



## Free Cisco 500-420 CAAPA Practice Questions

Shared by Odom on 13-08-2026

**For More Free Questions and Preparation Resources**

Check the Links on Last Page



## Question 1

---

Question Type: MultipleChoice

---

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?

Options:

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

P2P  
exams

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.

AppDynamics documentation on Transaction Score:

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

P2P  
exams

## Question 2

---

Question Type: MultipleChoice

---

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

Options:

- 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 real-time 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.

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 3

---

Question Type: MultipleChoice

---

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

Options:

---

- A- Tetration 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.

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 4

---

Question Type: MultipleChoice

---

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?

### Options:

- 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.

AppDynamics documentation on Health Rules:

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

## Question 5

---

Question Type: MultipleChoice

---

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

**Options:**

---

- 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.

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.

**Question 6**

---

**Question Type:** MultipleChoice

---

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

**Options:**

---

- 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.

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

## Question 7

---

Question Type: MultipleChoice

---

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

Options:

---

- 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.

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.



To Get Premium Files for 500-420 Visit

<https://www.p2pexams.com/products/500-420>

For More Free Questions Visit

<https://www.p2pexams.com/cisco/pdf/500-420>

**20%**  
**DISCOUNT**

**P2P**  
exams