

EXAM PREPARATION MATERIAL

Practice files designed to help you understand exam patterns, question styles and the overall assessment format.

IMPORTANT NOTE



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.



Stay connected with our support team
for the latest file updates and exam-related information.

Question: 1

A Performance Analyst is experiencing difficulty with the transaction response time on a node. The Performance Analyst attempts to create a health rule monitoring the business transactions' performance on that node. Which set of data should the Performance Analyst select when creating the health rule?

- A. Node health
- B. Service endpoints
- C. Error Rates
- D. Transaction performance

Answer: D

Explanation:

When a Performance Analyst is faced with issues related to transaction response times on a specific node, focusing on the "Transaction performance" data is most relevant. This is because transaction performance metrics provide direct insights into the business transactions' speed, response times, and execution times, which are critical for identifying and troubleshooting performance issues. By selecting "Transaction performance" when creating a health rule, the analyst can monitor specific metrics such as average response time, slow transaction count, and very slow transaction count, which are directly related to the performance of business transactions on the node in question. **Reference:**

AppDynamics documentation on Health Rules: This section explains how to create and configure health rules to monitor application performance metrics, including transaction performance. AppDynamics documentation on Business Transactions: Provides details on how AppDynamics identifies and monitors business transactions, which are key to understanding application performance.

Question: 2

Developers and operations personnel complain the Controller tracks many irrelevant Java classes and methods. How does a Performance Analyst exclude this unhelpful information?

- A. Configuration > Baselines and create a new dynamic baseline
- B. Edit the mbeans-server.xml file to delete the irrelevant methods
- C. Configuration > Instrumentation > Call Graph Settings and exclude irrelevant packages
- D. Edit the controller-info.xml file to exclude irrelevant methods

Answer: C

Explanation:

To streamline the monitoring process and enhance the relevance of collected data, AppDynamics allows Performance Analysts to exclude certain Java classes and methods from instrumentation. This is done through the "Configuration > Instrumentation > Call Graph Settings" menu, where irrelevant packages and classes can be specified for exclusion. This approach prevents the AppDynamics

agent from tracking unnecessary or irrelevant methods, thus reducing overhead and focusing on the most critical aspects of application performance.

Reference:

AppDynamics documentation on Call Graph Settings: This section details how to configure call graph settings, including the exclusion of specific classes and methods from instrumentation, to optimize performance monitoring.

Question: 3

Which three pieces of information are required when creating a new database collector? (Choose three.)

- A. Primary table name
- B. Database hostname/IP address
- C. System table name
- D. Database Type
- E. Port number

Answer: B, D

Explanation:

When creating a new database collector in AppDynamics, the essential pieces of information required include the "Database hostname/IP address," "Database Type," and "Port number." These details are crucial for establishing a connection to the database and ensuring accurate monitoring. The hostname/IP address identifies the server where the database is hosted, the Database Type specifies the database management system (e.g., MySQL, Oracle), and the Port number is necessary for network communication with the database server.

Reference:

AppDynamics documentation on Database Visibility: This section explains how to set up database collectors, including the required information for successful configuration.

Question: 4

Which two Key Performance Indicators (KPIs) accurately provide insight into server level resource consumption? (Choose two.)

- A. Calls per Minute
- B. Availability
- C. Average Response Time
- D. Application Restarts
- E. CPU %Busy
- F. Memory Used%

Answer: E, F

Explanation:

Key Performance Indicators (KPIs) such as "CPU %Busy" and "Memory Used%" are critical for providing insights into server-level resource consumption. "CPU %Busy" indicates the percentage of time the CPU is actively working on processes, reflecting the server's processing workload. "Memory Used%" shows the proportion of memory utilized, indicating how much of the server's RAM is being

consumed by applications and processes. These KPIs are essential for understanding and managing server performance and resource allocation.

Reference:

AppDynamics documentation on Server Monitoring: Includes information on monitoring server-level metrics, including CPU and memory utilization, to assess resource consumption.

Question: 5

What are two options for sharing snapshot details from the snapshot window? (Choose two.)

- A. Custom Report
- B. Custom Dashboard
- C. Download PDF
- D. Copy to Clipboard
- E. Export

Answer: C, E

Explanation:

To share snapshot details from the snapshot window in AppDynamics, the options to "Download PDF" and "Export" are typically used. "Download PDF" allows users to generate a PDF report of the snapshot details, providing a convenient and portable format for sharing and review. The "Export" option enables the extraction of snapshot data in various formats (e.g., CSV, XML), facilitating further analysis or sharing with other tools and stakeholders.

Reference:

AppDynamics documentation on Snapshots: This section covers how snapshots capture detailed performance data for transactions, including how to share and export this information for collaborative analysis.

Question: 6

In which two features of AppDynamics can Information Points metric data be used? (Choose two.)

- A. Alerting
- B. Analytics
- C. Flow Maps
- D. Custom Dashboards

Answer: A, D

Explanation:

Information Points in AppDynamics are custom metrics that track specific data within your applications, such as method invocations or the value of method arguments. These metrics can be utilized in various features of AppDynamics, most notably in "Alerting" and "Custom Dashboards." Alerting allows you to set up notifications based on the thresholds set for Information Points, ensuring that teams are promptly informed about significant changes or anomalies. Custom Dashboards enable the visualization of Information Points metrics alongside other key performance indicators, providing a comprehensive view of application health and performance tailored to specific needs.

Reference:

AppDynamics documentation on Information Points: Explains how to create and use Information Points to monitor specific business-relevant metrics.

AppDynamics documentation on Alerting: Details the process of setting up health rules and alerts based on various metrics, including those from Information Points.

AppDynamics documentation on Custom Dashboards: Guides on how to create dashboards that incorporate a wide range of metrics, including Information Points, for customized monitoring.

Question: 7

An E-commerce application is built using microservices architecture design with several components. In AppDynamics, how should the Transaction Detection rules be grouped logically?

- A. Use Automatic Discovery
- B. Use Scope
- C. Use Transaction Group
- D. Use Backend Detection

Answer: C

Explanation:

For an e-commerce application built using a microservices architecture, logically grouping Transaction Detection rules can be effectively achieved through "Use Transaction Group." This approach allows for the organization of business transactions into meaningful groups that reflect the application's structure and the interactions between its microservices. By grouping transactions, it becomes easier to monitor, analyze, and troubleshoot the application as a whole and its individual components, enhancing the visibility and management of the application's performance. Reference:

AppDynamics documentation on Business Transactions: Provides insights on how to configure and manage business transactions, including grouping and monitoring strategies.

AppDynamics documentation on Microservices Monitoring: Offers guidance on best practices for monitoring applications designed with microservices architecture, including transaction grouping.

Question: 8

A Performance Analyst has noticed a significant decrease in an application's workload (calls/min) and is trying to identify the root cause. Which option will give the Performance Analyst insight into the behavior of the affected Business Transactions?

- A. Review Business Transactions and enable Show Trends
- B. In Metric Browser Plot Calls/min for the Application
- C. Review Top Business Transactions by Load in the Application Dashboard
- D. Review the Transaction Score for the Application
- E. Review all elements in the Application Flow Map and identify variations in load

Answer: C

Explanation:

When a Performance Analyst observes a significant decrease in an application's workload (calls/min), reviewing the "Top Business Transactions by Load" in the Application Dashboard can provide valuable insights. This feature allows the analyst to quickly identify which business transactions have experienced the most significant changes in load, potentially pinpointing the root cause of the

overall workload decrease. It offers a focused view of the application's performance, highlighting areas that may require further investigation or immediate action.

Reference:

AppDynamics documentation on Application Dashboard: Describes the features and capabilities of the Application Dashboard, including how to view and analyze the top business transactions by load. AppDynamics documentation on Business Transactions: Details the importance of monitoring business transactions and how they can be used to understand application performance trends.

Question: 9

A Performance Analyst is reviewing Business Transactions with an Application team. The Application team would like to increase the Application Business Transaction limit because they need to have visibility into all the different transactions. What should the Performance Analyst do?

- A. Do nothing since the limit is not important during configuration
- B. Increase the limit to the requested value
- C. Increase the limit half way between the current level and the requested value
- D. Focus on the Business Transactions exceeding the limit and why

Answer: D

Explanation:

When an application team requests an increase in the Application Business Transaction limit for visibility purposes, it's crucial for the Performance Analyst to focus on the transactions that are currently exceeding the limit and understand why. This approach helps in identifying whether the limit is being reached due to genuinely essential transactions or if there are redundant, irrelevant, or improperly defined transactions contributing to the limit breach. By analyzing and rationalizing the transactions, the analyst can ensure that only valuable transactions are monitored, optimizing resource usage and maintaining effective observability without necessarily increasing the limit. **Reference:**

AppDynamics documentation on Business Transaction Limits: Discusses the implications of business transaction limits and strategies for managing and optimizing these limits within AppDynamics. AppDynamics Best Practices Guide: Offers recommendations for configuring and managing business transactions, including handling limits and ensuring meaningful transaction monitoring.

Question: 10

Which two views can a Controller-level custom dashboard provide? (Choose two.)

- A. Show a single view of only historical data.
- B. Aggregate and compare data from only a single application on the same Controller
- C. Aggregate and compare data from different applications on the same Controller
- D. A customized view of applications, servers, and database performance data

Answer: CD

Explanation:

Controller-level custom dashboards in AppDynamics have the flexibility to provide views that aggregate and compare data from different applications on the same Controller, as well as offer a customized view of applications, servers, and database performance data. This capability enables a holistic view of the IT environment, facilitating cross-application insights and the ability to correlate

performance across different layers of the technology stack. Such dashboards are invaluable for understanding overarching trends, pinpointing systemic issues, and driving informed decisionmaking across multiple applications and infrastructure components.

Reference:

AppDynamics documentation on Custom Dashboards: Provides detailed instructions on creating and customizing dashboards at the Controller level, including the aggregation and comparison of data from various sources.

AppDynamics documentation on Controller UI: Explains the functionalities available within the

Controller UI, including the creation of comprehensive dashboards that encompass application, server, and database metrics.

Question: 11

What AppDynamics Alert Action does a Performance Analyst need to select to post an AppDynamics event to a third-party collaboration tool?

- A. Make an HTTP Request
- B. Take a thread dump
- C. Create or Update a JIRA Ticket
- D. Run a script or executable on problematic nodes

Answer: A

Explanation:

To post an AppDynamics event to a third-party collaboration tool, the Performance Analyst needs to select the "Make an HTTP Request" alert action. This action allows AppDynamics to send an HTTP request to a specified URL, which can be the endpoint provided by the third-party tool's API. This integration capability enables the automatic posting of events, alerts, and notifications from AppDynamics to collaboration tools, enhancing communication and response times to performance issues.

Reference:

AppDynamics documentation on Alert and Respond: Details the various alert actions available within AppDynamics, including the ability to make HTTP requests to integrate with external systems.

Question: 12

Which item supplements business transaction and transaction analytics data with application data?

- A. Demarcation collectors
- B. Netflow data
- C. Data collectors
- D. Endpoint sensors

Answer: C

Explanation:

Data collectors in AppDynamics supplement business transaction and transaction analytics data with application data by capturing additional information during transaction execution. This can include method arguments, return values, and invocation context, which enriches the transaction data with more detailed application-level insights. This capability is crucial for in-depth performance analysis and troubleshooting, providing a more comprehensive view of application behavior.

Reference:

AppDynamics documentation on Data Collectors: Explains how to configure data collectors to capture and display detailed application data within business transactions.

Question: 13

A Business Transaction was registered and displayed on the Business Transaction Dashboard. It has continuous load on it. After an hour the Business Transaction stopped being displayed in the Business Transaction Dashboard. Which action stopped the display of the Business Transaction?

- A. The Business Transaction Lock Down was enabled an hour ago.
- B. The Business Transaction had been deleted an hour ago from the Business Transaction Dashboard.
- C. The Business Transaction Custom Match Rule was deleted an hour ago.
- D. The Business Transaction had been excluded an hour ago from the Business Transaction Dashboard.

Answer: C

Explanation:

If a Business Transaction stops being displayed on the Business Transaction Dashboard after continuous load, it could be due to the deletion of the Business Transaction Custom Match Rule. Custom Match Rules in AppDynamics are used to define custom business transactions based on specific criteria. If such a rule is deleted, transactions that were previously identified and displayed based on that rule may no longer be recognized as distinct business transactions, leading to their disappearance from the dashboard.

Reference:

AppDynamics documentation on Business Transactions: Provides insights on configuring and managing business transactions, including the use of custom match rules.

Question: 14

Which two methods are used to plot Host CPU and GC Time Spent in a single view? (Choose two.)

- A. Server tab under "Tiers and Nodes"
- B. JMX tab under "Tiers and Nodes"
- C. Memory tab under Tier and Nodes"
- D. Metrics Browser

Answer: BD

Explanation:

To plot Host CPU and GC (Garbage Collection) Time Spent in a single view, the "JMX tab under 'Tiers and Nodes'" and the "Metrics Browser" are the appropriate methods. The JMX tab provides access to Java Management Extensions (JMX) metrics, including those related to GC time. The Metrics Browser allows for the customization and aggregation of various metrics, including Host CPU usage and GC metrics, enabling a combined view of these critical performance indicators.

Reference:

AppDynamics documentation on Monitoring Tiers and Nodes: Discusses the JMX metrics available for Java applications, including garbage collection details.

AppDynamics documentation on the Metrics Browser: Describes how to use the Metrics Browser to view and analyze a wide range of performance metrics.

Question: 15

Which type of Data Collector will capture code data such as method arguments, variables, and return values?

- A. Method Invocation Data Collector
- B. Servlet Container Collector
- C. Transaction Data Collector
- D. URI Data Collector

Answer: A

Explanation:

The "Method Invocation Data Collector" is specifically designed to capture code-level data such as method arguments, variables, and return values. This type of data collector enables deep visibility into the execution of methods within transactions, providing valuable insights into the application's behavior and performance. This detailed level of monitoring is essential for diagnosing complex issues and understanding the inner workings of business transactions.

Reference:

AppDynamics documentation on Data Collectors: Details the types of data collectors available, including Method Invocation Data Collectors, and how they can be used to capture detailed codelevel data.

Question: 16

When using the REST interlace where would a Performance Analyst go to see all of the captured endpoints by service?

- A. Navigate to the Metric Browser then to the specific metric. Copy the REST URL from the Metric Browser and then use a wildcard query to get the metrics for other SEPs.
- B. Navigate to the Applications browser then drill down the requested endpoint with the available Dashboard & Metrics indicators then selected the desired metric.
- C. Navigate to the Metric Browser then to the Consumed Endpoints location. Drill down in the metric browser on the specific endpoint to call out its services.
- D. Access the MySQL database and find all the endpoints in an app. where can drill down to the tier and transactions using the endpoints. Select the appropriate REST feature enabled by the service endpoint.

Answer: C

Explanation:

To view all captured endpoints by service using the REST interface in AppDynamics, a Performance Analyst would navigate to the Metric Browser and proceed to the "Consumed Endpoints" section. From there, drilling down on specific endpoints within the Metric Browser allows the analyst to explore the services associated with each endpoint. This method enables detailed analysis of endpoint metrics, facilitating the monitoring and optimization of service performance.

Reference:

AppDynamics documentation on Metric Browser: Guides on navigating the Metric Browser to access and analyze various application metrics, including those related to endpoints and services.

AppDynamics documentation on REST API: Provides information on how to utilize the REST API for retrieving metrics, including

endpoint data.

Question: 17

Which two functions does the Business Transaction Discovery Feature allow a user to perform? (Choose two.)

- A. Identify servers on which to install an AppDynamics application agent
- B. Identify new Business transactions in the Business Transaction Dashboard
- C. Identify PHP Business Transactions and preview them
- D. Identify potential Business Transactions and preview them
- E. Identify entry points from uninstrumented code

Answer: B, D

Explanation:

The Business Transaction Discovery feature in AppDynamics allows users to identify new and potential business transactions directly within the Business Transaction Dashboard. This feature aids in the continuous monitoring and adaptation of the application's transaction tracking to changing business needs, ensuring comprehensive visibility. Users can also preview these identified transactions, assessing their relevance and impact on the application's performance before deciding to monitor them as official business transactions.

Reference:

AppDynamics documentation on Business Transaction Monitoring: Discusses how to manage and monitor business transactions, including the discovery and preview of new or potential transactions. AppDynamics documentation on Business Transaction Dashboard: Explains the functionalities and features of the Business Transaction Dashboard, including transaction discovery.

Question: 18

A Performance Analyst needs to send a summary report of specific events every two hours. Which feature can enable this automated report summary?

- A. Email Templates
- B. HTTP Request Templates
- C. Email Digests
- D. Email / SMS Configuration

Answer: C

Explanation:

To send a summary report of specific events every two hours, a Performance Analyst can use the "Email Digests" feature in AppDynamics. This feature allows for the automated aggregation and delivery of pertinent event information at predefined intervals, ensuring stakeholders remain informed about the application's performance and significant events without the need for constant manual monitoring.

Reference:

AppDynamics documentation on Alerting and Notifications: Details the configuration of email notifications and digests for events and

alerts, including scheduling and content customization options.

AppDynamics documentation on Reports: Provides information on various reporting features, including the setup of automated email digests.

Question: 19

Which option should a Performance Analyst use in an environment where virtual machines are routinely created and destroyed?

- A. Dappdynamics.agent.auto.node.prefix and -Dappdynamics.cron.vm
- B. Dappdynamics.agent.reuse.nodeName and - Dappdynamics.agent.reuse.nodeName.prefix
- C. Dappdynamics.http.proxyHost and -Dappdynamcs.http.proxy User
- D. Dappdynamics.agent.node.metainfo and -Dappdynamics.agent.nodeName

Answer: B

Explanation:

In environments where virtual machines are frequently created and destroyed, using the agent configuration properties - Dappdynamics.agent.reuse.nodeName and -

Dappdynamics.agent.reuse.nodeName.prefix can be beneficial. These settings allow AppDynamics agents to reuse node names for newly created instances, facilitating consistent monitoring and tracking across dynamic environments. This approach helps maintain clarity in application topology and ensures that performance data remains coherent even as the underlying infrastructure changes.

Reference:

AppDynamics documentation on Agent Configuration: Discusses advanced agent configuration options, including node name reuse, to accommodate dynamic environments such as those with frequently changing virtual machines.

AppDynamics documentation on Infrastructure Monitoring: Offers insights into best practices for monitoring in dynamic and virtualized environments.

Question: 20

Which built-in scheduled report Includes load, response time, and error graphs?

- A. Dashboard Report
- B. All Application Summary
- C. Application Health Report
- D. User Experience: Browser Apps

Answer: C

Explanation:

The "Application Health Report" is a built-in scheduled report in AppDynamics that includes vital metrics such as load, response time, and error graphs. This report provides a comprehensive overview of the application's health and performance, making it an essential tool for Performance Analysts to regularly review and share with stakeholders to ensure the application meets performance standards and user expectations.

Reference:

AppDynamics documentation on Reporting: Covers the types of reports available within AppDynamics, including the Application

Health Report, detailing its contents and how to schedule and customize it.

AppDynamics documentation on Application Performance Management: Provides an overview of key performance indicators and metrics critical for assessing application health, many of which are included in the Application Health Report.

Question: 21

What is the purpose of a transaction snapshot?

- A. To analyze issues with a specific business transaction
- B. To analyze issues only with a transaction flagged as stalled
- C. To analyze issues with a specific instances of a transaction
- D. To analyze issues only with a transaction flagged as slow

Answer: A

Explanation:

A transaction snapshot in AppDynamics is a detailed report of a single execution of a business transaction. Its primary purpose is to analyze issues with a specific business transaction by providing a comprehensive view of the transaction's execution path, including timing, call graphs, and database queries. This allows performance analysts and developers to drill down into individual transactions to diagnose performance bottlenecks, errors, or anomalies.

Reference:

AppDynamics documentation on Transaction Snapshots: Offers detailed guidance on how to capture and analyze transaction snapshots to troubleshoot and optimize application performance.

Question: 22

Which permission allows snapshot archiving?

- A. "Can view data from all applications"
- B. "Configure Business Transactions"
- C. "Agent Advanced Operation"
- D. "Application level - Can create applications"

Answer: C

Explanation:

The permission to enable snapshot archiving in AppDynamics typically falls under advanced operational capabilities, such as those categorized under "Agent Advanced Operation." This permission allows users to archive transaction snapshots for long-term storage and analysis, which is essential for historical performance analysis and auditing purposes.

Reference:

AppDynamics documentation on Role-Based Access Control: Explains the different permissions and roles within AppDynamics, including those related to advanced agent operations and snapshot archiving.

Question: 23

Which values can be used to identify a split exit point?

- A. Tetratation application values
- B. dynamic application values
- C. static application values
- D. variable application values

Answer: C

Explanation:

A split exit point in AppDynamics is identified using static application values. Static values provide a consistent and predictable way to categorize exit points, making it easier to aggregate and analyze similar types of interactions with external services or components.

Reference:

AppDynamics documentation on Exit Points: Provides insights into how exit points are defined and identified within AppDynamics, including the use of static values for split exit points.

Question: 24

A client needs to reference a metric from different applications on an AppDynamics dashboard.

Which of the following is true?

- A. Metrics from different applications can be cross referenced but cannot be combined in a single dashboard
- B. Metrics from different applications cannot be cross referenced or combined on an AppDynamics dashboard
- C. Metrics from different applications cannot be cross referenced on an AppDynamics dashboard
- D. Metrics from different applications can be cross referenced on an AppDynamics dashboard

Answer: D

Explanation:

In AppDynamics, metrics from different applications can indeed be cross-referenced on a single dashboard. This allows for a more holistic view of an organization's application ecosystem, facilitating the comparison and correlation of performance metrics across different applications within the same AppDynamics controller.

Reference:

AppDynamics documentation on Custom Dashboards: Details how to create custom dashboards that can include metrics from multiple applications, enabling cross-referencing and comprehensive performance monitoring.

Question: 25

Which two statements are correct about creating Information Points? (Choose two.)

- A. A wildcard can be used to select multiple methods for an Information Point.
- B. A wildcard can be used to select multiple classes for an Information Point.
- C. An Information Point can be created on the same class and method as a Custom Exit Point.
- D. An Information Point can be created on a class and method that is executed before the Business Transaction entry point.

Answer: A, B

Explanation:

When creating Information Points in AppDynamics, wildcards can be utilized to select multiple methods or classes, thereby broadening the scope of data collection. This feature is particularly useful for monitoring similar operations across different components of an application, enabling a more efficient and scalable approach to gathering custom metrics.

Reference:

AppDynamics documentation on Information Points: Elaborates on the process of setting up Information Points, including the use of wildcards to capture metrics from multiple methods or classes.

Question: 26

Which three Key Performance Indicators (KPIs) are automatically collected when you create an Information Point without adding custom data? (Choose three.)

- A. Maximum Response Time
- B. CPU Time
- C. Minimum Response Time
- D. Response Time
- E. Errors per Minute
- F. Calls per Minute

Answer: DEF

Explanation:

When an Information Point is created in AppDynamics without adding custom data, it automatically collects three key performance indicators (KPIs): Response Time, Errors per Minute, and Calls per Minute. Response Time measures the time taken to complete a transaction or operation, providing insights into application performance. Errors per Minute tracks the number of errors occurring within the scope of the Information Point, helping identify problematic areas. Calls per Minute counts the number of times the specified operation or transaction is invoked, indicating its usage frequency and potential impact on application performance.

Reference:

AppDynamics documentation on Information Points: Discusses the creation and configuration of Information Points, including the default metrics collected.

Question: 27

What are two types of Data Collectors in AppDynamics APM? (Choose two.)

- A. SQL data collectors
- B. HTTP data collectors
- C. Remote Service invocation data collectors
- D. Method invocation data collectors

Answer: A, D

Explanation:

In AppDynamics Application Performance Management (APM), two types of Data Collectors are SQL data collectors and Method invocation data collectors. SQL data collectors capture and record detailed information about SQL queries executed by the application, helping identify slow or inefficient database operations. Method invocation data collectors capture information about

specific method calls within the application code, including execution times and parameters, providing deep insights into code-level performance.

Reference:

AppDynamics documentation on Data Collectors: Provides detailed information on configuring SQL and Method invocation data collectors for in-depth application monitoring.

Question: 28

A Performance Analyst has opened a database collector view. Which tab should be clicked on next to find slow queries?

- A. Schemas
- B. Queries
- C. Live View
- D. Object Browser

Answer: B

Explanation:

When analyzing database performance in AppDynamics, the "Queries" tab within the database collector view is where a Performance Analyst should navigate to find slow queries. This tab presents a list of database queries executed by the application, highlighting those with long execution times, which are potential bottlenecks impacting application performance.

Reference:

AppDynamics documentation on Database Monitoring: Details the database collector feature, including how to use the Queries tab to identify and analyze slow database queries.

Question: 29

What does the Live Preview option in a Business Transaction Discovery Session allow a user to do?

- A. See if new Transaction Discovery Rules work
- B. Provide streaming logs from the application agent
- C. Search classes and methods currently running on the system
- D. Stream entry points in your application

Answer: A

Explanation:

The Live Preview option in a Business Transaction Discovery Session in AppDynamics allows a user to verify the effectiveness of newly configured Transaction Discovery Rules. This feature provides realtime feedback on how these rules identify and categorize business transactions, enabling users to adjust and fine-tune the rules to ensure accurate and meaningful transaction monitoring. **Reference:** AppDynamics documentation on Business Transaction Management: Includes information on Business Transaction Discovery Sessions and the use of Live Preview to validate Transaction Discovery Rules.

Question: 30

What are two differences between creating a Transaction Group using the 'Create Group' action and defining a Transaction Detection rule? (Choose two.)

- A. A Transaction Detection Rule changes the name of the incoming request and reduces the number of overall Business transactions.
- B. Create Transaction Group changes the name of the incoming request and reduces the number of overall Business transactions.
- C. A Transaction Group aggregates the data of multiple transactions.
- D. Transaction Groups create Transaction Detection Rules in a faster easier way.

Answer: CD

Explanation:

Creating a Transaction Group using the 'Create Group' action in AppDynamics allows for the aggregation of data from multiple transactions under a single group, facilitating a consolidated view of similar transactions. This differs from defining a Transaction Detection Rule, which typically focuses on identifying and categorizing individual transactions based on specific criteria. Transaction Groups do not change the names of incoming requests nor reduce the number of overall business transactions; instead, they provide a method for organizing and analyzing related transactions collectively, offering a streamlined approach compared to individually configuring Transaction Detection Rules for each transaction.

Reference:

AppDynamics documentation on Business Transactions: Covers the concept of Transaction Groups and their role in organizing and analyzing transaction data, as well as the process for creating and managing Transaction Detection Rules.

Question: 31

Which health rule violation event will be triggered when a Performance Analyst modifies the existing health rule that is already in critical violation?

- A. Health Rule Violation Ended - Critical
- B. Health Rule Violation Started - Critical
- C. Health Rule Violation Canceled - Critical
- D. Health Rule Violation Continues - Critical

Answer: D

Explanation:

When a Performance Analyst modifies an existing health rule that is already in a state of critical violation, the event that is typically triggered is "Health Rule Violation Continues - Critical." This event indicates that, despite the modification, the health rule is still being violated at a critical level. The system recognizes that the conditions for the health rule violation are still being met and continues to alert accordingly.

Reference:

AppDynamics documentation on Health Rules and Events: Explains the different types of health rule events and the conditions under which they are triggered.

Question: 32

A Performance Analyst received an alert that the Average Response Time is increasing after a new marketing offer was released. Which metric would quickly help the Performance Analyst determine that users may not be critically impacted?

- A. Errors per Minute increase
- B. Calls per Minute decrease
- C. Errors per Minute decrease
- D. Calls per Minute increase

Answer: D

Explanation:

If the Average Response Time is increasing after the release of a new marketing offer, seeing an increase in Calls per Minute could quickly help the Performance Analyst determine that users may not be critically impacted. This increase may indicate that despite the higher average response time, more users are engaging with the application, possibly due to interest generated by the marketing offer. It suggests that the system is handling more load, which could be a factor in the increased response time.

Reference:

AppDynamics documentation on Performance Metrics: Discusses the implications of various key performance indicators, including how an increase in Calls per Minute can be interpreted in the context of application performance.

Question: 33

Which two types of data are collected by Information Points? (Choose two.)

- A. Troubleshooting Metric Data
- B. Business Metric Data
- C. Code Metric Data
- D. Analytics Metric Data

Answer: B, C

Explanation:

Information Points in AppDynamics are designed to collect custom metrics that are specific to the business or code aspects of an application. They can capture Business Metric Data, which pertains to the performance metrics that directly impact business processes, and Code Metric Data, which relates to the performance of specific methods or segments of code within the application. This

allows for targeted monitoring and analysis of areas significant to the business's objectives and technical performance.

Reference:

AppDynamics documentation on Information Points: Provides details on how to set up Information Points to collect custom business and code metrics for in-depth performance analysis.

Question: 34

Refer to the exhibit.



Refer to the exhibit. On which tab will the configured transaction threshold be found?

- A. Call Graph
- B. Slow Calls & Error
- C. DB and Remote Services Calls
- D. Overview
- E. More

Answer: D

Explanation:

In Cisco AppDynamics, the transaction threshold configurations are typically found under the "Overview" tab. This is where you can view the health rule violations and performance baselines that are associated with transaction snapshots, which can include the configured transaction thresholds. These thresholds set the acceptable performance limits for transactions, and when these limits are exceeded, it may trigger health rule violations that are visible on the Overview tab.

Reference:

AppDynamics Documentation on Transaction Snapshots

AppDynamics Documentation on Health Rule Violations

Question: 35

Which feature can be used to determine if a given Java class is visible in AppDynamics?

- A. Tools in Business Transaction Discovery Session
- B. Preview Business transactions in Business Transaction Discovery Session
- C. Use the thread dump feature on the node agent
- D. Use the object instance tracking feature in memory

Answer: B

Explanation:

To determine if a given Java class is visible in AppDynamics, the "Preview Business transactions" feature in a Business Transaction Discovery Session can be used. This feature allows users to validate and preview the detection of business transactions, which includes ensuring that specific Java classes and methods are being correctly identified and monitored by AppDynamics.

Reference:

AppDynamics documentation on Business Transaction Detection: Explains how to conduct a Business Transaction Discovery Session and use the preview feature to validate the visibility and detection of business transactions, including specific Java classes.

Question: 36

Which two things should a Performance Analyst check if an agent has failed to register with the Controller after several minutes? (Choose two.)

- A. total free RAM
- B. network/firewall rules
- C. current CPU load average
- D. free license count

Answer: B, D

Explanation:

If an agent fails to register with the Controller, a Performance Analyst should check the network/firewall rules to ensure that the agent can communicate with the Controller over the network. Additionally, it is crucial to verify the free license count to confirm that there are available licenses for the agent to register. Insufficient licenses or network communication barriers can prevent successful agent registration.

Reference:

AppDynamics documentation on Agent Registration: Describes the prerequisites and troubleshooting steps for agent registration with the Controller.

AppDynamics documentation on Licensing: Provides details on how the licensing model works, including how to check for available licenses.

Question: 37

Within the configuration setting for Slow Transactions, under the "Configure Diagnostic Session Duration and Collection Rate" option, what is the default value pair for the setting "Collect up to snapshots per minute for minutes"?

- A. 2,2
- B. 5,5
- C. 10, 10
- D. 4,4

Answer: C

Explanation:

The default value pair for the setting "Collect up to snapshots per minute for minutes" in the configuration for Slow Transactions is 10, 10. This means that by default, the system is set to collect up to 10 snapshots per minute for a duration of 10 minutes during a diagnostic session.

Reference:

AppDynamics documentation on Transaction Snapshots: Outlines the default settings for diagnostic session duration and collection rates, including the collection of snapshots.

Question: 38

Which option, in addition to "minutes since creation", would a Performance Analyst use while configuring Automatic Cleanup of stale Business Transactions?

- A. Transaction Type
- B. Number of Calls
- C. Regular Expression
- D. Class/Method Filter

Answer: B

Explanation:

When configuring Automatic Cleanup of stale Business Transactions, the "minutes since creation" option is often paired with the "Number of Calls" metric. This setting ensures that business transactions that have not been called within a certain timeframe and have had a negligible number of calls are automatically cleaned up, helping to maintain an organized and relevant set of business transactions in the monitoring system.

Reference:

AppDynamics documentation on Business Transactions: Discusses best practices for managing business transactions, including the automatic cleanup based on time and call volume thresholds.

Question: 39

A Performance Analyst has an urgent need to gather more data for an ongoing issue. What should the Performance Analyst do?

- A. Enable Development Level Monitoring
- B. Browse the Metric Browser for errors
- C. Review the various transaction snapshots to identify anomalies
- D. Carefully monitor the snapshots for errors

Answer: A

Explanation:

If a Performance Analyst has an urgent need to gather more data for an ongoing issue, they should enable Development Level Monitoring. This monitoring level increases the amount of detailed diagnostic data collected by the agent, such as snapshots and transaction traces, which can provide deeper insights into the issue at hand.

Reference:

AppDynamics documentation on Monitoring Levels: Describes the different levels of monitoring available, including Development Level Monitoring and the types of data each level collects.

Question: 40

A Performance Analyst has enabled Development Level Monitoring for an application. For a default configuration, in which scenario will Development Level Monitoring get automatically disabled?

- A. A maximum of 500 calls per minute limit is exceeded, and Maximum heap utilization percentage goes above 90%
- B. A maximum of 1500 calls per minute limit is exceeded, and Maximum heap utilization percentage goes above 90%

- C. A maximum of 1000 calls per minute limit is exceeded, and Maximum heap utilization percentage goes above 95%
- D. A maximum of 2000 calls per minute limit is exceeded, and Maximum heap utilization percentage goes above 95%

Answer: A

Explanation:

Development Level Monitoring in AppDynamics is designed for use in a lower volume, nonproduction environment. For a default configuration, Development Level Monitoring will get automatically disabled if the monitored environment exceeds a threshold, typically a maximum of 1000 calls per minute, and the Maximum heap utilization percentage goes above 95%. These thresholds are in place to prevent excessive overhead in a production environment where such detailed monitoring could impact performance.

Reference:

AppDynamics documentation on Agent Configuration: Provides information on configuration settings for monitoring levels, including when Development Level Monitoring is automatically disabled due to exceeding thresholds.

Question: 41

An AppDynamics deployment has Business Transaction Lock Down turned on. The company has just added an important service to its application and wants to track this service as a unique Business Transaction. What action is needed to achieve this?

- A. Use the Business Transaction Dashboard for the tier-specific All Other Traffic to register the Business Transaction
- B. Modify the Automatic Transaction Discovery rule to include the Web Service Name and Operation Name
- C. Use live preview to identify the Business Transaction and Register it from there
- D. Create a Custom Transaction Match Rule based on the Web Service Name and Operation Name

Answer: D

Explanation:

When Business Transaction Lock Down is enabled in AppDynamics, no new business transactions will be automatically discovered to avoid uncontrolled growth in the number of business transactions. To track a new service as a unique Business Transaction, one needs to create a Custom Match Rule that specifies the criteria for identifying the business transaction. In this case, the Custom Match Rule should be based on the Web Service Name and Operation Name which are the distinguishing characteristics of the new service. This allows for the precise identification and monitoring of the service within the AppDynamics platform.

Reference: AppDynamics documentation on Business Transaction configuration and Custom Match Rules.

Question: 42

How does a Performance Analyst identify if automatic remediation has been taken for a health rule violation?

- A. Expand on the "Description" field to display "Actions Executed".
- B. Review the "Application Dashboard" and review "Actions Executed"
- C. Right-click on "view details" and click on the "Actions Executed" button.
- D. Click on the link inside the Health Rule field and look for the "Affects" tab to display the Executed Actions.

Answer: A

Explanation:

To identify if automatic remediation actions have been taken for a health rule violation in AppDynamics, a Performance Analyst should expand the "Description" field of the health rule violation event. This section will provide details on the actions executed as part of the automatic remediation process. These details help analysts understand the steps taken by the system to mitigate the issue without manual intervention.

Reference: AppDynamics documentation on Health Rule Violations and Automated Actions.

Question: 43

A team of developers deploys new Java servlet code that should create new business transactions in AppDynamics. After applying load on the new code function, there are no new Business Transactions on the Business Transaction Dashboard. Which two options should the developers check in

AppDynamics to make sure the Business Transactions can be discovered?

[Choose two.]

- A. The metric browser to see if the new transactions appear under Business Transaction Performance.
- B. The tier with the new code does not have any rules excluding it.
- C. Auto discovery for service endpoints is turned on.
- D. There is a health rule created to check for transaction performance.
- E. Auto Discovery for servlet is turned on for Java agents.

Answer: B, E

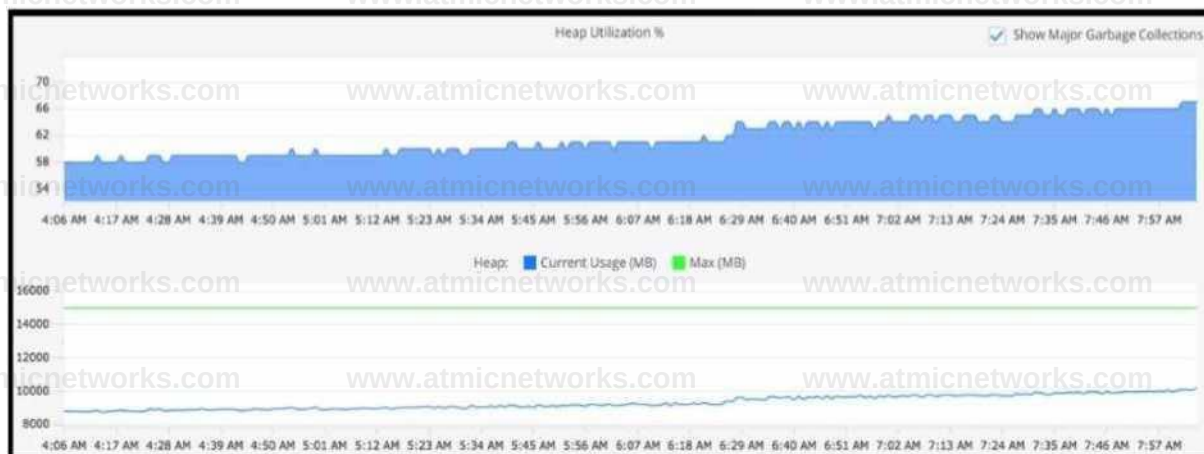
Explanation:

When new business transactions are not appearing on the Business Transaction Dashboard after deploying new code, developers should verify that there are no exclusion rules in place on the tier where the new code was deployed. Additionally, it is crucial to ensure that the Auto Discovery feature for servlets is enabled for Java agents, as this allows AppDynamics to automatically detect and name business transactions based on incoming requests to servlets. Both of these checks are necessary to ensure that new business transactions can be discovered and monitored.

Reference: AppDynamics documentation on Business Transaction detection and Java Agent configuration.

Question: 44

Refer to exhibit.



Refer to the exhibit. Using this heap utilization graph, which method is used to confirm if a memory leak is occurring during a certain time frame?

- A. In metric browser go through Application Infrastructure > Hardware Resources and select Memory Total (MB) and Used (MB)
- B. Refer to the Tiers and Nodes section and into the Memory tab and visualize Heap Utilization (%) and Heap Current Usage (MB) Vs Max (MB)
- C. Refer to the Tiers and Nodes section and into the JMX tab and select JVM > Memory > Heap > Max Available (MB) and Current usage (MB)
- D. In metric browser go through Application Infrastructure > Hardware Resources > Memory Total (MB) and Swap Used (MB)

Answer: B

Explanation:

To confirm if a memory leak is occurring, one should refer to the Tiers and Nodes section of the AppDynamics Controller UI, navigate to the Memory tab, and observe the Heap Utilization over time in relation to the Heap's Current Usage (MB) versus the Maximum (MB) allocated. Consistent growth in heap utilization or an upward trend that does not decrement even after garbage collection indicates a potential memory leak.

Reference: AppDynamics documentation on Memory Leak Detection and Heap Analysis.

Question: 45

Which two match conditions can be added when you configure an Information Point? (Choose two.)

- A. Match based on a regex applied to the method
- B. Match based on the invoked object
- C. Match based on the Business Transaction
- D. Match based on the return value

Answer: A, B

Explanation:

When configuring an Information Point in AppDynamics, you can add match conditions to refine what gets measured. Match conditions based on a regex applied to the method allow you to specify which methods to include based on a regular expression pattern. Matching based on the invoked object allows you to specify which objects' methods are included, filtering the data according to the object type or instance. These conditions help in pinpointing specific methods or objects for which you want to collect runtime information.

Reference: AppDynamics documentation on Information Points and Match Conditions.

Question: 46

What are two examples of backend calls? (Choose two.)

- A. a request coming from a browser
- B. a tier-to-tier request
- C. an asynchronous request
- D. a remote services call

Answer: B, D

Explanation:

Backend calls in AppDynamics are the interactions that an application component has with external components or services. These can include calls to databases, remote service calls, and interactions between different tiers of an application.

A tier-to-tier request refers to any internal call that happens between different tiers (or nodes) within the same application. For example, a web tier calling an API service tier within the same application ecosystem.

A remote services call is an external call from an application to a service that resides outside of the application's environment, like a call to an external web service, REST API, or a third-party service provider.

Reference:

AppDynamics documentation on Backend Detection:

<https://docs.appdynamics.com/21.6/en/application-monitoring/identify-backends>

Question: 47

A Performance Analyst notices an increase in Business Transaction error rate that is much higher than normal alerts. The Performance Analyst can see these are related to the Health Rules defined for the Shopping Service that is monitored with AppDynamics Browser RUM. While troubleshooting the Performance Analyst wants details on which Browsers and Devices are affected. Which section of the Browser App Dashboard will provide this detail for a given time period?

- A. Usage Stats
- B. Sessions
- C. Pages & Ajax Requests
- D. Overview

Answer: A

Explanation:

In the Browser Real User Monitoring (RUM) Dashboard, the Usage Stats section provides insights into user demographics, including the types of browsers and devices they are using. This section would help the Performance Analyst understand which browsers and devices are affected during a specific time period when there has been an increase in the Business Transaction error rate.

Reference:

AppDynamics documentation on Browser RUM: <https://docs.appdynamics.com/21.6/en/end-user-monitoring/browser-monitoring/browser-real-user-monitoring>

Question: 48

A development team responsible for the front-end shopping application has asked to receive an email every time the Java container thread count exceeds 25. Which alert and response capabilities settings will provide the email?

- A. Node Health - JMX Thread Pools (> Specific Value) + Notification Action (Send an Email)

B. Node Health - Thread Pools (> Specific Value) + Notification Action (Send an Email) C. Node Health - Thread Pools (> Specific Value) + Notification Action (Custom Action) D. Node Health - JMX Thread Pools (> Baseline) + Notification Action (Send an Email)

Answer: B

Explanation:

In AppDynamics, you can create health rules to monitor various metrics and set up actions based on the thresholds defined for these metrics. For monitoring Java container thread counts, you can set a health rule based on Node Health - specifically on thread pools - to trigger when the thread count exceeds a specific value. The Notification Action can then be configured to send an email to the development team whenever this threshold is breached.

Reference:

AppDynamics documentation on Health Rules:

<https://docs.appdynamics.com/21.6/en/infrastructure-visibility/health-rules>

Question: 49

With what frequency are widgets updated during a war room scenario?

- A. Near real-time
- B. Every 5 minutes
- C. Every 10 minutes
- D. Every 60 minutes

Answer: A

Explanation:

During a war room scenario, which is a real-time troubleshooting session, widgets in AppDynamics dashboards update in near real-time. This allows teams to observe the immediate impact of changes and identify issues as they occur.

Reference:

AppDynamics documentation on War Rooms:

Question: 50

Which three data points can be located by drilling down into a JDBC exit call for an Oracle backend? (Choose three.)

- A. Query type
- B. Statement type
- C. Query Id
- D. Weight %
- E. Originating node
- F. %Time

Answer: ABE

Explanation:

When drilling down into a JDBC exit call for an Oracle backend, AppDynamics provides detailed information about the call. The

data points include:

Query type, which can indicate whether it's a SELECT, INSERT, UPDATE, or DELETE statement.

Statement type, which describes the nature of the SQL statement being executed.

Originating node, which identifies the node from which the JDBC call originated.

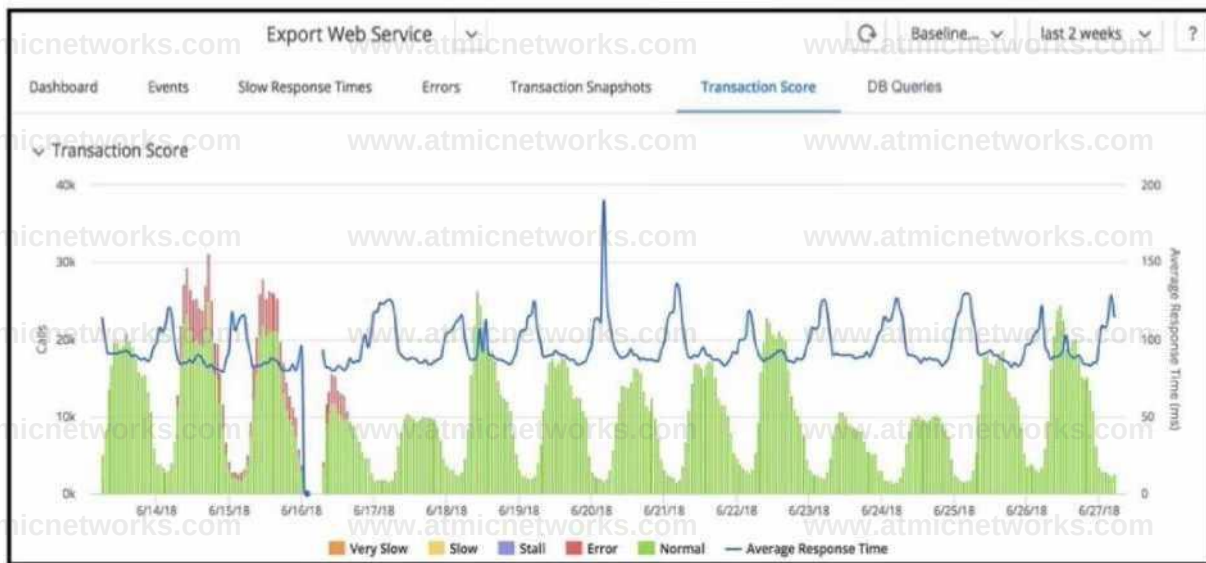
These data points help in understanding the nature and source of database operations, which can be critical for performance analysis and troubleshooting.

Reference:

AppDynamics documentation on Database Monitoring:

Question: 51

Refer to exhibit.



Refer to the exhibit. When looking at the Transaction Score for a specific transaction, how are errors in the transaction identified?

- A. Set the time range and drill down into the snapshots in the Error tab.
- B. Set the time range and examine the Slow Response Times tab.
- C. Set the time range and examine the dashboard for errors.
- D. Set the time range and drill down into the snapshots in the Events tab.

Answer: A

Explanation:

Errors in a transaction are identified by examining the snapshots that capture the problematic transactions. By setting the appropriate time range, a Performance Analyst can drill down into the snapshots within the Error tab to identify and analyze errors. These snapshots provide detailed diagnostic information, such as stack traces, slow SQL queries, and error logs, which are vital for pinpointing the root cause of transaction errors.

Reference:

AppDynamics documentation on Transaction Snapshots:

Question: 52

A Performance Analyst needs to define a set of Key Performance Indicators (KPIs) from a group of select metrics. The required performance information resides within the Transaction Analytics data set. Which method will accomplish this task?

- A. Experience Level Management
- B. Search Queries
- C. Business Outcome Milestones
- D. Metric Explorer

Answer: D

Explanation:

The Metric Explorer in AppDynamics allows Performance Analysts to define and visualize Key Performance Indicators (KPIs) from a selection of metrics. By accessing the Transaction Analytics data set, analysts can create custom dashboards that focus on the metrics they've determined to be critical KPIs for their application's performance.

Reference:

AppDynamics documentation on Metric Explorer:

Question: 53

Which tab within the Application Dashboard displays performance trends for each of Snapshots, Average Response Time, and Events within one central view?

- A. Application Flow Map
- B. Dashboard
- C. Events
- D. Transaction Score
- E. Network Dashboard

Answer: D

Explanation:

The Transaction Score tab within the Application Dashboard is designed to display performance trends across various metrics including Snapshots, Average Response Time, and Events. It gives a comprehensive view of the transaction performance, providing a score that reflects the health and reliability of transactions over time.

Reference:

AppDynamics documentation on Application Dashboard:

Question: 54

What is the Node limit of the maximum Service Endpoints per node?

- A. 50
- B. 100

- C. 250
- D. 1000

Answer: B

Explanation:

AppDynamics imposes a limit on the number of Service Endpoints that can be registered per node to ensure manageable performance and overhead. The limit per node is set to 100 Service Endpoints, which is a balance between providing detailed monitoring and maintaining application performance. Reference:

AppDynamics documentation on Service Endpoints:

<https://docs.appdynamics.com/latest/en/application-monitoring/monitor-service-endpoints>

Question: 55

A Performance Analyst has a third-party event management system API and wants to use the event management system to send alerts to Level 1.

Which actions are available in the AppDynamics Actions Menu to set this up?

- A. Send an SMS
- B. HTTP Request
- C. Remediation
- D. Send an Email

Answer: B

Explanation:

When integrating with a third-party event management system, the Performance Analyst can set up an HTTP Request action within AppDynamics. This allows for the creation of a custom HTTP request to the event management system's API, which can be configured to send alerts to Level 1 support or any other designated recipient.

Question: 56

Which AppDynamics option will allow a Performance Analyst to compare and filter related servers using custom metadata within your environment?

- A. Server Tagging
- B. Server Visibility
- C. Standalone Machine Agent
- D. Dynamic Monitoring Mode

Answer: A

Explanation:

Server Tagging in AppDynamics allows a Performance Analyst to compare and filter related servers by using custom metadata. This feature enables users to group and organize their servers based on criteria that make sense for their environment, such as role, location, environment type, or any other custom metadata.

Reference:

AppDynamics documentation on Server Tagging:

<https://docs.appdynamics.com/latest/en/infrastructure-visibility/infrastructure-visibility-settings/server-tags>

Question: 57

When should URI Segments be used in Transaction Detection rules?

- A. When identifying business transactions using particular parts of the SMTP request
- B. When the application uses message-oriented technologies
- C. When the application is a Web-Oriented Technology
- D. When the agent names the business transaction for the Web service name plus the operation name

Answer: C

Explanation:

URI Segments should be used in Transaction Detection rules when the application is web-oriented. Web-oriented applications often have meaningful information in the URI that can identify different business transactions. URI Segments can help in defining patterns that match specific parts of the URI to distinguish different transactions.

Reference:

AppDynamics documentation on Transaction Detection:

<https://docs.appdynamics.com/latest/en/application-monitoring/business-transactions>

Question: 58

Which statement about Service End Points and Business Transactions is true?

- A. Service End Points and Business Transactions can both be renamed by right clicking and selecting 'rename'.
- B. Service EndPoints and Business Transactions can both configure data collectors.
- C. Service EndPoints and Business Transactions can both be split.
- D. Service EndPoints and Business Transactions both provide percentile metrics.

Answer: B

Explanation:

Both Service Endpoints and Business Transactions in AppDynamics have the capability to configure data collectors. Data collectors are used to gather detailed information about transactions or endpoints, such as method parameters, return values, and SQL statements.

Reference:

AppDynamics documentation on Data Collectors:

<https://docs.appdynamics.com/latest/en/application-monitoring/configure-data-collection>

Question: 59

Which Application Dashboard view categorizes transactions by load, response time, errors, slow transactions, and stalled transactions in a single aggregated value for a specific time range?

- A. Transaction Snapshots
- B. Top Business Transactions
- C. Machine Snapshots
- D. Transaction Score

Answer: D

Explanation:

The Transaction Score view in the Application Dashboard categorizes transactions by load, response time, errors, slow transactions, and stalled transactions. It provides an aggregated value for a specific time range, giving an at-a-glance indication of the health and performance of business transactions. Reference:

AppDynamics documentation on Transaction Score:

<https://docs.appdynamics.com/latest/en/application-monitoring/application-dashboard>

Question: 60

What must a Performance Analyst first configure in order to monitor end-to-end latency performance metrics in AppDynamics?

- A. additional demarcator methods
- B. define the end-to-end latency transaction
- C. slow end-point identity markers
- D. endpoints for asynchronous transactions

Answer: B

Explanation:

To monitor end-to-end latency performance metrics, a Performance Analyst must first define the end-to-end latency transaction. This involves configuring the business transactions to include the necessary entry and exit points that capture the complete flow of the transaction, thereby enabling the measurement of the total latency.

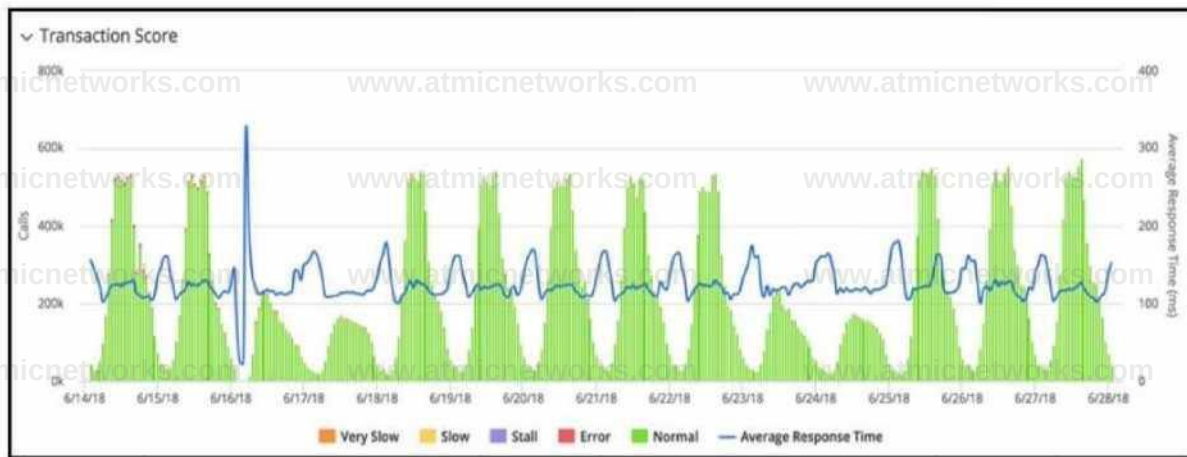
Reference:

AppDynamics documentation on Business Transactions:

<https://docs.appdynamics.com/latest/en/application-monitoring/business-transactions>

Question: 61

Refer to Exhibit.



Refer to the exhibit. The transaction score in the graphic displays an interesting performance pattern outside of business hours on 6/16/18. Which additional performance anomaly should be of most interest to a Performance Analyst?

- A. Elevated Response Time at 6/19/18
- B. Elevated Response Time at 6/25/18
- C. Lower than usual volume at 6/17/18
- D. Low volume at 6/15/18

Answer: A

Explanation:

The performance pattern outside of business hours on 6/16/18 that would be of most interest to a Performance Analyst is the elevated response time, as indicated on 6/19/18. This is because it shows a significant spike in response time, which could indicate a performance issue that needs to be addressed. Anomalies in response time can often be more indicative of underlying problems than changes in call volume, especially when they occur outside of expected peak periods.

Reference:

AppDynamics documentation on Transaction Score:

<https://docs.appdynamics.com/latest/en/application-monitoring/application-dashboard/transaction-score>

Question: 62

The performance impact on the _____ would lead a Performance Analyst to limit the duration and frequency of automatic diagnostic sessions.

- A. Application
- B. Controller
- C. Network
- D. Operating System

Answer: A

Explanation:

The primary concern for a Performance Analyst when considering the impact of automatic diagnostic sessions is the application itself. Intensive diagnostic sessions can be resource-heavy, potentially affecting the application's performance. Therefore, it is often necessary to limit the duration and frequency of these sessions to ensure they do not negatively impact the application's end-user experience.

Reference:

AppDynamics documentation on Diagnostic Sessions:

<https://docs.appdynamics.com/latest/en/troubleshoot/diagnostic-sessions>

Question: 63

Which three reports can be scheduled for periodic generation and delivery to email recipients? (Choose three.)

- A. Ad Hoc Data
- B. Application Health
- C. Error Trends
- D. Custom Dashboard
- E. Controller Audit
- F. Tier List

Answer: B, C, D

Explanation:

AppDynamics allows for several reports to be scheduled for periodic generation and sent to email recipients. Among these, Application Health, Error Trends, and Custom Dashboards are types of reports that can be automatically generated and delivered, providing regular insights into the application's performance and issues.

Reference:

AppDynamics documentation on Reports: <https://docs.appdynamics.com/latest/en/application-monitoring/reporting>

Question: 64

Which statement is correct regarding controller-level and tier/node-level dashboards?

- A. The Performance Analyst can associate a controller-level dashboard with a tier or node through the My Dashboards tab
- B. From the controller-level dashboards list the Performance Analyst can access any tier/node-level dashboards outside the application in which they were created.
- C. Controller-level and tier/node-level dashboards are two separate sets. The Performance Analyst cannot cross-reference between these dashboards.
- D. Controller-level and tier/node-level dashboards are not scoped to be separate entities.

Answer: C

Explanation:

Controller-level and tier/node-level dashboards in AppDynamics are treated as separate entities. They are scoped differently, with controller-level dashboards providing a global view across the entire AppDynamics domain, and tier/node-level dashboards being specific to particular tiers or nodes within an application. Performance Analysts do not have the ability to cross-reference directly between these two sets of dashboards within the AppDynamics UI.

Reference:

AppDynamics documentation on Dashboards: <https://docs.appdynamics.com/latest/en/application-monitoring/custom-dashboards>