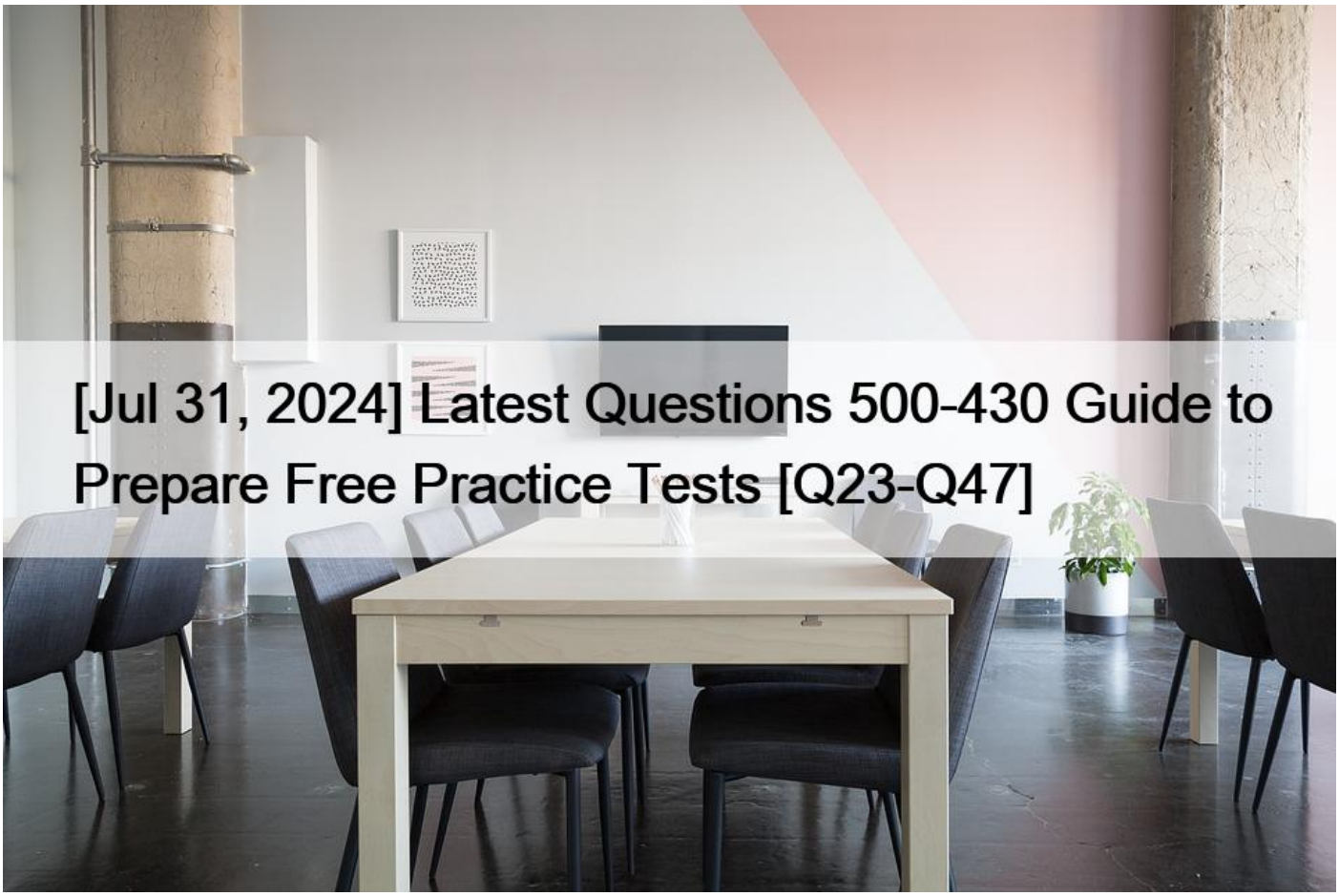
# [Jul 31, 2024 Latest Questions 500-430 Guide to Prepare Free Practice Tests [Q23-Q47




**[Jul 31, 2024 Latest Questions 500-430 Guide to Prepare Free Practice Tests Reliable 500-430 Dumps Questions Available as Web-Based Practice Test Engine**

To prepare for the Cisco 500-430 exam, there are a variety of resources available, including study guides, practice exams, and training courses. These resources can help you understand the exam objectives and give you the knowledge and skills you need to pass the exam on your first attempt. Additionally, Cisco offers a certification program that provides ongoing education and training to help IT professionals stay up-to-date with the latest technology trends and best practices.

Cisco 500-430 certification exam, also known as the Cisco AppDynamics Professional Implementer exam, is designed to test the skills and knowledge of IT professionals who specialize in implementing and managing AppDynamics solutions. Cisco AppDynamics Professional Implementer certification exam is ideal for professionals who have experience working with AppDynamics in large-scale environments and want to demonstrate their expertise to employers and peers.

**NO.23** Which two statements are true regarding the AppDynamics REST API for retrieving metrics? (Choose two.)

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

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. References: Retrieve Metric Data, Retrieve Metric Hierarchy

**NO.24** What are three recommended steps to prepare a Linux environment for the installation of an AppDynamics Controller with a Large performance profile? (Choose three.)

* Install libaio,
* Install MySQL.
* Verify the user account has root access,
* Verify the open file descriptor limit.
* Verify that Java is installed.
* Verify the process limit.

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. References: Prepare Linux for the Controller, Install the Controller on Linux, and Controller System Requirements in the AppDynamics documentation.

**NO.25** The Database Agent collects hardware metrics from a Windows database server using_____ . (Choose the correct option to complete the sentence.)
* Standalone Machine Agent
* PowerShell
* WHI
* SSH

Explanation

The Database Agent collects hardware metrics from a Windows database server using PowerShell

1. 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 host12. 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. References: Required Monitored Host Permissions, PowerShell Overview

**NO.26** Which two user accounts are created by the AppDynamies Controller during installation? (Choose two.)
* Elastic search root user
* GlassFish asadmin user
* Customer-specified Controller administrator account
* OS user that will run the controller
* REST API user
* MySQL appd admin user

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 tool1.

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 installation2.

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 installation3. References: Controller Installation, Manage Users and Groups, and Update the Root User and Glassfish Admin Passwords in the AppDynamics documentation.

**NO.27** Which framework would require the implementation of custom correlation?
* Custom TCP concurrent server
* Customer proprietary SOAP application
* Vendor-supplied enterprise application that uses JMS
* Pre-packaged WCF application

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. References: Custom Correlation for Java Applications and Configure Custom Correlation for

.NET Applications in the AppDynamics community.

**NO.28** What is the correct method to perform a NET Agent upgrade?
* Perform the agent upgrade on the application server host by running the MSI Installer Package.
* Perform the agent upgrade on a remote server host by using the AppDynamics Controller REST API.
* Perform the agent upgrade on the application server host by running the Agent Configuration Utility.
* Perform the agent upgrade via the AppDynamics Controller Ul.

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 youraccount 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.

(B) 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.

References:

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

**NO.29** A company set up an on-premises AppDynamics Controller and an on-premises Events Service cluster. What describes setting up this Events Service cluster?
* The Events Service API ports cannot be reconfigured from the default 9080/2081.
* The Controller must be able to initiate communication with the Events Service cluster.
* The Events Service cluster must be able to initiate communication with the Controller.
* There is no need for a load balancer or virtual IP address in front of the Events Service cluster.

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. References: Events Service Deployment, Install the Events Service on Linux, and Install the Events Service on Windows in the AppDynamics documentation.

**NO.30** What are three requirements to set up AppDynamics Controllers as a high availability pair? (Choose three.)
* Passwordless SSH must be configured between the two Controller servers.
* The Controller MySQL database must be installed on a shared location.
* The replicate sh script can be run only once.
* Both servers must have the Controller software installed prior to setting up high availability.
* A unique high availability license file is required for each Controller server.
* Both servers must have identical directory structures for the Controller installation.

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. References: Prerequisites for High Availability

**NO.31** What are two advantages of using an Events Service cluster? (Choose two.)
* Clusters allow data replication across multiple nodes.
* Clusters are easier to maintain than single-node instances.
* Clusters are horizontally scalable by adding nodes.
* Clusters reduce the load on the AppDynamics Controller.
* Clusters expose multiple channels for simultaneous queries.

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. References: Events Service Deployment, What are the Benefits of Server Clustering?

**NO.32** Which artifacts are needed, at minimum, to add a custom extension to a Machine Agent?
* Directory

Jar or Script file

Monitor.xml file
* Directory

Manitor.xml file

Custom-gxtension.xml file
* Directory

Zip Extension

Monitor.xml file
* Directory

Script file

Config.json file
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 yourextension12. References: Build a Monitoring Extension Using Scripts, Extensions and Custom Metrics

**NO.33** Default configuration of the Mobile SDK enables____and_____ . (Choose the two correct options to complete the sentence.)
* Crash Reports
* User Data
* Network Requests
* Custom Timers
* Breadcrumbs

Explanation

The default configuration of the Mobile SDK enables crash reports and network requests12. Crash reports capture and report any unhandled exceptions or signals that cause the app to terminate abnormally1. Network requests monitor the performance and errors of HTTP and HTTPS requests made by the app2. These features are enabled by default and do not require any additional code or configuration to work12. References: Crash Reports, Network Requests

**NO.34** 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)?
* dedicated EUM HTTP(s) ports
* GlassFish administration port
* Controller database and HTTP(s) ports
* Controller primary HTTP{s) port

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. References: Controller Port Settings, Configure the Port for the EUM Agent, Access the Administration Console, Controller System Requirements

**NO.35** What are the correct steps to install a .NET Agent patch?
* Uninstall the existing .NET Agent

Install the patch

Restart the instrumented application(s)
* Restart the machine

Install the patch over exiting .NET agent

Restart instrumented applications(s)
* Install the .NET Agent patch

Restart the instrumented application(s)
* Restart the instrumented application(s)

Apply the patch over existing NET agent
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.

References: .NET Agent and How do I deploy a .NET Agent? in the AppDynamics documentation and community.

**NO.36** Which implementation mode should be used when deploying an EUM Server with reverse proxy from a security and performance standpoint in a production environment?
* Install the EUM Server on the same host as the Controller, behind the reverse proxy.
* Install the EUM Server on a separate host from the Controller, in front of the reverse proxy.
* Install the EUM Server on the same host as the Controller, in front of the reverse proxy.
* Install the EUM Server on a separate host from the Controller, behind the reverse proxy.
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. References: EUM Server Deployment, Install a Production EUMServer, Reverse Proxy configuration for EUM, and Browser EUM installation in the AppDynamics documentation and community.

**NO.37** Which two user accounts are created by the AppDynamies Controller during installation? (Choose two.)
* Elastic search root user
* GlassFish asadmin user
* Customer-specified Controller administrator account
* OS user that will run the controller
* MySQL appd admin user
* REST API user

**NO.38** 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.)

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

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 .

References:

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

**NO.39** 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)?
* Controller database and HTTP(s) ports
* dedicated EUM HTTP(s) ports
* Controller primary HTTP{s) port
* GlassFish administration port

**NO.40** What is the minimum recommended number of nodes for a redundant Events Service?
* 1
* 2
* 3
* 4
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.

References:

1: Events Service Deployment – AppDynamics

2: Events Service Requirements – AppDynamics

**NO.41** Which two symptoms occur if an AppDynamics Controller is NOT scaled correctly? (Choose two.)
* Snapshots are NOT available after 2 weeks.
* Health rules violations occur more frequently.
* The average response times of tiers are higher than normal.
* The Controller’s metric reporting is 7 to 10 minutes behind the current time.
* The Controller Ul performs slowly.

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. 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 .

References:

1: Controller System Requirements – AppDynamics

2: Controller Sizing Guidelines – AppDynamics

3: Transaction Snapshots – AppDynamics

4: Configure Data Retention – AppDynamics

5: Health Rules – AppDynamics

**NO.42** The AppDynamics Controller is instrumented by an internal, out-of-the-box, AppDynamics Java agent. Which account and user name are used to connectto the Controller to view theinformation provided by the internal AppDynamics agent?
* The account is ‘root’ and the user is ‘admin’.
* The account is ‘customer!’ and the user is ‘root’.
* The account is ‘system’ and the user is “root.
* The account is internal’ and the user is ‘admin’.

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. References: Controller Self-Monitoring, Monitoring a Controller Using the Internal Monitoring Agent, Controller Accounts

**NO.43** What becomes more important as an AppDynamics Controller grows beyond supporting 500 agents?

* CPU utilization
* RAM allocated to the Controller
* Network throughput
* Disk VO
* Thread count on the GlassFish server

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 load123. References: Controller System Requirements, Performance and Controller Sizing Guidelines, How to Run AppDynamics in Microsoft Azure