

Latest Linux Foundation PCA Questions in Three Different Formats



Linux Foundation PCA Prometheus Certified Associate (PCA)

**Questions & Answers PDF
(Demo Version – Limited Content)**

For More Information – Visit link below:

<https://p2pexam.com/>

Visit us at: <https://p2pexam.com/pca>

As the authoritative provider of PCA actual exam, we always pursue high pass rate compared with our peers to gain more attention from those potential customers. We guarantee that if you follow the guidance of our PCA learning materials, you will pass the exam without a doubt and get a certificate. Our PCA Exam Practice is carefully compiled after many years of practical effort and is adaptable to the needs of the PCA exam.

Icertmaster's Linux Foundation PCA exam training materials' simulation is particularly high. You can encounter the same questions in the real exam. This only shows that the ability of our IT elite team is really high. Now many ambitious IT staff to make their own configuration files compatible with the market demand, to realize their ideals through these hot IT exam certification. Achieved excellent results in the Linux Foundation PCA Exam. With the Linux Foundation PCA exam training of Icertmaster, the door of the dream will open for you.

>> **PCA Latest Braindumps Book** <<

Splendid PCA Exam Materials: Prometheus Certified Associate Exam Present You a brilliant Training Dump - Icertmaster

The customer is God. PCA learning dumps provide all customers with high quality after-sales service. After your payment is successful, we will dispatch a dedicated IT staff to provide online remote assistance for you to solve problems in the process of download and installation. During your studies, PCA study tool will provide you with efficient 24-hour online services. You can email us anytime, anywhere to ask any questions you have about our PCA Study Tool. At the same time, our industry experts will continue to update and supplement PCA test question according to changes in the exam outline, so that you can concentrate on completing

the review of all exam content without having to pay attention to changes in the outside world.

Linux Foundation Prometheus Certified Associate Exam Sample Questions (Q29-Q34):

NEW QUESTION # 29

```
http_requests_total{verb="POST"} 30
```

```
http_requests_total{verb="GET"} 30
```

What is the issue with the metric family?

- A. Unit is missing in the `http_requests_total` metric name.
- B. The value represents two different things across the dimensions: code and verb.
- C. Metric names are missing a prefix to indicate which application is exposing the query.
- D. verb label content should be normalized to lowercase.

Answer: A

Explanation:

Prometheus metric naming best practices require that every metric name include a unit suffix that indicates the measurement type, where applicable. The unit should follow the base name, separated by an underscore, and must use base SI units (for example, `_seconds`, `_bytes`, `_total`, etc.).

In the case of `http_requests_total`, while the metric correctly includes the `_total` suffix-indicating it is a counter-it lacks a base unit of measurement (such as time, bytes, or duration). However, for event counters, `_total` is itself considered the unit, representing "total occurrences" of an event. Thus, the naming would be acceptable in strict Prometheus terms, but if this metric were measuring something like duration, size, or latency, then including a specific unit would be mandatory.

However, since the question implies that the missing unit is the issue and not the label schema, the expected answer aligns with ensuring metric names convey measurable units when applicable.

Reference:

Prometheus documentation - Metric and Label Naming Conventions, Instrumentation Best Practices, and Metric Type Naming (Counters, Gauges, and Units) sections.

NEW QUESTION # 30

What is the name of the official *nix OS kernel metrics exporter?

- A. `node_exporter`
- B. `Prometheus_exporter`
- C. `metrics_exporter`
- D. `os_exporter`

Answer: A

Explanation:

The official Prometheus exporter for collecting system-level and kernel-related metrics from Linux and other UNIX-like operating systems is the Node Exporter.

The Node Exporter exposes hardware and OS metrics including CPU load, memory usage, disk I/O, network traffic, and kernel statistics. It is designed to provide host-level observability and serves data at the default endpoint `:9100/metrics` in the standard Prometheus exposition text format.

This exporter is part of the official Prometheus ecosystem and is widely deployed for infrastructure monitoring. None of the other listed options (`Prometheus_exporter`, `metrics_exporter`, or `os_exporter`) are official components of the Prometheus project.

Reference:

Verified from Prometheus documentation - Node Exporter Overview, System Metrics Collection, and Official Exporters List.

NEW QUESTION # 31

Which of the following signal belongs to symptom-based alerting?

- A. CPU usage
- B. Disk space
- C. **API latency**

- D. Database memory utilization

Answer: C

Explanation:

Symptom-based alerting focuses on user-visible problems or service-impacting symptoms rather than low-level resource metrics. In Prometheus and Site Reliability Engineering (SRE) practices, alerts should signal conditions that affect users' experience - such as high latency, request failures, or service unavailability - instead of merely reflecting internal resource states.

Among the options, API latency directly represents the performance perceived by end users. If API response times increase, it immediately impacts user satisfaction and indicates a possible service degradation.

In contrast, metrics like disk space, CPU usage, or database memory utilization are cause-based metrics - they may correlate with problems but do not always translate into observable user impact.

Prometheus alerting best practices recommend alerting on symptoms (via RED metrics - Rate, Errors, Duration) while using cause-based metrics for deeper investigation and diagnosis, not for immediate paging alerts.

Reference:

Verified from Prometheus documentation - Alerting Best Practices, Symptom vs. Cause Alerting, and RED/USE Monitoring Principles sections.

NEW QUESTION # 32

What are the four golden signals of monitoring as defined by Google's SRE principles?

- A. Utilization, Load, Disk, Network
- B. Requests, CPU, Memory, Latency
- C. Availability, Logging, Errors, Throughput
- **D. Traffic, Errors, Latency, Saturation**

Answer: D

Explanation:

The Four Golden Signals-Traffic, Errors, Latency, and Saturation-are key service-level indicators defined by Google's Site Reliability Engineering (SRE) discipline.

Traffic: Demand placed on the system (e.g., requests per second).

Errors: Rate of failed requests.

Latency: Time taken to serve requests.

Saturation: How "full" the system resources are (CPU, memory, etc.).

Prometheus and its metrics-based model are ideal for capturing these signals.

NEW QUESTION # 33

What is a difference between a counter and a gauge?

- **A. Counters are only incremented, while gauges can go up and down.**
- B. Counters have no labels while gauges can have many labels.
- C. Counters change value on each scrape and gauges remain static.
- D. Counters and gauges are different names for the same thing.

Answer: A

Explanation:

The key difference between a counter and a gauge in Prometheus lies in how their values change over time. A counter is a cumulative metric that only increases-it resets to zero only when the process restarts. Counters are typically used for metrics like total requests served, bytes processed, or errors encountered. You can derive rates of change from counters using functions like `rate()` or `increase()` in PromQL.

A gauge, on the other hand, represents a metric that can go up and down. It measures values that fluctuate, such as CPU usage, memory consumption, temperature, or active session counts. Gauges provide a snapshot of current state rather than a cumulative total.

This distinction ensures proper interpretation of time-series trends and prevents misrepresentation of one-time or fluctuating values as cumulative metrics.

Reference:

Extracted and verified from Prometheus official documentation - Metric Types section explaining Counters and Gauges definitions

myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt,
myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, boldbranding in, www.stes.tyc.edu.tw,
myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt,
myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes