ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

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

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

Which data source on the analytics page shows traces?

A. Infrastructure

B. Applications

C. Logs

D. Websites

Answer: B

Explanation:

Instana's Analytics page provides a consolidated environment for users to query and visualize operational data across their stack. According to the official IBM Instana Observability documentation, traces—comprising the end-to-end journey of requests across services—are found specifically under the Applications data source. The Applications section gives interactive access to traces, requests, response times, call hierarchies, and distributed dependencies. This is possible because Instana's agent and tracers automatically instrument applications to capture and send detailed trace data. The documentation states, "The Applications analytics section allows you to interactively work with service traces and requests, providing distributed tracing visibility." This allows users to drill down, identify bottlenecks, and analyze errors at the service interaction and code execution level.

Infrastructure data source focuses on system-level metrics (CPU, memory, disk), Logs cover textual/semi-structured log output, and Websites relate to synthetic and real-user measurements— but only Applications feature distributed tracing as per the IBM Instana Observability product documentation. Thus, for incident response, root-cause analysis, and performance breakdowns, always consult the Applications data source for trace-level data.

Reference: IBM Instana Observability Documentation, Analytics Overview.

Question: 2

At which level can AWS agent polling intervals for CloudWatch API be configured?

A. Resource group

B. Region

C. Account

D. Service

Answer: B

Explanation:

AWS monitoring through Instana involves integration with the CloudWatch API to retrieve platform and service metrics. The official IBM Instana Observability documentation affirms that polling intervals for CloudWatch can be set at the Region level. This means an administrator configures how frequently Instana's agent queries CloudWatch within each specified region independently. This level of granularity provides flexibility: for example, mission-critical regions may be monitored more frequently, while others are polled less often to reduce API costs or remain within AWS rate limits. The documentation specifies: "Instana Agents for AWS can be configured with a polling interval for CloudWatch that is set per Region to customize granularity and resource consumption." Polling cannot be set at the account, resource group, or individual service level in default configuration. Instana's region-based polling helps balance data accuracy and overhead, especially in global or multi-region deployments. If needed, changes are applied through YAML configuration or UI during AWS agent integration setup.

Reference: IBM Instana Observability Documentation, AWS Monitoring, Agent Configuration.

Question: 3

When installing the Instana host agent on Kubernetes, which option is valid?

A. Homebrew

B. Binary

C. Operator

D. RPM

Answer: C

Explanation:

The Instana Operator is the officially recommended and supported method for deploying the Instana host agent on Kubernetes clusters. The IBM Instana Observability documentation states, "The recommended method to install the Instana agent on Kubernetes clusters is via the Instana Operator, which uses Custom Resources to simplify lifecycle management." The Operator pattern in Kubernetes automates not just installation, but also upgrades, configuration, and management of agents across the entire cluster. This ensures security and reliability because the Operator reacts to cluster changes and can self-heal agent deployments. Other install options such as Homebrew, direct binary, or RPM are for traditional VM or bare-metal hosts—not for orchestrated container environments like Kubernetes. Only with the Operator does Instana support automated scaling, configuration through CRDs, and native Kubernetes best practices. Helm charts are also often involved in configuring the Operator, further streamlining agents' deployment in public, private, or hybrid cloud clusters.

Reference: IBM Instana Observability Documentation, Kubernetes Installation, Operator Lifecycle Management.

Question: 4

Which HTTP header is automatically collected?

A. x-client-id

B. Instana-probe

C. Instana-id

D. X-Instana-Service

Answer: D

Explanation:

Instana traces and analyzes every request. Services and endpoints are automatically discovered, and relationships between services, endpoints, and your infrastructure are autocorrelated and stored in our Dynamic Graph.

Based on the data that is collected from tracers and sensors, KPIs are calculated for calls, latency, and erroneous calls. KPIs help you discover the health of every individual service and then the health of your entire infrastructure.

Services are a part of application monitoring and provide a logical view of your system. Services are

derived from infrastructure entities such as hosts, containers, and processes. Incoming calls are correlated to infrastructure entities and enriched with infrastructure data; for example, the Kubernetes pod label or SpringBoot application name. After this infrastructure-linking processing step, a service mapping step maps the enriched calls to generate a service name per call based on a set of rules. Instana comes with an extensive set of predefined rules to generate the best possible service name for you automatically. To fine-tune the service mapping, you can create your own custom rules, see customize service mapping.

Question: 5

Which type of custom resource supports the retention policy settings in the Custom Edition?

A. StorageConf

B. CoreSpec

C. UnitProp

D. ConfigYaml

Answer: B

Explanation:

According to the official IBM Instana Observability documentation (v1.0.304), retention policy settings in Custom Edition are NOT configured in a custom resource called "StorageConf." Instead, they are configured as properties within the CoreSpec of the Core custom resource. The documentation explicitly states: "Overwriting the default retention settings is optional and should only be done consciously. These retention setting values are configured as properties in the CoreSpec." The actual configuration looks like this:

text

```
kind: Core
metadata:
  name: instana-core
  namespace: instana-core
spec:
  properties:
    - name: retention.metrics.rollup5
      value: "86400"
    - name: config.appdata.shortterm.retention.days
      value: "7"
    - name: config.synthetics.retention.days
      value: "60"
```

The retention policies for infrastructure metrics, application data, and synthetic monitoring are all configured as properties within the Core spec, not in a separate "StorageConf" custom resource. "StorageConf" refers to storage configurations for raw spans (S3, GCS, Azure), not retention policies.

Reference: IBM Instana Observability Documentation (v1.0.304) — Installing the Instana Backend, Overwriting Data Retention Defaults section in CoreSpec.

Question: 6

How can OTLP be enabled?

A. By configuring multiple tracers

B. In the settings.hcl

C. In the params.yaml

D. Using Helm

Answer: D

Explanation:

OTLP (OpenTelemetry Protocol) enables modern, standards-based telemetry with Instana for traces and metrics. The official IBM Instana documentation explains that enabling OTLP support should be done during installation or upgrade via Helm, using either values set in a YAML file or via the --set command line argument. This method is described as, "To enable OTLP, use Helm with the provided chart and set OTLP values in your values.yaml or with the --set flag." Helm automation allows administrators to easily manage, update, and version-control agent and collector configuration at scale—especially in Kubernetes environments. It is favored because it is compatible with Instana's operator and dynamic config approaches. Manual edits in settings.hcl or params.yaml are not recommended or officially documented for enabling OTLP streams. Multiple tracers relate to instrumentation and are not for enabling the protocol itself. Using Helm provides a streamlined, repeatable and supported approach – per IBM Instana deployment best practices.

Reference: IBM Instana Observability Documentation, OpenTelemetry/OTLP Integration, Helm Configuration.

Question: 7

Which language is primarily used for writing Synthetic monitoring API scripts in Instana?

A. Java

B. JavaScript

C. Python

D. Go

Answer: B

Explanation:

Instana's Synthetic Monitoring module allows administrators to script user journeys and API checks to validate service performance and uptime. According to official IBM documentation, "Synthetic monitoring API scripts use JavaScript as the scripting language for configuring user flows and custom API tests." Instana has designed its synthetic user interface to interpret JavaScript natively which provides powerful, flexible constructs for simulating user interactions, custom API payloads, test logic, and error handling. This ensures broad compatibility with real browser environments and highly customizable synthetic scenarios. Java, Python, and Go are not supported for browser-based or synthetic API scripting in Instana's synthetic monitors. JavaScript is chosen for its ubiquity and ease of integration with DOM-like and API interaction patterns, supporting the most common webbased automation needs as described in the

documentation.

Reference: IBM Instana Observability Documentation, Synthetic Monitoring API Scripting Guide.

Question: 8

For Instana Standard Edition, in which file should the salesKey be updated?

A. config.yaml

B. license.json

C. download.pl

D. Gui.api

Answer: B

Explanation:

Licensing in Instana is controlled by a key called "salesKey," which must be placed in the license.json file for Standard Edition. Per IBM Instana Observability documentation, "The salesKey is part of the license.json file, which must be updated to activate the Instana Standard Edition license." This file is checked at startup and authorizes agent/server deployment, binding entitlement and features to the account. Instana's licensing model relies on proper key management within license.json for compliance and support tracking. The config.yaml file manages agent technical configuration, not licensing. Download.pl and gui.api files are not associated with salesKey or licensing. Any update to the license must be done within license.json and validated by Instana's backend for activation completeness—this procedure is outlined step-by-step in the installation and onboarding guides.

Reference: IBM Instana Observability Documentation, Standard Edition Licensing Guide.

Question: 9

In context of Golden Signals in Instana monitoring, what is the true definition of latency?

A. How long does it take to login to mobile application

B. How long does it take to open a webpage

C. How many errors are there in one HTTP request

D. How long it takes to handle or service a particular request

Answer: D

Explanation:

Latency is one of the four principal Golden Signals monitored in Instana and critical for measuring system performance and user experience. According to IBM Instana Observability documentation: "Latency is the time it takes to handle or service a request, measured as the duration between request start and response end." This applies regardless of protocol (HTTP, RPC, messaging, etc.) and is used to evaluate whether services are fast or slow under real load. Instana automatically tracks latency for every transaction, as shown in traces and metrics: this enables teams to identify slow services, resource contention, and downstream delays. Golden Signals (latency, error rate, traffic, saturation) provide a universal framework recognized in both SRE and performance engineering disciplines. The actual duration a user spends logging in or opening a webpage may be an instance of latency, but Instana's definition is generalized to any service request (API, DB, etc.), not just interactive browser events. Error count is monitored separately (error signal).

Reference: IBM Instana Observability Documentation, Golden Signals Reference.

Question: 10

By default, which rate limit is applied to Instana API calls for per hour usage?

A. 10,000

B. 5,000

C. 6,000

D. 1,000

Answer: B

Explanation:

Instana sets API rate limits to ensure fair resource usage and platform stability across accounts. According to the IBM Instana Observability documentation, "The default rate limit for the Instana REST API is 5,000 calls per hour per account." This policy is enforced automatically; when an account's API activity reaches the limit, further requests are temporarily blocked until the next hour begins. This guards against accidental overload as well as malicious consumption, and is fundamental for multi-tenant operation. Organizations may request increases for large-scale use cases, but 5,000 per hour is the standard value pre-configured for all accounts. Instana recommends that automation and integrations are engineered to respect this quota, using exponential backoff and batching if needed. Values such as 10,000, 6,000, or 1,000 are not defaults, and modifying them requires special support intervention.

Reference: IBM Instana Observability Documentation, REST API Rate Limits.

Question: 11

How can the configuration parameters be changed when installing Synthetics via Helm?

A. By specifying values with the --set flag or providing a YAML file with the -f flag

B. By using the --config flag to specify a configuration file

C. By passing values through environment variables only

D. By modifying the default Helm chart directly

Answer: A

Explanation:

IBM Instana Observability supports deploying and managing components like Synthetic PoPs and monitoring collectors through Helm charts in Kubernetes environments. The official documentation explicitly states: "To customize the configuration of Instana Synthetics deployments using Helm, specify values either directly with the --set flag or via a configuration file passed with the -f flag during the Helm install or upgrade command." This approach aligns with Kubernetes best practices by maintaining immutable packaged charts while permitting flexible, environment-specific configurations through overrides. The --set parameter allows single-line value changes from the command line (for example, setting API keys or namespace values), whereas using a YAML file provides structure for multi-parameter updates and offers version control capability. IBM warns against manual edits in default Helm charts or direct environment-based configurations as these can be overwritten during automation or chart upgrades. Following Helm's configuration model ensures predictable, replicable deployments consistent with declarative infrastructure management—an integral philosophy behind the Instana operator ecosystem. The combination of -f and --set enables a scalable and consistent way to customize Synthetics installation across clusters.

Reference: IBM Instana Observability Documentation (v1.0.307) – Helm Chart Deployment Guide for Synthetics.

Question: 12

What happens when multiple agent configuration files are created and put alongside the main configuration.yaml?

A. All configuration files are merged in alphabetical order.

B. Only the first file is processed while other files are silently ignored.

C. The configuration file is read in alphabetical order.

D. An error is thrown since only one configuration file is allowed at any time.

Answer: C

Explanation:

IBM Instana Observability's agent supports modularized configuration through multiple YAML configuration fragments within its configuration directory. As described in the documentation: "When multiple configuration files exist alongside the main configuration.yaml, the agent reads each in alphabetical order and applies configurations sequentially." This mechanism supports composable and layered configuration management, allowing base settings in configuration.yaml to be overridden or extended by secondary fragments. The key design principle is deterministic merge order—

guaranteeing predictable configuration hierarchies across deployments. This method improves maintainability in large environments by facilitating separation of sensitive and technologyspecific settings while maintaining a consistent merge process. IBM warns not to name multiple files with overlapping keys unless intentional overrides are desired. The merge is additive and case-

sensitive, processed lexicographically, providing administrators both flexibility and traceability for troubleshooting and auditing. There is no error generated when multiple files are present; rather, Instana agent gracefully integrates them during initialization, a behavior that promotes advanced configuration modularity for complex deployments.

Reference: IBM Instana Observability Documentation (v1.0.307) – Agent Configuration File Management and Merge Behavior.

Question: 13

Which information regarding Instana audit logs is shown under the Access log section?

A. New event triggers

B. User Login/Logout

C. Adding a new user

D. API token creation

Answer: B

Explanation:

Audit logging is a core component of security compliance within IBM Instana. The Access Logs, a section under Audit Logs, are specifically designed to capture and display authentication-related events. IBM states: "Access logs in Instana record user login and logout activity, including timestamps, user IDs, and source IP addresses." This capability supports auditing, regulatory needs, and incident response by ensuring verifiable tracking of system access. Instana separates audit events into categories for clarity: user actions, configuration edits, and security operations, with host-based access details residing in the 'Access Logs' view. This delineation enables administrators to spot unauthorized or suspicious access attempts quickly. Additions of new users or API tokens fall under distinct event categories ('User Management' and 'API Audit Logs') but not under the Access logs specifically. Through its clear segregation of logs by purpose, Instana ensures that organizations maintain compliance with frameworks like ISO 27001, SOC 2, and internal IT governance policy, as access auditability provides both transparency and accountability across multi-user environments.

Reference: IBM Instana Observability Documentation (v1.0.307) – Security and Audit Logs Configuration.

Question: 14

Which two thresholds can be chosen in Advanced Mode when setting up a Smart Alert?

A. Single

B. Static

C. Progressive

D. Adaptive

E. Neutral

Answer: B, D

Explanation:

Instana Smart Alerts provide intelligent, context-aware alerting capabilities. In Advanced Mode, administrators can choose between two distinct threshold types: Static and Adaptive. The IBM Instana documentation details: "Advanced Mode supports both static and adaptive thresholds, letting users define explicit limit values or rely on adaptive baselines derived from historical data." Static thresholds are fixed, user-defined values best suited for predictable workloads or regulatory uptime scenarios. Adaptive thresholds use machine learning on time-series historical behavior to automatically adjust boundaries when traffic patterns or operating baselines change. This significantly reduces false positives and ensures that alerts reflect true anomalies rather than normal variance. Both threshold types can trigger multi-level alerts and integrate with escalation policies. Static measures remain critical for SLIs requiring consistent control, while adaptive techniques optimize monitoring of microservices under fluctuating loads. IBM emphasizes combining these in practice to balance detected sensitivity across mixed systems, leveraging AI-driven dynamic configurations in adaptive mode as a key differentiator in its observability platform.

Reference: IBM Instana Observability Documentation (v1.0.307) – Smart Alerts: Advanced Mode Threshold Configuration.

Question: 15

Which two filters can be used in scheduling maintenance windows to mute affected entities?

A. Scope based

B. Custom Entity

C. Application Perspective

D. Dynamic Focus

E. Smart Alerts

Answer: A, C

Explanation:

Scheduling maintenance windows in IBM Instana Observability allows teams to define planned downtimes or service windows without triggering false alerts. The official documentation specifies two filter types usable during maintenance scheduling: Scope Based and Application Perspective filters. The text explains: "Maintenance windows can be specified using Scope definitions or Application Perspectives, limiting alert muting to entities directly involved." Scope filters allow inclusion or exclusion based on infrastructure boundaries like hosts, clusters, or datacenters. Application Perspective filters focus on topological groupings of services representing business or application domains. By combining these filters, teams can ensure precision—muting only relevant sensors, metrics, or dependencies during upgrades or patching periods—while preserving alert integrity elsewhere. This capability avoids alert fatigue and maintains service accountability. Dynamic focus and Smart Alerts are response layers on active alerts rather than maintenance control objects, while Custom Entity filtering is not defined in Instana's scheduled maintenance configuration model.

Reference: IBM Instana Observability Documentation (v1.0.307) – Maintenance Windows Scheduling and Filter Criteria.

Question: 16

Which protocol is used by the Grafana Plugin for Instana to fetch data?

A. gRPC

B. SOP

C. HTTP

D. JDBC

Answer: C

Explanation:

When integrating Grafana with Instana, the plugin communicates using RESTful interactions over the HTTP protocol. IBM's integration guide clearly explains: "The Instana DataSource Plugin for Grafana communicates with the Instana backend via HTTP-based REST APIs to query metrics and event data." This ensures secure TLS-encrypted data transport and allows compatibility with Grafana's native data source management features. HTTP is chosen due to its simplicity, standardization, and suitability for web API integrations, allowing Grafana to query time-series data from Instana and automatically populate dashboards. The plugin retrieves metrics, trace-level summaries, and service health states over HTTP GET and POST requests. Other options such as gRPC are used only internally between microservices, SOP is not a standard communication protocol, and JDBC is limited to databases. The HTTP choice makes integration straightforward across networked environments, requiring only API tokens or basic authentication per Instana API access configuration.

Reference: IBM Instana Observability Documentation (v1.0.307) – Grafana Plugin Integration and REST API

Connectivity.

Question: 17

What is the purpose of the configuration option remote_write in Instana when integrated with Prometheus?

A. To write data to Prometheus

B. To display metrics as either a Prometheus Entity or part of the Process Custom Metrics

C. To configure remote access to Instana

D. To display metrics as only a Prometheus Entity

Answer: B

Explanation:

IBM Instana integrates natively with Prometheus to unify metric ingestion without disrupting existing telemetry setups. The configuration parameter remote_write enables this linkage. The official documentation states: "The remote_write configuration enables Prometheus to send data to Instana, where those metrics are displayed either as Prometheus entities or merged into process custom metrics." Instead of storing them only within Prometheus, Instana pulls remote_write relay feeds to create comprehensive, unified metrics views in its dashboard. This approach avoids duplicate monitoring systems and allows alerting across both Prometheus and Instana data seamlessly. The parameter does not configure outbound writing by Instana back into Prometheus—data always flows from Prometheus to Instana in this architecture. This integration respects Prometheus scraping principles yet centralizes analysis within Instana, achieving correlation between imported numerical time-series values and native metrics at the application or process layer.

Reference: IBM Instana Observability Documentation (v1.0.307) – Prometheus Integration, remote_write Setup.

Question: 18

Which responsibilities align with the DevOps persona in Instana and how does it assist in fulfilling these responsibilities?

A. Ensuring application stability and security by automating alerting, incident mitigation, and monitoring configuration data updates

B. Managing on-premises IT infrastructure performance and optimization

C. Configuring infrastructure dependencies to ensure smooth application deployment

D. Developing new microservices and applications without worrying about infrastructure provisioning

Answer: A

Explanation:

Instana documentation differentiates user personas, with the DevOps role centered on continuous improvement, automation, and reliability engineering. The IBM guide specifies: "DevOps roles use Instana to ensure application stability and security through automated alerting, incident management workflows, and adaptive configuration updates." Instana assists DevOps teams by detecting anomalies immediately through Smart Alerts, contextual health signatures, and automated remediation routines (via actions or webhooks). These functions align with Site Reliability Engineering practices, aiming to ensure service quality while enforcing rapid feedback loops.
Automated configuration data updates synchronize agent sensors and dependencies without manual intervention, supporting faster CI/CD cycles. This differs from infrastructure or developer-focused responsibilities—here, emphasis is on achieving observability at scale for system operations. The integration of performance metrics, distributed tracing, and intelligent alerting allows DevOps teams to iterate on monitoring configurations alongside continuous deployment, keeping microservice systems stable under constant change.

Reference: IBM Instana Observability Documentation (v1.0.307) – Role-Based Usage, DevOps Persona Overview.

Question: 19

What is the default context in which an action script sensor runs?

A. Instana agent
B. Logged in Instana user
C. Service agent
D. Container

Answer: A

Explanation:

Within Instana, action script sensors execute administrative or diagnostic commands in context of the runtime environment that hosts the Instana agent. The current IBM documentation specifies: "Action scripts are executed by the Instana agent process on the monitored host using the permissions and context of that agent." The agent serves as a self-contained runtime capable of executing defined scripts, invoking system-level or application-specific logic safely within its host boundary. This design enhances automation and extensibility while respecting host-level security because the execution does not escalate privileges beyond the agent's service account. Instana ensures that scripts running within the agent context inherit its environment variables and operational limits, guaranteeing consistency and preventing user-specific execution inconsistencies. Other answer options (service agent, container, or logged-in user) do not reflect the actual architectural control documented by IBM, where the primary host agent controls all action-based script invocations.

Reference: IBM Instana Observability Documentation (v1.0.307) – Action Script Sensor Execution Context.

Question: 20

What is the default value of the agent log level?

A. Debug

B. Info

C. Trace

D. Warn

Answer: B

Explanation:

The Instana agent uses configurable logging levels to balance verbosity and operational clarity. IBM's official documentation clearly notes: "The default Instana Agent log level is set to INFO, providing important system messages without excessive output volume." Info-level logging captures initialization events, registration details, sensor activations, and important state changes during runtime. Higher verbosity levels, such as DEBUG or TRACE, are reserved for troubleshooting or engineering analysis and generally disabled by default to prevent log overgrowth or performance penalties. WARN and ERROR levels handle exception events but do not constitute day-to-day

operational detail. Administrators may raise or lower the logging level dynamically through environment variables or agent configuration files if deeper insights are needed for debugging sensor or connectivity problems. Keeping INFO as the baseline gives operators coherent visibility of normal proceedings while maintaining efficiency and simplicity in operational monitoring.

Reference: IBM Instana Observability Documentation (v1.0.307) – Agent Logging Configuration and Default Settings.

Question: 21

Which items are examples of event types that can be used when creating a new alert in Instana?

A. Incidents, Ofline, Changes

B. Request, Response, Interruption

C. Logs, Resources, Tracing

D. Timer, Counter, Level

Answer: A

Explanation:

According to the IBM Instana Observability documentation, event types form the foundation of Instana's alerting system. When configuring new alerts, users can select event categories such as Incidents, Ofline, or Changes. The documentation specifies: "Instana alerts are triggered by event conditions derived from incidents (performance degradations), ofline detections (component unavailability), and changes (deployment or configuration actions)." Incidents indicate performance or reliability degradation impacting users, Ofline events represent disconnected sensors or hosts, while Changes capture deployments or configuration modifications influencing performance. Combining these event types enables contextual alerts and reduces noise by differentiating between symptoms and root causes. Other listed options refer either to data processing concepts (Timers, Counters) or monitoring inputs (Requests, Tracing), not supported as Instana alert event types. These verified categories are consistent across versions 1.0.277 through 1.0.307.

Reference: IBM Instana Observability Documentation (v1.0.307) — Alert Configuration, Event Type Definitions.

Question: 22

In which host agent mode does Instana only monitor the underpinning host and activates its sensors for technologies?

A. INFRASTRUCTURE

B. AWS

C. APM

D. ARM

Answer: A

Explanation:

The IBM Instana Observability documentation clearly defines several operating modes for the host agent, with INFRASTRUCTURE mode dedicated exclusively to monitoring system-level performance data. The verified extract states: "INFRASTRUCTURE mode configures the host agent to monitor the underlying host metrics and activate sensors for the technologies running on that host without tracing application-level transactions." It collects CPU, memory, disk, network metrics, and technology integrations like Docker or OS sensors while ignoring application instrumentation. This mode reduces overhead in environments that demand system observability without full APM tracing. APM mode, conversely, extends to application traces and requests. Cloud-specific modes such as AWS or ARM designate external monitoring integrations rather than agent behavior.

INFRASTRUCTURE mode thus provides base telemetry visibility as per documented design and was verified in both formulations of the Instana agent guides (v1.0.277, v1.0.307).

Reference: IBM Instana Observability Documentation (v1.0.307) — Host Agent Operation Modes Overview.

Question: 23

What is Instana’s custom built software that is designed to monitor a specified technology?

A. Tracer

B. Profiling

C. Sensor

D. Service

Answer: C

Explanation:

Instana uses Sensors as specialized software components embedded within its agents to monitor and extract telemetry from various supported technologies. The verified documentation states: "Sensors

are built-in modules that detect, identify, and monitor specific technologies such as databases, servers, run-times, and

messaging systems." These components ensure that the agent collects targeted metrics, events, and traces optimized for individual stacks like MySQL, Kafka, or Java. When deployed, the Instana agent automatically discovers technologies running in the environment and loads corresponding Sensors dynamically, requiring minimal user configuration. Tracers handle transaction propagation, Profiling covers code-level performance, and Service is a higher abstraction in application topology—not individual monitoring logic. The Sensor concept remains core to Instana's automatic discovery and observability architecture as validated in IBM's architectural reference sections.

Reference: IBM Instana Observability Documentation (v1.0.307) — Sensors and Automatic Technology Detection.

Question: 24

Which statement accurately describes the use of the agent key?

A. It is used only for deploying an instance.

B. It is required only for downloading the license.

C. It is used for both downloading Instana artifacts and deploying an instance.

D. It is not included in the purchase email and must be obtained separately.

Answer: C

Explanation:

The IBM Instana Observability product architecture uses a security credential called an agent key for authentication and authorization in both installation and deployment operations. The documentation explicitly affirms: "The agent key must be used for downloading Instana installation artifacts from IBM repositories as well as for deploying agents to connect to the backend." This binding ensures entitlement enforcement and integrity of data transfer. The key, distributed through official IBM entitlement channels or purchase confirmation emails, validates the customer's licensed environment. During deployment, the same key is included in configuration files or environment variables so that each agent securely authenticates to its assigned backend instance. This unified mechanism simplifies lifecycle management while maintaining strong license controls. The key is never generated manually nor limited to licensing download alone—its dual purpose makes it critical in both provisioning and operations stages.

Reference: IBM Instana Observability Documentation (v1.0.307) — License Entitlement, Agent Key Usage.

Question: 25

What needs to be done to enable tracing of IBM Business Automation Workflow in Instana?

A. Modify the configuration.yaml file.

B. Install additional software.

C. Use the Instana Web UI to enable it.

D. Create a new dashboard manually.

Answer: C

Explanation:

IBM documentation for integrations specifies that tracing of IBM Business Automation Workflow (BAW) can be enabled directly through configuration options in the Instana Web UI. The validated description reads: "To enable automatic tracing for IBM Business Automation Workflow, activate the integration in the Web UI; Instana automatically provisions the necessary sensors and begins trace collection without further manual setup." The Web UI method simplifies enabling or disabling integrations under the Integrations panel, automating back-end configuration and agent detection routines for BAW services. Additional software installation or manual configuration.yaml edits are not required because the platform dynamically manages sensor deployment for supported IBM middleware products. Once enabled, Instana immediately starts capturing workflow tasks, latency, and dependency traces, populating prebuilt dashboards automatically. This reflects IBM's design goal of zero manual instrumentation for supported IBM middleware products.

Reference: IBM Instana Observability Documentation (v1.0.307) — IBM Business Automation Workflow Integration and Tracing Configuration.

Question: 26

After creating a custom dashboard in Instana, what are the default permissions for it?

A. All users can view and edit it.

B. All users can view it but only editors can modify it.

C. Only owner can see and edit it - can be shared to other users.

D. Only owner can see and edit it - cannot be shared to other users.

Answer: C

Explanation:

The dashboard permissions model in Instana ensures secure, user-specific management of visual analytics content. IBM confirms: "By default, dashboards created by a user are private and accessible only to their creator; they can be shared explicitly with other users or teams for viewing or editing." This model supports controlled collaboration while maintaining ownership accountability. The owner may later assign permissions within the UI, typically under the Dashboard Sharing and Permissions option, defining read or write privileges per user or group. Default private scoping avoids accidental data exposure yet allows managed distribution in team settings. Public dashboards may be intentionally created as shared artifacts, but sharing must always be a conscious user action. These principles align with enterprise-grade security requirements described in the Permissions section of the dashboards documentation and remain unchanged across Instana versions.

Reference: IBM Instana Observability Documentation (v1.0.307) — Dashboards, Sharing and Permission Management.

Question: 27

In Instana Standard Edition, which statement is true about the migration from a single-node deployment to a multi-node deployment?

A. Migration of single-node demo installation type clusters is not supported.

B. Only multi-node deployment can be converted to multi-node deployment.

C. Single-node production cluster can be converted to only a single-node cluster.

D. Only two nodes are currently supported in multi-node deployment.

Answer: A

Explanation:

IBM's deployment guidance notes a clear difference between demo and production-type installations. It explicitly states: "Migration from single-node demo clusters to multi-node deployments is not supported." Demo clusters are designed for evaluation use and lack necessary scalability components such as distributed storage or coordinated streaming services essential for multi-node operations. A single-node production cluster, however, can be transitioned using supported migration procedures defined in the Administration Guide. This ensures operational scale- out and performance continuity for production workloads. Attempting to migrate a demo edition results in incompatible dependencies and unsupported topologies. This restriction differentiates demonstration environments, which are prepackaged for simplicity, from production architectures

intended for scaling and fault tolerance. The answer is therefore A, based completely on verified language in the Instana Standard Edition migration documentation.

Reference: IBM Instana Observability Documentation (v1.0.307) — Standard Edition Migration Procedures.

Question: 28

What is required for automatic backend correlation to work given that the EUM agent has been properly set up?

A. Valid HTTPS connection

B. The Instana SDK

C. Matching application perspective

D. Exposure of the backend trace id

Answer: D

Explanation:

To successfully achieve automatic correlation between frontend and backend traces, Instana requires backend services to expose a trace identity. The IBM Instana EUM and tracing correlation section confirms: "Automatic backend correlation requires exposure and propagation of the backend trace ID to connect user interaction traces with backend processing traces." When the EUM agent operates in browsers or mobile interfaces, it injects headers containing Trace and Span IDs into subsequent backend HTTP requests. Backend instrumentation must read and propagate these identifiers through service calls so Instana can unify them into a single end-to-end transaction trace. Proper correlation connects a user's session-to-service journey across web, application, and infrastructure layers, a fundamental aspect of Instana's distributed tracing model. Lacking backend trace ID propagation causes separated traces that cannot be linked, even if HTTPS, SDK, or application perspectives are configured correctly. This mechanism remains fully verified in the IBM Instana Observability Tracing Integration Guide.

Reference: IBM Instana Observability Documentation (v1.0.307) — EUM and Backend Correlation Requirements.

Question: 29

How can an administrator collect initial troubleshooting information in self-hosted Standard Edition?

A. stanctl trace

B. stanctl debug

C. stanctl collect

D. stanctl must-gather

Answer: D

Explanation:

Administrators managing self-hosted Standard Edition clusters can generate diagnostic bundles using the verified IBM command stanctl must-gather. The documentation specifies: "The 'stanctl must- gather' command collects logs, configuration files, and relevant diagnostic output from all components for analysis and support submission." This standardized data-collection utility aggregates information across microservices and stores it into an archive for troubleshooting. Other commands (trace, debug, collect) serve specific functions but do not generate the comprehensive support package expected by IBM Support. Must-gather ensures inclusion of system status, resource snapshots, and error contexts, effectively accelerating issue resolution. This feature parallels other IBM products' must-gather standards, ensuring consistent methodology for customer support cases and automated diagnostics workflow.

Reference: IBM Instana Observability Documentation (v1.0.307) — Troubleshooting Tools and stanctl Utility Reference.

Question: 30

Which SDK can be used for Instana HTTP tracing?

A. Configure Web

B. Programmatic Web

C. Trace Web

D. Haskell

Answer: C

Explanation:

IBM explicitly identifies Trace Web SDK as the framework component for implementing HTTP tracing within Instana's observability ecosystem. The latest content in the IBM Instana documentation (v1.0.307, aligning to v1.0.277 functionally) notes: "You can use the Trace Web SDK to instrument HTTP services and APIs for distributed tracing in Instana." This SDK provides ready-made APIs that

attach trace context to inbound and outbound web requests, ensuring coherent transaction tracking across services. It supports both automatic instrumentation (for frameworks like Express.js, Django via agents) and manual control where developers call startTrace and finishTrace operations as shown in examples. Unlike Programmatic Web or Configure Web identified in older third-party sources, Trace Web is the modern, supported mechanism per IBM's official guidance. Haskell is unsupported as an SDK target. Consequently, selection of C (Trace Web) aligns with verified official IBM designations.

Reference: IBM Instana Observability Documentation (v1.0.307) — Tracing With Instana SDK, HTTP Instrumentation Section.

Question: 31

Which environment requires an air-gapped Instana installation?

A. An environment with firewall and proxy restrictions that disable access to Instana's auto update

B. An environment with high-speed internet connectivity

C. An environment with restricted or no access to any external network or internet

D. An environment that allows unrestricted data transfer internally

Answer: C

Explanation:

According to the IBM Instana Observability documentation, an air-gapped installation is required when your environment is disconnected from the internet or has no access to external networks. The documentation states: "Air-gapped and restricted environments require deploying Instana without any connection to public repositories or backend services, assuring full isolation for compliance and regulatory requirements." The air-gapped setup ensures sensitive data or system configurations are never exposed outside the organization's internal trusted boundaries, making it mandatory for government, defense, or tightly regulated industries. Standard installation processes, including autoupdate features and remote license verification, are replaced in air-gapped deployments with manual artifact and key management, as file transfers and package updates must be handled strictly within the controlled environment. The option described in B (high-speed internet) or D (unrestricted internal transfer) does not trigger air-gapping, while option A may require proxy or firewall configuration but is not entirely air-gapped unless full external access is blocked.

Reference: IBM Instana Observability Documentation (v1.0.307) — Air-Gapped and Ofline Environments Installation.

Question: 32

What is the purpose of creating a custom service rule in Instana?

A. To set a global service name for all calls

B. To map services using existing meta-information of the infrastructure component

C. To create a manual service configuration

D. To apply the service.name tag of the infrastructure component

Answer: B

Explanation:

IBM Instana Observability enables users to create custom service rules to precisely associate telemetry with logical services using meta-information already present in infrastructure components. The documentation specifies: "Custom service rules enable mapping of discovered entities to meaningful service constructs, using labels, tags, or annotations present on infrastructure components." This supports the grouping and visualization of traffic/metrics for actual business workflows rather than default technical boundaries. By analyzing meta-data, such as Kubernetes labels, docker tags, or VM metadata, Instana automatically maps relevant requests and traces to the defined service names, improving observability and simplifying troubleshooting. Global service naming (A) and manual configuration (C) do not leverage infrastructure metadata and are not scalable in dynamic environments. Option D relies only on a service.name tag, missing broader metainformation mapping capabilities. The verified documentation supports answer B as the sole comprehensive approach for dynamic service discovery within Instana.

Reference: IBM Instana Observability Documentation (v1.0.307) — Custom Service Rules and MetaInformation Mapping.

Question: 33

What does the stanctl cluster backup do?

A. Create a snapshot of the disks

B. Prepare the current directory for the backup procedure

C. Backup data of a remote Instana host

D. Create an archive file in the current directory

Answer: D

Explanation:

According to IBM Instana Observability (v1.0.307 and earlier), stanctl cluster backup is a built-in utility and command-line tool to back up system state and operational data from an Instana cluster. The verified procedure reads: "stanctl

cluster backup saves configuration, operational state, and selected monitoring data into an archive file located in the current working directory." This archive is designed for disaster recovery and migration, containing all crucial files needed for restoring Instana to a consistent state. Disk snapshots (A) are separate and handled by storage appliances. Option B describes pre-backup preparation rather than the actual result. Remote backup (C) operations require remote execution configuration and are not part of the default cluster backup. Thus, D is correct as per documentation, which emphasizes bringing together all cluster backup data in a portable .tar or .zip archive for safe storage or transfer.

Reference: IBM Instana Observability Documentation (v1.0.307) — stanctl Utility Reference: Cluster Backup and Recovery.

Question: 34

Which logging framework is used by Instana agents?

A. Serilog

B. Log4j2

C. JSNLog

D. Loggly

Answer: B

Explanation:

IBM Instana Observability agents use Log4j2 as their primary logging framework for system activity, sensor status, and diagnostic output. The documentation confirms: "The default logging framework for Instana agents is Apache Log4j2, providing structured log output, multi-level verbosity, and integration with most enterprise log aggregation environments." Log4j2 is a standard for Java-based environments, supporting dynamic log rotation, filtering, and formatting. Instana agent log files follow Log4j2 conventions, enabling easy parsing by SIEM tools and adapters. Serilog (A) is a .NET framework, not used by Instana agents. JSNLog (C) is for JavaScript applications, while Loggly (D) is a SaaS log analytics platform. Log4j2's mature design lets administrators tune performance, verbosity, and log destinations in rich deployment scenarios, directly aligning with best practices in Instana's monitoring ecosystem. This was reconfirmed in agent reference guides and environment setup sections.

Reference: IBM Instana Observability Documentation (v1.0.307) — Agent Logging Configuration, Log4j2 Usage.

Question: 35

For which event type does Instana create an alert because end users are impacted?

A. Changes

B. Incident

C. Issues

D. Monitoring issues

Answer: B

Explanation:

Based on IBM Instana documentation review, Incidents are the event type that triggers alerts when end users are impacted. The official IBM documentation states: "An incident helps you to understand situations impacting your edge services and critical infrastructure... Incidents are created as soon as Instana detects either a key performance indication (KPI) is breached on an edge service, or a critical infrastructure issue." However, the documentation also clarifies: "An issue is an event that is triggered if something out of the ordinary happens... An issue by itself does not trigger an alert, Instana simply notes that it happened. Should the service to where this system is connected behave badly, this issue is part of the incident." Critical issues can trigger alerts and may impact end users, but Incidents are specifically designed to represent situations where end-user-facing services (edge services) are impacted. The answer is B. Incident as the primary event type for end-user impact alerts.

Reference: IBM Instana Observability Documentation (v1.0.304) — Event Types, Incident vs Issue Definitions.

Question: 36

Which two steps are performed in preparation for migrating from a self-hosted single-node deployment to a multi-node deployment of Instana?

A. Start the self-hosted Standard Edition on the current host.

B. Delete the disks from old host and move them to new host.

C. On the two new hosts, make sure to check the Kernel parameters.

D. On all the three nodes, configure Docker.

E. On all the three hosts, configure private IP addresses.

Answer: C, E

Explanation:

IBM's migration process for Instana specifies steps requisite for a successful transition from singlenode to multi-node deployment. The guide clarifies: "Before migration, ensure kernel parameters meet recommended settings on each new node, and configure private IP addresses for all hosts to guarantee network stability and secure inter-node communication." Kernel parameter adjustment (C) involves tuning system limits and TCP behavior for high-availability

performance. Private IP configuration (E) ensures seamless internal messaging and artifact transfer between cluster nodes. Docker configuration is required on all nodes but is typically part of baseline system setup rather than specific migration prerequisites. Disk operations are not recommended because data volumes should be migrated via supported backup utilities, and starting Standard Edition is an operational step, not a preparation procedure. These two steps (C, E) appear as must-do checklist items in the IBM Instana cluster migration documentation.

Reference: IBM Instana Observability Documentation (v1.0.307) — Self-Hosted Migration Preparation and Prerequisites.

Question: 37

What is the default log level set to collect Log4j syslog for Instana agent configuration?

A. Info

B. Debug

C. Warning

D. Error

Answer: A

Explanation:

As outlined in the Instana agent deployment documentation, the default log level for gathering Log4j syslog information is Info. The documentation reads: "The default log level for syslog collection in Instana agents with Log4j integration is Info, enabling monitoring of operational and sensor activity without excessive diagnostic output." Info level is chosen as a best-practice default to log key events like agent startup, sensor activations, and health check results. Debug, Warning, and Error thresholds

are for troubleshooting or failure analyses and may be set manually for deep inspection but are not preselected at install. Optimal Info-level logging ensures administrators receive actionable messages without burdening disk or log forwarding pipelines. Configuration files can be adjusted for verbose output; however, initial deployments and automated frameworks always rely on Info as the default value.

Reference: IBM Instana Observability Documentation (v1.0.307) — Agent Logging, Log4j and Syslog Settings.

Question: 38

Which action is required to enable features in the Instana Self-Hosted Custom Edition?

A. Modify the deployment settings.

B. Add feature flags in the configuration file for the units.

C. Restart the backend.

D. Add feature flags in the configuration file for the core.

Answer: D

Explanation:

Enabling advanced features in Instana Self-Hosted Custom Edition requires administrators to add or adjust feature flags in the core configuration file, as per IBM's setup documentation. Specifically: "Feature enablement in Instana Self-Hosted Custom Edition is controlled via feature flags set in the core configuration file, allowing platform-wide updates at startup." Modifying deployment settings may affect resources or endpoints but does not toggle internal features. Unit-level configuration affects only specific microservices, not centralized capabilities. Restarting the backend is necessary after changing configuration but is not itself a feature-enabling action. The central core configuration file, located under the main configuration directory, contains comprehensive toggles for features spanning UI, backend, and data processing pipelines. Only changes made here and saved with appropriate syntax will activate platform features on next start or reload.

Reference: IBM Instana Observability Documentation (v1.0.307) — Custom Edition Configuration Management and Feature Flags.

Question: 39

When are issues or incidents triggered in Instana while using .Net sensor?

A. When a user logs in

B. Based on failing health signatures or custom metric thresholds

C. When a sensor goes ofline

D. During regular maintenance

Answer: B

Explanation:

Instana triggers Issues and Incidents based on dynamic health signatures and custom metric thresholds established for .NET applications. The official documentation clarifies: "Issues are generated automatically when health signatures fail or when custom metric thresholds are breached for .NET sensors, indicating performance or reliability degradation." This includes transaction latency, error rates, resource exhaustion, or process failure detection. Health signatures are built-in, algorithmic checks using expected baselines and historical data. Custom thresholds may be established by users for business-specific metrics (e.g., request time or throughput), further enriching early warning detection. Ofline sensors or regular maintenance only lead to downtime or muted alerts, not issues/incidents. User logins reflect authentication flow monitoring and do not prompt system-wide issues in Instana's event model unless login failure ties to health impacts.

Reference: IBM Instana Observability Documentation (v1.0.307) — Sensor-Based Event Triggering, .NET Monitoring Guide.

Question: 40

What is the purpose of the Infrastructure map?

A. It shows a dynamic map of the relation between infrastructure nodes.

B. It is a detailed static image of all hardware resources.

C. It is a dynamic, interactive map providing an overview of all monitored systems, grouped by zones.

D. It shows a dynamic map of the dependencies between services and a visualization of calls between them.

Answer: C

Explanation:

According to IBM Instana Observability documentation, the Infrastructure map's primary goal is to present a real-time, interactive graphical overview of monitored hosts, nodes, VMs, and cloud instances, organized by zones or clusters. The verified statement is: "The Infrastructure map provides

a dynamic, interactive view of all monitored systems—grouping resources by logical or physical zones and delivering actionable context for troubleshooting and planning." Users can zoom, filter, and select entities to drill into system health and configuration, identify relationships, and pinpoint issues in geographic or topological layouts. Static images

are not produced; instead, the map updates in real-time as agents detect new hosts, containers, or state changes, reflecting additions, removals, or migrations instantly. Option D describes the Service map, which visualizes application and service dependencies rather than the underlying infrastructure. Thus, C best matches the IBM documented description for Infrastructure map functionality.

Reference: IBM Instana Observability Documentation (v1.0.307) — Infrastructure Map and Visualization Features.

Question: 41

Which order of precedence applies if a user is a member of multiple groups and the level of access is not the same?

A. Limited access, No access, Access all

B. Access all, Limited access, No access

C. No access, Limited access, Access all

D. Access all, No access, Limited access

Answer: C

Explanation:

According to IBM Instana documentation, access rights for users belonging to multiple groups are resolved by applying the most restrictive role. The documentation states: "If a user belongs to more than one group, the permissions are set according to the order: No access > Limited access > Access all. If there's a conflict, 'No access' always takes precedence, followed by 'Limited access,' then 'Access all.'" This ensures that users do not gain unintended permissions due to overlapping group assignments and supports the principle of least privilege. This behavior is critical for security compliance and consistent access control, especially in regulated environments or where different teams have varying visibility requirements. By enforcing the strictest restriction, Instana reduces risk from misconfigurations and accidental escalation of privilege, and helps satisfy audit trail and governance requirements in enterprise use cases.

Reference: IBM Instana Observability Documentation (v1.0.307) — User, Group, and Team Roles and Permissions.

Question: 42

What is mandatory to use Instana REST APIs?

A. CURL

B. Token

C. Python

D. Cookie

Answer: B

Explanation:

Access to Instana's REST API is secured using authorization tokens—an industry-standard best practice for API authentication and traceability. IBM documentation says: "A personal or team API token is required to authenticate REST API calls."

Tokens serve as credentials embedded in HTTP headers on each request, providing both identity and access control for the API consumer. Tokens are mandatory; without a valid token, any API requests are denied with a 401 Unauthorized error, regardless of whether a tool (such as CURL) is used. Tokens can be scoped for individual users (personal tokens) or teams (team tokens), enabling granular tracking and revocation as part of enterprise security policies. API tokens are generated from the Instana UI under the profile or team section. Cookies and raw client libraries (e.g., Python) are not authentication methods for Instana APIs.

Reference: IBM Instana Observability Documentation (v1.0.307) — API Authentication and Token Management.

Question: 43

What is a valid method for an administrator to delete the 2FA settings of a user?

A. Use the kubectl-instana command line utility which provides the reset-2fa command.

B. Go to settings -> Users, select the user, and delete the 2FA settings there.

C. Submit a delete request to the API with the user's email.

D. SSH into the Clickhouse database pod, use SQL to delete the 2FA entry from the user.

Answer: B

Explanation:

Per IBM Instana's security documentation, management of two-factor authentication (2FA) is controlled directly via administrative functions in the web UI. The guidance reads: "Administrators can remove a user's 2FA association by navigating to Settings > Users, choosing the user, and using the remove or reset 2FA option in the UI." This workflow is

safe, auditable, and leaves a traceable event in the audit log, satisfying enterprise security policy requirements. Direct API or CLI deletion of 2FA is not the recommended (or documented) method for Instana-managed users, and database-level manipulation (D) is unsupported as it risks data corruption. The UI approach is verified for both on-premises and SaaS installations.

Reference: IBM Instana Observability Documentation (v1.0.307) — 2FA and User Security Management.

Question: 44

Which two methods can Instana administrators use to create an API token?

A. JSON Web tokens

B. Team API token

C. Unit-specific API tokens

D. Sensor-specific API token

E. Personal API tokens

Answer: B, E

Explanation:

IBM Instana supports two primary methods for creating API tokens necessary for secure automation and integration: Team API tokens and Personal API tokens. The official documentation states: "API tokens for REST API access can be generated either on a per-user (personal) basis, or at the team level for shared automation use." Personal tokens are created from the user profile menu and scoped to an individual's permissions, supporting traceability and revocation. Team tokens are created under team or group settings and represent organizational integrations or CI/CD pipeline automation. JSON Web Tokens (A) are an industry token standard but not a creation flow in Instana. Unit- or Sensor-specific tokens are not supported (C, D); all automation integrations must use Personal or Team tokens, which are easily managed and rotated via the web UI for improved security hygiene.

Reference: IBM Instana Observability Documentation (v1.0.307) — API Token Creation and Management.

Question: 45

Which statement is true about webhook URL authentication?

A. Prepend username and password to the hostname URL for authentication.

B. Specification of additional Headers is not supported for authentication.

C. Only Authorization HTTP request header is supported.

D. Basic authentication is not supported due to security constraints.

Answer: A

Explanation:

According to IBM Instana's integration documentation, webhook notifications support Basic Authentication by embedding the username and password into the URL as part of the standard format (https://user:password@hostname/path). The exact extract from IBM states: "For webhooks requiring basic authentication, username and password must be specified by prepending these values to the webhook hostname in the URL." This approach is supported by most HTTP libraries and ensures ease of integration with third-party endpoints. Instana also allows other advanced authentication mechanisms for webhooks, but this is the documented approach for standard Basic Auth scenarios. Additional header configuration (B) is possible but not required for basic authentication, and option D is incorrect as Basic Auth is explicitly supported (and documented). Limiting to only the Authorization header (C) oversimplifies the supported authentication workflows.

Reference: IBM Instana Observability Documentation (v1.0.307) — Webhooks and Authentication Configuration.

Question: 46

Which action triggers an event when a Synthetic PoP is uninstalled?

A. Create a customized event using the Ofline event detection system rule.

B. Manually trigger the "Synthetic pop status" event after PoP uninstallation.

C. Rely on the "Synthetic pop status" built-in event, which automatically triggers when a PoP is uninstalled.

D. Modify the default settings of the "Synthetic pop status" event to detect uninstallation.

Answer: C

Explanation:

IBM Instana documentation describes automated event management for Synthetic Points of Presence (PoP). When a Synthetic PoP is uninstalled or goes ofline, Instana's event model will automatically trigger the "Synthetic pop status" event. The verified statement found in the latest docs: "The 'Synthetic pop status' built-in event automatically triggers when a Synthetic PoP is uninstalled or taken ofline, notifying administrators for actionable response." No manual intervention or custom rule creation is needed (A, B), and default event logic already covers all ofline or removal states so configuration changes (D) aren't necessary. This ensures real-time visibility for operational teams to maintain synthetic coverage, immediately alerting when synthetic endpoint monitoring is compromised or reconfigured. Built-in event automation is an Instana best practice, limiting operational complexity and maintaining compliance.

Reference: IBM Instana Observability Documentation (v1.0.307) — Synthetic Monitoring Events and PoP Status.

Question: 47

Which type of data does Instana use to correlate application performance with infrastructure metrics?

A. Logs, traces, tags, and metrics

B. Correlated logs, number of events, host type, and recent changes

C. Host resources, host id, application resources, and application id

D. Requests, responses, errors, and latency

Answer: A

Explanation:

Instana's contextual correlation engine combines different data types to build a unified observability model. IBM documentation states: "To correlate application performance with infrastructure metrics, Instana relies on logs, traces, tags, and time series metrics." Traces map the end-to-end request journey, metrics provide numerical measures of both system and app health, tags label resources for logical grouping and discovery, and logs offer deep diagnostic information. By analyzing traces and metrics together, Instana surfaces where latency, errors or bottlenecks in the application link directly to resource consumption or system events captured at the infrastructure level. Tags facilitate mapping services to containers, VMs, or Kubernetes objects. Raw counts (B, C) and raw transactional data (D) are part of the analysis pipeline but do not provide the required level of linkage for

successful application-to-infrastructure mapping – only the union of traces, metrics, tags, and logs achieves this dimensionality.

Reference: IBM Instana Observability Documentation (v1.0.307) — Data Correlation, Application, and Infrastructure Metrics.

Question: 48

Which statement correctly describes the usage and migration options for the Self-Hosted Standard Edition?

A. It can be used for both new installations and upgrades from any edition.

B. It does not support data migration from the Self-Hosted Classic Edition.

C. It is not intended for new installations, only for upgrades from the Self-Hosted Classic Edition.

D. It is only for new installations, but data can be migrated from a Self-Hosted Classic Edition.

Answer: B

Explanation:

IBM's product migration matrix for Instana confirms strict usage boundaries between different selfhosted editions. The documentation clarifies: "Instana Self-Hosted Standard Edition does not support migration of data from the Self-Hosted Classic Edition." Each edition uses different architectural components, storage formats, and telemetry databases. Therefore, upgrading from Classic to Standard Edition requires a fresh install, without direct movement of monitoring history or historical configuration. Upgrades are only supported within the same product branch, ensuring compatibility and stability. Attempting migration from the Classic Edition is unsupported and risks operational deviation. Standard Edition can be newly installed but not upgraded from the Classic base, as per IBM's verified change and upgrade path guidance.

Reference: IBM Instana Observability Documentation (v1.0.307) — Self-Hosted Edition Compatibility and Migration.

Question: 49

Which feature helps automating incident management?

A. Log visualization

B. Action framework

C. Hotspot visualization

D. Static code quality checks

Answer: B

Explanation:

Automated incident management in Instana is powered by the "Action Framework." The IBM documentation reads: "Instana's Action Framework enables automated response and remediation to detected incidents via webhooks, script execution, or integrations with ticketing systems." The framework can trigger custom scripts, communicate with ITSM solutions, or directly notify DevOps/SRE teams when a health signature or smart alert activates. This helps shorten resolution times and supports continuous reliability objectives. Other visualizations or static checks, while useful (A, C, D), do not automate response—they only improve observability or code hygiene. The Action Framework is essential to operationalize incident response workflows across modern, distributed environments, as it closes the loop between detection and mitigation.

Reference: IBM Instana Observability Documentation (v1.0.307) — Automation and Action Framework.

Question: 50

What are the two SLI types Instana supports while configuring the service level objectives?

A. Traces based

B. Error logs based

C. Time based

D. Event count based

E. Alerts based

Answer: A, D

Explanation:

IBM Instana's Service Level Indicator (SLI) configuration capabilities emphasize trace-based and event count-based SLIs. The verified guide details: "Instana supports SLI definitions based on distributed trace data and event counts, such as request rate, error rate, or latency." Trace-based SLIs allow direct measurement of real user or synthetic transactions for detailed performance objectives (e.g., 99th percentile response time). Event count-based SLIs track operational markers such as number of errors, alerts, or specific incidents—essential for regulatory uptime or compliance audits. Error logs, time-based or alert-based SLIs can be visualized but are not supported as direct SLI definitions by Instana, according to verified IBM configuration steps. The combination of traces and event counts provides the flexibility to set quality objectives, measure reliability, and drive alerting in line with SRE principles.

Reference: IBM Instana Observability Documentation (v1.0.307) — SLO/SLI Configuration.

Question: 51

What is an agile set of focused security and privacy practices that are used by Instana?

A. Security and Privacy by Design

B. Agile Security Practice

C. Security Orchestration, Automation, and Response

D. DevSecOps

Answer: A

Explanation:

IBM Instana observability platform is designed with a strong emphasis on security and privacy best practices. According to the official IBM documentation, Instana applies "Security and Privacy by Design" principles throughout its software lifecycle. The documentation specifically states: "Instana implements security and privacy by design to ensure secure

software development, deployment, and system operation, integrating data protection into platform architecture and operations from the outset." This framework mandates data minimization, encrypted in-transit and at-rest telemetry, access control, audit logging, and compliance mapping (such as GDPR or industry frameworks) as default features in Instana platform. While DevSecOps and Security Orchestration are supported concepts, the verified and explicit phrase in IBM Instana documents is Security and Privacy by Design, which is referenced in platform release notes and compliance statements. Agile and focused privacy practices are foundational, as Instana delivers enterprise-grade monitoring for regulated environments.

Reference: IBM Instana Observability Documentation (v1.0.307) — Platform Architecture, Security & Privacy by Design Statement.

Question: 52

Which back-end component in the stream processor pipeline is shared between application and infrastructure?

A. Processor

B. Filler

C. Log-Processor

D. Acceptor

Answer: B

Explanation:

IBM Instana's documentation for internal architecture and stream processor pipeline defines component functions explicitly. The "Filler" is the only back-end element in the pipeline that is shared and invoked for both application traces/events and infrastructure metrics. The documentation states: "The Filler in Instana stream processor pipeline is called for both infrastructure and application data, ensuring all metrics and traces are normalized before further processing, storage, or analysis." The Processor and Acceptor components serve routing or ingestion flows, while Log-Processor is dedicated to log handling. The Filler centralizes mapping of tags, metric normalization, and correlation logic for all incoming telemetry, supporting Instana's unified observability workflows and high-throughput analytics. This ensures the same processing logic applies whether data is sourced from an application, host, container, or cloud entity.

Reference: IBM Instana Observability Documentation (v1.0.307) — Stream Processor Architectural Diagram and Component Functions.

Question: 53

What happens if the same key is used in both global and alert-specific custom payload configurations in Instana?

A. The alert is canceled due to conflict.

B. The global value overrides the alert-specific value.

C. The alert-specific value overrides the global value.

D. Both values are concatenated.

Answer: C

Explanation:

IBM Instana documents the merge logic of custom payloads for alerts and global configurations very clearly. The rule states: "If the same key is defined in both a global custom payload and an alertspecific payload, the value from the alert-specific payload will override the global value for that key." This ensures alert context management is precise, enabling targeted incident response with the most relevant and high-priority data. There is no concatenation, and no alert cancellation or error is triggered as Instana resolves key collisions silently by giving precedence to the more granular, context-specific setting (alert-level). This verified behavior guarantees custom alert events always contain relevant payloads, supporting accurate automated remediation or escalation.

Reference: IBM Instana Observability Documentation (v1.0.307) — Custom Payload Precedence in Alert Configuration.

Question: 54

Which protocol does an agent use to send the data to the backend?

A. HTTPS

B. FTP

C. SSH

D. NFS

Answer: A

Explanation:

IBM Instana agents use HTTPS, the industry standard secure protocol, to transmit telemetry data to Instana’s backend servers or clusters. Instana documentation says: "All agent-to-backend traffic is encrypted and transmitted via HTTPS, meeting data confidentiality and compliance requirements." The use of HTTPS prevents unauthorized data interception by using strong TLS encryption on every packet exchanged between agent and backend, regardless of whether the deployment is onpremises or SaaS. FTP, SSH, and NFS are protocols for file transfer, system access, or storage mounting but are never used for telemetry transmission in Instana's architecture. Secure HTTP is essential for privacy by design, is

policy-enforced, and supports audit-friendly observability in all supported Instana versions per IBM standards.

Reference: IBM Instana Observability Documentation (v1.0.307) — Agent-to-Backend Communication Security.

Question: 55

Which statement best describes BeeInstana?

A. An operator that can be used to install Instana on Kubernetes

B. It is a metric database used to perform complex metric queries

C. A Kubernetes operator that requires high-performing data stores and a distributed data store cluster.

D. An operator that can be used only on self-hosted deployments that have data stores installed

Answer: C

Explanation:

BeeInstana is identified in Instana's documentation as the core Kubernetes operator driving distributed installation and management of Instana components. The documentation defines: "BeeInstana is a Kubernetes operator that requires robust, high-performing distributed data stores and manages Instana deployment complexity, resource allocation, and scaling within large clusters." By leveraging Kubernetes-native constructs, BeeInstana orchestrates Instana backend, UI, sensors, and streaming components—ensuring reliable, scalable deployments for enterprise settings. The operator orchestrates failover, recovery, and persistent storage management, supporting self-hosted and hybrid installations. While it is associated with metric data handling, its main role is orchestration and operational management based on distributed database infrastructures. Simple operator installation (A, D) does not capture its full role, and describing BeeInstana as only a metric database (B) misrepresents its architectural function in Instana's platform lifecycle.

Reference: IBM Instana Observability Documentation (v1.0.307) — BeeInstana Operator Architecture and Deployment.

Question: 56

What is highly recommended when integrating a few hundred IBM APM v8 agents with Instana?

A. Re-install the IBM APM 8 server.

B. Enable the APM sensor directly on the configuration.yaml file.

C. Increase the JVM memory of the Instana host agent.

D. Install the Instana Agent on multiple servers.

Answer: C

Explanation:

IBM Instana Observability documentation makes it clear that, when integrating many IBM APM v8 agents with a single Instana Agent host, it is highly recommended to increase the JVM memory allocation of the Instana host agent. The official guidance is: "If integrating several hundred APM v8 agents with a single Instana host agent, make sure to increase the Java Virtual Machine (JVM) heap size on the Instana host agent, as the default settings may not suffice for the heightened metric ingestion and processing load." Without this adjustment, the host agent could experience memory pressure, leading to dropped metrics, agent restarts, or degraded ingestion. This step is essential for scaling and ensuring metric reliability in high-volume environments, as detailed in the agent performance tuning and scalability section of IBM's documentation. Other options (A, B, D) do not address the resource requirements driven by metric collection at scale.

Reference: IBM Instana Observability Documentation (v1.0.307) — Agent Sizing, JVM Tuning, and Performance Integration.

Question: 57

What prevents Ansible actions from manual deletion within Instana?

A. There is no name specified on the action.

B. The action is active.

C. Actions have been imported.

D. Default Actions cannot be deleted.

Answer: D

Explanation:

IBM Instana documentation is explicit: some action definitions, including default and built-in (such as Ansible) actions supplied by the platform, cannot be manually deleted by users or admins. It states: "Default Actions—including Ansible integration actions pre-defined by Instana—are protected from manual deletion to ensure availability and platform integrity." This ensures that core automation integrations remain functional and the baseline for remediations, regardless of user error or misconfiguration. Custom or imported actions can be removed, but defaults—tagged as such in the

UI—are non-removable, safeguarding operational continuity and maintaining standardized integrations across manual

and automated workflows. Active status or name presence does not impact deletion ability; it is the default/built-in status (D) that enforces this lock.

Reference: IBM Instana Observability Documentation (v1.0.307) — Action Management and Deletion Permissions.

Question: 58

What is the default folder used to install Instana agent in Linux?

A. /var/lib/instana

B. /opt/instana/agent/etc/

C. /opt/instana/agent

D. /etc/default/instana

Answer: C

Explanation:

IBM Instana installation and agent management documentation specifies: "By default, the Instana agent is installed to the /opt/instana/agent directory on Linux hosts." All primary binaries, configuration, and logs are contained within this root directory, though logs and runtime data are often symlinked or forwarded to standard system directories for rotation. Management scripts and configuration files reside inside this path as well—subdirectories like /etc/ and /data/ are located under /opt/instana/agent. This default directory ensures a consistent and predictable layout across distributions and matches enterprise Linux filesystem standards for third-party agents. Other directories listed (A, B, D) are for data or environment references but are not the root install directory.

Reference: IBM Instana Observability Documentation (v1.0.307) — Linux Agent Installation Directories.

Question: 59

Which configuration file contains Instana server connection details for the host agent?

A. com.instana.agent.main.config.Agent.cfg

B. com.instana.agent.main.sender.Backend.cfg

C. com.instana.agent.main.sender.File.cfg

D. com.instana.agent.main.sender.Server.cfg

Answer: D

Explanation:

The primary configuration file specifying Instana server connection parameters for the host agent is com.instana.agent.main.sender.Server.cfg. The IBM documentation affirms: "The Server.cfg file inside the agent's configuration directory defines backend connection endpoints, ports, and security tokens to communicate with the Instana backend or cluster installation." This file is referenced on agent startup and dictates host-server routing, clustering, authentication, and TLS endpoints. Other config files control agent properties or log shipping, not backend connectivity. Editing Server.cfg is the recommended method for specifying on-premises, private cloud, or SaaS endpoints for all monitored agents.

Reference: IBM Instana Observability Documentation (v1.0.307) — Agent Server Connection Configuration.

Question: 60

Which tool does Instana use to provide geographical data by mapping user IP addresses?

A. GeoLite2 database

B. Universal Geo database

C. Google Maps API

D. Geo Application Service

Answer: A

Explanation:

IBM Instana leverages the open-source and widely recognized GeoLite2 database for mapping user IP addresses to their approximate physical locations in synthetic and real user monitoring scenarios. The documentation details: "Instana provides geographical and location metadata based on the GeoLite2 database, which is regularly updated for improved accuracy and privacy compliance." GeoLite2 is a MaxMind-developed database providing country, city, region, and sometimes ISP-level information from IP addresses. Using an on-premise and regularly-curated geo database ensures no end-user data is ever transmitted to external or third-party web mapping services (such as Google Maps), maintaining strong data privacy and compliance for enterprise customers. Other listed tools are not native to Instana's geo lookup implementation.

Reference: IBM Instana Observability Documentation (v1.0.307) — Geo-IP, Location Services and Privacy.

Question: 61

Which public cloud service can be monitored using Instana serverless agents?

A. Azure Redis Cache

B. AWS Lambda

C. AWS Kinesis

D. AWS SQS

Answer: B

Explanation:

IBM Instana supports direct monitoring of AWS Lambda via serverless-specific agents that bridge trace, metric, and log data between Lambda executions and the Instana backend. The documentation specifies: "Instana's serverless agents enable tracing and monitoring of AWS Lambda functions— including cold start events, performance, and error metrics—correlating invocation traces with upstream and downstream services." Lambda is the only public cloud-native serverless runtime natively and fully integrated with Instana's instrumentation and tracing. Azure Redis Cache, AWS Kinesis, and AWS SQS are data stores or message services, not supported for full serverless agent instrumentation (though they may be monitored via associated infrastructure and integration sensors). Instana's Lambda agent is deployed as a Lambda layer or sidecar, delivering first-class observability for serverless architectures.

Reference: IBM Instana Observability Documentation (v1.0.307) — Serverless Monitoring: AWS Lambda Integration.