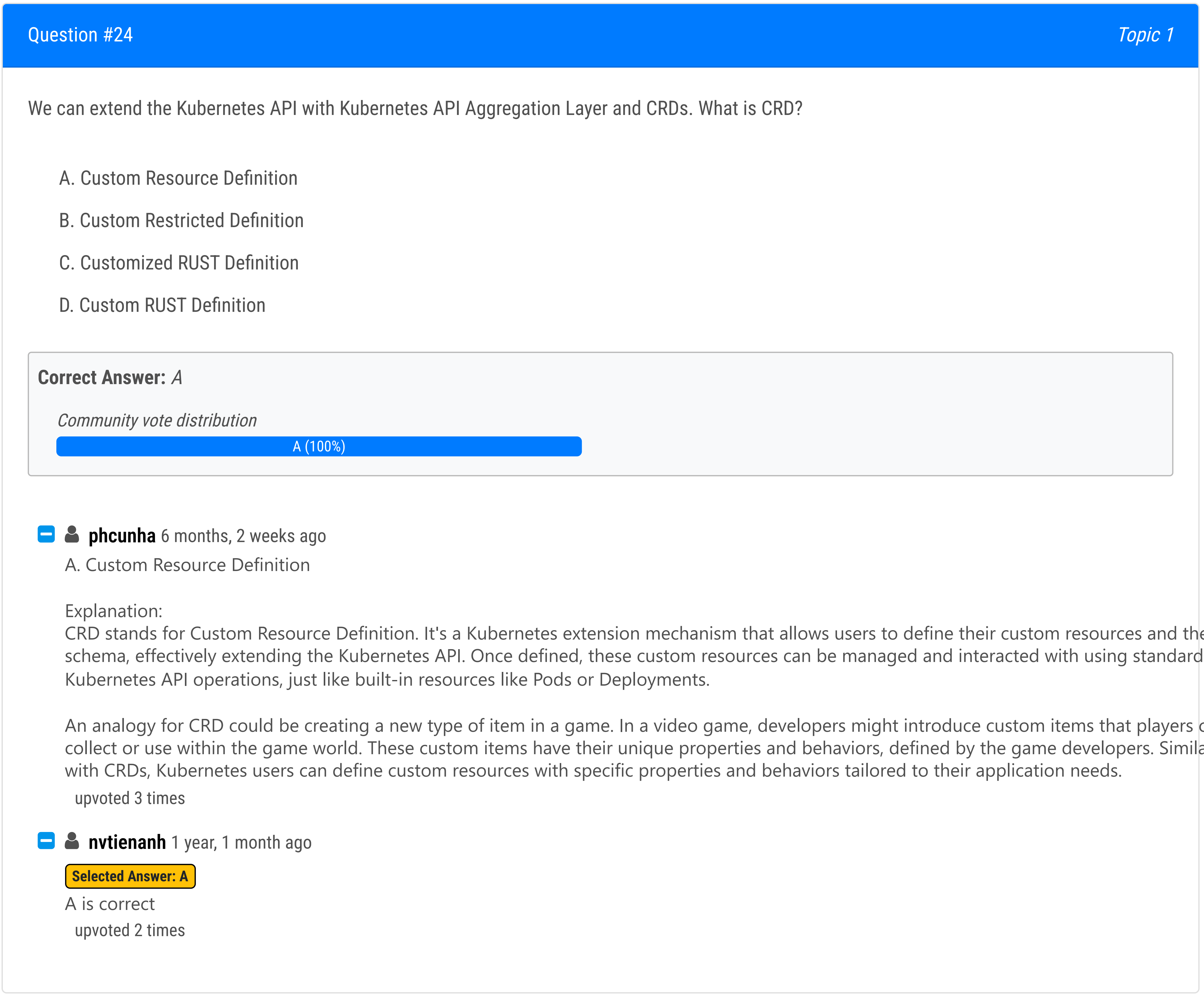
upvoted 1 times

nvtienanh 1 year, 1 month ago

Selected Answer: C

C is correct

upvoted 2 times



Question #24 *Topic 1*

We can extend the Kubernetes API with Kubernetes API Aggregation Layer and CRDs. What is CRD?

A. Custom Resource Definition

B. Custom Restricted Definition

C. Customized RUST Definition

D. Custom RUST Definition

Correct Answer: *A*

Community vote distribution

A (100%)

phcunha 6 months, 2 weeks ago

A. Custom Resource Definition

Explanation:
CRD stands for Custom Resource Definition. It's a Kubernetes extension mechanism that allows users to define their custom resources and their schema, effectively extending the Kubernetes API. Once defined, these custom resources can be managed and interacted with using standard Kubernetes API operations, just like built-in resources like Pods or Deployments.

An analogy for CRD could be creating a new type of item in a game. In a video game, developers might introduce custom items that players can collect or use within the game world. These custom items have their unique properties and behaviors, defined by the game developers. Similarly, with CRDs, Kubernetes users can define custom resources with specific properties and behaviors tailored to their application needs.

upvoted 3 times

nvtienanh 1 year, 1 month ago

Selected Answer: A

A is correct

upvoted 2 times

Question #25 *Topic 1*

The Kubernetes project work is carried primarily by SIGs. What does SIG stand for?

A. Special Interest Group

B. Software Installation Guide

C. Support and Information Group

D. Strategy Implementation Group

Correct Answer: *A*

Community vote distribution

A (100%)

phcunha 6 months, 2 weeks ago

A. Special Interest Group

Explanation:
SIG stands for Special Interest Group. In the context of Kubernetes, SIGs are groups of contributors focused on specific areas or domains within t project. Each SIG is responsible for overseeing and contributing to the development, maintenance, and improvement of its designated area, such as networking, storage, scalability, or documentation.

An analogy for SIGs could be departments within an organization. In a company, different departments (such as marketing, sales, or engineering focus on specific areas of expertise and collaborate to achieve common goals. Similarly, SIGs in Kubernetes focus on specific aspects of the proje and work together to advance the development and functionality of those areas.

upvoted 2 times

EzBL 6 months, 2 weeks ago

Selected Answer: A

In the Kubernetes project, SIG stands for Special Interest Group. SIGs are responsible for focused areas of the project and coordinate efforts relat to those areas. Each SIG has its own scope, responsibilities, and channels for communication, and they play a crucial role in the development, maintenance, and evolution of Kubernetes. Examples of SIGs include SIG-CLI, SIG-Storage, SIG-Network, etc.

upvoted 2 times

aouihw 1 year ago

Selected Answer: A

Correct https://www.cncf.io/announcements/2019/09/12/cloud-native-computing-foundation-announces-application-delivery-sig/

upvoted 2 times

nvtienanh 1 year, 1 month ago

Selected Answer: A

Given answer is correct.

upvoted 2 times

Question #26 *Topic 1*

What is the order of 4C’s in Cloud Native Security, starting with the layer that a user has the most control over?

A. Cloud -> Container -> Cluster -> Code

B. Container -> Cluster -> Code -> Cloud

C. Cluster -> Container -> Code -> Cloud

D. Code -> Container -> Cluster -> Cloud

Correct Answer: *D*

Community vote distribution

D (100%)

nvtienanh 1 year, 1 month ago

Selected Answer: D

D make sense

upvoted 3 times

Question #27 *Topic 1*

Which group of container runtimes provides additional sandboxed isolation and elevated security?

A. rune, cgroups

B. docker, containerd

C. runsc, kata

D. crun, cri-o

Correct Answer: *C*

Community vote distribution

C (100%)

phcunha 6 months, 2 weeks ago

An analogy for this could be security checkpoints in a high-security facility. Just as additional security measures like checkpoints and barriers enhance security within a facility, runsc and Kata Containers add extra layers of isolation and security to containerized environments, ensuring th even if one container is compromised, it doesn't affect the security of other containers or the host system.

upvoted 1 times

phcunha 6 months, 2 weeks ago

runsc (gVisor): It's a lightweight container runtime that runs containers inside a sandboxed environment, providing an additional layer of isolatio using user-space kernel emulation. This allows containers to have their own isolated kernel without the overhead of full virtualization.

Kata Containers: It's an open-source project that combines the security of virtual machines with the speed and manageability of containers. It use lightweight VMs to run each container, providing strong isolation between containers without the performance overhead of traditional virtual machines.

upvoted 1 times

phcunha 6 months, 2 weeks ago

C. runsc, kata

Explanation:
Runsc (gVisor) and Kata Containers are container runtimes that provide additional sandboxed isolation and elevated security compared to traditional container runtimes like Docker or containerd.

upvoted 2 times

pulsefire 6 months, 4 weeks ago

Selected Answer: C

C.

https://docs.openshift.com/container-platform/4.8/sandboxed_containers/understanding-sandboxed-containers.html#:~:text=OpenShift%20sandboxed%20containers%20support,containment%20through%20VM%20boundaries.

upvoted 2 times

Question #28 *Topic 1*

What is the common standard for Service Meshes?

A. Service Mesh Specification (SMS)

B. Service Mesh Technology (SMT)

C. Service Mesh Interface (SMI)

D. Service Mesh Function (SMF)

Correct Answer: *C*

phcunha 6 months, 2 weeks ago

C. Service Mesh Interface (SMI)

Explanation:
The Service Mesh Interface (SMI) is a specification for implementing service mesh functionality across different service mesh implementations. It provides a standard set of APIs for controlling and observing service mesh behavior, allowing for interoperability between various service mesh solutions.

An analogy for SMI could be a universal remote control standard. Just as different electronic devices from different manufacturers can be controlled using a universal remote that adheres to a common standard, service mesh implementations can communicate and interoperate effectively using the Service Mesh Interface (SMI) specification.

upvoted 3 times

Question #29 *Topic 1*

Which statement about Ingress is correct?

A. Ingress provides a simple way to track network endpoints within a cluster.

B. Ingress is a Service type like NodePort and ClusterIP.

C. Ingress is a construct that allows you to specify how a Pod is allowed to communicate.

D. Ingress exposes routes from outside the cluster to services in the cluster.

Correct Answer: *D*

Community vote distribution

D (100%)

phcunha 6 months, 2 weeks ago

D. Ingress exposes routes from outside the cluster to services in the cluster.

Explanation:
Ingress in Kubernetes is an API object that manages external access to services within a cluster. It provides HTTP and HTTPS routing to services based on incoming requests' hostnames, paths, or other criteria. Ingress exposes routes from outside the cluster to services inside the cluster, acting as an entry point for external traffic to reach services.

An analogy for Ingress could be a building's main entrance gate. Just as a main entrance gate controls access to different sections or floors withi building, Ingress controls access to different services within a Kubernetes cluster based on defined routing rules.

upvoted 2 times

AbhishekJoshi 9 months, 2 weeks ago

Selected Answer: D

https://kubernetes.io/docs/concepts/services-networking/ingress/#:~:text=Ingress%20exposes%20HTTP%20and%20HTTPS,defined%20on%20the%20Ingress%20resource.&text=An%20Ingre %20may%20be%20configured,offer%20name%2Dbased%20virtual%20hosting.

upvoted 2 times

Question #30 *Topic 1*

What best describes cloud native service discovery?

A. It's a mechanism for applications and microservices to locate each other on a network.

B. It's a procedure for discovering a MAC address, associated with a given IP address.

C. It's used for automatically assigning IP addresses to devices connected to the network.

D. It's a protocol that turns human-readable domain names into IP addresses on the Internet.

Correct Answer: *A*

phcunha 6 months, 2 weeks ago

A. It's a mechanism for applications and microservices to locate each other on a network.

Explanation:
Cloud native service discovery is a crucial aspect of distributed systems architecture, particularly in microservices environments. It enables applications and microservices to dynamically discover and communicate with each other without hardcoding IP addresses or relying on static configurations. This mechanism allows services to be scalable, resilient, and loosely coupled, facilitating efficient communication and interaction within the cloud environment.

An analogy for service discovery could be a directory in a large office building. Just as a directory helps individuals find the locations of various offices or departments within a building, service discovery enables components within a distributed system to locate and communicate with eacl other dynamically, regardless of their specific network locations or configurations.

upvoted 2 times

Question #31 *Topic 1*

What components are common in a service mesh?

A. tracing and log storage

B. circuit breaking and Pod scheduling

C. data plane and runtime plane

D. service proxy and control plane

Correct Answer: *D*

Community vote distribution

D (100%)

EzBL 6 months, 2 weeks ago

Selected Answer: D

In a service mesh architecture, service proxies (such as Envoy, Linkerd proxy, or Istio sidecar) are deployed alongside each service instance to handle communication between services. The control plane consists of various components responsible for configuring, managing, and monitori the behavior of the service proxies, including features like traffic routing, load balancing, encryption, authentication, and observability. Together, these components enable advanced traffic management, security, and observability capabilities in a distributed microservices environment.

upvoted 2 times

PinkAndBlack 9 months, 3 weeks ago

I can confirn that D is the corrct answer, anyway the question is a bit tricky, because the data plan can be considered as the set of service proxies even the C answer could be considered

upvoted 3 times

Question #32 *Topic 1*

Which storage operator in Kubernetes can help the system to self-scale, self-heal, etc?

A. Rook

B. Kubernetes

C. Helm

D. Container Storage Interface (CSI)

Correct Answer: *A*

Community vote distribution

A (100%)

dadaarce 6 months, 1 week ago

Selected Answer: A

Rook provides features like self-scaling, self-healing, monitoring, and automation of storage management tasks. It abstracts the complexities of managing distributed storage systems and integrates them seamlessly with Kubernetes, enabling operators to deploy and manage storage cluste as easily as deploying other Kubernetes resources.

upvoted 2 times

Question #33 *Topic 1*

What fields must exist in any Kubernetes object (e.g. YAML) file?

A. apiVersion, kind, metadata

B. kind, namespace, data

C. apiVersion, metadata, namespace

D. kind, metadata, data

Correct Answer: *A*

phcunha 6 months, 2 weeks ago

A. apiVersion, kind, metadata

Explanation:
In any Kubernetes object definition file, there are three mandatory fields that must exist:

apiVersion: Specifies the version of the Kubernetes API that the object uses. This field ensures compatibility and defines the structure of the objec

kind: Specifies the type of Kubernetes object being defined, such as Pod, Service, Deployment, etc. This field determines the behavior and functionality of the object.

metadata: Contains metadata about the object, such as its name, namespace, labels, and annotations. This metadata is used by Kubernetes to identify and manage the object within the cluster.

These fields are essential for Kubernetes to interpret and process the object correctly. The other options do not include all three mandatory field: or contain fields that are not universally required in every Kubernetes object.

upvoted 2 times

Question #34 *Topic 1*

Which of the following would fall under the responsibilities of an SRE?

A. Developing a new application feature.

B. Creating a monitoring baseline for an application.

C. Submitting a budget for running an application in a cloud.

D. Writing policy on how to submit a code change.

Correct Answer: *B*

phcunha 6 months, 2 weeks ago

Designing strategies for disaster recovery and fault tolerance: Just as the maintenance engineer creates contingency plans for breakdowns or accidents, the SRE devises plans for system failures or outages, ensuring that services can quickly recover and continue functioning.

Analyzing system performance metrics: Like reviewing maintenance logs and performance data for each truck, the SRE analyzes metrics from the application's monitoring systems to identify areas for improvement and optimize performance.

By considering the SRE's responsibilities in the context of managing a fleet of delivery trucks, we can draw parallels that illustrate the focus on reliability, efficiency, and proactive maintenance inherent in the role.

upvoted 1 times

phcunha 6 months, 2 weeks ago

B. Creating a monitoring baseline for an application.
Imagine an SRE as a maintenance engineer for a fleet of delivery trucks. Their primary responsibility is to ensure that the trucks operate reliably a efficiently, delivering goods to customers on time. Here's how the responsibilities of an SRE might align with this analogy:

Creating a monitoring baseline for an application: This is akin to installing sensors and monitoring systems on each truck to track parameters like fuel consumption, engine performance, and GPS location. By establishing a baseline for normal truck operation, the SRE can quickly identify deviations or potential issues.

Developing and implementing automation for tasks: Similar to designing automated loading and unloading systems for trucks, the SRE automat repetitive tasks in the operation of the fleet, such as scheduling maintenance checks or rerouting trucks in case of road closures.

upvoted 3 times

Question #35 *Topic 1*

What are the initial namespaces that Kubernetes starts with?

A. default, kube-system, kube-public, kube-node-lease

B. default, system, kube-public

C. kube-default, kube-system, kube-main, kube-node-lease

D. kube-default, system, kube-main, kube-primary

Correct Answer: *A*

Error_2k 8 months, 1 week ago

correct:
https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/

upvoted 2 times

Question #36 *Topic 1*

What is a probe within Kubernetes?

A. A monitoring mechanism of the Kubernetes API.

B. A pre-operational scope issued by the kubectl agent.

C. A diagnostic performed periodically by the kubelet on a container.

D. A logging mechanism of the Kubernetes API.

Correct Answer: *C*

Community vote distribution

C (100%)

SeaH0rse66 4 months, 2 weeks ago

Selected Answer: C

Within Kubernetes, a "probe" refers to a diagnostic mechanism used by the kubelet to check the health of containers running within pods.Probes are configured within the PodSpec of Kubernetes pods and are used to determine if the containers within the pod are healthy and ready to serve traffic. There are three types of probes:

Liveness Probe: Determines if the container is still running and healthy. If the liveness probe fails, Kubernetes restarts the container.

Readiness Probe: Determines if the container is ready to serve traffic. If the readiness probe fails, the pod is removed from load balancers, and nc traffic is routed to it.

Startup Probe: Similar to the liveness probe, but only runs during the initial startup of a container. It helps delay the liveness and readiness probe until the application inside the container has started.

upvoted 2 times

phcunha 6 months, 2 weeks ago

Think of a probe in Kubernetes like a heart rate monitor attached to a patient in a hospital. Just as the heart rate monitor continuously checks the patient's heart rate to ensure they are alive and functioning correctly, Kubernetes probes continuously monitor the health of containers to ensure they are running and capable of serving traffic. If the heart rate monitor detects irregularities or a lack of heartbeat, medical staff take action to revive or stabilize the patient. Similarly, if Kubernetes probes detect issues with a container, Kubernetes takes action to restart or replace the container, ensuring the application remains available and responsive.

upvoted 1 times

Question #37 *Topic 1*

Which Kubernetes feature would you use to guard against split brain scenarios with your distributed application?

A. Replication controllers

B. Consensus protocols

C. Rolling updates

D. StatefulSet

Correct Answer: *D*

Community vote distribution

D (63%) B (38%)

Andrei_Z 1 week, 5 days ago

Selected Answer: D

The question says: "with your distributed application" so it is D.

upvoted 1 times

EzBL 3 months, 1 week ago

Selected Answer: B

Consensus protocols, such as those provided by distributed systems like etcd or ZooKeeper, help prevent split-brain scenarios by ensuring that only one leader or primary instance is elected to make decisions or perform critical tasks within the distributed system at any given time. These protocols provide mechanisms for nodes to coordinate and agree on the state of the system, even in the presence of network partitions or failur thereby mitigating the risk of conflicting or divergent states that could lead to split-brain scenarios. While Kubernetes itself does not provide consensus protocols directly, it often relies on external systems like etcd for managing cluster state and coordination.

upvoted 1 times

SeaH0rse66 4 months, 2 weeks ago

Selected Answer: D

D. StatefulSet

While consensus protocols are generally used to prevent split brain scenarios in distributed systems, the provided information highlights that StatefulSets are specifically designed to ensure the stability and integrity of distributed and clustered applications. StatefulSets maintain "at most one" semantics, which helps prevent multiple instances of the same identity, reducing the risk of split brain scenarios and data loss in quorum-based systems.

StatefulSets are well-suited for applications that require stable network identity and storage, providing mechanisms to manage pod identities, persistent storage, and ordered deployment and scaling. Therefore, StatefulSets are the most appropriate Kubernetes feature to mitigate the risk split brain scenarios in this context.

upvoted 2 times

hovnival 5 months ago

Selected Answer: B

guys, both copilot and ChatGPT says Consensus protocols.

upvoted 2 times

SeaH0rse66 4 months, 2 weeks ago

chatGPT is wrong...D. StatefulSet

While consensus protocols are generally used to prevent split brain scenarios in distributed systems, the provided information highlights that StatefulSets are specifically designed to ensure the stability and integrity of distributed and clustered applications. StatefulSets maintain "at most one" semantics, which helps prevent multiple instances of the same identity, reducing the risk of split brain scenarios and data loss in quorum-based systems.

StatefulSets are well-suited for applications that require stable network identity and storage, providing mechanisms to manage pod identities persistent storage, and ordered deployment and scaling. Therefore, StatefulSets are the most appropriate Kubernetes feature to mitigate the risk of split brain scenarios in this context.

upvoted 1 times

phcunha 6 months, 2 weeks ago

Think of StatefulSet in Kubernetes as a traffic control system at an intersection with multiple lanes. Just as the traffic control system manages the flow of vehicles through the intersection, ensuring that only one lane can proceed at a time to prevent collisions or gridlock, StatefulSet manages the deployment of stateful applications, ensuring that only one instance can be active or "in charge" at any given time to prevent conflicts or inconsistencies in distributed systems.

upvoted 1 times

pulsefire 6 months, 4 weeks ago

Selected Answer: D

https://kubernetes.io/docs/tasks/run-application/force-delete-stateful-set-pod/#:~:text=split%20brain%20scenario%20in%20quorum%2Dbased%20systems

upvoted 2 times

sad_schedule 8 months, 3 weeks ago

https://unofficial-kubernetes.readthedocs.io/en/latest/tasks/manage-stateful-set/upgrade-pet-set-to-stateful-set/#:~:text=StatefulSet%20guards%20against%20split%20brain,brain%20scenarios%20with%20distributed%20applications.

upvoted 1 times

sadsak 9 months, 2 weeks ago

Shouldn't this be Consensus protocols?

upvoted 1 times

Question #38

Topic 1

What feature must a CNI support to control specific traffic flows for workloads running in Kubernetes?

A. Border Gateway Protocol

B. IP Address Management

C. Pod Security Policy

D. Network Policies

Correct Answer: *D*

Community vote distribution

D (100%)

phcunha 6 months, 2 weeks ago

Think of Network Policies in Kubernetes as security checkpoints at different entrances to a building. Just as security personnel at each entrance check IDs and verify permissions before allowing individuals to enter specific areas of the building, Network Policies control the flow of network traffic within the Kubernetes cluster, ensuring that only authorized Pods can communicate with each other and that access to sensitive services is restricted according to defined rules.

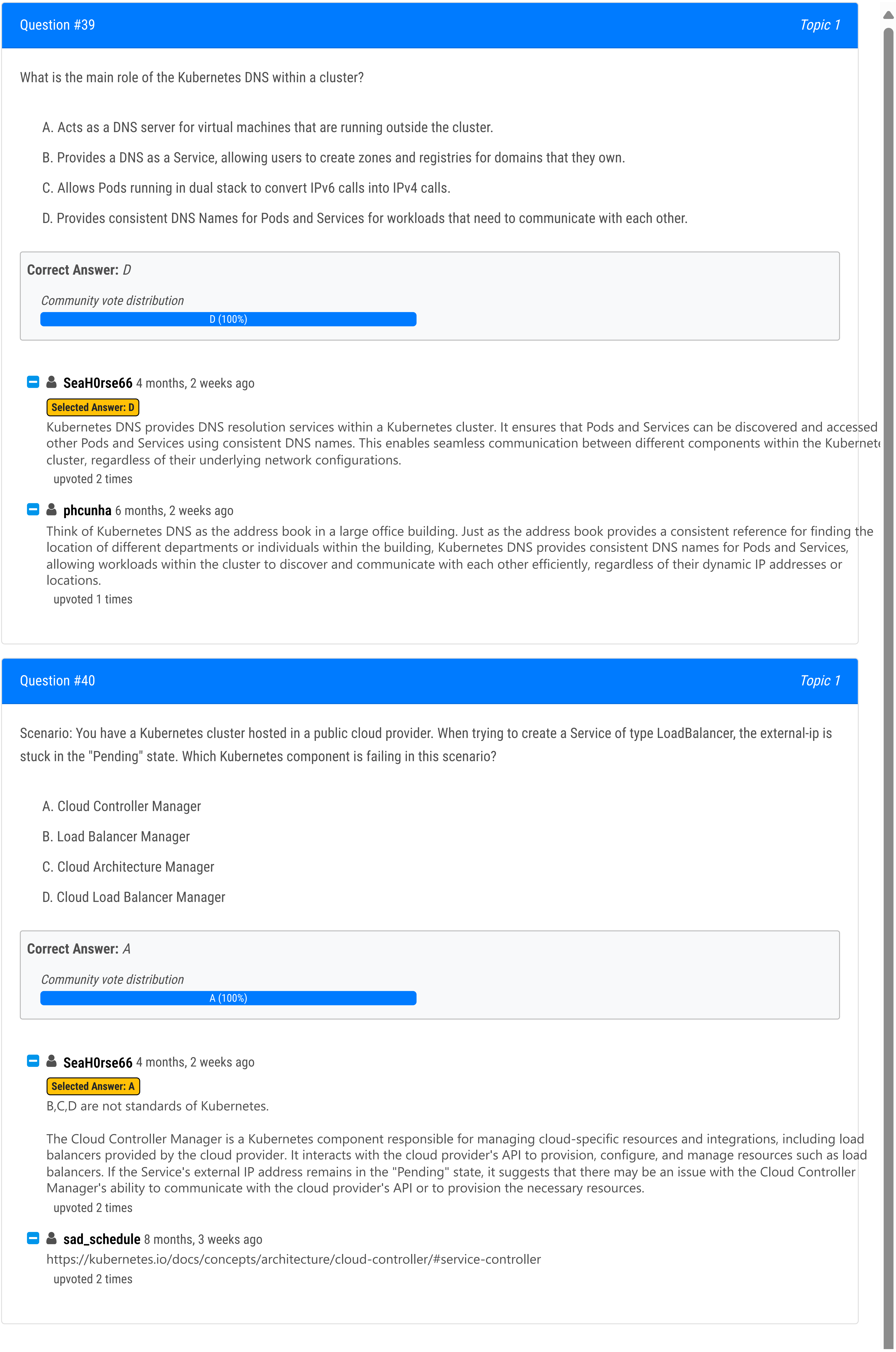
upvoted 2 times

pulsefire 6 months, 4 weeks ago

Selected Answer: D

https://kubernetes.io/docs/concepts/services-networking/network-policies/#:~:text=Network%20Policies-,Network%20Policies,might%20consider%20using%20Kubernetes%20NetworkPolicies%20for%20particula
20applications%20in%20your%20cluster.,-NetworkPolicies%20are%20an

upvoted 2 times



Question #39 *Topic 1*

What is the main role of the Kubernetes DNS within a cluster?

A. Acts as a DNS server for virtual machines that are running outside the cluster.

B. Provides a DNS as a Service, allowing users to create zones and registries for domains that they own.

C. Allows Pods running in dual stack to convert IPv6 calls into IPv4 calls.

D. Provides consistent DNS Names for Pods and Services for workloads that need to communicate with each other.

Correct Answer: *D*

Community vote distribution

D (100%)

SeaH0rse66 4 months, 2 weeks ago

Selected Answer: D

Kubernetes DNS provides DNS resolution services within a Kubernetes cluster. It ensures that Pods and Services can be discovered and accessed other Pods and Services using consistent DNS names. This enables seamless communication between different components within the Kubernete cluster, regardless of their underlying network configurations.

upvoted 2 times

phcunha 6 months, 2 weeks ago

Think of Kubernetes DNS as the address book in a large office building. Just as the address book provides a consistent reference for finding the location of different departments or individuals within the building, Kubernetes DNS provides consistent DNS names for Pods and Services, allowing workloads within the cluster to discover and communicate with each other efficiently, regardless of their dynamic IP addresses or locations.

upvoted 1 times

Question #40 *Topic 1*

Scenario: You have a Kubernetes cluster hosted in a public cloud provider. When trying to create a Service of type LoadBalancer, the external-ip is stuck in the "Pending" state. Which Kubernetes component is failing in this scenario?

A. Cloud Controller Manager

B. Load Balancer Manager

C. Cloud Architecture Manager

D. Cloud Load Balancer Manager

Correct Answer: *A*

Community vote distribution

A (100%)

SeaH0rse66 4 months, 2 weeks ago

Selected Answer: A

B,C,D are not standards of Kubernetes.

The Cloud Controller Manager is a Kubernetes component responsible for managing cloud-specific resources and integrations, including load balancers provided by the cloud provider. It interacts with the cloud provider's API to provision, configure, and manage resources such as load balancers. If the Service's external IP address remains in the "Pending" state, it suggests that there may be an issue with the Cloud Controller Manager's ability to communicate with the cloud provider's API or to provision the necessary resources.

upvoted 2 times

sad_schedule 8 months, 3 weeks ago

https://kubernetes.io/docs/concepts/architecture/cloud-controller/#service-controller

upvoted 2 times

Question #41 *Topic 1*

What are the characteristics for building every cloud-native application?

A. Resiliency, Operability, Observability, Availability

B. Resiliency, Containerd, Observability, Agility

C. Kubernetes, Operability, Observability, Availability

D. Resiliency, Agility, Operability, Observability

Correct Answer: *D*

Community vote distribution

D (71%) A (29%)

z2salman 4 months ago

Selected Answer: A

As stated above, Availability is a key rationality to implement containerization of applications as opposed to agility.

upvoted 1 times

alex78 5 months ago

Selected Answer: D

A cloud native application is engineered to run on a platform and is designed for resiliency, agility, operability, and observability. Resiliency embraces failures instead of trying to prevent them; it takes advantage of the dynamic nature of running on a platform. Agility allows for fast deployments and quick iterations. Operability adds control of application life cycles from inside the application instead of relying on external processes and monitors. Observability provides information to answer questions about application state.

https://www.oreilly.com/library/view/cloud-native-infrastructure/9781491984291/ch01.html

upvoted 2 times

alex78 5 months ago

Selected Answer: D

https://www.oreilly.com/library/view/cloud-native-infrastructure/9781491984291/ch01.html

upvoted 2 times

hovnival 5 months ago

Selected Answer: D

Folks I ran the same questions again in Copilot and week after it says correct answer is D. I am very sorry about that Apologies. Confirmed by ChatGPT answer is D. It is a bit frustrating that copilot is not consistent in aswer as I did copy paste like last time.

upvoted 2 times

hovnival 5 months, 1 week ago

Selected Answer: A

Copilot says A

upvoted 1 times

JBangura 6 months, 2 weeks ago

Selected Answer: D

https://lists.cncf.io/g/cncf-toc/message/1636

upvoted 2 times

EzBL 6 months, 2 weeks ago

Selected Answer: D

Availability is highly desirable, but it's a consequence of achieving resiliency through proper design and implementation.

upvoted 2 times

7a9e5e5 6 months, 3 weeks ago

I am agree with correct response is A. High Availability is so principal to deploy Cloud Native aplications and it is an crucial to keep the aplication without interruptions. Agility is important but not most importan than Availability.

upvoted 1 times

unkun 8 months ago

It should Be A

upvoted 2 times

majkisermi98 8 months, 2 weeks ago

Selected Answer: A

Answer is wrong, it should be A. Availability is a crucial principle, Agility is not
upvoted 2 times

Question #42 *Topic 1*

What does CNCF stand for?

A. Cloud Native Community Foundation

B. Cloud Native Computing Foundation

C. Cloud Neutral Computing Foundation

D. Cloud Neutral Community Foundation

Correct Answer: *B*

Question #43 *Topic 1*

Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called:

A. namespaces

B. containers

C. hypervisors

D. cgroups

Correct Answer: *A*

Community vote distribution

A (100%)

SeaH0rse66 4 months, 2 weeks ago

Selected Answer: A

https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/
"Namespaces are intended for use in environments with many users spread across multiple teams, or projects. For clusters with a few to tens of users, you should not need to create or think about namespaces at all. Start using namespaces when you need the features they provide."
upvoted 2 times

phcunha 6 months, 2 weeks ago

Think of namespaces in Kubernetes as different floors in a large office building. Each floor represents a separate workspace for different teams o departments within the organization. Just as each floor has its own set of offices, meeting rooms, and facilities, each namespace in Kubernetes h its own set of resources, configurations, and access controls, providing isolation and organization within the cluster.
upvoted 1 times

unkun 8 months ago

https://jamesdefabia.github.io/docs/user-guide/namespaces/

Answer is correct "A"
upvoted 2 times

Question #44

Topic 1

What component enables end users, different parts of the Kubernetes cluster, and external components to communicate with one another?

A. kubectl

B. AWS Management Console

C. Kubernetes API

D. Google Cloud SDK

Correct Answer: *C*

Community vote distribution

C (100%)

SeaH0rse66 4 months, 2 weeks ago

Selected Answer: C

The Kubernetes API serves as the primary interface for communication between end users, different parts of the Kubernetes cluster, and external components.

upvoted 2 times

Question #45

Topic 1

Which command will list the resource types that exist within a cluster?

A. kubectl api-resources

B. kubectl get namespaces

C. kubectl api-versions

D. curl https://kubectrl/namespaces

Correct Answer: *A*

Community vote distribution

A (100%)

vspringe 2 weeks ago

Selected Answer: A

kubectl api-resources

upvoted 1 times

Question #46 *Topic 1*

Which of these components is part of the Kubernetes Control Plane?

A. coredns

B. cloud-controller-manager

C. kube-proxy

D. kubelet

Correct Answer: *B*

unkun 8 months ago

https://kubernetes.io/docs/concepts/architecture/cloud-controller/

Answer is correct

upvoted 2 times

Question #47 *Topic 1*

Which of the following systems is NOT compatible with the CRI runtime interface standard?

A. CRI-0

B. dockershim

C. systemd

D. containerd

Correct Answer: *C*

Community vote distribution

C (100%)

hovnival 5 months, 1 week ago

Selected Answer: C

CRI - Container Runtime Interface. systemd is not container runtime interface.

upvoted 2 times

JBangura 6 months, 2 weeks ago

https://kubernetes.io/docs/tasks/administer-cluster/migrating-from-dockershim/check-if-dockershim-removal-affects-you/

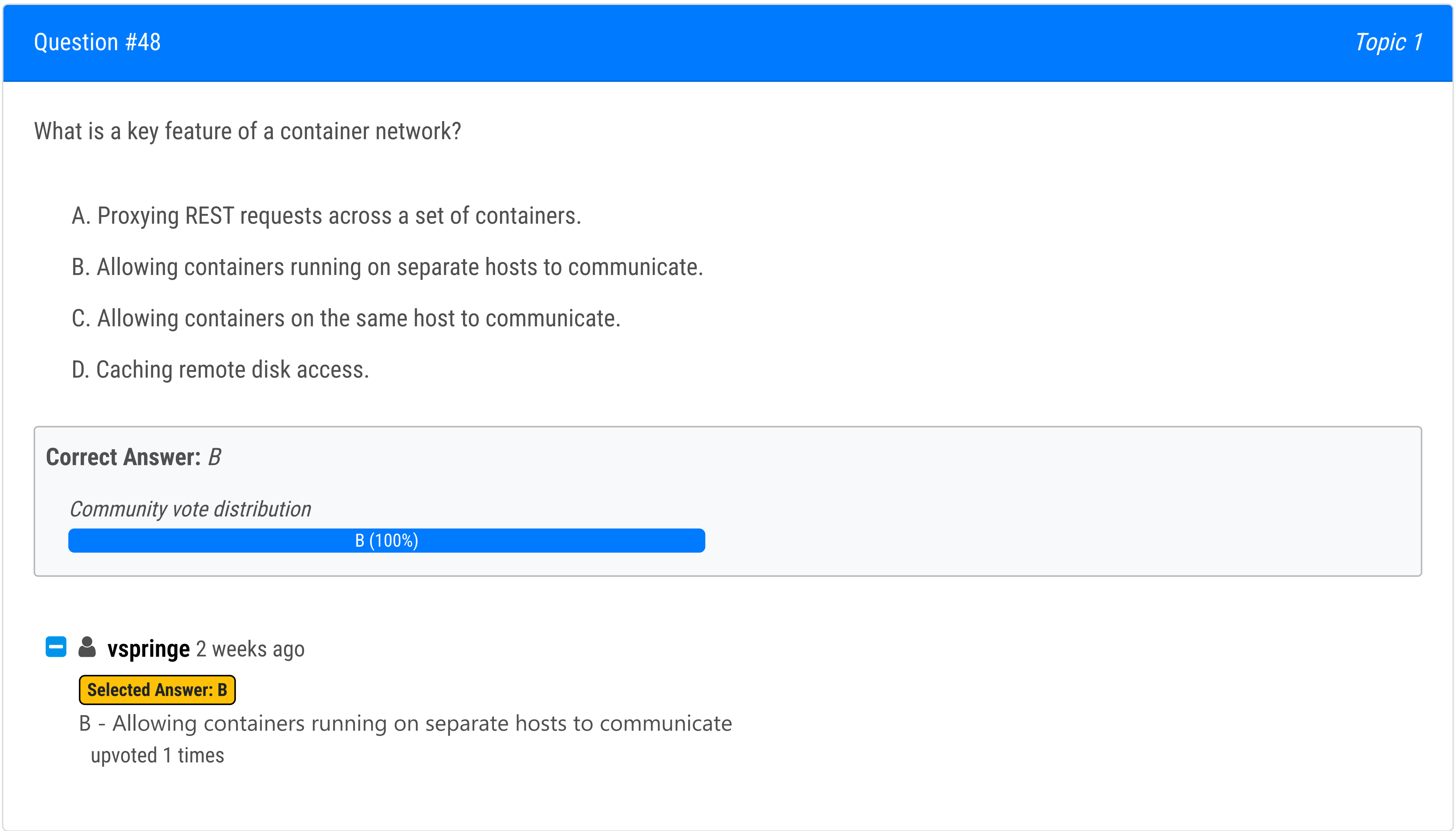
upvoted 1 times

omerco61 7 months, 3 weeks ago

Selected Answer: C

systemd: systemd is not a container runtime. It is an initialization system and service manager for Linux.

upvoted 3 times



Question #48 *Topic 1*

What is a key feature of a container network?

A. Proxying REST requests across a set of containers.

B. Allowing containers running on separate hosts to communicate.

C. Allowing containers on the same host to communicate.

D. Caching remote disk access.

Correct Answer: *B*

Community vote distribution

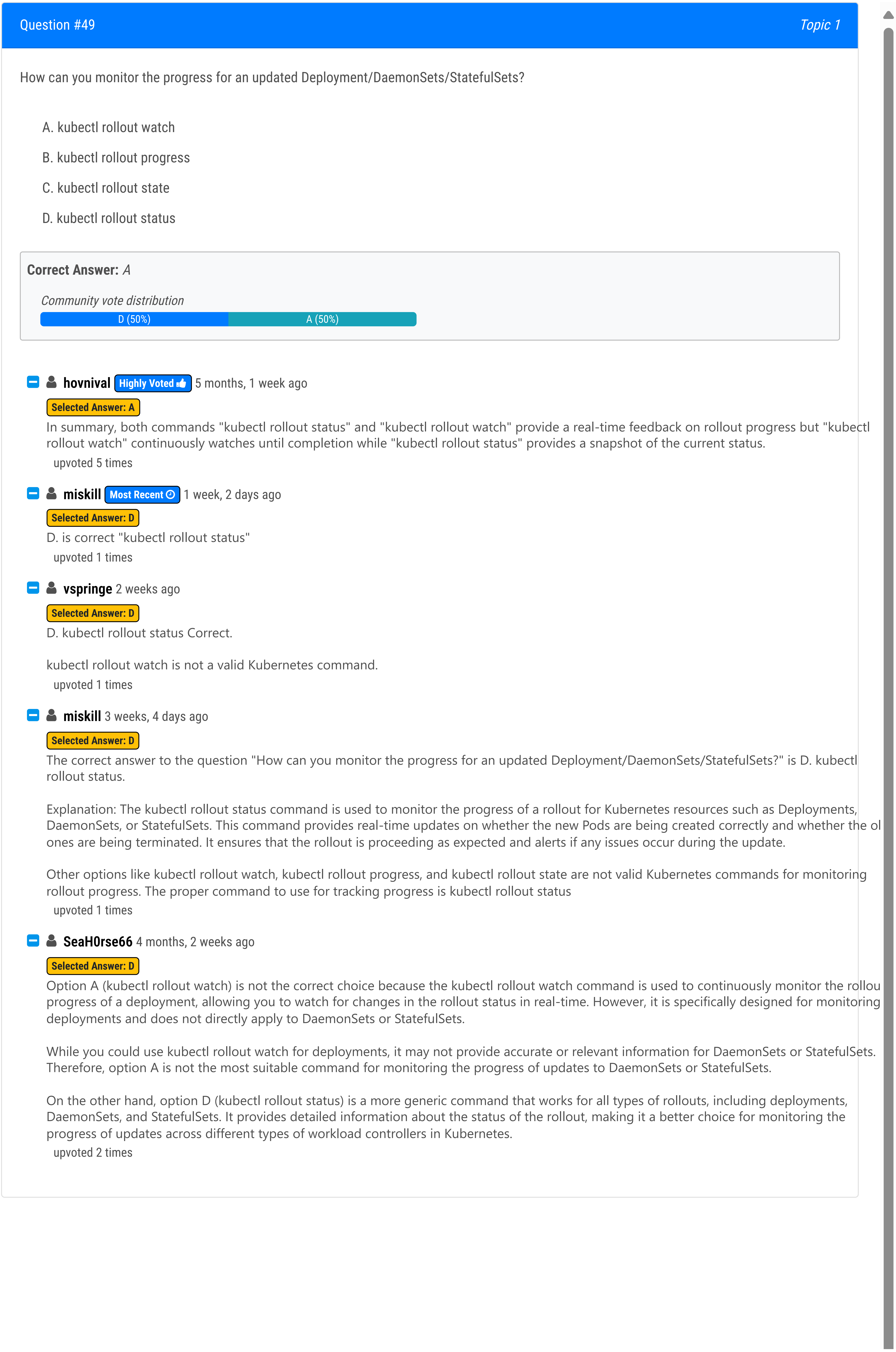
B (100%)

vspringe 2 weeks ago

Selected Answer: B

B - Allowing containers running on separate hosts to communicate

upvoted 1 times



Question #49 *Topic 1*

How can you monitor the progress for an updated Deployment/DaemonSets/StatefulSets?

A. kubectl rollout watch

B. kubectl rollout progress

C. kubectl rollout state

D. kubectl rollout status

Correct Answer: *A*

Community vote distribution

D (50%) A (50%)

hovnival Highly Voted 👍 5 months, 1 week ago

Selected Answer: A

In summary, both commands "kubectl rollout status" and "kubectl rollout watch" provide a real-time feedback on rollout progress but "kubectl rollout watch" continuously watches until completion while "kubectl rollout status" provides a snapshot of the current status.

upvoted 5 times

miskill Most Recent 🕘 1 week, 2 days ago

Selected Answer: D

D. is correct "kubectl rollout status"

upvoted 1 times

vspringe 2 weeks ago

Selected Answer: D

D. kubectl rollout status Correct.

kubectl rollout watch is not a valid Kubernetes command.

upvoted 1 times

miskill 3 weeks, 4 days ago

Selected Answer: D

The correct answer to the question "How can you monitor the progress for an updated Deployment/DaemonSets/StatefulSets?" is D. kubectl rollout status.

Explanation: The kubectl rollout status command is used to monitor the progress of a rollout for Kubernetes resources such as Deployments, DaemonSets, or StatefulSets. This command provides real-time updates on whether the new Pods are being created correctly and whether the ol ones are being terminated. It ensures that the rollout is proceeding as expected and alerts if any issues occur during the update.

Other options like kubectl rollout watch, kubectl rollout progress, and kubectl rollout state are not valid Kubernetes commands for monitoring rollout progress. The proper command to use for tracking progress is kubectl rollout status

upvoted 1 times

SeaH0rse66 4 months, 2 weeks ago

Selected Answer: D

Option A (kubectl rollout watch) is not the correct choice because the kubectl rollout watch command is used to continuously monitor the rollou progress of a deployment, allowing you to watch for changes in the rollout status in real-time. However, it is specifically designed for monitoring deployments and does not directly apply to DaemonSets or StatefulSets.

While you could use kubectl rollout watch for deployments, it may not provide accurate or relevant information for DaemonSets or StatefulSets. Therefore, option A is not the most suitable command for monitoring the progress of updates to DaemonSets or StatefulSets.

On the other hand, option D (kubectl rollout status) is a more generic command that works for all types of rollouts, including deployments, DaemonSets, and StatefulSets. It provides detailed information about the status of the rollout, making it a better choice for monitoring the progress of updates across different types of workload controllers in Kubernetes.

upvoted 2 times

Question #50 *Topic 1*

What is the goal of load balancing?

A. Automatically measure request performance across instances of an application.

B. Automatically distribute requests across different versions of an application.

C. Automatically distribute instances of an application across the cluster.

D. Automatically distribute requests across instances of an application.

Correct Answer: *D*

Community vote distribution

D (100%)

vspringe 2 weeks ago

Selected Answer: D

D. Automatically distribute requests across instances of an application.

upvoted 1 times

Question #51 *Topic 1*

How are ReplicaSets and Deployments related?

A. Deployments manage ReplicaSets and provide declarative updates to Pods.

B. ReplicaSets manage stateful applications, Deployments manage stateless applications.

C. Deployments are runtime instances of ReplicaSets.

D. ReplicaSets are subsets of Jobs and CronJobs which use imperative Deployments.

Correct Answer: *A*

Community vote distribution

A (100%)

SeaH0rse66 4 months, 2 weeks ago

Selected Answer: A

Deployments are higher-level abstractions in Kubernetes that manage ReplicaSets.ReplicaSets, on the other hand, are lower-level controllers responsible for maintaining a specified number of identical Pods to ensure high availability and fault tolerance. ReplicaSets are used by Deployments to manage the lifecycle of Pods, including scaling, rolling updates, and maintaining a desired number of replicas.

upvoted 2 times

AzureDP900 9 months, 4 weeks ago

A is correct

upvoted 2 times

LoloMaceto 10 months ago

Selected Answer: A

A is correct

upvoted 2 times

nvtienanh 1 year ago

Selected Answer: A

A is correct

upvoted 2 times

mazabel 1 year, 1 month ago

The correct is B

upvoted 1 times

Question #52 *Topic 1*

What factors influence the Kubernetes scheduler when it places Pods on nodes?

A. Pod memory requests, node taints, and Pod affinity.

B. Pod labels, node labels, and request labels.

C. Node taints, node level, and Pod priority.

D. Pod priority, container command, and node labels.

Correct Answer: *A*