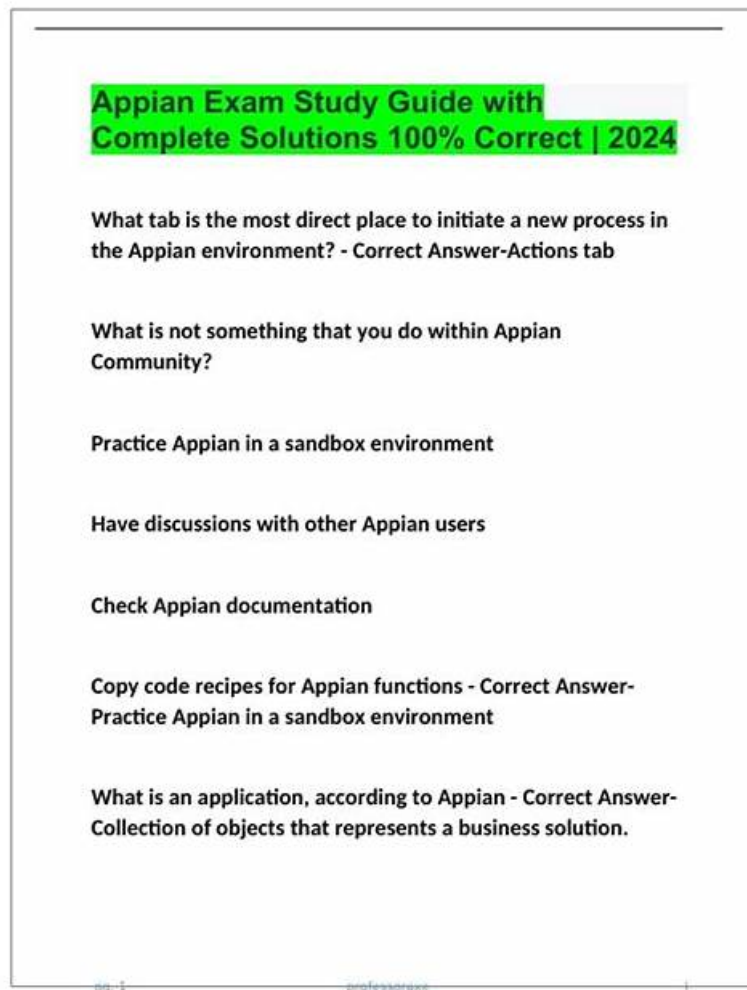


100% Pass Quiz Appian - ACD301–High Pass-Rate Valid Braindumps Files



P.S. Free 2026 Appian ACD301 dumps are available on Google Drive shared by Pass4Test: https://drive.google.com/open?id=1Nx_ZFaYbwUUYT9dkr8OuhbYbgwuf6-s

Of course, when we review a qualifying exam, we can't be closed-door. We should pay attention to the new policies and information related to the test ACD301 certification. For the convenience of the users, the ACD301 test materials will be updated on the homepage and timely update the information related to the qualification examination. As a result, the ACD301 Test Prep can help users to spend the least time, know the test information directly, let users save time and used their time in learning the new hot spot concerning about the knowledge content.

With the improvement of people's living standards, there are more and more highly educated people. To defeat other people in the more and more fierce competition, one must demonstrate his extraordinary strength. Today, getting ACD301 certification has become a trend, and ACD301 exam dump is the best weapon to help you pass certification. We all know that obtaining the ACD301 certification is very difficult, and students who want to pass the exam often have to spend a lot of time and energy. After years of hard work, the experts finally developed a set of perfect learning materials ACD301 practice materials that would allow the students to pass the exam easily. With our study materials, you only need 20-30 hours of study to successfully pass the exam and reach the peak of your career. What are you waiting for? Come and buy it now.

>> Valid Braindumps ACD301 Files <<

Quiz Appian - ACD301 –Latest Valid Braindumps Files

Appian certification is very helpful, especially the ACD301 which is recognized as a valid qualification in this industry. So far, ACD301 free download pdf has been the popular study material many candidates prefer. ACD301 questions & answers can assist you to make a detail study plan with the comprehensive and detail knowledge. Besides, we have money refund policy to ensure your interest in case of your failure in ACD301 Actual Test. Additional, if you have any needs and questions about the Appian test dump, our 24/7 will always be here to answer you.

Appian ACD301 Exam Syllabus Topics:

Topic	Details
Topic 1	<ul style="list-style-type: none">Project and Resource Management: This section of the exam measures skills of Agile Project Leads and covers interpreting business requirements, recommending design options, and leading Agile teams through technical delivery. It also involves governance, and process standardization.
Topic 2	<ul style="list-style-type: none">Application Design and Development: This section of the exam measures skills of Lead Appian Developers and covers the design and development of applications that meet user needs using Appian functionality. It includes designing for consistency, reusability, and collaboration across teams. Emphasis is placed on applying best practices for building multiple, scalable applications in complex environments.
Topic 3	<ul style="list-style-type: none">Extending Appian: This section of the exam measures skills of Integration Specialists and covers building and troubleshooting advanced integrations using connected systems and APIs. Candidates are expected to work with authentication, evaluate plug-ins, develop custom solutions when needed, and utilize document generation options to extend the platform's capabilities.
Topic 4	<ul style="list-style-type: none">Platform Management: This section of the exam measures skills of Appian System Administrators and covers the ability to manage platform operations such as deploying applications across environments, troubleshooting platform-level issues, configuring environment settings, and understanding platform architecture. Candidates are also expected to know when to involve Appian Support and how to adjust admin console configurations to maintain stability and performance.
Topic 5	<ul style="list-style-type: none">Proactively Design for Scalability and Performance: This section of the exam measures skills of Application Performance Engineers and covers building scalable applications and optimizing Appian components for performance. It includes planning load testing, diagnosing performance issues at the application level, and designing systems that can grow efficiently without sacrificing reliability.

Appian Lead Developer Sample Questions (Q46-Q51):

NEW QUESTION # 46

Your Appian project just went live with the following environment setup: DEV > TEST (SIT/UAT) > PROD.

Your client is considering adding a support team to manage production defects and minor enhancements, while 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 stream: DEV > TEST (SIT/UAT) > PROD Production support work stream: DEV > TEST2 (SIT/UAT) > PROD
- B. Phase 2 development work stream: DEV > TEST (SIT) > STAGE (UAT) > PROD Production support work stream: DEV2 > STAGE (SIT/UAT) > PROD
- C. Phase 2 development work stream: DEV > TEST (SIT) > STAGE (UAT) > PROD Production support work stream: DEV > TEST2 (SIT/UAT) > PROD
- D. Phase 2 development work stream: DEV > TEST (SIT/UAT) > PROD Production support work stream: DEV2 > TEST (SIT/UAT) > PROD

Answer: A

Explanation:

Comprehensive and Detailed In-Depth Explanation: The goal is to design an environment strategy that minimizes additional server costs and code retrofit effort while allowing the support team to manage production defects and minor enhancements without

disrupting the Phase 2 development team. The current setup (DEV > TEST (SIT/UAT) > PROD) uses a single development and testing pipeline, and the client wants to segregate support activities from Phase 2 development. Appian's Environment Management Best Practices emphasize scalability, cost efficiency, and minimal refactoring when adjusting environments.

* Option C (Phase 2 development work stream: DEV > TEST (SIT/UAT) > PROD; Production support work stream: DEV > TEST2 (SIT/UAT) > PROD): This option is the most cost-effective and requires the least code retrofit effort. It leverages the existing DEV environment for both teams but introduces a separate TEST2 environment for the support team's SIT/UAT activities. Since DEV is already shared, no new development server is needed, minimizing server costs. The existing code in DEV and TEST can be reused for TEST2 by exporting and importing packages, with minimal adjustments (e.g., updating environment-specific configurations). The Phase 2 team continues using the original TEST environment, avoiding disruption. Appian supports multiple test environments branching from a single DEV, and the PROD environment remains shared, aligning with the client's goal of low impact on Phase 2. The support team can handle defects and enhancements in TEST2 without interfering with development workflows.

* Option A (Phase 2 development work stream: DEV > TEST (SIT) > STAGE (UAT) > PROD; Production support work stream: DEV > TEST2 (SIT/UAT) > PROD): This introduces a STAGE environment for UAT in the Phase 2 stream, adding complexity and potentially requiring code updates to accommodate the new environment (e.g., adjusting deployment scripts). It also requires a new TEST2 server, increasing costs compared to Option C, where TEST2 reuses existing infrastructure.

* Option B (Phase 2 development work stream: DEV > TEST (SIT) > STAGE (UAT) > PROD; Production support work stream: DEV2 > STAGE (SIT/UAT) > PROD): This option adds both a DEV2 server for the support team and a STAGE environment, significantly increasing server costs. It also requires refactoring code to support two development environments (DEV and DEV2), including duplicating or synchronizing objects, which is more effort than reusing a single DEV.

* Option D (Phase 2 development work stream: DEV > TEST (SIT/UAT) > PROD; Production support work stream: DEV2 > TEST (SIT/UAT) > PROD): This introduces a DEV2 server for the support team, adding server costs. Sharing the TEST environment between teams could lead to conflicts (e.g., overwriting test data), potentially disrupting Phase 2 development. Code retrofit effort is higher due to managing two DEV environments and ensuring TEST compatibility.

Cost and Retrofit Analysis:

* Server Cost: Option C avoids new DEV or STAGE servers, using only an additional TEST2, which can often be provisioned on existing hardware or cloud resources with minimal cost. Options A, B, and D require additional servers (TEST2, DEV2, or STAGE), increasing expenses.

* Code Retrofit: Option C minimizes changes by reusing DEV and PROD, with TEST2 as a simple extension. Options A and B require updates for STAGE, and B and D involve managing multiple DEV environments, necessitating more significant refactoring. Appian's recommendation for environment strategies in such scenarios is to maximize reuse of existing infrastructure and avoid unnecessary environment proliferation, making Option C the optimal choice.

References: Appian Documentation - Environment Management and Deployment, Appian Lead Developer Training - Environment Strategy and Cost Optimization.

NEW QUESTION # 47

You need to connect Appian with LinkedIn to retrieve personal information about the users in your application. This information is considered private, and users should allow Appian to retrieve their information. Which authentication method would you recommend to fulfill this request?

- A. OAuth 2.0: Authorization Code Grant
- B. API Key Authentication
- C. Basic Authentication with dedicated account's login information
- D. Basic Authentication with user's login information

Answer: A

NEW QUESTION # 48

You are on a call with a new client, and their program lead is concerned about how their legacy systems will integrate with Appian. The lead wants to know what authentication methods are supported by Appian. Which three authentication methods are supported?

- A. API Keys
- B. SAML
- C. Biometrics
- D. Active Directory
- E. CAC
- F. OAuth

Answer: B,D,F

Explanation:

Comprehensive and Detailed In-Depth Explanation: As an Appian Lead Developer, addressing a client's concerns about integrating legacy systems with Appian requires accurately identifying supported authentication methods for system-to-system communication or user access. The question focuses on Appian's integration capabilities, likely for both user authentication (e.g., SSO) and API authentication, as legacy system integration often involves both. Appian's documentation outlines supported methods in its Connected Systems and security configurations. Let's evaluate each option:

* A. API Keys: API Key authentication involves a static key sent in requests (e.g., via headers). Appian supports this for outbound integrations in Connected Systems (e.g., HTTP Authentication with an API key), allowing legacy systems to authenticate Appian calls. However, it's not a user authentication method for Appian's platform login—it's for system-to-system integration. While supported, it's less common for legacy system SSO or enterprise use cases compared to other options, making it a lower-priority choice here.

* B. Biometrics: Biometrics (e.g., fingerprint, facial recognition) isn't natively supported by Appian for platform authentication or integration. Appian relies on standard enterprise methods (e.g., username/password, SSO), and biometric authentication would require external identity providers or custom clients, not Appian itself. Documentation confirms no direct biometric support, ruling this out as an Appian-supported method.

* C. SAML: Security Assertion Markup Language (SAML) is fully supported by Appian for user authentication via Single Sign-On (SSO). Appian integrates with SAML 2.0 identity providers (e.g., Okta, PingFederate), allowing users to log in using credentials from legacy systems that support SAML-based SSO. This is a key enterprise method, widely used for integrating with existing identity management systems, and explicitly listed in Appian's security configuration options—making it a top choice.

* D. CAC: Common Access Card (CAC) authentication, often used in government contexts with smart cards, isn't natively supported by Appian as a standalone method. While Appian can integrate with CAC via SAML or PKI (Public Key Infrastructure) through an identity provider, it's not a direct Appian authentication option. Documentation mentions smart card support indirectly via SSO configurations, but CAC itself isn't explicitly listed, making it less definitive than other methods.

* E. OAuth: OAuth (specifically OAuth 2.0) is supported by Appian for both outbound integrations (e.g., Authorization Code Grant, Client Credentials) and inbound API authentication (e.g., securing Appian Web APIs). For legacy system integration, Appian can use OAuth to authenticate with APIs (e.g., Google, Salesforce) or allow legacy systems to call Appian services securely. Appian's Connected System framework includes OAuth configuration, making it a versatile, standards-based method highly relevant to the client's needs.

* F. Active Directory: Active Directory (AD) integration via LDAP (Lightweight Directory Access Protocol) is supported for user authentication in Appian. It allows synchronization of users and groups from AD, enabling SSO or direct login with AD credentials. For legacy systems using AD as an identity store, this is a seamless integration method. Appian's documentation confirms LDAP/AD as a core authentication option, widely adopted in enterprise environments—making it a strong fit.

Conclusion: The three supported authentication methods are C (SAML), E (OAuth), and F (Active Directory).

These align with Appian's enterprise-grade capabilities for legacy system integration: SAML for SSO, OAuth for API security, and AD for user management. API Keys (A) are supported but less prominent for user authentication, CAC (D) is indirect, and Biometrics (B) isn't supported natively. This selection reassures the client of Appian's flexibility with common legacy authentication standards.

References:

* Appian Documentation: "Authentication for Connected Systems" (OAuth, API Keys).

* Appian Documentation: "Configuring Authentication" (SAML, LDAP/Active Directory).

* Appian Lead Developer Certification: Integration Module (Authentication Methods).

NEW QUESTION # 49

You are on a project with an application that has been deployed to Production and is live with users. The client wishes to increase the number of active users.

You need to conduct load testing to ensure Production can handle the increased usage. Review the specs for four environments in the following image.

Which environment should you use for load testing?

- A. acmetest
- **B. acmeuat**
- C. acmedev
- D. acme

Answer: B

Explanation:

The image provides the specifications for four environments in the Appian Cloud:

acmedev.appiancloud.com (acmedev): Non-production, Disk: 30 GB, Memory: 16 GB, vCPUs: 2
acmetest.appiancloud.com (acmetest): Non-production, Disk: 75 GB, Memory: 32 GB, vCPUs: 4
acmeuat.appiancloud.com (acmeuat): Non-production,

Disk: 75 GB, Memory: 64 GB, vCPUs: 8 acme.appiancloud.com (acme): Production, Disk: 75 GB, Memory: 32 GB, vCPUs: 4
Load testing assesses an application's performance under increased user load to ensure scalability and stability. Appian's Performance Testing Guidelines emphasize using an environment that mirrors Production as closely as possible to obtain accurate results, while avoiding direct impact on live systems.

Option A (acmeuat):

This is the best choice. The UAT (User Acceptance Testing) environment (acmeuat) has the highest resources (64 GB memory, 8 vCPUs) among the non-production environments, closely aligning with Production's capabilities (32 GB memory, 4 vCPUs) but with greater capacity to handle simulated loads. UAT environments are designed to validate the application with real-world usage scenarios, making them ideal for load testing. The higher resources also allow testing beyond current Production limits to predict future scalability, meeting the client's goal of increasing active users without risking live data.

Option B (acmedev):

The development environment (acmedev) has the lowest resources (16 GB memory, 2 vCPUs), which is insufficient for load testing. It's optimized for development, not performance simulation, and results would not reflect Production behavior accurately.

Option C (acme):

The Production environment (acme) is live with users, and load testing here would disrupt service, violate Appian's Production Safety Guidelines, and risk data integrity. It should never be used for testing.

Option D (acmetest):

The test environment (acmetest) has moderate resources (32 GB memory, 4 vCPUs), matching Production's memory and vCPUs. However, it's typically used for SIT (System Integration Testing) and has less capacity than acmeuat. While viable, it's less ideal than acmeuat for simulating higher user loads due to its resource constraints.

Appian recommends using a UAT environment for load testing when it closely mirrors Production and can handle simulated traffic, making acmeuat the optimal choice given its superior resources and non-production status.

NEW QUESTION # 50

You add an index on the searched field of a MySQL table with many rows (>100k). The field would benefit greatly from the index in which three scenarios?

- A. The field contains big integers, above and below 0.
- B. The field contains long unstructured text such as a hash.
- C. The field contains a textual short business code.
- D. The field contains a structured JSON.
- E. The field contains many datetimes, covering a large range.

Answer: A,C,E

Explanation:

Comprehensive and Detailed In-Depth Explanation: Adding an index to a searched field in a MySQL table with over 100,000 rows improves query performance by reducing the number of rows scanned during searches, joins, or filters. The benefit of an index depends on the field's data type, cardinality (uniqueness), and query patterns. MySQL indexing best practices, as aligned with Appian's Database Optimization Guidelines, highlight scenarios where indices are most effective.

* Option A (The field contains a textual short business code): This benefits greatly from an index. A short business code (e.g., a 5-10 character identifier like "CUST123") typically has high cardinality (many unique values) and is often used in WHERE clauses or joins. An index on this field speeds up exact-match queries (e.g., WHERE business_code = 'CUST123'), which are common in Appian applications for lookups or filtering.

* Option C (The field contains many datetimes, covering a large range): This is highly beneficial.

Datetime fields with a wide range (e.g., transaction timestamps over years) are frequently queried with range conditions (e.g., WHERE datetime BETWEEN '2024-01-01' AND '2025-01-01') or sorting (e.g., ORDER BY datetime). An index on this field optimizes these operations, especially in large tables, aligning with Appian's recommendation to index time-based fields for performance.

* Option D (The field contains big integers, above and below 0): This benefits significantly. Big integers (e.g., IDs or quantities) with a broad range and high cardinality are ideal for indexing. Queries like WHERE id > 1000 or WHERE quantity < 0 leverage the index for efficient range scans or equality checks, a common pattern in Appian data store queries.

* Option B (The field contains long unstructured text such as a hash): This benefits less. Long unstructured text (e.g., a 128-character SHA hash) has high cardinality but is less efficient for indexing due to its size. MySQL indices on large text fields can slow down writes and consume significant storage, and full-text searches are better handled with specialized indices (e.g., FULLTEXT), not standard B-tree indices. Appian advises caution with indexing large text fields unless necessary.

* Option E (The field contains a structured JSON): This is minimally beneficial with a standard index.

MySQL supports JSON fields, but a regular index on the entire JSON column is inefficient for large datasets (>100k rows) due to its variable structure. Generated columns or specialized JSON indices (e.g.,

g., using JSON_EXTRACT) are required for targeted queries (e.g., WHERE JSON_EXTRACT(json_col, '\$.key') = 'value'), but

References: Appian Documentation - Database Optimization Guidelines, MySQL Documentation - Indexing Strategies, Appian Lead Developer Training - Performance Tuning.

• • • • •

ACD301 Reliable Test Testking: <https://www.pass4test.com/ACD301.html>

- What's more, part of that Pass4Test ACD301 dumps now are free: https://drive.google.com/open?id=1Nx_ZFaYbwUUYT9dkr8OuhbYbgwuf6-s