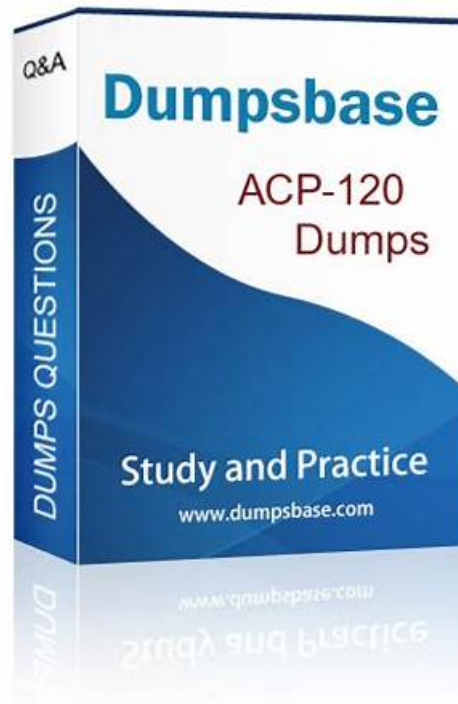


# ACP-120 Valid Exam Cost - ACP-120 Latest Dumps Free



DOWNLOAD the newest It-Tests ACP-120 PDF dumps from Cloud Storage for free: <https://drive.google.com/open?id=1U7TnxJ8hUno6ma3uSXVZe4RSb7tcMe1j>

Exam candidates are susceptible to the influence of ads, so our experts' know-how is impressive to pass the ACP-120 exam instead of making financial reward solely. We hypothesize that you fail the exam after using our ACP-120 learning engine we can switch other versions for you or give back full refund. In such a way, our ACP-120 Exam Questions can give you more choices to pass more exams and we do put our customers' interest as the first thing to consider.

It-Tests is a platform that will provide candidates with most effective ACP-120 study materials to help them pass their exam. It has been recognized by all of our customers, because it was compiled by many professional experts of our website. Not only did they pass their exam but also got a satisfactory score. These are due to the high quality of our ACP-120 study torrent that leads to such a high pass rate.

>> ACP-120 Valid Exam Cost <<

## ATLASSIAN ACP-120 Exam | ACP-120 Valid Exam Cost - ACP-120: Jira Cloud Administrator

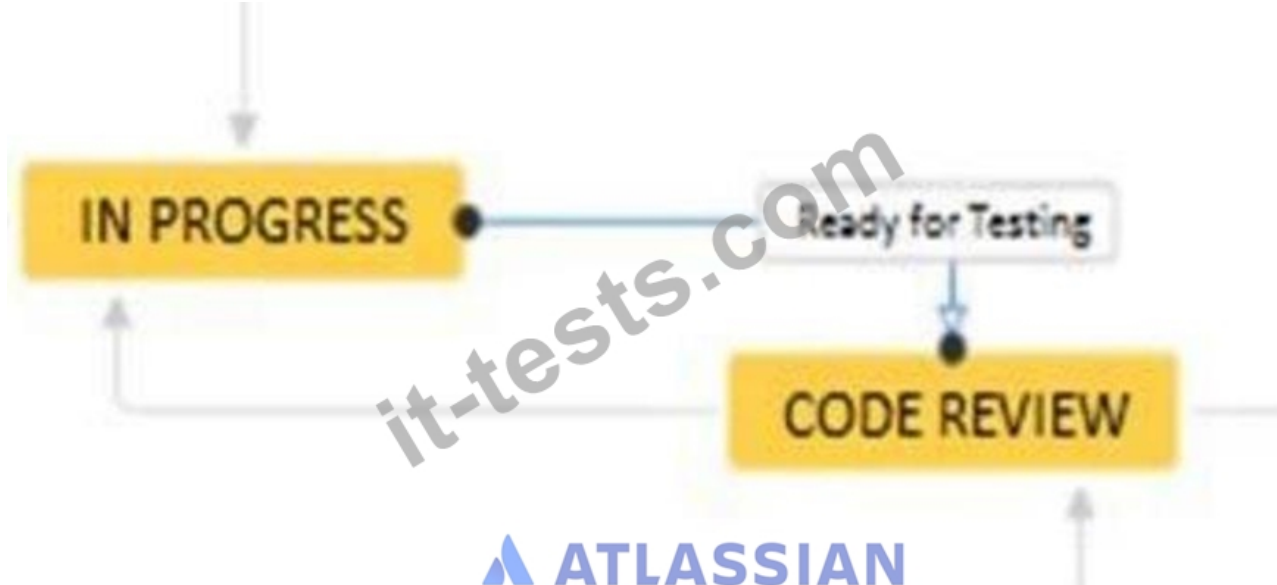
With the best quality and high accuracy, our ACP-120 vce braindumps are the best study materials for the certification exam among the dumps vendors. Our experts constantly keep the pace of the current exam requirement for ACP-120 Actual Test to ensure the accuracy of our questions. The pass rate of our ACP-120 exam dumps almost reach to 98% because our questions and answers always updated according to the latest exam information.

ATLASSIAN ACP-120 Certification Exam is a highly recognized certification in the field of Jira Cloud Administration. ACP-120 exam is designed to test the knowledge and skills of Jira Cloud administrators in managing Jira instances, configuring and customizing Jira projects, and troubleshooting common Jira issues. Jira Cloud Administrator certification is offered by ATLASSIAN, one of the leading software companies in the world that provides tools for software development and project management.

## ATLASSIAN Jira Cloud Administrator Sample Questions (Q64-Q69):

### NEW QUESTION # 64

A user says she can see the Ready for Testing transition but cannot successfully execute it.



Why can't the user execute this transition?

- A. A Condition on the transition requires the user to be in the Developer role.
- B. A Validator on the workflow transition requires the user to be in the Developer role.
- C. The status property of Code Review prevents the Developer role from executing the transition.
- **D. The status property of In Progress prevents the Developer role from executing the transition.**

**Answer: D**

### NEW QUESTION # 65

You need to create a filter that shows only issues that were created during the last calendar month. Identify the correct JQL query.

- **A. `createdDate >= startOfMonth(-1m) AND createdDate <= endOfMonth(-1m)`**
- B. `createdDate >= startOfMonth(-1) OR createdDate <= endOfMonth(-1)`
- C. `created changed during (startOfMonth(), endOfMonth())`
- D. `created >= startOfMonth(-1) AND created <= endOfMonth(-1)`

**Answer: A**

Explanation:

To create a filter that shows issues created during the last calendar month, the JQL query must use the `createdDate` field with the `startOfMonth(-1m)` and `endOfMonth(-1m)` functions to define the time range. The correct query is `createdDate >= startOfMonth(-1m) AND createdDate <= endOfMonth(-1m)` (Option A).

\* Explanation of the Correct Answer (Option A):

\* The `createdDate` field represents the date an issue was created. The `startOfMonth(-1m)` function returns the first day of the previous month (e.g., April 1, 2025, for May 2025), and `endOfMonth(-1m)` returns the last day of the previous month (e.g., April 30, 2025). The query `createdDate >= startOfMonth(-1m) AND createdDate <= endOfMonth(-1m)` selects issues created within this range, covering the entire last calendar month.

\* Exact Extract from Documentation:

Search for issues using JQL

The `createdDate` field (or `created` alias) stores the creation date of an issue. Use date functions like `startOfMonth()` and `endOfMonth()` to define time ranges.

Examples:

\* `createdDate >= startOfMonth(-1m) AND createdDate <= endOfMonth(-1m)` returns issues created in the previous calendar month.

\* Use `-1m` to indicate one month ago. Note: Ensure date fields use `>=` and `<=` for inclusive ranges.

(Source: Atlassian Support Documentation, "Search for issues using JQL")

\* Why This Fits: The query in Option A correctly uses `createdDate`, `startOfMonth(-1m)`, and `endOfMonth(-1m)` with `AND` to define the last calendar month, making it the correct answer.

\* Why Other Options Are Incorrect:

- \* created changed during (startOfMonth(), endOfMonth()) (Option A):
- \* Thechanged duringoperator is used for fields that track changes (e.g., status, assignee), not for static fields likecreated. Additionally,startOfMonth()andendOfMonth()without- 1 mrefer to the current month, not the last month.
- \* Extract from Documentation:  
Thechanged duringoperator is invalid for thecreatedfield. Use>=and<=for date comparisons.  
(Source: Atlassian Support Documentation, "Advanced searching - operators reference")
- \* createdDate >= startOfMonth(-1) OR createdDate <= endOfMonth(-1) (Option B):
- \* TheORoperator creates a logical error, selecting issues created after the start of the previous monthorbefore the end of the previous month, which includes almost all issues.  
Additionally,-1 is ambiguous and typically invalid without a unit (e.g.,-1 mfor one month).  
The correct operator isANDto define a range.
- \* Extract from Documentation:  
UseANDto define a date range (e.g., createdDate >= startOfMonth(-1m) AND createdDate <= endOfMonth(-1 m)).ORcreates an overly broad range.  
(Source: Atlassian Support Documentation, "Advanced searching - operators reference")
- \* created >= startOfMonth(-1) AND created <= endOfMonth(-1) (Option C):
- \* The-1 syntax without a unit (e.g.,-1m) is ambiguous and typically invalid inJQL. The correct syntax uses-1mto indicate one month ago. Additionally, whilecreatedis an alias for createdDate, the incorrect time unit makes this query invalid.
- \* Extract from Documentation:  
Date functions require units likemfor months (e.g.,startOfMonth(-1m)). Without a unit, the query is invalid.  
(Source: Atlassian Support Documentation, "Advanced searching - functions reference")
- \* Additional Notes:
- \* The filter can be created inIssues > Search for issues, saved as a filter, and shared if needed.
- \* Thecreatedalias can be used instead ofcreatedDate, butcreatedDateis more explicit and matches the options.
- \* Ensure the Jira instance's timezone aligns with the expected month range.

:

Atlassian Support Documentation:Search for issues using JQL  
Atlassian Support Documentation:Advanced searching - operators reference Atlassian Support Documentation:Advanced searching - functions reference

## NEW QUESTION # 66

You performed the following steps:

- \* Created a new filter with the JQL query project IN projectsWhereUserHasRole("Sales")
  - \* Created a filter subscription for the filter and set the recipients to the Sales group.
- Leila is a member of the Sales group. Which issues will Leila receive in her subscription?

- A. Issues from projects where Leila is in the Sales role
- B. Issues from projects where Leila is in the Sales group
- C. Issues assigned to anyone in the Sales role
- D. Issues assigned to anyone in the Sales group

**Answer: A**

Explanation:

The JQL query project IN projectsWhereUserHasRole("Sales") filters for issues in projects where the user running the query has theSalesproject role. Since Leila is a member of theSales groupand receives the filter subscription, the issues she receives are those from projects whereLeila is in the Sales role(Option A).

\* Explanation of the Correct Answer (Option A):

\* The JQL functionprojectsWhereUserHasRole("Sales")returns a list of projects where the user executing the query is a member of theSalesproject role. The query project IN projectsWhereUserHasRole("Sales") selects all issues from those projects.

\* Leila, as a member of theSales group, receives the filter subscription. When the subscription runs for Leila, the JQL query is evaluated in her context, meaning it returns issues from projects where Leilahas theSalesproject role.

\* Exact Extract from Documentation:

Advanced searching - functions reference

TheprojectsWhereUserHasRole("role")function returns projects where the user running the query is a member of the specified project role.

Example:

\* project IN projectsWhereUserHasRole("Sales") returns issues from projects where the user has theSalesproject role.Note: For filter subscriptions, the query is executed in the context of each recipient, so results depend on the recipient's role membership.

(Source: Atlassian Support Documentation, "Advanced searching - functions reference")

- \* Why This Fits: The filter subscription runs the JQL query for Leila, returning issues from projects where Leila is in the Sales role, making Option A the correct answer.
- \* Why Other Options Are Incorrect:
- \* Issues assigned to anyone in the Sales group (Option B):
- \* The JQL query does not filter by assignee or group membership. It only selects issues based on the projects where the user has the Sales role, regardless of who the issues are assigned to.
- \* Extract from Documentation:  
To filter by assignee, use assignee in membersOf("group"). The projectsWhereUserHasRole function filters by project, not assignee. (Source: Atlassian Support Documentation, "Advanced searching - fields reference")
- \* Issues from projects where Leila is in the Sales group (Option C):
- \* The query uses projectsWhereUserHasRole("Sales"), which checks for the Sales project role, not a group. Project roles and groups are distinct, and the query does not reference group membership.
- \* Extract from Documentation:  
Project roles are project-specific and distinct from groups. Use projectsWhereUserHasRole for roles, not membersOf for groups. (Source: Atlassian Support Documentation, "Advanced searching - functions reference")
- \* Issues assigned to anyone in the Sales role (Option D):
- \* The query does not filter by assignee or role membership of the assignee. It selects issues from projects where the user (Leila) has the Sales role, not issues assigned to users in that role.
- \* Extract from Documentation:  
To filter by assignees in a role, additional JQL or scripting is needed. projectsWhereUserHasRole only filters by project. (Source: Atlassian Support Documentation, "Advanced searching - fields reference")
- \* Additional Notes:
- \* The Sales project role must exist in the relevant projects, and Leila must be added to it in Project settings > People for those projects.
- \* The subscription sends emails to the Sales group, and each recipient (like Leila) sees issues based on their own role membership in the JQL query's context.
- \* Leila must have Browse Projects permission for the projects returned by the query to see the issues.

:

Atlassian Support Documentation: Advanced searching - functions reference Atlassian Support Documentation: Advanced searching - fields reference Atlassian Support Documentation: Manage filters in Jira Cloud

## NEW QUESTION # 67

Below are the requirements for two new fields to be used in a company-managed project:

- \* Allow a single option from a dropdown list of options.
- \* Allow bold and underlined text.

Identify the two field types you need to use. (Choose two.)

- A. Paragraph
- B. Short text
- C. Checkboxes
- D. Radio buttons

**Answer: A,D**

Explanation:

To meet the requirements for two new fields in a company-managed project, one field must allow selecting a single option from a dropdown list, and the other must allow bold and underlined text. The appropriate field types are Radio buttons (Option C) for the dropdown list and Paragraph (Option B) for bold and underlined text.

\* Explanation of the Correct Answers:

\* Radio buttons (Option C):

\* The requirement to allow a single option from a dropdown list of options is met by a Radio buttons custom field. This field type presents a list of predefined options in a radio button format, allowing users to select exactly one option, similar to a dropdown list. In Jira, radio buttons are often used interchangeably with Select List (single choice) for single-selection lists, and both fulfill the requirement.

\* Exact Extract from Documentation:

Custom field types

The Radio buttons field type allows users to select one option from a predefined list, presented as radio buttons.

To create:

- \* Go to Settings > Issues > Custom fields.
- \* Create a new field and select Radio buttons.

- \* Define the list of options. Note: Radio buttons are similar to Select List (single choice) and are used for single-selection dropdown-like functionality. (Source: Atlassian Support Documentation, "Manage custom fields in Jira Cloud")
- \* Why This Fits: The Radio buttons field type provides a single-selection list, meeting the requirement for a dropdown list of options.
- \* Paragraph (Option B):
- \* The requirement to allow bold and underlined text is met by a Paragraph custom field (also known as Text Field (multi-line)). This field type supports rich text input, including formatting options like bold and underlined text, using a WYSIWYG editor or wiki-style markup.
- \* Exact Extract from Documentation:

Custom field types

The Paragraph (or Text Field (multi-line)) field type allows users to enter multiple lines of text with rich text formatting, including bold, underline, and other styles.

To create:

- \* Go to Settings > Issues > Custom fields.
- \* Create a new field and select Paragraph. Note: The rich text editor supports formatting like bold and underline, or users can use wiki-style markup (e.g., bold, underline). (Source: Atlassian Support Documentation, "Manage custom fields in Jira Cloud")
- \* Why This Fits: The Paragraph field supports rich text formatting, including bold and underlined text, meeting the second requirement.
- \* Why Other Options Are Incorrect:
- \* Short text (Option A):
- \* A Short text (or Text Field (single line)) custom field allows single-line text input but does not support rich text formatting like bold or underlined text. It is unsuitable for the second requirement and does not provide a dropdown list for the first requirement.
- \* Extract from Documentation:

The Short text field type is for single-line plain text and does not support rich text formatting like bold or underline.

(Source: Atlassian Support Documentation, "Manage custom fields in Jira Cloud")

- \* Checkboxes (Option D):
- \* A Checkboxes custom field allows users to select multiple options from a predefined list.

The first requirement specifies a single option, making checkboxes unsuitable, as they are designed for multiple selections.

- \* Extract from Documentation:

The Checkboxes field type allows multiple selections from a list. For single-selection lists, use Radio buttons or Select List (single choice).

(Source: Atlassian Support Documentation, "Manage custom fields in Jira Cloud")

- \* Additional Notes:
- \* Steps to configure:
- \* Create a Radio buttons custom field in Settings > Issues > Custom fields, name it appropriately, and define the options.
- \* Create a Paragraph custom field in Settings > Issues > Custom fields for rich text input.
- \* Add both fields to the project's screens (Project settings > Screens).
- \* Configuring custom fields requires Jira administrator privileges.
- \* The Radio buttons field may display as a dropdown-like list in some interfaces, aligning with the requirement.

:

Atlassian Support Documentation: Manage custom fields in Jira Cloud

Atlassian Support Documentation: Configure screens in Jira Cloud

## NEW QUESTION # 68

No one should be allowed to edit issues in a particular workflow status of a company-managed project. Which element must be configured?

- A. Status Property
- B. Condition
- C. Trigger
- D. Post function

**Answer: A**

Explanation:

Okay, let's analyze this question again with the revised options.

The requirement is to prevent anyone from editing an issue when it resides in a specific workflow status.

As established previously:

- \* Conditions (A): Control whether a transition out of or into a status can occur. They check criteria before a transition starts. They don't prevent editing while an issue is sitting in a status.
- \* Post Functions (Implied by B & C): Actions that execute after a transition is successfully completed (e.

\* Triggers (D):Automatically initiate a workflow transition based on external events (like code commits). Not relevant to editing permissions within a status.

\* Navigate to Workflow configuration.

\* Go to its 'Properties'.

### NEW QUESTION # 69

• • • • •

**ACP-120 Latest Dumps Free:** <https://www.it-tests.com/ACP-120.html>

- DOWNLOAD the newest It-Tests ACP-120 PDF dumps from Cloud Storage for free: <https://drive.google.com/open?id=1U7TnxJ8hUno6ma3uSXVZe4RSb7tcMe1j>