熱門的PCA證照考試|第一次嘗試輕鬆學習並通過考試和免費下載的PCA：Prometheus Certified Associate Exam

P.S. PDFExamDumps在Google Drive上分享了免費的2026 Linux Foundation PCA考試題庫：https://drive.google.com/open?id=1RbhAtSjP9dUS8UNiXpD-0umTV9VBYmGP

總體來說，PDFExamDumps 的模擬試題還是比較實用的，知識點也比較明確，據廣大考生反應，真正的 PCA 考題都是我們考題網裡面的原題，而且題目的答案也比較隱晦一些，不懂不明白那個知識。或沒有認真看題目，是不可能選到正確答案的，如果你通過我們的 Linux Foundation PCA 考題模擬，就能在 PCA 考試中輕鬆過關，讓自己更加接近成功之路。

Linux Foundation PCA 考試大綱：

主題	簡介
主題 1	• 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.
主題 2	• 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.
主題 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.

主題 4	• 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.
主題 5	• 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.

最新的 Cloud & Containers PCA 免費考試真題 (Q49-Q54):

問題 #49
Which function would you use to calculate the 95th percentile latency from histogram data?

- A. topk(0.95, http_request_duration_seconds)
- B. percentile(http_request_duration_seconds, 0.95)
- C. quantile_over_time(0.95, http_request_duration_seconds[5m])
- D. histogram_quantile(0.95, sum(rate(http_request_duration_seconds_bucket[5m])) by (le))

答案：D

解題說明：
To calculate a percentile (e.g., 95th percentile) from histogram data in Prometheus, the correct function is histogram_quantile(). It estimates quantiles based on cumulative bucket counts.
Example:
histogram_quantile(0.95, sum(rate(http_request_duration_seconds_bucket[5m])) by (le)) This computes the 95th percentile request duration across all observed instances over the last 5 minutes.

問題 #50
Which PromQL expression computes how many requests in total are currently in-flight for the following time series data?
apiserver_current_inflight_requests{instance="1"} 5
apiserver_current_inflight_requests{instance="2"} 7

- A. sum(apiserver_current_inflight_requests)
- B. max(apiserver_current_inflight_requests)
- C. min(apiserver_current_inflight_requests)
- D. sum_over_time(apiserver_current_inflight_requests[10m])

答案：A

解題說明：
In Prometheus, when you have multiple time series that represent the same type of measurement across different instances, the sum() aggregation operator is used to compute their total value.
Here, each instance (1 and 2) exposes the metric apiserver_current_inflight_requests, indicating the number of active API requests

currently being processed.
To find the total number of in-flight requests across all instances, the correct expression is:
sum(apiserver_current_inflight_requests)
This returns 5 + 7 = 12.
min() would return the lowest value (5).
max() would return the highest value (7).
sum_over_time() calculates the cumulative sum over a range vector, not the current value, so it's incorrect here.
Reference:
Verified from Prometheus documentation - Aggregation Operators and Summing Across Dimensions sections.

問題 #51
Which field in alerting rules files indicates the time an alert needs to go from pending to firing state?

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

答案：D

解題說明：
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.

問題 #52
Which PromQL statement returns the average free bytes of the filesystems over the last hour?

- A. avg(node_filesystem_avail_bytes[1h])
- B. sum_over_time(node_filesystem_avail_bytes[1h])
- C. sum(node_filesystem_avail_bytes[1h])
- D. avg_over_time(node_filesystem_avail_bytes[1h])

答案：D

解題說明：
The avg_over_time() function calculates the average value of a time series over a specified range vector. It is used to measure how a gauge metric (like available filesystem bytes) behaves over time rather than at a single instant.
For example:
avg_over_time(node_filesystem_avail_bytes[1h])
This query returns the average amount of available filesystem space observed across all samples within the last hour for each time series.
By contrast:
avg() performs aggregation across different series at a single point, not over time.
sum() and sum_over_time() compute totals rather than averages.

Thus, only avg_over_time() provides the correct temporal average.
Reference:
Extracted and verified from Prometheus documentation - Range Vector Functions, avg_over_time() Definition, and Working with Gauge Metrics Over Time sections.

問題 #53
Given the metric prometheus_tsdb_lowest_timestamp_seconds, how do you know in which month the lowest timestamp of your Prometheus TSDB belongs?

- A. month(prometheus_tsdb_lowest_timestamp_seconds)
- B. (time() - prometheus_tsdb_lowest_timestamp_seconds) / 86400
- C. format_date(prometheus_tsdb_lowest_timestamp_seconds,"%M")
- D. prometheus_tsdb_lowest_timestamp_seconds % month

答案：B

解題說明：
The metric prometheus_tsdb_lowest_timestamp_seconds provides the oldest stored sample timestamp in Prometheus's local TSDB (in Unix epoch seconds). To determine the age or approximate date of this timestamp, you compare it with the current time (using time() in PromQL).
The expression:
(time() - prometheus_tsdb_lowest_timestamp_seconds) / 86400
converts the difference between the current time and the oldest timestamp from seconds into days (1 day = 86,400 seconds). This gives the number of days since the earliest sample was stored, allowing you to infer the time range and approximate month manually.
The other options are invalid because PromQL does not support direct date formatting (format_date) or month() extraction functions.
Reference:
Extracted and verified from Prometheus documentation - TSDB Internal Metrics, Time Functions in PromQL, and Using time() for Relative Calculations.

問題 #54
.....

- PCA考試備考經驗 □ PCA測試題庫 □ 最新PCA考題 □ 打開[www.vcesoft.com]搜尋⇒ PCA ⇐以免費下載考試資料PCA題庫更新資訊
- studentcenter.iodacademy.id, 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, credennz.in, emmaklewis.sites.gettysburg.edu, www.stes.tyc.edu.tw, a1technoclasses.com, 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, Disposable vapes

BONUS!!! 免費下載PDFExamDumps PCA考試題庫的完整版：https://drive.google.com/open?id=1RbhAtSjP9dUS8UNiXpD-0umTV9VBYmGP