SPLK-3002 Training Course

Splunk IT Service Intelligence Certified Admin Exam

Structured Learning & Certification Preparation

Table of Contents

Introduction

The SPLK-3002 Splunk IT Service Intelligence Certified Admin certification is intended to validate the knowledge required to administer and manage Splunk IT Service Intelligence (ITSI) environments. It reflects an understanding of how ITSI supports service monitoring, event management, and operational visibility. In modern IT environments, where service health and system observability are essential, this certification represents the ability to structure, maintain, and troubleshoot service intelligence platforms effectively.

About This Training / Certification

This certification focuses on administrative and implementation-level competencies within Splunk IT Service Intelligence. It is generally positioned at an intermediate to advanced level and assumes prior familiarity with Splunk fundamentals. The certification fits into a broader learning path by extending core Splunk knowledge into service-oriented monitoring, covering how to design, implement, and manage ITSI components as part of an integrated operational system.

What We Offer (AAAdemy)

AAAdemy provides structured training resources designed to support certification preparation and skill development across a wide range of IT domains. Our learning materials are built around clear knowledge structures, practical study guidance, and exam-oriented practice to help learners progress with confidence.

We offer well-organized knowledge explanations that break down complex topics into clear, understandable sections aligned with official exam objectives and real-world skill requirements. Each topic is designed to support both conceptual understanding and practical application.

Our study plans and learning guidance help learners follow a logical progression, focusing on key concepts, common pitfalls, and effective preparation strategies. This approach enables learners to study efficiently while maintaining a clear view of their learning goals.

To reinforce understanding, AAAdemy also provides practice questions and exam-focused insights that reflect typical certification scenarios. These resources are intended to help learners evaluate their readiness and strengthen their confidence before taking an exam.

All content is designed for flexible, self-paced learning, allowing individuals to study independently or alongside their existing professional or academic commitments.

Knowledge Overview

Area: Introducing ITSI
This area establishes the conceptual foundation of Splunk IT Service Intelligence. Candidates should understand what ITSI is designed to achieve, how it extends core Splunk capabilities, and why a service-centric monitoring model is useful in enterprise operations. This includes understanding the relationship between services, KPIs, entities, episodes, visual layers, and investigative workflows. The emphasis is on seeing ITSI not as a collection of separate features, but as an integrated framework that turns operational data into service-aware insight.

Area: Glass Tables
This area focuses on how ITSI communicates service health through visual context. Candidates should understand how glass tables are used to represent services, infrastructure components, and business-relevant status in a way that is easy to interpret. This includes the role of visual objects, status indicators, drilldown context, and the mapping between design elements and underlying monitoring logic. Conceptually, glass tables are important because they help operational teams move from raw system data to a shared, understandable view of service condition.

Area: Managing Notable Events
This area covers how ITSI surfaces and organizes important operational issues. Candidates are expected to understand how notable events are generated, how they are reviewed and managed, and how they support incident awareness within service monitoring workflows. This includes understanding the difference between raw alert signals and actionable operational events, as well as the role of event grouping and prioritization. The key idea is that event management should reduce noise and help teams focus on issues that matter most to service health.

Area: Investigating Issues with Deep Dives
This area focuses on investigative analysis using KPI-centric visual exploration. Candidates should understand how deep dives help teams analyze service behavior over time, compare KPI patterns, and identify possible causes of degradation or instability. This includes recognizing trend shifts, spikes, abnormal behavior, and timing relationships across related indicators. Deep dives are significant because they bridge the gap between high-level service status and detailed operational investigation.

Area: Installing and Configuring ITSI
This area addresses the administrative setup required to make ITSI functional and reliable. Candidates should understand the purpose of installation and configuration activities, the importance of environment readiness, and the basic administrative decisions that affect how ITSI will operate. This includes understanding how configuration supports later work with services, KPIs, event management, and visualizations. The deeper concept is that a stable ITSI deployment depends on correct preparation, sound configuration, and a clear awareness of system dependencies.

Area: Designing Services
This area focuses on the logical design of services before implementation. Candidates should understand how to translate technical systems and business capabilities into meaningful service models that can be monitored effectively. This includes identifying service boundaries, organizing supporting components, and defining relationships that reflect operational reality. Good service design is essential because inaccurate service structures can lead to misleading health calculations, weak visibility, and poor monitoring outcomes.

Area: Data Audit and Base Searches
This area emphasizes the quality and structure of the data that drives ITSI. Candidates should understand why source data must be validated, how base searches provide the foundation for KPI calculations, and how poor data quality can undermine service intelligence. The focus is on ensuring that monitoring logic is built on stable, relevant, and interpretable data. This area is conceptually important because ITSI can only produce meaningful service health assessments when its underlying search logic is sound.

Area: Implementing Services
This area covers the practical realization of service models within ITSI. Candidates should understand how designed services are configured in the platform, how KPIs are attached to them, and how the resulting service structure supports monitoring and analysis. This includes the relationship between design choices and implementation outcomes. The main expectation is not only to create services technically, but to implement them in a way that preserves clarity, maintainability, and operational relevance.

Area: Thresholds and Time Policies
This area addresses how KPI values are interpreted under changing conditions. Candidates should understand how thresholds define service states and how time policies allow different expectations to apply at different times. This includes understanding that normal and abnormal behavior may vary depending on workload patterns, business cycles, or operational windows. The deeper purpose of this area is to align monitoring logic with real-world operating conditions rather than treating all values as equally meaningful at all times.

Area: Entities and Modules
This area focuses on structure, organization, and reuse within ITSI. Candidates should understand how entities represent monitored objects and how modules support scalable organization of monitoring content. This includes understanding how these components help create consistency across services and reduce administrative complexity. Conceptually, this area is about building an ITSI environment that remains understandable and manageable as the scope of monitoring grows.

Area: Templates and Dependencies
This area covers standardization and relationship modeling. Candidates should understand how templates support repeatable service creation and how dependencies express the influence of one service on another. This includes recognizing how dependency modeling improves the interpretation of downstream and upstream health conditions. The value of this area lies in its ability to make service architectures more realistic, reusable, and aligned with how systems actually interact.

Area: Anomaly Detection
This area introduces a more adaptive approach to identifying unusual behavior. Candidates should understand the purpose of anomaly detection, how it differs from fixed-threshold evaluation, and why it can be useful in environments where normal behavior is variable or difficult to define statically. The expectation is to understand anomaly detection as an analytical capability that complements traditional monitoring logic by highlighting patterns that deserve attention even when standard thresholds may not be crossed.

Area: Correlation and Multi KPI Searches
This area focuses on extracting better operational meaning from multiple related signals. Candidates should understand how correlation helps connect events or indicators across the environment and how multi-KPI searches support more efficient or more context-rich service monitoring. This includes understanding the value of

combining signals rather than evaluating every metric in isolation. Conceptually, this area is about improving signal quality, reducing fragmentation, and making monitoring results more operationally useful.

Area: Aggregation Policies
This area covers how individual KPI states contribute to overall service health. Candidates should understand the logic used to combine multiple indicators into a single health representation and how different aggregation choices can change the interpretation of service condition. This is important because service health is rarely determined by one metric alone. The deeper understanding required here is that aggregation is not only a technical setting, but also a modeling decision that defines how operational importance is expressed.

Area: Access Control
This area addresses governance and controlled administration within ITSI. Candidates should understand how permissions and role-based access affect visibility, editing rights, and operational responsibility. This includes awareness of how access boundaries protect administrative integrity while still enabling collaboration across teams. The broader concept is that service intelligence environments must be managed in a way that is secure, organized, and appropriate for enterprise operational practices.

Area: Troubleshooting ITSI
This area focuses on diagnosing issues across the ITSI stack. Candidates should understand how to approach problems methodically, whether they originate in data inputs, searches, service configuration, KPI behavior, visual layers, or event handling logic. Troubleshooting requires the ability to trace how information moves through ITSI and identify where expected behavior breaks down. This area is especially important because effective administration depends not only on building monitoring content, but also on maintaining its accuracy, stability, and usefulness over time.

Detailed Knowledge Explanation

SPLK-3002 Introducing ITSI

In the landscape of modern enterprise observability, Splunk IT Service Intelligence (ITSI) functions as the strategic bridge between raw machine data and business resilience. Traditional monitoring often suffers from a "component-centric" bias—alerting on a single server's CPU spike without understanding if that spike actually degrades the customer’s checkout experience. Moving to a "service-centric" model is critical; it allows organizations to map technical health to business outcomes, ensuring that IT teams prioritize incidents based on their actual impact on revenue and operational stability.

1. Understanding the Basics — What is ITSI?

IT Service Intelligence (ITSI) is a specialized analytics solution built on the Splunk platform that serves as the "central brain" of an IT organization. While traditional tools provide siloed views of infrastructure, ITSI synthesizes data from across the stack to provide high-level situational awareness. By correlating disparate data points in real time, ITSI transforms fragmented logs into a unified narrative of service health.

1.1. ITSI as a Splunk Premium App

Architecturally, it is vital to distinguish ITSI as a premium offering. Unlike core Splunk features, ITSI requires a separate license and a specific installation package. For enterprise architects, this means planning for additional resource overhead and dedicated licensing management to unlock advanced capabilities such as machine learning-driven thresholds and service-level visualizations.

2. The Big Picture — What Makes ITSI Special?

ITSI's value is anchored by three primary pillars:

- **Service-Centric Monitoring:** Contextualizes technical metrics within the framework of business functions (e.g., identifying that a database lag is specifically impacting the "Payment Processing Service").
- **Real-Time Data Analysis:** Processes high-velocity data to update health scores and dashboards instantly, enabling a reactive speed that minimizes Mean Time to Resolution (MTTR).
- **Predictive Intelligence:** Leverages machine learning to establish dynamic baselines and identify anomalies, shifting operations from a reactive "break-fix" cycle to proactive prevention.

3. Core Components – Introduction

The ITSI ecosystem relies on five functionally interdependent components. To understand the "central brain" logic, consider the operational flow: a **Base Search** feeds a **KPI**, which determines the health of a **Service**. If a threshold is breached, an **Action Rule** generates a **Notable Event**, which is visualized on a **Glass Table**. For root cause analysis, an architect then pivots from that event into a **Deep Dive**.

3.1. Services

A service is the logical representation of a business or technical function (e.g., "Online Banking" or "Middleware Cluster"). Internally, services are comprised of KPIs, predefined thresholds, and parent-child dependencies. These elements aggregate to produce a **Health Score (0–100)**, providing an immediate numerical assessment of risk and stability.

3.2. KPIs (Key Performance Indicators)

KPIs are the fundamental metrics built using Splunk Processing Language (SPL) to quantify performance. Setting accurate thresholds (Normal, Warning, Critical) is a critical implementation step, as these values dictate how the KPI influences the broader service health score.

3.3. Expand KPI Sources: Logs, Metrics, Traces

Modern KPIs have evolved to support full-stack observability by incorporating **OpenTelemetry-formatted traces**. By integrating logs, metrics, and distributed traces, ITSI provides a granular view of microservice interactions, allowing architects to monitor complex, cloud-native environments with the same rigor as legacy infrastructure.

3.4. Glass Tables

Glass Tables are dynamic, interactive dashboards that provide a layered view of the environment. They support both real-time data binding for active monitoring and historical data for retrospective analysis. Their drag-and-drop interface allows for the creation of high-level maps that react visually to system changes.

3.5. Notable Events

Notable Events are actionable alerts triggered by threshold breaches. To prevent "alert fatigue," ITSI utilizes suppression policies and aggregation rules, ensuring that only significant, high-context incidents reach the operations team.

3.6. Deep Dives

Deep Dives are the primary investigation tool for root cause analysis. They provide a common timeline for multiple KPIs, allowing analysts to correlate trends and identify the specific moment a degradation began.

The seamless integration of Services, KPIs, Glass Tables, Notable Events, and Deep Dives provides the functional framework necessary for high-level monitoring, establishing the foundation for the technical deployment of the platform.

4. Introducing ITSI Practice Question

Q1: What is the primary purpose of IT Service Intelligence (ITSI) in a modern enterprise environment?
A. To provide a service-centric view that links technical metrics to business services
B. To monitor CPU and memory of individual servers
C. To store Splunk logs in a secure and encrypted format
D. To replace all traditional monitoring tools with a single dashboard

Q2: In ITSI, which component is primarily responsible for tracking the real-time health of a business function like "Login Service"?
A. Notable Event
B. KPI
C. Glass Table
D. Service

Q3: Which ITSI feature allows you to monitor metrics like “response time” or “error rate” and trigger alerts based on threshold breaches?
A. Deep Dive
B. Entity
C. KPI
D. Glass Table

Q4: How does a Glass Table differ from traditional dashboards in ITSI?
A. It is used to define search queries and filters for KPIs
B. It shows raw log data in tabular format
C. It can display service health using visual elements and real-time animations
D. It can only be used by Splunk admins for backend monitoring

Q5: Which of the following best describes a Notable Event in ITSI?
A. An alert generated when a KPI or correlation rule is violated
B. A list of all historical searches across services
C. A graphical element on a Glass Table
D. A time-scheduled data collection job

Q6: What is the role of a Deep Dive in ITSI?
A. To manage access control and role-based permissions
B. To configure base searches for KPI creation
C. To analyze the timeline and trend of multiple KPIs for investigation
D. To visualize logs and search results in raw format

Q7: What component of ITSI is used to combine CPU, memory, and error rate metrics to form a business-level view of system health?
A. KPI
B. Correlation Search
C. Glass Table
D. Service

Q8: What happens in ITSI when a KPI crosses its "Critical" threshold during scheduled monitoring hours?
A. The KPI search is skipped
B. A Notable Event may be created
C. The service is automatically deleted
D. A new Glass Table is generated

Q9: Which component allows users to interact with live visual indicators and drill down into service issues directly from a dashboard?
A. Notable Event
B. KPI
C. Glass Table
D. Service

Q10: Why are KPIs critical to ITSI’s service health calculation?
A. They form the inputs that determine the service's health score
B. They enable access to Splunk’s internal logging system
C. They display historical data only
D. They generate entity groups

SPLK-3002 Installing and Configuring ITSI

A stable ITSI foundation is the prerequisite for enterprise-grade performance. Proper configuration during the installation phase is not merely a checkbox; it is a performance mandate. Incorrectly tuned instances suffer from

sluggish search performance and high search head (SH) concurrency, which can lead to "skipped searches" and stale monitoring data.

1. Overview – What Does It Mean to "Install ITSI"?

Installing ITSI means deploying a premium real-time analytics platform on top of Splunk Enterprise. The goal is to establish a framework capable of high-frequency data collection, machine learning analysis, and complex service-level visualization.

2. What You Need Before You Begin

ITSI demands significantly more resources than standard apps due to its background search load. Architectural requirements include:

- **Splunk Enterprise:** Version 8.0+ is mandatory.
- **System Resources:** Minimum 16 GB RAM (though 32 GB+ is standard for production), multiple CPU cores, and high-performance Disk I/O.
- **Architectural Awareness:** In a **Search Head Cluster (SHC)** environment, installation requires specific attention to unpacking app packages across all members to ensure consistency.

3. Step-by-Step – How to Install ITSI

The workflow involves downloading the `.spl` package and installing via the UI or CLI. Post-installation, it is mandatory to create and verify the following specific indexes:

- `itsi_summary`: Stores KPI results and health scores.
- `itsi_tracked_alerts`: Stores notable events.
- `itsi_grouped_alerts`: Stores aggregated episode data.

4. Post-Installation Configuration

Following the app deployment, administrators must onboard data, configure **ITSI Teams** for role-based access control, and enable **data model acceleration**. This acceleration is essential for maintaining the responsiveness of the Service Analyzer and high-density Glass Tables.

5. Version Upgrade Considerations

Upgrading ITSI preserves existing data and services, whereas a fresh install starts with defaults. Architects should use the **Upgrade Readiness Dashboard** to identify incompatible configurations. **Pro Tip:** Always validate upgrades in a staging environment that mirrors production configurations before proceeding.

6. FAQ-Style Installation Troubleshooting Tips

- **Index Verification:** To verify if ITSI indexes are active and receiving data, run: `| eventcount summarize=false index=*`.

- **License Validation:** Standard licenses will not work; ensure the ITSI premium license is active in the License Manager.
- **UI Issues:** If the Service Analyzer fails to load, verify that the `itsi` app is enabled and the user role possesses the `itsi_admin` capability.

7. Configuration Paths and Log Directories

Deep debugging requires familiarity with the following directories and logs:

- **Configuration:** `$SPLUNK_HOME/etc/apps/SA-ITOA/`
- **Troubleshooting Logs:** `$SPLUNK_HOME/var/log/itsi/itsi_troubleshooting.log` (Toolkit logs).
- **Scheduler Logs:** `itsi_scheduler.log` (Monitors scheduled KPI jobs).
- **Indexing Logs:** `itsi_summary_indexing.log` (Tracks health of summary indexing).

With the platform stabilized and health checks verified, the focus transitions from technical infrastructure to the strategic design of the service model.

8. Installing and Configuring ITSI Practice Question

Q1: What version of Splunk Enterprise is generally required as a minimum to install ITSI?
A. Version 8.0
B. Version 6.5
C. Version 7.2
D. Version 9.2

Q2: What is the format of the ITSI application package that must be downloaded for installation?
A. .pkg
B. .spl
C. .tar.gz
D. .zip

Q3: Which index is primarily used by ITSI to store KPI results and health scores?
A. main
B. summary_index
C. itsi_tracked_alerts
D. itsi_summary

Q4: Why is high disk I/O important when installing and running ITSI?
A. To support multiple user accounts
B. To reduce licensing cost
C. To handle the volume of indexed KPI and alert data
D. To improve system security

Q5: What is the purpose of the ITSI Health Check dashboard?
A. To calculate licensing usage

B. To verify system readiness and detect configuration issues
C. To delete inactive teams and users
D. To schedule Splunk restarts

Q6: After installing ITSI, which task is considered part of post-installation configuration?
A. Creating teams and onboarding data pipelines
B. Verifying the web.conf file
C. Restarting the Splunk indexer
D. Reindexing all historical data

Q7: Which of the following is true about the ITSI license?
A. It must be installed separately from Splunk Enterprise
B. It is not required if only dashboards are used
C. It is included with the free Splunk license
D. It can be shared with the Splunkbase community

Q8: Which method can be used to install the ITSI app via the command line?
A. Use `splunk install app` command inside `$SPLUNK_HOME/bin`
B. Use `sudo dpkg -i` in the Splunk CLI directory
C. Extract `.spl` into `$SPLUNK_HOME/etc/apps/` and restart Splunk
D. Place the app in `/var/lib/splunk` and run `splunk apply`

Q9: Why is it important to configure modular inputs after installing ITSI?
A. To collect internal ITSI logs and support automation
B. To generate data for license enforcement
C. To secure the Splunk web interface
D. To store dashboards in the app context

Q10: Which of the following best describes the function of the `itsi_tracked_alerts` index?
A. Stores raw data used for entity creation
B. Saves KPI thresholds and templates
C. Tracks changes in configuration files
D. Stores notable events and alert data

SPLK-3002 Designing Services

The design phase is the blueprint for ITSI's value proposition. A well-designed service model ensures that technical monitoring is inextricably linked to business outcomes, providing a clear view of how infrastructure health affects the bottom line.

1. What Is a Service in ITSI?

A service is a logical model of a business-critical function. This includes high-level business functions (e.g., "Global Payment Gateway") and the technical infrastructure supporting them (e.g., "Linux Web Cluster").

2. Service Design Considerations

Architects must identify the "most important things" first. The strategy should be to break complex systems into focused, manageable pieces. For instance, rather than one "Retail Website" service, design a hierarchy of "Authentication," "Product Catalog," and "Checkout."

3. Key Elements of a Service

A robust service definition includes:

- **KPI Base Searches:** Efficient SPL queries that extract metric data.
- **Thresholds:** Logic defining the transition from "Normal" to "Critical."
- **Dependencies:** Parent-child relationships that propagate health impacts.

4. Best Practices for Service Design

To ensure scalability, use **Service Templates** for repeated patterns (e.g., regional branches). Maintain a focused scope for each service to prevent "alert storms" and clearly document ownership and business importance for every service created.

5. Health Score Calculation

The Health Score (0–100) is a weighted calculation of all active KPIs.

Scenario: Consider a Retail Banking service where the "Transaction Success Rate" is critical. If this KPI drops to 0%, the architect should configure its "Importance Weight" so high that the health score plummets to 0, even if "Disk Latency" or "CPU Load" remains healthy (Green).

6. Visualizing Services with Glass Tables

Services are the heart of visual monitoring. On Glass Tables, service states are represented via the **Traffic Light Model** (Green/Yellow/Red), allowing NOC teams to identify failing nodes instantly and initiate drill-downs to the underlying technical cause.

These design principles provide the theoretical framework required to begin the hands-on implementation within the ITSI Service Editor.

7. Designing Services Practice Question

Q1: In Splunk ITSI, what does a "Service" represent?
A. A specific metric from an application, like response time
B. A correlation search created for system-wide error detection
C. A dashboard built to show logs from different indexes
D. A logical group of KPIs that represent a business or technical function

Q2: Which of the following is a recommended best practice when designing services in ITSI?
A. Use templates to scale across similar service types
B. Configure thresholds after all dashboards have been created
C. Combine all business processes into one service to simplify visibility
D. Avoid defining dependencies to reduce complexity

Q3: Why should complex systems be broken into smaller services when designing ITSI architecture?
A. To allow more user dashboards to be created
B. To avoid the use of Glass Tables
C. To reduce licensing usage
D. To focus on measurable, manageable parts of the system

Q4: What is a "KPI base search" in the context of a service?
A. A Splunk search that defines how to retrieve KPI data
B. A UI layout showing related service components
C. A template for generating notable events
D. A background process that syncs services

Q5: What is the main reason for defining thresholds within KPIs?
A. To improve Splunk indexer memory usage
B. To evaluate KPI status and trigger Notable Events
C. To determine when the service is eligible for backup
D. To reduce the number of base searches

Q6: A "parent-child" relationship between services in ITSI is used to model:
A. Time synchronization between KPIs
B. Dependencies that affect a service's health score
C. Visual representation of dashboard elements
D. User account authentication between applications

Q7: Which of the following would be the best example of a service in ITSI?
A. An entity used in a correlation search
B. A system resource like memory usage on a single host
C. A collection of saved searches from the Splunk search app
D. A function like “Payment Gateway” with KPIs and dependencies

Q8: When building a service, why is documenting its purpose and scope important?
A. To assist with long-term maintenance and team ownership
B. To comply with Splunk licensing policies
C. To make it visible in the Glass Table
D. To reduce the number of KPIs required

Q9: What is the benefit of using "split-by" fields in KPI base searches?
A. They allow KPI calculations across different entities
B. They enable KPIs to trigger without thresholds
C. They simplify correlation search configurations
D. They increase the storage capacity of Splunk indexes

Q10: How do dependencies between services affect health score calculations?
A. They slow down KPI searches during system load
B. They are not factored into the overall health of the parent service
C. They allow health scores to reflect the state of related services
D. They are only used for Glass Table animations

SPLK-3002 Implementing Services

Implementation transforms a designed blueprint into a functional monitoring state. Utilizing the **Service Editor**, architects activate KPIs and dependencies to create a live representation of the environment's health.

1. From Service Design to Deployment

Implementation is the practical activation of the planned service model. This phase moves the project from conceptual diagrams to active data processing and health score calculation.

2. Steps in Service Implementation

1. **Create Service:** Define the name and assign the owner team.
2. **Add KPIs:** Build base searches or select from a library.
3. **Set Thresholds:** Define the judgement criteria (Static, Dynamic, or ML).
4. **Configure Dependencies:** Define the parent-child impact chain.
5. **Assign Access:** Enforce data governance through role-based permissions.

3. Threshold Management Options

- **Static:** Fixed values for metrics with well-known limits.
- **Dynamic:** Based on percentage changes or trending patterns.
- **Machine Learning (Anomaly Detection):** Uses historical data to learn "normal" patterns. Configuring ML thresholds requires setting the **sensitivity level**, **learning period**, and **re-training frequency**.

4. Time Policies – Why and How to Use Them

Time Policies adapt KPI sensitivity to business cycles. For example, higher error rates might be tolerated during a 2:00 AM maintenance window compared to peak trading hours. This adaptability is essential for reducing false positives and operational noise.

5. Using Service Templates During Implementation

Templates are the key to enterprise scalability. By applying a template, a new service inherits all predefined KPIs and health scoring logic. Templates enable **bulk updates**; modifying a single template can update hundreds of linked services simultaneously, ensuring configuration consistency.

6. Health Score Calculation and KPI Importance

The health score reflects the weighted severity of KPIs. Architects must align importance weights with business risk—a "Critical" state on a "Security" KPI should have a more drastic impact on the score than a "Warning" state on "Memory Usage."

7. Post-Implementation Verification and Adjustment

Validation is mandatory. Architects should push sample data and use **Deep Dives** to verify if thresholds are too strict or if status transitions occur as expected. Tuning importance weights post-deployment is a standard optimization task.

Effective implementation ensures that the subsequent data search foundation is built upon a verified and technically sound logical model.

8. Implementing Services Practice Question

Q1: What is the first step when implementing a service in ITSI?
A. Assign a time policy to all KPIs
B. Set up anomaly detection
C. Configure parent-child relationships
D. Create the service and provide a name and description

Q2: Why might you assign a parent-child structure between services in ITSI?
A. To suppress alert generation during non-critical failures
B. To enable static threshold evaluation
C. To model service dependencies and understand cascading health impacts
D. To automatically duplicate KPIs between services

Q3: What should be configured after creating a service in ITSI?
A. Add relevant KPIs with base searches, thresholds, and importance
B. Activate a modular input for each KPI
C. Generate a correlation search template
D. Upload KPI results to the Service Analyzer

Q4: How does assigning a service to a team benefit implementation?
A. Links the service directly to a correlation search
B. Organizes access control and ownership based on roles
C. Bypasses the need for threshold configuration
D. Allows services to run faster

Q5: Which of the following is true about KPI thresholds in service implementation?
A. They define what states (Normal, Warning, Critical) mean for each KPI
B. Thresholds are optional in production environments
C. They are set only at the global service level
D. Only dynamic thresholds are allowed

Q6: Which threshold type is best when KPI behavior is inconsistent or non-linear?
A. Static threshold
B. Hard-coded maximum value
C. Machine Learning-based threshold
D. Percentage threshold

Q7: What is a key benefit of using time policies in KPI configuration?
A. Generate alerts without thresholds
B. Adjust threshold sensitivity based on business hours
C. Allow KPIs to use multiple base searches
D. Increase the frequency of summary indexing

Q8: Which scenario best describes a use case for configuring dependencies between services?
A. A dashboard needs additional visualizations
B. An authentication service fails and impacts the user login process
C. A storage system manages multiple users across time zones
D. KPIs need to be split by host

Q9: What should be included when defining each KPI during service implementation?
A. A service module name and dashboard reference
B. A Notable Event severity preset
C. The SPL query, thresholds, importance, and optional split-by fields
D. An audit report filter

Q10: How do dynamic thresholds operate differently from static ones in ITSI?
A. They react to changes in percentage or trend over time
B. They use a fixed numeric value for evaluation
C. They ignore health score calculation
D. They require external data enrichment

SPLK-3002 Data Audit and Base Searches

Search efficiency is the cornerstone of ITSI performance. Base searches are the foundation upon which all KPIs are built; poorly written searches can cause significant system strain or provide inaccurate health data.

1. What Are Base Searches?

A Base Search is an SPL query that retrieves the specific data needed to measure a KPI. It is the primary engine behind service health scores and notable events.

2. Structure of a Base Search

A robust base search includes the SPL query, defined time ranges (e.g., last 5 minutes), and **split-by fields** (e.g., by `host` or `region`) to enable granular monitoring across entities.

3. Data Audit Tools in ITSI

To ensure data integrity, architects use:

- `itsi_data_integrity`: Detects if data is missing, stale, or incomplete.
- **Search Inspector:** Analyzes search duration and resource usage.
- **Audit Dashboards:** Track search success/failure trends and system performance.

4. Optimization Tips for Base Searches

Architectural Mandate: To ensure system stability, practitioners **must avoid** the use of `join`, `append`, or nested subsearches in KPI base searches, as these often slow down or fail silently. Best practices include:

- Limiting time ranges and pulling only required fields.
- Using **summary indexing** to build KPIs from lighter, pre-processed data.
- Staggering search schedules using **offset cron times** (e.g., running at `:02` instead of `:00`) to avoid Search Head and Indexer concurrency limits.

5. How to Validate Your Base Search

Always test searches in the SPL Editor before deployment. Use `| head 10` for quick validation, `| stats count` to confirm data matches, and review timestamps to ensure the search is retrieving events within the expected window.

6. Common Mistakes in Base Searches

Pitfalls include missing time ranges (triggering "All Time" searches), misspelled fields, and the misuse of real-time mode, which can consume excessive resources without providing real benefit if indexing is delayed.

7. Base Search Visibility and Permissions

Manage data governance via visibility scopes: **Private**, **App** (ITSI only), or **Global**. Role-based access ensures that only authorized teams can modify base searches, maintaining operational autonomy.

8. Using Base Searches in Templates

Base searches in templates utilize **tokens** (e.g., `$host$`, `$app_name$`). These tokens are replaced at runtime, making a single search logic flexible enough to apply to diverse entities without duplication.

Optimized searches provide the high-quality data stream necessary for high-level visual monitoring tools like Glass Tables.

9. Data Audit and Base Searches Practice Question

Q1: What is the primary function of a base search in ITSI?
A. To define how data is retrieved and calculated for a KPI
B. To configure user roles and access control settings
C. To suppress KPIs during maintenance windows
D. To display visual components on a Glass Table

Q2: Which component helps identify if a KPI's base search has missing or stale data?
A. Base Search Editor
B. itsi_data_integrity
C. Entity Management module
D. Glass Table preview mode

Q3: In ITSI, what is the benefit of using 'split-by' fields in a base search?
A. It enables the KPI to return results broken down by entities like host or region
B. It distributes searches across multiple indexers
C. It enhances real-time correlation search accuracy
D. It triggers automated scripts on threshold breach

Q4: What does the ITSI Search Inspector tool primarily help you analyze?
A. Notable Event severity levels
B. Service dependencies
C. Search performance metrics like duration and skipped intervals
D. Glass Table widget placement

Q5: What is a key optimization when scheduling large numbers of KPI base searches?
A. Run all searches simultaneously for consistency
B. Disable threshold evaluation
C. Use offset cron schedules to distribute system load
D. Enable only real-time searches

Q6: What is the purpose of using summary indexing for base searches in ITSI?
A. To delay alert generation during off-hours
B. To extract fields from the raw log data
C. To visualize raw events in Deep Dives
D. To reduce load by reusing pre-aggregated results

Q7: What part of a base search defines how often it runs in Splunk?
A. KPI severity threshold
B. Search Inspector settings
C. Notable Event prioritization
D. Search scheduling (cron or real-time mode)

Q8: Which of the following is a valid use of filters in base searches?
A. To exclude test or development data from KPI calculations
B. To automatically generate Glass Table layouts

C. To prioritize Notable Events based on source type
D. To manage access control for dashboards

Q9: How do audit dashboards in ITSI assist KPI reliability?
A. By providing drilldowns for KPI visualizations
B. By creating new KPI base searches automatically
C. By enabling user tagging for services
D. By displaying KPI search performance, failures, and skipped intervals

Q10: What is one risk of using a wide time range in a base search?
A. The resulting values may include future predictions
B. The KPI may be too sensitive to changes
C. The search may become slow and overload the system
D. The base search will automatically trigger a restart

SPLK-3002 Glass Tables

Glass Tables represent the "control room" of the enterprise. Beyond mere dashboards, they provide the visual situational awareness required in high-stakes environments like Network Operations Centers (NOCs) to reduce cognitive load and accelerate decision-making during crises.

1. What Are Glass Tables?

A Glass Table is an interactive, dynamic dashboard that uses custom layouts, icons, and shapes to visualize system health. It provides a transparent, layered view of the IT environment connected to live data.

2. What Is the Purpose of a Glass Table?

The objective is to provide a "big picture" summary that supports rapid triage. By visualizing the entire business chain, teams can identify which node is failing based on visual cues (color/animation) rather than parsing raw logs.

3. Key Features of Glass Tables

- **KPI Integration:** Linking shapes to KPIs for dynamic color-coding.
- **Drill-down Actions:** Clicking an icon can launch a Deep Dive or a ServiceNow ticket.
- **Animations:** Arrows can show "flow," and elements can "flash" to draw immediate focus to critical failures.

4. Design Best Practices for Beginners

Architects must maintain the **Traffic Light Model** for status clarity and limit clutter to essential services. Layouts should be tailored: Executives receive "Business Impact" views, while Engineers receive "Technical Metric" views with drill-down links.

5. Glass Table Data Sources

The data flow is linear: **Raw Data** is processed by a **Base Search**, which powers a **KPI**, which is bound to a **Glass Table Panel**. This ensures every visual indicator is backed by structured, query-driven measurements.

6. Animation Trigger Conditions

Animations like pulsing or flashing are specifically triggered by **KPI status changes**. If a service moves from "Normal" to "Critical," the configured panel property can trigger an animation, significantly increasing situational awareness on wall displays.

7. Where to Find the Glass Table Editor (UI Path)

Navigate to **Settings > ITSI > Glass Tables**. The drag-and-drop interface allows for easy binding of visual elements to specific health scores or KPIs.

While Glass Tables provide the summary, investigating the root cause of a failure requires the granular "microscope" of Deep Dives.

8. Glass Tables Practice Question

Q1: What is a key characteristic that differentiates a Glass Table from traditional dashboards in ITSI?
A. It collects raw logs from all indexes in real time
B. It automatically configures KPI thresholds for every service
C. It enables users to design animated, clickable service layouts connected to live KPI data
D. It provides detailed search logs in tabular format

Q2: Which of the following is an appropriate use of a Glass Table?
A. Visualizing the current health of interconnected services in real time
B. Running batch processing jobs during maintenance windows
C. Creating service templates with shared KPIs
D. Writing correlation searches for multiple KPIs

Q3: A company is building a Glass Table for an executive dashboard. Which of the following is a recommended design choice?
A. Use fewer visual elements with a focus on high-level service status
B. Include full log output beneath each visual indicator
C. Show all entity-level metrics across all services
D. Use the same design as an engineering-focused dashboard

Q4: Which visual cue in a Glass Table helps users immediately identify critical conditions?
A. Background images with static text
B. KPI split-by values

C. Color-coded elements linked to KPI thresholds
D. Search scheduler logs

Q5: What is the function of drill-down actions in a Glass Table?
A. They animate service lines to display system latency
B. They suppress event generation during low-priority incidents
C. They allow real-time deletion of Glass Table objects
D. They enable users to navigate directly to relevant dashboards, Deep Dives, or searches

Q6: Which of the following is a best practice when designing a Glass Table for a network operations center (NOC) screen?
A. Use static icons and labels for maximum clarity
B. Avoid using real-time data to improve performance
C. Use a large number of small components to fit all services
D. Include animations and transitions to highlight live system activity

Q7: What type of element would you most likely use to indicate CPU health across multiple servers on a Glass Table?
A. A log viewer widget
B. A shape or icon linked to a split-by KPI
C. A dashboard permissions role
D. An audit search filter

Q8: What is the primary benefit of using animations in a Glass Table design?
A. They make changes in system status immediately visible to the viewer
B. They suppress alerts during critical events
C. They reduce memory usage for large tables
D. They simplify KPI configuration in back-end searches

Q9: In a Glass Table, which component enables a shape or icon to reflect the status of a specific KPI?
A. KPI binding
B. Aggregation policy
C. Module dependency
D. Base search

Q10: A Glass Table is showing all components in green, except for a database node that is red. What can you conclude?
A. The Glass Table configuration needs to be reloaded
B. The database service has reached a critical KPI threshold
C. The database is scheduled for backup
D. The database service is currently under maintenance

SPLK-3002 Investigating Issues with Deep Dives

Deep Dives are the "microscope" of the ITSI ecosystem, providing a time-based methodology for root cause analysis (RCA). They are the essential tools for translating a high-level alert into a technical resolution.

1. What Is a Deep Dive?

A Deep Dive is an interactive investigation board that functions as a "virtual investigation room." It allows analysts to piece together trends and events to tell the story behind a system failure.

2. Purpose of Deep Dives

Deep Dives resolve complexity by showing multiple KPIs on a shared timeline. This exposes the "chain reaction" of failures—such as a database latency spike causing a subsequent surge in API errors.

3. Key Capabilities of Deep Dives

- **Multi-KPI Visualization:** Side-by-side line charts for trend comparison.
- **Event Timeline:** Notable Events are overlaid on KPI charts to show the exact sequence of alerts versus metrics.
- **Interactive Controls:** Analysts can zoom, pan, and rearrange panels to focus on the incident window.
- **Drill-Downs:** Clicking any point on a KPI line allows for a direct pivot into the underlying raw logs.

4. Use Cases for Deep Dives

Deep Dives are utilized for troubleshooting service degradation, performing RCA for major outages, and proactively monitoring new code releases for performance regressions.

5. Deep Dives as RCA Artifacts

Deep Dives can be saved as **Root Cause Analysis (RCA) records**. These records preserve the full investigative context, serving as snapshots for post-incident retrospectives and executive briefings.

6. Deep Dive vs. Episode Review – Key Differences

It is critical to distinguish these two views:

- **Episode Review:** Focuses on **managing** "who and what" (incident triage and response).
- **Deep Dive:** Focuses on **investigating** "how and why" (technical root cause analysis).

Identifying the root cause through a Deep Dive naturally leads to the long-term management of those incidents through Notable Events.

7. Investigating Issues with Deep Dives Practice Question

Q1: What is the primary function of a Deep Dive in ITSI?
A. To visualize real-time logs in a tabular format
B. To configure Notable Events and thresholds

C. To investigate time-based KPI trends and correlate events
D. To automatically suppress known false-positive alerts

Q2: Which feature of Deep Dives helps correlate KPI trends with Notable Events on the same timeline?
A. KPI thresholds
B. KPI Base Search
C. Event Timeline
D. KPI Weighting

Q3: What can users do in a Deep Dive to focus on a specific time window during an incident?
A. Use KPI importance weighting
B. Configure KPI thresholds in real time
C. Apply alert suppression filters
D. Zoom in and pan across the timeline

Q4: What action is possible when clicking on a KPI line in a Deep Dive chart?
A. Launch a Notable Event or related dashboard
B. Open a system access log
C. Create a new KPI
D. Adjust service templates

Q5: What is a key advantage of comparing multiple KPIs on a single timeline in Deep Dives?
A. It displays average KPIs only for one service
B. It enables manual tagging of log entries
C. It improves KPI suppression rules
D. It helps identify relationships between different service metrics

Q6: Which of the following is a common use case for a Deep Dive?
A. Creating entity filters
B. Post-incident analysis and documentation
C. Managing KPI threshold groups
D. Editing correlation searches

Q7: A Deep Dive shows a spike in database response time followed shortly by a rise in API error rates. What does this suggest?
A. The API errors are likely caused by database delays
B. The system is scheduled for backup
C. The KPIs are unrelated and should be viewed separately
D. There is a direct KPI suppression conflict

Q8: How can Deep Dives be used proactively rather than just reactively?
A. To delete events that are over one week old
B. To monitor performance changes after a new software release
C. To manage user access to dashboards
D. To disable KPI searches in low-traffic hours

Q9: What distinguishes Deep Dives from regular dashboards?
A. They use animated glass table overlays
B. They include service templates
C. They offer interactive, multi-KPI time-based analysis
D. They rely only on static KPI charts

Q10: What is typically the first step in using a Deep Dive to troubleshoot an issue?
A. Assigning roles to view dashboards
B. Setting up Splunk system indexes
C. Reconfiguring entity import fields
D. Correlating KPI trends and events

SPLK-3002 Managing Notable Events

Notable Event management defines the incident lifecycle in ITSI. The objective is to transform raw alerts into "actionable intelligence," guiding the response from detection to resolution.

1. What Is a Notable Event?

A Notable Event is a smart, actionable alert generated when a KPI crosses a threshold or a correlation search identifies a problematic data pattern.

2. Lifecycle of a Notable Event

The lifecycle consists of: **Triggering** (condition met), **Aggregation** (grouping into episodes), **Prioritization** (assigning severity), and **Management** (operator triage).

3. Event Management Features

- **Aggregation Policies:** Function as "smart folders" to group related events (e.g., grouping CPU, Memory, and Error alerts from one host into a single episode).
- **Suppression Rules:** Silence alerts during maintenance to reduce noise.
- **Workflows:** Automate responses, such as opening Jira/ServiceNow tickets.

4. Where Are Notable Events Displayed?

The central workspace is the **Episode Review** interface (`ITSI > Episode Review`). This view displays grouped events, metadata, and a timeline, allowing analysts to acknowledge and resolve incidents.

5. How KPIs Trigger Events (via Action Rules)

KPIs connect to the incident workflow through **Action Rules**. These rules define the specific conditions (e.g., "Critical status for > 2 minutes") required to generate a Notable Event and its associated severity.

6. How Prioritization Works

Priority is determined by the **Impact Score** (KPI severity weight), **Entity Importance** (prioritizing production over dev), and custom rules within the aggregation policy. This ensures teams address the most critical business risks first.

Mastery of ITSI for the SPLK-3002 certification requires a comprehensive understanding of this lifecycle—from optimized base searches and service design to high-level visualization and structured incident triage. By integrating these components, IT professionals ensure operational excellence and business resilience in any enterprise environment.

7. Managing Notable Events Practice Question

Q1: What best describes a Notable Event in ITSI?
A. A report showing KPI thresholds and values
B. An alert triggered by threshold breach or correlation pattern
C. A log-level search result saved from Deep Dive
D. A historical log that tracks KPI trends

Q2: What is the purpose of Aggregation Policies in Notable Event management?
A. To group related events to avoid alert fatigue
B. To automate KPI creation
C. To suppress events during maintenance
D. To assign color codes to KPI thresholds

Q3: During a system update, many KPIs are expected to breach thresholds. What feature in ITSI should you use to avoid false alerts?
A. Correlation Search
B. KPI Base Search
C. Event Suppression
D. Episode Review

Q4: What happens during the “Triggering” phase of the Notable Event lifecycle?
A. Events are automatically resolved based on suppression
B. A KPI threshold or correlation rule condition is met
C. ITSI escalates the event to external ticketing systems
D. ITSI checks for duplicate events to remove

Q5: What severity level would you assign to a Notable Event affecting a mission-critical service with major performance degradation?
A. Critical
B. Info
C. High
D. Warning

Q6: How can ITSI help teams act on Notable Events more efficiently?
A. By running a periodic dashboard export
B. By converting KPIs into log files
C. By defining workflows that send alerts or create tickets automatically
D. By suppressing event data for all services

Q7: A Notable Event includes fields like “team,” “region,” and “system type.” What is this functionality used for?
A. KPI threshold configuration
B. Log parsing
C. Custom fields and tagging
D. KPI normalization

Q8: What is a valid reason to escalate a Notable Event in ITSI?
A. The KPI threshold changes automatically
B. The event was triggered by a low-severity correlation
C. The issue remains unresolved after a defined time window
D. The service associated with the event is currently suppressed

Q9: Where do users typically manage and take action on Notable Events in ITSI?
A. Episode Review Dashboard
B. KPI Editor
C. Deep Dive Panel
D. Field Extractor

Q10: Which of the following actions can be taken on a Notable Event in the management phase?
A. Edit the KPI base search directly
B. Run a search filter against correlation search SPL
C. Schedule the event for historical reprocessing
D. Acknowledge, assign, escalate, or suppress the event

SPLK-3002 Access Control

Access control in ITSI functions as a dual-purpose tool. From a security perspective, it protects sensitive infrastructure metrics and critical configuration files from unauthorized access or accidental modification. From an operational clarity perspective, it ensures that teams remain focused on the systems they manage without being overwhelmed by irrelevant data from other departments. This dual role is fundamental to maintaining both the integrity of the platform and the productivity of the users.

1. Why Access Control Matters in ITSI

The core philosophy of access control is managing exactly who can perform specific actions within the environment. By defining clear boundaries, the system ensures that users only see data relevant to their roles,

which is critical for maintaining focus in high-pressure operations. Furthermore, this structure protects production-critical configurations from accidental edits by unauthorized personnel. It allows multiple operations teams to work in parallel without the risk of one team inadvertently disrupting the workflows or service definitions of another, fostering a stable and accountable environment.

2. Access Control Mechanisms in ITSI

While ITSI utilizes the standard Splunk Role-Based Access Control (RBAC) model, it introduces specialized, service-focused mechanisms to handle the nuances of modern IT environments.

2.1 Teams

Teams serve as the logical connective tissue in ITSI, linking users to specific services, notable events, and dashboards. For instance, a "Web Ops" team would be granted ownership of web-tier services and would receive alerts only for those systems. Similarly, a "DB Team" would focus exclusively on database performance. This grouping ensures that dashboards and Glass Tables are filtered to reflect only the assets a specific team is responsible for managing, making the interface user-specific and highly organized.

2.2 Roles and Capabilities

ITSI leverages standard Splunk roles such as `admin`, `power`, and `user`, while allowing for specialized custom roles defined by specific capabilities. These capabilities include viewing or editing services, managing Key Performance Indicators (KPIs), and creating Glass Tables. The `itsi_admin` role possesses full environmental access, while the `itsi_team_lead` is restricted to editing only their assigned team's services. The `itsi_viewer` role is designed for read-only observation, allowing stakeholders to monitor health without the risk of altering configurations.

2.3 Service-Level Restrictions

For environments requiring high precision, ITSI offers granular control at the individual service and KPI levels. Administrators can determine exactly who can modify or even view specific metrics. Such restrictions are vital for protecting production-critical services, enabling a structure where junior staff can observe system health without the authority to alter underlying logic or thresholds.

3. Common ITSI Roles

The administrative structure of ITSI typically follows a tiered permission model to ensure functional separation. The ITSI Admin occupies the highest tier, possessing unrestricted access to all features, global settings, and system configurations. The Service Owner acts as the functional lead for specific domains, holding the authority to manage services, KPIs, and alert configurations for their assigned systems. Finally, the Viewer role provides read-only access to dashboards and Glass Tables, designed for stakeholders who need to monitor high-level health scores without having the permission to modify the underlying ITSI environment.

4. Best Practices for Access Control

To maintain a scalable security posture, organizations should prioritize inherited roles, extending existing Splunk RBAC structures into ITSI to simplify management and maintain consistency. Periodic audits are essential; administrators should review user roles every few months to remove access for users who have changed responsibilities and to update team-service mappings as organizational structures shift. Ultimately, ITSI teams should mirror real-world structures—such as development teams managing their own applications while ops teams manage shared infrastructure—to improve ownership and communication.

5. Glass Table Access Control

Visualization access is a critical component of data confidentiality. Glass Table access can be restricted by team or role, ensuring that sensitive visualizations—such as those displaying executive-level KPIs or production environment specifics—are only visible to authorized personnel. This prevents unauthorized users from viewing high-level business metrics that fall outside their operational scope, enforcing both data confidentiality and operational boundaries.

6. Notable Event Ownership Routing

To enhance incident accountability, ITSI utilizes routing rules based on service ownership and entity mappings. This ensures that Notable Events are automatically directed to the correct team without manual filtering. By aligning alerts with the people responsible for the specific services involved, ITSI significantly reduces cross-team noise and ensures that incidents are addressed by the appropriate subject matter experts.

7. Access Control Audit via ITSI Audit Dashboards

For compliance and troubleshooting, ITSI provides audit dashboards that track "who, what, and when" regarding configuration changes. These dashboards specifically record modifications to services, KPIs, and thresholds. This level of visibility provides a transparent trail essential for security compliance and internal audits, allowing administrators to quickly identify the source of a misconfiguration during post-mortem analysis.

These security mechanisms create a scalable and secure posture, ensuring that data integrity is maintained as the environment grows. Once access is secured, the system focuses on the intelligent processing of the data itself via Aggregation Policies.

8. Access Control Practice Question

Q1: What is the main purpose of access control in ITSI?
A. To automatically suppress alerts across environments
B. To enable KPI threshold tuning based on user roles
C. To limit what users can see and do based on their role or team
D. To calculate service health scores using multiple teams

Q2: Which of the following is NOT a typical use for ITSI Teams?
A. Controlling Notable Event visibility
B. Assigning ownership of services
C. Managing Splunk index retention policies
D. Filtering access to dashboards

Q3: What is the primary capability of the `itsi_team_lead` role in ITSI?
A. Modify global correlation searches
B. Manage services and KPIs for their assigned team
C. Edit Splunk indexes and retention settings
D. Full system-wide administrative control

Q4: Why is aligning ITSI teams with real-world organizational structure considered a best practice?
A. It enforces license compliance automatically
B. It reduces dashboard load time
C. It increases correlation search frequency
D. It improves service ownership and operational clarity

Q5: Which of the following is an example of a service-level restriction in ITSI?
A. Defining KPIs using summary indexing
B. Automatically escalating suppressed Notable Events
C. Limiting KPI visibility to certain users or roles
D. Auto-closing events after 24 hours of inactivity

Q6: Why should access control configurations be reviewed periodically?
A. To reset license allocation quotas
B. To balance KPI weights across dashboards
C. To ensure the right users maintain the correct permissions
D. To increase correlation search accuracy

Q7: Which role in ITSI provides full administrative access across all services and configurations?
A. itsi_admin
B. dashboard_maintainer
C. itsi_user
D. itsi_team_viewer

Q8: How can dashboards and Glass Tables be restricted in visibility based on team membership?
A. Assign each user a custom event type
B. Configure the correlation search to update dashboard access
C. Set cron schedules per dashboard
D. Link dashboards to specific ITSI teams

Q9: What is the main characteristic of the `itsi_viewer` role?
A. Has read-only access to dashboards and services
B. Can create new KPIs
C. Can assign ownership to other teams
D. Can manage correlation searches

Q10: What is the main characteristic of the `itsi_user` role?
A. Full admin access to all ITSI features
B. Can only view dashboards and KPIs without making changes
C. Can create and manage services, KPIs, and Notable Events within their team
D. Can only manage Splunk index configurations

SPLK-3002 Aggregation Policies

In high-volume IT environments, a single root cause can trigger a deluge of alerts. Aggregation Policies are the mechanism that transforms this raw, high-volume data into actionable intelligence. By grouping related symptoms into consolidated episodes, ITSI helps operations teams move beyond "alert fatigue" toward a structured incident response, ensuring that the signal is never lost in the noise.

1. What Are Aggregation Policies?

The core function of an Aggregation Policy is consolidation. In a scenario where a database failure might generate hundreds of individual alerts, these policies tell the system to group similar events occurring within a specific timeframe into a single "Episode." This reduction in noise allows teams to avoid duplication and focus on the overarching issue rather than being distracted by a sea of individual, repetitive notifications.

2. Core Components of an Aggregation Policy

Technical precision in aggregation is achieved through four primary components:

2.1 Filtering Conditions

Filters determine which events are eligible for aggregation. Rules can be set based on severity, service name, entity, or custom fields. For example, a policy might be configured to only aggregate events where the severity is "Critical" and the service is the "Checkout API."

2.2 Aggregation Fields

These fields define the criteria for "similarity." By grouping events based on shared attributes—such as the same host (entity), the same KPI name, or a shared application source—the system can merge dozens of alerts from the same server into one logical incident.

2.3 Time Windows and Priority Rules

Temporal requirements define how close together events must occur to be grouped. If several events occur within a defined 10-minute window, they are treated as part of the same issue. Priority rules then determine the final severity of the episode; the system can be set to inherit the status of the most severe included event, ensuring that if one event is "Critical," the entire episode is prioritized as such.

3. Advanced Features

To further streamline response, aggregation policies support drill-down searches, allowing analysts to view the original raw events and logs from within the Episode Review interface. Auto-close settings can be configured to tidy up the environment once conditions return to normal or no new events are added, while workload prioritization ensures that high-impact business services are always highlighted at the top of the analyst's queue.

4. Best Practices for Aggregation Policies

Effectiveness requires a balance to avoid over-aggregation. Grouping all web server alerts together is a mistake if they belong to different services; aggregation should be tailored to specific monitoring goals. Policies must be reviewed regularly to adjust time windows and severity rules, ensuring the alerting remains accurate as the infrastructure evolves.

5. Where Aggregation Fits in the Notable Event Lifecycle

Aggregation is strictly a "post-processing" activity. It occurs *after* a correlation search or threshold breach has already generated a Notable Event. It sits between the initial detection of an issue and the display of that issue in the Episode Review interface. Its role is not to find problems, but to manage and organize the alerts generated by other detection mechanisms.

6. Differentiating Aggregation Policies from Correlation Searches

Understanding the distinction between detection and organization is vital for implementation success. Correlation Searches act as the "detectors"; they use logs, metrics, and custom SPL to find patterns and generate Notable Events. Aggregation Policies act as the "managers"; they take those generated events and organize them into episodes. In short: Correlation finds the problem, while Aggregation organizes the response.

By automating the management of event volume, ITSI significantly improves operational efficiency. This organized approach to known issues is complemented by the proactive identification of subtle patterns through Anomaly Detection.

7. Aggregation Policies Practice Question

Q1: What is the main benefit of using aggregation policies in ITSI?
A. They reduce alert noise by grouping related Notable Events
B. They suppress all KPI-based alerts
C. They increase the indexing rate during maintenance
D. They allow you to build Glass Tables automatically

Q2: Which of the following is a component of an aggregation policy that defines similarity between events?
A. Severity color mapping
B. Deep Dive link
C. Aggregation fields
D. Entity tagging

Q3: What does the time window in an aggregation policy control?
A. How often entity models are retrained
B. Whether events include historical logs
C. How close in time events must be to be grouped
D. When KPI thresholds are reset

Q4: In an aggregation policy, what determines the severity of the resulting grouped event?
A. The average of all KPI health scores

B. The time of day
C. The number of events generated per hour
D. The configured priority rules

Q5: Which use case best illustrates the value of auto-close settings in aggregation policies?
A. Automatically closing grouped events when no new events are added after a set time
B. Assigning the most severe tag to an incident
C. Escalating events to executive summary views
D. Linking services to dashboards

Q6: How can aggregation policies help prioritize business services during an incident?
A. By sending events to external APIs
B. By archiving old events
C. By disabling entity suppression
D. By ranking events based on service impact

Q7: What is a recommended best practice when configuring aggregation fields?
A. Avoid using entity or service fields
B. Use the same field across all policies to ensure uniformity
C. Tailor the fields based on the alerting context
D. Always group events by index

Q8: What can you view by drilling down into a grouped Notable Event?
A. The list of users assigned to the KPI
B. The correlation search that created the service
C. All original raw events and related search results
D. The latest configuration changes to dashboards

Q9: Why should you review and adjust aggregation policies regularly?
A. To replicate event routing across clusters
B. To ensure grouping logic remains effective and tuned
C. To deactivate inactive Glass Tables
D. To reduce indexer usage

Q10: What is a risk of over-aggregating events in ITSI?
A. Event color codes may conflict
B. It prevents event suppression from running
C. It may exceed memory thresholds
D. Real problems may be hidden in excessive grouping

SPLK-3002 Anomaly Detection

Modern IT environments are too dynamic for static thresholds alone. Anomaly Detection represents a shift from reactive monitoring to proactive, machine-learning-driven insights. By learning the "normal" rhythms of a system, ITSI can identify deviations—such as subtle spikes or irregular drops—that might otherwise go unnoticed until they cross a catastrophic threshold.

1. How Anomaly Detection Works

This proactive insight is generated through a three-step process. First, ITSI performs **Historical Behavior Learning**, analyzing past KPI data (typically 14 to 90 days) to build a model of normal behavior. Second, **Real-Time Evaluation** compares incoming data against this learned model. Finally, the system provides **Alerting and Visualization**, updating KPI status or generating Notable Events if the current behavior diverges significantly from the expected pattern.

2. Key Configuration Options

Strategic configuration is essential for model accuracy. The **Learning Window** determines the depth of historical data; more data creates a more robust model. **Sensitivity** controls how aggressively deviations are flagged; it is generally recommended to start with medium sensitivity. Finally, **Retraining Frequency** (daily or weekly) ensures the model adapts to system changes, such as seasonal traffic shifts or new application deployments.

3. Use Cases and Limitations

Anomaly detection excels at catching early warning signs before KPIs cross fixed thresholds and reducing false positives in noisy environments where static limits are too rigid. However, it requires sufficient historical data to be effective and may struggle in extremely unpredictable environments without proper sensitivity tuning. It is intended to complement, not replace, traditional thresholds.

4. What Is an Anomaly Score?

Every detected deviation is assigned an Anomaly Score ranging from 0 to 100. This score is a machine-learned confidence indicator; higher values represent a more significant divergence from the learned baseline. This score can drive KPI status color changes (e.g., green to red) and is used to trigger specific alert rules or action conditions before a hard threshold is ever violated.

5. Split-by Fields and Per-Entity Anomaly Detection

When KPIs use split-by fields (like host or region), ITSI builds independent models for each unique entity. This allows for fine-grained tracking, meaning the system learns the "normal" for Server A independently of Server B. This entity-specific modeling reduces false positives caused by system-wide data aggregation and identifies localized issues affecting only a subset of the environment.

6. Visual Representation in Deep Dives

In ITSI Deep Dives, anomalies are visualized using **Anomaly Bands** (shaded areas representing the expected range) and **Outlier Dots** (individual points where the KPI diverged significantly). These visual cues are essential

for Root Cause Analysis (RCA), allowing operators to see exactly when and by how much a system moved outside its learned normal behavior.

Anomaly Detection adds a layer of intelligence that anticipates issues before they escalate. This proactive capability is further strengthened by Correlation and Multi-KPI Searches, which link multiple data points to identify systemic failures.

7. Anomaly Detection Practice Question

Q1: What is the core function of anomaly detection in ITSI?
A. Automatically adjust index retention periods
B. Delete duplicate events from dashboards
C. Learn normal KPI behavior and flag unusual patterns
D. Schedule all KPI base searches hourly

Q2: What configuration setting determines how much historical data is used to train the anomaly detection model?
A. Entity Lookup Policy
B. Sensitivity
C. Base Search Interval
D. Learning Window

Q3: Which anomaly detection setting controls how aggressively ITSI flags deviations?
A. Sensitivity
B. KPI priority
C. Lookup interval
D. Entity mapping level

Q4: What happens if anomaly detection is applied to a KPI with only one day of data?
A. All values will be flagged as “normal” by default
B. It automatically switches to static thresholding
C. The KPI is ignored by the Service Analyzer
D. The model will not be able to learn accurate baseline behavior

Q5: How can anomaly detection help reduce false positives?
A. It disables KPI threshold coloring
B. It filters out known event types
C. It learns what constitutes normal noise in KPI patterns
D. It disables Notable Event generation

Q6: Which option best describes a situation where anomaly detection is more useful than static thresholds?
A. When using a correlation search based on entity type
B. When KPIs change frequently in unpredictable ways
C. When monitoring disk space with a fixed 90% limit
D. When you want to control dashboard visual spacing

Q7: What is the role of retraining frequency in anomaly detection configuration?
A. It controls when search filters are reapplied
B. It sets the number of Notable Events allowed per hour
C. It determines how often the KPI summary index is rebuilt
D. It defines how frequently the machine learning model is updated

Q8: What limitation must users consider when applying anomaly detection to a noisy KPI?
A. It will ignore time policies
B. It may generate too many false alerts without tuning
C. The KPI must be split-by entity
D. No threshold evaluation will occur

Q9: Why is anomaly detection considered a complement, not a replacement, for static thresholds in ITSI?
A. Because it requires admin privileges to run
B. Because it lacks event tagging capabilities
C. Because both methods provide different kinds of insights
D. Because anomaly detection cannot visualize KPI changes

Q10: What is one benefit of using anomaly detection in environments with seasonal behavior?
A. It filters all Notable Events by default
B. It enforces global thresholds
C. It learns and adjusts to time-based behavior patterns
D. It disables base search acceleration

SPLK-3002 Correlation and Multi KPI Searches

In a complex environment, isolated symptoms rarely tell the whole story. Correlation and Multi-KPI Searches are designed to identify systemic failures by linking disparate data points, providing a holistic view of environmental health that a single metric might miss.

1. Multi-KPI Searches

Multi-KPI searches apply mathematical or logical conditions (such as AND/OR) across several KPIs simultaneously. For instance, an architect might configure an alert to trigger only if CPU usage is over 90% *and* the application error rate exceeds 5%. This ensures that alerts are only generated when a truly systemic issue is present, significantly improving the signal-to-noise ratio.

2. Correlation Search Features

These searches use Splunk's Search Processing Language (SPL) to compare metrics across different services or entities. They can incorporate time-based logic, such as alerting if an API error rate increases within 10

minutes of a database query spike. This enables ITSI to identify cascading failures and behavioral patterns across the entire technology stack.

3. Best Practices and Performance

To ensure system scalability, correlation searches must be optimized. This includes limiting time ranges, using summary indexing where appropriate, and validating that the underlying KPIs are providing reliable data. High-performance logic is essential to prevent these resource-intensive searches from lagging or impacting platform stability.

4. Clarifying Scope and SPL Usage

Correlation searches are a general-purpose detection mechanism; they can incorporate log patterns, metric anomalies, and external system alerts in addition to KPI data. While conceptual logic is often described as "CPU > 90," real-world implementation involves sophisticated SPL querying the `itsi_summary` index. A realistic architectural search would look like this: `index=itsi_summary | stats count by itsi_service_id` or more specifically, utilizing `mstats` to pull from metric summaries for maximum performance.

5. Triggering Actions

Once a correlation search matches a condition, it initiates automated workflows. These actions include generating Notable Events in Episode Review, creating ServiceNow tickets, sending Slack or PagerDuty notifications, or running custom scripts for automated remediation. This creates a direct path from detection to resolution.

6. Distinguishing Correlation Rules from Trend Views

It is critical to differentiate between **Correlation Searches**, which are active alerting logic that generates events, and **Multi-KPI Trend Panels**, which are purely observational monitoring tools for side-by-side comparison. The search is a detector; the trend view is a visualization used by analysts to manually spot correlations.

By improving the signal-to-noise ratio, correlation ensures that every alert is meaningful. These advanced logic structures rely on the fundamental building blocks of the system: Entities and Modules.

7. Correlation and Multi KPI Searches Practice Question

Q1: What is the primary purpose of a correlation search in ITSI?
A. To track KPI value history over a rolling time window
B. To store historical KPI search results
C. To combine multiple KPIs and detect complex patterns or failures
D. To calculate static thresholds for machine learning models

Q2: Which of the following is an example of a Multi-KPI condition?
A. CPU usage > 90
B. error_rate > 5

C. response_time over 100ms
D. CPU > 90 AND Memory > 85

Q3: What feature allows correlation searches to consider the order or timing of KPI events?
A. Entity normalization
B. Time-based logic
C. Base search acceleration
D. Time window filtering

Q4: Why is correlation useful in reducing false positives?
A. It adds delays to all alerts, helping avoid duplicates
B. It narrows the data scope to just one metric
C. It replaces all static thresholds with anomaly models
D. It requires multiple KPIs to trigger an alert, increasing alert reliability

Q5: Before designing a correlation search, what should be ensured?
A. That entity rules are removed from the data model
B. That the dashboard layout is preconfigured
C. That all services are under the same team
D. That KPIs are accurate and properly thresholded

Q6: How can correlation searches help detect cascading failures?
A. By summarizing events into hourly reports
B. By combining performance drops across dependent services
C. By comparing KPI baselines
D. By focusing only on external alerts

Q7: What is the benefit of tagging correlation search results in ITSI?
A. Tags allow KPIs to be split by field
B. Tags prevent alert generation during off-hours
C. Tags categorize alerts and help with filtering and triaging
D. Tags simplify visualization in Glass Tables

Q8: Which scenario best describes when to use a correlation search?
A. When a single KPI value slightly increases
B. When dashboards need real-time color updates
C. When one base search is skipped
D. When multiple related KPIs show anomalies within the same timeframe

Q9: What SPL concept is most commonly used to build a correlation search?
A. Lookup tables
B. Boolean logic (AND, OR, NOT)
C. Transaction commands
D. Base acceleration scheduling

Q10: What is a recommended practice for performance when using Multi-KPI correlation searches?
A. Use summary indexing or scoped searches

B. Avoid filters and allow full data access
C. Run all correlation searches in real time
D. Only apply correlation to one service at a time

SPLK-3002 Entities and Modules

For ITSI to provide meaningful insights, its data must be grounded in real-world infrastructure. Entities provide the necessary granularity for monitoring individual components, while Modules offer the efficiency of pre-built, standardized content for rapid deployment.

1. What Are Entities in ITSI?

An entity is a real, identifiable component of an infrastructure, such as a server, virtual machine, or container. By associating KPI data with specific entities, ITSI can isolate issues to individual components. This allows operators to see if a performance dip is isolated to "Server A" or affecting the entire service, enabling precise troubleshooting and per-entity anomaly detection.

2. Entity Properties and Creation

Entities are defined by standard properties (IP, hostname, OS) and custom metadata (Data Center, Business Unit). They are managed in the **Entity Management UI** (Settings > ITSI > Entity Management) and can be discovered automatically from KPI data or added manually via CSV uploads.

3. What Are Modules in ITSI?

Modules are pre-packaged "starter kits" designed for specific platforms like Linux, AWS, or Docker. They provide a full-stack monitoring solution out of the box, including predefined services, KPIs, Glass Tables, and Aggregation Policies. They drastically reduce configuration time by providing standardized best-practice monitoring logic.

4. Module vs. Service Template

While both offer standardization, their scope differs. A **Module** is a "turnkey solution" for an entire technology stack (including dashboards and searches). A **Service Template** is a "design pattern" or blueprint used specifically to standardize KPIs and thresholds across custom services. Modules are full environments; templates are service blueprints.

5. Configuration and Entity Aliases

Entity Aliases are a critical advanced configuration for environments with inconsistent data sources. They allow one entity to be referenced by multiple field names (e.g., mapping `host` to `machine_name` or `hostname`). This ensures naming consistency and cross-source consistency across the platform, allowing KPI bindings to function correctly regardless of the source data's field naming conventions.

Entities provide granularity, while modules provide the speed required for enterprise-scale monitoring. These components are managed and scaled through Templates and Dependencies.

6. Entities and Modules Practice Question

Q1: What is the primary function of an entity in ITSI?
A. To represent individual infrastructure components like servers or applications
B. To manage access roles within the dashboard
C. To group KPIs into services
D. To store raw log data for correlation

Q2: Which of the following fields would most likely be a custom metadata field for an entity?
A. earliest=-24h@h
B. region=EMEA
C. host=web01
D. index=_internal

Q3: How does ITSI typically discover new entities automatically?
A. From manually configured dashboards
B. Through the Service Analyzer
C. From KPI searches that include split-by fields like host or application
D. By importing them through the role manager

Q4: What is one advantage of using entities in dashboards?
A. They simplify log formatting
B. They eliminate the need for threshold settings
C. They allow dashboards to display entity-specific KPI values
D. They increase license limits

Q5: What is an ITSI module?
A. A license validation tool
B. A role-based access manager
C. A search policy for aggregating logs
D. A pre-built monitoring bundle that includes KPIs, services, and dashboards

Q6: What is a reason for using a module in ITSI?
A. To reassign teams to unrelated services
B. To quickly deploy standardized monitoring for common platforms
C. To manually configure KPIs one by one
D. To customize correlation search priorities

Q7: Which of the following can be used to create entities manually in ITSI?
A. A Splunk base image
B. A correlation search
C. A deep dive template
D. A CSV file upload

Q8: How does entity-level anomaly detection improve alerting?
A. It learns patterns per entity to reduce false positives
B. It filters out all data points below average
C. It only evaluates KPIs for parent services
D. It uses static thresholds to detect change

Q9: What kind of components are typically included in an ITSI module?
A. REST API integrations and firewall settings
B. Access logs and index rotation policies
C. Notable Event dashboards and team chat integrations
D. Services, KPIs, dashboards, and correlation searches

Q10: Why are entities important for KPI tracking in ITSI?
A. They automatically create modules based on service design
B. They provide static global thresholds
C. They increase the retention period of event logs
D. They allow per-component evaluation and visualization of KPI behavior

---

SPLK-3002 Templates and Dependencies

Standardization is the key to managing ITSI at scale. Service Templates ensure consistency across departments, while dependency modeling allows for sophisticated impact analysis across the service tree, ensuring that administrators can manage thousands of services with the same effort as dozens.

1. Using Service Templates

Templates enable rapid scaling by allowing a single definition of KPIs and thresholds to be applied to multiple services. Changes made to a template can be pushed to all linked services automatically. This "define once, inherit everywhere" model ensures that monitoring logic remains consistent across the entire organization, whether monitoring one app or ten identical regional instances.

2. Technical Composition of Templates

A template acts as a reusable blueprint, typically including KPI base searches, threshold rules, health score rules, and time policies. This ensures that every service derived from the template follows the same operational standards and performance expectations.

3. Understanding Service Dependencies

Dependencies define parent-child relationships between services (e.g., a "Checkout Service" depending on a "Database"). Modeling these relationships allows ITSI to perform impact analysis, showing how a failure in a child service propagates upward to affect the health of the parent. This reflects real-world business impact and

supports weighted health scoring, where critical dependencies (like a DB) can be weighted more heavily than non-critical ones.

4. Dynamic Tokenization and Time Policies

To maintain reusability, templates use dynamic tokens such as `$host$`, which are replaced with actual entity names when the template is applied. Templates also incorporate Time Policies (like business hours or maintenance windows), ensuring that all derived services automatically inherit consistent, time-aware threshold behavior without manual configuration.

5. Visualization and Propagation

The **Service Analyzer's Dependency View** is the primary tool for visualizing these relationships and conducting root cause analysis. Health status changes and Notable Events propagate through the service hierarchy, allowing operators to see exactly where a failure originates (the symptom) and how it affects the overarching business service (the impact).

Effective templates and meaningful weighting prevent circular dependencies and ensure that health scores accurately reflect business reality. This logic is finalized through the configuration of Thresholds and Time Policies.

6. Templates and Dependencies Practice Question

Q1: What is the primary benefit of using a service template in ITSI?
A. It allows per-entity anomaly detection across all services
B. It automates threshold evaluation during correlation searches
C. It enforces role-based access control
D. It standardizes KPIs and configurations across multiple services

Q2: In ITSI, what component is inherited when applying a service template to a new service?
A. Raw log event definitions
B. Threshold and scoring configurations
C. Custom Glass Table layout
D. Event Aggregation Policy

Q3: Why is it important to avoid circular dependencies between ITSI services?
A. They increase search load on indexers
B. They cause delays in KPI base search scheduling
C. They prevent modular input configuration
D. They disrupt health score calculations and impact analysis

Q4: What is one valid reason to assign weights to child dependencies in a parent service?
A. To prevent the creation of duplicate services
B. To reduce disk space used by KPIs
C. To reflect the real-world impact of child service health on the parent
D. To adjust base search retention

Q5: A monitoring team wants to deploy the same service design to multiple environments. What should they use?
A. Threshold tuning
B. Deep Dive export
C. Service template
D. Aggregation policy

Q6: What does a service dependency structure allow you to model in ITSI?
A. Physical network topology
B. Splunk index storage patterns
C. Logical relationships and health impact between services
D. License consumption across search heads

Q7: Which of the following is a best practice when designing a reusable service template?
A. Use tokenized KPI names and avoid hardcoded identifiers
B. Bind the template to a specific time window only
C. Hard-code the hostname for all KPIs
D. Include static severity mappings for all time zones

Q8: What is the function of health score rules in a service template?
A. To establish base search acceleration parameters
B. To define indexing quotas for service data
C. To calculate overall service health based on KPI performance
D. To link thresholds with entity suppression logic

Q9: What happens when you update a KPI definition in a service template that is linked to multiple services?
A. The update must be manually applied to each service
B. All linked services automatically inherit the updated KPI
C. Only newly created services inherit the update
D. The change only affects base search retention policies

Q10: How do templates and dependencies work together in ITSI?
A. Templates define UI layout while dependencies generate alerts
B. Templates trigger alert suppression, and dependencies disable KPI scoring
C. Templates store raw events, and dependencies organize indexers
D. Templates scale configuration, and dependencies define service impact relationships

---

SPLK-3002 Thresholds and Time Policies

Thresholds and Time Policies are the tools that transform raw metrics into meaningful health indicators. They provide the necessary context to determine whether a metric value represents a normal state, a warning, or a critical failure within the specific context of the business clock.

1. Types of Thresholds

ITSI supports several models tailored to different behaviors: **Static** thresholds (fixed numbers for predictable metrics like disk usage), **Percentage-Based** (ideal for utilization), **Trend-Based** (focusing on the rate of change), and **Machine Learning** (using historical baselines to adapt to irregular patterns).

2. Strategic Use of Time Policies

Time policies acknowledge that system behavior varies by context. They allow for stricter monitoring during peak business hours and more relaxed rules during nightly maintenance windows or low-activity periods. This prevents false alerts during known heavy-load periods (like backups) and ensures high sensitivity when user activity is at its highest.

3. Configuration Locations and Action Rules

Thresholds are configured within individual KPI settings or the **Service Editor**. Time Policies are managed via the **Time Policy Manager** (Settings > ITSI > Time Policies). These configurations drive the "Action Rules" that determine exactly when a status change should trigger a Notable Event.

4. Triggering Notable Events

The operational flow moves from a threshold breach to a KPI state change (e.g., Normal to Critical). If event generation is enabled for that KPI, this breach triggers a Notable Event. This seamless flow from Metric to Threshold to Event is the backbone of ITSI's monitoring architecture, ensuring a direct link between performance and response.

These features form the backbone of ITSI's context-aware alerting system. To maintain this system, administrators must be proficient in troubleshooting the platform itself.

5. Thresholds and Time Policies Practice Question

Q1: What is the main purpose of a threshold in ITSI?
A. To assign users to different teams
B. To schedule when a search runs
C. To define when a KPI's status should change based on its value
D. To generate Glass Table visualizations

Q2: Which threshold type is most appropriate for a utilization metric like CPU usage?
A. Percentage-based thresholds
B. Static thresholds
C. Machine learning thresholds
D. Trend-based thresholds

Q3: What makes trend-based thresholds useful in monitoring KPIs?
A. They measure error counts across services
B. They prevent anomalies from generating Notable Events

C. They highlight rapid changes in KPI behavior over time
D. They detect gradual improvement in performance

Q4: Why are time policies important in ITSI?
A. They filter out raw log data from base searches
B. They configure Notable Event grouping rules
C. They determine which users can acknowledge Notable Events
D. They allow different thresholds to be used during different times of day

Q5: How does a machine learning threshold work in ITSI?
A. It ignores short-term anomalies to reduce alert fatigue
B. It learns normal patterns from historical data and flags deviations
C. It uses fixed limits based on KPI type
D. It compares KPI values against known static benchmarks

Q6: Which severity level in ITSI represents the most urgent condition?
A. Info
B. Critical
C. High
D. Warning

Q7: What is a valid use case for a static threshold?
A. Defining exact limits for disk usage
B. Observing weekly usage changes
C. Detecting gradual trends in user login time
D. Monitoring fluctuating API error rates

Q8: How do time zones improve the effectiveness of time policies in ITSI?
A. They match thresholds to users' preferred display format
B. They ensure alerts are always in UTC
C. They align threshold rules with the local time of each region
D. They allow KPIs to run faster by limiting data range

Q9: What happens when a KPI crosses a threshold into a "Critical" state?
A. The KPI automatically resets to Normal
B. A Notable Event may be generated
C. The search is paused until reset
D. The threshold is deleted and reset

Q10: What feature allows thresholds to behave differently during specific hours, such as nights or weekends?
A. KPI groups
B. Health score modifiers
C. Time policies
D. Threshold templates

SPLK-3002 Troubleshooting ITSI

Maintaining the health of the ITSI platform is fundamental to ensuring that service insights remain trustworthy. Troubleshooting focuses on identifying and resolving performance bottlenecks, data gaps, and configuration errors that could lead to inaccurate health scores.

1. Common Troubleshooting Domains

Administrators must monitor search performance using the **Search Inspector** to identify execution delays. The **Data Audit dashboards** are vital for spotting data latency or gaps. Furthermore, checking scheduler logs for skipped searches is essential, as search concurrency issues can lead to "stale" KPIs.

2. Specialized KPI States

Identifying the state of a KPI is the first step in debugging environmental issues:

- **Stale:** Data is not arriving on schedule. This is commonly caused by skipped searches, search concurrency overload, or index delays.
- **Invalid:** The base search returns non-numeric or malformed data. Common causes include SPL syntax errors, incorrect `eval` logic, or missing fields in the source data.
- **No Result:** The search runs but returns zero matching events. This is often due to field typos (e.g., `hostnme` vs `hostname`), time range misalignments, or insufficient permissions.

3. Technical Debugging Tools

The `itsi_troubleshooting_toolkit` provides environment diagnostics and health reports. Administrators should leverage the `_internal` logs for system warnings and query the `itsi_summary` index to verify raw KPI results. The `itsi_notable_archive` index is equally vital for forensic analysis, allowing architects to verify if a correlation search ever fired an event.

4. Debugging Correlation Searches

When correlation searches fail to trigger expected events, architects should narrow the query scope using lightweight SPL for testing. Validating live data retrieval can be done by appending `| stats count` or `| head 10` to the base search. Enabling debug logging can reveal hidden issues such as syntax parsing errors, aggregation timeouts, or skipped searches due to role-based permission restrictions.

By following these diagnostic practices, administrators can ensure a stable, efficient, and accurate ITSI environment that provides reliable intelligence for the business.

5. Troubleshooting ITSI Practice Question

Q1: What is the primary purpose of the Search Inspector tool in ITSI troubleshooting?
A. To analyze search execution time and identify delays
B. To configure role-based access to KPIs

C. To tune static thresholds in real-time
D. To deploy updated service templates

Q2: Which index should you examine to investigate skipped searches and search concurrency issues in ITSI?
A. index=itsi_audit
B. index=itsi_events
C. index=_internal
D. itsi_summary

Q3: A KPI is showing “No Result.” Which of the following is the most likely cause?
A. The base search may be pointing to a wrong or empty index
B. The KPI’s correlation search is not enabled
C. The KPI does not have a team assignment
D. The KPI is using too many threshold levels

Q4: What component in ITSI helps you detect stale or incomplete KPI results?
A. itsi_troubleshooting_toolkit
B. Service Template Manager
C. Glass Table Preview
D. ITSI Data Audit Dashboards

Q5: Which of the following is the BEST way to reduce the load from concurrent KPI searches?
A. Set all KPIs to run in real-time mode
B. Use cron scheduling offsets to stagger searches
C. Disable entity filtering
D. Increase the number of correlation searches

Q6: A Glass Table is not displaying KPI data correctly. What is a potential cause?
A. The KPI has a threshold of 'Info' only
B. The base search is returning multiple fields
C. The entity is not mapped to the KPI
D. The table size exceeds the storage limit

Q7: What does the `itsi_troubleshooting_toolkit` provide in ITSI environments?
A. Dashboards, diagnostics, and health checks for ITSI components
B. Machine learning-based anomaly detection
C. Automation for correlation rule creation
D. Performance profiling for Glass Tables

Q8: If Notable Events are not appearing as expected, which troubleshooting step should be prioritized?
A. Check for updates to the Splunkbase
B. Verify correlation searches are enabled and returning results
C. Rebuild the summary index
D. Adjust user roles and capabilities

Q9: Why is documenting service changes and KPI edits important in ITSI troubleshooting?
A. To comply with data retention policies

B. To reduce the number of Notable Events
C. To support root cause analysis and version rollback
D. To improve anomaly detection accuracy

Q10: What is the main risk of failing to troubleshoot threshold misconfigurations in ITSI?
A. It results in unreliable service health scores and false alerts
B. It disables correlation search execution
C. It prevents KPIs from being split by entity
D. It causes KPIs to inherit incorrect time zones

Learning Path & Study Advice

Candidates should begin by understanding the foundational concepts of ITSI, including its architecture and core components. After establishing this base, focus should shift to installation and configuration, ensuring clarity on how the platform is prepared for use. The next stage should emphasize service design and implementation, followed by KPI construction, thresholds, and aggregation logic.

Once these core elements are understood, candidates should explore operational aspects such as event management, anomaly detection, and visualization through glass tables and deep dives. Finally, attention should be given to governance and troubleshooting, ensuring the ability to manage access and resolve issues effectively. A consistent focus on how different components interact will support a deeper and more practical understanding.

Who This PDF Is For

This document is intended for IT professionals who are preparing for the SPLK-3002 certification and are responsible for administering Splunk IT Service Intelligence environments. It is suitable for system administrators, IT operations engineers, and observability practitioners with prior Splunk experience. Individuals seeking to develop structured knowledge in service modeling, KPI management, event handling, and ITSI administration will benefit most from this material.

Call To Action

This document provides an overview of structured learning and certification preparation approaches. For learners seeking clear knowledge organization, guided study planning, and exam-focused practice resources, AAAdemy offers a comprehensive platform to support independent and effective learning.

Explore additional training materials, study guidance, and practice resources at:

Splunk SPLK-3002 Splunk IT Service Intelligence Certified Admin Certification Training Course - AAAdemy

Online Flashcards (Quizlet):

https://quizlet.com/user/AAAdemy/folders/splk-3002-splunk-it-service-intelligence-certified-admin?i=6zfa5t&x=1xqt

Attachment : Answers by Knowledge Point

Introducing ITSI Practice Question
A1: Answer: A
Explanation: ITSI is designed to provide a service-centric view of IT systems by linking low-level technical metrics (such as CPU, memory, error rates) to high-level business services (like “Checkout System” or “Email Notification”). This helps organizations understand how technical issues impact service delivery.

A2: Answer: D
Explanation: A Service in ITSI is a logical grouping of technical components that work together to support a business function. It contains KPIs and calculates a real-time health score to indicate how well the service is performing.

A3: Answer: C
Explanation: A KPI (Key Performance Indicator) in ITSI is a metric derived from a Splunk search that reflects the performance of part of a service. Thresholds can be configured for each KPI to determine when alerts (Notable Events) should be triggered.

A4: Answer: C
Explanation: A Glass Table is a highly visual, customizable dashboard in ITSI that uses icons, shapes, and animations to represent service health. It is particularly useful for executive and NOC views that need at-a-glance insights.

A5: Answer: A
Explanation: A Notable Event in ITSI is a structured, actionable alert triggered when a KPI breaches a threshold or when a correlation search detects a defined condition. These events are managed in the Episode Review dashboard.

A6: Answer: C
Explanation: A Deep Dive provides an interactive, timeline-based view of KPIs that allows users to analyze trends, correlate metrics, and investigate the root cause of incidents. It is ideal for operational troubleshooting and post-incident review.

A7: Answer: D
Explanation: A Service in ITSI is built by combining relevant KPIs like CPU, memory, and error rate into a single, high-level health score, offering a unified view of business impact.

A8: Answer: B
Explanation: When a KPI breaches a defined threshold (like Critical), ITSI may automatically generate a Notable Event, depending on its configuration. This event is used to alert operations teams and may be grouped via Aggregation Policies.

A9: Answer: C
Explanation: A Glass Table is a dashboard interface that displays service and KPI health visually using icons, color states, and drilldowns. Users can click these indicators to explore deeper data views such as Deep Dives.

A10: Answer: A
Explanation: KPIs are the metrics that measure parts of a service. Their status—based on thresholds—feeds directly into ITSI's health score calculation, which represents overall service condition.

Glass Tables Practice Question
A1: Answer: C
Explanation: A Glass Table is not just a standard dashboard; it allows users to visually represent services using icons, shapes, and animations, and link these to real-time KPI data, enabling clickable and interactive monitoring.

A2: Answer: A
Explanation: Glass Tables are used to create real-time visual overviews of systems and services, including live KPI status and drill-down actions. They are not designed for search creation or batch job execution.

A3: Answer: A
Explanation: When designing for executives, the dashboard should be minimal and business-focused, avoiding technical overload and instead showing the high-level health of key services.

A4: Answer: C
Explanation: Glass Tables support color-coded visual elements, where the color (e.g., green/yellow/red) reflects the real-time threshold status of a KPI.

A5: Answer: D
Explanation: Drill-down actions enable user interaction by allowing clicks on Glass Table elements to open Deep Dives, dashboards, or search results, making Glass Tables highly functional.

A6: Answer: D
Explanation: Animations (like blinking or flowing arrows) are helpful in NOC scenarios to highlight data flow and attention areas, making real-time issues more visible.

A7: Answer: B
Explanation: Shapes or icons in Glass Tables can be linked to KPIs with split-by fields, allowing the same visual element to represent per-entity metrics like CPU usage by host.

A8: Answer: A
Explanation: Animations such as blinking, fading, or directional movement help operators quickly detect system status changes, especially on large wall displays.

A9: Answer: A
Explanation: Visual components in Glass Tables are bound to KPIs, allowing them to dynamically change color or behavior based on the real-time status of the underlying metric.

A10: Answer: B
Explanation: In Glass Tables, red indicates a critical threshold breach. If a database icon is red, its linked KPI is in a critical state, suggesting performance or availability issues.

Managing Notable Events Practice Question
A1: Answer: B
Explanation: A Notable Event is created in ITSI when a KPI crosses a threshold or a correlation search matches a pattern. It is actionable and signifies a condition needing investigation.

A2: Answer: A
Explanation: Aggregation Policies define how multiple related Notable Events are grouped together, helping reduce duplication, confusion, and alert fatigue.

A3: Answer: C
Explanation: Event Suppression allows you to define rules to prevent Notable Events from being created during known periods of activity such as maintenance or testing.

A4: Answer: B
Explanation: In the Triggering phase, a KPI threshold is breached or a correlation search condition is satisfied, initiating the creation of a Notable Event.

A5: Answer: A
Explanation: If a mission-critical service is severely affected, the Notable Event should be prioritized as Critical, ensuring it's handled immediately by the relevant team.

A6: Answer: C
Explanation: Workflows allow automation of actions such as sending notifications or creating incident tickets, enabling teams to respond to events consistently and quickly.

A7: Answer: C
Explanation: Custom fields and tags are used to enrich events with metadata, helping teams filter, categorize, and route events effectively.

A8: Answer: C
Explanation: Escalation happens when an event hasn't been resolved within a defined time, indicating it requires additional attention or higher-level intervention.

A9: Answer: A
Explanation: The Episode Review Dashboard is the central UI where users view, group, assign, and take action on Notable Events.

A10: Answer: D
Explanation: In the management phase, users can acknowledge, assign, escalate, or suppress Notable Events to keep incident handling structured and efficient.

Investigating Issues with Deep Dives Practice Question
A1: Answer: C
Explanation: A Deep Dive is a time-based, visual interface in ITSI that allows users to explore KPI behavior over time, correlate with Notable Events, and identify root causes.

A2: Answer: C
Explanation: The Event Timeline is overlaid in the Deep Dive interface and shows exactly when Notable Events occurred, allowing comparison with changes in KPI values.

A3: Answer: D
Explanation: Deep Dives offer interactive controls such as zooming and panning, enabling users to analyze specific time intervals around performance changes or incidents.

A4: Answer: A
Explanation: Users can drill down from a KPI line to explore underlying data, view logs, or open related Notable Events or dashboards, which makes investigation efficient.

A5: Answer: D
Explanation: Visualizing multiple KPIs on the same timeline allows you to spot trends and relationships, such as whether CPU usage and error rate increased together.

A6: Answer: B
Explanation: Deep Dives are commonly used in post-incident reviews to replay what happened, identify the root cause, and create documentation for future prevention.

A7: Answer: A
Explanation: A time-based correlation where one KPI spikes just before another suggests a causal relationship, indicating the root cause might be the database delay.

A8: Answer: B
Explanation: Deep Dives can be used proactively to observe how KPIs behave after changes or deployments, helping detect regressions or improvements early.

A9: Answer: C
Explanation: Deep Dives provide an interactive workspace where users can view and explore several KPIs across a shared timeline, with dynamic controls like zoom and drilldown.

A10: Answer: D
Explanation: Troubleshooting with Deep Dives begins by correlating KPI behaviors and identifying when and how different metrics changed—this is essential for finding the root cause.

Installing and Configuring ITSI Practice Question
A1: Answer: A
Explanation: ITSI requires Splunk Enterprise version 8.0 or higher to ensure compatibility with its features and performance requirements.

A2: Answer: B
Explanation: ITSI is distributed as a `.spl` file, which is Splunk's application packaging format used for installation via the UI or CLI.

A3: Answer: D
Explanation: The `itsi_summary` index is used to store summarized KPI results, including health scores for services and KPIs.

A4: Answer: C
Explanation: ITSI runs many background searches and stores large volumes of data in indexes like `itsi_summary`, requiring high disk throughput for efficiency.

A5: Answer: B
Explanation: The Health Check dashboard helps administrators verify whether the environment is properly configured, including permissions, indexes, and search status.

A6: Answer: A
Explanation: After installation, it's important to create ITSI Teams, configure access control, and set up data pipelines for KPIs and services.

A7: Answer: A
Explanation: The ITSI license is separate from the Splunk Enterprise license and must be installed to activate all features of the ITSI platform.

A8: Answer: C
Explanation: You can install ITSI by placing the `.spl` file in the `apps` directory and then restarting Splunk to complete the app installation.

A9: Answer: A
Explanation: Modular inputs in ITSI are essential for collecting logs, triggering background processes, and enabling functionality such as alert actions and service templates.

A10: Answer: D
Explanation: The `itsi_tracked_alerts` index is where Notable Events are stored, including metadata about triggered KPIs or correlation rules.

Designing Services Practice Question
A1: Answer: D

Explanation: A Service in ITSI is a logical model of a business or technical function. It groups together KPIs, thresholds, and dependencies to monitor and evaluate performance in a meaningful way.

A2: Answer: A
Explanation: Templates provide a way to quickly deploy and manage multiple similar services with consistency in KPIs, thresholds, and structure. They are highly recommended for scalability.

A3: Answer: D
Explanation: Splitting large systems into smaller, focused services makes them easier to manage, analyze, and troubleshoot. Each service should reflect a meaningful component.

A4: Answer: A
Explanation: A KPI base search is a Splunk Processing Language (SPL) query that retrieves and aggregates raw data to generate KPI metrics used within services.

A5: Answer: B
Explanation: Thresholds help define the KPI's status levels such as Normal, Warning, and Critical, which are used to calculate service health and trigger alerts.

A6: Answer: B
Explanation: Dependencies in ITSI are modeled using parent-child relationships, where child service health can influence the health score of a parent service.

A7: Answer: D
Explanation: A service such as "Payment Gateway" includes KPIs, thresholds, and dependencies that represent a real business or technical function, making it ideal for service modeling in ITSI.

A8: Answer: A
Explanation: Documentation helps teams understand what the service does, who owns it, and how to maintain or troubleshoot it in the future.

A9: Answer: A
Explanation: Split-by fields allow a single KPI to track values across multiple entities, such as servers or locations, helping with detailed per-entity monitoring.

A10: Answer: C
Explanation: Service dependencies enable the health score of a parent service to be influenced by its child services, giving a more accurate picture of overall system health.

Data Audit and Base Searches Practice Question
A1: Answer: A
Explanation: A base search in ITSI defines how data is retrieved from Splunk using SPL. It is the foundation for KPIs, services, and dashboards.

A2: Answer: B
Explanation: `itsi_data_integrity` is a built-in utility in ITSI that detects if KPI searches are returning missing, stale, or incomplete data.

A3: Answer: A
Explanation: Split-by fields allow a single KPI to provide per-entity data (e.g., per host, application), which enhances the granularity of service monitoring.

A4: Answer: C
Explanation: Search Inspector shows detailed information on search performance, including duration, skipped results, and resource usage.

A5: Answer: C
Explanation: Offsetting search schedules avoids simultaneous execution of multiple searches, which improves system performance and reliability.

A6: Answer: D
Explanation: Summary indexing allows you to store the result of a heavy search and re-use it, improving performance and reducing resource usage.

A7: Answer: D
Explanation: Search scheduling determines the execution frequency of a base search, using either cron expressions or real-time configuration.

A8: Answer: A
Explanation: Filters in base searches are used to exclude irrelevant data (e.g., test environments) and ensure accurate KPI measurement.

A9: Answer: D
Explanation: Audit dashboards allow users to track KPI performance metrics, helping identify where data is missing or searches have failed.

A10: Answer: C
Explanation: Using a large time range causes the search to scan more data, which can slow down performance and overload indexers unnecessarily.

Implementing Services Practice Question
A1: Answer: D
Explanation: The first step in implementing a service is to create the service using the Service Editor, providing it with a clear name and optional description before KPIs or dependencies are added.

A2: Answer: C
Explanation: Parent-child relationships are used to define dependencies, where the health status of a child service can influence the health of the parent, helping model real-world impact.

A3: Answer: A
Explanation: After creating a service, the next step is to add KPIs, defining the base search, thresholds, and importance weight for each KPI used in health scoring.

A4: Answer: B
Explanation: Assigning services to a team enables better access control, accountability, and delegation of monitoring responsibilities to relevant operations or application groups.

A5: Answer: A
Explanation: Thresholds define KPI states, such as "Normal", "Warning", or "Critical", allowing ITSI to assess the health score and generate Notable Events accordingly.

A6: Answer: C
Explanation: Machine Learning-based thresholds are ideal for complex or inconsistent KPIs because they automatically learn a baseline and flag outliers, even if you don’t know the ideal values.

A7: Answer: B
Explanation: Time policies allow you to define different thresholds for different times of day, such as being stricter during business hours and more tolerant at night.

A8: Answer: B
Explanation: When one service, such as Authentication, goes down and causes another (e.g., User Login) to fail, it's ideal to use service dependencies to reflect this impact hierarchy.

A9: Answer: C
Explanation: Each KPI must include a base SPL search, thresholds for evaluation, an importance score (for health score weighting), and optionally a split-by dimension for granularity.

A10: Answer: A
Explanation: Dynamic thresholds are based on relative trends or percentage increases/decreases over time, making them suitable for metrics with natural fluctuations.

Thresholds and Time Policies Practice Question
A1: Answer: C
Explanation: Thresholds define when a KPI status changes—such as from Normal to Warning or Critical—based on specific value conditions. This helps drive alerts, health scores, and visibility.

A2: Answer: A
Explanation: Percentage-based thresholds are designed for KPIs that naturally use percentage scales, such as CPU or memory usage.

A3: Answer: C
Explanation: Trend-based thresholds evaluate how quickly a metric changes, which is useful for identifying sudden shifts or degradation.

A4: Answer: D
Explanation: Time policies allow administrators to configure different threshold values depending on time windows like business hours vs. off-hours.

A5: Answer: B
Explanation: Machine learning thresholds analyze historical KPI data to understand normal patterns and detect anomalies without needing predefined limits.

A6: Answer: B
Explanation: "Critical" is the highest severity level used to indicate the most urgent KPI condition requiring immediate action.

A7: Answer: A
Explanation: Static thresholds are ideal for metrics with well-defined, consistent limits like disk space capacity.

A8: Answer: C
Explanation: Time zones ensure time policies reflect each region’s local time, making threshold application more relevant across global environments.

A9: Answer: B
Explanation: Threshold violations such as entering a "Critical" state can trigger the creation of a Notable Event for team response.

A10: Answer: C
Explanation: Time policies let administrators set different threshold levels for different time windows, helping adapt to normal operational variances.

Entities and Modules Practice Question
A1: Answer: A
Explanation: Entities in ITSI are used to represent and track individual infrastructure components, allowing precise monitoring, filtering, and anomaly detection.

A2: Answer: B
Explanation: Custom metadata like `region=EMEA` can be added to entities for filtering, grouping, and access control.

A3: Answer: C
Explanation: When KPIs use split-by fields like “host,” ITSI can detect new values and automatically create entities for them.

A4: Answer: C
Explanation: Entities allow dashboards to be filtered and visualized on a per-entity basis, increasing clarity and value for specific users.

A5: Answer: D
Explanation: A module is a pre-configured package that helps users deploy monitoring for a technology or environment quickly using predefined KPIs and dashboards.

A6: Answer: B
Explanation: Modules enable fast deployment and consistency by providing out-of-the-box KPIs, dashboards, and services for known technologies.

A7: Answer: D
Explanation: Entities can be manually created using the ITSI UI or imported via CSV, especially useful for preloading entity details.

A8: Answer: A
Explanation: Entity-level anomaly detection builds unique baselines per entity, improving alert accuracy and reducing false alerts.

A9: Answer: D
Explanation: ITSI modules include services, KPIs, dashboards, and prebuilt searches to provide a complete monitoring framework.

A10: Answer: D
Explanation: Entities let you evaluate KPI behavior for specific components, enabling finer-grained analysis and anomaly detection.

Templates and Dependencies Practice Question
A1: Answer: D
Explanation: Service templates provide a reusable structure for KPIs, thresholds, and scoring logic, which ensures consistency and scalability across multiple services.

A2: Answer: B
Explanation: When a template is applied, the new service inherits thresholds, KPI base searches, and scoring configurations from the template.

A3: Answer: D
Explanation: Circular dependencies create logical loops that can break health score propagation and distort service impact analysis.

A4: Answer: C
Explanation: Assigning weights ensures that services with greater importance to the parent's functionality contribute more significantly to the health score.

A5: Answer: C
Explanation: A service template allows teams to apply a standardized set of configurations to multiple services quickly and uniformly.

A6: Answer: C
Explanation: Dependencies allow you to define parent-child service relationships that mirror operational hierarchies and help in impact visualization.

A7: Answer: A
Explanation: Using tokenized variables increases the flexibility and reusability of templates across teams and environments.

A8: Answer: C
Explanation: Health score rules determine how the performance of KPIs is aggregated to reflect the health status of a service.

A9: Answer: B
Explanation: Updates to templates are automatically pushed to all services that use that template, maintaining consistency.

A10: Answer: D
Explanation: Templates provide scalable, reusable service definitions, while dependencies create a hierarchy that models operational relationships and health score propagation.

Anomaly Detection Practice Question

A1: Answer: C
Explanation: Anomaly detection in ITSI uses machine learning to analyze historical KPI trends and detect behavior that deviates from the learned baseline.

A2: Answer: D
Explanation: The Learning Window controls how many days or weeks of historical KPI data ITSI uses to build its model of expected behavior.

A3: Answer: A
Explanation: Sensitivity determines how strict or lenient the anomaly detection model is when identifying outliers in KPI data.

A4: Answer: D
Explanation: Anomaly detection requires a sufficient amount of historical data to establish a reliable baseline. One day is typically too little for accuracy.

A5: Answer: C
Explanation: Unlike static thresholds, anomaly detection models learn typical fluctuations and can avoid triggering alerts for normal behavior variations.

A6: Answer: B
Explanation: In dynamic or unpredictable environments, anomaly detection adapts to trends more effectively than fixed limits.

A7: Answer: D
Explanation: Retraining frequency specifies how often ITSI updates the machine learning model with new KPI data.

A8: Answer: B
Explanation: Highly variable (noisy) KPIs can confuse the model and lead to excessive alerting unless sensitivity or smoothing is adjusted.

A9: Answer: C
Explanation: Anomaly detection provides pattern-based alerts, while static thresholds offer specific condition-based logic. They work best together.

A10: Answer: C
Explanation: Anomaly detection can adapt to seasonal trends by learning what is normal during different times of day or year.

Correlation and Multi KPI Searches Practice Question

A1: Answer: C
Explanation: Correlation searches are designed to evaluate the combined behavior of multiple KPIs, helping detect systemic or complex issues that a single KPI alone may not reveal.

A2: Answer: D
Explanation: A Multi-KPI condition evaluates multiple metrics at once using logical operators. "CPU > 90 AND Memory > 85" is a classic example.

A3: Answer: B
Explanation: Time-based logic enables correlation rules to specify sequences or simultaneous occurrences of anomalies across KPIs, enhancing situational analysis.

A4: Answer: D
Explanation: By correlating multiple KPIs, the system only generates alerts when multiple conditions are met, improving alert relevance and reducing noise.

A5: Answer: D
Explanation: Without accurate KPIs and meaningful thresholds, any correlation search will produce poor-quality results—this foundational data must be solid first.

A6: Answer: B
Explanation: Correlation logic lets you track when multiple failures occur in a sequence, such as one service failing and then impacting others, which is known as a cascading failure.

A7: Answer: C
Explanation: Adding tags (like “database”, “latency”, “critical”) allows teams to quickly identify alert types, filter dashboards, and prioritize incidents accordingly.

A8: Answer: D
Explanation: Correlation searches are ideal for identifying patterns involving multiple KPIs, especially when they occur within a short time window.

A9: Answer: B
Explanation: Correlation searches rely heavily on logical operators like AND, OR, and NOT in SPL queries to relate multiple KPIs and conditions.

A10: Answer: A
Explanation: For performance, you should optimize correlation searches using techniques like summary indexing, scoped filters, and limited time windows.

Aggregation Policies Practice Question
A1: Answer: A
Explanation: Aggregation policies are used to reduce alert noise by combining similar Notable Events into a single grouped event, improving clarity and response efficiency.

A2: Answer: C
Explanation: Aggregation fields determine which attributes (e.g., service, host, KPI) must match for events to be grouped together.

A3: Answer: C
Explanation: The time window defines the allowed time range within which similar events are grouped into a single Notable Event.

A4: Answer: D
Explanation: Priority rules define how the final severity of the grouped event is set, often based on the highest-severity included event.

A5: Answer: A
Explanation: Auto-close allows a grouped event to close automatically when no additional events are added for a configured period.

A6: Answer: D
Explanation: Aggregation policies can prioritize grouped events based on business importance, allowing critical incidents to be surfaced first.

A7: Answer: C
Explanation: You should customize aggregation fields based on the monitoring goal—different services may require grouping by different attributes.

A8: Answer: C
Explanation: Drill-downs let users see all original Notable Events, raw logs, and search results for deeper investigation.

A9: Answer: B
Explanation: As your environment changes, aggregation logic must be adjusted to avoid missed groupings or excessive event noise.

A10: Answer: D
Explanation: Over-aggregation can obscure real issues by combining unrelated events, making root cause detection more difficult.

Access Control Practice Question
A1: Answer: C
Explanation: Access control ensures that users only have access to the services, dashboards, and KPIs relevant to their responsibilities, improving both security and clarity.

A2: Answer: C
Explanation: Teams in ITSI manage access to services and events, but they do not manage Splunk's index-level settings.

A3: Answer: B
Explanation: The `itsi_team_lead` role allows users to create and manage services and KPIs associated with their own team.

A4: Answer: D
Explanation: When teams reflect actual org structures, ownership and accountability for services and incidents are clearer and more effective.

A5: Answer: C
Explanation: Service-level restrictions allow administrators to control which users can view or modify specific KPIs or services.

A6: Answer: C
Explanation: Regular audits of access help maintain secure and accurate role assignments as users change teams or responsibilities.

A7: Answer: A
Explanation: The `itsi_admin` role grants unrestricted access to all ITSI features, including service configuration, KPI creation, and alert management.

A8: Answer: D
Explanation: Dashboards can be associated with ITSI teams to ensure only relevant users can access and view them.

A9: Answer: A
Explanation: The `itsi_viewer` role is designed for users who need visibility into services and metrics but should not make configuration changes.

A10: Answer: C
Explanation: The `itsi_user` role typically allows users to create and manage services, KPIs, and Notable Events within the scope of their assigned team.

Troubleshooting ITSI Practice Question
A1: Answer: A
Explanation: The Search Inspector tool helps analyze search execution details such as runtime, skipped searches, and delays, making it useful for diagnosing KPI and service performance issues.

A2: Answer: C
Explanation: The `_internal` index contains Splunk's internal logs, including scheduler activity and skipped search information, which are critical when troubleshooting concurrency issues.

A3: Answer: A
Explanation: If a KPI shows "No Result," it often means the base search is not returning data—commonly because it is pointing to the wrong index or an index with no matching events.

A4: Answer: D
Explanation: ITSI Data Audit Dashboards help identify stale, incomplete, or missing KPI results, making them essential for troubleshooting KPI data quality issues.

A5: Answer: B
Explanation: Staggering KPI search schedules using cron offsets helps reduce peak load and avoid concurrency problems caused by too many searches running at once.

A6: Answer: C
Explanation: If the entity is not correctly mapped to the KPI, the Glass Table may fail to display the expected KPI data for that object.

A7: Answer: A
Explanation: The `itsi_troubleshooting_toolkit` provides diagnostic dashboards, health checks, and troubleshooting aids for identifying issues in ITSI deployments.

A8: Answer: B
Explanation: If Notable Events are missing, the first thing to verify is that the underlying correlation searches are enabled and actually returning matching results.

A9: Answer: C
Explanation: Documenting service and KPI changes helps teams trace issues, support root cause analysis, and roll back problematic changes when needed.

A10: Answer: A
Explanation: Incorrect threshold settings can distort KPI severity, resulting in misleading service health scores and unnecessary or missing alerts.