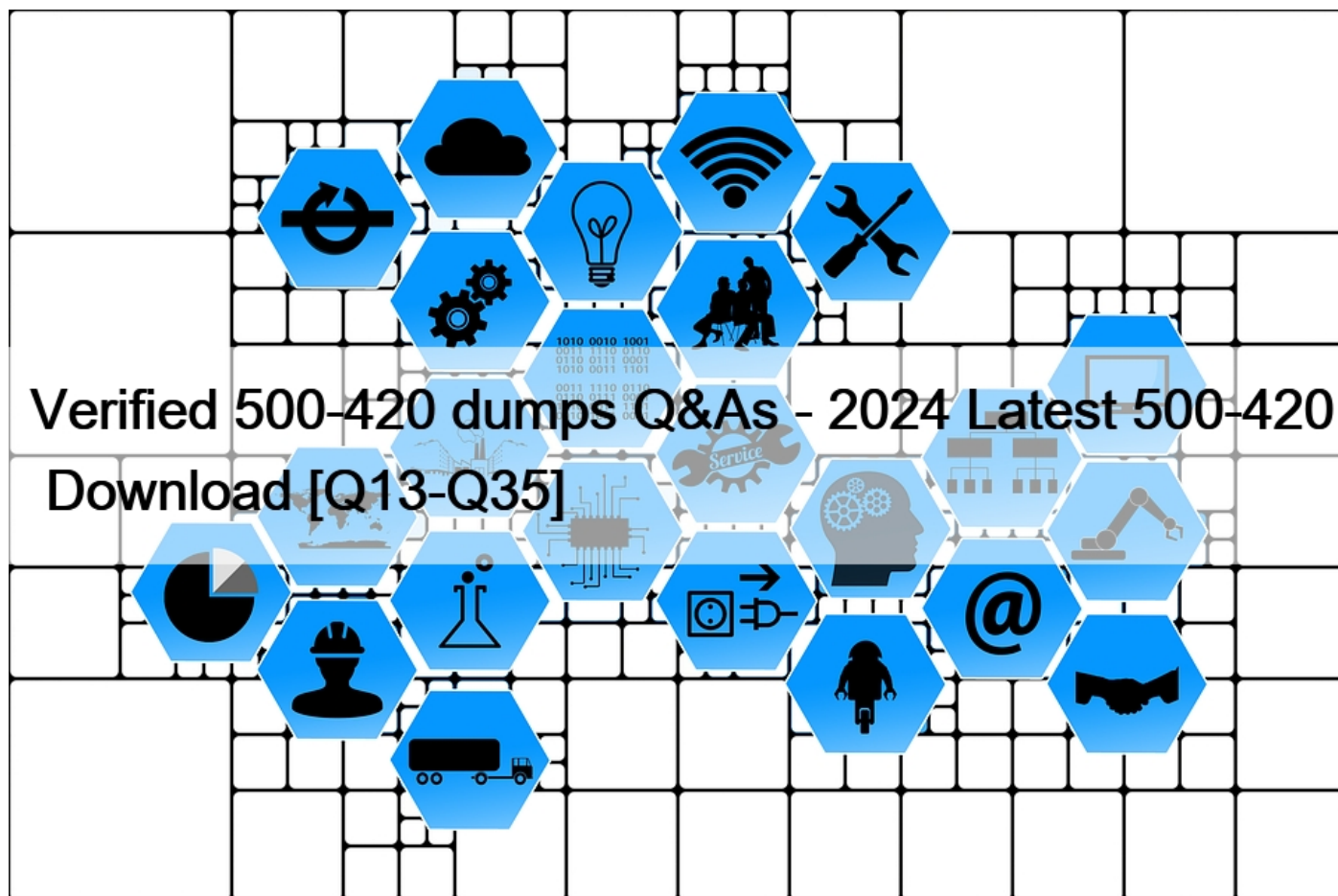


Verified 500-420 dumps Q&As - 2024 Latest 500-420 Download [Q13-Q35]



Verified 500-420 dumps Q&As - 2024 Latest 500-420 Download
Updated 100% Cover Real 500-420 Exam Questions - 100% Pass Guarantee

NO.13 Which two functions does the Business Transaction Discovery Feature allow a user to perform? (Choose two.)

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

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.

References:

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.

NO.14 Which two types of data are collected by Information Points? (Choose two.)

- * Troubleshooting Metric Data
- * Business Metric Data
- * Code Metric Data
- * Analytics Metric Data

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.

References:

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.

NO.15 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?

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

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

NO.16 Which type of Data Collector will capture code data such as method arguments, variables, and return values?

- * Method Invocation Data Collector
- * Transaction Data Collector
- * Servlet Container Collector
- * URI Data Collector

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.

References:

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 code-level data.

NO.17 What is the Node limit of the maximum Service Endpoints per node?

- * 50

- * 100
- * 250
- * 1000

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.

References:

AppDynamics documentation on Service Endpoints:

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

NO.18 What are two types of Data Collectors in AppDynamics APM? (Choose two.)

- * SQL data collectors
- * HTTP data collectors
- * Remote Service invocation data collectors
- * Method invocation data collectors

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.

References:

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

NO.19 Which two Key Performance Indicators (KPIs) accurately provide insight into server level resource consumption? (Choose two.)

- * Calls per Minute
- * Availability
- * Average Response Time
- * Application Restarts
- * CPU %Busy
- * Memory Used%

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.

References:

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

NO.20 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?

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

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.

References:

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.

NO.21 Which two match conditions can be added when you configure an Information Point? (Choose two.)

- * Match based on a regex applied to the method
- * Match based on the invoked object
- * Match based on the Business Transaction
- * Match based on the return value

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. References: AppDynamics documentation on Information Points and Match Conditions.

NO.22 Which two methods are used to plot Host CPU and GC Time Spent in a single view? (Choose two.)

- * Server tab under Tiers and Nodes
- * JMX tab under Tiers and Nodes
- * Memory tab under Tier and Nodes
- * Metrics Browser

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.

References:

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.

NO.23 Which item supplements business transaction and transaction analytics data with application data?

- * Demarcation collectors
- * Netflow data
- * Data collectors
- * Endpoint sensors

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.

References:

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

NO.24 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?

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

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.

References:

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.

NO.25 Which statement is correct regarding controller-level and tier/node-level dashboards?

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

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.

References:

AppDynamics documentation on Dashboards:

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

NO.26 What is the purpose of a transaction snapshot?

- * To analyze issues with a specific business transaction
- * To analyze issues only with a transaction flagged as stalled
- * To analyze issues with a specific instances of a transaction
- * To analyze issues only with a transaction flagged as slow

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.

References:

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

NO.27 A Performance Analyst has an urgent need to gather more data for an ongoing issue. What should the Performance Analyst do?

- * Enable Development Level Monitoring
- * Browse the Metric Browser for errors
- * Review the various transaction snapshots to identify anomalies
- * Carefully monitor the snapshots for errors

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.

References:

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

NO.28 What AppDynamics Alert Action does a Performance Analyst need to select to post an AppDynamics event to a third-party collaboration tool?

- * Make an HTTP Request
- * Take a thread dump
- * Create or Update a JIRA Ticket
- * Run a script or executable on problematic nodes

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.

References:

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.

NO.29 Which three data points can be located by drilling down into a JDBC exit call for an Oracle backend? (Choose three.)

- * Query type
- * Statement type
- * Query Id
- * Weight %
- * Originating node
- * %Time

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.

References:

AppDynamics documentation on Database Monitoring:

NO.30 Which built-in scheduled report Includes load, response time, and error graphs?

- * Dashboard Report
- * All Application Summary
- * Application Health Report
- * User Experience: Browser Apps

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.

References:

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.

NO.31 The performance impact on the _____ would lead a Performance Analyst to limit the duration and frequency of automatic diagnostic sessions.

- * Application

- * Controller
- * Network
- * Operating System

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.

References:

AppDynamics documentation on Diagnostic Sessions:

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

NO.32 A Performance Analyst needs to send a summary report of specific events every two hours. Which feature can enable this automated report summary?

- * Email Templates
- * HTTP Request Templates
- * Email Digests
- * Email / SMS Configuration

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.

References:

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.

NO.33 How does a Performance Analyst identify if automatic remediation has been taken for a health rule violation?

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

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

NO.34 What are two differences between creating a Transaction Group using the **Create Group** action and defining a Transaction Detection rule? (Choose two.)

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

NO.35 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?

- * Usage Stats
- * Sessions
- * Pages & Ajax Requests
- * Overview

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.

References:

AppDynamics documentation on Browser RUM:

<https://docs.appdynamics.com/21.6/en/end-user-monitoring/browser-monitoring/browser-real-user-monitoring>

Use Real Dumps - 100% Free 500-420 Exam Dumps: https://www.braindumpsit.com/500-420_real-exam.html