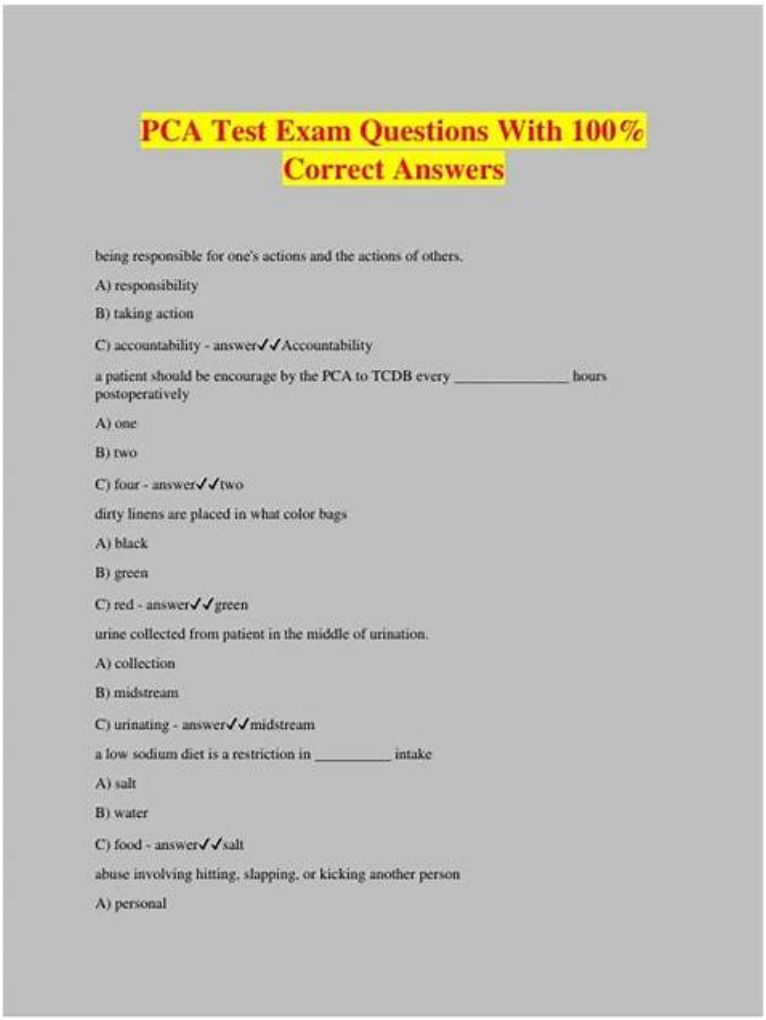
PCA Valid Mock Test | PCA Valid Test Topics



PCA Test Exam Questions With 100% Correct Answers

being responsible for one's actions and the actions of others.

A) responsibility

B) taking action

C) accountability - answer✓✓Accountability

a patient should be encourage by the PCA to TCDB every _____ hours postoperatively

A) one

B) two

C) four - answer✓✓two

dirty linens are placed in what color bags

A) black

B) green

C) red - answer✓✓green

urine collected from patient in the middle of urination.

A) collection

B) midstream

C) urinating - answer✓✓midstream

a low sodium diet is a restriction in _____ intake

A) salt

B) water

C) food - answer✓✓salt

abuse involving hitting, slapping, or kicking another person

A) personal

In such society where all people take the time so precious, choosing Exam-Killer to help you pass the Linux Foundation Certification PCA Exam is cost-effective. If you choose Exam-Killer, we promise that we will try our best to help you pass the exam and also provide you with one year free update service. If you fail the exam, we will give you a full refund.

Linux Foundation PCA Exam Syllabus Topics:

Topic	Details
Topic 1	• Observability Concepts: This section of the exam measures the skills of Site Reliability Engineers and covers the essential principles of observability used in modern systems. It focuses on understanding metrics, logs, and tracing mechanisms such as spans, as well as the difference between push and pull data collection methods. Candidates also learn about service discovery processes and the fundamentals of defining and maintaining SLOs, SLAs, and SLIs to monitor performance and reliability.
Topic 2	• Prometheus Fundamentals: This domain evaluates the knowledge of DevOps Engineers and emphasizes the core architecture and components of Prometheus. It includes topics such as configuration and scraping techniques, limitations of the Prometheus system, data models and labels, and the exposition format used for data collection. The section ensures a solid grasp of how Prometheus functions as a monitoring and alerting toolkit within distributed environments.

Topic 3	• PromQL: This section of the exam measures the skills of Monitoring Specialists and focuses on Prometheus Query Language (PromQL) concepts. It covers data selection, calculating rates and derivatives, and performing aggregations across time and dimensions. Candidates also study the use of binary operators, histograms, and timestamp metrics to analyze monitoring data effectively, ensuring accurate interpretation of system performance and trends.
Topic 4	• Alerting and Dashboarding: This section of the exam assesses the competencies of Cloud Operations Engineers and focuses on monitoring visualization and alert management. It covers dashboarding basics, alerting rules configuration, and the use of Alertmanager to handle notifications. Candidates also learn the core principles of when, what, and why to trigger alerts, ensuring they can create reliable monitoring dashboards and proactive alerting systems to maintain system stability.
Topic 5	• Instrumentation and Exporters: This domain evaluates the abilities of Software Engineers and addresses the methods for integrating Prometheus into applications. It includes the use of client libraries, the process of instrumenting code, and the proper structuring and naming of metrics. The section also introduces exporters that allow Prometheus to collect metrics from various systems, ensuring efficient and standardized monitoring implementation.

PCA Valid Test Topics, Certification PCA Exam Cost

Our services before, during and after the clients use our PCA certification material are considerate. Before the purchase, the clients can download and try out our PCA learning file freely. During the clients use our products they can contact our online customer service staff to consult the problems about our products. After the clients use our PCA Prep Guide dump if they can't pass the test smoothly they can contact us to require us to refund them in full and if only they provide the failure proof we will refund them at once. Our company gives priority to the satisfaction degree of the clients and puts the quality of the service in the first place.

Linux Foundation Prometheus Certified Associate Exam Sample Questions (Q30-Q35):

NEW QUESTION # 30
Which PromQL expression computes the rate of API Server requests across the different cloud providers from the following metrics?
apiserver_request_total{job="kube-apiserver", instance="192.168.1.220:6443", cloud="aws"} 1
apiserver_request_total{job="kube-apiserver", instance="192.168.1.121:6443", cloud="gcloud"} 5

- A. rate(apiserver_request_total{job="kube-apiserver"}[5m]) by (cloud)
- B. sum by (cloud) (apiserver_request_total{job="kube-apiserver"})
- C. rate(sum by (cloud)(apiserver_request_total{job="kube-apiserver"})[5m])
- D. sum by (cloud)(rate(apiserver_request_total{job="kube-apiserver"}[5m]))

Answer: D

Explanation:
The rate() function computes the per-second increase of a counter metric over a specified range, while sum by (label) aggregates those rates across dimensions - in this case, the cloud label.
The correct query is:
sum by (cloud)(rate(apiserver_request_total{job="kube-apiserver"}[5m])) This expression:
Calculates the rate of increase in API requests per second for each instance.
Groups and sums those rates by cloud, giving the total request rate per cloud provider.
Option A incorrectly places by (cloud) after rate(), which is not valid syntax.
Option B returns raw counter totals (not rates).
Option D incorrectly applies rate() after aggregation, which distorts the calculation since rate() must operate on individual time series before aggregation.
Reference:
Verified from Prometheus documentation - rate() Function, Aggregation Operators, and Querying Counters Across Labels sections.

NEW QUESTION # 31
How would you correctly name a metric that provides metadata information about the binary?

- A. app_metadata
- B. app_build_info
- C. app_build_desc
- D. app_build

Answer: B

Explanation:
The Prometheus naming convention for metrics that expose build or version information about an application binary uses the _info suffix. The standard pattern is:
<application>_build_info
This metric typically includes constant labels such as version, revision, branch, and goversion to describe the build environment.
For example:
app_build_info{version="1.2.3", revision="abc123", goversion="go1.22"} 1 This approach follows the official Prometheus instrumentation guidelines, where metrics ending in _info convey metadata or constant characteristics about the running process.
The other options do not conform to the Prometheus best practice of suffix-based semantic naming.
Reference:
Extracted and verified from Prometheus documentation - Metric Naming Conventions, Exposing Build Information, and Standard _info Metrics sections.

NEW QUESTION # 32
Where does Prometheus store its time series data by default?

- A. In-memory only.
- B. In an embedded TSDB on local disk.
- C. In etcd.
- D. In an external database such as InfluxDB.

Answer: B

Explanation:
By default, Prometheus stores its time series data in a local, embedded Time Series Database (TSDB) on disk. The data is organized in block files under the data/ directory inside Prometheus's storage path.
Each block typically covers two hours of data, containing chunks, index, and metadata files. Older blocks are compacted and deleted based on retention settings.

NEW QUESTION # 33
How can you use Prometheus Node Exporter?

- A. You can use it to instrument applications with metrics.
- B. You can use it to collect resource metrics from the application HTTP server.
- C. You can use it to probe endpoints over HTTP, HTTPS.
- D. You can use it to collect metrics for hardware and OS metrics.

Answer: D

Explanation:
The Prometheus Node Exporter is a core system-level exporter that exposes hardware and operating system metrics from *nix-based hosts. It collects metrics such as CPU usage, memory, disk I/O, filesystem space, network statistics, and load averages.
It runs as a lightweight daemon on each host and exposes metrics via an HTTP endpoint (default: :9100/metrics), which Prometheus scrapes periodically.
Key clarification:
It does not instrument applications (A).
It does not collect metrics directly from application HTTP endpoints (B).
It is unrelated to HTTP probing tasks - those are handled by the Blackbox Exporter (D).

Thus, the correct use of the Node Exporter is to collect and expose hardware and OS-level metrics for Prometheus monitoring.
Reference:
Extracted and verified from Prometheus documentation - Node Exporter Overview, Host-Level Monitoring, and Exporter Usage Best Practices sections.

NEW QUESTION # 34
Which field in alerting rules files indicates the time an alert needs to go from pending to firing state?

- A. timeout
- B. interval
- C. duration
- D. for

Answer: D

Explanation:
In Prometheus alerting rules, the for field specifies how long a condition must remain true continuously before the alert transitions from the pending to the firing state. This feature prevents transient spikes or brief metric fluctuations from triggering false alerts.
Example:
alert: HighRequestLatency
expr: http_request_duration_seconds_avg > 1
for: 5m
labels:
severity: warning
annotations:
description: "Request latency is above 1s for more than 5 minutes."
In this configuration, Prometheus evaluates the expression every rule evaluation cycle. The alert only fires if the condition (http_request_duration_seconds_avg > 1) remains true for 5 consecutive minutes. If it returns to normal before that duration, the alert resets and never fires.
This mechanism adds stability and noise reduction to alerting systems by ensuring only sustained issues generate notifications.
Reference:
Verified from Prometheus documentation - Alerting Rules Configuration Syntax, Pending vs. Firing States, and Best Practices for Alert Timing and Thresholds sections.

NEW QUESTION # 35
.....

In the matter of quality, our PCA practice engine is unsustainable with reasonable prices. Despite costs are constantly on the rise these years from all lines of industry, our PCA learning materials remain low level. That is because our company beholds customer-oriented tenets that guide our everyday work. The achievements of wealth or prestige is no important than your exciting feedback about efficiency and profession of our PCA Practice Engine. So our PCA practice materials are great materials you should be proud of and we are!

- New PCA Exam Notes □ Certification PCA Training □ PCA Torrent □ Search for 「 PCA 」 and obtain a free download on （ www.pdfvce.com ） □Test PCA Voucher
- Test PCA Voucher □ Examcollection PCA Questions Answers □ Examcollection PCA Questions Answers □ Search on [www.vce4dumps.com] for ▹ PCA ◃ to obtain exam materials for free download ✈Test PCA Voucher
- PCA Practice Exams □ Certification PCA Training □ PCA Test Question □ Search for ✔ PCA □✔□ and download it for free immediately on ➠ www.pdfvce.com □ □Examcollection PCA Questions Answers
- Test PCA Voucher □ PCA Test Question □ New Guide PCA Files □ Search for ▹ PCA ◃ on 「 www.pass4test.com 」 immediately to obtain a free download □Accurate PCA Answers
- building.lv, www.stes.tyc.edu.tw, 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, 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, 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, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, Disposable vapes