Vendor: Appian

Exam Code:ACD300

Exam Name: Appian Certified Lead Developer

Version: Demo

QUESTION 1

You are designing a process that is anticipated to be executed multiple times a day. This process retrieves data from an external system and then calls various utility processes as needed. The mam process will not use the results of the utility processes, and there are no user forms anywhere.

Which design choice should be used to start the utility processes and minimize the load on the execution engines?

A. Use the Start Process Smart Service to start the utility processes.

B. Start the utility processes via a subprocess synchronously.

C. Use Process Messaging lo star! the utility process.

D. Start the utility processes via a subprocess asynchronously

Correct Answer: C

To design a process that is anticipated to be executed multiple times a day, that retrieves data from an external system and then calls various utility processes as needed, you should use Process Messaging to start the utility process and minimize the load on the execution engines. Process Messaging is a feature that allows you to send and receive messages between processes in Appian. By using Process Messaging, you can start the utility process asynchronously, which means that the main process does not have to wait for the utility process to finish before continuing. This way, you can improve the performance and scalability of your process design, and reduce the load on the execution engines. The other options are not as effective. Option A, using the Start Process Smart Service to start the utility processes, would also start the utility process asynchronously, but it would require more configuration and maintenance than Process Messaging. Option B, starting the utility processes via a subprocess synchronously, would start the utility process as a part of the main process flow, which means that the main process would have to wait for the utility process to finish before continuing. This would reduce the performance and scalability of your process design, and increase the load on the execution engines. Option D, starting the utility processes via a subprocess asynchronously, would also start the utility process as a part of the main process flow,but it would not wait for the utility process to finish before continuing. However, this option would still create more overhead than Process Messaging, as it would create more instances of processes in Appian.

QUESTION 2

You are running an inspection as a part of the first deployment process from TEST to PROD. You receive a notice that one of your objects will not deploy because it is dependent on an object from an application owned by a separate team.

What should be your next step?

A. Create your own object with the same code base, replace (he dependent object in the application. and deploy to PROO.

B. Halt the production deployment and contact the other team tor guidance on promoting the object to PROD

C. Check the dependencies of the necessary object Deploy w PROO if there are few dependencies and it is low risk

D. Push a functionally viable package to PROD without the dependencies, and plan the rest o! the deployment accordingly with the other team\\'s constraints

Correct Answer: B

Deploying an object that is dependent on another object from a different application can cause errors and inconsistencies in the production environment. The best practice is to halt the production deployment and contact the other team for guidance on how to promote the object to PROD. The other team may have a different deployment schedule, or they may have some dependencies or customizations that need to be considered. By communicating with the other team, you can ensure that the object is deployed in a safe and coordinated manner, and avoid any potential conflicts or issues. Verified References: [Appian Deployment Guide], [Appian Best Practices]

QUESTION 3

You have created a Web API in Appian. with the following URL to call it:
https://exampleappiancloud.com/suite/webapi/usef_managefnent/ users ?username=)=john.smith.

Which is the connect syntax forreferring to the user name parameter\\'

A. httpirequest.queryParameters users username

B. httpirequest usees username

C. httpirequest formData username

D. httpirequest queryParameters.username

Correct Answer: D

The correct syntax for referring to the username parameter in the Web API URL is httpirequest.queryParameters.username. This syntax allows you to access the value of the username parameter that is passed in the query string of the URL after the question mark (?). For example, if the URL ishttps://exampleappiancloud.com/suite/webapi/user_management/users?username=john. smith, then httpirequest.queryParameters.username will return john.smith. Verified References: Appian Documentation, section "Web API".

QUESTION 4

You have an active development team (Team A) building enhancements for an application (App X\\'). and ate currently using the TEST environment for UAT.

A separate operations team (\\'Team B) discovers a critical error in the Production instance of App X that they must remediate However. Team 6 does not have a hotfix stream for which to accomplish this The available environments are DEV. TEST, and PROD

Which risk mitigation effort should both teams employ to ensure Team AS capital project is only minorly interrupted, and Team B s critical fix can be completed and deployed quickly to end users?

A. Team 8 must communicate to Team A which component will be addressed in the hotfix to avoid overlap of changes It overlap exists, the component must be versioned to its PROD state before being remediated and deployed, and then versioned back to its latest development state If overlap does not exist, the component may be remediated and deployed without any version changes

B. Team A must analyze their current codebase in OEV lo merge the hotfix changes into their latest enhancements. Team B is then requited to wait for the hotfix to follow regular deployment protocols from DEV to the PROO environment.

C. Team 8 must address changes in the TEST environment These changes can then be tested and deployed directly to

PROD. Once the deployment is complete. Team B can then communicate their changes to Teams to ensure they are Incorporated as a part of the next release.

D. Team 8 must address the changes directly in PROD. As there is no hotfix stream, and OEV and TEST are being utilized for active development it is best to avoid a conflict of components. Once Team A has completed their enhancements work. Team 6 can update DEV and TEST accordingly.

Correct Answer: A

This is the best risk mitigation effort that both teams can employ to ensure that Team A\\'s capital project is only minorly interrupted, and Team B\\'s critical fix can be completed and deployed quickly to end users. By communicating with Team A, Team B can identify which component is causing the critical error in PROD, and check if there is any overlap of changes with Team A\\'s enhancements. If there is an overlap, Team B can version the component to its PROD state, which is the last stable version, before making any changes to fix the error. Then, Team B can deploy the fixed component to PROD, and version it back to its latest development state, which includes Team A\\'s enhancements. This way, Team B can avoid overwriting or losing any of Team A\\'s work, and ensure that the component is consistent across all environments. If there is no overlap, Team B can simply make the changes to the component and deploy it to PROD, without affecting Team A\\'s work. The other options are not as effective. Option B, having Team A analyze their current codebase in DEV to merge the hotfix changes into their latest enhancements, would delay the deployment of the critical fix, as Team B would have to wait for Team A to finish their analysis and merge. Option C, having Team B address the changes in TEST, would interrupt Team A\\'s UAT process, and could cause conflicts or errors in TEST or PROD. Option D, having Team B address the changes directly in PROD, would be risky and not recommended, as it could introduce new errors or inconsistencies in PROD. Verified References: [Appian Deployment Guide], [Appian Best Practices]

QUESTION 5

You are planning a strategy around data volume testing for an Appian application that queries and writes to MySQL database.

You have administrator access to the Appian application and to the database.

What are two key considerations when designing a data volume testing strategy?

A. Data from previous tests needs to remain in the testing environment prior to loading prepopulated data

B. large datasets must be loaded via Applan processes

C. The amount of data that needs to be populated should be determined by the project sponsor and the stakeholders based on their estimation

D. Testing with the correct amount of data should be in the definition of done as part of each sprint.

E. Data model changes must wait until towards the end of the protect.

Correct Answer: DE

When designing a data volume testing strategy for an Appian application that queries and writes to MySQL database, you should consider two key considerations: Testing with the correct amount of data should be in the definition of done as part of each sprint. Data volume testing is a type of testing that verifies how well an application performs when handling large amounts of data. Data volume testing is important to ensure that the application meets the performance and quality requirements of the users and stakeholders. By including data volume testing in the definition of done as part of each sprint, you can ensure that each feature or functionality of your application is tested with realistic data volumes before being delivered to production. This way, you can identify and resolve any potential issues or bottlenecks early in the development cycle, and avoid any surprises or delays later on. Data model changes must wait until towards

the end of the project. Data model changes are changes that affect the structure or schema of your database, such as adding, modifying, or deleting tables, columns, indexes, or constraints. Data model changes are risky and costly to make, especially when dealing with large amounts of data. Data model changes can affect the performance, functionality, or integrity of your application and database. Therefore, data model changes must wait until towards the end of the project, when you have finalized your requirements and design decisions, and have minimized your data volume testing efforts. By waiting until towards the end of the project to make data model changes, you can reduce the impact and complexity of those changes, and avoid any unnecessary rework or regression. The other options are not as effective. Option A, data from previous tests needs to remain in the testing environment prior to loading prepopulated data, is not a key consideration for designing a data volume testing strategy, but rather a best practice for preparing your testing environment. Option B, large datasets must be loaded via Appian processes, is not a key consideration for designing a data volume testing strategy, but rather a technical implementation detail that may or may not be suitable for your application. Option C, the amount of data that needs to be populated should be determined by the project sponsor and the stakeholders based on their estimation, is not a key consideration for designing a data volume testing strategy, but rather an input or assumption that you need to validate before conducting your data volume testing.

QUESTION 6

You are presented with the following application requirement:

Users must be able to navigate throughout the application while maintaining complete visibility in the

application structure, and easily navigate to previous locations.\\'\\'

Which Appian Interface Pattern would you recommend?

A. Use Bullous as Cards pattern on the home page lo prominently display application choices.

B. Implement an Activity History pattern to track an organizations activity measures.

C. implement a drilldown report pattern to show detailed information about report data.

D. Include a breadcrumbs pattern on applicable inert aces to show the organizational hierarchy

Correct Answer: C

To meet the application requirement of allowing users to view summary and detailed information about report data, you should implement a drilldown report pattern to show detailed information about report data. A drilldown report pattern is a user interface component that displays data in a hierarchical structure, and allows users to expand or collapse different levels of data. For example, if the user is viewing a sales report by region, the drilldown report pattern could show something like "North America > USA > California > Los Angeles". The user can click on any level of data to see more or less details. This way, the user can see both summary and detailed information about report data, and explore different aspects of the data. The other options are not as effective. Option A, using Tiles as Cards pattern on the home page to prominently display application choices, would provide a way for users to access different parts of the application from the home page, but it would not show summary or detailed information about report data. Option B, implementing an Activity History pattern to track an organization\\'s activity measures, would provide a way for users to see the recent actions performed by themselves or others in the application, but it would not show summary or detailed information about report data. Option D, including a breadcrumbs pattern on applicable interfaces to show the organizational hierarchy, would provide a way for users to see where they are in the application, and easily go back to any previous level by clicking on the corresponding link, but it would not show summary or detailed information about report data.

QUESTION 7

You are taskedto build a large scale acquisition application for a prominent customer. The acquisition process tracks the time it takes is fulfill a purchase request with an award.

The customer has structured the contract so that there are multiple application dev teams.

How should you design for multiple processes and forms, while minimizing repeated code?

A. Create a Center of Excellence (CoE)

B. Create a common objects application.

C. Create a Scrum of Scrums sprint meeting for the team leads

D. Create duplicate processes and forms as needed

Correct Answer: B

To build a large scale acquisition application for a prominent customer, you should design for multiple processes and forms, while minimizing repeated code. One way to do this is to create a common objects application, which is a shared application that contains reusable components, such as rules, constants, interfaces, integrations, or data types, that can be used by multiple applications. This way, you can avoid duplication and inconsistency of code, and make it easier to maintain and update your applications. You can also use the common objects application to define common standards and best practices for your application development teams, such as naming conventions, coding styles, or documentation guidelines. Verified References: [Appian Best Practices], [Appian Design Guidance]

QUESTION 8

HOTSPOT

For each requirement, match the most appropriate approach to creating or utilizing plug-ins Each approach will be used once.

Note: To change your responses, you may deselect your response by clicking the blank space at the top of the selection list.

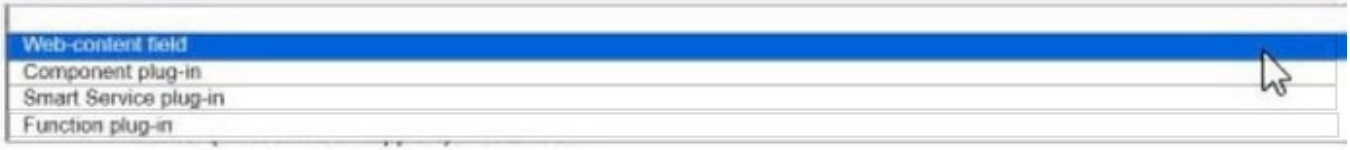
Hot Area:

Read barcode values from images containing barcodes and QR codes.

Select a match:

Display an externally hosted geolocation/mapping application's interface within Appian to allow users of Appian to see where a customer (stored within Appian) is located.

Select a match:

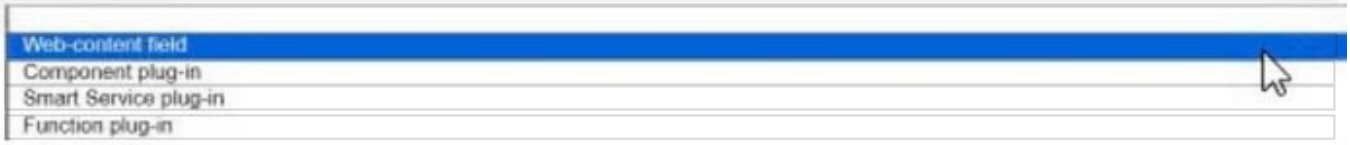


Display an externally hosted geolocation/mapping application's interface within Appian to allow users of Appian to select where a customer is located and store the selected address in Appian.

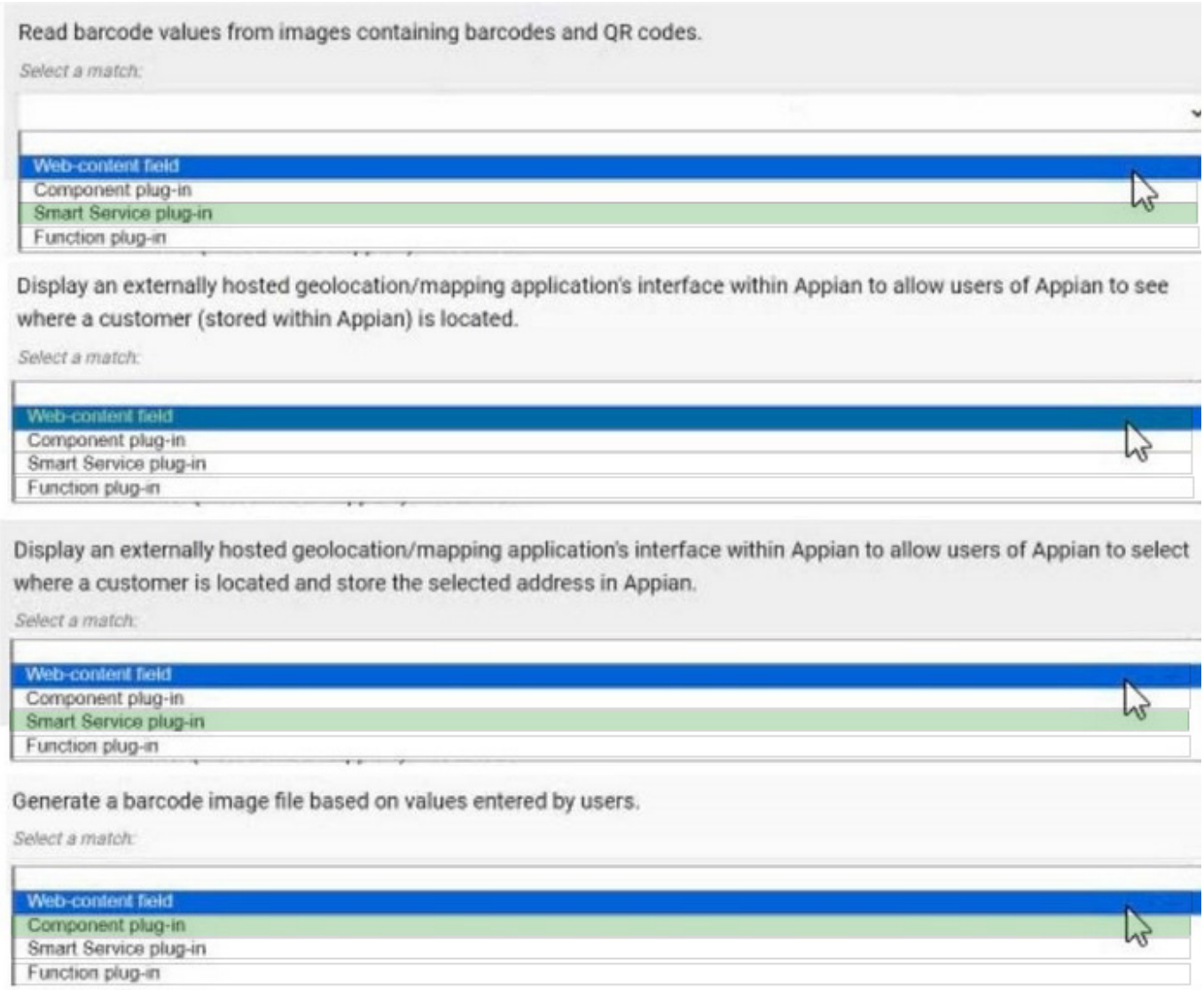
Select a match:

Generate a barcode image file based on values entered by users.

Select a match:



Correct Answer:



Requirement: Read barcode values from images containing barcodes and QR codes. Correct approach: C. Smart Service plug-in Exact explanation of correct approach taken from Appian Documentation: A smart service plug-in is a type of plug-in that allows you to create custom smart services that can be used in process models. A smart service can perform complex logic, interact with external systems, or manipulate data in Appian. A smart service plug-in can also leverage Java code to implement the functionality of the smart service. A smart service plug-in would be suitable for reading barcode values from images, as it can use Java libraries or APIs that can scan and decode barcodes and QR codes from image files. A smart service plug-in can also return the barcode values as outputs that can be used by other nodes or processes in Appian. A smart service plug-in can also be configured with input parameters, such as the image file, the barcode type, or the output format, that can customize the behavior of the smart service. A smart service plug-in can also have error handling and logging features that can handle any exceptions or failures that might occur during the barcode reading process. Requirement: Display an externally hosted geolocation mapping applications interface within Appian to allow users of Appian to see where a customer (stored within Appian)is located. Correct approach: A. Web-content field Exact explanation of correct approach taken from Appian Documentation: A web-content field is a type of user interface component that allows you to display web content from an external source in a SAIL interface. A web-content field would be suitable for displaying an externally hosted geolocation mapping applications interface, as it can embed the web content in an iframe and render it within the Appian interface. You can also pass parameters to the web content, such as the customer\\'s location, using the url parameter of the web-content field. A web- content field can also interact with other components in the Appian interface, such as buttons, grids, or forms, using the postMessage API. This way, you can create a seamless user experience that integrates the external geolocation mapping applications interface with the Appian functionality. Requirement: Display an externally hosted geolocation mapping applications

interface within Appian to allow users of Appian to select where a customer is located and store the selected address in Appian. Correct approach: A. Web-content field and C. Smart Service plug-in Exact explanation of correct approach taken from Appian Documentation: A web- content field and a smart service plug-in would be suitable for displaying an externally hosted geolocation mapping applications interface within Appian to allow users of Appian to select where a customer is located and store the selected address in Appian. A web- content field would be suitable for displaying the external geolocation mapping applications interface, as explained above. A smart service plug-in would be suitable for storing the selected address in Appian, as it can use Java code to receive the address data from the web content, validate it, and write it to a data store entity or a process variable. Requirement: Generate a barcode image file based on values entered by users. Correct approach: B. Component plug-in Exact explanation of correct approach taken from Appian Documentation: A component plug-in is a type of plug-in that allows you to create custom user interface components that can be used in SAIL interfaces. A component plug-in can also leverage Java code to implement the functionality of the component. A component plug-in would be suitable for generating a barcode image file, as it can use Java libraries or APIs that can encode values into barcode formats and generate image files. A component plug-in can also display the barcode image file in the Appian interface and allow users to download or print it. A component plug-in can also interact with other components in the Appian interface, such as text fields, buttons, or forms, using the a!refreshVariable() function. This way, you can create a dynamic user experience that updates the barcode image file based on the values entered by users.

QUESTION 9

You are reviewing the Engine Performance Logs in Production for a single application thathas been live for six months. This application experiences concurrent user activity and has a fairly sustained load during business hours. The client has reported performance issues with the application during business hours.

During your investigation, you notice a high Work Queue - Java Work Queue Size value in the logs You also notice unattended process activities, including timer events and sending notifications emails, are taking far longer to execute than normal.

The client Increased the number of CPU cores prior to the application going live

What is the next recommendation?

A. Add more engine replicas.

B. Optimize slow-performing user interfaces.

C. Add more application servers.

D. Add execution and analytics shards

Correct Answer: A

Adding more engine replicas will increase the number of threads available to execute unattended process activities, such as timer events and sending notification emails. This will reduce the Java Work Queue Size and improve the performance of the application. Verified References: Appian Engine Performance Logs, Appian Engine Configuration

---

QUESTION 10

Your Appian project just went live with the following environment setup; DEV > TEST (SIT/DAT) > PROD

Your client is considering adding a support team to manage production defects and minor enhancements, white the original development team focuses on Phase 2 Your client is asking you for a new environment strategy that will have the least impact on Phase 2 development work.

Which option involves the lowest additional server cost and the least code retrofit effort?

A. Phase 2 development work steam: DEV > TEST (SIT) > STAGE (UAT) > PROO Production support work stream DEV > TEST2 (SIT/UAT)>PROO

B. Phase 2 development work Stream: DEV > TEST (SIT) > STAGE (UAT) > PROO Production support work stream DEV2 > STAGE (S1T/UAT) > PROD

C. Phase 2 development work stream: DEV > TEST (SIT/UAT) >PROD Production support work stream DEV > TEST2 (SIT/UAT) > PROO

D. Phase 2 development work stream: OEV > TEST (Srr/DAT) > PROO Production support work stream. DEV2 > TEST (SIT/UAT) > PROD

Correct Answer: B

The option B involves the lowest additional server cost and the least code retrofit effort, as it only requires one additional environment (DEV2) for the production support work stream. The production support work stream can use the existing STAGE environment for testing and user acceptance testing, as it is shared with the phase 2 development work stream. This way, there is no need to create a separate TEST2 environment or to retrofit any code from TEST to STAGE or from STAGE to PROD. Verified References: [Appian Certified Lead Developer study guide], page 16, section "Environment Strategy".

QUESTION 11

On the latest Health Check report from your Cloud TEST environment utilizing a ManaDB add-on. you note the following findings

Category; User Experience Description; # of slow query rules Risk; High

Category; User Experience

Description: U of slow write to data store nodes

Risk: High

Which three things might you do to address this, without consulting the business?

A. Reduce the batch size for database queues to 10.

B. Optimize the database execution use standard database performance troubleshooting methods and tools (such as query execution plans)

C. Reduce the size and complexity of the inputs. If you ore passing in a list, consider whether (he data model can be redesigned lo pass single values instead

D. Optimize the database execution. Replace the new with a materialized view.

E. Use smaller CDTs or limit the fields selected in alqueryEntity()

Correct Answer: BCE

The three things that might help to address the findings of the Health Check report are:

B. Optimize the database execution using standard database performance troubleshooting methods and tools (such as

query execution plans). This can help to identify and eliminate any bottlenecks or inefficiencies in the database queries that are causing slow query rules or slow write to data store nodes. C. Reduce the size and complexity of the inputs. If you are passing in a list, consider whether the data model can be redesigned to pass single values instead. This can help to reduce the amount of data that needs to be transferred or processed by the database, which can improve the performance and speed of the queries or writes.

E. Use smaller CDTs or limit the fields selected in a!queryEntity(). This can help to reduce the amount of data that is returned by the queries, which can improve the performance and speed of the rules that use them. The other options are incorrect for the following reasons:

A. Reduce the batch size for database queues to 10. This might not help to address the findings, as reducing the batch size could increase the number of transactions and overhead for the database, which could worsen the performance and speed of the queries or writes.

D. Optimize the database execution. Replace the new with a materialized view. This might not help to address the findings, as replacing a view with a materialized view could increase the storage space and maintenance cost for the database, which could affect the performance and speed of the queries or writes. Verified References: Appian Documentation, section "Performance Tuning".

QUESTION 12

What are two advantages of having High Availability (HA) for Applan Cloud applications?

A. An Applan Cloud HA instance is composed of multiple active nodes running in different availability zones in differentregions.

B. Data andtransactions are continuouslyreplicated across the active nodes to achieve redundancy and avoid single points offailure.

C. A typical Appian Cloud HA instance is composed of two active nodes.

D. In the event of a system failure, your Appian instance will fie restored and available to your users in less than 15 minutes.having lost no more than the last 1minute worth of data.

Correct Answer: BD

The two advantages of having High Availability (HA) for Appian Cloud applications are:

B. Data and transactions are continuously replicated across the active nodes to achieve redundancy and avoid single points of failure. This is an advantage of having HA, as it ensures that there is always a backup copy of data and transactions in case one of the nodes fails or becomes unavailable. This also improves data integrity and consistency across the nodes, as any changes made to one node are automatically propagated to the other node. D. In the event of a system failure, your Appian instance will be restored and available to your users in less than 15 minutes, having lost no more than the last 1 minute worth of data. This is an advantage of having HA, as it guarantees a high level of service availability and reliability for your Appian instance. If one of the nodes fails or becomes unavailable, the other node will take over and continue to serve requests without any noticeable downtime or data loss for your users. The other options are incorrect for the following reasons:

A. An Appian Cloud HA instance is composed of multiple active nodes running in different availability zones in different regions. This is not an advantage of having HA, but rather a description of how HA works in Appian Cloud. An Appian Cloud HA instanceconsists of two active nodes running in different availability zones within the same region, not different regions.

C. A typical Appian Cloud HA instance is composed of two active nodes. This is not an advantage of having HA, but rather a description of how HA works in Appian Cloud. A typical Appian Cloud HA instance consists of two active nodes

running in different availability zones within the same region, but this does not necessarily provide any benefit over having one active node. Verified References: Appian Documentation, section "High Availability".