SPLK-4001 Q&As

Splunk O11y Cloud Certified Metrics User

Pass Splunk SPLK-4001 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/splk-4001.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Splunk Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

What is one reason a user of Splunk Observability Cloud would want to subscribe to an alert?

A. To determine the root cause of the Issue triggering the detector.

B. To perform transformations on the data used by the detector.

C. To receive an email notification when a detector is triggered.

D. To be able to modify the alert parameters.

Correct Answer: C

One reason a user of Splunk Observability Cloud would want to subscribe to an alert is C. To receive an email notification when a detector is triggered. A detector is a component of Splunk Observability Cloud that monitors metrics or events and triggers alerts when certain conditions are met. A user can create and configure detectors to suit their monitoring needs and goals A subscription is a way for a user to receive notifications when a detector triggers an alert. A user can subscribe to a detector by entering their email address in the Subscription tab of the detector page. A user can also unsubscribe from a detector at any time When a user subscribes to an alert, they will receive an email notification that contains information about the alert, such as the detector name, the alert status, the alert severity, the alert time, and the alert message. The email notification also includes links to view the detector, acknowledge the alert, or unsubscribe from the detector To learn more about how to use detectors and subscriptions in Splunk Observability Cloud, you can refer to these documentations. https://docs.splunk.com/Observability/alerts-detectors-notifications/detectors.html https://docs.splunk.com/Observability/alerts-detectors-notifications/subscribe-to-detectors.html

QUESTION 2

A customer deals with a holiday rush of traffic during November each year, but does not want to be flooded with alerts when this happens. The increase in traffic is expected and consistent each year. Which detector condition should be used when creating a detector for this data?

A. Outlier Detection

B. Static Threshold

C. Calendar Window

D. Historical Anomaly

Correct Answer: D

historical anomaly is a detector condition that allows you to trigger an alert when a signal deviates from its historical pattern. Historical anomaly uses machine learning to learn the normal behavior of a signal based on its past data, and then compares the current value of the signal with the expected value based on the learned pattern. You can use historical anomaly to detect unusual changes in a signal that are not explained by seasonality, trends, or cycles. Historical anomaly is suitable for creating a detector for the customer\\'s data, because it can account for the expected and consistent increase in traffic during November each year. Historical anomaly can learn that the traffic pattern has a seasonal component that peaks in November, and then adjust the expected value of the traffic accordingly. This way, historical anomaly can avoid triggering alerts when the traffic increases in November, as this is not an anomaly, but rather a normal variation. However, historical anomaly can still trigger alerts when the traffic deviates from the historical pattern in other ways, such as if it drops significantly or spikes unexpectedly.

QUESTION 3

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. Create the detector. Select Alert settings, then select Ephemeral Infrastructure and enter the expected lifetime of an instance.

C. Check the Dynamic checkbox when creating the detector.

D. Check the Ephemeral checkbox when creating the detector.

Correct Answer: B

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 down. 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.

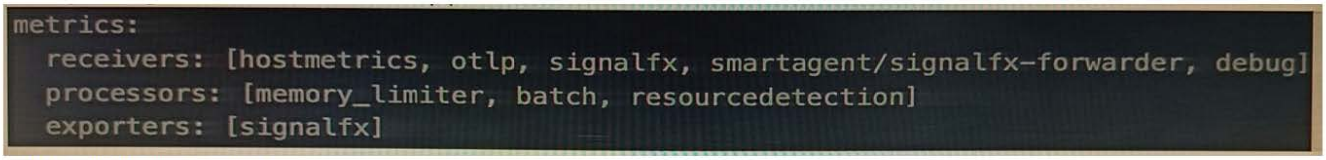
With this feature, the detector will only trigger alerts when an instance stops reporting a metric unexpectedly, based on its expected lifetime. If an instance stops reporting a metric within its expected lifetime, the detector will assume that it was

terminated on purpose and will not trigger an alert. Therefore, option B is correct.

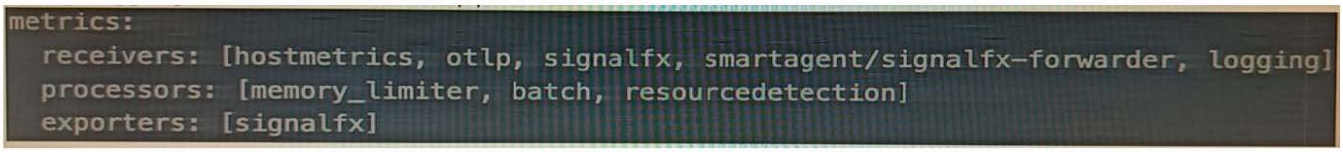
QUESTION 4

A customer is experiencing issues getting metrics from a new receiver they have configured in the OpenTelemetry Collector. How would the customer go about troubleshooting further with the logging exporter?

A. Adding debug into the metrics receiver pipeline:



B. Adding logging into the metrics receiver pipeline:



C. Adding logging into the metrics exporter pipeline:

D. Adding debug into the metrics exporter pipeline:

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

The correct answer is B. Adding logging into the metrics receiver pipeline. The logging exporter is a component that allows the OpenTelemetry Collector to send traces, metrics, and logs directly to the console. It can be used to diagnose and troubleshoot issues with telemetry received and processed by the Collector, or to obtain samples for other purposes To activate the logging exporter, you need to add it to the pipeline that you want to diagnose. In this case, since you are experiencing issues with a new receiver for metrics, you need to add the logging exporter to the metrics receiver pipeline. This will create a new plot that shows the metrics received by the Collector and any errors or warnings that might occur The image that you have sent with your question shows how to add the logging exporter to the metrics receiver pipeline. You can see that the exporters section of the metrics pipeline includes logging as one of the options. This means that the metrics received by any of the receivers listed in the receivers section will be sent to the logging exporter as well as to any other exporters listed To learn more about how to use the logging exporter in Splunk Observability Cloud, you can refer to this documentation.
https://docs.splunk.com/Observability/gdi/opentelemetry/components/logging- exporter.html
https://docs.splunk.com/Observability/gdi/opentelemetry/exposed- endpoints.html

QUESTION 5

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. Rate/Sec

B. Median

C. Mean (by host)

D. Mean (Transformation)

Correct Answer: D

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 interval. 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 fluctuations 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 documentation. https://docs.splunk.com/Observability/gdi/metrics/analytics.html#Mean-Transformation https://docs.splunk.com/Observability/gdi/metrics/analytics.html