Free Cisco 500-430 CAPI Practice Questions

Shared by Vinson on 13-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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

Options:

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:

A, D, F

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 2

Question Type: MultipleChoice

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

Options:

A- Directory
Jar or Script file
Monitor.xml file
B- Directory
Manitor.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 3

Question Type: MultipleChoice

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

Options:

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:

A, E

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 4

Question Type: MultipleChoice

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

Options:

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

Answer:

B, D

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 APIandRetrieve All Business Applicationsin the AppDynamics documentation.

Question 5

Question Type: MultipleChoice

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

Options:

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. 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.

1: Cisco AppDynamics Professional Implementer (500-430)

2: Events Service Requirements - AppDynamics

3: Events Service Deployment - AppDynamics

4: Events Service Deployment - AppDynamics

Question 6

Question Type: MultipleChoice

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

Options:

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. xmi file.

Answer:

A, B

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.

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 7

Question Type: MultipleChoice

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

Options:

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

Answer:

B

Explanation:

The Database Agent collects hardware metrics from a Windows database server usingPowerShell1.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.Reference:Required Monitored Host Permissions,PowerShell Overview

Question 8

Question Type: MultipleChoice

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

Options:

A- It can act as a forwarder for analytics events.
B- lt 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:

A, E

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 .