www.atmicnetworks.com

ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

www.atmicnetworks.com

Question: 1

If a search contains a subsearch, what is the order of execution?

A. The order of execution depends on whether either search uses a stats command.
B. The inner search executes first.
C. The outer search executes first.
D. The two searches are executed in parallel.

Answer: B

Explanation:

In a Splunk search containing a subsearch, the inner subsearch executes first. The result of the subsearch is then passed to the outer search, which often depends on the results of the inner subsearch to complete its execution.

Reference:
Splunk Documentation on Subsearches:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutsubsearches
Splunk Documentation on Search Syntax:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Usefieldsinsearches

Question: 2

How can the erex and rex commands be used in conjunction to extract fields?

A. The regex generated by the erex command can be edited and used with the rex command in a subsequent search.
B. The regex generated by the rex command can be edited and used with the erex command in a subsequent search.
C. The regex generated by the erex command can be edited and used with the erex command in a subsequent search.
D. The erex and rex commands cannot be used in conjunction under any circumstances.

Answer: A

Explanation:

The erex command in Splunk generates regular expressions based on example data. These generated regular expressions can then be edited and utilized with the rex command in subsequent searches.

www.atmicnetworks.com

Question: 3

What command is used to compute and write summary statistics to a new field in the event results?

A. tstats
B. stats
C. eventstats
D. transaction

Answer: C

Explanation:

The eventstats command in Splunk is used to compute and add summary statistics to all events in the search results, similar to stats, but without grouping the results into a single event.

Question: 4

Which commands can run on both search heads and indexers?

A. Transforming commands
B. Centralized streaming commands
C. Dataset processing commands
D. Distributable streaming commands

Answer: D

Explanation:

In Splunk's processing model, commands are categorized based on how and where they execute within the search pipeline. Understanding these categories is crucial for optimizing search performance.

Distributable Streaming Commands:

Definition: These commands operate on each event individually and do not depend on the context of other events. Because of this independence, they can be executed on indexers, allowing the processing load to be distributed across multiple nodes.

Execution: When a search is run, distributable streaming commands can process events as they are retrieved from the indexers, reducing the amount of data sent to the search head and improving efficiency.

Examples: eval, rex, fields, rename

Other Command Types:

Dataset Processing Commands: These commands work on entire datasets and often require all events to be available before processing can begin. They typically run on the search head.

Centralized Streaming Commands: These commands also operate on each event but require a centralized view of the data, meaning they usually run on the search head after data has been gathered from the indexers.

Transforming Commands: These commands, such as stats or chart, transform event data into statistical tables and generally run on the search head.

By leveraging distributable streaming commands, Splunk can efficiently process data closer to its source, optimizing resource

utilization and search performance.

Reference:

Splunk Documentation: Types of commands

Question: 5

What is returned when Splunk finds fewer than the minimum matches for each lookup value?

A. The default value NULL until the minimum match threshold is reached.
B. The default match value until the minimum match threshold is reached.
C. The first match unless the time_field attribute is specified.
D. Only the first match.

Answer: A

Explanation:

When Splunk's lookup feature finds fewer than the minimum matches for each lookup value, it returns the default value NULL for unmatched entries until the minimum match threshold is reached.

Question: 6

When would a distributable streaming command be executed on an indexer?

A. If any of the preceding search commands are executed on the search head.
B. If all preceding search commands are executed on the indexer, and a streamstats command is used.
C. If all preceding search commands are executed on the indexer.
D. If some of the preceding search commands are executed on the indexer, and a timerchart command is used.

Answer: C

Explanation:

A distributable streaming command would be executed on an indexer if all preceding search commands are executed on the indexer, enhancing search efficiency by processing data where it resides.

A distributable streaming command is executed on an indexer if all preceding search commands are executed on the indexer . This ensures that the entire pipeline up to that point can be processed locally on the indexer without requiring intermediate results to be sent to the search head.

Here's why this works:

Distributable Streaming Commands : These commands process data in a streaming manner and can run on indexers if all prior commands in the pipeline are also distributable. Examples include eval, fields, and rex.

Execution Location : For a command to execute on an indexer, all preceding commands must also be distributable. If any non-distributable command (e.g., stats, transaction) is encountered, processing shifts to the search head.

www.atmicnetworks.com

Question: 7

Why is the transaction command slow in large Splunk deployments?

A. It forces the search to run in fast mode.
B. The transaction runs on each indexer in parallel.
C. It forces all event data to be returned to the search head.
D. The transaction runs a hidden eval to format fields.

Answer: C

Explanation:

The transaction command can be slow in large deployments because it requires all event data relevant to the transaction to be returned to the search head, which can be resource-intensive.

Question: 8

What are the four types of event actions?

A. stats, target, set, and unset

B. stats, target, change, and clear
C. eval, link, change, and clear
D. eval, link, set, and unset

Answer: C

Explanation:

The four types of event actions in Splunk are:
eval : Allows you to create or modify fields using expressions.
link : Creates clickable links that can redirect users to external resources or other Splunk views. change : Triggers actions when a field's value changes, such as highlighting or formatting changes. clear : Clears or resets specific fields or settings in the context of an event action.
Here's why this works:
These event actions are commonly used in Splunk dashboards and visualizations to enhance interactivity and provide dynamic behavior based on user input or data changes.
Other options explained:
Option A : Incorrect because stats and target are not valid event actions.
Option B : Incorrect because set and unset are not valid event actions.
Option D : Incorrect because stats and target are not valid event actions.
Reference:
Splunk Documentation on Event Actions:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/EventActions
Splunk Documentation on Dashboard Interactivity:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/PanelreferenceforSimplifiedXML

www.atmicnetworks.com

Question: 9

When using the bin command, which argument sets the bin size?

A. maxDataSizeMB
B. max
C. volume
D. span

Answer: D

Explanation:

In Splunk, the span argument is used to set the size of each bin when using the bin command, determining the granularity of segmented data over a time range or numerical field.

Question: 10

How is a cascading input used?

A. As part of a dashboard, but not in a form.
B. Without notation in the underlying XML.
C. As a way to filter other input selections.
D. As a default way to delete a user role.

Answer: C

Explanation:

A cascading input is used to filter other input selections in a dashboard or form, allowing for a dynamic user interface where one input influences the options available in another input.

Cascading Inputs:

Definition: Cascading inputs are interconnected input controls in a dashboard where the selection in one input filters the options available in another. This creates a hierarchical selection process, enhancing user experience by presenting relevant choices based on prior selections.

Implementation:

Define Input Controls:

Create multiple input controls (e.g., dropdowns) in the dashboard.

Set Token Dependencies:

Configure each input to set a token upon selection.

Subsequent inputs use these tokens to filter their available options.

Example:

Consider a dashboard analyzing sales data:

Input 1: Country Selection

Dropdown listing countries.

Sets a token $country$ upon selection.

Input 2: City Selection

Dropdown listing cities.

Uses the $country$ token to display only cities within the selected country.

XML Configuration:

```
<input type="dropdown" token="country">
<label>Select Country</label>
<choice value="USA">USA</choice>
<choice value="Canada">Canada</choice>
</input>

<input type="dropdown" token="city">
<label>Select City</label>
<search>
  <query>index=sales_data country=$country$ | stats count by city</query>
</search>
</input>
```

In this setup:

Selecting a country sets the $country$ token.

The city dropdown's search uses this token to display cities relevant to the selected country. Benefits:

Improved User Experience: Users are guided through a logical selection process, reducing the chance of invalid or irrelevant selections.

Data Relevance: Ensures that dashboard panels and visualizations reflect data pertinent to the user's selections.

Other Options Analysis:

B . As part of a dashboard, but not in a form:

Cascading inputs are typically used within forms in dashboards to collect user input. This option is incorrect as it suggests a limitation that doesn't exist.

C . Without token notation in the underlying XML:

Cascading inputs rely on tokens to pass values between inputs. Therefore, token notation is essential in the XML configuration.

D . As a default way to delete a user role:

This is unrelated to the concept of cascading inputs.

Conclusion:

Cascading inputs are used in dashboards to create a dependent relationship between input controls, allowing selections in one input to filter the options available in another, thereby enhancing data relevance and user experience.

Reference:

Splunk Documentation: Set up cascading or dependent inputs

Question: 11

Which of the following is accurate regarding predefined drilldown tokens?

A. They capture data from a form input.
B. They vary by visualization type.
C. There are eight categories of predefined drilldown tokens.
D. They are defined by a panel's base search.

Answer: B

Explanation:

Predefined drilldown tokens in Splunk vary by visualization type. These tokens are placeholders that capture dynamic values based on user interactions with dashboard elements, such as clicking on a chart segment or table row. Different visualization types may have different drilldown tokens.

Question: 12

Which of the following statements is accurate regarding the append command?

A. It is used with a subsearch and only accesses real-time searches.
B. It is used with a subsearch and only accesses historical data.
C. It cannot be used with a subsearch and only accesses historical data.
D. It cannot be used with a subsearch and only accesses real-time searches.

Answer: B

Explanation:

The append command in Splunk is used with a subsearch to add additional data to the end of the primary search results and can access historical data, making it useful for combining datasets from different time ranges or sources.

Question: 13

What happens to panels with post-processing searches when their base search is refreshed?

A. The panels are deleted.
B. The panels are only refreshed if they have also been configured.
C. The panels are refreshed automatically.
D. Nothing happens to the panels.

Answer: C

Explanation:

When the base search of a dashboard panel with post-processing searches is refreshed, the panels with these post-processing searches are refreshed automatically to reflect the updated data.

Question: 14

Which of the following are potential string results returned by the typeof function?

A. True, False, Unknown
B. Number, String, Bool

C. Number, String, Null

D. Field, Value, Lookup

Answer: B

Explanation:

The typeof function in Splunk is used to determine the data type of a field or value. It returns one of the following string results:

Number : Indicates that the value is numeric.

String : Indicates that the value is a text string.

Bool : Indicates that the value is a Boolean (true/false).

Here's why this works:

Purpose of typeof : The typeof function is commonly used in conjunction with the eval command to inspect the data type of fields or expressions. This is particularly useful when debugging or ensuring that fields are being processed as expected.

Return Values : The function categorizes values into one of the three primary data types supported by Splunk: Number, String, or Bool.

Example:

```
| makeresults
| eval example_field = "123"
| eval type = typeof(example_field)
```

This will produce:

_time	example_field	type
<current_timestamp>	123	String

Other options explained:

Option A : Incorrect because True, False, and Unknown are not valid return values of the typeof function. These might be confused with Boolean logic but are not related to data type identification. Option C : Incorrect because Null is not a valid return value of typeof. Instead, Null represents the absence of a value, not a data type.

Option D : Incorrect because Field, Value, and Lookup are unrelated to the typeof function. These terms describe components of Splunk searches, not data types.

Reference:

Splunk Documentation on typeof:

https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/CommonEvalFunctions

Splunk Documentation on Data Types:

https://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutfields

Question: 15

Which search generates a field with a value of "hello"

A. | makeresults field="hello"
B. | makeresults | fields="hello"
C. | makeresults | eval field="hello"
D. | makeresults | eval field=make{"hello"}

Answer: C

Explanation:
The correct search to generate a field with a value of "hello" is: Copy 1
| makeresults | eval field="hello"
Here's why this works:
makeresults : This command creates a single event with no fields.
eval : The eval command is used to create or modify fields. In this case, it creates a new field named field and assigns it the value "hello".
Example:

| makeresults | eval field="hello"

This will produce a result like:

time	field
<current_timestamp>	hello

Reference:
Splunk Documentation on makeresults:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Makeresults
Splunk Documentation on eval:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Eval

Question: 16

What is one way to troubleshoot dashboards?

A. Create an HTML panel using tokens to verify that they are being set.
B. Delete the dashboard and start over.
C. Go to the Troubleshooting dashboard of the Searching and Reporting app.
D. Run the previous_searches command to troubleshoot your SPL queries.

Answer: A

Explanation:
Comprehensive and Detailed Step by Step
One effective way to troubleshoot dashboards in Splunk is to create an HTML panel using tokens to verify that tokens are being set correctly. This allows you to debug token values and ensure that dynamic behavior (e.g., drilldowns, filters) is functioning as expected.
Here's why this works:

HTML Panels for Debugging : By embedding an HTML panel in your dashboard, you can display the current values of tokens dynamically. For example:

<html>
Token value: $token_name$
</html>

This helps you confirm whether tokens are being updated correctly based on user interactions or other inputs.

Token Verification : Tokens are essential for dynamic dashboards, and verifying their values is a critical step in troubleshooting issues like broken drilldowns or incorrect filters.

Other options explained:

Option B : Incorrect because deleting and recreating a dashboard is not a practical or efficient troubleshooting method.

Option C : Incorrect because there is no specific "Troubleshooting dashboard" in the Searching and Reporting app.

Option D : Incorrect because the previous_searches command is unrelated to dashboard troubleshooting; it lists recently executed searches.

Reference:

Splunk Documentation on Dashboard Troubleshooting:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/Troubleshootdashboards

Splunk Documentation on Tokens:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/UseTokenstoBuildDynamicInputs

Question: 17

How is a multivalue field treated from product="a, b, c, d"

A .. | makemv delim{product, ","}
B .. | eval mvexpand{makemv{product, ","}}
C .. | mvexpand product
D . | makemv delim="," product

Answer: D

Explanation:

The makemv command with delim="," is used to split a multivalue field like product="a, b, c, d" into separate values, making it easier to manipulate each value individually.

Question: 18

How can the inspect button be disabled on a dashboard panel?

A. Set inspect.link.disabled to 1

B. Set link.inspect.visible to 0
C. Set link.inspectSearch.visible to 0
D. Set link.search.disabled to 1

Answer: B

Explanation:

To disable the inspect button on a dashboard panel, set the link.inspect.visible attribute to 0. This hides the button, preventing users from accessing the search inspector for that panel.

To disable the Inspect button on a dashboard panel in Splunk, you need to set the attribute link.inspect.visible to 0. This hides the Inspect button for that specific panel.

Here's why this works:

Purpose of link.inspect.visible : The link.inspect.visible attribute controls the visibility of the Inspect button in a dashboard panel. Setting it to 0 disables the button, while setting it to 1 (default) keeps it visible.

Customization : This is useful when you want to restrict users from inspecting the underlying search queries or data for a specific panel.

Question: 19

Which of the following is valid syntax for the split function?

A .. | eval split phoneNumber by "" as areaCodes.
B .. | eval areaCodes = split(phoneNumber, "")
C .. | eval phoneNumber split("-", 3, areaCodes)
D . | eval split(phone-Number, "_", areaCodes)

Answer: B

Explanation:

The valid syntax for using the split function in Splunk is ..... | eval areaCodes = split(phoneNumber, "_"). This function splits the string based on the specified delimiter, creating an array of substrings.

Question: 20

Which field is required for an event annotation?

A. annotation_category
B. _time
C. eventtype
D. annotation_label

Answer: B

Explanation:

The _time field is required for event annotations in Splunk. This field specifies the time point or range where the annotation should be applied, helping correlate annotations with the correct temporal data.

Question: 21

How is regex passed to the makemv command?

A. makemv must be preceded by the erex command.
B. It is specified by the delim argument.
C. It is specified by the tokenizer argument.
D. makemv must be preceded by the rex command.

Answer: B

Explanation:

The regex is passed to the makemv command in Splunk using the delim argument. This argument specifies the delimiter used to split a single string field into multiple values, effectively creating a multivalue field.

Question: 22

Which of the following best describes the process for tokenizing event data?

A. The event data is broken up by values in the punch field.
B. The event data is broken up by major breakers and then broken up further by minor breakers.
C. The event data is broken up by a series of user-defined regex patterns.
D. The event data has all punctuation stripped out and is then space-delimited.

Answer: B

Explanation:

The process for tokenizing event data in Splunk involves breaking the event data up by major breakers (which typically identify the boundaries of events) and further breaking it up by minor breakers (which segment the event data into fields). This hierarchical approach allows Splunk to efficiently parse and structure the data.

Question: 23

What qualifies a report for acceleration?

A. Fewer than 100k events in search results, with transforming commands used in the search string.
B. More than 100k events in search results, with only a search command in the search string.
C. More than 100k events in the search results, with a search and transforming command used in the search string.
D. Fewer than 100k events in search results, with only a search and transaction command used in the search string.

Answer: A

Explanation:

A report qualifies for acceleration in Splunk if it involves fewer than 100,000 events in the search results and uses transforming commands. Transforming commands aggregate data, which helps reduce the dataset's size and complexity, making the report suitable for acceleration.

Question: 24

Assuming a standard time zone across the environment, what syntax will always return events from between 2:00 AM and 5:00 AM?

A. datehour>-2 AND date_hour<5
B. earliest=-2h@h AND latest=-5h@h
C. time_hour>-2 AND time_hour>-5
D. earliest=2h@ AND latest=5h3h

Answer: B

Explanation:

The correct syntax to return events from between 2:00 AM and 5:00 AM is earliest=-2h@h AND latest=-5h@h. This uses relative time modifiers to specify a range starting at 2 AM and ending at 5 AM.

Question: 25

What capability does a power user need to create a Log Event alert action?

A. edit_search_server
B. edit_udp
C. edit_tcp
D. edit_alerts

Answer: D

Explanation:

To create a Log Event alert action in Splunk, a power user needs the edit_alerts capability. This capability allows the user to

configure and manage alert actions within Splunk.

Question: 26

Where can wildcards be used in the tstats command?

A. No wildcards can be used with tstats.
B. In the where clause.
C. In the from clause.
D. In the by clause.

Answer: C

Explanation:

Wildcards can be used in the from clause of the tstats command in Splunk. This allows users to query across multiple datasets or data models that share a common naming pattern.

Question: 27

What is the result of the xyseries command?

A. To transform single series output into a multi-series output.
B. To transform a stats-like output into chart-like output.
C. To transform a multi-series output into single series output.
D. To transform a chart-like output into a stats-like output.

Answer: B

Explanation:

The xyseries command in Splunk transforms a stats-like output into a chart-like output, making it easier to visualize complex relationships between multiple data points.

Question: 28

What XML element is used to pass multiple fields into another dashboard using a dynamic drilldown?

A. <drilldown field="sources_Field_name">
B. <condition field="sources_Field_name">
C. <pass_token field="sources_field_name">
D. <link field="sources_field_name">

Answer: D

Explanation:

In Splunk Simple XML for dashboards, the <link> element is used within a <drilldown> configuration to pass multiple fields to another dashboard using dynamic drilldown.

Question: 29

Which function of the stats command creates a multivalue entry?

A. mvcombine
B. eval
C. makemv
D. list

Answer: D

Explanation:

The list function of the stats command creates a multivalue entry, combining multiple occurrences of a field into a single multivalue field.

The list function of the stats command creates a multivalue entry by aggregating values from multiple events into a single field. This is particularly useful when you want to group data and collect all matching values into a list.

Here's why this works:

Purpose of list : The list function collects all values of a specified field for each group and stores them as a multivalue field. For example, if you group by user_id, the list function will create a multivalue field containing all corresponding product values for that user.

Multivalue Fields : Multivalue fields allow you to handle multiple values within a single field, which can be expanded or manipulated using commands like mvexpand or foreach.

Reference:

Splunk Documentation on stats:

https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/stats

Splunk Documentation on Multivalue Fields:

https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/MultivalueEvalFunctions

Question: 30

What is the recommended way to create a field extraction that is both persistent and precise?

A. Use the rex command.
B. Use the Field Extractor and manually edit the generated regular expression.
C. Use the Field Extractor and let it automatically generate a regular expression.
D. Use the erex command.

Answer: B

Explanation:

The recommended way to create a field extraction that is both persistent and precise is to use the Field Extractor and manually edit the generated regular expression. This ensures accuracy and allows for customization beyond the automatically generated regex.

Question: 31

What is the value of base lispy in the Search Job Inspector for the search index=sales clientip=170.192.178.10?

A. [index::sales AND 192 AND 10 AND 178 AND 170]
B. [index::sales AND 469 10 702 390]
C. [192 AND 10 AND 178 AND 170 index::sales]
D. [AND 10 170 178 192 index::sales]

Answer: A

Explanation:

The base lispy expression represents how Splunk parses and simplifies a search command. In this case, the lispy format shows how Splunk is breaking down the search terms to effectively perform the search.

Question: 32

What is an example of the simple XML syntax for a base search and its post-process search?

A. <search id="myBaseSearch">, <search base="myBaseSearch">
B. <search globalsearch="myBaseSearch">, <search globalsearch>

C. <panel id="myBaseSearch">, <panel base="myBaseSearch">
D. <search id="myGlobalSearch">, <search base="myBaseSearch">

Answer: A

Explanation:

In Splunk, a base search is defined using <search id="myBaseSearch"> and is referenced by postprocess searches using the base attribute, as seen in the syntax <search base="myBaseSearch">.

www.atmicnetworks.com

Question: 33

What arguments are required when using the spath command?

A. input, output, index
B. input, output path
C. No arguments are required.
D. field, host, source

Answer: C

Explanation:

The spath command in Splunk is used to extract fields from structured data formats like JSON or XML. No arguments are required for basic usage, as spath automatically parses the _raw field by default. Here's why this works:
Default Behavior : By default, spath extracts fields from the _raw field of events without requiring any arguments. It intelligently parses JSON or XML data and creates new fields based on the structure.
Optional Arguments : While spath does not require arguments, you can optionally specify: input: To specify a field other than _raw to parse.
output: To rename the extracted fields.
path: To extract specific subfields within the structured data.
Example:

```
| makeresults
| eval _raw="{\"name\":\"Alice\",\"age\":30}"
| spath
```

Reference:
Splunk Documentation on spath:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/spath
Splunk Documentation on Parsing Structured Data:
https://docs.splunk.com/Documentation/Splunk/latest/Data/Extractfieldsfromstructureddata

Question: 34

When possible, what is the best choice for summarizing data to improve search performance?

A. Use the fieldsummary command.
B. Data model acceleration
C. Report acceleration
D. Summary indexing

Answer: B

Explanation:

When possible, data model acceleration is the best choice for summarizing data to improve search performance. It is specifically designed for optimizing searches over large datasets and complex data models.

www.atmicnetworks.com

Here's why this works:

Data Model Acceleration : Data model acceleration precomputes summaries of data models, enabling faster pivot operations and searches. It is ideal for use cases involving large datasets and complex relationships between fields.

Performance Benefits : By accelerating data models, Splunk reduces the computational overhead of searching raw data, making it significantly faster to generate reports and visualizations.

Other options explained:

Option A : Incorrect because summary indexing is better suited for aggregating data over long time ranges but is less flexible than data model acceleration.

Option C : Incorrect because report acceleration is limited to specific reports and does not provide the same level of flexibility as data model acceleration.

Option D : Incorrect because the fieldsummary command provides statistical summaries of fields but does not improve search performance for large datasets.

Example: To enable data model acceleration:

Navigate to Settings > Data Models in Splunk.

Select the data model you want to accelerate.

Configure acceleration settings, such as the summary range and update frequency.

Reference:

Splunk Documentation on Data Model Acceleration:

https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Acceleratedatamodels

Splunk Documentation on Summary Indexing:

https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Usesummaryindexing

Question: 35

Which syntax is used when referencing multiple CSS files in a view?

A. <dashboard stylesheet="custom.css | userapps.css">
B. <dashboard style="custom.css, userapps.css">
C. <dashboard stylesheet=custom.css stylesheet=userapps.css>
D. <dashboard stylesheet="custom.css, userapps.css">

Answer: D

Explanation:

To reference multiple CSS files in a Splunk dashboard, you use the stylesheet attribute with a comma- separated list of file names enclosed in quotes. The correct syntax is: xml

Copy

```
1
<dashboard stylesheet="custom.css, userapps.css">
```

Here's why this works:

stylesheet Attribute : The stylesheet attribute allows you to specify one or more CSS files to style your dashboard.

Comma-Separated List : Multiple CSS files are referenced by listing their names separated by commas within a single

stylesheet attribute.

Quotes : The entire list of CSS files must be enclosed in quotes to ensure proper parsing.

Other options explained:

Option A : Incorrect because the pipe (|) character is not valid for separating CSS file names.

Option B : Incorrect because the style attribute is not used for referencing CSS files in Splunk dashboards.

Option C : Incorrect because the stylesheet attribute cannot be repeated; instead, all CSS files must be listed in a single stylesheet attribute.

Example:

```
<dashboard stylesheet="custom.css, userapps.css">
<label>Styled Dashboard</label>
<row>
  <panel>
   <title>Panel Title</title>
   <table>
    <search>
      <query>index=_internal | head 10</query>
    </search>
   </table>
  </panel>
</row>
</dashboard>
```

Reference:

Splunk Documentation on Dashboard Styling:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/CustomizeDashboardCSS

Splunk Documentation on XML Structure:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/PanelreferenceforSimplifiedXML

Question: 36

How can a lookup be referenced in an alert?

A. Use the lookup dropdown in the alert configuration window.
B. Follow a lookup with an alert command in the search bar.
C. Run a search that uses a lookup and save as an alert.
D. Upload a lookup file directly to the alert.

Answer: C

Explanation:

In Splunk, a lookup can be referenced in an alert by running a search that incorporates the lookup and saving that search as an alert. This allows the alert to use the lookup data as part of its logic.

www.atmicnetworks.com

Question: 37

Where does the output of an append command appear in the search results?

A. Added as a column to the right of the search results.
B. Added as a column to the left of the search results.
C. Added to the beginning of the search results.
D. Added to the end of the search results.

Answer: D

Explanation:

The output of the append command is added to the end of the current search results. This is useful for concatenating additional data from a subsearch.

Question: 38

Which stats function is used to return a sorted list of unique field values?

A. values

B. sum
C. count
D. list

Answer: A

Explanation:

The values function in the stats command returns a sorted list of unique values from a specified field, making it helpful for summarizing and analyzing data.

Question: 39

How can form inputs impact dashboard panels using inline searches?

A. Panels powered by an inline search require a minimum of one form input.
B. Form inputs cannot impact panels using inline searches.
C. Adding a form input to a dashboard converts all panels to prebuilt panels.
D. A token in a search can be replaced by a form input value.

Answer: D

Explanation:

www.atmicnetworks.com

Form inputs in Splunk dashboards allow users to dynamically interact with the data displayed in panels. When a panel uses an inline search, you can use tokens to replace parts of the search query with values provided by form inputs.

Here's how this works:

Tokens : Tokens are placeholders in a search query that can be dynamically replaced with user- provided values from form inputs (e.g., dropdowns, text boxes).

Dynamic Searches : When a user interacts with a form input, the token value is updated, and the search query is re-executed with the new value.

Inline Searches : Inline searches are defined directly within the panel's XML or configuration, and they can include tokens to make them dynamic.

For example:

```
<input type="dropdown" token="selected_product">
<label>Select Product</label>
<choice value="productA">Product A</choice>
<choice value="productB">Product B</choice>
</input>

<panel>
<title>Sales for $selected_product$</title>
<table>
  <search>
    <query>index=sales product="$selected_product$" | stats count by region</query>
  </search>
</table>
</panel>
```

Other options explained:

Option A : Incorrect because form inputs can indeed impact panels using inline searches.

Option B : Incorrect because adding a form input does not automatically convert panels to prebuilt panels.

Option D : Incorrect because panels using inline searches do not require a minimum of one form input.

Reference:

Splunk Documentation on Tokens:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/UseTokenstoBuildDynamicInputs

Splunk Documentation on Inline Searches:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/PanelreferenceforSimplifiedXML

Question: 40

Which of the following has a schema or structure embedded in the data itself?

A. Dark data
B. Unstructured data
C. Embedded data

D. Self-describing data

Answer: D

Explanation:

Self-describing data includes information about its structure within the data itself. Examples include formats like JSON and XML, where the data schema is embedded and can be easily interpreted without external references.

Question: 41

Which of the following functions' primary purpose is to convert epoch time to a string format?

A. tostring
B. strptime
C. tonumber
D. strftime

Answer: D

Explanation:

The strftime function in Splunk is used to convert epoch time into a human-readable string format. It takes an epoch time value and a format string as arguments and returns the time as a formatted string. Other options, like strptime, convert string representations of time into epoch format, while tostring converts values to strings, and tonumber converts values to numbers.

Question: 42

Which of the following can be used to access external lookups?

A. Perl and Python
B. Python and Ruby
C. Perl and binary executable
D. Python and binary executable

Answer: D

Explanation:

Splunk supports external lookups that enrich search results using scripts or binary executables.
Python and binary executables are commonly used for creating these external lookups, as Python is widely supported, and binary executables can handle performance-critical tasks.

www.atmicnetworks.com

Question: 43

What file types does Splunk use to define geospatial lookups?

A. GPX or GML files
B. TXT files
C. KMZ or KML files
D. CSV files

Answer: C

Explanation:

Splunk uses KMZ or KML files to define geospatial lookups. These formats are designed for geographic annotation and mapping, making them ideal for geospatial data in Splunk.

Question: 44

If a nested macro expands to a search string that begins with a generating command, what additional syntax is needed?

A. Double tick marks around the nested macro.
B. A comma before the nested macro.
C. Square brackets around the nested macro.
D. A pipe character before the nested macro.

Answer: C

Explanation:

When a nested macro expands to a search string that begins with a generating command, square brackets are required to ensure proper interpretation. Square brackets allow the nested macro to be treated as a subsearch or command.

Question: 45

What is the correct hierarchy of XML elements in a dashboard panel?

A. <panel><dashboard><row>
B. <dashboard><row><panel>
C. <dashboard><panel><row>
D. <panel><row><dashboard>

Answer: B

Explanation:

www.atmicnetworks.com

The correct XML hierarchy for a dashboard panel is <dashboard><row><panel>. The <dashboard> element contains rows, and within each <row>, there are panels that hold visualizations or searches.

Question: 46

Why use the tstats command?

A. As an alternative to the summary command.
B. To generate statistics on indexed fields.
C. To generate an accelerated data model.
D. To generate statistics on search-time fields.

Answer: B

Explanation:

The tstats command is used to generate statistics on indexed fields, particularly from accelerated data models. It operates on indexed-time summaries, making it more efficient than using raw data.

The tstats command is used to generate statistics on indexed fields . It is highly efficient because it operates directly on indexed data (e.g., metadata or data model datasets) rather than raw event data.
Here's why this works:
Indexed Fields : Indexed fields include metadata fields like _time, host, source, and sourcetype, as well as fields defined in data models. Since these fields are preprocessed and stored in the index, querying them with tstats is faster than searching raw events.
Performance : tstats is optimized for large-scale searches and is particularly useful for summarizing data across multiple indexes or time ranges.
Data Models : tstats can also query data model datasets, making it a powerful tool for working with accelerated data models.

Question: 47

Which commands should be used in place of a subsearch if possible?

A. untable and/or xyseries
B. stats and/or eval
C. mvexpand and/or where
D. bin and/or where

Answer: B

Explanation:

stats and eval are recommended over subsearches because they are more efficient and scalable. Subsearches can be slow and resource-intensive, whereas stats aggregates data, and eval performs calculations within the search.

www.atmicnetworks.com

The stats and eval commands should be used instead of subsearches whenever possible because subsearches have performance limitations. They return only a maximum of 10,000 results or execute within 60 seconds by default, which may cause incomplete results. Using stats allows aggregation of large datasets efficiently, while eval can manipulate field values within a search rather than relying on subsearches.

Reference:
Splunk Documentation - Stats Command

Splunk Documentation - Eval Command

Question: 48

Which of the following would exclude all entries contained in the lookup file baditems.csv from search results?

A. NOT [inputlookup baditems.csv]
B. NOT (lookup baditems.csv OUTPUT item)
C. WHERE item NOT IN (baditems.csv)
D. [NOT inputlookup baditems.csv]

Answer: A

Explanation:

The correct way to exclude entries from the lookup file baditems.csv is using NOT [inputlookup baditems.csv]. This syntax excludes all entries in the lookup from the main search results.

Question: 49

What order of incoming events must be supplied to the transaction command to ensure correct results?

A. Reverse lexicographical order
B. Ascending lexicographical order
C. Ascending chronological order
D. Reverse chronological order

Answer: C

Explanation:

The transaction command requires events in ascending chronological order to group related events correctly into meaningful transactions.

www.atmicnetworks.com

Question: 50

What type of drilldown passes a value from a user click into another dashboard or external page?

A. Visualization
B. Event
C. Dynamic
D. Contextual

Answer: D

Explanation:

Contextual drilldown allows values from user clicks to be passed into another dashboard or external page, making dashboards interactive and responsive to user input.

Question: 51

When running a search, which Splunk component retrieves the individual results?

A. Indexer
B. Search head
C. Universal forwarder
D. Master node

Answer: B

Explanation:

The Search head (Option B) is responsible for initiating and coordinating search activities in a distributed environment. It sends search requests to the indexers (which store the data) and consolidates the results retrieved from them. The indexers store and retrieve the data, but the search head manages the user interaction and result aggregation.

Question: 52

What does the query | makeresults generate?

A. A timestamp
B. A results field
C. An error message
D. The results of the previously run search

Answer: B

Explanation:

The | makeresults command generates a single event containing default fields, such as _time. It's mainly used to create sample data or placeholder events for testing purposes. The primary field it generates is _time, but the command is used to generate a base event that can be manipulated further.

Question: 53

When using a nested search macro, how can an argument value be passed to the inner macro?

A. The argument value may be passed to the outer macro.
B. An argument cannot be used with an inner nested macro.
C. An argument cannot be used with an outer nested macro.
D. The argument value must be specified in the outer macro.

Answer: A

Explanation:

When using nested search macros, the argument value can be passed to the inner macro by specifying it in the outer macro. This allows dynamic arguments to flow into the inner macro, enabling flexible and reusable search logic.

Question: 54

What default Splunk role can use the Log Event alert action?

A. Power
B. User
C. can_delete
D. Admin

Answer: D

Explanation:

The Admin role (Option D) has the privilege to use the Log Event alert action, which logs an event to an index when an alert is triggered. Admins have the broadest range of permissions, including configuring and managing alert actions in Splunk.

The Admin role in Splunk has the necessary permissions to use the Log Event alert action . This action allows alerts to generate log entries in the _internal index, which can be useful for auditing or tracking alert activity.

Here's why this works:

Permissions Required : The Log Event alert action requires administrative privileges because it involves writing data to the _internal index, which is typically restricted to users with elevated permissions.

Default Roles : By default, only the Admin role has the required capabilities (edit_roles, schedule_search, and

write_to_internal_index) to configure and execute this alert action.

Question: 55

Which predefined drilldown token passes a clicked value from a table row?

A. $table.$
B. $rowclick.$
C. $row.$
D. $tableclick.$

Answer: C

Explanation:

The predefined drilldown token $row.$ passes the clicked value from a table row in Splunk dashboards. It allows you to capture the entire row of data when a user clicks on a table visualization. Here's why this works:
Purpose of $row.$: When a user clicks on a table row, $row.$ captures all the fields and their values for that row. This token is particularly useful for creating contextual drilldowns or passing multiple values to subsequent searches or panels.
Dynamic Behavior : Drilldown tokens like $row.$ enable dynamic interactions in dashboards, allowing users to filter or explore data based on their selections.
Other options explained:
Option A : Incorrect because $table.$ is not a valid predefined drilldown token.
Option B : Incorrect because $rowclick.$ is not a valid predefined drilldown token.
Option D : Incorrect because $tableclick.$ is not a valid predefined drilldown token.
Example:

```
<drilldown>
<set token="selected_row">$row.$</set>
</drilldown>
```

This sets the selected_row token to the clicked row's data, which can then be used in other parts of the dashboard.
Reference:
Splunk Documentation on Drilldown Tokens:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/DrilldownIntro
Splunk Documentation on Tokens:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/UseTokenstoBuildDynamicInputs

Question: 56

Which statement about the coalesce function is accurate?

A. It can take only a single argument.
B. It can take a maximum of two arguments.
C. It can be used to create a new field in the results set.
D. It can return null or non-null values.

Answer: C

Explanation:

The coalesce function returns the first non-null value from a list of fields, and it can be used within an eval expression to create a new field in the results set. This is useful when handling missing or inconsistent data across multiple fields.

Question: 57

Which command processes a template for a set of related fields?

A. bin
B. xyseries
C. foreach
D. untable

Answer: C

Explanation:

The foreach command applies a processing step to each field in a set of related fields. It allows repetitive operations to be applied to multiple fields in one go, streamlining tasks across several fields.

The foreach command in Splunk is used to process a template for a set of related fields. It allows you to iterate over multiple fields that share a common naming pattern and apply a transformation or operation to each of them. This is particularly useful when you have a series of similarly named fields (e.g., field1, field2, field3) and want to perform the same action on all of them without specifying each field individually.

For example, if you have fields like price1, price2, and price3, and you want to convert their values to integers, you can use the following syntax:

Reference:
Splunk Documentation on foreach:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/foreach

Question: 58

Which is a regex best practice?

A. Use complex expressions rather than simple ones.
B. Avoid backtracking.
C. Use greedy operators (.*) instead of non-greedy operators (.*?).
D. Use * rather than +.

Answer: B

Explanation:

One of the best practices in regex is to avoid backtracking, which can degrade performance by revisiting parts of the input multiple times. Optimizing regex patterns to prevent unnecessary backtracking improves efficiency, especially when dealing with large datasets.

Question: 59

What does using the tstats command with summariesonly=false do?

A. Returns results from only non-summarized data.
B. Returns results from both summarized and non-summarized data.
C. Prevents the use of wildcard characters in aggregate functions.
D. Returns no results.

Answer: B

Explanation:

Setting summariesonly=false in the tstats command retrieves results from both summarized (accelerated) and non-summarized (raw) data, allowing a more comprehensive analysis of both types of data in the same query.

Question: 60

Which of the following is an event handler action?

A. Run an eval statement based on a user clicking a value on a form.
B. Set a token to select a value from the time range picker.
C. Pass a token from a drilldown to modify index settings.
D. Cancel all jobs based on the number of search job results captured.

Answer: A

Explanation:

An event handler action can trigger an eval statement based on a user's interaction with a form. This makes dashboards interactive by allowing real-time updates based on user input, modifying the data presented dynamically.

Question: 61

Which of the following fields are provided by the fieldsummary command? (Select all that apply)

www.atmicnetworks.com

A. count
B. stdev
C. mean
D. dc

Answer: A, D

Explanation:

The fieldsummary command provides statistical summaries of fields, including the count of events containing the field (count) and the distinct count of field values (dc). Standard deviation (stdev) and mean are not provided by fieldsummary, but can be calculated using commands like stats.

Question: 62

Which of the following is accurate about cascading inputs?

A. They can be reset by an event handler.
B. The final input has no impact on previous inputs.
C. Only the final input of the sequence can supply a token to searches.
D. Inputs added to panels cannot participate.

Answer: A

Explanation:

Cascading inputs allow one input's selection to determine the options available in subsequent inputs. An event handler can reset the cascading sequence based on user interactions, ensuring the following inputs reflect appropriate options based on prior selections.

Cascading inputs in Splunk dashboards allow one input to dynamically update or influence another input. These inputs are often used to create dependent dropdowns or filters. One key feature of cascading inputs is that they can be reset by an event handler .

Here's why this works:

Cascading Behavior : Cascading inputs are designed to update dynamically based on user selections. For example, selecting a value in one dropdown might populate or filter the options in another

dropdown.

Resetting Inputs : Event handlers (e.g., change events) can reset or clear the values of cascading inputs when certain conditions are met. This ensures that the dashboard remains consistent and avoids invalid combinations of inputs.

Dynamic Tokens : Cascading inputs use tokens to pass values between inputs and searches. These tokens can be updated or cleared dynamically using event handlers.

Reference:

Splunk Documentation on Cascading Inputs:

https://docs.splunk.com/Documentation/Splunk/latest/Viz/Cascadinginputs

Splunk Documentation on Event Handlers:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/EventHandlerReference

Question: 63

Which element attribute is required for event annotation?

A. <search type="event_annotation">
B. <search style="annotation">
C. <search type=$annotation$>
D. <search type="annotation">

Answer: D

Explanation:

In Splunk dashboards, event annotations require the attribute <search type="annotation"> to define an event annotation, which marks significant events on visualizations like timelines.

Question: 64

Repeating JSON data structures within one event will be extracted as what type of fields?

A. Single value
B. Lexicographical
C. Multivalue
D. Mvindex

Answer: C

Explanation:

When Splunk encounters repeating JSON data structures in an event, they are extracted as multivalue fields. These allow multiple values to be stored under a single field, which is common with arrays in JSON data.

When Splunk extracts repeating JSON data structures within a single event, it represents them as multivalue fields . A multivalue field is a field that contains multiple values, which can be iterated over or expanded using commands like mvexpand or foreach.

Here's why this works:

JSON Data Extraction : Splunk automatically parses JSON data into fields. If a JSON key has an array of values (e.g., "products": ["productA", "productB", "productC"]), Splunk creates a multivalue field for that key.

Multivalue Fields : These fields allow you to handle multiple values for the same key within a single event. For example, if the JSON key products contains an array of product names, Splunk will store all the values in a single multivalue field named products.

www.atmicnetworks.com

{
"event": "purchase",
"products": ["productA", "productB", "productC"] }

Reference:
Splunk Documentation on JSON Data Extraction:
https://docs.splunk.com/Documentation/Splunk/latest/Data/ExtractfieldsfromJSON
Splunk Documentation on Multivalue Fields:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/MultivalueEvalFunctions

Question: 65

A report named "Linux logins" populates a summary index with the search string sourcetype=linux_secure | sitop src_ip user. Which of the following correctly searches against the summary index for this data?

A. index=summary sourcetype="linux_secure" | top src_ip user
B. index=summary search_name="Linux logins" | top src_ip user
C. index=summary search_name="Linux logins" | stats count by src_ip user
D. index=summary sourcetype="linux_secure" | stats count by src_ip user

Answer: C

Explanation:

The correct way to search against the summary index for this data is: index=summary search_name="Linux logins" | stats count by src_ip user

Here's why this works:
Summary Index : Summary indexes store pre-aggregated data generated by scheduled reports or saved searches. To query this data, you must specify the index=summary and filter by the

search_name field, which identifies the specific report that populated the summary index. Aggregation : The original search used sitop, which is designed for summary indexing. When querying the summary index, you should use stats to aggregate the pre-aggregated data further. Example:
index=summary search_name="Linux logins" | stats count by src_ip user

Reference:
Splunk Documentation on Summary Indexing:
https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Usesummaryindexing
Splunk Documentation on sitop:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/sitop

Question: 66

Which statement about .tsidx files is accurate?

A. A .tsidx file consists of a lexicon and a posting list.
B. Splunk removes outdated .tsidx files every 5 minutes.
C. Splunk updates .tsidx files every 30 minutes.
D. Each bucket in each index may contain only one .tsidx file.

Answer: A

Explanation:

A .tsidx (time-series index) file in Splunk consists of two main components:

Lexicon : A dictionary of unique terms (e.g., field names and values) extracted from indexed data.

Posting List : A mapping of terms in the lexicon to the locations (offsets) of events containing those terms.

Here's why this works:

Purpose of .tsidx Files : These files enable fast searching by indexing terms and their locations in the raw data. They are critical for efficient search performance.

Structure : The lexicon ensures that each term is stored only once, while the posting list links terms to their occurrences in events.

Other options explained:

Option B : Incorrect because Splunk does not remove .tsidx files every 5 minutes. These files are part of the index and persist until the associated data is aged out or manually deleted.

Option C : Incorrect because .tsidx files are updated as data is indexed, not at fixed intervals like every 30 minutes.

Option D : Incorrect because each bucket can contain multiple .tsidx files, depending on the volume of indexed data.

Reference:

Splunk Documentation on .tsidx Files:

https://docs.splunk.com/Documentation/Splunk/latest/Indexer/HowSplunkstoresindexes

Splunk Documentation on Indexing:

https://docs.splunk.com/Documentation/Splunk/latest/Indexer/Howindexingworks

Question: 67

Which of the following is not a common default time field?

A. date_zone
B. date_minute
C. date_year
D. date_day

Answer: A

Explanation:

Fields like date_minute, date_year, and date_day are common default time fields in Splunk, while date_zone is not typically a default field for time-related data.

Question: 68

What is a performance improvement technique unique to dashboards?

A. Using stats instead of transaction
B. Using global searches
C. Using report acceleration
D. Using data model acceleration

Answer: B

Explanation:

In Splunk, dashboards are powerful tools for visualizing and analyzing data. However, as dashboards grow in complexity and the volume of data increases, performance optimization becomes critical. One technique unique to dashboards is the use of global searches .

What Are Global Searches?

A global search allows multiple panels within a dashboard to share the same base search. Instead of each panel running its own independent search, all panels derive their results from a single, shared search. This reduces the computational load on the Splunk instance because it eliminates redundant searches and ensures that the data is processed only once.

Why Is This Unique to Dashboards?

Global searches are specifically designed for dashboards where multiple panels often rely on the

same dataset or search logic. By consolidating the search into one query, Splunk avoids duplicating effort, which improves performance significantly. This technique is not applicable to standalone searches or reports, making it unique to dashboards.

Comparison with Other Options:

B . Using data model acceleration:

Data model acceleration (DMA) is a powerful feature for speeding up searches over large datasets by precomputing and storing summarized data. However, it is not unique to dashboards—it can be used in any type of search or report.

C . Using stats instead of transaction:

Replacing transaction commands with stats is a general best practice for improving search performance. While this is a valid optimization technique, it applies universally across Splunk and is not specific to dashboards.

D . Using report acceleration:

Report acceleration is another general-purpose optimization technique that speeds up saved searches by creating summaries of the data. Like DMA, it is not exclusive to dashboards.

Benefits of Global Searches:

Reduced Search Load: By sharing a single search across multiple panels, the number of searches executed is minimized.

Faster Dashboard Loading: Since the data is fetched once and reused, dashboards load faster. Consistent Results: All panels using the global search will display consistent results derived from the same dataset.

Example of Global Search in a Dashboard:

```
<dashboard>
<search id="base_search">
  <query>index=main sourcetype=access_combined | fields clientip, status, method</query>
</search>
<panel>
  <title>Status Codes</title>
  <table>
    <search base="base_search">
      <query>| stats count by status</query>
    </search>
  </table>
</panel>
<panel>
  <title>Top Clients</title>
  <chart>
    <search base="base_search">
      <query>| top clientip</query>
    </search>
  </chart>
</panel>
</dashboard>
```

In this example, the base_search is defined once and reused by both panels. Each panel adds additional processing (stats or top) to the shared results, reducing redundancy.

Reference:

Splunk Documentation - Dashboard Best Practices:

https://docs.splunk.com/Documentation/Splunk/latest/Viz/BestPractices

This document highlights the importance of global searches for optimizing dashboard performance.

Splunk Documentation - Global Searches:

https://docs.splunk.com/Documentation/Splunk/latest/Viz/PanelreferenceforSimplifiedXML#Global_searches

Detailed explanation of how global searches work and their implementation in dashboards.

Splunk Core Certified Power User Learning Path:

The official Splunk training materials emphasize the use of global searches as a key technique for improving dashboard performance.

By leveraging global searches, users can ensure their dashboards remain efficient and responsive even as data volumes grow.

This makes Option A the correct and verified answer.

Question: 69

Which of these generates a summary index containing a count of events by product_id?

A. stats si(product_id)
B. stats count by product_id
C. sistats count by product_id
D. sistats summary index by product_id

Answer: C

Explanation:

The correct command to generate a summary index containing a count of events by product_id is: sistats count by product_id

Here's why this works:

sistats : This command is specifically designed for creating summary indexes. It pre-aggregates data and stores it in a format optimized for fast retrieval.

count by product_id : This part of the command calculates the count of events grouped by the product_id field.

Summary indexing is useful when you want to store pre-aggregated data for faster reporting. For example, instead of querying raw data every time, you can query the summary index to get quick results.

Other options explained:

Option A : Incorrect because stats si(product_id) is invalid syntax.

Option B : Incorrect because stats is used for real-time aggregation but does not create summary indexes.

Option D : Incorrect because sistats summary index by product_id is invalid syntax.

Example:

index=main | sistats count by product_id

Reference:

Splunk Documentation on sistats:

https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/sistats

Splunk Documentation on Summary Indexing:

https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Usesummaryindexing

Question: 70

When and where do search debug messages appear to help with troubleshooting views?

A. In the Dashboard Editor, while the search is running.
B. In the Search Job Inspector, after the search completes.
C. In the Search Job Inspector, while the search is running.
D. In the Dashboard Editor, after the search completes.

Answer: C

Explanation:

Search debug messages appear in the Search Job Inspector while the search is running. This tool provides detailed insights into search performance and potential issues, making it helpful for troubleshooting.

www.atmicnetworks.com

Question: 71

What function can be used as an alternative to coalesce to return the first value from a list of fields that is not null?

A. bin
B. case
C. exact
D. mvzip

Answer: B

Explanation:
Comprehensive and Detailed Step by Step
The case function can be used as an alternative to coalesce to return the first non-null value. While coalesce(field1, field2, field3) will return the first non-null value, case(condition1, value1, condition2, value2, ...) allows more flexibility by evaluating conditions.

Reference:
Splunk Documentation - case Function

Question: 72

Which of the following cannot be accomplished with a webhook alert action?

A. Retrieve data from a web page
B. Create a ticket in a support app
C. Post a notification on a web page
D. Post a message in a chatroom

Answer: A

Explanation:
Comprehensive and Detailed Step by Step
A webhook in Splunk is designed to send HTTP POST requests to a specified URL when an alert is triggered. This mechanism allows Splunk to communicate with external systems by pushing data to them. Common use cases for webhooks include:
Creating a ticket in a support application: By sending a POST request to the support application's API endpoint with the necessary details, a new ticket can be created automatically.
Posting a notification on a web page: If the web page has an API that accepts POST requests, Splunk can send data to it, resulting in a notification being displayed.
Posting a message in a chatroom: Many chat platforms offer webhook integrations where POST requests can send messages to specific channels or chatrooms.
However, retrieving data from a web page is not within the capabilities of a webhook. Webhooks are designed for outbound communication (sending data) and do not handle inbound requests or data retrieval. To fetch or retrieve data from external sources, other methods such as scripted inputs or custom scripts would be required.
Reference:

www.atmicnetworks.com

Splunk Documentation: Set up alert actions

Question: 73

What is used to separate multiple tokens when creating a drilldown in XML?

A. A pipe character (|)
B. A comma (,)
C. An escaped ampersand (&)
D. An escaped double quote (\")

Answer: C

Explanation:

Comprehensive and Detailed Step by Step

In Splunk XML dashboards, multiple tokens must be separated using an escaped ampersand (&), which prevents syntax errors and ensures that tokens are correctly passed in drilldowns.

Reference:

Splunk Documentation - Token Usage

Question: 74

Which of the following most accurately defines a base search?

A. A dashboard panel query used by a drilldown.
B. A search query used by post-process searches.
C. A search query hidden in the XML.
D. A search query that uses | tstats used by post-process searches.

Answer: B

Explanation:

A base search in Splunk is a foundational search query defined within a dashboard that can be referenced by multiple panels. This approach promotes efficiency by allowing multiple panels to display different aspects or visualizations of the same dataset without executing separate searches for each panel.

Key Points:

Definition: A base search is a primary search defined once in a dashboard's XML and referenced by other panels through post-process searches.

Post-Process Searches: These are additional search commands applied to the results of the base search. They refine or transform

the base search results to meet specific panel requirements. Benefits:

Performance Optimization: Reduces the number of searches executed, thereby conserving system resources.

Consistency: Ensures all panels referencing the base search use the same dataset, maintaining uniformity across the dashboard.

Example:

Consider a dashboard that needs to display various statistics about web traffic:

Base Search:

```
<search name="base_search">
index=web_logs | stats count by status_code
</search>
```

Panel 1 (Total Requests):

```
<panel>
<title>Total Requests</title>
<search base="base_search">
  | stats sum(count) as total_requests
</search>
</panel>
```

Panel 2 (Error Rate):

```
<panel>
<title>Error Rate</title>
<search base="base_search">
  | where status_code >= 400
  | stats sum(count) as error_count
</search>
</panel>
```

In this example:

The base_search retrieves the count of events grouped by status_code from the web_logs index.

Panel 1 calculates the total number of requests by summing the count field.

Panel 2 filters for error status codes (400 and above) and calculates the total number of errors. By defining a base search, both panels utilize the same initial dataset, ensuring consistency and reducing redundant processing.

Reference:

Splunk Documentation - Base Search

Question: 75

Which of the following elements sets a token value of sourcetype=access_combined?

A. <set token="NewToken">sourcetype=$click.value$</set>

B. <set token="NewToken"> prefix="sourcetype=">$click.value$</set>

C. <set token="NewToken">sourcetype=$click.value$</set>
D. <set token="NewToken" prefix="sourcetype=">$click.value$</set>

Answer: D

Explanation:

In Splunk, tokens are used in dashboards to dynamically pass values between different components, such as dropdowns, text inputs, or clickable elements. The <set> tag is a Simple XML element that allows you to define or modify the value of a token. When setting a token value, you can use attributes like prefix and suffix to construct the desired value format.

Question Analysis:

The goal is to set a token named NewToken with the value sourcetype=access_combined. This requires constructing the token value by combining a static prefix (sourcetype=) with a dynamic value (e.g., $click.value$, which represents the value clicked or selected by the user).

Why Option D Is Correct:

The prefix attribute in the <set> tag allows you to prepend a static string to the dynamic value. In this case:
The prefix="sourcetype=" ensures that the token starts with the string sourcetype=.
The $click.value$ dynamically appends the selected or clicked value to the token.
For example, if $click.value$ is access_combined, the resulting token value will be sourcetype=access_combined.

Example Use Case:

Suppose you have a dashboard with a clickable chart where users can select a sourcetype. You want to set a token (NewToken) to capture the selected sourcetype in the format sourcetype=<selected_value>. The following XML snippet demonstrates how this works:

```
<dashboard>
<row>
  <panel>
    <html>
      <a href="#" onclick="setToken('NewToken', 'sourcetype=access_combined')">Set Token</a>
    </html>
  </panel>
</row>
<row>
  <panel>
    <table>
      <search>
        <query>index=_internal $NewToken$ | stats count by sourcetype</query>
      </search>
    </table>
  </panel>
</row>
</dashboard>
```

In this example:
Clicking the link triggers the <set> logic.
The token NewToken is set to sourcetype=access_combined.
The search query uses $NewToken$ to filter results based on the selected sourcetype.

Reference:

www.atmicnetworks.com

Splunk Documentation - Token Usage in Dashboards:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/TokenReference
This document explains how tokens work in Splunk dashboards, including the use of <set> tags and attributes like prefix and suffix.
Splunk Documentation - Dynamic Drilldowns:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/Dynamicdrilldownindashboards
This resource provides examples of how to use tokens for dynamic interactions in dashboards.
Splunk Core Certified Power User Learning Path:
The official training materials cover token manipulation and dynamic dashboard behavior, including the use of <set> tags.
By using the prefix attribute correctly, Option D ensures that the token value is constructed in the desired format (sourcetype=access_combined), making it the verified and correct answer.

Question: 76

Which of the following drilldown methods does not exist in dynamic dashboards?

A. Contextual Drilldown
B. Dynamic Drilldown
C. Custom Drilldown
D. Static Drilldown

Answer: D

Explanation:
Comprehensive and Detailed Step-by-Step
In Splunk dashboards, drilldown methods define how user interactions with visualizations (such as clicking on a chart or table) trigger additional actions or navigate to more detailed information. Understanding the available drilldown methods is crucial for designing interactive and responsive dashboards.
Drilldown Methods in Dynamic Dashboards:
A . Contextual Drilldown:
Contextual drilldown refers to the default behavior where clicking on a visualization element filters the dashboard based on the clicked value. For example, clicking on a bar in a bar chart might filter the dashboard to show data specific to that category.
B . Dynamic Drilldown:
Dynamic drilldown allows for more advanced interactions, such as navigating to different dashboards or external URLs based on the clicked data. This method can be customized using tokens and conditional logic to provide a tailored user experience.
C . Custom Drilldown:
Custom drilldown enables developers to define specific actions that occur upon user interaction. This can include setting tokens, executing searches, or redirecting to custom URLs. It provides flexibility to design complex interactions beyond the default behaviors.
D . Static Drilldown:
The term "Static Drilldown" is not recognized in Splunk's documentation or dashboard configurations. Drilldowns in Splunk are inherently dynamic, responding to user interactions to provide more detailed insights. Therefore, "Static Drilldown" does not exist as a method in dynamic dashboards.

www.atmicnetworks.com

Conclusion:

Among the options provided, Static Drilldown is not a recognized drilldown method in Splunk's dynamic dashboards. Splunk's drilldown capabilities are designed to be interactive and responsive, allowing users to explore data in depth through contextual, dynamic, and custom interactions. Reference:

Splunk Documentation: Drilldown actions in dashboards

The stats command in Splunk is used to perform statistical operations on data, such as calculating counts, averages, sums, and other aggregations. When working with accelerated data models or report acceleration, Splunk may generate summaries of the data to improve performance. These summaries are precomputed and stored to speed up searches.

The summariesonly argument in the stats command controls whether the search should use only summarized data (summariesonly=true) or include both summarized and non-summarized (raw) data (summariesonly=false). By default, summariesonly is set to false.

Question Analysis:

The question asks what happens when you use the stats command with summariesonly=false. Let's analyze each option:

A . Returns results from both summarized and non-summarized data.

This is the correct answer. When summariesonly=false, Splunk includes both summarized data (if available) and raw data in the results. This ensures that all relevant data is considered, even if some data has not been summarized yet.

B . Returns results from only non-summarized data.

This is incorrect. Setting summariesonly=false does not exclude summarized data; it includes both summarized and non-summarized data.

C . Returns no results.

This is incorrect. The stats command will always return results unless there is an issue with the query or no data matches the search criteria. Setting summariesonly=false does not cause the search to return no results.

D . Prevents use of wildcard characters in aggregate functions.

This is incorrect. The summariesonly argument has no effect on the use of wildcard characters in aggregate functions. Wildcard behavior is unrelated to this setting.

Why Option A Is Correct:

When summariesonly=false, Splunk combines summarized data (from accelerated data models or report acceleration) with raw data to ensure completeness. This is particularly useful in scenarios where:

Not all data has been summarized yet.

You want to ensure that your results are comprehensive and include the latest data that may not yet be part of the summary.

For example, consider a scenario where you have an accelerated data model summarizing logs for the past 30 days. If you run a search with stats summariesonly=false, Splunk will include both the summarized data (for the past 30 days) and any new, non-summarized data (e.g., logs from today).

| stats count by sourcetype summariesonly=false

In this example:

If summaries exist for some data, they will be included in the results.

Any raw data that has not been summarized will also be included.

The final output will reflect the combined results from both summarized and non-summarized data. Key Points About summariesonly:

Default Behavior: The default value of summariesonly is false, meaning both summarized and non-summarized data are included by default.

Use Case for summariesonly=true: If you want to restrict the search to only summarized data (e.g., for faster performance),

www.atmicnetworks.com

you can set summariesonly=true.

Impact on Results: Using summariesonly=false ensures that your results are complete, even if some data has not been summarized.

Reference:

Splunk Documentation - stats Command:

https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/stats

This document explains the stats command and its arguments, including summariesonly.

Splunk Documentation - Data Model Acceleration:

https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Acceleratedatamodels

This resource provides details about how data model acceleration works and the role of summaries in accelerated searches.

Splunk Core Certified Power User Learning Path:

The official training materials cover the use of the stats command and its interaction with summarized data.

By ensuring that both summarized and non-summarized data are included, summariesonly=false provides the most comprehensive results, making Option A the verified and correct answer.

Question: 77

What does Splunk recommend when using the Field Extractor and Interactive Field Extractor (IFX)?

A. Use the Field Extractor for structured data and the IFX for unstructured data.
B. Use the IFX for structured data and the Field Extractor for unstructured data.
C. Use both tools interchangeably for any data type.
D. Avoid using both tools for field extraction.

Answer: A

Explanation:

Comprehensive and Detailed Step-by-Step

Splunk provides two primary tools for creating field extractions: the Field Extractor and the Interactive Field Extractor (IFX). Each tool is optimized for different data structures, and understanding their appropriate use cases ensures efficient and accurate field extraction. Field Extractor:

Purpose: Designed for structured data, where events have a consistent format with fields separated by common delimiters (e.g., commas, tabs).

Method: Utilizes delimiter-based extraction, allowing users to specify the delimiter and assign names to the extracted fields.

Use Case: Ideal for data like CSV files or logs with a predictable structure.

Interactive Field Extractor (IFX):

Purpose: Tailored for unstructured data, where events lack a consistent format, making it challenging to extract fields using simple delimiters.

Method: Employs regular expression-based extraction. Users can highlight sample text in events, and IFX generates regular expressions to extract similar patterns across events.

Use Case: Suitable for free-form text logs or data with varying structures.

Best Practices:

Structured Data: For data with a consistent and predictable structure, use the Field Extractor to define field extractions based on delimiters. This method is straightforward and efficient for such data types.

Unstructured Data: When dealing with data that lacks a consistent format, leverage the Interactive Field Extractor (IFX). By highlighting sample text, IFX assists in creating regular expressions to accurately extract fields from complex or irregular data.

Conclusion:

Splunk recommends using the Field Extractor for structured data and the Interactive Field Extractor (IFX) for unstructured data. This approach ensures that field extractions are tailored to the data's structure, leading to more accurate and efficient data parsing.

Reference:

Splunk Documentation: Build field extractions with the field extractor

Question: 78

Which of the following is a valid use of the eval command?

A. To filter events based on a condition.
B. To calculate the sum of a numeric field across all events.
C. To create a new field based on an existing field's value.
D. To group events by a specific field.

Answer: C

Explanation:

Comprehensive and Detailed Step-by-Step

The eval command in Splunk is a versatile tool used for manipulating and creating fields during search time. It allows users to perform calculations, convert data types, and generate new fields based on existing data.

Primary Uses of the eval Command:

Creating New Fields: One of the most common uses of eval is to create new fields by transforming existing data. For example, extracting a substring, performing arithmetic operations, or concatenating strings.

Example:

spl

CopyEdit

```
| eval full_name = first_name . " " . last_name
```

This command creates a new field called full_name by concatenating the first_name and last_name fields with a space in between.

Conditional Processing: eval can be used to assign values to a field based on conditional logic, similar to an "if-else" statement.

Example:

spl

CopyEdit

```
| eval status = if(response_time > 1000, "slow", "fast")
```

This command creates a new field called status that is set to "slow" if the response_time exceeds 1000 milliseconds; otherwise, it's set to "fast".

Analysis of Options:

A . To filter events based on a condition:

Filtering events is typically achieved using the where command or by specifying conditions directly in the search criteria. While eval can be used to create fields that represent certain conditions, it doesn't directly filter events.

B . To calculate the sum of a numeric field across all events:

Calculating the sum across events is performed using the stats command with the sum() function. eval operates on a per-event basis and doesn't aggregate data across multiple events.

C . To create a new field based on an existing field's value:

This is a primary function of the eval command. It allows for the creation of new fields by transforming or manipulating existing field values within each event.

D . To group events by a specific field:

Grouping events is accomplished using commands like stats, chart, or timechart with a by clause. eval doesn't group events but can be used to create or modify fields that can later be used for grouping.

Conclusion:

The eval command is best utilized for creating new fields or modifying existing fields within individual events. Therefore, the valid use of the eval command among the provided options is to create a new field based on an existing field's value.

Reference:

Splunk Documentation: eval command

Question: 79

What is the purpose of the rex command in Splunk?

A. To extract fields using regular expressions.
B. To remove duplicate events from search results.
C. To rename fields in the search results.
D. To sort events based on a specified field.

Answer: A

Explanation:

The rex command in Splunk is a powerful tool used for field extraction by applying regular expressions (regex) to raw event data. It allows users to define patterns that match specific parts of the data and extract them as fields. This is particularly useful when working with unstructured or semi-structured data, where fields are not automatically extracted.

Question Analysis:

The question asks about the purpose of the rex command. Let's analyze each option:

A . To extract fields using regular expressions.

This is the correct answer. The primary purpose of the rex command is to extract fields from raw data using regex patterns. For example, you can use rex to parse key-value pairs, timestamps, or other structured elements embedded in unstructured logs.

B . To remove duplicate events from search results.

This is incorrect. The dedup command is used to remove duplicate events, not the rex command.

C . To rename fields in the search results.

This is incorrect. The rename command is used to rename fields, not the rex command.

D . To sort events based on a specified field.

This is incorrect. The sort command is used to sort events, not the rex command.

Why Option A Is Correct:

The rex command is specifically designed for field extraction using regular expressions . Regular expressions are patterns that

describe how to match text in the data. By defining these patterns, you can extract specific portions of the raw data and assign them to fields.

For example, consider the following log entry:

Copy

```
1
User=john Action=login Status=success
```

You can use the rex command to extract the User, Action, and Status fields:

spl

Copy

```
1
| rex "User=(?<user>\w+) Action=(?<action>\w+) Status=(?<status>\w+)"
```

In this example:

The rex command uses a regex pattern to identify and extract the values for User, Action, and Status.

The extracted values are assigned to the fields user, action, and status.

Key Features of the rex Command:

Field Extraction: Extracts fields from raw data using regex patterns.

Customization: Allows you to define custom field names for the extracted values.

Flexibility: Works with both structured and unstructured data, making it versatile for various use cases.

Example Use Cases:

Extracting Key-Value Pairs:

Suppose your logs contain key-value pairs like key=value. You can use rex to extract these pairs into fields:

```
| rex "key1=(?<field1>\w+) key2=(?<field2>\w+)"
```

Parsing Timestamps:

If your logs include timestamps in a specific format, you can use rex to extract and parse them:

```
| rex "EventTime=(?<timestamp>\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2})"
```

Extracting IP Addresses:

To extract IP addresses from logs:

```
| rex "ClientIP=(?<ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
```

Reference:

Splunk Documentation - rex Command:

https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/rex

This document provides detailed information about the syntax and usage of the rex command.

Splunk Documentation - Regular Expressions:

https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Aboutregularexpressions

This resource explains how regular expressions work and their role in field extraction.

Splunk Core Certified Power User Learning Path:

The official training materials cover the rex command extensively, including examples and best practices for field extraction.

By enabling users to extract fields using regular expressions, the rex command plays a critical role in transforming raw data into structured, queryable fields. This makes Option A the verified and correct answer.

www.atmicnetworks.com

Question: 80

The field products contains a multivalued field containing the names of products. What is the result of the command mvexpand products limit=<x>?

A. Compressed values in products will be uncompressed.
B. Separate events will be created for each product in products.
C. products will be converted from a single value field to a multivalue field.
D. All multivalue fields will be converted to single value fields.

Answer: B

Explanation:
Comprehensive and Detailed Step by Step
The mvexpand command in Splunk is used to expand multivalue fields into separate events. When you use mvexpand on a field like products, which contains multiple values, it creates a new event for each value in the multivalue field. For example, if the products field contains the values [productA, productB, productC], running mvexpand products will create three separate events, each containing one of the values (productA, productB, or productC).
The optional limit=<x> parameter specifies the maximum number of values to expand. If limit=2, only the first two values (productA and productB) will be expanded into separate events, and any remaining values will be ignored.
Key points about mvexpand:
It works only on multivalue fields.
It does not modify the original field but creates new events based on its values.
The limit parameter controls how many values are expanded.
Example:

```
| makeresults
| eval products="productA,productB,productC"
| makemv delim="," products
| mvexpand products
```

This will produce three separate events, one for each product.
Reference:
Splunk Documentation on mvexpand:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/mvexpand

Question: 81

Which of the following groups of commands can use multivalue functions?

A. eval, fieldformat, and where
B. eval, fields, and where
C. fieldformat, search, and where
D. eval, mvexpand, and makemv

Answer: D

Explanation:

Comprehensive and Detailed Step by Step

Multivalue functions in Splunk are used to manipulate fields that contain multiple values. The correct group of commands that can use multivalue functions is: Copy 1 eval, mvexpand, and makemv Here's why this works:

eval : This command can use multivalue functions like mvappend(), mvcount(), and mvjoin() to manipulate multivalue fields.

mvexpand : This command expands multivalue fields into separate events, making it easier to work with individual values.

makemv : This command splits a single-value field into a multivalue field based on a delimiter.

Other options explained:

Option A : Incorrect because fieldformat is used for formatting display values and does not support multivalue functions.

Option B : Incorrect because fields is used to include or exclude fields but does not handle multivalue fields.

Option C : Incorrect because fieldformat and search do not support multivalue functions. Example: | makeresults

```
| eval products="productA,productB,productC"
| makemv delim="," products
| mvexpand products
```

Reference:

Splunk Documentation on Multivalue Functions:

https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/MultivalueEvalFunctions

Splunk Documentation on mvexpand:

https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/mvexpand

Question: 82

What is the value of base lispy in the Search Job Inspector for the search index=web clientip=76.169.7.252?

A. [index::web AND 169 252 7 76]

B. [AND 169 252 7 76 index::web]

C. [169 AND 252 AND 7 AND 76 index::web]

D. [index::web 169 AND 252 AND 7 AND 76]

Answer: A

Explanation:

Comprehensive and Detailed Step by Step

The base lispy value in the Search Job Inspector represents the internal representation of the search query after it has been parsed and optimized by Splunk. It shows how Splunk interprets the query in terms of logical operations and field-value pairs.

For the search:

Copy 1

```
index=web clientip=76.169.7.252
```

The base lispy value will be: Copy 1

```
[ index::web AND 169 252 7 76 ]
```

Here's why this is correct:

Index Matching : The index::web part specifies that the search is scoped to the web index. Field-Value Matching : The clientip field is broken down into its individual components (76, 169, 7, 252) for efficient matching using bloom filters and other optimizations.

Logical AND : Splunk combines these components with an AND operator to ensure all conditions are met.

Other options explained:

Option B : Incorrect because the order of AND and the components is incorrect.

Option C : Incorrect because the components are not properly grouped with the index.

Option D : Incorrect because the AND operator is misplaced, and the structure does not match Splunk's internal representation.

Reference:

Splunk Documentation on Search Job Inspector:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Viewsearchjobproperties

Splunk Documentation on Bloom Filters:
https://docs.splunk.com/Documentation/Splunk/latest/Indexer/Bloomfilters

Question: 83

Which of the following statements is correct regarding bloom filters?

A. Hot buckets have no bloom filters as their contents are always changing.
B. Bloom filters could return false positives or false negatives.
C. Each bucket uses a unique hashing algorithm to create its bloom filter.
D. The bloom filter contains trinary values: 0, 1, and 2.

Answer: A

Explanation:

Comprehensive and Detailed Step by Step

The correct statement about bloom filters in Splunk is:

Copy

1

Hot buckets have no bloom filters as their contents are always changing.

Here's why this is correct:

Bloom Filters : Bloom filters are data structures used by Splunk to quickly determine whether a specific value exists in a bucket. They are designed for cold and warm buckets where the data is static.

Hot Buckets : Hot buckets contain actively ingested data, which is constantly changing. Since bloom filters are precomputed and immutable, they cannot be applied to hot buckets.

Other options explained:

Option B : Incorrect because bloom filters can only return false positives (indicating a value might exist when it doesn't), but they never return false negatives.

Option C : Incorrect because all buckets use the same hashing algorithm to create bloom filters.

Option D : Incorrect because bloom filters only contain binary values (0 or 1), not trinary values. Reference:

Splunk Documentation on Bloom Filters:
https://docs.splunk.com/Documentation/Splunk/latest/Indexer/Bloomfilters

Splunk Documentation on Buckets:
https://docs.splunk.com/Documentation/Splunk/latest/Indexer/HowSplunkstoresindexes

www.atmicnetworks.com

Question: 84

Which is generally the most efficient way to run a transaction?

A. Run the search query in Smart Mode.
B. Using | sort before the transaction command.
C. Run the search query in Fast Mode.
D. Rewrite the query using stats instead of transaction.

Answer: D

Explanation:
Comprehensive and Detailed Step by Step
The most efficient way to run a transaction is to rewrite the query using stats instead of transaction whenever possible. The transaction command is computationally expensive because it groups events based on complex criteria (e.g., time constraints, shared fields, etc.) and performs additional operations like concatenation and duration calculation.
Here's why stats is more efficient:
Performance : The stats command is optimized for aggregating and summarizing data. It is faster and uses fewer resources compared to transaction.
Use Case : If your goal is to group events and calculate statistics (e.g., count, sum, average), stats can often achieve the same result without the overhead of transaction.
Limitations of transaction : While transaction is powerful, it is best suited for specific use cases where you need to preserve the raw event data or calculate durations between events.
Example: Instead of:
| transaction session_id

You can use:

| stats count by session_id

Other options explained:
Option A : Incorrect because Smart Mode does not inherently optimize the transaction command. Option B : Incorrect because sorting before transaction adds unnecessary overhead and does not address the inefficiency of transaction.
Option C : Incorrect because Fast Mode prioritizes speed but does not change how transaction operates.
Reference:
Splunk Documentation on transaction:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Transaction
Splunk Documentation on stats:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Stats

Question: 85

Which command is the opposite of untable?

A. chart
B. table
C. bin
D. xyseries

Answer: A

Explanation:
Comprehensive and Detailed Step by Step
The untable command in Splunk converts tabular data (rows and columns) into a format where each row represents a key-value pair. Its opposite is the chart command, which aggregates data into a tabular format with rows and columns.
Here's why chart is the opposite of untable:
untable : This command takes structured data (e.g., a table with columns A, B, C) and transforms it into a long format where each row contains a key-value pair (e.g., field, value).
chart : This command aggregates data into a structured table format, grouping data by specified fields and calculating statistics (e.g., count, sum).
Example: Using untable: spl
Copy 1 | untable _time field value This converts a table into key-value pairs. Using chart: spl
Copy 1 | chart count by field
This aggregates data into a structured table.
Other options explained:
Option B : Incorrect because table simply selects specific fields for display but does not aggregate data like chart.
Option C : Incorrect because bin is used for bucketing numeric or time-based data, not for creating tables.
Option D : Incorrect because xyseries transforms data into a series format but does not directly reverse the effect of untable.
Reference:
Splunk Documentation on untable:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/untable
Splunk Documentation on chart:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/chart

Question: 86

What is the default time limit for a subsearch to complete?

A. 10 minutes
B. 120 seconds
C. 5 minutes
D. 60 seconds

Answer: D

Explanation:

The default time limit for a subsearch to complete in Splunk is 60 seconds . If the subsearch exceeds this time limit, it will terminate,

and the outer search may fail or produce incomplete results.

Here's why this works:

Subsearch Timeout : Subsearches are designed to execute quickly and provide results to the outer search. To prevent performance issues, Splunk imposes a default timeout of 60 seconds.

Configuration : The timeout can be adjusted using the subsearch_maxout and subsearch_timeout settings in limits.conf, but the default remains 60 seconds.

Other options explained:

Option A : Incorrect because 10 minutes (600 seconds) is far longer than the default timeout.

Option B : Incorrect because 120 seconds is double the default timeout.

Option C : Incorrect because 5 minutes (300 seconds) is also longer than the default timeout.

Example: If a subsearch takes longer than 60 seconds to complete, you might see an error like:

Error in 'search': Subsearch exceeded configured timeout.

Reference:

Splunk Documentation on Subsearches:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutsubsearches

Splunk Documentation on limits.conf:
https://docs.splunk.com/Documentation/Splunk/latest/Admin/Limitsconf

Question: 87

Which command calculates statistics on search results as each search result is returned?

A. streamstats
B. fieldsummary
C. eventstats
D. appendpipe

Answer: A

Explanation:

Comprehensive and Detailed Step by Step

The streamstats command calculates statistics on search results as each event is processed , maintaining a running total or other cumulative calculations. Unlike eventstats, which calculates statistics for the entire dataset at once, streamstats processes events sequentially.

Here's why this works:

Purpose of streamstats : This command is ideal for calculating cumulative statistics, such as running totals, averages, or counts, as events are returned by the search.

Sequential Processing : streamstats applies statistical functions (e.g., count, sum, avg) incrementally to each event based on the order of the results.

```
| makeresults count=5
| streamstats count as running_count
```

This will produce:

www.atmicnetworks.com

_time running_count

<current_timestamp> 1
<current_timestamp> 2
<current_timestamp> 3
<current_timestamp> 4
<current_timestamp> 5

Other options explained:
Option B : Incorrect because fieldsummary generates summary statistics for all fields in the dataset, not cumulative statistics.
Option C : Incorrect because eventstats calculates statistics for the entire dataset at once, not incrementally.
Option D : Incorrect because appendpipe is used to append additional transformations or calculations to existing results, not for cumulative statistics.
Reference:
Splunk Documentation on streamstats:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Streamstats
Splunk Documentation on Statistical Commands:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/StatisticalAggregatingCommands

Question: 88

Which of the following is true about a KV Store Collection when using it as a lookup?

A. Each collection must have at least 3 fields, one of which needs to match values of a field in your event data.
B. Each collection must have at least 2 fields, one of which needs to match values of a field in your event data.
C. Each collection must have at least 2 fields, none of which need to match values of a field in your event data.
D. Each collection must have at least 3 fields, none of which need to match values of a field in your event data.

Answer: B

Explanation:
Comprehensive and Detailed Step by Step
When using a KV Store Collection as a lookup in Splunk, each collection must have at least 2 fields , and one of these fields must match values of a field in your event data . This matching field serves as the key for joining the lookup data with your search results.
Here's why this works:
Minimum Fields Requirement : A KV Store Collection must have at least two fields: one to act as the key (matching a field in your event data) and another to provide additional information or context.
Key Matching : The matching field ensures that the lookup can correlate data from the KV Store with your search results. Without this, the lookup would not function correctly.
Other options explained:
Option A : Incorrect because a KV Store Collection does not require at least 3 fields; 2 fields are sufficient.
Option C : Incorrect because at least one field in the collection must match a field in your event data for the lookup to work.

Option D : Incorrect because a KV Store Collection does not require at least 3 fields, and at least one field must match event data.

Example: If your event data contains a field user_id, and your KV Store Collection has fields user_id and user_name, you can use the lookup command to enrich your events with user_name based on the matching user_id.

Reference:

Splunk Documentation on KV Store Lookups:

https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/ConfigureKVstorelookups

Splunk Documentation on Lookups:

https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Aboutlookupsandfieldactions

Question: 89

What are the default time and results limits for a subsearch?

A. 60 seconds and 10,000 results
B. 60 seconds and 50,000 results
C. 300 seconds and 10,000 results
D. 300 seconds and 50,000 results

Answer: A

Explanation:

Comprehensive and Detailed Step by Step

The default time and results limits for a subsearch in Splunk are:

Time Limit : 60 seconds

Results Limit : 10,000 results

Here's why this works:

Time Limit : Subsearches are designed to execute quickly to avoid performance bottlenecks. By default, Splunk imposes a timeout of 60 seconds for subsearches. If the subsearch exceeds this limit, it will terminate, and the outer search may fail.

Results Limit : Subsearches are also limited to returning a maximum of 10,000 results by default. This ensures that the outer search does not get overwhelmed with too much data from the subsearch.

Other options explained:

Option B : Incorrect because the results limit is 10,000, not 50,000.

Option C : Incorrect because the time limit is 60 seconds, not 300 seconds.

Option D : Incorrect because both the time limit (300 seconds) and results limit (50,000) exceed the default values.

Example: If a subsearch exceeds the default limits, you might see an error like:

Copy

1

Error in 'search': Subsearch exceeded configured timeout or result limit.

Reference:

Splunk Documentation on Subsearch Limits:

https://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutsubsearches

Splunk Documentation on limits.conf:

https://docs.splunk.com/Documentation/Splunk/latest/Admin/Limitsconf

www.atmicnetworks.com

Question: 90

Which of the following is true about nested macros?

A. The inner macro should be created first.
B. The outer macro should be created first.
C. The outer macro name must be surrounded by backticks.
D. The inner macro passes arguments to the outer macro.

Answer: A

Explanation:
Comprehensive and Detailed Step by Step
When working with nested macros in Splunk, the inner macro should be created first . This ensures that the outer macro can reference and use the inner macro correctly during execution.
Here's why this works:
Macro Execution Order : Macros are processed in a hierarchical manner. The inner macro is executed first, and its output is then passed to the outer macro for further processing.
Dependency Management : If the inner macro does not exist when the outer macro is defined, Splunk will throw an error because the outer macro cannot resolve the inner macro's definition. Other options explained:
Option B : Incorrect because the outer macro depends on the inner macro, so the inner macro must be created first.
Option C : Incorrect because macro names are referenced using dollar signs ($macro_name$), not backticks. Backticks are used for inline searches or commands.
Option D : Incorrect because arguments are passed to the inner macro, not the other way around.
The inner macro processes the arguments and returns results to the outer macro.
Example:

```
# Define the inner macro
[inner_macro(1)]
args = arg1
definition = eval result = $arg1$ * 2

# Define the outer macro
[outer_macro(1)]
args = arg1
definition = `inner_macro($arg1$)`
```

In this example, inner_macro must be defined before outer_macro.
Reference:
Splunk Documentation on Macros:
https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Definesearchmacros
Splunk Documentation on Nested Macros:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Usesearchmacros

Question: 91

Which of the following is true about the multikv command?

A. The multikv command derives field names from the last column in a table-formatted event.
B. The multikv command creates an event for each column in a table-formatted event.
C. The multikv command requires field names to be ALL CAPS when multitable=false.
D. The multikv command displays an event for each row in a table-formatted event.

Answer: D

Explanation:
Comprehensive and Detailed Step by Step
The multikv command in Splunk is used to extract fields from table-like events (e.g., logs with rows and columns). It creates a separate event for each row in the table, making it easier to analyze structured data.
Here's why this works:
Purpose of multikv : The multikv command parses table-formatted events and treats each row as an individual event. This allows you to work with structured data as if it were regular Splunk events. Field Extraction : By default, multikv extracts field names from the header row of the table and assigns them to the corresponding values in each row.
Row-Based Events : Each row in the table becomes a separate event, enabling you to search and filter based on the extracted fields.
Example: Suppose you have a log with the following structure:

Name Age Location
Alice 30 New York
Bob 25 Los Angeles

Using the multikv command:

| multikv

This will create two events:

Event 1: Name=Alice, Age=30, Location=New York
Event 2: Name=Bob, Age=25, Location=Los Angeles

Other options explained:
Option A : Incorrect because multikv derives field names from the header row, not the last column.
Option B : Incorrect because multikv creates events for rows, not columns.
Option C : Incorrect because multikv does not require field names to be in ALL CAPS, regardless of the multitable setting.
Reference:
Splunk Documentation on multikv:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Multikv
Splunk Documentation on Parsing Structured Data:
https://docs.splunk.com/Documentation/Splunk/latest/Data/Extractfieldsfromstructureddata

www.atmicnetworks.com

Question: 92

Which of the following could be used to build a contextual drilldown?

A. <set> and <unset> elements with a depend? attribute.
B. $earliest$ and $latest$ tokens set by a global time range picker.
C. <set> and <reset> elements with a rejects attribute.
D. <set> and <offset> elements with depends and rejects attributes.

Answer: A

Explanation:
Comprehensive and Detailed Step by Step
To build a contextual drilldown in Splunk dashboards, you can use <set> and <unset> elements with a depend? attribute. These elements allow you to dynamically update tokens based on user interactions, enabling context-sensitive behavior in your dashboard.

Here's why this works:

Contextual Drilldown : A contextual drilldown allows users to click on a visualization (e.g., a chart or table) and navigate to another view or filter data based on the clicked value.
Dynamic Tokens : The <set> element sets a token to a specific value when a condition is met, while <unset> clears the token when the condition is no longer valid. The depend? attribute ensures that the behavior is conditional and context-aware.
Example:

```
<drilldown>
<set token="selected_product">$click.value$</set>
<unset token="selected_product" depend="?"></unset>
</drilldown>
```

In this example:
When a user clicks on a value, the selected_product token is set to the clicked value ($click.value$).
If the condition specified in depend? is no longer true, the token is cleared using <unset>.
Other options explained:

Option B : Incorrect because $earliest$ and $latest$ tokens are related to time range pickers, not contextual drilldowns.
Option C : Incorrect because <reset> is not a valid element in Splunk XML, and rejects is unrelated to drilldown behavior.
Option D : Incorrect because <offset> is not used for building drilldowns, and depends/rejects do not apply in this context.
Reference:
Splunk Documentation on Drilldowns:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/DrilldownIntro
Splunk Documentation on Tokens:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/UseTokenstoBuildDynamicInputs

Question: 93

Which of the following are predefined tokens?

A. $earliest_tok$ and now
B. ?click.field? and ?click.value?
C. ?earliest_tok$ and ?latest_tok?
D. ?click.name? and ?click.value?

Answer: A

Explanation:
Comprehensive and Detailed Step by Step
The predefined tokens in Splunk include $earliest_tok$ and now. These tokens are automatically available for use in searches, dashboards, and alerts.
Here's why this works:
Predefined Tokens :
$earliest_tok$: Represents the earliest time in a search's time range.
now: Represents the current time when the search is executed.
These tokens are commonly used to dynamically reference time ranges or timestamps in Splunk queries.
Dynamic Behavior : Predefined tokens like $earliest_tok$ and now are automatically populated by Splunk based on the context of the search or dashboard.
Other options explained:
Option B : Incorrect because ?click.field? and ?click.value? are not predefined tokens; they are contextual drilldown tokens that depend on user interaction.
Option C : Incorrect because ?earliest_tok$ and ?latest_tok? mix invalid syntax (? and $) and are not predefined tokens.
Option D : Incorrect because ?click.name? and ?click.value? are contextual drilldown tokens, not predefined tokens.
Reference:
Splunk Documentation on Tokens:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/UseTokenstoBuildDynamicInputs
Splunk Documentation on Time Tokens:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Specifytimemodifiersinyoursearch

Question: 94

When using the bin command, what attributes are used to define the size and number of sets created?

A. bins and start and end
B. bins and minspan
C. bins and span
D. bins and limit

Answer: C

Explanation:
Comprehensive and Detailed Step by Step
The bin command in Splunk is used to group numeric or time-based data into discrete intervals
(bins). The attributes used to define the size and number of sets are bins and span.

Here's why this works:

bins Attribute : Specifies the number of bins (intervals) to create. For example, bins=10 divides the data into 10 equal-sized intervals.

span Attribute : Specifies the size of each bin. For example, span=10 creates bins of size 10 for numeric data or span=1h creates bins of 1-hour intervals for time-based data.

Combination : You can use either bins or span to control the binning process, but not both simultaneously. If you specify both, span takes precedence.

Other options explained:

Option A : Incorrect because start and end are not attributes of the bin command; they are unrelated to defining bin size or count.

Option B : Incorrect because minspan is not a valid attribute of the bin command.

Option D : Incorrect because limit is unrelated to the bin command; it is typically used in other commands like stats or top.

Example:

```
index=_internal
| bin _time span=1h
```

This groups events into 1-hour intervals based on the _time field.

Reference:

Splunk Documentation on bin:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/bin

Splunk Documentation on Time-Based Binning:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Chartbinneddata

Question: 95

When enabled, what drilldown action is performed when a visualization is clicked in a dashboard?

A. A visualization is opened in a new window.
B. Search results are refreshed for the selected visualization.
C. Search results are refreshed for all panels in a dashboard.
D. A search is opened in a new window.

Answer: B

Explanation:

Comprehensive and Detailed Step by Step

When drilldown is enabled in a Splunk dashboard, clicking on a visualization triggers a refresh of the search results for the selected visualization . This allows users to interact with the data and refine the displayed results based on the clicked value.

Here's why this works:

Drilldown Behavior : Drilldown actions are configured to dynamically update tokens or filters based on user interactions. When a user clicks on a chart, table, or other visualization, the underlying search query is updated to reflect the selected value.

Contextual Updates : The refresh applies only to the selected visualization, ensuring that other panels in the dashboard remain unaffected unless explicitly configured otherwise.

Other options explained:

Option A : Incorrect because visualizations are not automatically opened in a new window during drilldown.

Option C : Incorrect because drilldown actions typically affect only the selected visualization, not all panels in the dashboard.

Option D : Incorrect because a new search window is not opened unless explicitly configured in the drilldown settings.

Example:

```
<drilldown>
<set token="selected_value">$click.value$</set>
</drilldown>
```

In this example, clicking on a value updates the selected_value token, which can be used to filter the visualization's search results.

Reference:

Splunk Documentation on Drilldowns:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/DrilldownIntro

Splunk Documentation on Tokens:
https://docs.splunk.com/Documentation/Splunk/latest/Viz/UseTokenstoBuildDynamicInputs

Question: 96

Which of the following is true about the preview feature and macros?

A. The preview feature expands only the selected macro within the search.
B. The preview feature can be launched using Tab-Shift-E on Mac or Windows.
C. The preview feature can be launched by right-clicking on the macro name in the search string.
D. The preview feature expands all macros within the search, including nested macros.

Answer: D

Explanation:

Comprehensive and Detailed Step by Step

The preview feature in Splunk expands all macros within a search, including any nested macros , to show their full definitions. This allows users to review the complete structure of the search query after all macros have been resolved.

Here's why this works:

Macro Expansion : Macros are placeholders for reusable search logic. When the preview feature is used, Splunk replaces all macro references with their corresponding definitions, including those nested within other macros.

Full Visibility : Expanding all macros ensures that users can see the entire search logic, which is especially helpful for debugging or understanding complex queries.

Other options explained:

Option A : Incorrect because the preview feature expands all macros, not just the selected one.

Option B : Incorrect because the keyboard shortcut Tab-Shift-E is not valid for launching the preview feature.

Option C : Incorrect because right-clicking on a macro name does not launch the preview feature; it is typically accessed through the Splunk UI or specific commands.

Reference:

Splunk Documentation on Macros:
https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Definesearchmacros

Splunk Documentation on Search Preview:
https://docs.splunk.com/Documentation/Splunk/latest/Search/Previewsearches

www.atmicnetworks.com

Question: 97

When should summary indexing be used?

A. For reports that run on small datasets over long time ranges.
B. For reports that do not qualify for report or data model acceleration.
C. For reports that run over short time ranges.
D. For reports that run in Smart Mode.

Answer: A

Explanation:
Comprehensive and Detailed Step by Step
Summary indexing should be used for reports that run on small datasets over long time ranges . It is particularly useful when you need to aggregate data over extended periods without querying raw events repeatedly.
Here's why this works:
Efficiency : Summary indexing pre-aggregates data into summary indexes, reducing the amount of data that needs to be processed during runtime. This improves performance for reports that span long time ranges.
Small Datasets : Summary indexing is most effective when working with smaller datasets because aggregating large volumes of data can become resource-intensive.
Other options explained:
Option B : Incorrect because summary indexing is not a fallback for reports that fail to qualify for acceleration methods like report or data model acceleration.
Option C : Incorrect because summary indexing is less beneficial for short time ranges, where querying raw data is often faster.
Option D : Incorrect because Smart Mode is unrelated to summary indexing; it is a search optimization feature.
Example: Suppose you want to calculate daily sales totals over a year. Instead of querying raw sales data every time, you can use summary indexing to store daily totals and query the summary index instead.
Reference:
Splunk Documentation on Summary Indexing:
https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Usesummaryindexing
Splunk Documentation on Report Acceleration:
https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Acceleratedatamodels

Question: 98

Which of the following is true about the summariesonly=t argument of the tstats command?
A. Applies only to accelerated data models.
B. When using an unaccelerated data model, the search produces a larger result count than with summariesonly=f.
C. Applies only to unaccelerated data models.
D. When using an accelerated data model, the search produces a larger result count than with summariesonly=f.

Answer: A

Explanation:

Comprehensive and Detailed Step by Step

The summariesonly=t argument of the tstats command applies only to accelerated data models . It ensures that the search uses only the precomputed summaries of the data model, ignoring raw data. Here's why this works:

Purpose of summariesonly=t : When set to true, the tstats command restricts the search to use only the accelerated summaries of the data model. This improves performance but may exclude events that are not part of the summary.

Accelerated Data Models : Acceleration creates summaries of data models, making them faster to query. Using summariesonly=t ensures that only these summaries are queried, avoiding raw data entirely.

Other options explained:

Option B : Incorrect because summariesonly=t does not apply to unaccelerated data models; it requires acceleration to function.

Option C : Incorrect because summariesonly=t applies only to accelerated data models, not unaccelerated ones.

Option D : Incorrect because summariesonly=t typically produces fewer results, as it excludes raw data that is not part of the summary.

Example:

| tstats count WHERE index=_internal summariesonly=t BY sourcetype

This query uses only the accelerated summaries of the _internal index.

Reference:

Splunk Documentation on tstats:
https://docs.splunk.com/Documentation/Splunk/latest/SearchReference/tstats

Splunk Documentation on Data Model Acceleration:
https://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Acceleratedatamodels

Question: 99

Which Job Inspector component displays the time taken to process field extractions?

A. command.search.filter
B. command.search.fields
C. command.search.kv
D. command.search.regex

Answer: C

Explanation:

The Splunk Job Inspector provides detailed metrics about the execution of search jobs, including the time taken by various components. The component responsible for measuring the time taken to apply field extractions is command.search.kv.

According to Splunk Documentation:

command.search.kv – tells how long it took to apply field extractions to the events.

This component specifically measures the duration of key-value field extraction processes during a search job.

Reference: View search job properties - Splunk Documentation

www.atmicnetworks.com

Question: 100

What happens when a bucket's bloom filter predicts a match?

A. Event data is read from journal.gz using the .tsidx files from that bucket.
B. Field extractions are used to filter through the .tsidx files from that bucket.
C. The filter is deleted from the indexer and wiped from memory.
D. Event data is read from the .tsidx files using the postings from that bucket.

Answer: A

Explanation:

In Splunk, a bloom filter is a probabilistic data structure used to quickly determine whether a given term or value might exist in a dataset, such as an index bucket. When a bloom filter predicts a match, it indicates that the term may be present, prompting Splunk to perform a more detailed check.
Specifically, when a bloom filter predicts a match:
Event data is read from journal.gz using the .tsidx files from that bucket.
This means that Splunk proceeds to read the raw event data stored in the journal.gz files, guided by the index information in the .tsidx files, to confirm the presence of the term.
Reference: Built-in optimization - Splunk Documentation

Question: 101

When a user opens a dataset in Pivot that has not been accelerated, an ad hoc data model acceleration is created. How long does this accelerated data model last?

A. For the time specified by a Splunk administrator in limits.conf
B. For the duration of the user's Pivot session
C. For 24 hours after Pivot was opened
D. For 7 days after Pivot was opened

Answer: B

Explanation:

In Splunk, when a user accesses a dataset in Pivot that lacks persistent acceleration, Splunk automatically creates an ad hoc data model acceleration. This temporary acceleration is designed to enhance performance during the user's current session.
According to Splunk Documentation:
"Ad hoc summaries are always created in a dispatch directory at the search head."
"These summaries are temporary and exist only for the duration of the user's Pivot session."
This means that the accelerated data model persists only while the user is actively engaged in the Pivot session. Once the session ends, the ad hoc acceleration is discarded.
Reference: Accelerate data models - Splunk Documentation

www.atmicnetworks.com

Question: 102

Where can wildcards be used in the tstats command?

A. In the where clause
B. In the by clause
C. In the from clause
D. No wildcards can be used with tstats

Answer: A

Explanation:

The tstats command in Splunk is optimized for performance and has specific limitations regarding the use of wildcards.
According to Splunk Documentation:
"The tstats command does not support wildcard characters in field values in aggregate functions or BY clauses."
"You can use wildcards in the where clause to filter results."
This means that while wildcards are not permitted in the by or from clauses, they can be effectively used within the where clause to filter data based on pattern matching.
Reference: tstats - Splunk Documentation

Question: 103

What are the results from the transaction command when keepevicted=true?

A. All closed transaction values are set to 0
B. The search results include data from failed transactions
C. All closed values are set to 1
D. Only failed transactions are kept in the data

Answer: B

Explanation:

The keepevicted parameter in the transaction command controls whether evicted transactions are included in the search results.
Evicted transactions are those that were not completed within specified constraints like maxspan, maxpause, or maxevents.
According to Splunk Documentation:
"keepevicted: Whether to output evicted transactions. Evicted transactions can be distinguished from non-evicted transactions by checking the value of the 'closed_txn' field."
"The 'closed_txn' field is set to '0' for evicted transactions and '1' for closed transactions."
By setting keepevicted=true, you ensure that these incomplete or failed transactions are included in your search results, allowing for comprehensive analysis.
Reference: transaction - Splunk Documentation

www.atmicnetworks.com

Question: 104

Which of the following is true when comparing the rex and erex commands?

A. The rex command is similar to automatic field extraction while erex isn't
B. The erex command uses data samples to generate regular expressions while rex doesn't
C. The rex command requires knowledge of regular expressions while erex doesn't
D. The erex command requires knowledge of regular expressions while rex doesn't

Answer: C

Explanation:

The rex and erex commands in Splunk are both used for field extraction, but they differ in their approach and requirements.
According to Splunk Documentation:
"rex: Specify a Perl regular expression named groups to extract fields while you search."
"erex: Use the erex command to extract data from a field when you do not know the regular expression to use. The command automatically extracts field values that are similar to the example values you specify."
This indicates that:
The rex command requires users to have knowledge of regular expressions to define the extraction patterns.
The erex command is designed for users who may not be familiar with regular expressions, allowing them to provide example values, and Splunk generates the appropriate regular expression.
Reference: erex - Splunk Documentation

Question: 105

What is the function of the |s token filter?

A. |s is not a valid token filter.
B. To wrap a value in double quotes.
C. To force no encoding to occur.
D. To encode URL values.

Answer: B

Explanation:

In Splunk's Simple XML dashboards, token filters modify how token values are rendered. The |s token filter specifically wraps the token value in double quotes and escapes any internal quotation marks.
This is particularly useful when constructing search strings that require quoted values.
For example, using $token_name|s$ ensures that the value of token_name is enclosed in double quotes, which is essential when the value contains spaces or special characters.
Reference: Token usage in dashboards - Splunk Documentation

Question: 106

Which of the following attributes only applies to the form element, and not the dashboard root element of a SimpleXML dashboard?

A. hideEdit
B. hideTitle
C. hideFilters
D. hideChrome

Answer: C

Explanation:

In Splunk's Simple XML, certain attributes are specific to the <form> element and do not apply to the <dashboard> root element.

The hideFilters attribute is one such attribute that is exclusive to the <form> element. It controls the visibility of form input elements (filters) in the dashboard.

Setting hideFilters="true" within the <form> element hides the input fields, allowing for a cleaner dashboard view when inputs are not necessary.

Reference: Simple XML Reference - Splunk Documentation

Question: 107

When using the bin command, what attributes are used to define the size and number of sets?

A. bins and minspan
B. bins and span
C. bins and start and end
D. bins and limit

Answer: B

Explanation:

The bin command in Splunk is used to group continuous numerical values into discrete buckets or bins. The span attribute defines the size of each bin, while the bins attribute specifies the number of bins to create.

For example:

spl

Copy

... | bin span=10ms bins=5 duration

This command creates 5 bins, each spanning 10 milliseconds, for the duration field.

Reference: bin - Splunk Documentation

Question: 108

Which of the following correctly uses mvfilter?

A. mvfilter(isnotnull(X))
B. mvfilter(x, isnotnull)
C. where mvfilter(isnotnull(X))
D. eval new_field=mvfilter(*)

www.atmicnetworks.com

Answer: A

Explanation:

The mvfilter function in Splunk is used to filter the values of a multivalue field based on a Boolean expression. The correct syntax is:

mvfilter(expression)

Where expression is a condition applied to each value in the multivalue field. For instance:

eval filtered_field = mvfilter(isnotnull(X))

This command filters out null values from the multivalue field X.

Reference: mvfilter - Splunk Documentation

Question: 109

How can an underlying search be optimized to improve dashboard performance?

A. Limit the results to a specific time window.
B. Convert the search to an inline search.
C. Use NOT expressions to filter results.
D. Use the transaction command instead of stats.

Answer: A

Explanation:

One of the most effective ways to enhance dashboard performance in Splunk is by narrowing the time range of the underlying searches. Limiting the search to a specific time window reduces the amount of data Splunk needs to process, leading to faster search execution and improved dashboard responsiveness.

According to Splunk Documentation:

"One of the most effective ways to limit the data that is pulled off from disk is to limit the time range. Use the time range picker or specify time modifiers in your search to identify the smallest window of time necessary for your search."

Reference: Quick tips for optimization - Splunk Documentation

Question: 110

What is one way to troubleshoot dashboards?

A. Create an HTML panel using tokens to verify that they are set.
B. Run the | previous_searches command to your SPL queries.
C. Go to the Troubleshooting dashboard of the Searching and Reporting app.
D. Delete the dashboard and start over.

Answer: A

Explanation:

When troubleshooting dashboards in Splunk, it's essential to verify that tokens are being set and passed correctly, especially when using dynamic inputs. Creating an HTML panel that displays token values can help confirm that tokens are populated as expected.

For example, you can add a panel with the following Simple XML to display token values: xml

Copy

```
<panel>
<html>
  <p>Token value: $your_token$</p>

</html>
</panel>
```

This approach allows you to see the current value of your_token directly on the dashboard, aiding in debugging issues related to token usage.

Reference: Master Splunk Dashboards: Expert Guide to Troubleshooting Tokens!

Question: 111

Which of the following is a valid event action in Splunk?

A. Execute an eval statement.
B. Edit an event in the raw data.
C. Execute a stats statement.
D. Create a new REST API endpoint.

Answer: A

Explanation:

In Splunk, event actions are operations that can be performed on events within the Search & Reporting app. One valid event action is executing an eval statement, which allows users to compute and add new fields to events dynamically.

According to Splunk Documentation:

"You can define workflow actions that perform tasks such as running a search, opening a URL, or executing an eval expression."

Reference: Control workflow action appearance in field and event menus - Splunk Documentation

Question: 112

When should the fill_summary_index.py script be used?

A. To create a summary index.
B. To backfill gaps in a summary index.
C. To reset a summary index that includes overlapping data.
D. To populate a summary index from a saved report.

Answer: B

Explanation:

www.atmicnetworks.com

The fill_summary_index.py script is a utility provided by Splunk to backfill data into a summary index. It's particularly useful when there are gaps in the summary index due to missed scheduled searches or when initializing a summary index with historical data.

According to Splunk Documentation:

"You can use the fill_summary_index.py script, which backfills gaps in summary index collection by

running the saved searches that populate the summary index as they would have been executed at their regularly scheduled times for a given time range."

Reference: Manage summary index gaps - Splunk Documentation

Question: 113

Consider the following search:

```
(index=_internal log group=tcpin connections) earliest
| stats count as _count by sourceHost guid fwdType version
| eventstats dc(sourceHost) as dc_sourceHost by guid
| where dc_sourceHost > 1
| fields - dc_sourceHost
| xyseries guid fwdType sourceHost
| search guid="00507345-CE09-4A5E-428-D3E8718CB065"
| appendpipe [ stats count | eval "Duplicate GUID" = if(count==0, "Yes", "No") ]
```

Which of the following are transforming commands?

A. where and search
B. fields and appendpipe
C. stats and xyseries
D. eval and eventstats

Answer: C

Explanation:

In Splunk, transforming commands are those that process events to produce statistical summaries, often changing the shape of the data. Among the commands listed:

stats is a transforming command that computes aggregate statistics, such as count, sum, average, etc., and transforms the data into a tabular format.

xyseries is also a transforming command that reshapes the data into a matrix format suitable for charting, converting three columns into a two-dimensional table.

The other commands:

where and search are filtering commands.

fields is a field selector command.

appendpipe is a generating command.

www.atmicnetworks.com

eval is an evaluation command.
eventstats is a reporting command that adds summary statistics to each event.
Reference:
stats - Splunk Documentation
xyseries - Splunk Documentation

Question: 114

What is the value of base lispy in the Search Job Inspector for the search index=sales clientip=170.192.178.10?

A. [index::sales 192 AND 10 AND 178 AND 170]
B. [index::sales AND 469 10 702 390]
C. [192 AND 10 AND 178 AND 170 index::sales]
D. [AND 10 170 178 192 index::sales]

Answer: A

Explanation:

In Splunk, the "base lispy" is an internal representation of the search query used by the Search Job Inspector. It breaks down the search into its fundamental components for processing. For the search index=sales clientip=170.192.178.10, Splunk tokenizes the IP address into its individual octets and combines them with the index specification.
Therefore, the base lispy representation would be:
[index::sales 192 AND 10 AND 178 AND 170]
This indicates that the search is constrained to the sales index and is looking for events containing all the specified IP address components.

Question: 115

When working with an accelerated data model acc_datmodel and an unaccelerated data model unacc_datmodel, what tstats query could be used to search one of these data models?

A. | tstats count from datamodel=acc_datmodel summariesonly=false
B. | tstats count where datamodel=acc_datmodel summariesonly=false
C. | tstats count where index=datamodel by index, datamodel
D. | tstats count from datamodel=unacc_datmodel summariesonly=true

Answer: A

Explanation:

The tstats command in Splunk is optimized for performance and is typically used with accelerated data models. The summariesonly parameter determines whether the search should use only the summarized (accelerated) data or fall back to raw data if necessary.
Setting summariesonly=false allows the search to use both summarized and raw data, making it suitable for both accelerated and unaccelerated data models.
Setting summariesonly=true restricts the search to only summarized data, which would result in no data returned if the data

www.atmicnetworks.com

model is not accelerated.

Therefore, to search an accelerated data model and allow fallback to raw data if needed, the correct query is:

| tstats count from datamodel=acc_datmodel summariesonly=false Reference:

tstats - Splunk Documentation

Question: 116

What does it mean when a command is run and the is_exact column is 0?

A. The distinct count of values for that field is exactly 0.
B. The distinct count of fields in the field summary is 1.
C. The distinct count of values in that field is approximated.
D. The distinct count of values for that field is exact.

Answer: C

Explanation:

In Splunk, the is_exact field indicates whether the count of distinct values for a particular field is exact or estimated. A value of:

1 means the count is exact.

0 means the count is an approximation.

Therefore, when is_exact is 0, it signifies that the distinct count of values for that field is an estimate, not an exact count.

Reference:

fields - Splunk Documentation

Question: 117

How is a multivalue field created from product="a, b, c, d"

A .. | mvexpand product
B .. | eval mvexpand(makemv(product, ","))
C .. | makemv delim="," product
D . | makemv delim(product)

Answer: C

Explanation:

To create a multivalue field from a single string with comma-separated values, the makemv command is used with the delim parameter to specify the delimiter.

The correct syntax is:

... | makemv delim="," product

This command splits the product field into multiple values wherever a comma is found, effectively creating a multivalue field.

www.atmicnetworks.com

Reference:
makemv - Splunk Documentation

Question: 118

Which SPL command converts the hour into a user's local time based upon the user's time zone preference setting?

A. time(_time, "%H")
B. local_time(_time, "%H")
C. relative_time(_time, "%H")
D. strftime(_time, "%H")

Answer: D

Explanation:

The strftime function in Splunk is used to format timestamps into human-readable strings. When you use strftime(_time, "%H"), it converts the _time field into the hour (00 to 23) based on the user's time zone preference setting.
Splunk stores all timestamps in Coordinated Universal Time (UTC). However, when displaying time, it adjusts according to the user's time zone preference set in their profile. Therefore, using strftime will reflect the local time for the user.
Reference: Splunk Community Discussion on Time Zone Conversion

Question: 119

Which of the following will best optimize dashboard performance?

A. Use inline searches.
B. Use base searches.
C. Use accelerated data models.
D. Use scheduled reports.

Answer: C

Explanation:

Accelerated data models in Splunk create summaries of data that can be queried more efficiently, significantly improving dashboard performance. By precomputing and storing results, dashboards can retrieve data faster, reducing load times and resource consumption.
According to Splunk Documentation:
"Data model acceleration speeds up reporting for the entire set of fields that you define in a data model and which you and your Pivot users want to report on."

Reference: Accelerate Data Models - Splunk Documentation

www.atmicnetworks.com

Question: 120

Which of the following is true about Log Event alerts?

A. They must be used with other alert actions.
B. They cannot use tokens to reference event fields.
C. They require at least Power User role.
D. They create new searchable events.

Answer: D

Explanation:

Log Event alerts in Splunk are designed to create new events in the index when specific conditions are met. These events are then searchable like any other event, allowing for further analysis and correlation.

This functionality is particularly useful for tracking occurrences of specific conditions over time or triggering additional workflows based on the logged events.

Reference: Splunk Documentation on Alert Actions