New SPLK-4001 Braindumps Free - Latest SPLK-4001 Braindumps Free

What's more, part of that PDF4Test SPLK-4001 dumps now are free: https://drive.google.com/open?id=1O8W5re9A3sGqls84MydJHS0y9kR6Ceh8

As we all know, if you want to pass the SPLK-4001 exam, you need to have the right method of study, plenty of preparation time, and targeted test materials. However, most people do not have one or all of these. That is why I want to introduce our SPLK-4001 Original Questions to you. So why not try our Splunk original questions, which will help you maximize your pass rate? Even if you unfortunately fail to pass the exam, we will give you a full refund.

Upon successfully completing the Splunk SPLK-4001 Certification Exam, candidates will receive a Splunk O11y Cloud Certified Metrics User certification, which is recognized internationally as an industry-standard credential. Splunk O11y Cloud Certified Metrics User certification program will help validate the candidate's knowledge, skills, and experience in the area of cloud monitoring and analytics, which will ultimately enhance their career prospects and enable them to build a successful career in this exciting field.

>> New SPLK-4001 Braindumps Free <<

Advantages Of These Splunk SPLK-4001 Exam Questions Formats

If you buy online classes, you will need to sit in front of your computer on time at the required time; if you participate in offline counseling, you may need to take an hour or two of a bus to attend class. But if you buy SPLK-4001 test guide, things will become completely different. Unlike other learning materials on the market, Splunk O11y Cloud Certified Metrics User torrent prep has an APP version. You can download our app on your mobile phone. And then, you can learn anytime, anywhere. Whatever where you are, whatever what time it is, just an electronic device, you can do exercises. With Splunk O11y Cloud Certified Metrics User torrent prep, you no longer have to put down the important tasks at hand in order to get to class; with SPLK-4001 Exam Questions, you don't have to give up an appointment for study.

Splunk O11y Cloud Certified Metrics User Sample Questions (Q48-Q53):

NEW QUESTION # 48
An SRE creates a new detector to receive an alert when server latency is higher than 260 milliseconds. Latency below 260 milliseconds is healthy for their service. The SRE creates a New Detector with a Custom Metrics Alert Rule for latency and sets a Static Threshold alert condition at 260ms.
How can the number of alerts be reduced?

- A. Adjust the notification sensitivity. Duration set to 1 minute.
- B. Choose another signal.
- C. Adjust the threshold.
- D. Adjust the Trigger sensitivity. Duration set to 1 minute.

Answer: D

Explanation:
According to the Splunk O11y Cloud Certified Metrics User Track document1, trigger sensitivity is a setting that determines how long a signal must remain above or below a threshold before an alert is triggered. By default, trigger sensitivity is set to Immediate, which means that an alert is triggered as soon as the signal crosses the threshold. This can result in a lot of alerts, especially if the signal fluctuates frequently around the threshold value. To reduce the number of alerts, you can adjust the trigger sensitivity to a longer duration, such as 1 minute, 5 minutes, or 15 minutes. This means that an alert is only triggered if the signal stays above or below the threshold for the specified duration. This can help filter out noise and focus on more persistent issues.

NEW QUESTION # 49
What information is needed to create a detector?

- A. Alert Signal, Alert Criteria, Alert Settings, Alert Message, Alert Recipients
- B. Alert Signal, Alert Condition, Alert Settings, Alert Message, Alert Recipients
- C. Alert Status, Alert Criteria, Alert Settings, Alert Message, Alert Recipients
- D. Alert Status, Alert Condition, Alert Settings, Alert Meaning, Alert Recipients

Answer: B

Explanation:
According to the Splunk Observability Cloud documentation1, to create a detector, you need the following information:
Alert Signal: This is the metric or dimension that you want to monitor and alert on. You can select a signal from a chart or a dashboard, or enter a SignalFlow query to define the signal.
Alert Condition: This is the criteria that determines when an alert is triggered or cleared. You can choose from various built-in alert conditions, such as static threshold, dynamic threshold, outlier, missing data, and so on. You can also specify the severity level and the trigger sensitivity for each alert condition.
Alert Settings: This is the configuration that determines how the detector behaves and interacts with other detectors. You can set the detector name, description, resolution, run lag, max delay, and detector rules. You can also enable or disable the detector, and mute or unmute the alerts.
Alert Message: This is the text that appears in the alert notification and event feed. You can customize the alert message with variables, such as signal name, value, condition, severity, and so on. You can also use markdown formatting to enhance the message appearance.
Alert Recipients: This is the list of destinations where you want to send the alert notifications. You can choose from various channels, such as email, Slack, PagerDuty, webhook, and so on. You can also specify the notification frequency and suppression settings.

NEW QUESTION # 50
What information is needed to create a detector?

- A. Alert Signal, Alert Criteria, Alert Settings, Alert Message, Alert Recipients
- B. Alert Signal, Alert Condition, Alert Settings, Alert Message, Alert Recipients
- C. Alert Status, Alert Criteria, Alert Settings, Alert Message, Alert Recipients
- D. Alert Status, Alert Condition, Alert Settings, Alert Meaning, Alert Recipients

Answer: B

Explanation:
Explanation
According to the Splunk Observability Cloud documentation1, to create a detector, you need the following information:
Alert Signal: This is the metric or dimension that you want to monitor and alert on. You can select a signal from a chart or a dashboard, or enter a SignalFlow query to define the signal.
Alert Condition: This is the criteria that determines when an alert is triggered or cleared. You can choose from various built-in alert conditions, such as static threshold, dynamic threshold, outlier, missing data, and so on. You can also specify the severity level and the trigger sensitivity for each alert condition.
Alert Settings: This is the configuration that determines how the detector behaves and interacts with other detectors. You can set the detector name, description, resolution, run lag, max delay, and detector rules. You can also enable or disable the detector, and mute or unmute the alerts.
Alert Message: This is the text that appears in the alert notification and event feed. You can customize the alert message with variables, such as signal name, value, condition, severity, and so on. You can also use markdown formatting to enhance the message appearance.
Alert Recipients: This is the list of destinations where you want to send the alert notifications. You can choose from various channels,

such as email, Slack, PagerDuty, webhook, and so on. You can also specify the notification frequency and suppression settings.

NEW QUESTION # 51
To smooth a very spiky cpu.utilization metric, what is the correct analytic function to better see if the cpu.
utilization for servers is trending up over time?

- A. Mean (by host)
- B. Mean (Transformation)
- C. Median
- D. Rate/Sec

Answer: B

Explanation:
Explanation
The correct answer is D. Mean (Transformation).
According to the web search results, a mean transformation is an analytic function that returns the average value of a metric or a dimension over a specified time interval1. A mean transformation can be used to smooth a very spiky metric, such as cpu.utilization, by reducing the impact of outliers and noise. A mean transformation can also help to see if the metric is trending up or down over time, by showing the general direction of the average value. For example, to smooth the cpu.utilization metric and see if it is trending up over time, you can use the following SignalFlow code:
mean(1h, counters("cpu.utilization"))
This will return the average value of the cpu.utilization counter metric for each metric time series (MTS) over the last hour. You can then use a chart to visualize the results and compare the mean values across different MTS.
Option A is incorrect because rate/sec is not an analytic function, but rather a rollup function that returns the rate of change of data points in the MTS reporting interval1. Rate/sec can be used to convert cumulative counter metrics into counter metrics, but it does not smooth or trend a metric. Option B is incorrect because median is not an analytic function, but rather an aggregation function that returns the middle value of a metric or a dimension over the entire time range1. Median can be used to find the typical value of a metric, but it does not smooth or trend a metric. Option C is incorrect because mean (by host) is not an analytic function, but rather an aggregation function that returns the average value of a metric or a dimension across all MTS with the same host dimension1. Mean (by host) can be used to compare the performance of different hosts, but it does not smooth or trend a metric.
Mean (Transformation) is an analytic function that allows you to smooth a very spiky metric by applying a moving average over a specified time window. This can help you see the general trend of the metric over time, without being distracted by the short-term fluctuations1 To use Mean (Transformation) on a cpu.utilization metric, you need to select the metric from the Metric Finder, then click on Add Analytics and choose Mean (Transformation) from the list of functions. You can then specify the time window for the moving average, such as 5 minutes, 15 minutes, or 1 hour. You can also group the metric by host or any other dimension to compare the smoothed values across different servers2 To learn more about how to use Mean (Transformation) and other analytic functions in Splunk Observability Cloud, you can refer to this documentation2.
1: https://docs.splunk.com/Observability/gdi/metrics/analytics.html#Mean-Transformation 2:
https://docs.splunk.com/Observability/gdi/metrics/analytics.html

NEW QUESTION # 52
A customer has a very dynamic infrastructure. During every deployment, all existing instances are destroyed, and new ones are created Given this deployment model, how should a detector be created that will not send false notifications of instances being down?

- A. Create the detector. Select Alert settings, then select Auto-Clear Alerts and enter an appropriate time period.
- B. Check the Dynamic checkbox when creating the detector.
- C. Create the detector. Select Alert settings, then select Ephemeral Infrastructure and enter the expected lifetime of an instance.
- D. Check the Ephemeral checkbox when creating the detector.

Answer: C

Explanation:
Explanation
According to the web search results, ephemeral infrastructure is a term that describes instances that are auto-scaled up or down, or are brought up with new code versions and discarded or recycled when the next code version is deployed1. Splunk Observability Cloud has a feature that allows you to create detectors for ephemeral infrastructure without sending false notifications of instances

being down2. To use this feature, you need to do the following steps:
Create the detector as usual, by selecting the metric or dimension that you want to monitor and alert on, and choosing the alert condition and severity level.
Select Alert settings, then select Ephemeral Infrastructure. This will enable a special mode for the detector that will automatically clear alerts for instances that are expected to be terminated.
Enter the expected lifetime of an instance in minutes. This is the maximum amount of time that an instance is expected to live before being replaced by a new one. For example, if your instances are replaced every hour, you can enter 60 minutes as the expected lifetime.
Save the detector and activate it.
With this feature, the detector will only trigger alerts when an instance stops reporting a metric unexpectedly, based on its expected lifetime. If an instance stops reporting a metric within its expected lifetime, the detector will assume that it was terminated on purpose and will not trigger an alert. Therefore, option B is correct.

NEW QUESTION # 53
.....

id=1O8W5re9A3sGqls84MydJHS0y9kR6Ceh8