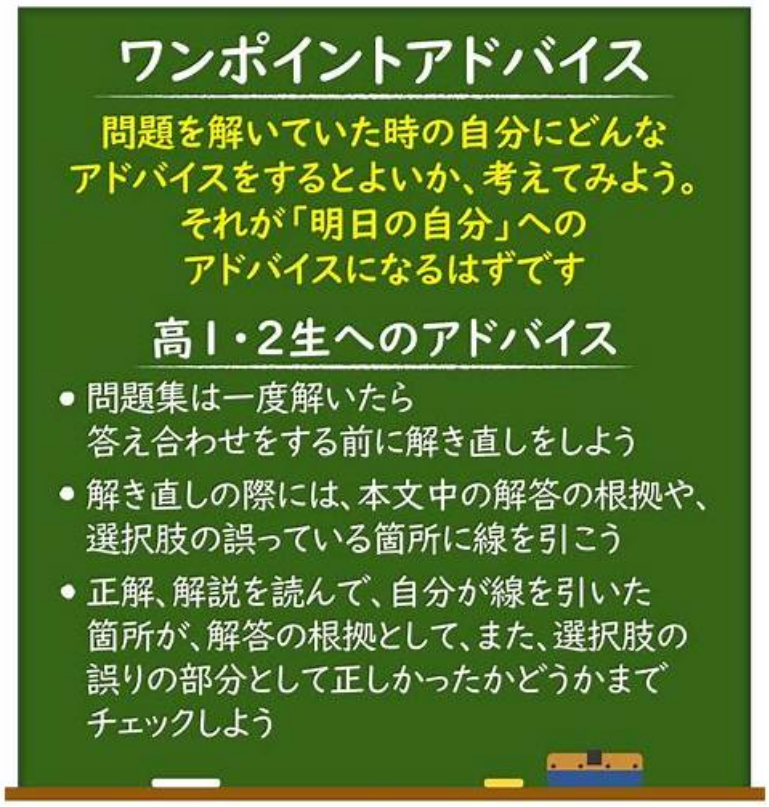
Linux Foundation PCA Exam | PCAミシュレーション問題 -信頼できるプランフォームPCA資格トレーリング



さらに、Pass4Test PCAダンプの一部が現在無料で提供されています：https://drive.google.com/open?id=1ndRYCEfl9BrACB2iMAKCNOTsyo1vdIaj

PCA試験問題の継続的な刷新により、当社は大きな市場シェアを占めています。強力な研究センターを構築し、PCAトレーニングガイドでより良い仕事をするために強力なチームを所有しています。これまで、PCA学習教材に関する多くの特許を取得しています。一方で、当社Linux Foundationは改修の恩恵を受けています。お客様は当社の製品を選択する可能性が高くなります。一方、私たちが投資したお金は有意義なものであり、PCA試験の新しい学習スタイルを刷新するのに役立ちます。

Linux Foundation PCA 認定試験の出題範囲：

トピック	出題範囲
トピック 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.
トピック 2	• 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.

トピック 3	• 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.
トピック 4	• 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.
トピック 5	• 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.

Linux Foundation Prometheus Certified Associate Exam 認定 PCA 試験問題 (Q27-Q32):

質問 # 27
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. sum by (cloud)(rate(apiserver_request_total{job="kube-apiserver"}[5m]))
- B. sum by (cloud) (apiserver_request_total{job="kube-apiserver"})
- C. rate(apiserver_request_total{job="kube-apiserver"}[5m]) by (cloud)
- D. rate(sum by (cloud)(apiserver_request_total{job="kube-apiserver"})[5m])

正解：A

解説：
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.

質問 # 28
Which of the following is an invalid @ modifier expression?

- A. sum(http_requests_total{method="GET"} @ 1609746000)
- B. go_goroutines @ start()
- C. go_goroutines @ end()
- D. sum(http_requests_total{method="GET"}) @ 1609746000

正解：A

解説：
The @ modifier in PromQL allows querying data as it existed at a specific point in time rather than the evaluation time. It can be applied after a selector or an entire expression, but the syntax rules are strict.
 go_goroutines @ start() → Valid; queries value at the start of the evaluation range.
 sum(http_requests_total{method="GET"}) @ 1609746000 → Valid; applies the modifier after the full expression.
 go_goroutines @ end() → Valid; queries value at the end of the evaluation range.
 sum(http_requests_total{method="GET"} @ 1609746000) → Invalid, because the @ modifier cannot appear inside the selector braces; it must appear after the selector or aggregation expression.
This invalid placement violates PromQL's syntax grammar for subquery and modifier ordering.
Reference:
Verified from Prometheus documentation - PromQL @ Modifier Syntax, Evaluation Modifiers, and PromQL Expression Grammar sections.

質問 # 29
Which PromQL statement returns the sum of all values of the metric node_memory_MemAvailable_bytes from 10 minutes ago?

- A. sum(node_memory_MemAvailable_bytes offset 10m)
- B. offset sum(node_memory_MemAvailable_bytes[10m])
- C. sum(node_memory_MemAvailable_bytes) offset 10m
- D. sum(node_memory_MemAvailable_bytes) setoff 10m

正解：A

解説：
In PromQL, the offset modifier allows you to query metrics as they were at a past time relative to the current evaluation. To retrieve the value of node_memory_MemAvailable_bytes as it was 10 minutes ago, you place the offset keyword inside the aggregation function's argument, not after it.
The correct query is:
sum(node_memory_MemAvailable_bytes offset 10m)
This computes the total available memory across all instances, based on data from exactly 10 minutes in the past.
Placing offset after the aggregation (as in option B) is syntactically invalid because modifiers apply to instant and range vector selectors, not to complete expressions.
Reference:
Verified from Prometheus documentation - PromQL Evaluation Modifiers: offset, Aggregation Operators, and Temporal Query Examples.

質問 # 30
With the following metrics over the last 5 minutes:
up{instance="localhost"} 1 1 1 1 1
up{instance="server1"} 1 0 0 0 0
What does the following query return:
min_over_time(up[5m])

- A. {instance="server1"} 0
- B. {instance="localhost"} 1 {instance="server1"} 0

正解：B

解説：
The min_over_time() function in PromQL returns the minimum sample value observed within the specified time range for each time series.
In the given data:
For up{instance="localhost"}, all samples are 1. The minimum value over 5 minutes is therefore 1.
For up{instance="server1"}, the sequence is 1 0 0 0 0. The minimum observed value is 0.
Thus, the query min_over_time(up[5m]) returns two series - one per instance:
{instance="localhost"} 1
{instance="server1"} 0
This query is commonly used to check uptime consistency. If the minimum value over the time window is 0, it indicates at least one scrape failure (target down).
Reference:
Verified from Prometheus documentation - PromQL Range Vector Functions, min_over_time() definition, and up Metric Semantics sections.

質問 # 31
Given the metric prometheus_tsdb_lowest_timestamp_seconds, how do you know in which month the lowest timestamp of your Prometheus TSDB belongs?

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

正解：A

解説：
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.

質問 # 32
.....

- ハイパスレートのPCAミシュレーション問題一回合格-認定するPCA資格トレーリング □ URL ☀ www.passtest.jp □☀□をコピーして開き、✔ PCA □✔□を検索して無料でダウンロードしてくださいPCAファンデーション
- PCA無料試験 □ PCA無料試験 □ PCA受験対策書 □ 「 PCA 」の試験問題は▷ www.goshiken.com ◁で無料配信中PCA練習問題
- 認定するLinux Foundation PCA | 真実的なPCAミシュレーション問題試験 | 試験の準備方法Prometheus Certified Associate Exam資格トレーリング □ □ www.xhs1991.com □は、[PCA]を無料でダウンロードするのに最適なサイトですPCA最新試験情報
- 最新のLinux Foundation PCAミシュレーション問題 - 合格スムーズPCA資格トレーリング | 権威のあるPCA関連合格問題 □ “www.goshiken.com”から簡単に□ PCA □を無料でダウンロードできますPCA無料サンプル
- PCA試験の準備方法 | 完璧なPCAミシュレーション問題試験 | 有効的なPrometheus Certified Associate Exam資格トレーリング □ ☀ www.mogiexam.com □☀□を開いて□ PCA □を検索し、試験資料を無料でダウンロードしてくださいPCA復習時間
- PCA認定試験 □ PCA試験関連赤本 □ PCA受験料過去問 □ 《 PCA 》を無料でダウンロード{ www.goshiken.com }で検索するだけPCA試験問題集
- 実際的-ユニークなPCAミシュレーション問題試験-試験の準備方法PCA資格トレーリング □ □ www.passtest.jp □に移動し、【 PCA 】を検索して、無料でダウンロード可能な試験資料を探しますPCAファンデーション
- presenciaschool.com, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, soulroutes.org.in, Disposable vapes

無料でクラウドストレージから最新のPass4Test PCA PDFダンプをダウンロードする：https://drive.google.com/open?id=1ndRYCEf9BrACB2iMAKCNOTsyo1vdIaj