DUMPSBOSS.

Splunk Enterprise Certified Admin

Splunk SPLK-1003

Version Demo

Total Demo Questions: 28

Total Premium Questions: 287

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Splunk Deployment Overview	2
Topic 2, License Management	14
Topic 3, Splunk Apps	2
Topic 4, Splunk Configuration Files	45
Topic 5, User Management	32
Topic 6, Data Management	64
Topic 7, Search Management	3
Topic 8, Index Management	22
Topic 9, Forwarder Management	73
Topic 10, Distributed Search	27
Topic 11, Splunk Deployment Monitoring	3
Total	287

QUESTION NO: 1

A new forwarder has been installed with a manually createddeploymentclient.conf.

What is the next step to enable the communication between the forwarder and the deployment server?

A. Restart Splunk on the deployment server.

B. Enable the deployment client in Splunk Web under Forwarder Management.

C. Restart Splunk on the deployment client.

D. Wait for up to the time set in thephoneHomeIntervalInSecssetting.

ANSWER: C

Explanation:

Restart Splunk on the deployment client. A forwarder acts as a deployment client when its `deploymentclient.conf` configuration specifies a deployment server, typically through the `targetUri` setting in the `[target-broker:deploymentServer]` stanza. Because this configuration was created manually after the forwarder installation, the running Splunk process must be restarted to load the new file and initialize the deployment-client service with the configured deployment-server address. Once restarted, the forwarder can phone home to the deployment server and request any applicable server classes, apps, and configuration content. This is the normal activation step for manually adding or changing deployment-client configuration on a Splunk instance. Splunk documents that changes to deployment client configuration require restarting the deployment client for the changes to take effect. See the Splunk documentation for configuring deployment clients and the deploymentclient.conf specification for the required deployment-client settings and behavior.

QUESTION NO: 2

Which layers are involved in Splunk configuration file layering? (Choose all that apply.)

A. App context

B. User context

C. Global context

D. Forwarder context

ANSWER: A B C

Explanation:

Splunk configuration file layering uses global context, app context, and user context to determine the effective settings that Splunk Enterprise applies. Global context represents system-wide configuration, typically stored under the `system` directory, and establishes settings that can apply across the entire Splunk instance. App context provides configuration packaged with or local to a specific app, allowing an app to define knowledge objects and operational settings without necessarily affecting unrelated apps. User context supplies user-specific configuration and is used for private knowledge objects or preferences associated with an individual user within an app.

When the same setting is defined at multiple levels, Splunk resolves the setting through its documented configuration-file precedence rules. This layered model lets administrators establish broad defaults globally while allowing app-specific and, where appropriate, user-specific customization. Understanding these contexts is essential when troubleshooting unexpected configuration behavior, because the effective value may come from a higher-precedence configuration layer rather than the file initially being inspected. Splunk documents the configuration-directory structure and precedence order in its administration guidance: Configuration file directories and Where to find configuration files.

QUESTION NO: 3

The Splunk administrator wants to ensure data is distributed evenly amongst the indexers. To do this, he runs the following search over the last 24 hours:

index=*

What field can the administrator check to see the data distribution?

A. host

B. index

C. linecount

D. splunk_server

ANSWER: D

Explanation:

The `splunk_server` field identifies the Splunk Enterprise instance that supplied each event to the search. In a distributed deployment, this lets an administrator group or count returned events by the indexer that owns and searches the relevant data. For example, after running `index=*` over the selected time range, adding `| stats count by splunk_server` provides event counts per participating indexer. Comparable counts across the indexer names indicate that data is broadly distributed evenly, while a significant imbalance can identify an indexer holding or returning disproportionately more data for that time range.

Splunk documents `splunk_server` as a default field containing the name of the Splunk server from which an event originates in distributed search results. This makes it the appropriate field for assessing indexer-level distribution from the specified search. The field is especially useful for an operational validation of indexer balance because it reflects the server participation associated with the actual result set being examined. See Splunk's documentation for default fields and its reference for the stats command.

QUESTION NO: 4

Which of the methods listed below supports muti-factor authentication?

A. Lightweight Directory Access Protocol (LDAP)

B. Security Assertion Markup Language (SAML)

C. Single Sign-on (SSO)

D. OpenID

ANSWER: B

Explanation:

Security Assertion Markup Language (SAML) supports multi-factor authentication in Splunk Enterprise by allowing Splunk to delegate user authentication to an external SAML identity provider. The identity provider, rather than Splunk itself, can enforce the organization's authentication policy before issuing a signed SAML assertion to Splunk. That policy can require an additional factor such as a push approval, time-based one-time passcode, hardware security key, smart card, or biometric verification in addition to the user's password. After the user completes the configured factors successfully, the identity provider sends the assertion containing the authenticated user identity and relevant attributes; Splunk then uses that assertion to establish the user's session and apply role mappings. This makes SAML a practical integration point for enterprise MFA services while retaining centralized identity and access-policy management. Splunk documents that SAML single sign-on uses a third-party identity provider to authenticate users and pass SAML assertions to Splunk. See How SAML SSO works in Splunk Enterprise and Configure SAML SSO in Splunk Enterprise.

QUESTION NO: 5

Which feature in Splunk allows Event Breaking, Timestamp extractions, and any advanced configurations found in props.conf to be validated all through the UI?

A. Apps

B. Search

C. Data preview

D. Forwarder inputs

ANSWER: C

Explanation:

Data preview is correct because it provides Splunk Web's interactive interface for examining incoming sample data and validating how Splunk will process it before or while data is onboarded. In Data Preview, an administrator can select or create a source type and inspect the resulting events, including where Splunk breaks a data stream into individual events. It also displays timestamp recognition results, enabling the administrator to verify that event times are extracted as intended.

The feature supports configuration choices that correspond to source-type processing settings stored in `props.conf`, such as line breaking, event boundary handling, timestamp extraction, and other parsing behavior. This lets an administrator test and refine parsing settings visually rather than relying solely on configuration-file edits and subsequent indexing checks. The preview is especially useful because parsing decisions affect how events are indexed and therefore should be validated with representative sample data. Splunk documents the role of source types and `props.conf` in parsing data at index time, including event breaking and timestamp processing. See Configure timestamp recognition and props.conf specification.

QUESTION NO: 6

What is the **default purpose** of a **Splunk Deployment Server** ?

A. To stage and deploy updates to /etc/pcer-apps/

B. To stage and deploy updates to $SPLUNK_HOME/etc/apps/

C. To stage and deploy updates to /etc/manager-apps/

D. To stage and deploy updates to $SPLUNK_HOME/etc/deployment-apps/

ANSWER: D

Explanation:

To stage and deploy updates to `$SPLUNK_HOME/etc/deployment-apps/` is correct because a Splunk Deployment Server centrally distributes Splunk apps and configuration content to deployment clients. Administrators place deployment apps in this staging directory on the deployment server. Each deployment app is a directory containing the relevant configuration files, such as inputs, outputs, props, transforms, or other app content that must be consistently delivered to a group of Splunk instances.

The Deployment Server uses server classes to determine which clients receive each deployment app. Clients, including universal forwarders and other eligible Splunk Enterprise instances, phone home to the Deployment Server and receive applicable app updates. This lets an administrator manage distributed configuration changes from one location instead of manually copying files to every client. The staging location is specifically `$SPLUNK_HOME/etc/deployment-apps`; the `$SPLUNK_HOME` prefix identifies the Splunk installation directory and should be included when specifying the full default path. See Splunk's documentation on Deployment Server deployment and the serverclass.conf specification.

QUESTION NO: 7

A user recently installed an application to index NCINX access logs. After configuring the application, they realize that no data is being ingested. Which configuration file do they need to edit to ingest the access logs to ensure it remains unaffected after upgrade?

A. $SPLUNK_HOME/etc/apps/splunk_TA_nginx/local/inputs.conf

B. $SPLUNK_HOME/etc/apps/splunk_TA_nginx/default/inputs.conf

C. $SPLUNK_HOME/etc/system/local/inputs.conf

D. $SPLUNK_HOME/etc/apps/splunk_TA_nginx/local/props.conf

ANSWER: A

Explanation:

`$SPLUNK_HOME/etc/apps/splunk_TA_nginx/local/inputs.conf` is the appropriate file for defining or enabling the data input that monitors NGINX access logs while preserving the administrator's changes through upgrades. In a Splunk app or add-on, `inputs.conf` contains input stanzas, such as file and directory monitor inputs, that tell a Splunk instance what data to collect. A monitoring stanza for the access-log path must be enabled and configured on the Splunk component that performs the collection, typically a universal forwarder or heavy forwarder.

Splunk's configuration-directory structure intentionally separates vendor-supplied settings from site-specific settings. The app's `local` directory is the proper location for locally created or overridden configuration. Splunk gives local configuration precedence over the app's default configuration, and upgrade processes preserve the local directory rather than replacing it with the updated app package. Therefore, placing the required access-log input configuration in `splunk_TA_nginx/local/inputs.conf` both activates the intended collection configuration and protects that customization during a future Splunk or app upgrade.

See Splunk's documentation on configuration file locations and the default/local directories and the inputs.conf specification.

QUESTION NO: 8

Which authentication methods are natively supported within Splunk Enterprise? (select all that apply)

A. LDAP

B. SAML

C. RADIUS

D. Duo Multifactor Authentication

ANSWER: A B

Explanation:

LDAP and SAML are natively supported authentication methods in Splunk Enterprise. LDAP integration allows Splunk Enterprise to authenticate users against an organization's existing LDAP directory, including Microsoft Active Directory. Administrators configure one or more LDAP strategies in Splunk Web or configuration files, map directory groups to Splunk roles, and can use the directory as the central source for user authentication and authorization-related group membership.

SAML is also built into Splunk Enterprise as an authentication option. It enables single sign-on through a SAML identity provider, allowing users to authenticate with the organization's established identity service and then access Splunk Enterprise without maintaining a separate Splunk password. Splunk Enterprise can act as a SAML service provider and supports the configuration of SAML identity-provider settings, user attributes, and role mappings. These integrations are first-class Splunk Enterprise authentication configurations rather than requiring a custom authentication script. See Splunk's authentication overview and its documentation for SAML single sign-on.

QUESTION NO: 9

Where can scripts for scripted inputs reside on the host file system? (select all that apply)

A. $SFLUNK_HOME/bin/scripts

B. $SPLUNK_HOME/etc/apps/bin

C. $SPLUNK_HOME/etc/system/bin

D. $S?LUNK_HOME/etc/apps/ < your_app > /bin_

ANSWER: A C D

Explanation:

Scripts invoked by a scripted input must be located in one of Splunk's supported script directories on the Splunk instance that runs the input. The supported global locations are `$SPLUNK_HOME/etc/system/bin` and `$SPLUNK_HOME/bin/scripts`. An app can also supply a script in its own `bin` directory, using the path `$SPLUNK_HOME/etc/apps/<your_app>/bin`. These locations allow Splunk to locate and execute the command specified by the scripted-input stanza in `inputs.conf`.

For maintainability and app portability, Splunk recommends placing the script in the `bin` directory nearest to the `inputs.conf` file that invokes it. Thus, when an app defines the scripted input, storing the executable under that app's `bin` directory keeps the input definition and its supporting script together. The system-level and general scripts directories remain valid choices for scripts intended to be shared more broadly on the host. See Splunk's inputs.conf specification and its guidance on scripted inputs.

QUESTION NO: 10

Which setting in indexes.conf allows data retention to be controlled by time?

A. maxDaysToKeep

B. moveToFrozenAfter

C. maxDataRetentionTime

D. frozenTimePeriodInSecs

ANSWER: D

Explanation:

`frozenTimePeriodInSecs` is the `indexes.conf` setting that controls time-based retention. It specifies, in seconds, how long indexed data is retained before Splunk moves it from the warm or cold bucket state to the frozen state. A bucket becomes eligible for freezing when the age of its newest event exceeds the configured period. The outcome for frozen data depends on the index configuration: Splunk can delete it, archive it using a frozen-data script, or, where applicable, manage it through an external storage workflow. This setting therefore defines the retention period based on event time rather than a storage-capacity limit. Administrators should choose a value that meets their organization's retention policy and compliance requirements, while confirming that timestamps are correctly extracted because bucket aging is based on event timestamps. The default retention period is commonly six years, expressed as 188,697,600 seconds, unless a different value is configured for a specific index. See Splunk's indexes.conf specification and its guidance on setting a data retirement and archiving policy.

QUESTION NO: 11

Which of the following are methods for adding inputs in Splunk? (select all that apply)

A. CLI

B. Splunk Web

C. Editing inputs.conf

D. Editing monitor.conf

ANSWER: A B C

Explanation:

Splunk Enterprise supports configuring data inputs through Splunk Web, the Splunk command-line interface (CLI), and the `inputs.conf` configuration file. Splunk Web provides the graphical workflow for creating common inputs, such as monitored files and directories, network ports, and scripted inputs. The CLI can also create and manage inputs from a command shell, which is useful for repeatable administrative procedures and environments where a graphical interface is unavailable. Directly editing `inputs.conf` is the file-based configuration method, commonly used for deployment automation, app-based configuration, and settings that need to be managed consistently across Splunk instances. Input definitions created through Splunk Web or the CLI are ultimately stored as configuration settings, including in `inputs.conf`. These methods apply to Splunk Enterprise components that perform data collection, such as indexers and heavy forwarders, subject to the specific input type and deployment architecture. Splunk documents the available input-configuration methods and the relationship between Web, CLI, and configuration files in its data onboarding guidance. See Configure your inputs and the inputs.conf specification for details.

QUESTION NO: 12

Which of the following are supported configuration methods to add inputs on a forwarder? (select all that apply)

A. CLI

B. Edit inputs.conf

C. Edit forwarder.conf

D. Forwarder Management

ANSWER: A B D

Explanation:

CLI, Edit inputs.conf, and Forwarder Management are supported ways to configure inputs on a forwarder. The universal forwarder CLI can create and manage supported input definitions directly, such as monitored files and directories. This is useful when an administrator needs to configure a local forwarder from a command line or automate configuration through scripts. Inputs can also be defined declaratively in `inputs.conf`; Splunk reads the input stanzas from this configuration file and collects data according to those definitions. The file can be placed in an appropriate local or app configuration directory, depending on how the forwarder is managed.

Forwarder Management supports centralized administration through a deployment server. An administrator can package an `inputs.conf` file in a deployment app, assign that app to a server class, and deploy it to matching forwarders. When the app is received, the forwarder uses the deployed input configuration. This approach is especially valuable for consistently enabling the same data inputs across many forwarders and maintaining those settings centrally. Splunk documents CLI and configuration-file input definition for universal forwarders, as well as deployment-server management of forwarder apps, at Forward data to Splunk Enterprise and About the deployment server.

QUESTION NO: 13

Which valid bucket types are searchable? (Choose all that apply.)

A. Hot buckets

B. Cold buckets

C. Warm buckets

D. Frozen buckets

ANSWER: A B C

Explanation:

Hot buckets, cold buckets, and warm buckets are searchable bucket types in Splunk Enterprise. Splunk writes incoming events to a hot bucket, which remains searchable while data is actively being indexed. When a hot bucket rolls to warm, its data is still stored locally and remains available for normal search activity. As data ages further, warm buckets can roll to cold storage; cold buckets also remain searchable because Splunk retains the bucket's index files and can access them from the configured cold path. These bucket states together represent the searchable portion of an index's data lifecycle. Bucket rolling is controlled by index configuration and can occur due to age, size, or volume limits, but the transition among these states does not remove the data from normal search availability. Splunk's index-management documentation describes hot, warm, and cold as searchable buckets and explains that retention actions eventually move data beyond searchable index storage. See Splunk's How Splunk stores indexes documentation and its index storage configuration guidance for details on bucket lifecycle and storage paths.

QUESTION NO: 14

What event-processing pipelines are used to process data for indexing? (select all that apply)

A. fifo pipeline

B. Indexing pipeline

C. Parsing pipeline

D. Typing pipeline

ANSWER: B C

Explanation:

Indexing pipeline and **Parsing pipeline** are the event-processing pipelines involved in preparing incoming data for indexing in Splunk Enterprise. The parsing pipeline processes raw input streams into individual events. At this stage, Splunk performs key event-processing activities such as line breaking, timestamp extraction, and assigning metadata including host, source, and sourcetype. These operations ensure that raw machine data is structured as timestamped events before it enters an index.

The indexing pipeline then writes those processed events to the configured index. It creates the on-disk index structures that make event retrieval efficient, including raw-data storage and time-series index files. Together, these pipelines form the index-time data path: incoming data is first converted into properly bounded and annotated events, then persisted in Splunk indexes for later search and reporting. Splunk's data-routing and index-time configuration documentation describes parsing as the stage where event boundaries and metadata are determined, while the indexing stage stores the processed data in the index. See How indexing works and Configure index-time field extraction.

QUESTION NO: 15

What is the correct order of steps in Duo Multifactor Authentication?

A. 2. Connect to SAML server

B. 3. Authentication Granted 4 Connect to SAML server

C. 2 Check authentication / group mapping

D. 3. Check authentication / group mapping

ANSWER: C

Explanation:

The correct step is **"2 Check authentication / group mapping"**. Before Duo can perform secondary authentication, Splunk must evaluate the user's primary authentication state and determine the authorization context that applies to that identity, including any configured group-to-role mappings. This establishes which Splunk roles and permissions should be assigned if the user successfully completes the login process. Once that initial authentication and mapping evaluation has occurred, the integration can proceed with contacting Duo's cloud service over HTTPS for the second-factor challenge. Duo then returns the authentication result to Splunk, allowing Splunk to complete or deny the session creation according to the response. Treating authentication and group mapping as the second stage reflects that authorization information must be available as part of the login workflow before the protected Splunk session is finalized. Duo's Splunk integration documentation describes the connection between Splunk and the Duo service and the secondary-authentication flow: Duo Splunk documentation. Splunk's documentation also describes how authentication mechanisms and role mapping govern user access: Splunk role mapping documentation.

QUESTION NO: 16

Which configuration file contains the `master_uri` setting used by a license peer to identify its license manager?

A. license.conf

B. server.conf

C. authentication.conf

D. deploymentclient.conf

ANSWER: B

Explanation:

`server.conf` contains the `[license]` stanza, including the `master_uri` setting used by a license peer to identify the license manager. The URI specifies the management endpoint of the Splunk Enterprise instance responsible for license administration.

Configuring a license peer to communicate with a license manager centralizes license usage reporting and license entitlement management. See the server.conf specification and Splunk documentation for managing licenses.

QUESTION NO: 17

What is a role in Splunk? (select all that apply)

A. A classification that determines what capabilities a user has.

B. A classification that determines if a Splunk server can remotely control another Splunk server.

C. A classification that determines what functions a Splunk server controls.

D. A classification that determines what indexes a user can search.

ANSWER: A D

Explanation:

In Splunk Enterprise, a role is a collection of permissions and settings that is assigned to users, enabling administrators to manage access consistently rather than configuring each user individually. "A classification that determines what capabilities a user has." is correct because capabilities are the granular permissions that authorize actions in Splunk, such as searching, managing knowledge objects, or administering particular features. Roles can inherit capabilities from other roles, which supports efficient, reusable access-control design.

“A classification that determines what indexes a user can search.” is also correct. A role’s index settings control which indexes its users may search, including both indexes searched by default and indexes that may be searched when explicitly specified. This is a core part of Splunk’s role-based access control: users receive access to data through their assigned roles, subject to the role’s configured index permissions. Splunk administrators configure roles and assign users to them through Splunk Web or authorization configuration, allowing security requirements to be applied uniformly across users with comparable responsibilities. See Splunk’s documentation on users and roles and adding and editing roles.

QUESTION NO: 18

Which forwarder is recommended by Splunk to use in a production environment?

A. Heavy forwarder

B. SSL forwarder

C. Lightweight forwarder

D. Universal forwarder

ANSWER: D

Explanation:

The **Universal forwarder** is the Splunk-recommended forwarder for most production data-collection deployments. It is a lightweight, dedicated Splunk component designed to reliably monitor files, directories, and operating-system event sources, then forward the resulting data to indexers or intermediate forwarders. Because it performs minimal processing, it has a small CPU and memory footprint and is appropriate for installation across large fleets of production servers and endpoints. It also supports production-relevant capabilities such as load balancing across receiving indexers, automatic failover, acknowledgment-based delivery where configured, encrypted forwarding, and deployment-server-based configuration management. Splunk documentation describes the universal forwarder as the best way to forward data from production systems in the overwhelming majority of cases. A heavier processing tier is used only when requirements call for parsing, filtering, routing, or other index-time transformations before data reaches indexers. For standard, scalable collection and forwarding from production hosts, the universal forwarder provides the intended architecture and operational efficiency. See Splunk’s Universal Forwarder documentation and its deployment guidance.

QUESTION NO: 19

Which of the following statements describe deployment management? (select all that apply)

A. Requires an Enterprise license

B. Is responsible for sending apps to forwarders.

C. Once used, is the only way to manage forwarders

D. Can automatically restart the host OS running the forwarder.

ANSWER: A B

Explanation:

Deployment management in Splunk Enterprise is provided by the deployment server, which centrally distributes configuration changes, apps, and other content to deployment clients. Forwarders are common deployment clients, so sending apps to forwarders is a core deployment-management use case. For example, an administrator can use server classes to target particular groups of universal forwarders and distribute a technology add-on, inputs configuration, or outputs configuration to those clients. This allows standardized configuration management at scale rather than requiring app installation individually on every forwarder.

A Splunk Enterprise instance acting as a deployment server is also a management component and must have access to a Splunk Enterprise license. Splunk’s distributed-deployment licensing guidance explicitly identifies the deployment server as a

management component requiring Enterprise licensing access. These two characteristics describe deployment management: it is a licensed Splunk Enterprise management function and it distributes apps/configuration to forwarders and other deployment clients. See Splunk's deployment server documentation and its distributed deployment licensing documentation.

QUESTION NO: 20

An index stores its data in buckets. Which default directories does Splunk use to store buckets? (Choose all that apply.)

A. bucketdb

B. frozendb

C. colddb

D. db

ANSWER: C D

Explanation:

Splunk stores index buckets in path locations configured in `indexes.conf`. By default, the `db` directory is the index's home path, where Splunk keeps hot and warm buckets. Hot buckets are actively being written as data is indexed; once rolled, warm buckets remain in that same home-path location until they age or meet configured rollover criteria. The `colddb` directory is the default cold path, where Splunk moves cold buckets after they roll from warm storage. Thus, both directories are standard default bucket-storage locations used by an index lifecycle. The actual parent path is normally based on `$SPLUNK_DB`, with per-index directories beneath it. Administrators can change these locations through the `homePath` and `coldPath` settings in `indexes.conf`, such as when hot/warm and cold data require different storage tiers. Splunk's documentation describes the bucket lifecycle and the default home and cold paths, while the configuration reference defines the corresponding index settings. See How Splunk stores indexes and the indexes.conf specification.

QUESTION NO: 21

Which of the following enables compression for universal forwarders in outputs. conf ?

A)

B)

C)

D)

A. Option A

B. `compressed = true`

C. Option C

D. Option D

ANSWER: B

Explanation:

`compressed = true` enables compression for data sent by a universal forwarder through a TCP or SSL output configured in `outputs.conf`. This setting is normally placed in the applicable `[tcpout]` stanza or a target-group stanza used by the forwarder. Once enabled, the forwarder compresses event data before transmitting it to the receiving Splunk Enterprise instance. Compression can reduce network bandwidth consumption, which is particularly useful when forwarding substantial data volumes or when the connection between the forwarder and indexer has limited capacity.

The receiving side must also be configured to accept compressed forwarded data on its receiving port. Therefore, compression is an end-to-end forwarding configuration: the universal forwarder enables it for its output, and the receiver enables it for the corresponding Splunk-to-Splunk input. Splunk documents `compressed` as a Boolean output setting and specifies that it applies to TCP and SSL outputs. See the outputs.conf configuration reference and Splunk's forwarding configuration documentation for the output-stanza configuration model.

QUESTION NO: 22

How is data handled by Splunk during the input phase of the data ingestion process?

A. Data is treated as streams.

B. Data is broken up into events.

C. Data is initially written to disk.

D. Data is measured by the license meter.

ANSWER: C

Explanation:

Data is initially written to disk. During the input phase, Splunk Enterprise receives raw data from configured inputs, such as monitored files, network ports, scripted inputs, or forwarding sources. Splunk handles this incoming data as a continuous stream and divides it into manageable chunks. These chunks are placed into queues and written to disk before the later stages perform event processing and indexing. This disk-based handling provides buffering and helps Splunk manage incoming data reliably, including when downstream processing is temporarily slower than the input rate.

At this point in the pipeline, the data is still raw: it has not yet undergone the parsing activities that establish event boundaries, extract timestamps, or apply line-breaking rules. Persisting the incoming data in Splunk's pipeline queues supports durable processing as it moves onward through parsing and indexing. Splunk's data pipeline documentation describes the input phase as the stage where Splunk acquires incoming data and writes the resulting chunks to disk for subsequent pipeline processing. See Splunk's data pipeline overview and documentation on event processing and timestamp extraction.

QUESTION NO: 23

When are knowledge bundles distributed to search peers?

A. After a user logs in.

B. When Splunk is restarted.

C. When adding a new search peer.

D. When a distributed search is initiated.

ANSWER: D

Explanation:

Knowledge bundles are distributed when a distributed search is initiated. In a distributed search environment, the search head must make the knowledge objects and configuration context needed to execute the search available to its search peers. That context can include items such as props.conf and transforms.conf settings, event types, tags, lookups, field extractions, and other relevant knowledge objects. Splunk packages the applicable configuration into a knowledge bundle and sends it to the peers before they perform their portions of the search.

This mechanism helps ensure that search-time processing is consistent across the search head and all participating peers. Bundle replication is tied to dispatching distributed searches, rather than being a general action performed simply at sign-in, service restart, or peer enrollment. Splunk also uses bundle checks and replication controls to avoid unnecessarily

transferring an unchanged bundle for every search, but the operational trigger is the initiation and dispatch of a distributed search. See Splunk's documentation on what search heads send to search peers and propagating knowledge objects to indexers.

QUESTION NO: 24

Within props. conf, which stanzas are valid for data modification? (select all that apply)

A. Host

B. Server

C. Source

D. Sourcetype

ANSWER: A C D

Explanation:

Host, **Source**, and **Sourcetype** are valid stanza scopes in `props.conf` for applying data-processing settings. Splunk identifies incoming events using host, source, and sourcetype metadata, and a props stanza can target one of those values to define parsing and transformation behavior. In particular, these stanzas can contain data-modification directives such as `TRANSFORMS-`, which reference transform definitions in `transforms.conf`. Those transforms can perform index-time actions including masking event content, routing data, setting or overriding metadata, and other supported transformations. A host-targeted stanza uses the `host::<host>` form, while a source-targeted stanza uses `source::<source>`; a sourcetype-targeted stanza is named with the sourcetype itself. This lets administrators apply the same processing rule at the metadata scope that best matches the incoming data. Splunk's props.conf specification documents host, source, and sourcetype as supported stanza types and describes their use in assigning event-processing properties. See the props.conf configuration reference and Splunk's documentation on index-time operations.

QUESTION NO: 25

What are responsibilities of an indexer cluster manager? (Choose all that apply.)

A. Coordinate bucket replication.

B. Maintain peer-node membership.

C. Enforce replication and search factors.

D. Store all primary copies of indexed data.

ANSWER: A B C

Explanation:

The cluster manager coordinates key administrative functions for an indexer cluster. It maintains cluster state, tracks bucket copies across peer nodes, and directs peer nodes to meet the configured replication factor and search factor. It also manages peer-node membership, including the process of adding or removing peer nodes from the cluster.

The cluster manager does not store the primary searchable copies of indexed data or execute normal distributed searches. Those functions are performed by peer nodes. See Splunk documentation on indexer clusters.

QUESTION NO: 26

Which valid bucket types are searchable? (select all that apply)

A. Hot buckets

B. Cold buckets

C. Warm buckets

D. Frozen buckets

ANSWER: A B C

Explanation:

Hot buckets, Cold buckets, and Warm buckets are searchable because they are active stages of Splunk's index-data lifecycle. Hot buckets receive newly indexed events and remain searchable while data is being written. When a hot bucket rolls to warm, it is no longer actively written but remains in the index and available to searches. As data ages further, warm buckets roll to cold storage; cold buckets are also retained by the indexer and remain searchable, although their location and storage characteristics can differ from warm data.

Splunk searches the buckets that are currently managed as part of the index's searchable data. The bucket lifecycle can subsequently move data to frozen status when its retention period is reached. Frozen data is no longer maintained as searchable index data unless it is restored and thawed; restored data is placed in a thawed location and can then be searched. Thus, the listed searchable lifecycle states are the hot, warm, and cold bucket types. For Splunk's bucket lifecycle definitions and retention behavior, see How Splunk stores indexes and Set a retirement and archiving policy.

QUESTION NO: 27

What is the **order of precedence** (from **lowest** → **highest**) within **serverclass.conf** in which attributes will be expressed?

A. [global] → [serverClass: < name >] → [serverClass: < name > :client: < client.name >]

B. [global] → [serverClass: < name >] → [app: < appname >]

C. [global] → [serverClass: < name >] → [serverClass: < name > :app: < appname >]

D. [global] → [serverClass: < name >] → [serverClass: < name > :client: < client.name > :user: < username >]

ANSWER: C

Explanation:

The correct lowest-to-highest attribute precedence in `serverclass.conf` is `[global]`, then `[serverClass:<name>]`, then `[serverClass:<name>:app:<appname>]`. This hierarchy lets a deployment server establish broad default settings in the global stanza, specialize those settings for a particular server class, and then provide the most targeted setting for an individual deployment app assigned to that server class. When the same supported attribute is defined at more than one of these levels, the value in the more specific stanza takes precedence. Thus, settings at the server-class-and-app level override values inherited from the server class, and server-class values override global defaults. This structure is important when administering deployment clients because it supports reusable global configuration while retaining precise control of how a particular app is deployed to a particular collection of clients. Splunk documents these stanza types and their precedence in the serverclass.conf specification. See the serverclass.conf specification and Splunk's guidance on using the deployment server.

QUESTION NO: 28

Which of the following indexes come pre-configured with Splunk Enterprise? (select all that apply)

A. _license

B. _internal

C. _external

D. _thefishbucket

ANSWER: B D

Explanation:

`_internal` and `_thefishbucket` are preconfigured Splunk Enterprise indexes. The `_internal` index stores Splunk Enterprise's own operational data, including log events generated by Splunk processes. Administrators use these events to monitor platform activity, diagnose operational problems, and investigate messages produced by Splunk components.

`_thefishbucket` is also supplied by default. It contains file-tracking information used by the input-processing system, particularly the seek addresses that help Splunk determine where it previously stopped reading a monitored file. This tracking enables Splunk to resume file ingestion appropriately and helps prevent unnecessary re-indexing of data that has already been processed. Although it is a preconfigured index, it is intended for Splunk's internal input-management use rather than as a general destination for user data.

Splunk documents both indexes among the default indexes installed with Splunk Enterprise and describes their respective roles in internal logging and file-input tracking. See About default indexes and How log file rotation works.