

Linux Foundation

Cilium-Associate

Cilium Certified Associate (CCA)



For More Information – Visit link below:

<https://www.examsboost.com/>

Product Version

- ✓ Up to Date products, reliable and verified.
- ✓ Questions and Answers in PDF Format.

Latest Version: 6.0

Question: 1

Which one of the following service mesh features and use cases is natively supported by Cilium?

- A. API request limiting
- B. API authorization
- C. Layer 7 Load Balancing of gRPC
- D. Fault injection, such as injection of delay

Answer: A

Explanation:

The intended answer is A because API request limiting corresponds to rate limiting, which Cilium identifies as a core Layer 7 traffic-management capability. Cilium combines its eBPF datapath with Envoy for application-layer processing. The official Service Mesh documentation expressly includes rate limiting among the functions that must understand protocols such as HTTP, REST, gRPC, and WebSocket. It is therefore not merely packet-rate policing at Layer 3 or Layer 4; it can be applied with application-protocol context.

However, this question is no longer valid as a strict single-answer item. Current Cilium documentation also describes proxy-based Layer 7 load balancing as useful for gRPC and provides an Envoy-backed implementation for Kubernetes Services. Consequently, option C is also supportable under the current product documentation, although the feature is identified as beta. API authorization and fault-delay injection are not presented as equivalent first-class Cilium Service Mesh use cases in the cited feature overview.

For certification-bank purposes, retain A as the intended answer, but revise option C or qualify it to restore a unique correct choice.

Official references

Service Mesh; Proxy Load Balancing for Kubernetes Services.

Study Guide topic: Service Mesh.

Question: 2

Please review the output of `cilium status` below and answer the question that follows. Please note, the output has been modified for accessibility purposes.

```

Cilium: OK
Operator: OK
Envoy DaemonSet: disabled (using embedded mode)
Hubble Relay: OK
ClusterMesh: disabled

Deployment hubble-ui Desired: 1, Ready: 1/1, Available: 1/1
Deployment hubble-relay Desired: 1, Ready: 1/1, Available: 1/1
Deployment cilium-operator Desired: 1, Ready: 1/1, Available: 1/1
DaemonSet cilium Desired: 3, Ready: 3/3, Available: 3/3
Containers: cilium Running: 3
             hubble-ui Running: 1
Deployment hubble-ui Desired: 1, Ready: 1/1, Available: 1/1
Deployment hubble-relay Desired: 1, Ready: 1/1, Available: 1/1
Deployment cilium-operator Desired: 1, Ready: 1/1, Available: 1/1
DaemonSet cilium Desired: 3, Ready: 3/3, Available: 3/3
Containers: cilium Running: 3
             hubble-ui Running: 1
             hubble-relay Running: 1
             cilium-operator Running: 1
Cluster Pods: 6/6 managed by Cilium
Helm chart version: 1.14.1
Image versions cilium
quay.io/cilium/cilium:v1.14.1@sha256:edc1d05ea1365c4a8f6ac6982247d5c145181704894bb698619c3827b6963a72: 3
             hubble-ui quay.io/cilium/hubble-
backend:v0.12.0@sha256:8a79a1aad4fc9c2aa2b3e4379af0af872a89fcec9d99e117188190671c66fc2e: 1
             hubble-relay quay.io/cilium/hubble-
relay:v1.14.1@sha256:db30e85a7abc10589ce2a97d61ee18696a03dc5ea04d44b4d836d88bd75b59d8: 1
             cilium-operator quay.io/cilium/operator-
generic:v1.14.1@sha256:e061de0a930534c7e3f8feda8330976367971238ccafff42659f104effd4b5f7: 1

```

Question 2 cilium status exhibit

Assume the cluster is healthy. What is correct about the `ci 1 ium status` command output above?

- A. The Kubernetes cluster where Cilium has been deployed should consist of four nodes.
- B. When specific Cilium features require Layer 7 processing, the Cilium agent starts an Envoy proxy as a separate process within the Cilium agent pod.
- C. The component that allows you to query multiple Hubble instances simultaneously and aggregate the results is unhealthy.
- D. Cilium has been installed and configured in a multi-cluster deployment model to provide load balancing and service discovery.

Answer: B

Explanation:

The status output shows Envoy DaemonSet: disabled (using embedded mode). In embedded mode, Cilium starts Envoy as a separate process inside the Cilium agent pod when a feature requiring Layer 7 processing—such as an L7 network policy, protocol visibility, Ingress, or Gateway API—is used. This is precisely the behavior described in option B.

The remaining choices contradict the exhibit. The Cilium DaemonSet has a desired count of three, which ordinarily indicates three schedulable nodes running Cilium, not four. Hubble Relay: OK means the aggregation component is healthy. Hubble Relay connects to the Hubble instances associated with Cilium agents and presents a multi-node API to clients such as Hubble

CLI and Hubble UI. Finally, ClusterMesh: disabled rules out the claim that the installation is currently configured for Cilium Cluster Mesh service discovery and cross-cluster load balancing. Cilium can alternatively run Envoy through the independently life-cycled cilium-envoy DaemonSet. That is not the deployment displayed here because the DaemonSet is explicitly disabled and embedded mode is reported.

Official references

Envoy deployment modes; Hubble internals.

Study Guide topic: Architecture.

Question: 3

Which question does Hubble provide the information to answer?

- A. What is the configuration of Cilium B6P?
- B. Which container database query ran the longest?
- C. Which containers have the highest CPU utilization?
- D. Which services had connections blocked due to network policy?

Answer: D

Explanation:

Hubble provides network and security flow observability derived from Cilium's datapath. Its flow records include source and destination workloads, namespaces, identities, ports, protocols, forwarding verdicts, and drop reasons. Hubble can therefore identify services or workloads whose connections were rejected by network policy. The Hubble CLI supports filtering by verdict, such as --verdict DROPPED, and official examples show events labeled Policy denied DROPPED. These capabilities directly answer the question posed in option D.

Option B, which is marked as correct in the supplied bank, is not supported by the stated functionality. Hubble may expose network-layer or supported application-layer metadata, including DNS and HTTP information when Layer 7 visibility is configured, but it is not a database query profiler and cannot determine which internal SQL statement ran longest merely from ordinary Hubble flows. Similarly, CPU utilization is a workload-resource metric normally obtained through Kubernetes metrics, Prometheus, or another monitoring system. Cilium BGP configuration is examined through Cilium configuration and BGP status commands rather than inferred from Hubble network flows.

The supplied answer key should therefore be corrected from B to D.

Official references

Inspecting Network Flows with the Hubble CLI; Hubble internals.

Study Guide topic: Network Observability.

Question: 4

A Kubernetes cluster is not currently running Cilium as a CNI, but the user would like to benefit from Hubble's observability capabilities on your cluster. Which one of the following options is NOT possible?

- A. Migrating to Cilium and Hubble by reconfiguring `/etc/cni/net .6/` to point to Cilium and restarting the nodes, accepting an outage.
- B. Install Hubble on the cluster without Cilium. Hubble can be deployed in a standalone model by downloading and installing the Hubble binary
- C. Install Cilium and Hubble on top of the cluster in a CNI chaining model, without disrupting the existing CNI.
- D. Migrating to Cilium and Hubble on a node-by-node basis, without requiring a complete cluster outage, by using Dual Overlays.

Answer: B

Explanation:

Hubble is Cilium's integrated observability layer and consumes flow events produced by Cilium's eBPF datapath and embedded Hubble servers. The Hubble CLI is only a client; downloading its binary does not install a standalone datapath or create flow data on a cluster that lacks Cilium.

Option B is therefore the operation that is not possible.

The other approaches represent recognized Cilium deployment or migration models. A direct migration can replace the CNI configuration and recycle workloads or nodes, although a naive cluster-wide transition can disrupt connectivity. CNI chaining allows Cilium to operate with another CNI: the existing plugin continues to provide basic connectivity and IP address management, while Cilium attaches eBPF programs to the created interfaces to provide visibility, policy, and other functions. Cilium also documents migration through dual overlays. In that model, the old and new networks coexist temporarily, nodes are moved in a controlled sequence, and workloads attached to either overlay retain connectivity when the documented addressing and routing requirements are met.

The supplied bank incorrectly marks A. The verified answer is B because Hubble requires Cilium-managed observability data.

Official references

Setting up Hubble Observability; CNI Chaining; Migrating a cluster to Cilium.

Study Guide topic: Installation and Configuration.

Question: 5

Which command is used to enable logging at the debug log level of Cilium agents?

- A. `cilium log level --set=debug`
- B. `cilium logging.level=debug`
- C. `cilium config set debug true`
- D. `cilium logging debug`

Answer: C

Explanation:

`cilium config set debug true` follows the supported Cilium CLI configuration syntax and sets the debug configuration key to true. The `cilium config set` command accepts a key/value pair and, by default, restarts the Cilium pods so that the changed configuration is applied. The Cilium configuration documentation defines debug as the setting that enables full debug mode. This increases agent logging verbosity and causes eBPF programs to emit additional visibility events for diagnostic use.

Options A, B, and D do not match documented Cilium CLI command structures. There is no `cilium log level --set=debug` command in the Cilium CLI hierarchy, and neither `cilium logging.level=debug` nor `cilium logging debug` is valid configuration syntax. A Helm-managed installation may also enable debugging through the chart value `debug.enabled=true`, but that does not make any of the alternative commands correct.

Debug mode should be enabled deliberately because it increases log volume and may generate additional datapath visibility information. After troubleshooting, operators should normally restore the previous setting to avoid unnecessary operational overhead.

The supplied answer key incorrectly identifies B. The verified answer is C.

Official references

Cilium configuration; Cilium CLI `config set`; Helm values.

Study Guide topic: Installation and Configuration.

Thank You for Trying Our Product

For More Information – **Visit link below:**

<https://www.examsboost.com/>

15 USD Discount Coupon Code:

G74JA8UF

FEATURES

- ✓ **90 Days Free Updates**
- ✓ **Money Back Pass Guarantee**
- ✓ **Instant Download or Email Attachment**
- ✓ **24/7 Live Chat Support**
- ✓ **PDF file could be used at any Platform**
- ✓ **50,000 Happy Customer**



Visit us at: <https://www.examsboost.com/test/cilium-associate>