DUMPSBOSS.

Alfresco Process Services Certified Administrator

Alfresco APSCA

Version Demo

Total Demo Questions: 9

Total Premium Questions: 92

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1 - (SIMULATION)

When a new user is created, which capabilities groups is the new user assigned to by default? (Choose two.)

A. app-designer-users

B. analytics-users

C. tenant-managers

D. Superusers

E. Administrators

ANSWER: Check the explanation for the answer

Explanation:

Correct answers: A. `app-designer-users` and B. `analytics-users`.

Newly created APS users are assigned by default to these two capability groups. They are not automatically assigned elevated administrative capability groups such as `tenant-managers`, `Superusers`, or `Administrators`.

QUESTION NO: 2

Which of the following approaches are valid for listing all of the available tenants? (Choose two.)

A. Log into activiti-app as tenant manager and view tenants under identity management (IDM).

B. Select list of tenants from database table ACT_TENANTS.

C. Use Alfresco Process Services REST API to list tenants.

D. Log into activiti-admin as tenant manager and view tenants under configuration.

E. Generate an Analytics report in activiti-app for tenant configurations.

ANSWER: C D

Explanation:

Alfresco Process Services supports tenant administration through both its administrative user interface and its REST API. “Use Alfresco Process Services REST API to list tenants.” is valid because the enterprise administration API exposes tenant resources, including an endpoint for retrieving the configured tenants. Access to these administration endpoints requires a user with the appropriate tenant-management or administrator privileges, so it is intended for authorized operational and integration use rather than ordinary end users.

“Log into activiti-admin as tenant manager and view tenants under configuration.” is also valid. The Activiti Admin application provides the tenant-management interface under its configuration area, where a tenant manager can view the tenant definitions available in the APS installation. This UI is the supported administrative route for inspecting and managing tenant configuration.

These two approaches represent the supported application-level mechanisms: an authenticated administrative API for programmatic retrieval and the Admin UI for interactive administration. APS tenancy documentation describes the tenant-management role and the use of the Admin application for tenant configuration. See the Alfresco Process Services 1.7 tenant documentation and the Alfresco Process Services REST API documentation.

QUESTION NO: 3

Which records can be retained in the history tables after a process instance has completed, when history is enabled? (Choose two.)

A. A completed process instance.

B. A completed user task.

C. A deployed BPMN XML resource.

D. An Identity Management user account.

E. An active asynchronous job.

ANSWER: A B

Explanation:

`ACT_HI_PROCINST` stores historic process-instance information, including records for completed process instances. `ACT_HI_TASKINST` stores historic task-instance information, including completed user tasks and their relevant lifecycle information.

Historic data is separate from runtime data. Once a process completes, its active execution and task records are no longer required in the runtime tables, but the historical records can be retained for audit, reporting, and operational investigation. The exact level of retained information depends on the configured process-engine history level. See the Activiti User Guide and the Alfresco Process Services Administration Guide.

QUESTION NO: 4

Which of the following configuration snippets enables Active Directory authentication?

A. ldap.authentication.active-directory.enabled=true ldap.authentication.active-directory.domain=your domain

B. ldap.authentication.enabled=true ldap.authentication.active-directory.domain=your domain

C. ldap.authentication=true ldap.authentication.active-directory.domain=your domain

D. ldap.authentication.active-directory=true ldap.authentication.active-directory.domain=your domain

ANSWER: A

Explanation:

The configuration snippet `ldap.authentication.active-directory.enabled=true ldap.authentication.active-directory.domain=your domain` is correct because Alfresco Process Services uses the `ldap.authentication.active-directory.enabled` property as the explicit switch for its Active Directory authentication configuration. Setting this property to `true` activates the Active Directory-specific authentication integration. The accompanying `ldap.authentication.active-directory.domain` setting identifies the AD domain that APS should use when authenticating users; in a real deployment, `your domain` must be replaced with the organization's actual domain value.

These properties belong in the APS application configuration, typically as entries in the `activiti-app.properties` file or supplied through the corresponding deployment configuration mechanism. Additional LDAP connection, bind-account, user-query, and synchronization properties may be required for a functional production setup, depending on the directory topology and security model. However, the stated pair is the relevant snippet for enabling the Active Directory authentication mode itself. Alfresco's Process Services documentation describes Active Directory authentication under its LDAP authentication settings: Alfresco Process Services authentication documentation. General APS administration resources are also available from the Alfresco Process Services 1.9 documentation.

QUESTION NO: 5

Which configuration option stores the query used to test database connectivity?

A. database.preferred-test-query

B. datasource.preferred-test-query

C. datasource.test.query

D. datasource.keep.alive.query

ANSWER: B

Explanation:

`datasource.preferred-test-query` is the Alfresco Process Services datasource property that contains the SQL validation query used to verify that a database connection remains usable. APS uses a pooled datasource, so connections can exist in the pool for some time before an application request needs them. A preferred test query lets the connection pool execute a lightweight, database-specific SQL statement, such as `SELECT 1`, to validate connectivity before supplying a connection to the application or while checking idle connections, depending on the associated pool-validation settings.

This setting is particularly useful where firewalls, database timeouts, or network devices can silently close otherwise idle connections. Configuring an appropriate query helps APS detect and discard stale connections, reducing database-related failures when a process operation needs persistence access. The query must be valid for the database platform in use and should be inexpensive to execute. Alfresco's Process Services database configuration reference identifies `datasource.preferred-test-query` as the setting for this purpose. See the Alfresco Process Services database configuration documentation and the Apache Tomcat JDBC pool documentation for the underlying validation-query behavior.

QUESTION NO: 6

What library is used for the Generate Document Task?

A. Aspose.

B. Jquery.

C. Angular.

D. Drools.

ANSWER: A

Explanation:

The correct answer is **Aspose.** In Alfresco Process Services (formerly Activiti Enterprise), the Generate Document Task creates a document from a Microsoft Word template and process data. This capability is implemented using the Aspose document-processing library, specifically its Word-processing functionality. The task can merge values from the process context into placeholders in a template, then produce a generated document that can be stored or used in subsequent process steps. Aspose is designed for server-side creation and manipulation of Office-format documents without requiring a desktop Office installation, which makes it suitable for a workflow engine running on application servers. Alfresco's Activiti development documentation describes the Generate Document Task as part of its document-generation support, while Aspose's product documentation describes the underlying Java library's ability to create, populate, convert, and save Word documents programmatically. See the Alfresco Activiti Developer Guide and the Aspose.Words for Java documentation.

QUESTION NO: 7

Alfresco Process Services uses the 3rd party library Aspose to transform documents. Which statements are correct regarding Aspose? (Choose two.)

A. When activiti-app is installed with installer, no manual action is needed to deploy an Aspose license to fully function.

B. When activiti-app is installed using the war distributions, an Aspose license named "transform.lic" has to be deployed manually.

C. Regardless of how activiti-app is installed, Aspose does not require a license to be deployed in order to fully function.

D. When activiti-app is installed using the war distributions, no manual action is needed to deploy an Aspose license to fully function.

E. Regardless of how activiti-app is installed, the path to the required Aspose license can be configured in the activiti-admin.properties file.

ANSWER: B E

Explanation:

Alfresco Process Services uses Aspose for document transformation capabilities, and a valid Aspose license is required for the complete supported transformation functionality. In a WAR-based installation, the application does not receive the installer-managed license deployment; therefore, the administrator must manually deploy the license file named `transform.lic`. This makes the statement, “When activiti-app is installed using the war distributions, an Aspose license named “transform.lic” has to be deployed manually,” correct. The license location is also configurable, which supports environments where the license must reside outside the deployed application or be supplied through an environment-specific configuration. The relevant setting is maintained in `activiti-admin.properties`, so the statement that the path to the required Aspose license can be configured there regardless of installation method is also correct. This configuration approach allows administrators to manage the license consistently while retaining control over deployment-specific file locations. Consult the Alfresco Process Services administration documentation for deployment and configuration guidance: Alfresco Process Services Administration. Product documentation and supported-version resources are also available through the Alfresco Process Services documentation portal.

QUESTION NO: 8

By default, how often does a node in a cluster send information from the engine to that admin-app?

A. Every 10 seconds.

B. Every 15 seconds.

C. Every 20 seconds.

D. Every 30 seconds.

ANSWER: D

Explanation:

Every 30 seconds. is correct. In an Alfresco Process Services clustered deployment, each Process Services engine node periodically reports its status and runtime information to the Admin Application. This heartbeat lets the Admin Application maintain an up-to-date view of the nodes participating in the cluster, including their availability and engine-related state. The default reporting interval is 30 seconds, so an administrator can expect a normally functioning node to communicate with the Admin Application twice per minute. This regular node-to-admin communication supports cluster monitoring and helps the administrative interface identify nodes that are active or no longer reporting. The interval is a default configuration value and may be adjusted where the deployment’s operational requirements call for different monitoring responsiveness or network behavior. See the Alfresco Process Services administration documentation on clustering at Alfresco Process Services clustering and the Process Services documentation archive at Alfresco Process Services documentation.

QUESTION NO: 9

Which file extensions are supported for process models imported using App Designer? (Choose two.)

A. .bpm

B. .bpml

C. .bpmn

D. .bpmn20

E. .bpmn20.xml

ANSWER: C E

Explanation:

App Designer supports importing BPMN 2.0 process definitions when the source file uses the `.bpmn` or `.bpmn20.xml` extension. These are recognized filename conventions for BPMN 2.0 XML interchange files, enabling a process model created or exported by a compatible BPMN tool to be brought into Alfresco Process Services for further editing and use in an app. The imported definition represents the process structure in BPMN XML, including elements such as start events, user tasks, gateways, sequence flows, and end events. After import, the model can be opened in the Process Editor and configured with the forms, assignments, and implementation details required by the application.

The `.bpmn` extension is the common short extension used for BPMN 2.0 files. The `.bpmn20.xml` extension is also explicitly supported because it is frequently used by BPMN and Activiti-compatible tooling to identify BPMN 2.0 XML content. Alfresco Process Services documentation describes App Designer and its process-model capabilities at Alfresco Process Services documentation; the BPMN specification and standard XML interchange context are maintained by OMG BPMN 2.0.