

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

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. Reference: [Controller System Requirements, Performance and Controller sizing guidelines](#), [How to Run AppDynamics in Microsoft Azure](#), [Platform Requirements](#)

Question: 2

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. Reference: [Health Rule API](#) and [Retrieve All Business Applications](#) in the AppDynamics documentation.

Question: 3

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 requirements1:](#)

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. Reference: [Prerequisites for High Availability](#)

Question: 4

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 steps12:](#)

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.

Therefore, the correct statements are:

The Database Agent must be stopped and restarted during the upgrade. (A)

Controller-info.xml is the only file that needs to be migrated from the previous agent to the new agent. (D)

The incorrect statements are:

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. (B) This is not true because the controller- info.xml file contains the information about the Controller host, port, account name, access key, and SSL settings. As long as this file is copied to the new agent location, the Controller does not need to be reconfigured.

All data collectors created from the previous agent must be migrated to the new agent. © This is not true because the data collectors are configured on the Controller UI, not on the agent. The agent collects the metrics from the databases and sends them to the Controller. The data collectors do not need to be migrated to the new agent.

After the Database Agent is upgraded, the AppDynamics Controller must be restarted. (E) This is not true because the Controller does not depend on the agent version. The agent and the Controller are compatible as long as they meet the Agent and Controller Compatibility requirements.

Reference:

- 1: Upgrade the Database Agent - AppDynamics
- 2: Release Upgrade Checklist for Database Agents - AppDynamics

Question: 5

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 mode¹. 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 alerts².

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 schedule³.

Reference: [Access the Administration Console](#), [License Rules](#), [Data Retention](#)

Question: 6

If using SSL for agent communication with an AppDynamics Controller, it is recommended that agent SSL traffic

- A. is encrypted with a 64-bit encryption key
- B. is decrypted and then encrypted again at the firewall
- C. terminates at a reverse proxy or a load balancer in front of the Controller

D. terminates at the Controller

Answer: C

Explanation:

AppDynamics recommends that you terminate SSL connections from agents at a reverse proxy or a load balancer in front of the Controller, rather than at the Controller itself. This improves the performance and scalability of the Controller by offloading the SSL processing to the proxy or load balancer. It also simplifies the configuration and management of SSL certificates and truststores. To enable this option, you need to configure the proxy or load balancer to accept SSL connections from agents and forward them to the Controller using HTTP. You also need to configure the agents to use SSL and point to the proxy or load balancer host and port. See [Enable SSL for the Java Agent](#) for an example. Reference: [Agent-to-Controller Connections](#) and [Enable SSL and SSH for Database Agent Communications](#) in the AppDynamics documentation.

Question: 7

Which artifacts are needed, at minimum, to add a custom extension to a Machine Agent?

A. Directory

Jar or Script file

Monitor.xml file

B. Directory Monitor.xml file Custom-gxtension.xml file C. Directory

Zip Extension Monitor.xml file D. Directory Script file Config.json file

Answer: A

Explanation:

[To add a custom extension to a Machine Agent, you need at least three artifacts: a directory, a jar or script file, and a monitor.xml file¹²](#). The directory is where you place your extension files under the <machine_agent_home>/monitors directory. The jar or script file is the executable file that collects and reports the custom metrics to the Machine Agent. The monitor.xml file is the configuration file that defines the name, frequency, and parameters of your extension. [You can also optionally include other files such as libraries, configuration files, or templates for your extension¹²](#). Reference: [Build a Monitoring Extension Using Scripts, Extensions and Custom Metrics](#)

Question: 8

An administrator is asked to improve the capacity of an Events Service cluster. What is the recommended way to add capacity to the cluster?

A. Add a new Events Service cluster to share the load.

B. Add nodes running on machines with identical hardware matching the existing nodes.

C. Add more storage to the master nodes of the cluster.

D. Add more storage to as many of the existing nodes as possible.

Answer: B

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the recommended way to add capacity to the Events Service cluster is to add nodes running on machines with identical hardware matching the existing nodes¹². This will increase the data storage, replication, and redundancy of the cluster, as well as the processing power for queries. The Events Service cluster is horizontally scalable, so nodes can be added as your data storage requirements grow³². The Events Service must run on dedicated machines with identical directory structures, user account profiles, and hardware profiles¹. For heap space allocation, AppDynamics recommends allocating half of the available RAM to the Events Service process, with a minimum of 7 GB up to 31 GB¹. Solid-state drives (SSD) can significantly outperform hard disk drives (HDD), and are therefore recommended for production deployments¹.

The incorrect options are:

Add a new Events Service cluster to share the load. (A) This is not recommended because it will

create additional complexity and overhead for managing multiple clusters and routing data and queries to the appropriate cluster. It will also require more license units to enable analytics on multiple clusters.

Add more storage to the master nodes of the cluster. (C) This is not recommended because it will not increase the data replication and redundancy of the cluster, nor the processing power for queries. It will also create an imbalance in the cluster, as the master nodes will have more storage than the worker nodes, which may affect the performance and stability of the cluster.

Add more storage to as many of the existing nodes as possible. (D) This is not recommended because it will not increase the data replication and redundancy of the cluster, nor the processing power for queries. It will also create an imbalance in the cluster, as some nodes will have more storage than others, which may affect the performance and stability of the cluster.

Reference:

¹: Cisco AppDynamics Professional Implementer (500-430)

²: Events Service Requirements - AppDynamics

³: Events Service Deployment - AppDynamics

⁴: Events Service Deployment - AppDynamics

Question: 9

Which type of Alert and Respond action is available only on an on-premises Controller?

- A. Remediation action
- B. Cloud auto-scaling
- C. Diagnostic action
- D. Custom action

Answer: D

Explanation:

A custom action is a type of Alert and Respond action that allows you to integrate AppDynamics notifications with your own alerting or ticketing system. You can create a custom action by writing an executable script and an XML file that describe how to pass information from AppDynamics to your system. A custom action can be triggered by a policy based on a health rule violation or an event. A custom action is available only on an on-premises Controller, because it requires access to the Controller file system and the ability to execute scripts on the Controller host machine. For a SaaS Controller, you can use HTTP request templates instead of custom actions to integrate with external systems. Reference: Build a Custom Action, Alert and Respond, Actions, Custom Actions

Question: 10

Which implementation mode should be used when deploying an EUM Server with reverse proxy from a security and performance standpoint in a production environment?

- A. Install the EUM Server on the same host as the Controller, behind the reverse proxy.
- B. Install the EUM Server on a separate host from the Controller, in front of the reverse proxy.
- C. Install the EUM Server on the same host as the Controller, in front of the reverse proxy.
- D. Install the EUM Server on a separate host from the Controller, behind the reverse proxy.

Answer: D

Explanation:

The EUM Server is the on-premises version of the EUM Cloud that receives, processes, and stores data from EUM agents. For production installation, AppDynamics recommends that the EUM Server and the Controller sit on different hosts, and the EUM Server hosts its own MySQL instance. This improves the scalability and performance of both components. Moreover, AppDynamics strongly recommends that SSL traffic from agents to the EUM Server is terminated at a reverse proxy that sits in front of the EUM Server in the network path, and forwards connections to the EUM Server using HTTP. This relieves the SSL processing load from the EUM Server, simplifies the certificate management, and enhances the security by isolating the EUM Server from the external Internet. Therefore, the best implementation mode for deploying an EUM Server with reverse proxy in a production environment is option D. Reference: [EUM Server Deployment](#), [Install a Production EUM Server](#), [Reverse Proxy configuration for EUM](#), and [Browser EUM installation](#) in the AppDynamics documentation and community.

Question: 11

Default configuration of the Mobile SDK enables and . (Choose the two correct options to complete the sentence.)

- A. Crash Reports
- B. User Data
- C. Network Requests
- D. Custom Timers
- E. Breadcrumbs

Answer: AC

Explanation:

[The default configuration of the Mobile SDK enables crash reports and network requests](#)¹². [Crash reports capture and report any unhandled exceptions or signals that cause the app to terminate abnormally](#)¹. [Network requests monitor the performance and errors of HTTP and HTTPS requests made by the app](#)². [These features are enabled by default and do not require any additional code or configuration to work](#)¹². Reference: [Crash Reports](#), [Network Requests](#)

Question: 12

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 application¹.](#) 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 action¹.](#) 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 NOW¹.](#) 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.

The incorrect options are:

[For updating an AppDynamics dashboard \(A\): This is not a valid reason for using the REST API for health rule violations, because the AppDynamics dashboards already display the health rule violations that occurred in the selected time frame, along with the severity, status, affected entities, and actions².](#) You do not need to use the REST API to update the dashboard, as the dashboard is automatically refreshed with the latest data from the Controller.

For sending emails (D): This is not a valid reason for using the REST API for health rule violations, because the REST API does not send emails directly. The REST API only returns the information about the health rule violations, and the actions that were triggered by them. [If you want to send emails based on the health rule violations, you need to configure an email action in the health rule configuration, or use a custom action that invokes an external email service³.](#)

When pushing events to the Event Management System is NOT possible (E): This is not a valid reason for using the REST API for health rule violations, because the REST API does not push events to the Event Management System. The REST API only returns the information about the health rule violations, and the actions that were triggered by them. [If you want to push events to the Event Management System, you need to configure an HTTP request action in the health rule configuration, or use a custom action that invokes an external API³.](#)

Reference:

- [1: Health Rule Violations API - AppDynamics](#)
- [2: Health Rule Violations - AppDynamics](#)
- [3: Actions - AppDynamics](#)

Question: 13

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 81811. 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 Controller2. The GlassFish administration port is used to access the Controller Admin Console, not by the EUM Server3. The Controller database port is used by the Controller to connect to the MySQL database, not by the EUM Server4.](#) Reference: [Controller Port Settings](#), [Configure the Port for the EUM Agent](#), [Access the Administration Console](#), [Controller System Requirements](#)

Question: 14

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 libaio1.](#)

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 /tmp2. 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 installer3.](#)

[Other preparatory tasks include verifying the user account permissions, configuring the virus scanners, installing the netstat network utility, and setting the file descriptor limit2.](#) Reference: [Prepare Linux for the Controller](#), [Install the Controller on Linux](#), and [Controller System Requirements] in the AppDynamics documentation.

Question: 15

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 itself¹. To access the information provided by the internal agent, you need to log in to the Controller UI with the following credentials²:

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 it². The root user is the default administrator user for the system account and has the same password as the admin user for the customer¹ account³. Reference: [Controller Self-Monitoring](#), [Monitoring a Controller Using the Internal Monitoring Agent](#), [Controller Accounts](#)

Question: 16

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 Package¹². 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.

The incorrect options are:

Perform the agent upgrade on a remote server host by using the AppDynamics Controller REST API.

(8) This is not a valid method for upgrading the NET Agent, because the AppDynamics Controller REST API does not provide any endpoint for agent installation or upgrade. The REST API is mainly used for retrieving or updating configuration data, metrics, events, snapshots, and other information from the Controller³.

Perform the agent upgrade on the application server host by running the Agent Configuration Utility. © This is not a valid method for upgrading the NET Agent, because the Agent Configuration Utility is a tool for modifying the agent configuration after installation, not for installing or upgrading the agent. The Agent Configuration Utility allows you to change the Controller connection settings, the agent logging level, the proxy settings, and other advanced options⁴.

Perform the agent upgrade via the AppDynamics Controller UI. (D) This is not a valid method for upgrading the

NET Agent, because the AppDynamics Controller UI does not provide any feature for agent installation or upgrade. [The Controller UI is mainly used for monitoring, analyzing, and troubleshooting the performance of the applications, business transactions, tiers, nodes, and other entities that are instrumented by the agents5.](#)

Reference:

- [1:](#) Upgrade the .NET Agent for Windows - AppDynamics
- [2:](#) Release Upgrade Checklist for .NET Agents - AppDynamics
- [3:](#) REST API - AppDynamics
- [4:](#) Configure the .NET Agent - AppDynamics
- [5:](#) AppDynamics Application Performance Monitoring Platform - AppDynamics

Question: 17

Which directory should an administrator back up if the goal is to back up the EUM Server?

- A. <eum_server_home> directory
- B. <controller_home>/bin directory
- C. <controller_home>
- D. <controller_home>/bin/eum_server directory

Answer: A

Explanation:

The <eum_server_home> directory contains the EUM Server installation files, configuration files, and data files. It is recommended to back up this directory as a precaution before upgrading or migrating the EUM Server. The default location of this directory is <installDir>/AppDynamics/EUM, where <installDir> is the directory where you installed the Controller. [You can also use the backup-eum.sh script to back up the EUM Server data12.](#) Reference: [Upgrade the Production EUM Server](#), [Configure the EUM Server](#)

Question: 18

Which framework would require the implementation of custom correlation?

- A. Custom TCP concurrent server
- B. Customer proprietary SOAP application
- C. Vendor-supplied enterprise application that uses JMS
- D. Pre-packaged WCF application

Answer: A

Explanation:

Custom correlation is needed when the default detection mechanisms of AppDynamics are not capable of auto-correlating transactions across tiers or across parent-child threads in complex multithreaded applications. Custom correlation enables the user to configure AppDynamics to propagate a unique correlation key by using the extension points of the distributed protocol or by decorating the payload. Among the four options, a custom TCP concurrent server is the most likely to require the implementation of custom correlation, as it is an unsupported framework and protocol that may not have easily-defined method calls or payload objects to configure as exit points or entry points. The other options, such as SOAP, JMS, and WCF, are supported by AppDynamics and can be automatically correlated by the agents without the need for custom configuration. Reference: [Custom Correlation for Java Applications](#) and [Configure Custom Correlation for .NET Applications](#) in the AppDynamics

community.

Question: 19

Which two statements are true regarding the AppDynamics REST API for retrieving metrics? (Choose two.)

- A. Metrics can be retrieved for a fixed time range.
- B. Median is one of the returned values,
- C. End-time value must be provided if using the time-range-type of AFTER_TIME.
- D. Minimum and maximum values are meaningful for all metric types.
- E. Wildcards can be used in the REST API metric path.

Answer: AE

Explanation:

The AppDynamics REST API for retrieving metrics allows you to get values generated for metrics by specifying the path of the metric and the time frame for the data¹. The following statements are true regarding this API¹²:

Metrics can be retrieved for a fixed time range. You can use the time-range-type parameter to specify a fixed time range such as BEFORE_NOW, AFTER_TIME, or BETWEEN_TIMES. You can also use the duration-in-mins parameter to specify the length of the time range in minutes.

Wildcards can be used in the REST API metric path. You can use the asterisk (*) character as a wildcard to match any metric name or part of a metric name. For example, you can use the metric path Business Transaction Performance|Business Transactions|*|Average Response Time (ms) to retrieve the average response time for all business transactions in all tiers. Reference: [Retrieve Metric Data](#), [Retrieve Metric Hierarchy](#)

Question: 20

Which two methods are available to define JVM options for an AppDynamics Controller so that the JVM options are retained across upgrades of the Controller? (Choose two.)

- A. Use the modifyJvmOptions utility provided by AppDynamics.
- B. Define JVM options on the Controller Settings page of the Enterprise Console.
- C. Pass JVM options to the Controller via java -javaagent:"options jar".
- D. Use the controller.sh script provided by AppDynamics.
- E. Define JVM options manually in the domain.xml file.

Answer: AB

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the two methods that are available to define JVM options for an AppDynamics Controller so that the JVM options are retained across upgrades of the Controller are:

Use the modifyJvmOptions utility provided by AppDynamics. (A) This is a valid method because the modifyJvmOptions utility is a script that allows you to add, remove, or list the JVM options for the Controller without manually editing any files. The utility also validates the syntax and format of the JVM options and creates a backup of the original configuration. The utility is located in the <controller_home>/bin directory and can be run on Linux or Windows platforms. The utility updates the <controller_home>/appserver/glassfish/domains/domain1/config/domain.xml file with the specified JVM options, which are preserved during the Controller upgrade¹².

Define JVM options on the Controller Settings page of the Enterprise Console. (B) This is a valid method because the Controller Settings page of the Enterprise Console is a graphical user interface that allows you to configure various settings for the Controller, including the JVM options. The Enterprise Console is a web-based application that provides a centralized way to manage the AppDynamics platform components, such as the Controller, the Events Service, and the EUM Server. [The Enterprise Console also handles the Controller upgrade process and preserves the JVM options that are defined on the Controller Settings page34.](#)

The incorrect options are:

Pass JVM options to the Controller via java -javaagent:"options jar". © This is not a valid method because the java -javaagent option is used to specify the path to the AppDynamics agent jar file, not the JVM options for the Controller. The agent jar file is used to instrument the Java applications that are monitored by the AppDynamics platform, not the Controller itself. The agent jar file also contains the agent configuration properties, such as the Controller host, port, account name, access key, and application name. [Passing JVM options to the Controller via this option will not have any effect on the Controller performance or behavior5.](#)

Use the controller.sh script provided by AppDynamics. (D) This is not a valid method because the controller.sh script is used to start, stop, or restart the Controller, not to define the JVM options for the Controller. The controller.sh script is located in the <controller_home>/bin directory and can be run on Linux platforms. The controller.sh script does not accept any arguments or parameters for the JVM options, and does not update any configuration files with the JVM options. [Using this script to define the JVM options for the Controller will not have any effect on the Controller performance or behavior6.](#)

Define JVM options manually in the domain.xmi file. (E) This is not a valid method because the domain.xmi file is not a configuration file for the JVM options for the Controller, but a configuration file for the WebSphere Application Server. The WebSphere Application Server is a Java EE application server that can be used to host the Java applications that are monitored by the AppDynamics platform, not the Controller itself. The domain.xmi file contains the settings for the WebSphere Application Server, such as the server name, port, security, data sources, and class loaders. [Defining JVM options manually in this file will not have any effect on the Controller performance or behavior, and may cause errors or conflicts with the WebSphere Application Server configuration7.](#)

Reference:

- [1:](#) Modify JVM Options for the Controller - AppDynamics
- [2:](#) Release Upgrade Checklist for Controllers - AppDynamics
- [3:](#) Configure the Controller Using the Enterprise Console - AppDynamics
- [4:](#) Upgrade the Controller Using the Enterprise Console - AppDynamics
- [5:](#) Install the Java Agent - AppDynamics
- [6:](#) Start and Stop the Controller - AppDynamics
- [7:](#) Configuring the WebSphere Application Server - IBM

Question: 21

What becomes more important as an AppDynamics Controller grows beyond supporting 500 agents?

- A. CPU utilization
- B. RAM allocated to the Controller
- C. Network throughput
- D. Disk I/O
- E. Thread count on the GlassFish server

Answer: C

Explanation:

As an AppDynamics Controller grows beyond supporting 500 agents, network throughput becomes more

important. This is because the Controller needs to handle a large volume of data from the agents, as well as serve requests from the UI and API clients. Network throughput is the measure of how much data can be transferred over a network in a given time. A low network throughput can cause delays, errors, or timeouts in the communication between the Controller and the agents or clients. [Therefore, it is recommended to monitor the network throughput of the Controller and ensure that it meets the minimum requirements for the expected load](#)¹²³. Reference: [Controller System Requirements](#), [Performance and Controller Sizing Guidelines](#), [How to Run AppDynamics in Microsoft Azure](#)

Question: 22

Which two user accounts are created by the AppDynamics Controller during installation? (Choose two.)

- A. Elastic search root user
- B. GlassFish asadmin user
- C. Customer-specified Controller administrator account
- D. OS user that will run the controller
- E. REST API user
- F. MySQL appd admin user

Answer: BC

Explanation:

The AppDynamics Controller is a Java web application that runs on a GlassFish application server and uses a MySQL database. During the installation of the Controller, two user accounts are created by default:

The GlassFish asadmin user is the administrative user for the GlassFish server. This user has the authority to start, stop, and configure the GlassFish server and its domains. The default username for this user is admin and the default password is appdynamics. [You can change the password for this user after the installation by using the asadmin command-line tool](#)¹.

The customer-specified Controller administrator account is the user account that you provide during the installation wizard. This is the account that you use to access the AppDynamics User Interface (UI) for the first time and perform various tasks such as creating applications, configuring agents, managing users and groups, and so on. You can choose any username and password for this account, but AppDynamics recommends using only ASCII characters. [You can also create additional user accounts in the Controller UI after the installation](#)².

The other options are not user accounts that are created by the Controller installation. The Elastic search root user, the REST API user, and the MySQL appd admin user are user accounts that are used internally by the Controller components and are not exposed to the end user. [The OS user that will run the controller is a user account that you need to create on the host machine before the installation, and it is not created by the Controller installation](#)³. Reference: [Controller Installation](#), [Manage Users and Groups](#), and [Update the Root User and Glassfish Admin Passwords](#) in the AppDynamics documentation.

Question: 23

What are two advantages of using an Events Service cluster? (Choose two.)

- A. Clusters allow data replication across multiple nodes.
- B. Clusters are easier to maintain than single-node instances.
- C. Clusters are horizontally scalable by adding nodes.
- D. Clusters reduce the load on the AppDynamics Controller.
- E. Clusters expose multiple channels for simultaneous queries.

Answer: AC

Explanation:

An Events Service cluster is a group of two or more Events Service nodes that work together to store and process unstructured data generated by AppDynamics components such as Application Analytics, Database Visibility, and End User Monitoring¹. Using an Events Service cluster has two main advantages over a single-node instance¹²:

Clusters allow data replication across multiple nodes. This means that the data is duplicated and distributed among the nodes in the cluster, providing data redundancy and protection against data loss in case of a node failure. Data replication also improves data availability and query performance, as the cluster can handle concurrent requests from multiple clients.

Clusters are horizontally scalable by adding nodes. This means that the cluster can grow in size and capacity by adding more nodes to the cluster, without affecting the existing nodes or data. Horizontal scaling allows the cluster to handle increasing data volumes and performance demands, as well as balance the workload among the nodes.

Reference: [Events Service Deployment](#), [What are the Benefits of Server Clustering?](#)

Question: 24

What are two reasons that would require an administrator to install the Events Service cluster manually? (Choose two.)

- A. Installation on SUSE Linux
- B. Security concerns with passwordless SSH
- C. Security requirements to install using a non-root user account
- D. Installation on Windows

Answer: BC

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the two reasons that would require an administrator to install the Events Service cluster manually are:

Security concerns with passwordless SSH (B): This is a valid reason because the automated installation of the Events Service cluster requires passwordless SSH access to the target hosts. Passwordless SSH allows the Enterprise Console to execute commands on the remote hosts without prompting for a password. However, some organizations may have security policies that prohibit passwordless SSH access, as it may pose a risk of unauthorized access or malicious attacks. [In such cases, the administrator can install the Events Service cluster manually, by following the steps described in the Manual Installation of the Events Service Cluster document¹.](#)

Security requirements to install using a non-root user account ©: This is a valid reason because the automated installation of the Events Service cluster requires root privileges on the target hosts. Root privileges allow the Enterprise Console to create directories, change permissions, and install packages on the remote hosts. However, some organizations may have security policies that restrict root access, as it may pose a risk of accidental or intentional damage to the system. [In such cases, the administrator can install the Events Service cluster manually, by following the steps described in the Manual Installation of the Events Service Cluster document¹.](#) The administrator can use a non-root user account that has sudo privileges to perform the manual installation.

The incorrect options are:

Installation on SUSE Linux (A): This is not a valid reason for manual installation, because the automated installation of the Events Service cluster supports SUSE Linux as one of the compatible operating systems. [The Enterprise Console can install the Events Service cluster on SUSE Linux hosts using the automated installation process, as long as the hosts meet the prerequisites described in the Events Service Requirements document².](#)

Installation on Windows (D): This is not a valid reason for manual installation, because the Events Service cluster

does not support Windows as an operating system. The Events Service cluster can only run on Linux hosts, as it is based on Apache Cassandra, which is a Linux-based distributed database. [The Events Service cluster cannot be installed on Windows hosts, either manually or automatically2.](#)

Reference:

- [1:](#) Manual Installation of the Events Service Cluster - AppDynamics
- [2:](#) Events Service Requirements - AppDynamics

Question: 25

Which URL retrieves all AppDynamics business applications from an AppDynamics Controller using the AppDynamics Rest API?

- A. `http(s)://<controller-host>:<port>/controller/rest/businessapplications`
- B. `http(s)://<controller-host>:<port>/controller/allapplications`
- C. `http(s)://<controller-host>:<port>/controller/rest/applications`
- D. `http(s)://<controller-host>:<port>/controller/applications`

Answer: C

Explanation:

The AppDynamics Rest API allows you to retrieve information and perform operations on the AppDynamics platform using HTTP requests. To retrieve all AppDynamics business applications from an AppDynamics Controller using the AppDynamics Rest API, you need to use the following URL format: `http(s)://<controller-host>:<port>/controller/rest/applications`

This URL returns the business application names and internal numeric identifiers for all the applications that are monitored by the Controller. You can use the application name or ID as a parameter for other API methods that require the application context. You can also specify the output format as XML (default) or JSON by adding the output query parameter. For example: `http(s)://<controller-host>:<port>/controller/rest/applications?output=JSON`

This URL returns the same information as the previous one, but in JSON format. You can also filter the applications by their status (alive or not) by adding the time-range-type query parameter. For example:

`http(s)://<controller-host>:<port>/controller/rest/applications?time-range-type=BEFORE_NOW&duration-in-mins=60`

This URL returns only the applications that are alive in the last 60 minutes. [An alive application is an application with at least one node that submits at least one metric to the Controller in the provided time range12.](#)

Reference: [Application Model API](#), [Metric and Snapshot API](#)

Question: 26

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 keywords1.](#)

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 keywords2.](#)

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. Therefore, the best answer is option C and E. Reference: [Sensitive Data Collection and Security](#), [SQL Bind Variables](#), and [Diagnostic Data Collectors](#) in the AppDynamics documentation.

Question: 27

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 data1. Some of the valid reasons to use this API are12:](#)

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. Reference: [Retrieve Metric Data](#), [Health Rule API](#)

Question: 28

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](#)

The incorrect options are:

Should be installed using the same user account (A): This is not a valid recommendation because the Events Service Cluster does not require the same user account for installation. The user account that is used to install the Events Service Cluster should have sudo privileges on the target hosts, but it does not have to be the same for all the hosts. [The user account that is used to run the Events Service Cluster should have read and write permissions on the installation directory, but it does not have to be the same for all the hosts either](#)

Should have eight or more CPU cores (B): This is not a valid recommendation because the Events Service Cluster does not require eight or more CPU cores for the servers. The minimum requirement for the CPU cores is four, and the recommended requirement is six. [Having more CPU cores may improve the performance of the cluster, but it is not a mandatory recommendation](#)

Should be running the same operating system version (E): This is not a valid recommendation because the Events Service Cluster does not require the same operating system version for the servers. The Events Service Cluster supports various Linux operating systems, such as CentOS, Red Hat, Ubuntu, and SUSE. [The operating system version should be compatible with the Events Service version, but it does not have to be the same for all the hosts](#)

Reference:

[1: Events Service Requirements - AppDynamics](#)

[2: Events Service Deployment - AppDynamics](#)

Question: 29

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 file1](#). 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 separately2](#).

The other statements are false because:

A) AppDynamics .NET Agent does support instrumenting processes running multiple App domains. The agent can monitor multiple App domains within a single process, as long as they are running the same .NET Framework version. [The agent can also monitor multiple applications within a single App domain, by using the Standalone Applications element in the agent configuration file3](#).

B) AppDynamics .NET Agent does not automatically discover all the Windows services to be instrumented. The agent automatically instruments IIS applications only. For Windows services or standalone applications, you need to manually configure the agent by editing the agent configuration file and adding the Windows Services or Standalone Applications element. [You also need to provide the executable name, tier name, and node name for each service or application4](#).

D) AppDynamics .NET Agent does not require that the Windows services are running under the "App.pool identity user" account. The agent can instrument Windows services running under any user account, as long as the account has sufficient permissions to load the agent profiler DLL and access the agent configuration and log files. [The "App.pool identity user" account is only required for IIS applications that run in an application pool5](#).

Reference: [.NET Agent Configuration Properties](#), [Configure the .NET Agent for Windows Services and Standalone Applications](#), [Instrument the DefaultDomain for Standalone Applications](#), [Install the .NET Agent for Windows](#), [Administer the .NET Agent](#)

Question: 30

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 libaio1.](#)

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 file2.](#)

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 file2.](#)

[Other steps that are not required but recommended for the Controller installation are verifying the user account permissions, configuring the virus scanners, installing the netstat network utility, and setting the NUMA configuration2.](#) [You do not need to install MySQL or Java separately, as they are included in the Controller installation package3.](#) Reference: [Prepare Linux for the Controller](#), [Install the Controller on Linux](#), and [Controller System Requirements](#) in the AppDynamics documentation.

Question: 31

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 properties1.](#) [The correct URL format for this API is1:](#)

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

transactions

[The other options are incorrect because12:](#)

Option A uses an invalid endpoint /allbts, which does not exist in the API.

Option C uses an incorrect base URL /controller/applications, which is used for the Controller UI, not the Rest API.

Option D uses a misspelled endpoint /allbis, which does not exist in the API. Reference: [Retrieve All Business Transactions in a Business Application, AppDynamics APIs](#)

Question: 32

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 troubleshooting12.](#)

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 productivity12.](#)

The incorrect options are:

Snapshots are not available after 2 weeks. (A) This is not a valid symptom because the AppDynamics Controller does not store snapshots for more than 2 weeks by default. Snapshots are detailed records of the execution context and call graphs of the business transactions that are monitored by the AppDynamics platform. Snapshots are useful for diagnosing performance issues and errors, but they also consume a lot of disk space. The AppDynamics Controller automatically purges the snapshots that are older than 2 weeks, unless the retention policy is changed by the user. [The availability of snapshots is not affected by the Controller scaling, unless the disk space is exhausted34.](#)

Health rule violations occur more frequently. (B) This is not a valid symptom because the AppDynamics Controller does not cause health rule violations to occur more frequently. Health rule violations are triggered when the performance or availability metrics of the monitored entities exceed the thresholds that are defined by the user. Health rule violations indicate the presence of performance issues or errors in the monitored applications, tiers, nodes, or other entities, not in the Controller itself. [The frequency of health rule violations is not affected by the Controller scaling, unless the Controller fails to collect or process the metrics data5.](#)

The average response times of tiers are higher than normal. (C) This is not a valid symptom because the

AppDynamics Controller does not affect the average response times of tiers. The average response time of a tier is the average time that the tier takes to process the incoming requests from the business transactions that are monitored by the AppDynamics platform. The average response time of a tier is influenced by the performance and behavior of the application code, the infrastructure, the dependencies, and the workload of the tier, not by the Controller itself. The average response time of a tier is not affected by the Controller scaling, unless the Controller fails to collect or process the metrics data .

Reference:

- [1:](#) Controller System Requirements - AppDynamics
- [2:](#) Controller Sizing Guidelines - AppDynamics
- [3:](#) Transaction Snapshots - AppDynamics
- [4:](#) Configure Data Retention - AppDynamics
- [5:](#) Health Rules - AppDynamics

Question: 33

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](#)¹². Reference: [Upgrade the Java Agent](#), [Release Upgrade Checklist for Java Agents](#)

Question: 34

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. Reference: [Events Service Deployment](#), [Install the Events Service on Linux](#), and [Install the Events Service on Windows](#) in the AppDynamics documentation.

Question: 35

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 unreachable1. This query does not require any authentication or parameters, and can be used as a quick and easy way to check the Controller status1.](#) Reference: [Ping the Controller](#)

Question: 36

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

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

Answer: C

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the minimum recommended number of nodes for a redundant Events Service is three ©. 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 scalability12.](#)

The incorrect options are:

- 1 (A): This is not a valid option because a single-node Events Service cluster does not provide any redundancy or resilience. If the node fails, the cluster becomes unavailable and the data is lost. [A single-node Events Service cluster is only suitable for testing or development purposes, not for production environments12.](#)
- 2 (B): This is not a valid option because a two-node Events Service cluster does not provide sufficient redundancy or resilience. If one node fails, the cluster becomes unstable and may lose data or availability. [A two-node Events Service cluster is not recommended for production environments12.](#)
- 4 (D): This is not a valid option because a four-node Events Service cluster is not optimal for redundancy or resilience. A four-node Events Service cluster has an even number of nodes, which may cause a split-brain scenario, where the cluster is divided into two equal partitions that cannot communicate with each other. This may result in data inconsistency or unavailability. [A four-node Events Service cluster can be improved by adding a fifth node to avoid the split-brain scenario12.](#) Reference:

[1:](#) Events Service Deployment - AppDynamics

[2:](#) Events Service Requirements - AppDynamics

Question: 37

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 users12.](#)

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 metrics34.](#)

The other statements are false because:

- B) The Machine Agent cannot send SNMP alerts. The Machine Agent can only receive SNMP traps from external sources and report them as events to the AppDynamics Controller. [The AppDynamics Controller can send SNMP alerts to external systems based on health rule violations or events, but this is not a function of the Machine Agent5.](#)
- C) The Machine Agent cannot communicate with multiple AppDynamics Controllers. The Machine Agent can only communicate with one Controller at a time, which is specified in the controller- info.xml file in the agent

configuration directory. If you want to monitor the same host machine with multiple Controllers, you need to install multiple Machine Agents on the same machine, each with a different Controller configuration and port number .

D) The Machine Agent cannot restart itself if it goes down. The Machine Agent does not have a built-in mechanism to automatically restart itself in case of a failure or a crash. You need to use an external tool or script to monitor the Machine Agent process and restart it if necessary. Alternatively, you can use the AppDynamics Agent Installer to deploy the Machine Agent as a service, which can be configured to restart automatically on failure .

Reference: [Analytics Agent](#), [Analytics Data](#), [HTTP Listener](#), [Custom Metrics](#), [SNMP Trap Alerting Integration](#), [SNMP Integration], [Machine Agent Configuration Properties], [Install the Machine Agent], [Agent Installer], [Start and Stop the Machine Agent]

Question: 38

What are two actions that an administrator should take to upgrade an EUM Server that is currently in production? (Choose two.)

- A. Stop the EUM agents,
- B. Upgrade the EUM agents.
- C. Update the EUM Server access key.
- D. Stop the EUM server before the upgrade.
- E. Run the new installer on the EUM host machine.

Answer: DE

Explanation:

To upgrade an EUM Server that is currently in production, the administrator should follow these steps:
Stop the EUM Server before the upgrade. This ensures that the EUM Server does not process any incoming data from the EUM agents during the upgrade process. [The administrator can use the eum.sh or eum.bat script to stop the EUM Server gracefully1.](#)

Run the new installer on the EUM host machine. The installer will detect the existing EUM Server installation and prompt the administrator to upgrade it to the latest version. The installer will also migrate the EUM data from the old version to the new version, if needed. [The administrator should follow the instructions on the installer wizard to complete the upgrade2.](#)

The other options are not necessary or correct for upgrading the EUM Server. The administrator does not need to stop or upgrade the EUM agents, as they are compatible with the new version of the EUM Server. The administrator does not need to update the EUM Server access key, as it remains the same after the upgrade. [The administrator does not need to install MySQL, as it is bundled with the EUM Server installation package2.](#)

Reference: [Upgrade the Production EUM Server](#) and [Start and Stop the EUM Server](#) in the AppDynamics documentation.

Question: 39

What describes the EUM agent?

- A. It communicates with the Machine Agent.
- B. It communicates with the AppDynamics Controller.
- C. It communicates with the Events Service.
- D. It communicates with the EUM Collector.

Answer: D

Explanation:

The EUM agent is a special agent that runs in web, mobile, or IoT applications and collects metrics on the end user experience. [The EUM agent communicates with the EUM Collector, which is a component that verifies, aggregates, and packages the raw metrics sent by the EUM agent¹². The EUM Collector can be either a SaaS service provided by AppDynamics or an on-premises server installed by the customer¹². The EUM Collector then sends the processed metrics to the](#)

[AppDynamics Controller and the Events Service for storage, analysis, and visualization¹²](#). Reference: [Overview of End User Monitoring, EUM Data](#)

Question: 40

A Java-based web application was instrumented. The browser snapshots provide a detailed look at an individual page request, however the correlated server-side snapshots are missing for all requests. What are two reasons for this missing correlated server-side snapshots? (Choose two.)

- A. Server has set the the HttpOnly flag on all cookies.
- B. Correlated server-side snapshots work only for .NET Applications.
- C. Correlated snapshots are visible only if the injection mechanism is Automatic.
- D. Correlated snapshots are visible only if browser is Chrome.
- E. Server side application is not instrumented with server agent.
- F. Correlated server-side snapshots are visible only if Java version is 1.7+.

Answer: AE

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the two reasons for the missing correlated server-side snapshots are:

Server has set the HttpOnly flag on all cookies. (A) This is a valid reason because the HttpOnly flag is a security feature that prevents client-side scripts from accessing the cookies. However, the AppDynamics JavaScript Agent relies on the cookies to correlate the browser snapshots with the server-side snapshots. The JavaScript Agent injects a cookie named `_appdyn_browser` into the browser requests, which contains the correlation information. If the server sets the HttpOnly flag on all cookies, including the `_appdyn_browser` cookie, the JavaScript Agent cannot read or modify the cookie, and the correlation fails. [To enable the correlation, the server should not set the HttpOnly flag on the `\_appdyn\_browser` cookie¹².](#)

Server-side application is not instrumented with server agent. (E) This is a valid reason because the server-side snapshots are collected by the AppDynamics app agents that instrument the application servers. The app agents monitor the business transactions that are executed by the server-side application, and capture the execution context, call graphs, errors, and metrics. If the server-side application is not instrumented with the app agent, the server-side snapshots are not available, and the correlation fails. [To enable the correlation, the server-side application should be instrumented with the app agent that is compatible with the application server and the Controller³⁴.](#)

The incorrect options are:

Correlated server-side snapshots work only for .NET Applications. (B) This is not a valid reason because the correlated server-side snapshots work for any application server that is instrumented with the AppDynamics app agent, not only for .NET applications. The AppDynamics platform supports various application servers, such as Java, .NET, PHP, Node.js, Python, and C/C++. [The app agents collect the server-side snapshots for the business transactions that are executed by the application server, regardless of the programming language or](#)

[framework34.](#)

Correlated snapshots are visible only if the injection mechanism is Automatic. © This is not a valid reason because the correlated snapshots are visible regardless of the injection mechanism. The injection mechanism refers to the way the AppDynamics JavaScript Agent is inserted into the web pages. There are two injection mechanisms: Automatic and Manual. The Automatic injection mechanism uses the app agent to inject the JavaScript Agent into the web pages that are served by

the application server. The Manual injection mechanism requires the user to manually insert the JavaScript Agent into the web pages. Both injection mechanisms support the correlation of the browser snapshots and the server-side snapshots, as long as the JavaScript Agent and the app agent are configured correctly .

Correlated snapshots are visible only if browser is Chrome. (D) This is not a valid reason because the correlated snapshots are visible regardless of the browser. The AppDynamics JavaScript Agent supports various browsers, such as Chrome, Firefox, Safari, Edge, and Internet Explorer. The JavaScript Agent collects the browser snapshots for the web pages that are loaded by the browser, and correlates them with the server-side snapshots, regardless of the browser type or version .

Correlated server-side snapshots are visible only if Java version is 1.7+. (F) This is not a valid reason because the correlated server-side snapshots are visible regardless of the Java version. The AppDynamics Java Agent supports various Java versions, such as 1.5, 1.6, 1.7, 1.8, and 11. The Java Agent collects the server-side snapshots for the business transactions that are executed by the Java application server, and correlates them with the browser snapshots, regardless of the Java version or vendor .

Reference:

- [1:](#) Browser Snapshots - AppDynamics
- [2:](#) Troubleshoot Browser RUM - AppDynamics
- [3:](#) Transaction Snapshots - AppDynamics
- [4:](#) Supported Environments and Versions - AppDynamics
- [5]: Browser Real User Monitoring - AppDynamics
- [6]: Set Up and Configure Web EUM - AppDynamics
- [7]: Browser Support - AppDynamics
- [8] : Java Agent - AppDynamics
- [9] : Java Supported Environments - AppDynamics

Question: 41

The user under which the JVM runs must have write privileges to the _____ directories in the Java Agent home. (Choose the correct option to complete the sentence.)

- A. conf and logs
- B. lib and logs
- C. bin and logs
- D. conf and lib

Answer: A

Explanation:

The user under which the JVM runs must have write privileges to the conf and logs directories in the Java Agent home. This is because the conf directory contains the agent configuration files, such as controller-info.xml and agent-config.xml, which may need to be modified or updated by the user. The logs directory contains the agent log files, such as agent.log and agent-errors.log, which are generated by the agent and may need to be accessed or rotated by the user. The user can install the agent as the same user that owns the JVM or as an administrator on the host machine. [The user can restrict the remaining contents of the agent directory, such as the lib and bin](#)

[directories, to read-only access12](#). Reference: [Install the Java Agent](#), [Java Agent fails to start](#)

Question: 42

Which data is unavailable in a hybrid deployment of AppDynamics where the AppDynamics Controller and Events Service are installed on-premises and the EUM Server is hosted in AppDynamics' SaaS cloud?

- A. Analytics metrics for End-User Monitoring data sets
- B. End-User Monitoring resource loading times
- C. End-User Monitoring session information
- D. End-User Monitoring browser snapshots

Answer: A

Explanation:

In a hybrid deployment of AppDynamics, where the AppDynamics Controller and Events Service are installed on-premises and the EUM Server is hosted in AppDynamics' SaaS cloud, the data that is unavailable is the analytics metrics for End-User Monitoring data sets. This is because the analytics metrics require the Events Service to store and process the unstructured data generated by the EUM agents. However, in a hybrid deployment, the EUM Server and the Events Service are not connected, and the EUM Server does not send the EUM data to the Events Service. [Therefore, the analytics metrics for EUM data sets, such as browser records, mobile snapshots, network requests, and custom events, are not available in the Controller UI or the Analytics UI1. The other data, such as resource loading times, session information, and browser snapshots, are available in the EUM Server UI, as they are stored and displayed by the EUM Server itself2](#). Reference: [Hybrid Deployment](#) and [EUM Data Sets](#) in the AppDynamics documentation.

Question: 43

Which three AppDynamics Controller properties govern how long metric data is retained in the database? (Choose three.)

- A. metrics.ten.min.retention.period
- B. metrics.ten.sec.retention.period
- C. metrics.retention.period
- D. metrics.min.retention. period
- E. metrics. day retention period
- F. metrics week retention period

Answer: ACD

Explanation:

[The AppDynamics Controller properties that govern how long metric data is retained in the database are1:](#)

metrics.ten.min.retention.period: This property specifies the number of days to retain metric data at

10-minute granularity. The default value is 32 days.

metrics.retention.period: This property specifies the number of days to retain metric data at 1-hour granularity. The default value is 365 days.

metrics.min.retention.period: This property specifies the number of hours to retain metric data at 1- minute granularity. The default value is 4 hours.

The other options are incorrect because1:

metrics.ten.sec.retention.period: This property does not exist in the AppDynamics Controller. The finest granularity for metric data is 1 minute.

metrics.day.retention.period: This property does not exist in the AppDynamics Controller. The coarsest granularity for metric data is 1 hour.

metrics.week.retention.period: This property does not exist in the AppDynamics Controller. The metric data retention is based on days, not weeks. Reference: [Database Size and Data Retention](#)

Question: 44

Instead of using the Enterprise Console UI, how can an administrator import an existing keypair to manage the Controller SSL certificate?

- A. Add the keypair to the keystore.jks using a third-party tool.
- B. Re-run the Controller installer and specify the new keypair.
- C. Upload a new keystore.jks file through the Controller UI.
- D. Upload the keypair from within the Controller UL.

Answer: A

Explanation:

According to the Cisco AppDynamics Professional Implementer (CAPI) documents, the method to import an existing keypair to manage the Controller SSL certificate without using the Enterprise Console UI is to add the keypair to the keystore.jks using a third-party tool (A). The keystore.jks file is the default keystore for the Controller that contains the private keys and certificates for the secure communication on port 8181. If the administrator already has a custom keypair that is signed by a third-party Certificate Authority (CA) or an internal CA, they can use a third-party tool, such as KeyStore Explorer or OpenSSL, to import the keypair into the keystore.jks file. The administrator should also import the root or intermediate certificates of the CA into the cacerts.jks file, which is the default truststore for the Controller. The administrator should use the keytool utility, which is bundled with the Controller installation, to import the certificates into the cacerts.jks file. [The administrator should also update the password for the keystore.jks and cacerts.jks files, and restart the Controller to apply the changes12.](#)

The incorrect options are:

Re-run the Controller installer and specify the new keypair. (B) This is not a valid method because the Controller installer does not allow the administrator to specify a custom keypair for the Controller SSL certificate. The Controller installer only allows the administrator to specify the Controller host name, port, account name, access key, and database settings. [The Controller installer does not modify the keystore.jks or cacerts.jks files, and does not import any custom keypair or certificate into the Controller keystore or truststore3.](#)

Upload a new keystore.jks file through the Controller UI. (C) This is not a valid method because the Controller UI does not provide any feature to upload a new keystore.jks file for the Controller SSL certificate. The Controller UI only allows the administrator to view and edit the Controller settings, such as the license, the security, the email, the analytics, and the EUM. [The Controller UI does not access or modify the keystore.jks or cacerts.jks files, and does not import any custom keypair or certificate into the Controller keystore or truststore4.](#)

Upload the keypair from within the Controller UI. (D) This is not a valid method because the Controller UI does not provide any feature to upload a custom keypair for the Controller SSL certificate. The Controller UI only allows the administrator to view and edit the Controller settings, such as the license, the security, the email, the analytics, and the EUM. [The Controller UI does not access or modify the keystore.jks or cacerts.jks files, and does not import any custom keypair or certificate into the Controller keystore or truststore4.](#)

Reference:

[1:](#) Controller SSL and Certificates - AppDynamics

- [2:](#) How do I resolve SSL certificate validation errors in the .NET Agent? - AppDynamics
- [3:](#) Install the Controller - AppDynamics
- [4:](#) Controller Settings - AppDynamics

Question: 45

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 data12.](#)

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 needed34.](#) The other statements are false because:

- A) They do not need to be installed on a bare metal server. The large and extra large performance profiles can be installed on any supported operating system, such as Linux or Windows, and on any supported platform, such as physical, virtual, or cloud. [However, the host machine should have enough CPU, memory, and network resources to meet the performance profile requirements12.](#)
- B) They do not need to be run with the High Availability Toolkit. The High Availability Toolkit is an optional tool that can be used to configure and manage a high availability deployment of the AppDynamics platform, where multiple Controllers are clustered together to provide redundancy and failover. The High Availability Toolkit can be used with any performance profile, not just the large and extra large ones. However, the High Availability Toolkit requires a license and additional hardware and software resources .
- C) They are supported on Windows. The large and extra large performance profiles can be installed on Windows Server 2012, 2012 R2, 2016, or 2019, as long as the host machine meets the performance profile requirements. [However, some features or configurations may not be available or supported on Windows, such as the High Availability Toolkit, the Enterprise Console, or the Events Service12.](#)

Reference: [Platform Requirements](#), [Controller System Requirements](#), [Controller Disk Space Requirements](#), [Controller Disk Space Alert](#), [High Availability], [High Availability Toolkit]

Question: 46

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 nodes¹](#).

The other options are not valid reasons for deploying a load balancer for a single-node Events Service cluster. Option A is incorrect, because the embedded Events Service is not meant to be used along with the single-node cluster, as it runs on the same machine as the Controller and does not offer data replication or scalability. [The embedded Events Service is only used by the Database Visibility product by default, and it is not recommended for production Application Analytics or EUM installations¹](#). Option B is incorrect, because hiding the Events Service server's real name is not a security or performance benefit, and it can be achieved by other means, such as DNS or firewall rules. Option D is incorrect, because a load balancer cannot provide redundancy for a single-node, as there is no other node to fail over to in case of a node failure. [A load balancer can only provide redundancy for a multi-node cluster, which has data replication and fault tolerance¹](#). Reference: [Events Service Deployment](#) in the AppDynamics documentation.

Question: 47

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 are12:](#)

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: 48

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 version12.](#)

Verify the Java Agent Version metric for that node in the Metric Browser. © 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 upgrade³⁴](#).

The incorrect options are:

V erify that the application log contains a message indicating success. (A) This is not a valid way because the application log does not contain any message indicating the success of the Java Agent upgrade. The application log is a file that records the events and messages that occur in the application server, such as the startup, shutdown, errors, warnings, and debug information. The application log does not record the events and messages that occur in the Java Agent, such as the installation, upgrade, configuration, or reporting. The Java Agent has its own log file, which is located in the <agent_home>/logs directory, and which contains the messages indicating the success or failure of the Java Agent upgrade .

V erify that the Java Agent log contains a message indicating the agent started successfully. (D) This is not a valid way because the Java Agent log does not contain any message indicating the agent started successfully after the upgrade. The Java Agent log is a file that records the events and messages that occur in the Java Agent, such as the installation, upgrade, configuration, or reporting. The Java Agent log contains a message indicating the agent started successfully when the application server is started, not when the Java Agent is upgraded. The Java Agent log does not contain any message indicating the success or failure of the Java Agent upgrade. The Java Agent log is located in the <agent_home>/logs directory, and the message indicating the agent started successfully is “Agent Startup Complete” .

Reference:

[1:](#) Tiers and Nodes - AppDynamics

[2:](#) Tier Dashboard - AppDynamics

[3:](#) Metric Browser - AppDynamics

[4:](#) Java Agent Metrics - AppDynamics

[5]: Java Agent Logs - AppDynamics

[6]: Troubleshoot the Java Agent - AppDynamics

Question: 49

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. WMI
- D. SSH

Answer: B

Explanation:

[The Database Agent collects hardware metrics from a Windows database server using PowerShell¹. 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 host². The Database Agent uses PowerShell to collect metrics such as CPU, memory, disk, and network utilization from the Windows database server¹. To enable hardware monitoring for a Windows database server, the Database Agent requires the following permissions¹:](#)

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.

Reference: [Required Monitored Host Permissions](#), [PowerShell Overview](#)

Question: 50

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.

Reference: [.NET Agent](#) and [How do I deploy a .NET Agent?](#) in the AppDynamics documentation and community.