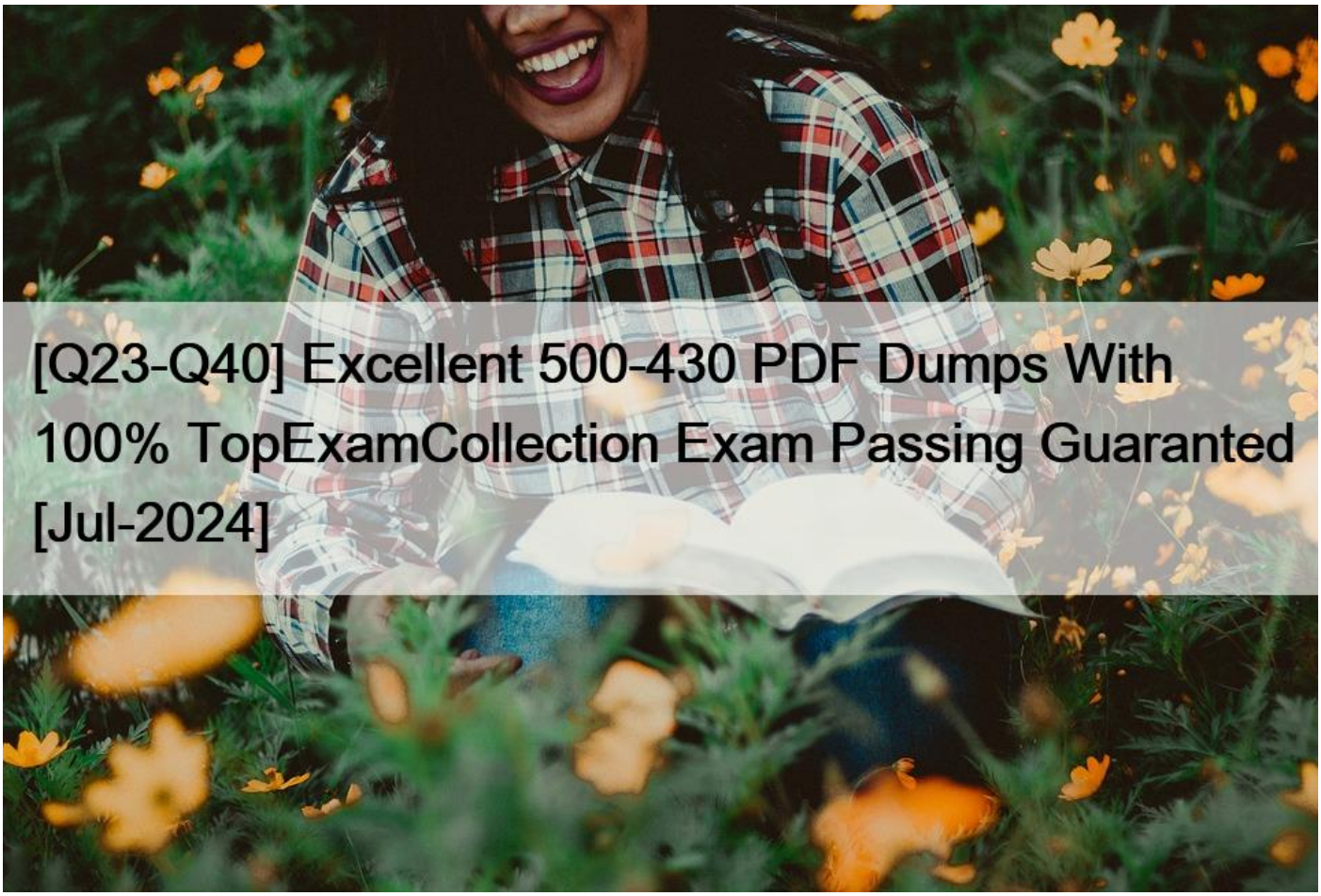
[Q23-Q40 Excellent 500-430 PDF Dumps With 100% TopExamCollection Exam Passing Guaranted [Jul-2024



Excellent 500-430 PDF Dumps With 100% TopExamCollection Exam Passing Guaranted [**Jul-2024**]
100% Pass Your 500-430 Cisco AppDynamics Professional Implementer at First Attempt with TopExamCollection

The Cisco AppDynamics Professional Implementer certification exam consists of 60-70 multiple-choice questions and has a time limit of 90 minutes. 500-430 exam is available in English and Japanese and can be taken online or in person at a Pearson VUE testing center. The passing score for the exam is 70%, and candidates who pass the exam receive a Cisco Certified AppDynamics Professional Implementer certification.

Cisco AppDynamics platform is a powerful tool for monitoring and managing application performance in complex environments. By earning the Cisco Certified AppDynamics Professional Implementer certification, IT professionals can demonstrate their ability to use this platform to its full potential, making them valuable assets for organizations looking to improve their application performance and user experience.

NEW QUESTION 23

Which URL retrieves all AppDynamics business applications from an AppDynamics Controller using the AppDynamics Rest API?
* http(s)://<cantroller-host=:<part=/controller/rest/businessapplications
* http(s)://<controller-host>:<port>/controller/allapplications
* httpis)://<controller-host>:<port>/controller/rest/applications
* http(s)://<controller-host>:<port>/controller/applications
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-mi 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. References: Application Model API, Metric and Snapshot API

NEW QUESTION 24

Which directory should an administrator back up if the goal is to back up the EUM Server?
* <eum_server_home= directory
* <controller_home=/bin directory
* <controller_home>
* <controller_home>/bin/eum_server directory
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. References: Upgrade the Production EUM Server, Configure the EUM Server

NEW QUESTION 25

Which URL retrieves all AppDynamics business transactions from an application using the AppDynamics Rest API?
* http(s)://<controller-host>:<port=/controller/rest/applications/<application_name>/allbts
* http(s)://<controller-host>:<port>/controller/rest/applications/<application_name=>/business-transactions
* http(s)://<controller-host>:<port>/controller/applications/<application_name>/business-transactions
* httpis)://<controller-host>:<port>/controller/applications/<application_name=>/allbis
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. References: Retrieve All Business Transactions in a Business Application, AppDynamics APIs

NEW QUESTION 26

Which two choices are available when specifying an application in a URL string for the Health Rule REST API? (Choose two.)
* Application Alias
* Application ID
* Application GUID
* Application Name
* Application REGEX
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. References: Health Rule API and Retrieve All Business Applications in the AppDynamics documentation.

NEW QUESTION 27

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

NEW QUESTION 28

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

NEW QUESTION 29

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

* Install JRE.
* Ensure that MySQL port (3388) is opened.
* Install SSH.
* Install libaio.
* Verify that sufficient temporary (tmp) space is available (at least 1 GB).

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

NEW QUESTION 30

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

* It can act as a forwarder for analytics events.
* lt can send SNMP alerts.
* It can communicate with multiple AppDynamics Controllers.
* It can restart itself if it goes down.
* It can start an HTTP listener for custom metrics.

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 youcan 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 .

References: 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]

NEW QUESTION 31

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 inAppDynamics’ SaaS cloud?

* Analytics metrics for End-User Monitoring data sets
* End-User Monitoring resource loading times
* End-User Monitoring session information

* End-User Monitoring browser snapshots

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. References: Hybrid Deployment and EUM Data Sets in the AppDynamics documentation.

NEW QUESTION 32

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

NEW QUESTION 33

What describes the EUM agent?

* It communicates with the Machine Agent.
* It communicates with the AppDynamics Controller.
* It communicates with the Events Service.
* It communicates with the EUM Collector.

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. References: Overview of End User Monitoring, EUM Data

NEW QUESTION 34

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

* Add a new Events Service cluster to share the load.
* Add nodes running on machines with identical hardware matching the existing nodes.
* Add more storage to the master nodes of the cluster.
* Add more storage to as many of the existing nodes as possible.

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.

References:

1: Cisco AppDynamics Professional Implementer (500-430)

2: Events Service Requirements – AppDynamics

3: Events Service Deployment – AppDynamics

4: Events Service Deployment – AppDynamics

NEW QUESTION 35

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

NEW QUESTION 36

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

NEW QUESTION 37

What are three reasons you would create custom events using the Machine Agent REST API? (Choose three.)
* to create an event to track application deployment
* to create an event to be displayed in a Controller Audit report
* to create an alert that is to be triggered when a custom event is created
* to create an event to be displayed along with Time Series data in a custom dashboard
* to create an event to be used to trigger a health rule violation
* to create a new metric

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. References: Machine Agent HTTP Listener, Create Custom Events

NEW QUESTION 38

Which two statements are true when updating the Database Agent? (Choose two.)
* The Database Agent must be stopped and restarted during the upgrade.
* 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.
* All data collectors created from the previous agent must be migrated to the new agent.
* Controller-info.xml is the only file that needs to be migrated from the previous agent to the new agent.
* After the Database Agent is upgraded, the AppDynamics Controller must be restarted.

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>confcontroller-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>confcontroller-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.

References:

1: Upgrade the Database Agent – AppDynamics

2: Release Upgrade Checklist for Database Agents – AppDynamics

NEW QUESTION 39

What are two recommendations for servers in an Events Service Cluster? (Choose two.)
* Should be installed using the same user account,
* Should have eight or more CPU cares,
* Should be on the same local network,
* Should have identical hardware specifications.
* Should be running the same operating system version

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 : 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 Cluster12.

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 type12.

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 either12.

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 recommendation12.

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 hosts12.

References:

1: Events Service Requirements – AppDynamics

2: Events Service Deployment – AppDynamics

NEW QUESTION 40

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.