

DUMPSBOSS.

Jira Administration for Cloud

Atlassian ACP-120

Version Demo

Total Demo Questions: 7

Total Premium Questions: 74

Buy Premium PDF

<https://dumpsboss.co>

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, General Configuration	10
Topic 2, User Management	10
Topic 3, Project Configuration	15
Topic 4, Issue Configuration	24
Topic 5, Workflow Configuration	12
Topic 6, Mix Questions	3
Total	74

QUESTION NO: 1

A company-managed project uses components for product areas. Each new issue has exactly one component, and it must be assigned automatically to that component's lead when no assignee is selected during creation. Which two configurations are required? (Choose two.)

- A. Set a component lead for each relevant component
- B. Set the project Default Assignee to Component Lead
- C. Set the project Default Assignee to Project lead
- D. Grant every component lead the Administer Projects permission
- E. Add the Component field to an issue security scheme

ANSWER: A B

Explanation:

Each relevant **component must have a component lead** so Jira has a user to assign for issues in that component. The project's **Default Assignee** must then be set to Component Lead. Together, these settings assign an otherwise unassigned new issue to the lead of its selected component.

Setting the default assignee to Project lead would assign all such issues to one project-level user instead. Assign Issues permission controls whether users can assign issues, but it does not configure the automatic default-assignment behavior.

QUESTION NO: 2

What is the impact of changing the name of a project?

- A. Jira will need to be reindexed.
- B. The URL for issues will change.
- C. The old name can no longer be reused.
- D. The project key will also change.
- E. Issue filters may need to be updated.

ANSWER: E

Explanation:

Issue filters may need to be updated. Changing a Jira Cloud project's name changes its human-readable display name but does not change the project's underlying identity. A saved filter, dashboard gadget, board query, or other JQL-based configuration can explicitly contain the project name, such as `project = "Old Project Name"`. After the rename, that query text still refers to the former name, so an administrator or filter owner should review and update it to use the new name where necessary.

Using the project key in JQL, for example `project = ABC`, is the more durable approach because the key is independent of a routine name change. This helps keep saved searches and reporting configurations stable when a project is renamed. Jira's JQL documentation confirms that the `project` field can be searched using a project name or key, while Atlassian's project administration guidance distinguishes changing project details, including its name, from changing its key. See the [JQL fields reference](#) and [Edit a project's details](#).

QUESTION NO: 3

A company-managed project has separate Create, Edit, and View screens. Users must be able to see the Root Cause custom field after an issue is closed, but they must not enter or change it during issue creation or editing. Which configuration meets this requirement?

- A. Add Root Cause only to the View screen.
- B. Add Root Cause only to the Edit screen.
- C. Mark Root Cause as required in the field configuration.
- D. Add Root Cause to every screen and hide it in the issue layout.

ANSWER: A

Explanation:

Add Root Cause only to the View screen is correct. In a company-managed project, the screen scheme can use different screens for Create, Edit, and View operations. Adding the field only to the View screen makes its stored value visible on the issue while preventing users from supplying or changing it through the standard Create and Edit operations.

A field configuration cannot make a field view-only for one operation while leaving it visible for another. Adding the field to the Edit screen would allow users to change it, and adding it to the Create screen would allow users to set it when creating an issue.

QUESTION NO: 4

You were asked to modify the only workflow in a company-managed project. Which two requirements can be satisfied using only out-of-box functionality? (Choose two.)

- A. Automatically set Due Date based on the issue type.
- B. Only a subset of project users should be able to create epics.
- C. Automatically clear Assignee when transitioning to Pending.
- D. Automatically reassign an issue when Priority is edited.
- E. Only members of Testers project role can transition to Passed.

ANSWER: C E

Explanation:

“Automatically clear Assignee when transitioning to Pending.” can be implemented directly on the transition that moves an issue to Pending. Jira Cloud’s built-in workflow post functions run after a transition occurs, and the *Clear field value* post function can clear the Assignee field as part of that transition. Because this behavior is configured in the workflow itself, it applies consistently whenever users take that transition and does not require an app, custom script, or separate rule.

“Only members of Testers project role can transition to Passed.” is also supported natively through a workflow transition condition. A condition controls whether a user may perform a transition. The built-in *User is in project role* condition can be configured for the Testers role on the transition leading to Passed. Project roles are project-scoped, so administrators can manage the relevant users or groups through the project’s role membership while the workflow enforces the transition restriction.

These are standard capabilities of company-managed Jira Cloud workflows: post functions perform actions after a transition, while conditions limit who can execute it. Atlassian documents both workflow configuration and transition-rule behavior in [Configure advanced issue workflows](#) and [Configure workflow transition rules](#).

QUESTION NO: 5

Users need to be able to select Browser Type when editing bugs in the DEV project. However, they cannot see the field. Identify three possible root causes. (Choose three.)

- A. Project roles
- B. Field configurations
- C. Custom field context
- D. Project permissions
- E. Issue security scheme

ANSWER: B C D

Explanation:

Field configurations, **Custom field context**, and **Project permissions** are the applicable areas to investigate. A field configuration controls whether a field is hidden, required, or optional for the issue type that uses that configuration. If Browser Type is hidden in the field configuration associated with Bug issues in DEV, Jira suppresses it from issue operations, including editing.

A custom field's context controls the projects and issue types for which the field exists. Browser Type must have a context that includes both the DEV project and the Bug issue type. Otherwise, Jira does not make the field available for those issues, regardless of its presence in broader configuration.

Project permissions are also relevant because users need the Edit Issues permission to edit issue fields. Permission grants can be assigned through users, groups, or project roles, so the effective permissions of the affected users must allow issue editing in DEV. Once the field is applicable and not hidden, users with Edit Issues permission can select a value during the edit operation. Atlassian documents field behavior in [Configure a field configuration scheme](#), custom-field applicability in [Configure custom field contexts](#), and access control in [Manage project permissions](#).

QUESTION NO: 6

Gary is no longer receiving notifications when his colleagues modify issues that he is watching. You already verified that watchers should be notified. Which personal setting did Gary update?

- A. Watch your issues
- B. You make changes to the issue
- C. Email notifications format
- D. Notifications for relevant issues

ANSWER: D

Explanation:

Notifications for relevant issues is the applicable personal setting. In Jira Cloud, a user can control whether they receive email notifications for issue activity that is relevant to them through their personal settings. Relevant activity includes updates to issues where the user has an association such as watcher, assignee, or reporter. Therefore, although the project notification scheme continues to designate watchers as recipients, disabling this personal email-notification preference prevents Gary from receiving those watcher-related emails. This setting acts at the individual user level, so it can change Gary's experience without changing the issue's watcher list, the project notification scheme, or the notification behavior for other users. The scenario specifically establishes that watcher notifications are configured correctly, which makes Gary's personal preference the determining factor. Atlassian documents that users can open their personal settings and select whether to receive email notifications for relevant issue activity; when the setting is off, they do not receive issue-activity emails even when they are associated with an issue. See Atlassian's guidance on [managing Jira notification emails](#) and its overview of [watching issues](#).

QUESTION NO: 7

You need to identify issues that meet both conditions:

- Tom set the priority value to “Highest” sometime this month.
- The priority value may or may not be “Highest” now.

Which JQL query returns the expected results?

- A. priority changed to Highest by tom before endOfMonth()
- B. priority was Highest by tom after startOfMonth()
- C. priority was Highest by tom during (startOfMonth(), endOfMonth()) and priority = Highest or priority != Highest
- D. priority was Highest by tom and (priority = Highest by tom after startOfMonth() or priority != Highest)

ANSWER: B

Explanation:

`priority was Highest by tom after startOfMonth()` is correct because it queries the issue's historical field changes rather than its current priority. The `WAS` operator searches for issues where a field held a specified value at some point in its recorded history. Therefore, an issue remains in the results even if its priority was subsequently changed from `Highest` to another value, which directly satisfies the requirement that the current value is irrelevant.

The `BY tom` predicate narrows that historical event to changes attributed to Tom. The `after startOfMonth()` predicate limits the matching history to events after the beginning of the current month. Together, these predicates identify issues for which Tom set the priority to `Highest` during the current month, without adding any condition that requires the priority to still be `Highest`.

Atlassian documents that historical JQL operators such as `WAS` can be qualified with predicates including `BY` and date predicates such as `AFTER`. See the [JQL operators reference](#) and the [JQL functions reference](#) for the behavior of `startOfMonth()`.