PassQueen
a.k.a dumpsqueen.com

# Splunk Core Certified Consultant

## Splunk SPLK-3003

**Version Demo**

**Total Demo Questions: 12**

**Total Premium Questions: 120**

**Buy Premium PDF**

https://passqueen.com

support@passqueen.com

# Topic Break Down

| Topic | No. of Questions |
|---|---|
| Topic 1, Deployment Planning | 14 |
| Topic 2, Deployment Implementation | 23 |
| Topic 3, Data Collection and Management | 30 |
| Topic 4, Splunk Configuration | 34 |
| Topic 5, Troubleshooting and Optimization | 19 |
| Total | 120 |

QUESTION NO: 1

Which statement is true about sub searches?

**A.** Sub searches are faster than other types of searches.

**B.** Sub searches work best for joining two large result sets.

**C.** Sub searches run at the same time as their outer search.

**D.** Sub searches work best for small result sets.

**ANSWER: D**

**Explanation:**

Sub searches work best for small result sets. Splunk executes a subsearch first, then formats its results and passes them to the outer search as search criteria or as input to the command containing the subsearch. This makes subsearches useful when a preliminary search can identify a limited set of values—such as a small list of hosts, users, session IDs, or other field values—that the main search should use. Because the subsearch must complete and return its results before the outer search can use them, its output is constrained by configurable limits, including a default maximum of 10,000 results and a default execution time limit of 60 seconds. Large result sets can therefore be truncated or cause the subsearch to time out, potentially producing incomplete outer-search results. Designing a subsearch to return only the fields and values needed by the outer search, and reducing its result set early with focused filters and transforming commands where appropriate, helps keep searches reliable and efficient. Splunk recommends considering alternatives such as lookups, summary indexing, or other search designs when the intermediate data set is large. See Splunk's About subsearches documentation and its join command reference for subsearch behavior and limits.

**QUESTION NO: 2**

A customer requires an administrator to determine why a setting in props.conf is taking effect differently than expected on an indexer.

Which command shows the effective value and the configuration file that supplied it?

**A.** $SPLUNK_HOME/bin/splunk btool props list --debug

**B.** $SPLUNK_HOME/bin/splunk list props --effective

**C.** $SPLUNK_HOME/bin/splunk show config props.conf

**D.** $SPLUNK_HOME/bin/splunk validate props --source

**ANSWER: A**

**Explanation:**

The command `splunk btool props list --debug` displays the effective merged props.conf configuration and identifies the source file for each setting. The `--debug` option is particularly useful because it shows the path of the configuration file that provides each effective value.

Splunk configuration is layered across system and app directories, with local settings generally taking precedence over default settings. Btool resolves that precedence and is therefore more reliable than inspecting a single configuration file manually. An administrator can add a stanza name to narrow the output to a particular sourcetype. See Splunk documentation on using btool to troubleshoot configurations and how configuration files work.

**QUESTION NO: 3**

A site from a multi-site indexer cluster needs to be decommissioned. Which of the following actions must be taken?

**A.** Nothing. Decommissioning a site is not possible.

**B.** Create an alias for where the new data should be sent.

**C.** Remove the site from the list of available sites.

**D.** Remove the site from the list of available sites and create an alias for where the new data should be sent.

**ANSWER: D**

**Explanation:**

Remove the site from the list of available sites and create an alias for where the new data should be sent. In a multisite indexer cluster, a site can be referenced by data-routing and site-affinity configurations. Before retiring that site, administrators must ensure that traffic intended for it is redirected to a remaining site. A site alias provides that continuity: components that still refer to the retiring site can resolve the alias and send new data to the designated surviving site instead. After routing has been redirected, removing the retired site from the available-site configuration prevents cluster components from treating it as an active destination for new indexing activity. This is a coordinated configuration change, not merely a matter of shutting down the indexers at the location. The cluster must continue to meet its multisite data-placement requirements, including replication and searchable-copy policies, while the retiring site's data is handled according to the decommissioning plan. Splunk's multisite-cluster documentation describes site-aware deployment and the role of site configuration in routing and data availability. See Multisite indexer cluster deployment and How multisite indexer clusters work.

## QUESTION NO: 4

A customer wants to understand how Splunk bucket types (hot, warm, cold) impact search performance within their environment. Their indexers have a single storage device for all data. What is the proper message to communicate to the customer?

**A.** The bucket types (hot, warm, or cold) have the same search performance characteristics within the customer's environment.

**B.** While hot, warm, and cold buckets have the same search performance characteristics within the customers environment, due to their optimized structure, the thawed buckets are the most performant.

**C.** Searching hot and warm buckets result in best performance because by default the cold buckets are miniaturized by removing TSIDX files to save on storage cost.

**D.** Because the cold buckets are written to a cheaper/slower storage volume, they will be slower to search compared to hot and warm buckets which are written to Solid State Disk (SSD).

**ANSWER: A**

**Explanation:**

The bucket types (hot, warm, or cold) have the same search performance characteristics within the customer's environment. In Splunk, hot, warm, and cold are lifecycle states that describe a bucket's age and writability rather than an inherent search-speed tier. Hot buckets are actively receiving data; warm buckets are no longer written to; and cold buckets are older buckets that Splunk has rolled from warm storage. A bucket's actual search performance is primarily determined by the storage media and I/O available at the location where the bucket resides, along with factors such as concurrent workload, index configuration, and the volume of data being searched.

Because this environment uses one storage device for all indexed data, each bucket type is being read from the same underlying storage characteristics. Therefore, merely transitioning a bucket between hot, warm, and cold does not make it faster or slower to search. Different performance behavior could arise only if the deployment were configured to place bucket states on storage with different latency or throughput characteristics. Splunk's index lifecycle documentation describes the bucket states and explains that buckets roll through them as data ages; storage placement is configurable through index volume and path settings. See How Splunk stores indexes and Configure index storage.

## QUESTION NO: 5

A customer uses a monitor input to ingest application export files. Files that are more than 30 days old should not be ingested when the input is first enabled.

Which inputs.conf setting should be used?

**A.** ignoreOlderThan

**B.** followTail

**C.** move_policy

**D.** crcSalt

**ANSWER: A**

**Explanation:**

`ignoreOlderThan` is correct because it specifies the age threshold for files that Splunk should ignore when it discovers files for a monitor input. Configuring the setting to 30 days prevents old files from being consumed when the input is initially enabled.

This setting is useful when a monitored directory contains historical files that should remain on disk but are outside the intended ingestion scope. It should be used carefully with file-delivery processes so that valid files are not excluded before the forwarder has an opportunity to read them. See the inputs.conf specification and file and directory monitoring documentation.

## QUESTION NO: 6

When utilizing a subsearch within a Splunk SPL search query, which of the following statements is accurate?

**A.** Subsearches have to be initiated with the | subsearch command.

**B.** Subsearches can only be utilized with | inputlookup command.

**C.** Subsearches have a default result output limit of 10000.

**D.** There are no specific limitations when using subsearches.

**ANSWER: C**

**Explanation:**

Subsearches have a default result output limit of 10000. A subsearch runs first, and Splunk formats its returned results so they can be used by the outer search. To prevent subsearches from consuming excessive resources or generating an impractically large expanded search expression, Splunk applies result and time limits. By default, the `maxout` setting for subsearches limits the number of results returned to the invoking search to 10,000. Therefore, even if a subsearch finds more matching events or rows, only up to that default number is available for use by the outer search unless an administrator changes the applicable limit configuration. This behavior is important when designing searches that use brackets, such as filtering a primary search with values produced by another search. Search authors should account for the result cap when completeness matters and should validate whether a more scalable SPL design is appropriate. Splunk documents the default subsearch result limit and its operational constraints in its subsearch documentation and in the `limits.conf` configuration reference. See Splunk: About subsearches and Splunk: limits.conf.

## QUESTION NO: 7

A customer has the following Splunk instances within their environment: An indexer cluster consisting of a cluster master/master node and five clustered indexers, two search heads (no search head clustering), a deployment server, and a license master. The deployment server and license master are running on their own single-purpose instances. The customer would like to start using the Monitoring Console (MC) to monitor the whole environment.

On the MC instance, which instances will need to be configured as distributed search peers by specifying them via the UI using the settings menu?

**A.** Just the cluster master/master node.

**B.** Indexers, search heads, deployment server, license master, cluster master/master node.

**C.** Search heads, deployment server, license master, cluster master/master node

**D.** Deployment server, license master

**ANSWER: C**

**Explanation:**

**Search heads, deployment server, license master, cluster master/master node** is correct. In a distributed Monitoring Console deployment, the Monitoring Console must be able to run searches against the Splunk Enterprise instances whose monitoring data it needs to collect. The two standalone search heads, the deployment server, and the license master are separate, non-clustered instances, so each must be explicitly supplied to the Monitoring Console as a distributed search peer through the Monitoring Console's settings workflow. The cluster master/master node must likewise be configured so that the Monitoring Console can obtain indexer-cluster topology and health information and use the cluster configuration to discover the clustered indexers. Once the indexer cluster is configured through its manager node, the Monitoring Console can automatically identify the indexer peers in that cluster; individual clustered indexers do not need to be manually entered through the settings menu. This configuration lets the Monitoring Console consolidate health, performance, licensing, deployment, search, and indexing visibility for the entire described deployment. Splunk documents the distributed Monitoring Console setup and the required peer configuration in its Configure a distributed deployment of the Monitoring Console guide. See also the Monitoring Console overview for its role in monitoring Splunk Enterprise deployments.

**QUESTION NO: 8**

What happens to the indexer cluster when the indexer Cluster Master (CM) runs out of disk space?

**A.** A warm standby CM needs to be brought online as soon as possible before an indexer has an outage.

**B.** The indexer cluster will continue to operate as long as no indexers fail.

**C.** If the indexer cluster has site failover configured in the CM, the second cluster master will take over.

**D.** The indexer cluster will continue to operate as long as a replacement CM is deployed within 24 hours.

**ANSWER: B**

**Explanation:**

The indexer cluster will continue to operate as long as no indexers fail. The Cluster Master, now termed the cluster manager in current Splunk documentation, coordinates peer-node membership, bucket fix-up, replication and search-factor maintenance, and cluster configuration. If it runs out of disk space and can no longer function correctly, the peer nodes do not immediately stop indexing or serving searches. They retain the bucket copies and cluster state already available to them, so normal cluster activity can continue during the manager outage.

However, the cluster cannot reliably respond to a peer-node failure while the Cluster Master is unavailable. Restoring the manager's disk capacity or replacing and recovering the manager should therefore be treated as an urgent operational task. Once the manager is functioning again, it can resume monitoring the cluster and coordinating any necessary remediation to maintain configured replication and search factors. Splunk documents this behavior as part of indexer-cluster manager failure handling: peer nodes continue their normal operations, but cluster maintenance activities requiring manager coordination are unavailable until the manager returns. See What happens when the cluster manager goes down and About indexer clusters.

**QUESTION NO: 9**

When adding a new search head to a search head cluster (SHC), which of the following scenarios occurs?

**A.** The new search head connects to the captain and replays any recent configuration changes to bring it up to date.

**B.** The new search head connects to the deployer and replays any recent configuration changes to bring it up to date.

**C.** The new search head connects to the captain and pulls the most recently deployed bundle. It then connects to the deployer and replays any recent configuration changes to bring it up to date.

**D.** The new search head connects to the deployer and pulls the most recently deployed bundle. It then connects to the captain and replays any recent configuration changes to bring it up to date.

**ANSWER: A**

**Explanation:**

The new search head connects to the captain and replays any recent configuration changes to bring it up to date. In a search head cluster, the captain coordinates cluster membership and replicates the cluster's replicated configuration and knowledge objects among members. When a new member is added, it joins the existing cluster and synchronizes with the captain so that it receives the current replicated state. The member can then participate consistently in scheduled-search execution, artifact replication, and other clustered search-head activities.

This synchronization is a core responsibility of the search head cluster's captain-based configuration replication process. The deployer has a different operational purpose: administrators use it to distribute apps and other non-replicated configuration content to search head cluster members. It is not the source a newly joining member uses to establish the cluster's replicated configuration state. After the member has joined and synchronized, administrators may use the deployer as appropriate to distribute applicable apps or deployment content consistently across the cluster. Splunk documents the joining process and the captain's role in maintaining cluster configuration in its Add search head cluster members guidance and How search head clustering works overview.

**QUESTION NO: 10**

What happens when an index cluster peer freezes a bucket?

**A.** All indexers with a copy of the bucket will delete it.

**B.** The cluster master will ensure another copy of the bucket is made on the other peers to meet the replication settings.

**C.** The cluster master will no longer perform fix-up activities for the bucket.

**D.** All indexers with a copy of the bucket will immediately roll it to frozen.

**ANSWER: C**

**Explanation:**

The cluster master will no longer perform fix-up activities for the bucket. Freezing is the final stage of an index bucket's normal retention lifecycle. In an indexer cluster, when a peer freezes its copy of a bucket, that copy is no longer considered part of the cluster's actively managed bucket set. Consequently, the cluster manager (called the cluster master in older Splunk terminology) stops attempting to satisfy replication-factor and search-factor requirements for that bucket through fix-up operations. It does not treat the loss of that frozen copy as a condition requiring creation or streaming of a replacement copy.

This behavior is important because bucket retention is governed by index settings, and frozen data has reached the end of the searchable lifecycle configured for that index. Peers can freeze their bucket copies independently, based on the bucket-aging process and their local bucket state. Once freezing begins, the cluster's normal high-availability maintenance mechanisms are intentionally no longer applied to maintain the bucket's configured number of copies. Splunk documents this cluster-specific behavior in its discussion of buckets and clustering: Buckets and clusters. For the broader bucket lifecycle, including the frozen state, see How Splunk Enterprise stores indexes.

## QUESTION NO: 11

What is the primary driver behind implementing indexer clustering in a customer's environment?

**A.** To improve resiliency as the search load increases.

**B.** To reduce indexing latency.

**C.** To scale out a Splunk environment to offer higher performance capability.

**D.** To provide higher availability for buckets of data.

## ANSWER: D

**Explanation:**

Implementing indexer clustering is primarily driven by the need to provide higher availability for buckets of data. In an indexer cluster, Splunk maintains multiple copies of each bucket across peer nodes according to the replication factor. This means that, if an individual peer becomes unavailable, the cluster can continue to retain and serve the data from replicated bucket copies on other peers. The cluster manager monitors peer status and coordinates bucket replication so that the configured data-availability requirements are maintained.

Search availability is also supported through the search factor, which determines how many searchable copies of a bucket are maintained. Search heads can therefore access suitable bucket copies even when a peer is down or undergoing maintenance. This architecture protects indexed data from a single indexer failure and enables continued access to that data, making resilient bucket storage the fundamental purpose of indexer clustering. Capacity and performance planning remain important parts of a deployment, but the defining architectural value is maintaining replicated, available data buckets. Splunk documents the relationship between replication factor, search factor, bucket copies, and peer-node availability in its indexer-cluster administration guidance. See About indexer clusters and How clustered search works.

## QUESTION NO: 12

A customer has enabled data model acceleration for a data model that is used by several high-volume security dashboards.

Which indexes.conf setting specifies the location where the accelerated data model summaries are stored?

**A.** summaryHomePath

**B.** tstatsHomePath

**C.** thawedPath

**D.** coldPath

## ANSWER: B

**Explanation:**

`tstatsHomePath` is correct because data model acceleration creates TSIDX summary files that are used by the `tstats` command to return results from accelerated data models. The setting defines the storage location for those accelerated data model summaries.

This path is separate from `summaryHomePath`, which is used for report acceleration and summary indexing data. Storage capacity and performance for `tstatsHomePath` should be considered when sizing an environment that uses data model acceleration. See the indexes.conf specification and accelerate data models documentation.