

- **Vendor: Cisco**
- **Exam Code: 500-430**
- **Exam Name: Cisco AppDynamics Professional Implementer**
- **New Updated Questions from [Braindump2go](#)**
- **(Updated in [March/2026](#))**

[Visit Braindump2go and Download Full Version 500-430 Exam Dumps](#)

QUESTION 1

What are two valid reasons for using the REST API to retrieve health rule violations? (Choose two.)

- A. For updating an AppDynamics dashboard
- B. For determining which actions have been executed
- C. When searching for historical events
- D. For sending emails
- E. When pushing events to the Event Management System is NOT possible

Answer: BC

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the REST API for health rule violations allows you to retrieve information about the health rule violations that occurred in a specified time range for a given application. You can use the REST API for health rule violations for the following valid reasons:
For determining which actions have been executed (B): The REST API response includes the details of the actions that were triggered by the health rule violation, such as email, SMS, HTTP request, or custom action. You can use this information to verify if the actions were executed successfully, or to troubleshoot any issues with the action execution.
When searching for historical events (C): The REST API allows you to specify a custom time range for retrieving the health rule violations, such as BEFORE_TIME, AFTER_TIME, BETWEEN_TIMES, or BEFORE_NOW. You can use this feature to search for historical events that occurred in the past, or to analyze the trends and patterns of the health rule violations over time.

QUESTION 2

Which AppDynamics Controller port(s) does the EUM Server require access to in a configuration where the EUM Server and Controller are on separate hosts (split-host configuration)?

- A. dedicated EUM HTTP(s) ports
- B. GlassFish administration port
- C. Controller database and HTTP(s) ports
- D. Controller primary HTTP(s) port

Answer: D

Explanation:

In a split-host configuration, where the EUM Server and Controller are on separate hosts, the EUM Server requires access to the Controller primary HTTP(s) port. This is because the EUM Server needs to communicate with the Controller API server to send data and receive configuration information. The default primary HTTP port for the Controller is 8090 and the default primary HTTPS port is 8181. The dedicated EUM HTTP(s) ports are used by the EUM agents to send data to the EUM Server, not by the EUM Server to access the Controller. The GlassFish administration port is used to access the Controller Admin Console, not by the EUM Server. The Controller database port is used by the Controller to connect to the MySQL database, not by the EUM Server.

QUESTION 3

Which two preparatory tasks are required prior to installing an AppDynamics Controller on Linux? (Choose two.)

- A. Install JRE.
- B. Ensure that MySQL port (3388) is opened.
- C. Install SSH.
- D. Install libaio.
- E. Verify that sufficient temporary (tmp) space is available (at least 1 GB).

Answer: DE

Explanation:

Before installing an AppDynamics Controller on Linux, you need to perform some preparatory tasks to ensure the system meets the requirements and the installation runs smoothly. Two of these tasks are:

Install libaio on the host machine if it does not already have it installed. This library facilitates asynchronous I/O operations on the system, which are required by the Controller. You can use the package manager of your Linux distribution to install libaio, such as yum or apt-get. For example, on CentOS, you can run `yum install libaio`.

Verify that you have enough temporary (tmp) space available on the system, at least 1 GB. The Controller installation uses the tmp space to extract and install the software components. You can check the tmp space by running `df -h /tmp`. If the tmp space is insufficient, you can either free up some space by deleting unnecessary files, or specify a different temporary directory for the installation by passing the `-Djava.io.tmpdir` parameter to the installer. Other preparatory tasks include verifying the user account permissions, configuring the virus scanners, installing the netstat network utility, and setting the file descriptor limit.

QUESTION 4

The AppDynamics Controller is instrumented by an internal, out-of-the-box, AppDynamics Java agent. Which account and user name are used to connect to the Controller to view the information provided by the internal AppDynamics agent?

- A. The account is 'root' and the user is 'admin'.
- B. The account is 'customer!' and the user is 'root'.
- C. The account is 'system' and the user is 'root'.
- D. The account is 'internal' and the user is 'admin'.

Answer: C

Explanation:

The AppDynamics Controller is instrumented by an internal, out-of-the-box, AppDynamics Java agent that monitors the performance and health of the Controller itself. To access the information provided by the internal agent, you need to log in to the Controller UI with the following credentials:

Account = system

Username = root

Password = <root_user_password>

The system account is a special account that is used only for internal monitoring and troubleshooting purposes. It is not visible in the normal Controller UI and requires a special URL to access it. The root user is the default administrator user for the system account and has the same password as the admin user for the customer account.

QUESTION 5

What is the correct method to perform a NET Agent upgrade?

- A. Perform the agent upgrade on the application server host by running the MSI Installer Package.
- B. Perform the agent upgrade on a remote server host by using the AppDynamics Controller REST API.
- C. Perform the agent upgrade on the application server host by running the Agent Configuration Utility.
- D. Perform the agent upgrade via the AppDynamics Controller UI.

Answer: A

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the correct method to perform a NET Agent upgrade is to perform the agent upgrade on the application server host by running the MSI Installer Package. This method will install updated agent files and maintain legacy configurations. You do not need to uninstall the old agent first when you upgrade from the NET Agent ≥ 3.9 , except for patch releases. You need to stop IIS, instrumented Windows services, and instrumented standalone applications before running the MSI Installer Package. You also need to launch an elevated command prompt with full administrator privileges and specify your account access key for single-tenant Controller accounts. After the installation, you need to restart Windows services and standalone applications.

QUESTION 6

Which two AppDynamics user permissions should an administrator configure in order to protect private data such as credit card numbers? (Choose two.)

- A. Memory Monitoring
- B. Error Detection
- C. SQL Bind Variables
- D. Monitoring Level
- E. Diagnostic Data Collectors
- F. Policies

Answer: CE

Explanation:

AppDynamics can collect sensitive data such as credit card numbers from various sources, such as SQL queries, error messages, HTTP headers, cookies, and so on. To protect this data from unauthorized access or exposure, an administrator should configure the user permissions for the following features:

SQL Bind Variables: This feature enables the agents to capture the values of the parameters that are passed to SQL queries. These values can contain sensitive data such as credit card numbers, passwords, or personal information. To prevent this data from being stored or displayed in the Controller UI, the administrator should disable the Capture SQL Bind Values permission for the users who do not need to see this data. Alternatively, the administrator can enable the Mask SQL Bind Values permission, which replaces the values with asterisks (*) in the UI. The administrator can also configure the agent properties to exclude or mask certain bind variables based on patterns or keywords.

Diagnostic Data Collectors: This feature enables the agents to collect additional data from the application code, such as method arguments, return values, HTTP headers, cookies, and so on. These data can also contain sensitive information that should not be exposed to unauthorized users. To control the access to this data, the administrator should disable the View Diagnostic Data Collectors permission for the users who do not need to see this data. The administrator can also configure the agent properties to exclude or mask certain data collectors based on patterns or keywords. Other features that can collect sensitive data and require user permissions are Error Detection, Memory Monitoring, and Policies. However, these features are less likely to capture credit card numbers than SQL Bind Variables and Diagnostic Data Collectors.

QUESTION 7

What are three valid reasons to use the AppDynamics REST API to retrieve metrics? (Choose three.)

- A. to archive 1-minute granularity data
- B. to create a custom report to be run monthly to show average node availability
- C. to calculate a new metric based on two existing metrics
- D. to evaluate health rules
- E. to retrieve health rule violations
- F. to create an alert using a baseline to send to an internal ticketing system

Answer: ACE

Explanation:

The AppDynamics REST API to retrieve metrics allows you to get values generated for metrics by specifying the path of the metric and the time frame for the data. Some of the valid reasons to use this API are:

To archive 1-minute granularity data. The AppDynamics Controller stores metric data at different levels of granularity depending on the retention period. For example, it stores 1-minute granularity data for 8 days, 10-minute granularity data for 32 days, and 1-hour granularity data for 365 days. If you want to archive the 1-minute granularity data for longer than 8 days, you can use the API to retrieve and store the data in an external database or file system. To

calculate a new metric based on two existing metrics. The AppDynamics Controller provides some built-in metrics such as average response time, calls per minute, errors per minute, etc. However, you may want to calculate a new metric that is not available in the Controller, such as the ratio of errors to calls, or the percentage of slow transactions. You can use the API to retrieve the values of the existing metrics and perform the calculation using your own logic or formula. To retrieve health rule violations. Health rules are the rules that define the performance and availability thresholds for your application components. When a health rule is violated, the AppDynamics Controller generates an event and optionally triggers a policy action. You can use the API to retrieve the list of health rule violations for a given application, time range, and severity level. This can help you monitor and troubleshoot the health of your application and take corrective actions if needed.

QUESTION 8

What are two recommendations for servers in an Events Service Cluster? (Choose two.)

- A. Should be installed using the same user account,
- B. Should have eight or more CPU cores,
- C. Should be on the same local network,
- D. Should have identical hardware specifications.
- E. Should be running the same operating system version

Answer: CD

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the two recommendations for servers in an Events Service Cluster are:

Should be on the same local network(C): This is a valid recommendation because the Events Service Cluster is based on Apache Cassandra, which is a distributed database that relies on network communication between the nodes. Having the servers on the same local network reduces the network latency and improves the performance and reliability of the cluster. The network bandwidth should be at least 1 Gbps, and the network firewall should allow the required ports for the Events Service Cluster.

Should have identical hardware specifications (D): This is a valid recommendation because the Events Service Cluster is horizontally scalable, which means that the nodes share the data storage and processing load equally. Having identical hardware specifications for the servers ensures that the cluster is balanced and efficient, and avoids performance bottlenecks or failures due to hardware differences. The hardware specifications should meet the minimum requirements for the Events Service Cluster, such as CPU cores, RAM, disk space, and disk type.

QUESTION 9

Which two statements are true about instrumenting standalone Windows services with NET Agent? (Choose two.)

- A. AppDynamics .NET Agent does NOT support instrumenting process running multiple App domains.
- B. AppDynamics .NET Agent automatically discovers all the Windows services to be instrumented.
- C. AppDynamics .NET Agent can instrument both 32-bit as well 64-bit processes.
- D. AppDynamics .NET Agent requires that the Windows services is running under the "App.pool identity user" account.
- E. AppDynamics .NET Agent supports instrumentation of multiple instances of the same application.

Answer: CE

Explanation:

The AppDynamics .NET Agent can instrument both 32-bit and 64-bit processes, as long as they are running on a supported .NET Framework version and operating system. The agent automatically detects the process architecture and loads the appropriate profiler DLL. You can also specify the process architecture manually in the agent configuration file. The AppDynamics .NET Agent also supports instrumentation of multiple instances of the same application, such as Windows services or standalone applications. You can configure the agent to assign different tier and node names for each instance, based on the process name, process ID, or command line arguments. This allows you to monitor the performance and health of each instance separately.

QUESTION 10

What are three recommended steps to prepare a Linux environment for the installation of an AppDynamics Controller with a Large performance profile? (Choose three.)

- A. Install libaio,
- B. Install MySQL.
- C. Verify the user account has root access,
- D. Verify the open file descriptor limit.
- E. Verify that Java is installed.
- F. Verify the process limit.

Answer: ADF

Explanation:

To prepare a Linux environment for the installation of an AppDynamics Controller with a Large performance profile, which is suitable for monitoring up to 1000 agents, you need to perform the following steps:

Install libaio on the host machine if it does not already have it installed. This library facilitates asynchronous I/O operations on the system, which are required by the Controller. You can use the package manager of your Linux distribution to install libaio, such as yum or apt-get. For example, on CentOS, you can run yum install libaio.

Verify the open file descriptor limit on the system. The file descriptor limit determines how many files a process can open at a time. The Controller requires a high file descriptor limit to handle the large number of connections and transactions. AppDynamics recommends setting the file descriptor limit to at least 65535 for the user account that runs the Controller. You can check the current file descriptor limit by running ulimit -n and modify it by editing the /etc/security/limits.conf file. Verify the process limit on the system. The process limit determines how many processes a user can run at a time. The Controller requires a high process limit to handle the large number of threads and subprocesses. AppDynamics recommends setting the process limit to at least 65535 for the user account that runs the Controller. You can check the current process limit by running ulimit -u and modify it by editing the /etc/security/limits.conf file.

QUESTION 11

Which URL retrieves all AppDynamics business transactions from an application using the AppDynamics Rest API?

- A. http(s) ://<controller-host>:<port>/controller/rest/applications/<application_name>/allbts
- B. http(s) ://<controller-host>:<port>/controller/rest/applications/<application_name=>/business-transactions
- C. http(s) ://<controller-host>:<port>/controller/applications/<application_name>/business-transactions
- D. http(s) ://<controller-host>:<port>/controller/applications/<application_name=>/allbis

Answer: B

Explanation:

The AppDynamics Rest API to retrieve business transactions allows you to get a list of all business transactions in a business application, along with their key metrics and properties. The correct URL format for this API is:

http(s) ://<controller-host>:<port>/controller/rest/applications/<application_name>/business-transactions

QUESTION 12

Which two symptoms occur if an AppDynamics Controller is NOT scaled correctly? (Choose two.)

- A. Snapshots are NOT available after 2 weeks.
- B. Health rules violations occur more frequently.
- C. The average response times of tiers are higher than normal.
- D. The Controller's metric reporting is 7 to 10 minutes behind the current time.
- E. The Controller UI performs slowly.

Answer: DE

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the two symptoms that occur if an AppDynamics Controller is not scaled correctly are:

The Controller's metric reporting is 7 to 10 minutes behind the current time. (D) This is a valid symptom because the AppDynamics Controller collects, processes, and stores metrics from the agents that monitor the applications, tiers, nodes, and other entities. If the Controller is not scaled correctly, it may not have enough resources, such as CPU, memory, disk space, or network bandwidth, to handle the incoming metrics data. This may result in a backlog of metrics data that causes the Controller to lag behind the current time. The Controller's metric reporting delay can affect the

accuracy and timeliness of the performance analysis and troubleshooting.

The Controller UI performs slowly. (E) This is a valid symptom because the AppDynamics Controller UI is a web-based application that allows users to access, visualize, and interact with the performance data and configuration settings of the AppDynamics platform. If the Controller is not scaled correctly, it may not have enough resources, such as CPU, memory, disk space, or network bandwidth, to serve the UI requests. This may result in a slow or unresponsive UI that affects the user experience and productivity.

QUESTION 13

What is required in order for an upgrade of the Java Agent to succeed?

- A. The host machine must be restarted.
- B. The Machine Agent must be upgraded first,
- C. The AppDynamics Controller must be restarted.
- D. The application JVM must be restarted after the agent had been updated.

Answer: D

Explanation:

To upgrade the AppDynamics Java Agent, you need to copy the existing agent directory to a backup location and replace it with the new agent directory. You also need to copy any configuration file changes made in the old directory to the new agent directory. After that, you need to restart the application JVM where the Java Agent is installed, so that the new agent can take effect. Restarting the application JVM is the only required step for the upgrade to succeed. You do not need to restart the host machine, the Machine Agent, or the AppDynamics Controller.

QUESTION 14

A company set up an on-premises AppDynamics Controller and an on-premises Events Service cluster. What describes setting up this Events Service cluster?

- A. The Events Service API ports cannot be reconfigured from the default 9080/2081.
- B. The Controller must be able to initiate communication with the Events Service cluster.
- C. The Events Service cluster must be able to initiate communication with the Controller.
- D. There is no need for a load balancer or virtual IP address in front of the Events Service cluster.

Answer: B

Explanation:

The Events Service is the on-premises data storage facility for unstructured data generated by Application Analytics, Database Visibility, and End User Monitoring deployments. The Events Service can be deployed as a single node or a cluster of three or more nodes, depending on the data volume and availability requirements. The Controller and other Events Service clients connect to the Events Service to store and retrieve analytics data. Therefore, it is essential that the Controller is able to initiate communication with the Events Service cluster, using the Events Service API ports (default 9080/2081). The Events Service cluster does not need to initiate communication with the Controller, as it only responds to the requests from the clients. The Events Service API ports can be reconfigured from the default values, if needed, by modifying the conf/events-service-api-store.properties file on each node. However, the clients must also be updated with the new port values. It is also recommended to use a load balancer or a virtual IP address in front of the Events Service cluster, to provide a single endpoint for the clients and to enable load balancing and failover among the nodes.

QUESTION 15

Which REST query could be used to verify the availability of an AppDynamics Controller?

- A. `http://<controller_host>:<port>/controller/rest/uptime`
- B. `http://<controller_hosts>=<port>/controller/rest/serverstatus`
- C. `http://<controller_host>:<port>/controller/rest/ping`
- D. `http://<controller_host>:<port>/controller/rest/healthcheck`

Answer: C

Explanation:

The REST query that could be used to verify the availability of an AppDynamics Controller is1:
`http://<controller_host>:<port>/controller/rest/ping`

This query returns a simple text response of "pong" if the Controller is up and running, or an error message if the Controller is down or unreachable. This query does not require any authentication or parameters, and can be used as a quick and easy way to check the Controller status.

QUESTION 16

What is the minimum recommended number of nodes for a redundant Events Service?

- A. 1
- B. 2
- C. 3
- D. 4

Answer: C

Explanation:

The Events Service is a distributed database that stores and processes the analytics data collected by the AppDynamics platform. The Events Service cluster consists of multiple nodes that share the data load and provide fault tolerance and high availability. The minimum number of nodes for a functional Events Service cluster is one, but this is not recommended for production environments, as it does not provide any redundancy or resilience. The minimum number of nodes for a redundant Events Service cluster is three, as this allows the cluster to tolerate the failure of one node without losing any data or availability. The recommended number of nodes for a redundant Events Service cluster is five or more, as this provides better performance and scalability.

QUESTION 17

What are two capabilities of the standalone Machine Agent running on Linux? (Choose two.)

- A. It can act as a forwarder for analytics events.
- B. It can send SNMP alerts.
- C. It can communicate with multiple AppDynamics Controllers.
- D. It can restart itself if it goes down.
- E. It can start an HTTP listener for custom metrics.

Answer: AE

Explanation:

The AppDynamics standalone Machine Agent is a Java program that runs on a host machine and collects hardware and infrastructure metrics, such as CPU, memory, disk, and network usage. The Machine Agent can also perform additional functions, such as:

Acting as a forwarder for analytics events: The Machine Agent can be configured to forward business transaction, log, browser, mobile, and synthetic events from the application agents to the AppDynamics Events Service, which is a distributed, scalable data store for analytics data. The Machine Agent can also forward custom events from the SDK or API to the Events Service. This allows you to use the AppDynamics Analytics features, such as dashboards, queries, funnels, and metrics, to analyze the performance and behavior of your applications and users. Starting an HTTP listener for custom metrics: The Machine Agent can be configured to start an HTTP listener that can receive custom metrics from external sources, such as scripts, tools, or other applications. The Machine Agent can then report these custom metrics to the AppDynamics Controller, where you can view them in the Metric Browser or use them in health rules, policies, or dashboards. This allows you to monitor any aspect of your system that is not covered by the default Machine Agent metrics.

QUESTION 18

What are two ways in which large and extra large performance profiles differ from other profile types? (Choose two.)

- A. They must be installed on a bare metal server.
- B. They must be run with the High Availability Toolkit.
- C. They are not supported on Windows.
- D. They require an enterprise-grade database.
- E. An alert is generated when disk space falls below 2 GB.

Answer: DE

Explanation:

AppDynamics performance profiles are predefined sets of system requirements and configuration settings that are designed to support different levels of load and scalability for the AppDynamics platform. The performance profiles range from small to extra large, depending on the number of agents, metrics, and events that the platform needs to handle. The large and extra large performance profiles differ from other profile types in the following ways: They require an enterprise-grade database: The large and extra large performance profiles require an external MySQL database that is enterprise-grade, meaning that it has high availability, scalability, performance, and security features. The database should also have enough disk space, memory, and CPU resources to handle the expected load and growth. The AppDynamics platform uses the database to store configuration data, metric data, event data, and analytics data. An alert is generated when disk space falls below 2 GB: The large and extra large performance profiles have a built-in alert mechanism that notifies the administrator when the disk space on the Controller host falls below 2 GB. This is because the Controller needs enough disk space to store temporary files, logs, backups, and snapshots. If the disk space is insufficient, the Controller may experience performance degradation, data loss, or corruption. The administrator should monitor the disk space usage and free up space or add more disk capacity as needed.

QUESTION 19

Why would a load balancer be deployed in production for a single-node events cluster?

- A. to use the embedded Events Service along with the single-node cluster
- B. to hide the events server's real name
- C. to allow for deployment growth in the events cluster
- D. to provide redundancy for the single-node

Answer: C

Explanation:

A load balancer is a network device that distributes incoming traffic among a group of servers or nodes. A load balancer can improve the performance, availability, and scalability of a service by balancing the load and providing failover mechanisms. In the context of AppDynamics, a load balancer can be used to route the traffic from the Controller and other Events Service clients to the Events Service nodes. The Events Service is the on-premises data storage facility for unstructured data generated by Application Analytics, Database Visibility, and End User Monitoring deployments. One of the reasons why a load balancer would be deployed in production for a single-node Events Service cluster is to allow for deployment growth in the future. A single-node Events Service cluster is suitable for test environments or small-scale deployments, but it does not offer data replication or scalability. If the data volume or availability requirements increase, the Events Service cluster needs to be expanded to a multi-node cluster, which consists of three or more nodes. Deploying a load balancer in front of a single-node Events Service cluster makes it easier to add more nodes later, without having to modify the configuration of the Controller and other Events Service clients. The load balancer can also provide a single endpoint for the clients and enable load balancing and failover among the nodes.

QUESTION 20

What are three reasons you would create custom events using the Machine Agent REST API? (Choose three.)

- A. to create an event to track application deployment
- B. to create an event to be displayed in a Controller Audit report
- C. to create an alert that is to be triggered when a custom event is created
- D. to create an event to be displayed along with Time Series data in a custom dashboard
- E. to create an event to be used to trigger a health rule violation
- F. to create a new metric

Answer: ADE

Explanation:

The Machine Agent REST API allows you to create custom events that can be used for various purposes in AppDynamics. Some of the reasons you would create custom events using this API are:

To create an event to track application deployment. You can use the API to send a custom event that marks the start and end of an application deployment process. This can help you monitor the impact of the deployment on the application performance and availability, as well as correlate any issues or anomalies with the deployment event.

To create an event to be displayed along with Time Series data in a custom dashboard. You can use the API to send a custom event that contains any relevant information or context that you want to display in a custom dashboard. For example, you can send a custom event that contains the details of a configuration change, a maintenance window, a business transaction, or a user action. You can then use the custom dashboard to visualize the custom event data along with the Time Series data for the metrics you are interested in.

To create an event to be used to trigger a health rule violation. You can use the API to send a custom event that contains a metric value that you want to use as a condition for a health rule. For example, you can send a custom event that contains the CPU utilization of a machine, and then create a health rule that evaluates the CPU utilization metric and triggers a violation if it exceeds a certain threshold. You can then use the health rule violation to generate alerts, notifications, or remediation actions. Reference: Machine Agent HTTP Listener, Create Custom Events

QUESTION 21

The application server was restarted after an upgrade. What are two valid ways to confirm that the upgraded Java Agent is running successfully? (Choose two.)

- A. Verify that the application log contains a message indicating success.
- B. Verify that the node within the Controller UI indicates the app agent is reporting.
- C. Verify the Java Agent Version metric for that node in the Metric Browser.
- D. Verify that the Java Agent log contains a message indicating the agent started successfully.

Answer: BC

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the two valid ways to confirm that the upgraded Java Agent is running successfully are:

Verify that the node within the Controller UI indicates the app agent is reporting.

(B) This is a valid way because the Controller UI is a web-based application that allows users to monitor and manage the performance of the applications, tiers, nodes, and other entities that are instrumented by the AppDynamics agents. The Controller UI displays the status of the nodes within each tier, and indicates whether the app agent is reporting or not. If the app agent is reporting, the node icon is green and shows the agent version. If the app agent is not reporting, the node icon is gray and shows the last time the agent reported. The user can also hover over the node icon to see more details, such as the agent type, the agent version, the agent runtime directory, and the agent properties file. The user can verify that the upgraded Java Agent is running successfully by checking that the node icon is green and shows the latest agent version. Verify the Java Agent Version metric for that node in the Metric Browser.

(C) This is a valid way because the Metric Browser is a feature of the Controller UI that allows users to view and analyze the metrics collected by the AppDynamics agents. The Metric Browser displays the metrics in a hierarchical tree structure, where each node represents a metric category, a metric name, or a metric value. The user can expand or collapse the nodes, and select or deselect the metrics to view them in a chart. The user can also apply filters, time ranges, baselines, and other options to customize the chart. The user can verify that the upgraded Java Agent is running successfully by navigating to the Java Agent Version metric for that node in the Metric Browser. The Java Agent Version metric shows the version number of the Java Agent that is running on the node. The user can compare the metric value with the expected agent version, and check that the metric is updated after the upgrade.

QUESTION 22

The Database Agent collects hardware metrics from a Windows database server using_____. (Choose the correct option to complete the sentence.)

- A. Standalone Machine Agent
- B. PowerShell
- C. WHI
- D. SSH

Answer: B

Explanation:

The Database Agent collects hardware metrics from a Windows database server using PowerShell. PowerShell is a scripting language and a command-line shell that allows the Database Agent to execute commands and access Windows Performance Counters on the target host. The Database Agent uses PowerShell to collect metrics such as CPU, memory, disk, and network utilization from the Windows database server. To enable hardware monitoring for a Windows database server, the Database Agent requires the following permissions:

The user that runs the Database Agent must have permission to execute PowerShell scripts on the local machine. The user that runs the Database Agent or the Collector Service user (if using Windows Authentication) must have permission to establish a WMI connection to the target host and collect Windows Performance Counters.

QUESTION 23

What are the correct steps to install a .NET Agent patch?

- A. Uninstall the existing .NET Agent
Install the patch
Restart the instrumented application(s)
- B. Restart the machine
Install the patch over exiting .NET agent
Restart instrumented applications(s)
- C. Install the .NET Agent patch
Restart the instrumented application(s)
- D. Restart the instrumented application(s)
Apply the patch over existing NET agent

Answer: C

Explanation:

To install a .NET Agent patch, which is a minor update to the existing .NET Agent version, you do not need to uninstall the old agent or restart the machine. You only need to follow these steps:

Download the .NET Agent patch from the AppDynamics Download Center. Launch an elevated command prompt with full administrator privileges. Execute the Installer.bat file from the patch archive. The batch file installs the patch and starts the AppDynamics Agent Coordinator service.

Restart the instrumented applications, such as IIS, Windows services, or standalone applications.

QUESTION 24

Creating alerting extensions via APIs allows for:

- A. Development of custom notification channels
- B. Integration with external incident management tools
- C. Customization of alert thresholds based on dynamic conditions
- D. Automatic adjustment of the platform's UI theme

Answer: ABC

QUESTION 25

Which API endpoint is typically used to retrieve a list of applications monitored by the platform?

- A. /api/metrics/apps
- B. /api/v1/apps/
- C. /api/apps/list
- D. /ui/apps/dashboard

Answer: B

QUESTION 26

Essential considerations for high availability setup include:

- A. Load balancing methods
- B. The theme of the monitoring dashboard
- C. Network redundancy and failover mechanisms
- D. Database replication and synchronization

Answer: ACD

QUESTION 27

What is the most important factor in determining sizing for AppDynamics Controller?

- A. Projected metric load per minute on the Controller
- B. Number of administrators/end users logging on to the Controller to monitor application performance

- C. Type of agents reporting to the Controller
- D. Number of AppDynamics applications to be created on the Controller

Answer: A

Explanation:

The most important factor in determining sizing for AppDynamics Controller is the projected metric load per minute on the Controller. This is because the metric load represents the actual workload on the Controller, which depends on the nature of the application, the AppDynamics configuration, and the usage patterns. The number of agents, the type of agents, the number of administrators/end users, and the number of AppDynamics applications are only rough estimates that can vary greatly depending on the specific scenario. Therefore, it is recommended to test the performance of the system in a staging environment and verify the Controller sizing using the metric upload rate before deploying to production.

QUESTION 28

Which two choices are available when specifying an application in a URL string for the Health Rule REST API? (Choose two.)

- A. Application Alias
- B. Application ID
- C. Application GUID
- D. Application Name
- E. Application REGEX

Answer: BD

Explanation:

The Health Rule REST API allows you to create, configure, update, and delete health rules for multiple applications simultaneously. To use this API, you need to specify the application in the URL string. You can use either the application ID or the application name for this purpose. The application ID is a unique numeric identifier for each application in the Controller. The application name is the display name of the application in the AppDynamics UI. You cannot use the application alias, GUID, or REGEX for the Health Rule REST API.

QUESTION 29

What are three requirements to set up AppDynamics Controllers as a high availability pair? (Choose three.)

- A. Passwordless SSH must be configured between the two Controller servers.
- B. The Controller MySQL database must be installed on a shared location.
- C. The replicate sh script can be run only once.
- D. Both servers must have the Controller software installed prior to setting up high availability.
- E. A unique high availability license file is required for each Controller server.
- F. Both servers must have identical directory structures for the Controller installation.

Answer: ADF

Explanation:

To set up AppDynamics Controllers as a high availability pair, you need to meet the following requirements: Passwordless SSH must be configured between the two Controller servers. This allows the Enterprise Console to automate the configuration and administration tasks associated with a highly available deployment on Linux systems. Both servers must have the Controller software installed prior to setting up high availability. The Controllers in an HA pair must be equivalent versions, and be in the same data center. Both servers must have identical directory structures for the Controller installation. The individual machines in the Controller HA pair need to have an equivalent amount of disk space.

QUESTION 30

Which two statements are true when updating the Database Agent? (Choose two.)

- A. The Database Agent must be stopped and restarted during the upgrade.
- B. If the agent is moved to a new location during the upgrade, the AppDynamics Controller must be reconfigured to reference the new location of the agent.
- C. All data collectors created from the previous agent must be migrated to the new agent.

- D. Controller-info.xml is the only file that needs to be migrated from the previous agent to the new agent.
- E. After the Database Agent is upgraded, the AppDynamics Controller must be restarted.

Answer: AD

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, when updating the Database Agent, you need to follow these steps:

Stop the agent as described for your specific installation in Start and Stop the Database Agent. Make a copy of the existing agent directory, <db_agent_home>.

Backing up allows you to revert to the previous agent installation if you need to. You can also copy over the controller-info.xml configuration file to the new installation to ensure the agent configuration is maintained. Install the Database Agent as described for your specific installation in Administer the Database Agent.

Copy the <backup_db_agent_home>\conf\controller-info.xml file to the new installation directory, <db_agent_home>\conf.

To ensure the agent configuration is maintained, copy the <backup_db_agent_home>\conf\controller-info.xml file to the new installation directory, <db_agent_home>\conf.

Start the new agent.

See Start and Stop the Database Agent.

Verify the Database Agent Installation.

See Verify the Database Agent Installation.

QUESTION 31

What are two settings that an administrator can configure from the AppDynamics Controller Admin Console? (Choose two.)

- A. Metrics baseline calculation
- B. Controller log file rotation
- C. Controller heap size
- D. License allocation
- E. Retention periods

Answer: DE

Explanation:

The AppDynamics Controller Admin Console lets you configure certain global settings for the Controller, such as metric retention periods, UI notification triggers, tenancy mode, and accounts in multi-tenancy mode. Two of the settings that an administrator can configure from the AppDynamics Controller Admin Console are:

License allocation: This setting allows you to view and manage the license usage and availability for your Controller. You can see the total number of licenses, the number of licenses in use, the number of licenses available, and the license expiration date for each agent type. You can also allocate licenses to specific applications or accounts, and set license limits and alerts.

Retention periods: This setting allows you to specify how long the Controller retains the metric data and the event data for your monitored applications. You can configure the retention periods for different types of data, such as minute-level metrics, hour-level metrics, day-level metrics, transaction snapshots, and events. You can also configure the data purge schedule and the data backup schedule.