

The Ultimate Guide to Jira Migrations

How to migrate from Jira Server
to Data Center or Cloud

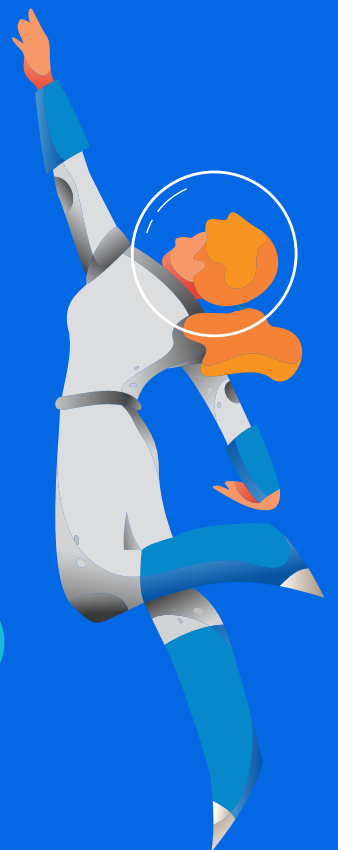
By Rachel Wright

With a foreword by George Dinkov,
Director of Product Engineering at Botron

Notice of Liability

The information in this book is distributed on an "As Is" basis, without warranty. The publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained in this book. Use of the information and instructions contained in this work is at your own risk.

© 2020 Botron Atlassian Apps, an Appfire company. All rights reserved.



Foreword

Hi, I'm George Dinkov, co-founder and Director of Product Engineering at Botron.

Since 2013, our company has focused on creating Jira apps that aid administrators with enterprise-level challenges, like system configuration, governance and scaling, backups, migrations, system integrity, and more. And through these efforts, we have become intimately familiar with the “gotchas” of Jira migration — the failure to plan and test, the lack of communication with the right stakeholders, not cleaning up before migrating to a new system, and the list goes on. We realized that a lot of administrators, and the enterprises they worked for, were facing huge obstacles when they attempted system-level changes in Jira. What the ecosystem desperately needed was a resource on Jira migrations.



Enter Rachel Wright... I personally met Rachel earlier this year, but her reputation has been well-known throughout the Atlassian ecosystem for years. Not only is Rachel a frequent speaker, user group leader, and champion within the community, she has also developed some of the best Jira training courses on LinkedIn, authored THE book on Jira strategy, and regularly offers herself as a resource to companies facing Jira administrative challenges. She has a deep understanding of Jira configurations, best practices, and how decisions today will affect the application in the future. These qualifications queue her up as the ideal author of a resource on migrations, which is why we asked her to write it.

Migrations are one of the hardest things you will have to do as a Jira admin. This book is useful because it's not just high-level guidance about migrations — it goes into the specifics about which problems are likely to be encountered, so you can plan appropriately, have the right conversations with stakeholders, think through tricky scenarios, and feel confident about each step in the migration process.

As one of the technical reviewers of this book, I can assure you that Rachel's thought of and addressed just about every scenario one could encounter when it comes to migrating off of Jira Server. There are worksheets, case studies, a quiz, and a wealth of expert commentary that will help you make the right decisions for your own organization.

This book couldn't have come at a better time, either. Although companies have been migrating and moving Jira servers for many years, Atlassian has recently

announced that their Server offering is going away. Now more than ever, it's crucial that the ecosystem comes together to share best practices, tips, and tricks for performing successful migrations.

Although you may be reading this book because of a required move, I want to assure you that regardless of the reason for moving, there is a lot of good that can come from migrating to Data Center or Cloud.

Moving your current system is no small task, so you'll want to make sure you clean up your instance, remove or archive old projects and issues, and only migrate what you really need. The entire process can help you gain operational efficiencies, reduce overhead, start fresh, bring in new features, reduce cruft and complexity, and improve your system infrastructure. Think of how good it will feel when the process is done, and you end up in a better spot than when you started.

I will leave you with this. If you're faced with the mighty challenge of a Jira migration, rest easy knowing that this book covers everything you might face in the process. Read it, plan your approach, test test test, and you will save yourself a lot of headaches. So, grab a cup of coffee and settle in — the Ultimate Guide to Jira Migrations awaits!

Sincerely,

George Dinkov
Director of Product Engineering
Botron Atlassian Apps, an Appfire Company



Contents

1.0 Introduction	7
1.1 Who is this Book for?	8
1.2 Why Migrate?.	8
1.3 Terminology	10
1.4 Conventions	11
2.0 Jira Deployment Differences	12
2.1 Server vs. Data Center vs. Cloud.	12
2.2 Quiz	16
2.3 Creating a Plan.	26
2.4 Tracking Apps and Customizations	30
3.0 Migration Options	32
3.1 Jira Server to Jira Cloud	32
3.2 Jira Server to Jira Data Center	34
3.3 Full vs. Partial Migration	36
3.4 Phased Migration	39
3.5 Consolidation	39
3.6 Starting Fresh	43
4.0 Governance, Approval, and Change Management	44
4.1 Identifying Stakeholders	44
4.2 Getting Buy-in	47
4.3 Approval	48
4.4 Tracking Changes	49

5.0 Migration Process	51
5.1 Migration Checklist	51
5.2 Migration Preparation	59
Configuration Assessment	61
Back Up	76
Clean Up	77
Archiving Projects	78
Health and Performance Tests	80
5.3 Helper Apps	84
Jira Cloud Migration Assistant	84
Jira Site Import	85
CSV Import	86
Jira Data Center Migration App	86
Configuration Manager for Jira	87
5.4 Example Migration Walkthrough using Configuration Manager for Jira	89
Step 1: Install App and Run Initial Checks	89
Step 2: Create Snapshot	93
Step 3: Deploy Snapshot	97
User and Group Migration	107
Attachments	108
Apps Migration	110
Case Study	111
5.5 Migration Execution Strategy	112
Choosing the Right Migration Partner	112
Migrating on Your Own	113
Must-have Apps	113
Additional Worksheets	114
5.6 Migration Considerations	122
Refresh Environment	122
Freeze Configuration	122

Launch	124
Pre-Migration Training	126
5.7 Merge Considerations.	127
Application Comparison	127
Conflict Resolution	128
5.8 Migration Testing	135
Test Team	135
Test Cases	136
6.0 After Migration	155
6.1 Test Environment	156
6.2 Project Templates	161
6.3 Disaster Recovery and Backup.	164
6.4 Archive Strategy	164
6.5 Post-Migration Training.	164
7.0 Q&A	166
7.1 What are the worst migration mistakes?.	166
7.2 What is the biggest migration misconception?	167
7.3 How can you manage risk during the migration process?	168
Conclusion	171
Worksheets	172
Sources and Documentation	173
Index	177
About the Author	179
Special Thanks	179

1.0 Introduction

There are many reasons organizations switch deployment types or environments over the years. Maybe your Jira Server application is mission-critical and requires the high availability and redundancy features of Jira Data Center. Maybe you're hosting Jira on-premises but crave the simplicity and scalability of a Cloud-hosted infrastructure. Maybe your organization has other Cloud tools, and it makes sense to use Atlassian's Jira Cloud product, too. Maybe you have multiple Jira applications, and you need to consolidate them into a single instance for easier management. Or maybe your current application is a bit of a mess, and you'd like to start fresh with a clean configuration. If so, this is the guide for you!

Hi, I'm Rachel Wright, Certified Jira Administrator and Author of the *Jira Strategy Admin Workbook*. As a consultant, I've seen a lot of different Jira configurations and helped plan many migrations, merges, and moves. Sometimes the effort is small, like moving a Jira project from one application to another. Other times it's porting an entire application to a different hosting environment or combining multiple large instances together. I've even advised a company to start fresh in a new application.

This guide will help you understand Jira deployment type differences, so you can determine which is best for your organization. I'll discuss the migration options, how to obtain buy-in from leadership, and how to plan and track a migration project. I'll help you understand the migration process so you can properly prepare, execute, and test it. I'll suggest helper apps to do the heavy lifting and reduce risk. Finally, I'll suggest post-migration actions to keep the application happy and healthy for the future.

I recommend reading this guide once, start to finish. Then, use the worksheets, tips, and step-by-step instructions to plan a migration from Jira Server to Jira Data Center or Jira Cloud.



1.1 Who is this Book for?

This book is written for:

- **Jira Administrators** and **Project Managers** who are involved in every step of the migration process, including planning, execution, and communication.
- **Application Owners, Leaders, or Executives** who approve and oversee the overall effort. See the [Jira Deployment Differences](#), [Migration Options](#), and [Governance, Approval, and Change Management](#) chapters for initial decision and procedure information.
- **Systems Administrators, Database Administrators, and Network Engineers** who support the infrastructure. See the [Migration Process](#) chapter for all the technical details and considerations.
- **Test Team Members** who validate application data and functionality. See the [Migration Testing](#) section in the [Migration Process](#) chapter for test strategy details and sample test cases.



BIG NEWS: On October 16, 2020, Atlassian announced that it will stop selling new licenses of its Server products on February 2, 2021 and end support for its Server products on February 2, 2024. Customers will need to switch to Data Center, migrate to Atlassian-hosted Cloud products, or make alternate plans by February 2, 2024. Additionally, Server license renewal prices will increase on February 2, 2021, so customers may wish to act sooner depending on budgeting and resource planning considerations.

Read more: botron.io/atl-cloud-journey

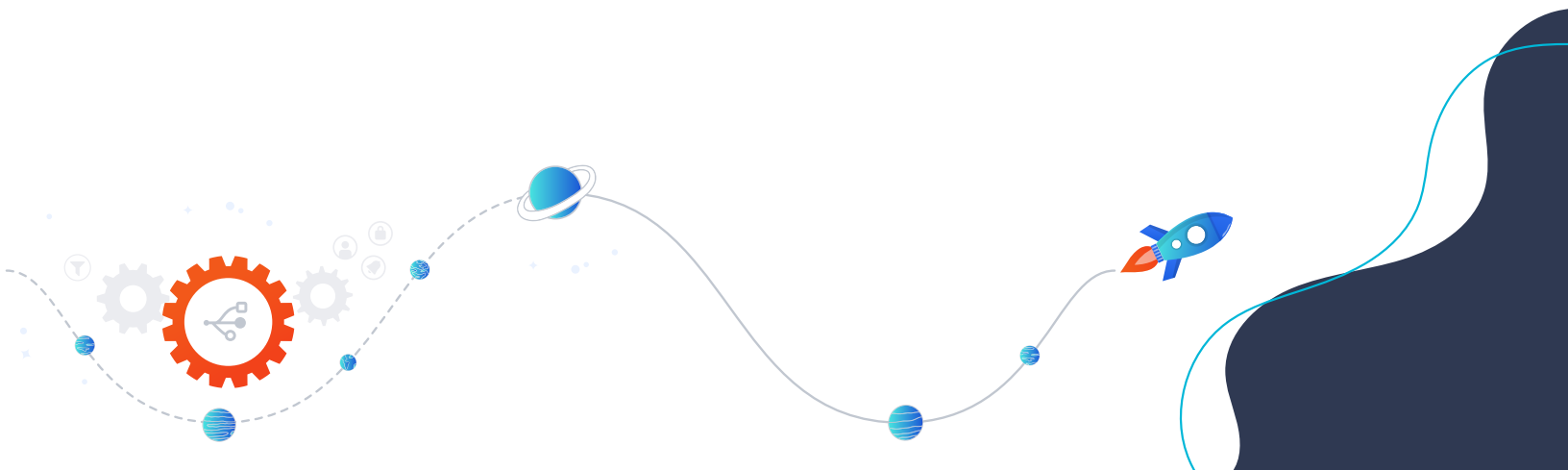
1.2 Why Migrate?

No solution fits forever and organizations must pivot as needs and industry capabilities change. Organizations often migrate to take advantage of new features or technical advances not previously available. For example, there are features in Jira Data Center and Jira Cloud that simply don't exist in Jira Server.

Additional reasons to consider migrating include when the current infrastructure is obsolete, unsupported, or expensive to maintain. It may be *possible* to upgrade the old hardware in the server closet, but it could be faster or cheaper to move to a third-party Data Center or a Cloud-hosted infrastructure instead. Or, reduce overhead and save money by consolidating hosting strategies for internal applications. You may gain other efficiencies, too. For example, you may require less IT staff. Resources could work on other strategic priorities instead.

A migration or merge is also an opportunity to reduce the complexity of the Jira configuration, reduce the amount of data, and remove the cruft that accumulates over the years. Do you really need to access Jira data from five years ago? Do you need to retain unused statuses, issue types, and custom fields? If not, consider backing up the info and leaving it behind. Use the migration as an opportunity to start fresh or cleaner than you started.

Finally, if you have multiple Jira applications, you may be able to consolidate user accounts, too.



Risks

Ideally, you're proactively considering migration and have plenty of time to think, research, and plan this process. But I know some of you don't have that luxury. Instead, you're responding to an emergency, repeated downtime incidents, or a critical piece of hardware or software is suddenly broken or unsupported. If that's your situation, take a deep breath. I promise you will get through it! We've all survived Jira emergencies, and so will you. Think of it as an opportunity to learn and be stronger for the next challenge.

There's never a perfect time to migrate, but sooner is usually better than later. There are risks to migrating and risks to *not* migrating as well. Migrations are complex, require careful planning and execution, and impact business operations with some level of downtime.

Each Jira deployment type has different features, capabilities, user interfaces, and terminology. There may be a slight learning curve, especially for end users. The navigation may be different, a favorite feature may work differently, or expected data may be missing. Third-party apps and integrations may be incompatible or work differently, too.

Careful consideration of the risks, options, and differences is a vital part of the migration planning process. Take as much time as you can to get it done right.

1.3 Terminology

The following terms are used throughout this book:

App – Software that extends the capabilities of Atlassian software. Apps were previously called add-ons and plugins. Install apps from the Atlassian Marketplace at botron.io/amp or from the *Find new apps* administration page in Jira.

“Classic” Jira Project – Traditional “Business”, “Software”, and “Service Management” Jira project types in Jira Server and Jira Data Center. Jira Cloud has all these projects and one additional project type. See also: “Next-gen” Jira Project

Cluster – A collection of multiple Jira applications, also called instances or nodes. A Jira Data Center environment is either clustered (with multiple applications) or non-clustered (with a single application). See also: Node

Jira Cloud – This deployment type is hosted by Atlassian. Atlassian is the System Administrator and is responsible for application uptime, upgrades, the database, and the mail server.

Jira Core – This application type contains all the main Jira features like projects, issues, workflows, and users. This type is best for business teams and for managing projects, processes, and tasks.

Jira Data Center – This deployment type is for mission-critical environments that require high availability, load balancing, redundancy, and disaster recovery features.

Jira Server – This application type is hosted on-premises (e.g., on a server in your office), in a data center with other internal applications, in a Cloud server environment (like Amazon AWS, Microsoft Azure, or Google Cloud Platform), or by a third party.

Jira Software – This application type runs on top of Jira Core. It is designed for development teams and teams using a Scrum or Kanban methodology. It includes dev-specific features like sprints, story points, and backlogs, as well as integration with tools like Bamboo and Bitbucket. See also: Jira Core

Jira Service Management – This application type runs on top of Jira Core and is built for support teams. This was previously called Jira Service Desk. It contains additional features like service level agreements, queues for grouping requests based on type and severity, and additional reporting for workload and customer satisfaction. It has a simple request entry interface called the *Customer Portal* and integrates with Confluence to display self-service help articles. See also: Jira Core

Load Balancer – A way to distribute traffic to application nodes. If there are many users on one node, this software routes new users to a different node. This helps to manage performance and impact. See also: Node

“Next-gen” Jira Project – Short for “Next-generation”. An additional project type in Jira Cloud. These projects are created by Jira administrators and regular users. They are schemeless, meaning they have independent settings not shared with other projects.

Node – An individual application in a clustered Data Center environment. Each node has a unique ID visible in the application footer. See also: Cluster

Production Environment – A live, real-time environment where applications are used for daily business operations.

Snapshot – A package of configuration information. For example, use the Configuration Manager for Jira app to take a snapshot of Jira projects to deploy on another server.

Source Application – The original location of data. In a Server to Data Center migration, Server is the *source* application and Data Center is the *target* application. See also: Target Application

Staging Application – A *pre-production* application specifically for verifying changes in real-life scenarios. Use this environment to test upgrades, try out third-party apps, experiment with scripts and customizations, and understand the impact of large clean-up efforts.

Target Application – The destination for migrated data. In a Server to Data Center migration, Server is the *source* application and Data Center is the *target* application. See also: Source Application

1.4 Conventions

The following styles are used throughout this book:

Item	Format	Example
Links to book sections	Underlined Text	<u>Jira Deployment Differences</u>
Links to external websites	Underlined URL	<u>botron.io/amp</u>
Breadcrumbs	Link name > link name	Admin > System > System Info
Web page names	Italic	<i>Find new apps</i>
Worksheets	Bold	Worksheet 1: Jira Migration Survey

2.0 Jira Deployment Differences

There are three Jira deployment types: Server, Data Center, and Cloud. Each type has a different user interface, features, capabilities, terminology, and app availability.

2.1 Server vs. Data Center vs. Cloud

Here are some differences between the deployment types:

Overview

	Jira Server	Jira Data Center	Jira Cloud
Setup	<ul style="list-style-type: none">• You set up the application server, database, and mail server• You install and configure the application• Single application	<ul style="list-style-type: none">• Same setup as Jira Server• Same user interface as Jira Server• Clustered or single (non-clustered) application	<ul style="list-style-type: none">• Fast setup• Fewer configuration settings to administer• Simple user interface• Single application
Hosting Options	<ul style="list-style-type: none">• On-premises• In a data center• In a cloud server environment (AWS, Azure, etc.)	<ul style="list-style-type: none">• Same options as Jira Server	<ul style="list-style-type: none">• On Atlassian's servers
Pricing	<ul style="list-style-type: none">• Tiered by user count• Perpetual licenses• 1-year or 2-year annual maintenance and support fees	<ul style="list-style-type: none">• Tiered by user count• 1-year or 2-year annual subscriptions	<ul style="list-style-type: none">• Per user• Monthly or 1-year or 2-year annual subscriptions• Free, standard, and premium plans

Overview

	Jira Server	Jira Data Center	Jira Cloud
Best For	<ul style="list-style-type: none">• Organizations with existing IT infrastructure and experienced System, Database, and Application Administrators• Strict compliance or governance requirements• Organizations already hosting other on-premises tools	<ul style="list-style-type: none">• Same as Jira Server plus:<ul style="list-style-type: none">◦ Mission-critical environments◦ High availability◦ Redundancy◦ Disaster recovery	<ul style="list-style-type: none">• Organizations without IT infrastructure, a Database Administrator, or a System Administrator• Organizations already using other cloud tools• Organizations that want immediate access to the newest features

Feature Differences

	Jira Server	Jira Data Center	Jira Cloud
User Limit	<ul style="list-style-type: none">• Unlimited	<ul style="list-style-type: none">• Unlimited	<ul style="list-style-type: none">• Max 10,000 users currently• Max 20,000 users planned for 2021
Control	<ul style="list-style-type: none">• Advanced control over:<ul style="list-style-type: none">◦ Data◦ Performance◦ Maintenance◦ Import• Own application server, database, and mail server• Listeners, logging, and profiling information	<ul style="list-style-type: none">• Same as Jira Server plus:<ul style="list-style-type: none">◦ Instant scalability◦ Load balancing◦ Rate limiting◦ Cluster monitoring	<ul style="list-style-type: none">• Frequent new features• Frequent user interface changes

Feature Differences

	Jira Server	Jira Data Center	Jira Cloud
Responsibilities	<ul style="list-style-type: none"> You're responsible for: <ul style="list-style-type: none"> Uptime Application server Database Mail server Indexing Privacy Security Compliance 	<ul style="list-style-type: none"> Same as Jira Server 	<ul style="list-style-type: none"> Atlassian is responsible for: <ul style="list-style-type: none"> Uptime Application server Database Mail server Indexing Privacy Security Compliance
Upgrades	<ul style="list-style-type: none"> Manual Custom schedule 	<ul style="list-style-type: none"> Same as Jira Server plus: <ul style="list-style-type: none"> Zero downtime 	<ul style="list-style-type: none"> Automatic
Backups	<ul style="list-style-type: none"> Manual Custom schedule Custom retention 	<ul style="list-style-type: none"> Same as Jira Server 	<ul style="list-style-type: none"> Automatic Daily 30-day retention
Access	<ul style="list-style-type: none"> Limit via IP Limit via firewall Single sign-on with Atlassian Crowd and apps 	<ul style="list-style-type: none"> Same as Jira Server plus: <ul style="list-style-type: none"> Single sign-on with SAML and OpenID Connect 	<ul style="list-style-type: none"> Single Atlassian account Single sign-on with apps
URL	<ul style="list-style-type: none"> Custom 	<ul style="list-style-type: none"> Same as Jira Server 	<ul style="list-style-type: none"> In the format <i>company.atlassian.net</i>
User Management	<ul style="list-style-type: none"> Local user management or central management through LDAP, Atlassian Crowd, Okta, etc. 	<ul style="list-style-type: none"> Same as Jira Server 	<ul style="list-style-type: none"> Central management through cloud services like Atlassian Access, G Suite, Okta, etc.

Feature Differences

	Jira Server	Jira Data Center	Jira Cloud
Custom Development	<ul style="list-style-type: none">• Scripting and REST API access• Custom email templates• HTML and JavaScript in issue fields	<ul style="list-style-type: none">• Same as Jira Server	<ul style="list-style-type: none">• Through the Atlassian Connect framework or Forge• Scripting through an app
Apps	<ul style="list-style-type: none">• Over 1,000	<ul style="list-style-type: none">• Over 500• Additional validation requirements	<ul style="list-style-type: none">• Almost 1,000
Project Types	<ul style="list-style-type: none">• Classic	<ul style="list-style-type: none">• Same as Jira Server	<ul style="list-style-type: none">• Classic and Next-gen
Custom Email Templates	<ul style="list-style-type: none">• Yes	<ul style="list-style-type: none">• Same as Jira Server	<ul style="list-style-type: none">• With apps
Archival	<ul style="list-style-type: none">• No	<ul style="list-style-type: none">• Yes	<ul style="list-style-type: none">• With premium plan
Content Delivery Network Support (CDN)	<ul style="list-style-type: none">• No	<ul style="list-style-type: none">• Yes	<ul style="list-style-type: none">• No
Custom Field Optimizer	<ul style="list-style-type: none">• With apps	<ul style="list-style-type: none">• Yes	<ul style="list-style-type: none">• With apps



Resources

- Compare Atlassian cloud vs. server - botron.io/atl-cloud-vs-server
- Functional differences in Jira Cloud product - botron.io/atl-jira-cloud-vs-server
- User management differences in cloud and server - botron.io/atl-cloud-server-user-management
- Server vs. Data Center – what’s right for you? - botron.io/atl-server-v-dc
- Jira Server and Data Center feature comparison - botron.io/atl-jira-server-dc-compare

2.2 Quiz

Download this worksheet! Get your customizable copy at botron.io/worksheets

Jira Migration Quiz

Take this quiz to determine which deployment type is best for your situation.

Questions	Answer
1. Do you have an existing IT infrastructure (e.g., an application server, database server, mail server, etc.)? A. Yes B. No C. Unknown	<input type="checkbox"/>
2. Where are other internal tools hosted (e.g., email, word processing, billing, invoicing software, etc.)? A. On-premises B. In a data center C. In the cloud (Ex: Amazon AWS, Microsoft Azure, Google Cloud Platform, etc.) D. Unknown	<input type="checkbox"/>
3. Do you have a Systems Administrator and a Database Administrator? A. Yes B. No C. Unknown	<input type="checkbox"/>
4. How experienced is the Jira Application Administrator? A. Beginner B. Intermediate C. Advanced D. Unknown	<input type="checkbox"/>
5. How many Jira users do you have? A. 1 - 10 users B. 11 - 1,999 users C. 2,000 - 5,000 users D. 5,000 users or more E. Unknown	<input type="checkbox"/>
6. Where do you want Jira user accounts stored? A. Locally in the Jira application B. In an LDAP directory C. In an on-premises application like Atlassian Crowd D. In a cloud application like Atlassian Access, G Suite, or Okta E. Unknown	<input type="checkbox"/>



Questions

Answer

7. How do you want to be billed?

- A. Per user
- B. By tier (e.g., Flat fee for 10-25 users)
- C. Unknown

☐

8. How often do you want to be billed?

- A. Monthly
- B. Annually
- C. Unknown

☐

9. What industry are you in?

- A. Non-profit
- B. Education or Open Source
- C. Finance, Health Care, or highly regulated area
- D. Technology, Professional Services, Media, Communication, Other
- E. Unknown

☐

10. Do you have specific security needs?

- A. Yes
- B. No
- C. Unknown

☐

11. Do you need to restrict access to the Jira application?

- A. Yes
- B. No
- C. Unknown

☐

12. Do you have specific compliance, privacy, governance, or audit requirements?

- A. Yes
- B. No
- C. Unknown

☐

13. How often would you like to upgrade?

- A. Manually, on a custom schedule
- B. Automatically, as new features are available
- C. Unknown

☐

14. How would you like to back up?

- A. On a custom schedule with a custom retention policy
- B. Automatically, daily, with a 30-day retention policy
- C. Unknown

☐



Questions

Answer

15. How much control do you need over data, performance, and indexing?

- A. More control (Your System Administrator handles everything)
- B. Less control (Atlassian handles everything)
- C. Unknown

☐

16. Do you need apps, integrations, or connections to other applications?

- A. Yes
- B. No
- C. Unknown

☐

17. Do you need custom configuration files or scripts?

- A. Yes
- B. No
- C. Unknown

☐

18. Do you need database access?

- A. Yes
- B. No
- C. Unknown

☐

19. Do you need additional customizations?

- A. Yes
- B. No
- C. Unknown

☐

20. Do you require a custom URL?

- A. Yes (e.g., jira.company.com)
- B. No (e.g., company.atlassian.net)
- C. Unknown

☐

21. Do you require high availability, redundancy, or disaster recovery features?

- A. Yes
- B. No
- C. Unknown

☐



Answers

1. Do you have an existing IT infrastructure (e.g., an application server, database server, mail server, etc.)?

Response	Recommendation	Notes
A. Yes	Data Center	It's easy to add Jira to existing servers and infrastructure
B. No	Cloud	Use Atlassian's application server, database server, and mail server
C. Unknown	n/a	n/a

2. Where are other internal tools hosted (e.g., email, word processing, billing, invoicing software, etc.)?

Response	Recommendation	Notes
A. On-premises	Data Center	Manage costs by hosting all applications with the same infrastructure
B. In a data center	Data Center	Same as above
C. In the cloud	Cloud or Data Center	Host server or data in Cloud infrastructure (Ex: Amazon AWS, Microsoft Azure, Google Cloud Platform, etc.) or use Atlassian's hosting for Jira Cloud
D. Unknown	n/a	n/a

3. Do you have a Systems Administrator and a Database Administrator?

Response	Recommendation	Notes
A. Yes	Data Center	A System Administrator and Database Administrator are needed to set up and maintain the application
B. No	Cloud	No System Administrator or Database Administrator required
C. Unknown	n/a	n/a



4. How experienced is the Jira Application Administrator?

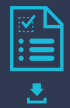
Response	Recommendation	Notes
A. Beginner	Cloud	There are fewer settings to configure and maintain
B. Intermediate	Data Center	There is more to set up and manage, like multiple application nodes. Regular maintenance is needed (e.g., periodically re-index issue data).
C. Advanced	Data Center	Same as above
D. Unknown	n/a	n/a

5. How many Jira users do you have?

Response	Recommendation	Notes
A. 1 - 10	Cloud	Use the free plan for 10 or less users. See the feature comparison at: botron.io/atl-jira-pricing
B. 11 - 1,999	Cloud or Data Center	n/a
C. 2,000 - 5,000	Cloud or Data Center	Organizations with over 2,000 users often have more complex needs
D. 5,000+	Data Center	Cloud is currently limited to 10,000 users
E. Unknown	n/a	n/a

6. Where do you want Jira user accounts stored?

Response	Recommendation	Notes
A. Locally	Data Center	User accounts and groups only exist in the Jira application. Credentials are Jira-specific. Only recommended for less than 50 users.
B. LDAP Directory	Data Center	Connect Jira to a corporate directory. Ex: Microsoft Active Directory, ApacheDS, Apple Open Directory, etc.
C. In an on-premises application	Data Center	e.g., Atlassian Crowd
D. In a cloud application	Cloud	e.g., Atlassian Access, G Suite, or Okta (Okta connects to any type of Jira)
E. Unknown	n/a	n/a



7. How do you want to be billed?

Response	Recommendation	Notes
A. Per user	Cloud	n/a
B. By tier	Data Center	First tier starts at 500 users
C. Unknown	n/a	n/a

8. How often do you want to be billed?

Response	Recommendation	Notes
A. Monthly	Cloud	Payable via credit card or PayPal
B. Annually	Cloud or Data Center	Payable via credit card, bank transfer, or check
C. Unknown	n/a	n/a

9. What industry are you in?

Response	Recommendation	Notes
A. Non-profit	Cloud or Data Center	Discounts exists for qualifying charitable organizations
B. Education or Open Source	Cloud or Data Center	Discounts exists for classroom use
C. Finance, Health Care, or highly regulated area	Depends	Regulations may dictate where and how data is accessed or stored
D. Technology, Professional Services, Media, Communication, Other	Cloud or Data Center	n/a
E. Unknown	n/a	n/a



10. Do you have specific security needs?

Response	Recommendation	Notes
A. Yes	Depends	Security needs may dictate where and how data is accessed or stored. See Atlassian's security info at: botron.io/atlassian-security
B. No	Cloud or Data Center	See above
C. Unknown	n/a	n/a

11. Do you need to restrict access to the Jira application?

Response	Recommendation	Notes
A. Yes	Cloud Premium or Data Center	Limit by IP address with Cloud Premium. Limit by IP address or firewall with Data Center.
B. No	Cloud or Data Center	Require login or allow anonymous access
C. Unknown	n/a	n/a

12. Do you have specific compliance, privacy, governance, or audit requirements?

Response	Recommendation	Notes
A. Yes	Depends	Compliance or regulatory needs may dictate where and how data is accessed or stored. See Atlassian's programs at: botron.io/atlassian-compliance and botron.io/atlassian-privacy .
B. No	Cloud or Data Center	See above
C. Unknown	n/a	n/a

13. How often would you like to upgrade?

Response	Recommendation	Notes
A. Manually	Data Center	Upgrade manually to manage change, downtime, and user training
B. Automatically	Cloud	Upgrades are applied automatically, as new features are available, on Atlassian's schedule
C. Unknown	n/a	n/a



14. How would you like to back up?

Response	Recommendation	Notes
A. Custom	Data Center	Back up application files, database records, and the attachments directory on a custom schedule. Retain backups according to your data retention policy.
B. Automatically	Cloud	Atlassian data backups are automatic, daily, and retained for 30 days
C. Unknown	n/a	n/a

15. How much control do you need over data, performance, and indexing?

Response	Recommendation	Notes
A. More control	Data Center	Your System Administrator handles everything
B. Less control	Cloud	Atlassian handles everything
C. Unknown	n/a	n/a

16. Do you need apps, integrations, or connections to other applications?

Response	Recommendation	Notes
A. Yes	Depends	Apps, integrations, and connections have varying compatibility. Check requirements for each in individual documentation and the Atlassian Marketplace at: botron.io/amp
B. No	Cloud or Data Center	See above
C. Unknown	n/a	n/a

17. Do you need custom configuration files or scripts?

Response	Recommendation	Notes
A. Yes	Data Center or Cloud with an app	Customize configuration files, property files, and more with Data Center. Run scripts in Cloud with an app.
B. No	Cloud or Data Center	Same as above
C. Unknown	n/a	n/a



18. Do you need database access?

Response	Recommendation	Notes
A. Yes	Data Center	Query for additional information. Make direct changes (not recommended). REST API is also available.
B. No	Cloud	No database access; however, there is a REST API.
C. Unknown	n/a	n/a

19. Do you need additional customizations?

Response	Recommendation	Notes
A. Yes	Data Center	Full access to application files
B. No	Cloud	No access to application files
C. Unknown	n/a	n/a

20. Do you require a custom URL?

Response	Recommendation	Notes
A. Yes	Data Center	Customize the URL with any domain or subdomain (e.g., <i>jira.company.com</i>)
B. No	Cloud	Use Atlassian's URL format (e.g., <i>company.atlassian.net</i>)
C. Unknown	n/a	n/a

21. Do you require high availability, redundancy, or disaster recovery features?

Response	Recommendation	Notes
A. Yes	Data Center	These features are <i>only</i> available in Data Center
B. No	Cloud	n/a
C. Unknown	n/a	n/a



Analyzing your Answers

Were the response recommendations mostly Cloud or Data Center? You may have the ability to compromise on some of these, or you may have unique needs that are deal-breakers. For example, you may be able to achieve a Jira Server customization through a third-party Jira Cloud app instead. If you need high availability, redundancy, or disaster recovery, then Jira Data Center is the only option. Consider which of your answers are flexible vs. must-haves for your organization.

Hopefully, this quiz helped you further understand the differences and considerations and helped point you in the right direction. You can further explore both deployment options using the available free trials at:

botron.io/atl-explore-cloud and botron.io/atl-jira-dc.

Note: This ebook contains 16 worksheets to help you plan your migration. All the worksheets are available in separate, editable files for you to download, customize, fill in, or print. Feel free to add your branding, change the information, or remove sections that don't apply. Use them all or just the ones you need. Store this information in Google Docs, Confluence, or wherever your other Jira documentation is located.

Download the worksheets at: botron.io/worksheets

2.3 Creating a Plan

I hope the previous section helped you determine which deployment type is best for your organization. Now it's time to gather some initial information about the source application and its intended target destination.

Use **Worksheet 1: Jira Migration Survey** to consider your goals and record your application details. Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 1: Jira Migration Survey



Planning a migration? Use this worksheet to gather some initial information about your plan, *source* application, and the intended *target* destination.

Migration Goals

Plan

Are you migrating to a different deployment type, changing hosting environments, or merging applications?

- | | | | |
|---|--|---|---|
| <input type="checkbox"/> Cloud to Cloud | <input type="checkbox"/> Server to Server | <input type="checkbox"/> Data Center to Data Center | <input type="checkbox"/> Merge Applications |
| <input type="checkbox"/> Cloud to Server | <input type="checkbox"/> Server to Data Center | <input type="checkbox"/> Data Center to Server | |
| <input type="checkbox"/> Cloud to Data Center | <input type="checkbox"/> Server to Cloud | <input type="checkbox"/> Data Center to Cloud | |

☐ Other _____

Motivations

Why are you migrating? Check all that apply.

- | | | | |
|--|--|---|---------------------------------------|
| <input type="checkbox"/> Save money | <input type="checkbox"/> Reduce complexity | <input type="checkbox"/> Improve infrastructure | <input type="checkbox"/> Start fresh |
| <input type="checkbox"/> Consolidate users | <input type="checkbox"/> Reduce overhead | <input type="checkbox"/> Consolidate infrastructure | <input type="checkbox"/> New features |
| <input type="checkbox"/> Reduce data | <input type="checkbox"/> Reduce cruft | <input type="checkbox"/> Previous solution obsolete | |
| <input type="checkbox"/> Gain efficiencies | | | |

☐ Other _____

Scope

How much data will the migration/merge include?

- | | | |
|--|---|----------------------------------|
| <input type="checkbox"/> Full data set | <input type="checkbox"/> Partial data set | <input type="checkbox"/> Unknown |
|--|---|----------------------------------|



Application Details

Source Application

The original location of data. In a Server to Data Center migration, Server is the *source* application and Data Center is the *target* application.

URL: _____

Deployment Type:	<input type="checkbox"/> Cloud	Version: _____	User Count: _____
	<input type="checkbox"/> Server	(Server or Data Center only)	License Exp: _____
	<input type="checkbox"/> Data Center		

Known problems or challenges:

Staging Application

The application for testing the migration procedure and migrated data. This is usually the same deployment type and configuration as the *target* application.

URL: _____

Deployment Type:	<input type="checkbox"/> Cloud	Version: _____	User Count: _____
	<input type="checkbox"/> Server	(Server or Data Center only)	License Exp: _____
	<input type="checkbox"/> Data Center		

Known problems or challenges:



Target Application

The destination for migrated data. In a Server to Data Center migration, Server is the *source* application and Data Center is the *target* application.

URL: _____

Deployment Type:	<input type="checkbox"/> Cloud	Version: _____	User Count: _____
	<input type="checkbox"/> Server	(Server or Data Center only)	License Exp: _____
	<input type="checkbox"/> Data Center		

Known problems or challenges:

Source Application Stats

Understand your source application by compiling some basic stats.

- In Jira Server or Data Center, visit Admin > System > System Info. Scroll down to *Database Statistics* for a count of records, assets, and schemes.
- In Jira Cloud, visit each administration page and count the records.

Record the totals in the table below.

Tip: Expect totals to fluctuate during the migration project. Compile these stats in the beginning and at the end of the migration so you can track growth and change over time. See a useful comparison graph in the “Post-migration Stats” section in the [After Migration](#) chapter.

Area	Count	Notes
Archived Issues (Total)		
Archived Projects		
Attachments		
Comments		
Components		
Custom Fields		
Groups		
Individually Archived Issues		



Area	Count	Notes
Issue Security Levels		
Issue Types		
Issues		
Issues In Archived Projects		
Priorities		
Projects		
Resolutions		
Screen Schemes		
Screens		
Statuses		
Users		
Versions		
Workflows		

2.4 Tracking Apps and Customizations

How many apps does the application have? Some apps are only compatible with specific deployment types and Jira versions. It's important to know which apps work in the target environment and which do not. Sometimes organizations have a lot of apps but not all are extensively used. Decide which are critical and which are "nice to have" if app availability or cost is an issue.

What other applications is Jira connected to or integrated with? Have you made visual or functional customizations? These factors impact the migration plan. Start by auditing the customizations you have, so you can decide which to retain. Then, verify compatibility, consider costs, and decide what to do with app data.

Use **Worksheet 2: Jira Apps and Customizations** to research and record apps, connections, integrations, and customizations. Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 2: Jira Apps and Customizations



Use this worksheet to gather initial information about Jira apps, connections, integrations, and customizations.

Apps

See the apps installed by visiting Admin > Manage Apps.

Tip: Use the Power Admin app by Botron to see how apps are currently used. Read more: botron.io/amp-pa

ID	Name & Marketplace URL	Version	Expiration	Cost	Use	Notes
Ex:	Configuration Manager for Jira	6.6.8	19/Apr/22	\$400	Test environment sync, quarterly maintenance, etc.	Also needed for June migration
1						
2						
3						
...						



Connections & Integrations

Tip: Check for connections and integrations in the following Jira administration areas: Admin > Manage Apps, Admin > Applications, and Admin > System > WebHooks.

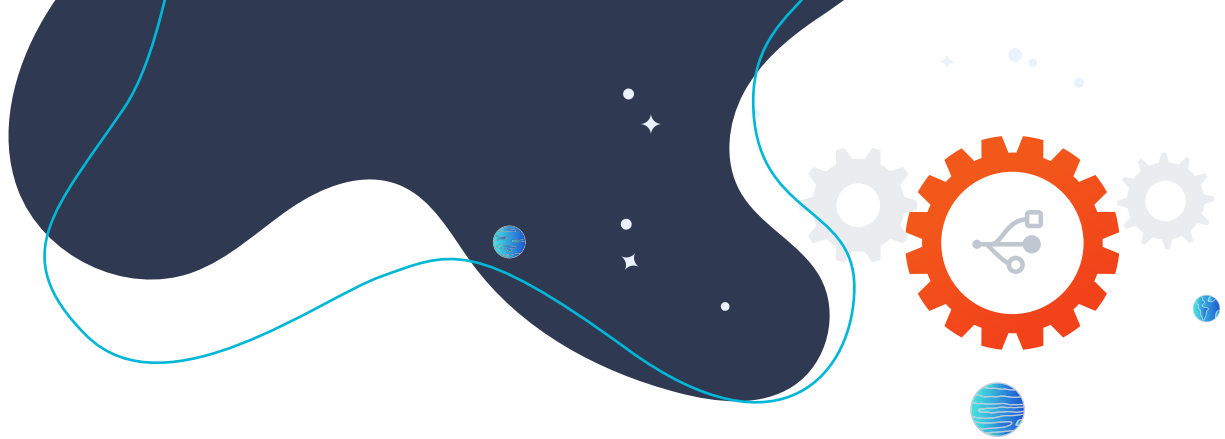
ID	Name & URL	Type	Expiration	Cost	Use	Notes
Ex:	<u>Confluence</u>	Application link	10/Jun/21	\$1,250	Document repository, collaboration portal, etc.	Same expiration as Jira
Ex:	<u>Slack</u>	WebHook	15/Dec/20	\$2,001	Chat	
1						
2						
3						
...						

Customizations

Enter any additional customizations.

ID	Name	Type	Use	Notes
Ex:	Outgoing mail disabled	Properties file	Disable email notifications in test environment.	Set "Datlassian.mail.senddisabled" to "true" in setenv.sh file
Ex:	Groovy script	Workflow validator	If the "Component" selected is "MS Office" require the user to select a value (e.g., Word, Excel, etc.) for the "Application Name" field in the "Request Details" transition.	If a value is not provided during the transition for field(s) Application Name, show the following error: Please select the application. This validation only applies if the following condition is true: <code>issue.get("components")?.any{it.name == "MS Office"}</code>
1				
2				
3				
...				

Tip: Check scripts and workflow behaviors for any hard-coded or application-specific values. If these values exist, be ready to make manual updates immediately after the migration.



3.0 Migration Options

Depending on your unique circumstances, you may have one or more potential migration paths. Here's a high-level overview of some common ones. While the considerations differ, the methods share many of the same initial planning, project management, execution, and testing steps detailed in later chapters. Use this section to determine which path to take and answer initial questions.

3.1 Jira Server to Jira Cloud

Atlassian has invested heavily in their Cloud infrastructure and they want customers to use it! Additionally, Atlassian announced they are decommissioning their Server products. The end of life date is February 2, 2024, and there are incentives to consider migrating sooner.

Atlassian has many tools and resources to encourage customers to migrate to Cloud. Among them is a generous Cloud free trial (read more: botron.io/atl-explore-cloud) and a dedicated Migration Support Team to assist with product, pricing, billing, licensing, and support questions. If you encounter a technical problem, or have a large enterprise application, Atlassian may offer additional services such as executing the migration on your behalf.

Learn more about Atlassian's support and resources available to teams migrating from Jira Server to Cloud: botron.io/atl-cloud-migration-support.

The support team doesn't help with apps, however. To migrate apps and their data, you'll need to use a tool or work with a Solution Partner. See the [Helper Apps](#) and [Choosing the Right Migration Partner](#) sections for recommendations.

The main Cloud migration tools are the Jira Cloud Migration Assistant (read more: botron.io/amp-cloud-migration-asst) and the Jira Site Import tool (read more: botron.io/atl-jira-site-import). You can also import issue data via CSV, with limitations.

Finally, there's a Cloud version of the Configuration Manager for Jira app in the works. Until it's ready, there's a Server to Cloud workaround. Read more: botron.io/projects-server-to-cloud

Read more about all these tools in the [Helper Apps](#) section in the "Migration Process" chapter.

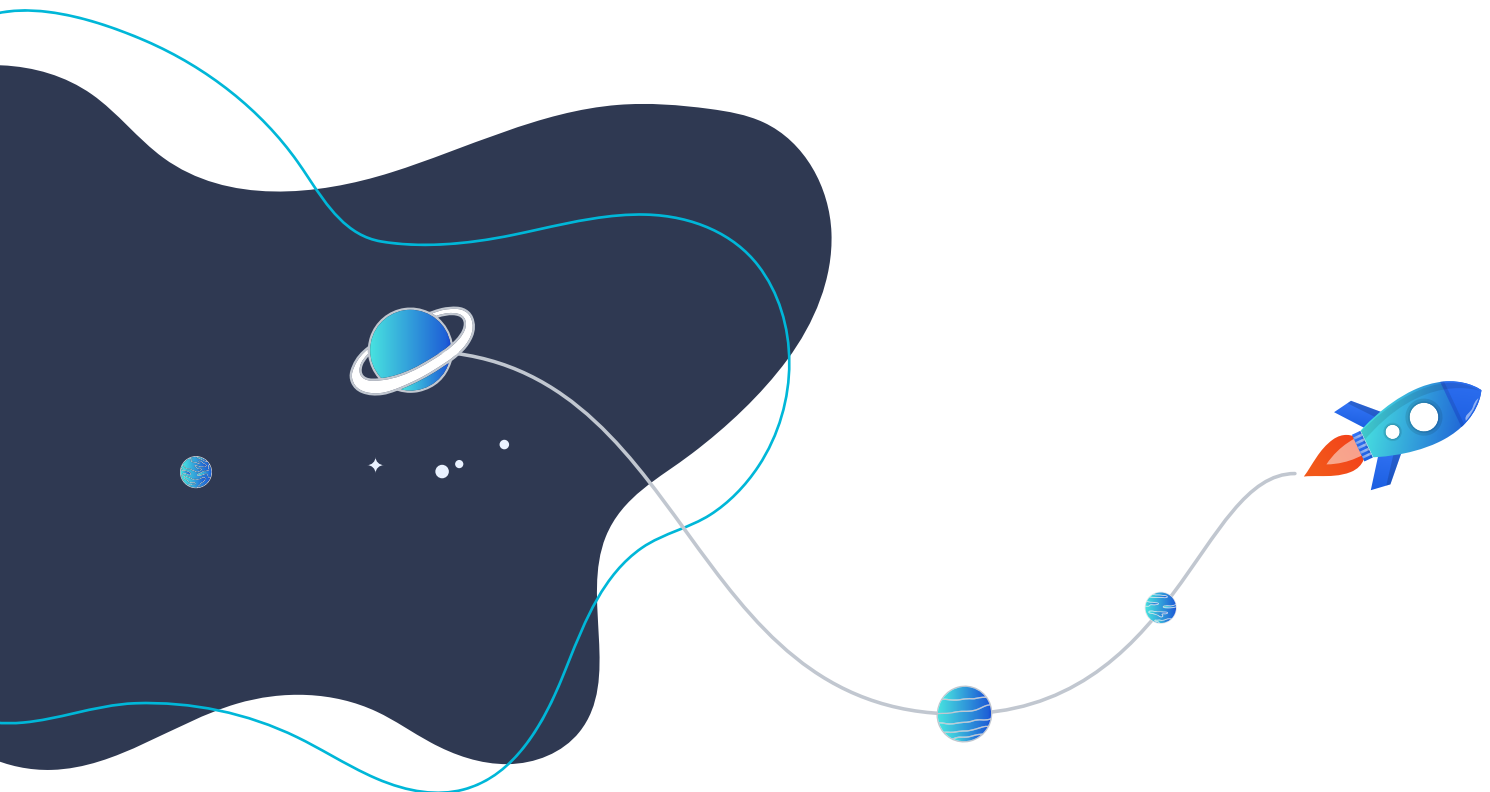
Cloud Differences

When Atlassian launched their first cloud-based version of Jira in 2011, it looked and functioned just like the Server version. But these days, Cloud and Server are quite different. They have different features, user interfaces, apps, terminology, and documentation. If this is your chosen path, be sure to factor these differences into initial and post migration plans.



Resources

- Cloud Migration Center - botron.io/atl-cloud-migration
- Cloud migration free trial - botron.io/atl-explore-cloud
- Compare cloud migration methods - botron.io/atl-cloud-migrate-methods
- Atlassian cloud migration support scope - botron.io/atl-cloud-migration-support
- Shared migration roles and responsibilities - botron.io/atl-shared-roles
- Use the Jira Cloud Migration Assistant to migrate from server to cloud - botron.io/atl-jira-cloud-migration-assistant
- Use Jira Site Import to migrate from server to cloud - botron.io/atl-jira-site-import
- What gets migrated with the Jira Cloud Migration Assistant - botron.io/atl-jira-cloud-migration-assistant-what
- FAQs about migrating to Jira or Confluence Cloud - botron.io/atl-cloud-migrate-FAQ
- Configuration Manager for Jira Server to Cloud Workaround Instructions - botron.io/projects-server-to-cloud



3.2 Jira Server to Jira Data Center

If your path is from Jira Server to Jira Data Center, there's good news. Data Center looks and functions just like Server from an end user perspective, so there's no user re-training needed. Managing Jira Data Center is like managing Jira Server, except there are a few new features and capabilities.

With this path, decide whether Jira Data Center will run as a **clustered application** or a **single application**.

A cluster is a collection of multiple Jira applications. Multiple Jira applications are also called *instances* or *nodes*. When Jira is mission-critical, multiple applications provide a safety net. If one application goes down, a load balancer automatically redirects users to a different Jira application. This process is invisible to the end user. The applications share the same file system and the same database, so they all function the same. Users see the exact same configuration and data, regardless of which instance they are using.

Benefits of Clustering

- Less drain on a single application. When demand is split between applications, users experience better performance and response time.
- Easy to scale. Quickly add and remove instances without downtime or additional license fees. Fees are based on total user count, not number of applications.
- Application count is only limited by size and system capabilities (e.g., RAM, heap space, etc.).
- Each application has its own logs, caches, indexes, and configuration files. If needed, you can modify one instance without impacting the others.
- Upgrade during business hours without impacting users. Simply put Jira into “upgrade mode” which shuts down, upgrades, and restarts each application, one at a time.



Tip: Devote one instance solely to Jira administration so clean-up, maintenance, and scheduled activities don't impact performance for regular users. Simply create an “administration only” instance and take it out of rotation in the load balancer.



Note: While clustering allows for more concurrent users, it doesn't inherently improve performance. For example, if too many Jira configuration elements impact application performance now, it's likely you'll experience the same performance issues in a clustered environment later.

Usage Example

Atlassian uses Jira Data Center to power its customer support service desk and to collect bug and feature requests for all their products. Using a clustered Data Center deployment model, the applications can support many millions of users and issues.

See their use case, infrastructure, sample load, and scale details at:

botron.io/atl-jira-dc-sample-deploy

Additional sizing information is also available here: botron.io/atl-jira-dc-size-profiles

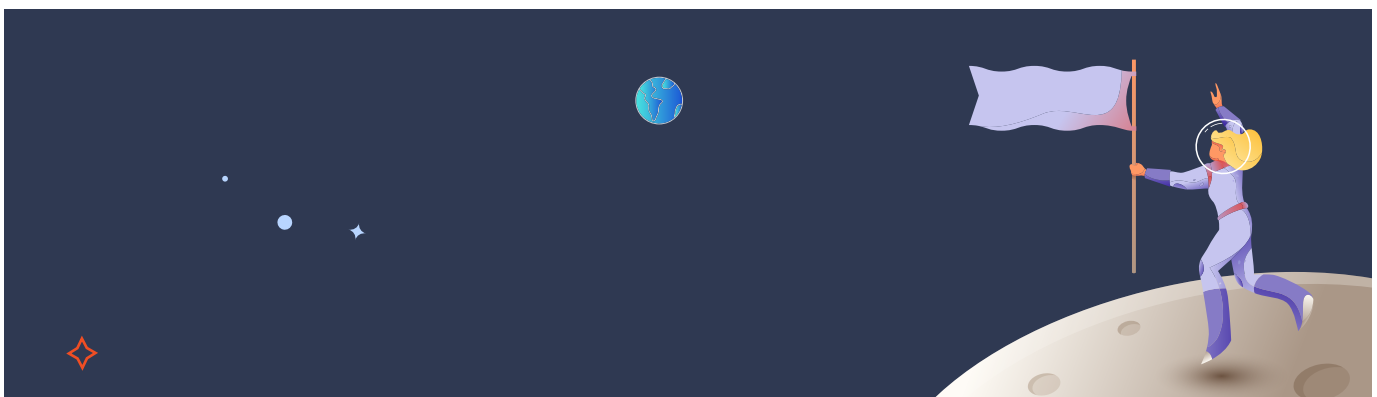
Benefits of a Single Application

- Upgrade the license without changing the infrastructure. The requirements and deployment steps are the same for a non-clustered application as Jira Server.
- Updating the license in Jira automatically unlocks Data Center-only features like SAML single sign-on, rate limiting, content delivery network support, and project archiving.

If clustering is more power than you need, simply run Jira Data Center as a single application and add additional instances in the future.

Usage Example

Moving to Jira Data Center simply because Atlassian is decommissioning Jira Server.



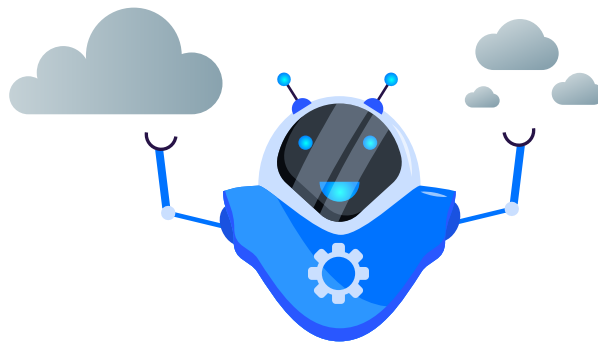
App Differences

Regardless of whether you choose a single or clustered application, you'll need to switch the apps to compatible Data Center versions. Data Center apps are built differently to handle the demands of highly available, clustered applications. Checking app compatibility is an important planning step for all migration paths.



Resources

- Jira Data Center FAQs - botron.io/atl-jira-dc-faq
- Moving to Jira Data Center - botron.io/atl-migrate-server-to-dc
- Running Jira Data Center in a cluster - botron.io/atl-jira-dc-cluster
- Set up a Jira Data Center cluster - botron.io/atl-jira-dc-cluster-setup
- Upgrading Jira Data Center with zero downtime - botron.io/atl-jira-upgrade-dc
- Running Jira Data Center on a single node - botron.io/atl-jira-dc-single
- Evaluate apps for Data Center migration - botron.io/atl-dc-apps-eval



3.3 Full vs. Partial Migration

Do you need to migrate inactive projects? How about custom field data used by expired apps? Are there five-year-old bugs you truly won't ever fix? Would it be nice to have less cruft for users to wade through?

For any migration path, decide whether to migrate all data or just some data. Scrutinize everything to make sure it's truly needed in the target application. Limiting the scope of information makes the migration easier.

Determine whether it's better to clean up unneeded data and settings before or after the migration. If you plan to clean up after, make sure it happens! Otherwise, you'll have a bigger configuration mess than before. Be sure to consider the long-term impact of adding more data and settings to the target application.

How to Limit Scope

- Determine the list of needed projects and issues. Exclude projects unused for X months
Exclude issues that are not likely to be addressed. (We all have them!)
- Determine which users to include. Do you need to migrate data from users who left the company years ago?
- Remove unneeded apps, schemes, and settings. For example, ask users to remove any old or unneeded filters, filter subscriptions, boards, and dashboards. Alternatively, ask users to specify which user objects to migrate. Visit each page of the Admin > Issues administration area, looking for unused or duplicate schemes to remove.
- Clean up any known problems. For example, fix or remove broken connections, webhooks, mail handlers, and other settings.

Getting the source application into the best possible state makes migration easier.

Export Old Data

Is it acceptable to remove unneeded data, or do you need to keep it for legal and historical reasons? What is your organization's backup and retention policy? Should you export for archival and restore purposes? Consider exporting old data, placing it in a long-term storage location, and removing it from the application.

When choosing an export format, determine the end goal. There's a big difference between formats for importing data back into Jira and formats for simply viewing issue information outside of Jira. Here are some options:

Goal	Format Options
Back up, re-import, archive	<ul style="list-style-type: none">• Copy database (Jira Server, Jira Data Center)• Copy attachments (Jira Server, Jira Data Center)• Copy server files (Jira Server, Jira Data Center)• Backup manager (Jira Cloud)
Viewing	<ul style="list-style-type: none">• Printable (Jira Server, Jira Data Center)• Print list (Jira Cloud)• Full Content (Jira Server, Jira Data Center)• Print details (Jira Cloud)• Word• HTML

Goal	Format Options
Manipulation	<ul style="list-style-type: none"> • Excel • Google Sheets (Jira Cloud) • CSV • XML

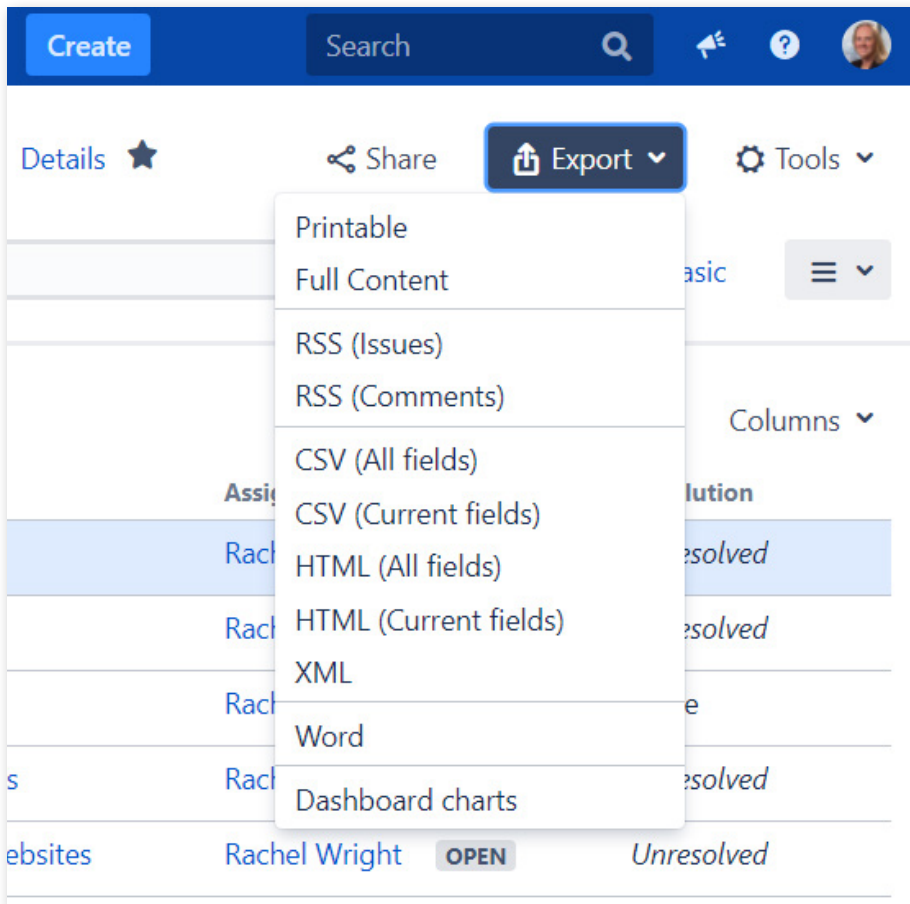


Image 1: Export Options in Jira Server

Caveats

There are some export caveats to be aware of. Exporting issues from the user interface is not as robust as a full backup. Important information like transitions, change history, comments, and attachments are not included. Additionally, most applications limit exports to 1,000 issues at a time. For example, if there are 5,000 issues, you'll need to export 1,000 issues five times.



Resources

- Exporting issues - botron.io/atl-cloud-export-issues
- Export Content to Word, PDF, HTML and XML - botron.io/atl-export-content

3.4 Phased Migration

What happens when you need to migrate, but one department isn't ready? What if some Jira projects depend on apps not yet compatible with the target application? The best answer for these circumstances may be to migrate in phases. Migrate the people and projects that are ready now, and tackle the others later. This means you'll complete multiple mini migrations. The entire migration process is much longer, but you gain timing flexibility.

3.5 Consolidation

Have more than one Jira application? It happens more than you'd think. Often there's an "official" Jira application and "unofficial" applications lurking in the background. Maybe there's a public application plus an internal one. Maybe the IT team has one running on an old server under someone's desk. Maybe the Marketing team set up their own Cloud application (or two!). After mergers, organizations often discover multiple applications to consolidate. If you're in this situation, you may need multiple migration events completed in a specific order.

Consolidation Questions

If you have many Jira applications, you'll need to determine the best order to merge them. Answer these questions to help determine a plan of attack.

1. What is the long-term application deployment strategy for the entire organization (e.g., Cloud, Server, Data Center)?
2. How many Jira applications exist?
3. Which deployment types exist?
4. What is each application used for?
5. How many users does each application have?
6. Who uses each application?
7. How large is each application?
8. Which apps, integrations, or customizations exist?

Consolidation Use Case

Consider the following scenario: A large technology company in the United States just completed the first of many acquisitions. They bought a small software development firm in Germany. Now the combined organization has multiple Jira instances with multiple purposes, two sets of users, and different location-based compliance requirements.

Questions		Answers
1.	What is the long-term application deployment strategy for the entire organization (e.g., Cloud, Server, Data Center)?	The on-premises infrastructure at headquarters is aging and costly to maintain. The organization is considering moving to a cloud-hosted environment.
2.	How many Jira applications exist?	<ul style="list-style-type: none"> • One on-premises Jira Server application in the US • One Atlassian-hosted Jira Cloud application • One cloud-hosted Jira Server application in Germany
3.	What deployment types exist?	<ul style="list-style-type: none"> • Application 1: Jira Server (US) • Application 2: Jira Cloud • Application 3: Jira Server (Germany)
4.	What is each application used for?	<ul style="list-style-type: none"> • Application 1: Strategic company priorities, department requests, team tasks • Application 2: Marketing tasks and campaigns • Application 3: Development features and bugs
5.	How many users does each application have?	<ul style="list-style-type: none"> • Application 1: 2,000 • Application 2: 50 • Application 3: 200
6.	Who uses each application?	<ul style="list-style-type: none"> • Application 1: All US staff and some German staff • Application 2: Only Marketing staff • Application 3: Only German staff
7.	How large is each application?	<ul style="list-style-type: none"> • Application 1: 500,000 issues, 30 projects, 85 custom fields • Application 2: 400 issues, 3 projects, 15 custom fields • Application 3: 2,000 issues, 10 projects, 250 custom fields

Questions	Answers
<p>8. Which apps, integrations, or customizations exist?</p>	<ul style="list-style-type: none"> • Application 1: <ul style="list-style-type: none"> ◦ ScriptRunner for Jira ◦ Configuration Manager for Jira ◦ JSU Automation Suite for Jira Workflows • Application 2: <ul style="list-style-type: none"> ◦ JSU Automation Suite for Jira Workflows ◦ Trello connection • Application 3: <ul style="list-style-type: none"> ◦ Clone Plus for Jira ◦ GitHub connection ◦ Slack WebHook

Analyzing the Answers

Given the above information, what may impact this consolidation? What factors should be specifically considered?

First, the organization has other acquisitions in its future. It's important to understand the long-term strategy for internal applications and make sure the consolidation plan accommodates those goals. We already know that headquarters wants to decommission their on-premises servers and move to a cloud-hosted solution. What is the timeframe for this change? Will the existing infrastructure support additional data and load in the meantime? How well are the existing apps functioning right now?

Next, applications are currently hosted in two different countries. What are the security and compliance requirements for each country? For example, does country A allow customer data stored on servers in country B? Are there performance or latency concerns between different data locations? What other governance requirements exist?

Next, consider how the applications are used. Are there any material differences? Is there a business reason the Marketing department needs a dedicated Cloud application? Could all users track their work in a single Jira application instead? What are the benefits and challenges of consolidation?

In the scenario, there is some user access overlap. The Jira Server (US) application is accessed by some German staff members. If the usernames for these staff members are not the same in both applications, you'll need to address username format differences.

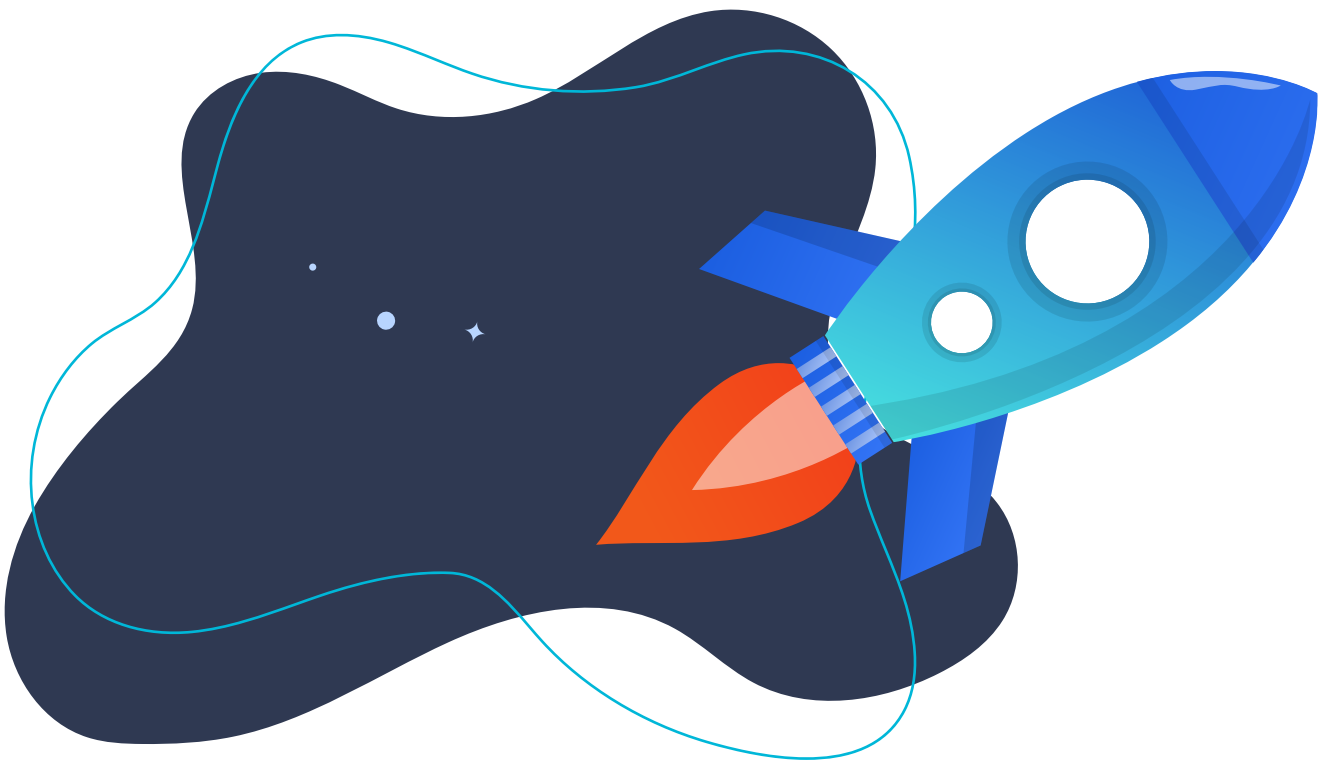
The Jira Server (Germany) application has a lot of custom fields for its size. Are all those fields really needed? Can clean-up be completed before consolidation? Do any custom fields have the same name in multiple applications?

Finally, consider customizations, integrations, connections, and apps. Both the Jira Server (US) and Jira Cloud applications use the JSU Automation Suite for Jira Workflows app. Are there any differences between how the app works in different environments? What license costs will change? Will the organization continue to use Slack as its chat tool or does another tool need to be connected?

As you can see, there are many considerations to uncover and discuss before determining a consolidation strategy.

“As Is” vs. Transformative Migration

Another decision to make is whether you’ll migrate the data and configuration exactly as it is in the source application or if you’ll need to transform the data. Transformations are often necessary to avoid conflicts and collisions. For example, if both the source and target applications have a project with the same key, you can’t migrate the project “as is”. You’ll need to change the key and any elements (e.g., queries, links, etc.) referencing the key. If both applications have a custom field with the same name, you’ll have duplicates to handle now or clean up later.



3.6 Starting Fresh

Sometimes it's smarter to start fresh instead of spending time and money fixing what you have. Wouldn't it be nice to have a brand new, current, and clean Jira application? It's every admin's dream and an opportunity to leave cruft, old issue data, and configuration problems behind. I recommend this strategy when an application requires multiple upgrades to catch up to the latest version, when there are recurring performance or instability issues, or when there's a lot of old data that's no longer needed. Here are some points to consider:

Pros	Cons
<ul style="list-style-type: none">• Reduce the size and scope of the application• Leave old or unused projects behind• Ditch ancient or unneeded issues• Regain attachment or database storage space• Revamp application usage, user access, or configuration strategies• Leave past configuration mistakes behind• Ditch needless customizations and use standard features instead• Gain new features and performance improvements• Escape old log errors or corruption• Ditch database tables created by old or unused apps• Update the database schema to the latest standard• Upgrade the database version or change database types• Update app and application versions• Remove old user accounts• Clear out unneeded objects (e.g., old boards, saved filters, etc.)• Ditch un-maintained groups• Restart issue numbering (e.g., ISSUE-1 instead of ISSUE-1058741)	<ul style="list-style-type: none">• Resistance to change (even when it's really needed)• Real (or perceived) business interruption or downtime• Fear of the unknown• Loss of specific app or application features• Loss of access to old data• Archival, retention, or long-term storage of previous data• Data stored in multiple places (e.g., old database and new database)• Potential inability to restore old data (if ever needed)• Potential license and support cost increase• Potential infrastructure upgrades needed (e.g., faster storage, better hardware, etc.)• User impact or adoption issues• User and administrator re-training

Hopefully, the previous information helped you determine the best migration option. Now that you've identified a path, the upcoming [Migration Process](#) chapter will help you create a detailed plan. But first, you'll need to get everyone in your organization on the same page.

4.0 Governance, Approval, and Change Management

Who are the Jira stakeholders? What is the approval process at your organization? How should you track large changes? A migration is a large event. It should be responsibly managed and communicated like any other large strategic project. The many groups using or overseeing Jira are all impacted in different ways.

4.1 Identifying Stakeholders

Before you start the migration or approval process, identify the impacted groups, how they are affected, what each needs to know, and any specific actions they need to take. Also think about the ways the migration might benefit them. Identifying preliminary info now will be useful later for creating a proposal and communications plan.

Impacted parties often include: Application owners, leadership, system administrators, database administrators, network engineers, help desk technicians, administrators of other connected applications, accounts payable reps, security/compliance officers, Jira application administrators, Jira project administrators, team or project leads, power users, and end users. What other areas are impacted in your organization?

Use **Worksheet 3: Jira Migration Stakeholders** to compile contacts and information. Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 3: Jira Migration Stakeholders



Use this worksheet to identify migration stakeholders and create a list of contacts to use with your migration communications plan. Next, start noting each area's impacts, needs, and any actions they will need to take throughout the project.

Examples

Group/Area:	Finance Department	Lead Contact:	Chris Jones
Benefits:	<ul style="list-style-type: none">One bill and one billing due date		
Impacts:	<ul style="list-style-type: none">Billing frequency changeCost change		
Concerns:	<ul style="list-style-type: none">TBD		
Actions:	<ul style="list-style-type: none">Update vendor terms in the payment system		



Group/Area:	Jira Administrators	Lead Contact:	Kelly Smith
Benefits:	<ul style="list-style-type: none">• New features like project archiving• New admin-only node		
Impacts:	<ul style="list-style-type: none">• Improved infrastructure• Different URLs		
Concerns:	<ul style="list-style-type: none">• New clustered environment to learn and maintain		
Actions:	<ul style="list-style-type: none">• Read Atlassian documentation• Attend load balancer training		

Stakeholders

Group/Area:	<enter group>	Lead Contact:	<enter name>
Benefits:	<ul style="list-style-type: none">•		
Impacts:	<ul style="list-style-type: none">•		
Concerns:	<ul style="list-style-type: none">•		
Actions:	<ul style="list-style-type: none">•		

Group/Area:	<enter group>	Lead Contact:	<enter name>
Benefits:	<ul style="list-style-type: none">•		
Impacts:	<ul style="list-style-type: none">•		
Concerns:	<ul style="list-style-type: none">•		
Actions:	<ul style="list-style-type: none">•		

Group/Area:	<enter group>	Lead Contact:	<enter name>
Benefits:	<ul style="list-style-type: none">•		
Impacts:	<ul style="list-style-type: none">•		
Concerns:	<ul style="list-style-type: none">•		
Actions:	<ul style="list-style-type: none">•		

After identifying stakeholders and impacted areas, use **Worksheet 4: Jira Migration Team** to build a team. Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 4: Jira Migration Team



Use this worksheet to record members of the migration team.

Sample participants:

- | | |
|---|---|
| <input type="checkbox"/> Project manager | <input type="checkbox"/> Security/compliance officer |
| <input type="checkbox"/> Application owner | <input type="checkbox"/> Jira application administrator |
| <input type="checkbox"/> IT leadership | <input type="checkbox"/> Jira project administrator |
| <input type="checkbox"/> System administrator | <input type="checkbox"/> Team lead |
| <input type="checkbox"/> Database administrator | <input type="checkbox"/> Power user |
| <input type="checkbox"/> Network engineer | <input type="checkbox"/> End user |
| <input type="checkbox"/> Help desk technician | <input type="checkbox"/> Atlassian representative |
| <input type="checkbox"/> Admins of other connected applications | <input type="checkbox"/> Vendor contact |

Team Members

ID	Name	Contact Details	Time Zone	Role
Ex:	Rachel Wright	rwright@jirastrategy.com	EST	Source Jira Admin
1				
2				
3				
4				
5				
6				
...				

4.2 Getting Buy-in

You may need to convince one or more groups that migration is needed. You may even need to submit a formal proposal. The key to a successful proposal is communicating the goals, the benefits, and the plan. You should tailor the proposal to its audience and include enough information so the reviewer can make an *informed* decision. Keep in mind that the reviewer may not be familiar with Jira specifics. If there's a company proposal template, use it. The template might contain additional aspects to consider, and the reviewer will appreciate a consistent format.

Don't think of a proposal as unnecessary extra work; use it as an opportunity to think through all the major project scenarios and communicate plans to others. Have a colleague review the proposal before you submit it. Finally, after submission, be ready to answer questions or provide additional details when needed.



Tip: Craft a project time estimate and then double it! Migrations are always more complex and take longer than you think.

Questions

Use these questions to help create a proposal. Use your answers from **Worksheet 1: Jira Migration Survey** in the [Creating a Plan](#) section.

1. What are the project goals and objectives?
2. What are the reasons to migrate? Why now?
3. What are the benefits?
 - a. Include new features, efficiencies, consolidation opportunities, etc.
4. What are the risks of not migrating?
 - a. Cite any recent events or pain points (e.g., "Prevent another downtime incident")
5. What are the expected costs or savings?
 - a. Include license, service, equipment, and time (manpower) costs.
6. What is the proposed migration path (e.g., Jira Server to Jira Data Center)?
7. Who is impacted?
8. How much data is included? (See the [Full vs. Partial Migration](#) section.)

9. What is the approximate timeline?

- a. Include extra time for research, planning, performing the migration in a staging environment, and testing.

10. What dependencies exist?

- a. Include resource needs, potential collisions with other strategic projects, release schedules, etc.

11. What are the known challenges and risks?

- a. Include downtime, knowledge gaps, etc.

12. What still needs to be researched, discovered, decided, or determined?



Tip: Need help calculating or comparing migration costs? Ask an Atlassian Solution Partner or Atlassian Technical Account Manager (TAM) for estimates.

4.3 Approval

What is your organization's approval process? There may be an approval at the beginning to start a project, or an approval at the end to accept the results. Or there may be both. Highly regulated organizations might require multiple approval steps. There may be approvals for plans, test cases, results, and more. There may even be signatures collected at specific milestones. Be sure to understand and follow your organization's approval process. Don't fail an audit or delay the project because needed procedures weren't correctly followed.

Regardless of what your company requires, make sure leadership is aware of the project and agrees on the overall execution strategy. Always document the strategy and any approval steps.

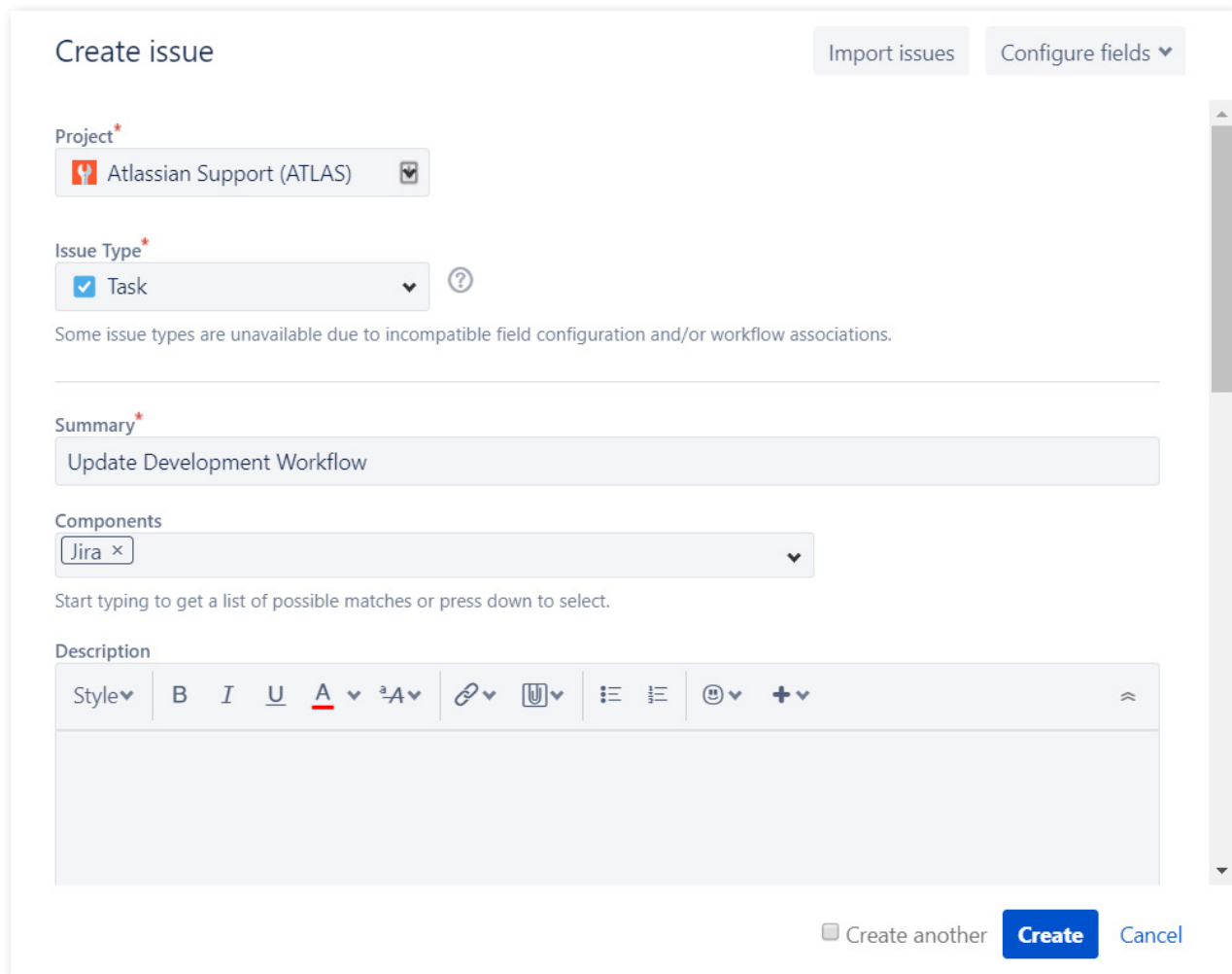


Tip: Use Jira and Confluence to track and document the entire migration project's lifecycle.

4.4 Tracking Changes

Where do you record Jira configuration changes? How do you communicate them to fellow administrators and users? Hopefully, the answer isn't sticky notes, emails, or nowhere at all!

I recommend using a Jira project to track the effort of supporting Jira and other internal applications. If you have Confluence, create a companion space, and link it to the Jira project, so you can document decisions, discussions, and changes. You can also use Jira Service Management to manage the request and approval process. Here's an example of tracking Jira changes using Jira.



The screenshot shows the 'Create issue' interface in Jira. At the top, there are buttons for 'Import issues' and 'Configure fields'. The 'Project' dropdown is set to 'Atlassian Support (ATLAS)'. The 'Issue Type' dropdown is set to 'Task'. Below this, a message states: 'Some issue types are unavailable due to incompatible field configuration and/or workflow associations.' The 'Summary' field contains the text 'Update Development Workflow'. The 'Components' section shows 'Jira' as the selected component. Below this, a prompt says: 'Start typing to get a list of possible matches or press down to select.' The 'Description' field is empty, with a rich text editor toolbar above it. At the bottom right, there are three buttons: 'Create another' (disabled), 'Create' (active), and 'Cancel'.

Image 2: Sample Jira Support Project Create Screen

Tracking Jira changes in Jira works perfectly — except when the application is unavailable for maintenance, upgrade, or migration activities! It's easy to plan around these events, however. Before any planned downtime, export the important project-specific Jira issues or Confluence pages so the information is available offline. Once the application is available again, update any changed or outdated details. Make sure the information in the application remains the source of truth. Delete any old, offline versions of exported data.

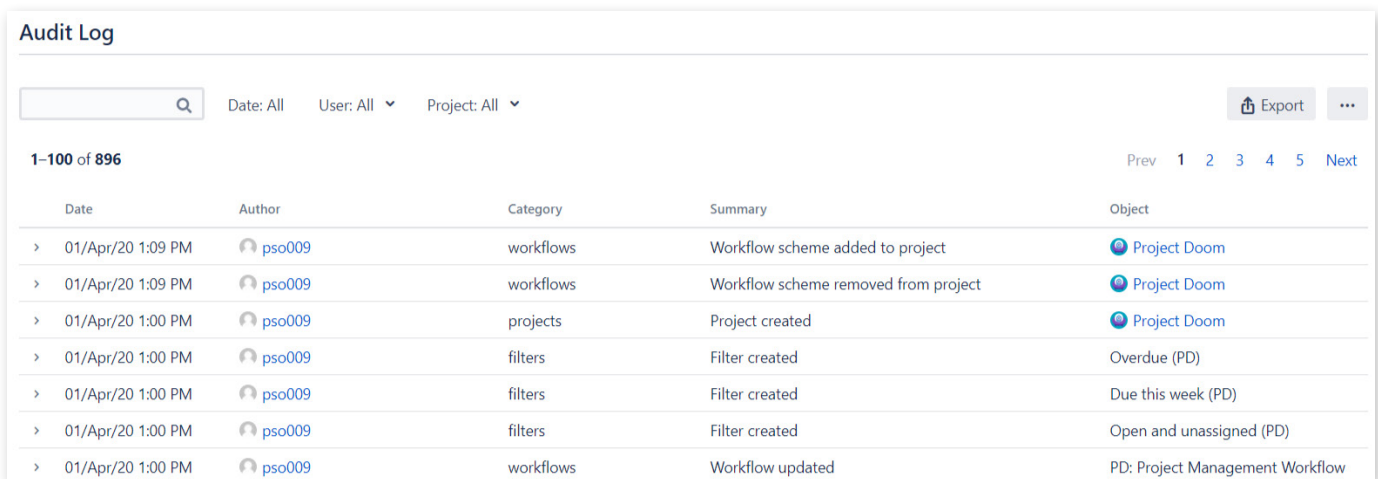
What to track?

My personal rule is to document any configuration changes that only an application administrator or system administrator can make. Those changes include requests to:

- Add, modify, remove, or archive a Jira project,
- Add, modify, remove, or deactivate a user,
- Change user or group permissions,
- Customize settings or schemes,
- Perform maintenance or clean-up activities,
- Upgrade the application, and
- Add, modify, remove, or test an app or integration.

I wouldn't record adding a Component, however, as any project-level administrator can make that type of change.

Jira has an audit log, but it provides little detail. The audit log records changes and timestamps, but it can't capture *why* a certain change was made, *who* requested it, and other important change details.



The screenshot shows the Jira Audit Log interface. At the top, there's a search bar and filters for Date, User, and Project. Below the filters, it indicates '1-100 of 896' results. A table lists the audit entries with columns for Date, Author, Category, Summary, and Object. The entries show various actions like adding/removing workflow schemes, creating projects and filters, and updating workflows, all performed by user 'pso009' on April 1st, 2020.

Date	Author	Category	Summary	Object
> 01/Apr/20 1:09 PM	pso009	workflows	Workflow scheme added to project	Project Doom
> 01/Apr/20 1:09 PM	pso009	workflows	Workflow scheme removed from project	Project Doom
> 01/Apr/20 1:00 PM	pso009	projects	Project created	Project Doom
> 01/Apr/20 1:00 PM	pso009	filters	Filter created	Overdue (PD)
> 01/Apr/20 1:00 PM	pso009	filters	Filter created	Due this week (PD)
> 01/Apr/20 1:00 PM	pso009	filters	Filter created	Open and unassigned (PD)
> 01/Apr/20 1:00 PM	pso009	workflows	Workflow updated	PD: Project Management Workflow

Image 3: Jira Audit Log

Historical change data is especially valuable to other administrators to understand why a setting is the way it is, to troubleshoot application issues, or to conduct research requested by the Legal, Security, or Compliance teams.

5.0 Migration Process

Now that you've explored the different migration options and engaged stakeholders, it's time to prepare and execute your migration.

5.1 Migration Checklist

Use **Worksheet 5: Jira Migration Checklist** to identify steps to complete before, during, and after a migration. Add your own situation-specific items and customize the checklist as you work through this chapter.

Download this worksheet! Get your customizable copy at botron.io/worksheets



Note: The checklist is an overview of the entire migration process. Items not previously discussed are addressed in future sections.

Worksheet 5: Jira Migration Checklist



Use this worksheet to plan and execute a Jira migration event.

Before

Project Preparation

1. ☐ **Determine goals and motivations**
Why migrate? What are the benefits? What features or efficiencies are gained? What is reduced? What are the risks of not migrating? Use **Worksheet 1: Jira Migration Survey** in the [Creating a Plan](#) section.
2. ☐ **Research deployment differences**
What feature, pricing, capability, and app differences exist between Jira Cloud, Server, and Data Center? Take the quiz in the [Jira Deployment Differences](#) section to determine which deployment type is best for your situation.
3. ☐ **Determine migration path**
e.g., Jira Server to Jira Data Center, Jira Server to Jira Server (merge/consolidation), start fresh, etc.
4. ☐ **Determine initial migration scope**
Do you plan to migrate all data or some data?



5. ☐ **Gather source application details**

What is the source application's URL, deployment type, and version? How many users are there? What are the known problems or challenges? Use **Worksheet 1: Jira Migration Survey** in the [Creating a Plan](#) section.

6. ☐ **Gather source application stats**

How many projects, issues, attachments, and custom fields exist? Use **Worksheet 2: Jira Apps and Customizations** in the [Tracking Apps and Customizations](#) section.

7. ☐ **Gather source app and customization details**

Which apps are installed in the source application? Use **Worksheet 2: Jira Apps and Customizations** in the [Tracking Apps and Customizations](#) section.

8. ☐ **Verify source license expiration date**

Verify expiration date is valid through the migration period. Visit: Admin > System > License

9. ☐ **Review migration tool features and limitations**

- To Cloud: Jira Cloud Migration Assistant, Jira Site Import, Configuration Manager for Jira, CSV Import, etc.
- To Data Center: Jira Data Center Migration App, Configuration Manager for Jira, CSV Import, etc.

10. ☐ **Determine application model (For Jira Data Center only)**

Will you run Jira as a clustered application or single application?

11. ☐ **Examine migration scope**

Are there projects, issues, or settings to exclude?

12. ☐ **Research long-term storage and retention policy**

Where can historical data be stored? How long does it need to remain available? In what format?

13. ☐ **Identify stakeholders**

Create a list of impacted groups and lead contacts. Record benefits, impacts, concerns, and actions. Use **Worksheet 3: Jira Migration Stakeholders** in the [Governance, Approval, and Change Management](#) section.



14. ☐ **Identify migration team members**
Use **Worksheet 4: Jira Migration Team** in the [Governance, Approval, and Change Management](#) section to build a project team.
15. ☐ **Get stakeholder buy-in**
Determine whether you need to convince others that migration is needed.
16. ☐ **Create migration proposal**
Create a proposal to communicate the migration goals, benefits, and high-level plan.
17. ☐ **Research approval requirements**
What approvals are needed? When? By whom?
18. ☐ **Get preliminary project approval**
Make sure leadership is aware of the project and agrees on the overall execution strategy.
19. ☐ **Create a project tracking issue in Jira**
Use Jira to track Jira support.
20. ☐ **Create a project documentation space in Confluence**
Create a companion Confluence space to document Jira support. Link the Confluence space to the Jira project.
21. ☐ **Setup regular project meetings**
Create a regular project schedule to share updates, discuss issues, and make decisions.
22. ☐ **Create a decisions list**
Use **Worksheet 6: Jira Migration Decisions List** in the [Migration Preparation](#) section to log and resolve issues.
23. ☐ **Conduct source configuration assessment**
Use **Worksheet 7: Jira Migration Configuration Assessment** in the [Configuration Assessment](#) section to assess the source application.



Application Preparation

24. ☐ **Upgrade source Jira and apps**

Upgrade Jira and apps, if needed. Update **Worksheet 2: Jira Apps and Customizations** in the [Tracking Apps and Customizations](#) section.
25. ☐ **Evaluate app compatibility**

Are all apps needed in the target application? Are all compatible with the target application? What functional differences exist? Use **Worksheet 10: Jira Migration App Decisions** in the [Must-have Apps](#) section.
26. ☐ **Evaluate connections, integrations, and customizations**

Are all connections and integrations needed in the target application? Are all compatible with the target application? What customizations need additional research or special treatment? Do customizations work in a clustered environment (e.g., Jira Data Center)?
27. ☐ **Back up source application**

Develop and test a backup strategy. Back up the source application.
28. ☐ **Archive unneeded source projects**

Archive projects or export unneeded issues for long-term storage.
29. ☐ **Clean up source application**

Remove any unused schemes or apps. Reduce duplication and unneeded settings.
30. ☐ **Run source health checks (For Jira Server and Jira Data Center only)**

Resolve any issues reported by the following tools:

 - „ Instance Health Tool (Admin > System > Troubleshooting and support tools)
 - „ Integrity Checker (Admin > System > Integrity checker)
31. ☐ **Set up target application**

Install and configure the target application. Verify that system settings (e.g., mode, default language, user locale, default user time zone, etc.) match the source application. Verify user limit is adequate. Verify the license expiration date is valid through the migration period. Visit: Admin > System > License



32. ☐ **Gather target application details**

What is the target application's URL, deployment type, and version? Use **Worksheet 1: Jira Migration Survey** in the [Creating a Plan](#) section.

33. ☐ **Conduct target configuration assessment (for merge/consolidation)**

Use **Worksheet 7: Jira Migration Configuration Assessment** in the [Migration Preparation](#) section to compare source and target application configurations.

34. ☐ **Set up staging application**

Install a staging application using the same settings and specs as the target application.

35. ☐ **Gather staging application details**

What is the staging application's URL, deployment type, and version? Use **Worksheet 1: Jira Migration Survey** in the [Creating a Plan](#) section.

36. ☐ **Run source performance tests (Jira Server and Jira Data Center)**

Run performance tests to ensure the source, staging, and target applications can handle the import/export process, additional data, and new user load. Increase system resources (e.g., computing memory, JVM memory, etc.) in the target application to handle additional data and user requests. Add a dedicated node in Jira Data Center.

37. ☐ **Determine helper apps**

Determine which apps are needed for the migration. Install apps on other applications as needed.

38. ☐ **Check source object integrity**

Use the [Integrity Check for Jira](#) app to detect and address filter, dashboard, and board errors.

Migration Preparation

39. ☐ **Determine user account strategy**

Use **Worksheet 13: Jira Migration User Account Strategy** in the [Merge Considerations](#) section to record account differences.

40. ☐ **Determine migration execution strategy**

Decide whether to enlist help from Atlassian, a Solution Partner, or migrate on your own.



41. ☐ **Create vendor access**

Create application, server, and database access accounts for any migration vendors.

42. ☐ **Determine timeline**

Use **Worksheet 8: Jira Migration Timeline** in the [Additional Worksheets](#) section to craft a high-level timeline.

43. ☐ **Build a communication plan**

Use **Worksheet 9: Jira Migration Communication Plan** in the [Additional Worksheets](#) section to communicate milestones and plans.

44. ☐ **Make final app and customization decisions**

Use **Worksheet 10: Jira Migration App Decisions** in the [Additional Worksheets](#) section to record a final decision for each app, connection, integration, and customization. Install desired apps in staging and target applications.

45. ☐ **Address and resolve conflicts**

Identify and discuss how to handle conflicts like duplicate project keys, naming collisions, similar schemes, etc.

46. ☐ **Form a migration test team**

Use **Worksheet 14: Jira Migration Test Team** in the [Migration Testing](#) section to create a contact list and determine what each person will test.

47. ☐ **Create data test cases**

Use **Worksheet 15: Jira Migration Configuration Testing** in the [Migration Testing](#) section to verify all configuration elements are as expected.

48. ☐ **Create test accounts**

Use **Worksheet 16: Jira Migration Test Cases** in the [Migration Testing](#) section to record test account details.

49. ☐ **Create functional test cases**

Use **Worksheet 16: Jira Migration Test Cases** in the [Migration Testing](#) section for sample test cases.



Migration Testing

50. ☐ **Create rollback plan**
Create, verify, and document a plan to handle an emergency rollback.
51. ☐ **Develop launch plan**
Reference **Worksheet 12: Jira Migration Sample Launch Plan** in the [Migration Considerations](#) section as an example.
52. ☐ **Refresh environment data**
Update the staging environment configuration and data if needed.
53. ☐ **Select and communicate configuration freeze window**
Use **Worksheet 11: Jira Migration Configuration Freeze** in the [Migration Considerations](#) section to communicate periods when the Jira configuration should not be changed.
54. ☐ **Test launch plan and migration**
Test the entire migration process in a staging environment.
55. ☐ **Update staging settings**
Update the base URL, application links, property files, and any other settings as needed.
56. ☐ **Execute test cases**
Verify that all data was transferred and functions as expected. Check the application logs for any unexpected issues.
57. ☐ **Address issues**
Fix any issues encountered during the test migration process.
58. ☐ **Repeat test migration as necessary**
Do as many trial runs as needed. Don't progress from the staging environment until everything is perfect.



Additional Planning

59. ☐ **Refine launch plan**
Update the launch plan to incorporate findings from the test migration.
60. ☐ **Select migration date**
Use the “Selecting a Launch Window” information in the [Migration Considerations](#) section to select a migration date.
61. ☐ **Communicate launch plan**
Communicate plans to stakeholders and users. Reference **Worksheet 9: Jira Migration Communication Plan** in the [Additional Worksheets](#) section.
62. ☐ **Conduct pre-migration training**
Refer to the sample “Demo Agendas” in the [Pre-Migration Training](#) section for information to include.

During

63. ☐ **Execute launch plan**
Execute the steps in the launch plan previously developed, tested, and refined.

After

64. ☐ **Perform post-migration clean-up**
Address any unneeded or duplicate settings or schemes.
65. ☐ **Decommission source application**
Start the decommission process. Prevent application changes, make data “read only”, redirect visitors, disable login ability, and finally, take the source application offline.
66. ☐ **Maintain staging environment**
Keep the staging environment’s configuration in sync with production.
67. ☐ **Create project templates**
Prevent over-customization by creating templates for development, support, and task projects.



68. ☐ **Revisit disaster recovery and backup strategy**

Make sure the target application is regularly backed up. Use the Configuration Manager for Jira app (botron.io/amp-cm) to take periodic snapshots and quickly deploy the “last known good” configuration.

69. ☐ **Revisit archive strategy**

Periodically archive any unneeded projects and issues in the target application.

70. ☐ **Conduct post-migration training**

Refer to the sample “Demo Agendas” in the [Pre-Migration Training](#) section and add any additional items from the [Post-Migration Training](#) section.

71. ☐ **Mark tasks complete**

Mark Jira tracking tasks complete and update project documentation as needed. Log any outstanding items to address.

72. ☐ **Conduct a retrospective**

Document any issues encountered, what went well, and what to improve for future projects.

73. ☐ **Thank upgrade team**

Recognize the efforts of everyone who participated in the project.



Resources

- Plan your cloud migration - botron.io/atl-server-to-cloud-guide
- Atlassian Data Center migration checklist - botron.io/atl-dc-migrate-checklist

5.2 Migration Preparation

Proper preparation is the *most important* part of the migration process. You need to thoroughly understand the application’s configuration, so you can make decisions and validate that the migration plan works. Don’t rush through preparation (or validation). It’s much cheaper to do the necessary prep work at the beginning than to roll back because you’ve encountered problems or missing data at the end.

Decisions List

During preparation, questions will arise that require additional research, consideration, or decisions. For example, you may discover an app in the source application is not compatible with the target application. How should you handle this? Should you look for a similar app that is compatible? Stop using the incompatible app? Delay the migration until the app is compatible? This decision requires input from different stakeholders.

Use **Worksheet 6: Jira Migration Decisions List** to record items for further discussion.

Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 6: Jira Migration Decisions List



Use this worksheet to collect and record decisions for items that require special consideration, further research, or discussion.

Decisions

ID	Category	Item	Description	Stakeholders	Decision
Ex:	Configuration	Two custom fields named "Support Type"	Custom field ID 10500 is a single select list used by HELP project. Custom field ID 10341 is a text field used by the CS project.	<ul style="list-style-type: none">Help Desk team leadCustomer Service department lead	<p>The selection field is better for reporting. Retain the selection field and remove the text field.</p> <p>Action items:</p> <ul style="list-style-type: none">Review text data and categorize issues with the closest select list option.Add additional select list options, if needed.Add select field to the CS project.Bulk edit CS issues to set select list values.Remove the text field from the CS project.Remove text field from Jira.
Ex:	App	Incompatible app	The app providing custom field X is not available in the target application	TBD	TBD
1					
2					
3					
...					

Configuration Assessment

Now you're ready to examine the source application. By documenting the configuration, you'll understand which schemes, settings, apps, and other important elements exist. Use the documentation to drive discussions, make scope decisions, craft test cases, and make sure all elements are accounted for.

Use **Worksheet 7: Jira Migration Configuration Assessment** to create an inventory of source application configuration elements. Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 7: Jira Migration Configuration Assessment



Enter the source application configuration elements. In the "Migrate" column, specify "Yes" to migrate, "No" to exclude, or "Exists" if the item is already present in the target application.

Application Details

URL: _____

User Count: _____

License Expiration: _____

Projects

Projects

List all projects in the source application. Note any project keys or names which also exist in the *target* application. Visit: Admin > Projects

ID	Key	Project Name	Project Lead	Migrate	Scope
Ex:	DEV	Development	Kelly Smith	Yes	All issues
Ex:	FIN	Finance	Chris Jones	Yes	JQL: project = FIN and (created > startOfYear() or updated > startOfYear())
Ex:	DEMO	Demo	Kelly Smith	No	n/a
Ex:	TEST	Test	Kelly Smith	Exists	n/a
1					
2					
3					
...					



Project Categories

List all project categories in the source application. Visit: Admin > Project categories

ID	Project Category Name	Projects Used By	Migrate
Ex:	Development	DEV, QA, APPS	Yes
Ex:	Test	TEST, TASKS	No
Ex:	Demo	DEMO	Exists
1			
2			
3			
...			

Project Roles

List all project roles in the source application. Visit: Admin > System > Project roles

ID	Project Role Name	Notes	Migrate
Ex:	Administrators	n/a	Exists
1			
2			
3			
...			

Issue Types

Issue Types

List all issue types in the source application. Visit: Admin > Issues > Issue types and Admin > Issues > Sub-tasks

ID	Issue Type Name	Notes	Migrate
Ex:	Bug	n/a	Exists
1			
2			
3			
...			



Issue Type Schemes

List all issue type schemes in the source application. Visit: Admin > Issues > Issue type schemes

ID	Issue Type Scheme Name	Issue Types	Notes	Migrate
Ex:	Default Issue Type Scheme	<ul style="list-style-type: none">• Task• Sub-task• Story (Default)• Bug• Epic	n/a	Exists
1				
2				
3				
...				

Workflows

Workflows

List all workflows in the source application. Visit: Admin > Issues > Workflows

Note: Do not migrate inactive workflows

ID	Workflow Name	Notes	Migrate
1			
2			
3			
...			



Workflow Schemes

List all workflow schemes in the source application. Visit: Admin > Issues > Workflow schemes

Note: Do not migrate inactive workflow schemes

ID	Workflow Scheme Name	Workflow > Issue Type	Usage	Migrate
Ex:	Software Simplified Workflow Scheme	<ul style="list-style-type: none">Approval workflow for Bug > BugSimple workflow for Task > Task	<ul style="list-style-type: none">DevQAApps	Exists
1				
2				
3				
...				

Screens

Screens

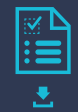
List all screens in the source application. Visit: Admin > Issues > Screens

ID	Screen Name	Notes	Migrate
1			
2			
3			
...			

Screen Schemes

List all screen schemes in the source application. Visit: Admin > Issues > Screen schemes

ID	Screen Scheme Name	Notes	Migrate
1			
2			
3			
...			



Issue Type Screen Schemes

List all issue type screen schemes in the source application. Visit: Admin > Issues > Issue type screen schemes

ID	Issue Type Screen Scheme Name	Notes	Migrate
1			
2			
3			
...			

Fields

Fields

List all custom fields in the source application. Note any conflicting fields which also exist in the target application. Visit: Admin > Issues > Custom fields

ID	Field Name	Notes	Migrate
1			
2			
3			
...			

Field Configurations

List all field configurations in the source application. Visit: Admin > Issues > Field configurations

ID	Field Configuration Name	Notes	Migrate
1			
2			
3			
...			



Field Configuration Schemes

List all field configuration schemes in the source application. Visit: Admin > Issues > Field configuration schemes

ID	Field Configuration Scheme Name	Notes	Migrate
1			
2			
3			
...			

Priorities

Priorities

List all priorities in the source application. Visit: Admin > Issues > Priorities

ID	Priority Name	Notes	Migrate
Ex:	Highest	n/a	Exists
1			
2			
3			
...			

Priority Schemes

List all priority schemes in the source application. Visit: Admin > Issues > Priority Schemes

ID	Priority Scheme Name	Priorities	Notes	Migrate
1				
2				
3				
...				



Issue Links

Issue Links

List all link types in the source application. Visit: Admin > Issues > Issue linking

ID	Issue Link Name	Notes	Migrate
Ex:	<i>Relates</i>	<i>n/a</i>	<i>Exists</i>
1			
2			
3			
...			

Issue Attributes

Statuses

List all statuses in the source application. Visit: Admin > Issues > Statuses

ID	Status Name	Notes	Migrate
Ex:	<i>Closed</i>	<i>n/a</i>	<i>Exists</i>
1			
2			
3			
...			

Resolutions

List all resolutions in the source application. Visit: Admin > Issues > Resolutions

ID	Resolution Name	Notes	Migrate
Ex:	<i>Duplicate</i>	<i>n/a</i>	<i>Exists</i>
1			
2			
3			
...			



Issue Security Schemes

Issue Security Schemes

List all issue security schemes in the source application. Visit: Admin > Issues > Issue security schemes

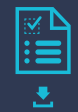
ID	Issue Security Scheme Name	Projects	Notes	Migrate
1				
2				
3				
...				

Notifications

Notification Schemes

List all notification schemes in the source application. Visit: Admin > Issues > Notification schemes

ID	Notification Scheme Name	Projects	Notes	Migrate
1				
2				
3				
...				



Default Notifications

List all notifications, in the default permission scheme, in the source application.

ID	Default Notification Name	Granted To	Notes
Ex:	Issue created	<ul style="list-style-type: none">Current AssigneeReporterAll Watchers	
1	Issue Created		
2	Issue Updated		
3	Issue Assigned		
4	Issue Resolved		
5	Issue Closed		
6	Issue Commented		
7	Issue Comment Edited		
8	Issue Comment Deleted		
9	Issue Reopened		
10	Issue Deleted		
11	Issue Moved		
12	Work Logged On Issue		
13	Work Started On Issue		
14	Work Stopped On Issue		
15	Issue Worklog Updated		
16	Issue Worklog Deleted		
17	Generic Event		



Permission

Permission Schemes

List all permission schemes in the source application. Visit: Admin > Issues > Permission schemes

ID	Permission Scheme Name	Projects	Notes	Migrate
1				
2				
3				
...				

Default Permissions

List all permissions, in the default permission scheme, in the source application.

ID	Default Permission Name	Granted To	Notes
Ex:	Administer Projects	Project role <ul style="list-style-type: none">Administrators Group <ul style="list-style-type: none">jira-administrators	
1	Administer Projects		
2	Browse Projects		
3	Manage Sprints		
4	Service Desk Agent		
5	View Development Tools		
6	View Read-Only Workflow		
7	Assignable User		
8	Assign Issues		
9	Close Issues		
10	Create Issues		
11	Delete Issues		
12	Edit Issues		
13	Link Issues		
14	Modify Reporter		



15	Move Issues		
16	Resolve Issues		
17	Schedule Issues		
18	Set Issue Security		
19	Transition Issues		
20	Manage Watchers		
21	View Voters and Watchers		
22	Add Comments		
23	Delete All Comments		
24	Delete Own Comments		
25	Edit All Comments		
26	Edit Own Comments		
27	Create Attachments		
28	Delete All Attachments		
29	Delete Own Attachments		
30	Delete All Worklogs		
31	Delete Own Worklogs		
32	Edit All Worklogs		
33	Edit Own Worklogs		
34	Work On Issues		



User Objects

Note: Only migrate objects associated with migrated projects listed above. Do not migrate objects with errors (e.g. boards with deleted filters or filters with invalid JQL). Use the Integrity Check for Jira app (botron.io/amp-ic) to detect and address errors.

Filters

List all filters in the source application. Visit: Admin > System > Shared filters

ID	Filter Name	Notes	Migrate
1			
2			
3			
...			

Dashboards

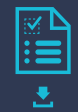
List all dashboards in the source application. Visit: Admin > System > Shared dashboards

ID	Dashboard Name	Filters	Notes	Migrate
Ex:	Development Dashboard	<ul style="list-style-type: none">filter=12234filter=10587	n/a	Exists
1				
2				
3				
...				

Boards

List all boards in the source application. Visit: Boards > View all boards

ID	Board Name	Filter	Notes	Migrate
1				
2				
3				
...				



Customization

Apps

List all apps in the source application. Also see **Worksheet 2: Jira Apps and Customizations** and **Worksheet 10: Jira Migration App Decisions**. Visit: Admin > Manage apps > Manage apps

ID	App Name	Notes	Migrate
1			
2			
3			
...			

Application Links

List all application links in the source application. Visit: Admin > Applications > Application links

ID	Application Link Name	Notes	Migrate
Ex:	Confluence	https://confluence.domain.com	Yes
1			
2			
3			
...			

Connections, Integrations, WebHooks, Listeners, Mail Handlers, Issue Collectors, and other customizations

List all customizations in the source application. Visit: Admin > Applications, Admin > System > WebHooks, Admin > System > Listeners, etc.

ID	Customization Name	Notes	Migrate
1			
2			
3			
...			



User Management

Users

List all users in the source application. Visit: Admin > User management

ID	User Name	Directory	Notes	Migrate
1				
2				
3				
...				

Groups

List all groups in the source application. Visit: Admin > User management > Groups

ID	Group Name	Members	Notes	Migrate
Ex:	<i>jira-administrators</i>	<i>Kelly Smith</i>	<i>n/a</i>	<i>Exists</i>
1				
2				
3				
...				

Global Permissions

List all global permissions in the source application. Visit: Admin > System > Global permissions

ID	Global Permission Name	Granted To	Notes	Migrate
Ex:	<i>Jira System Administrators</i>	<i>Group:</i> • <i>jira-administrators</i>	<i>n/a</i>	<i>Exists</i>
1	Jira System Administrators			
2	Jira Administrators			
3	Browse Users			
4	Create Shared Objects			
5	Manage Group Filter Subscriptions			
6	Bulk Change			
7	Browse Archive			

Comparison

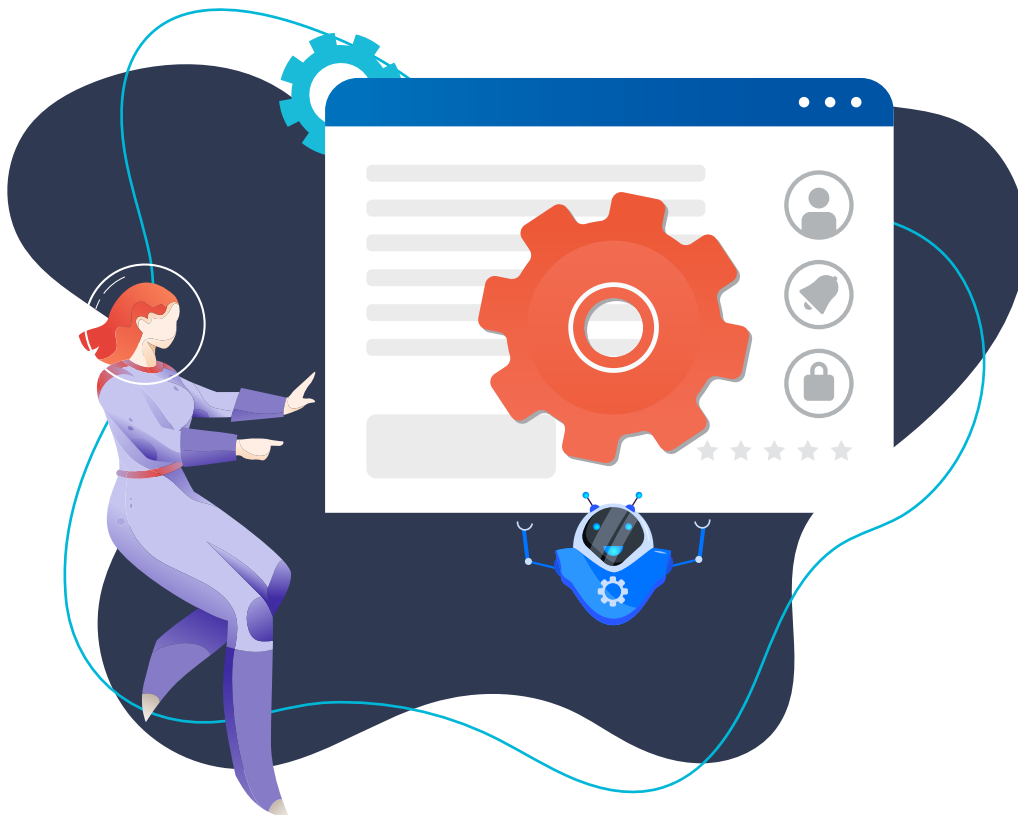
Are you consolidating applications? Does the target application have its own data and configuration? If so, conduct the above assessment process for the target application, too. Then, compare both results. Specifically look for conflicts and collisions.

For example, the source application has older priority names like “Blocker”, “Critical”, and “Major”. The target application has newer priority names like “Highest”, “High”, and “Medium”. How will you handle this situation? You’ll need to consolidate the naming strategy or agree to support both strategies.

Look for conflicting project names, project keys, custom fields, etc. If you find collisions, add them to **Worksheet 6: Jira Migration Decisions List** in the previous section.



Tip: Take screenshots of the administration pages in the source, staging, and target applications (e.g., the list of statuses on the Admin > Issues > Status page). These will be useful when comparing before and after states.



Back Up

Before you make major changes, always back up the file system, database, and attachment directory. After, test the restore process in a staging environment to verify that the backup files are corruption-free and usable.

If the backup procedure is automated, periodically ensure that the most recent backup occurred as expected.

A database and file system backup is better than Jira's XML backup function, especially for large applications.



Tip: Consider taking multiple backups during the migration process for partial roll backs or comparison purposes. Example: Take a backup before validation, make any needed fixes, then take another backup.

Rollback

Practice the restore procedure and determine how to handle an emergency roll back situation. The best time to plan for a roll back is before you need it!

Determine scenarios where you'd stop the migration process and roll back or where you'd continue and troubleshoot the problem after. For example, if there's a permission problem, it might be faster to fix the permission schemes *after* the migration, rather than roll back and start over.

Roll backs most commonly occur because important data is missed, not because the migration copy process failed.



Resources

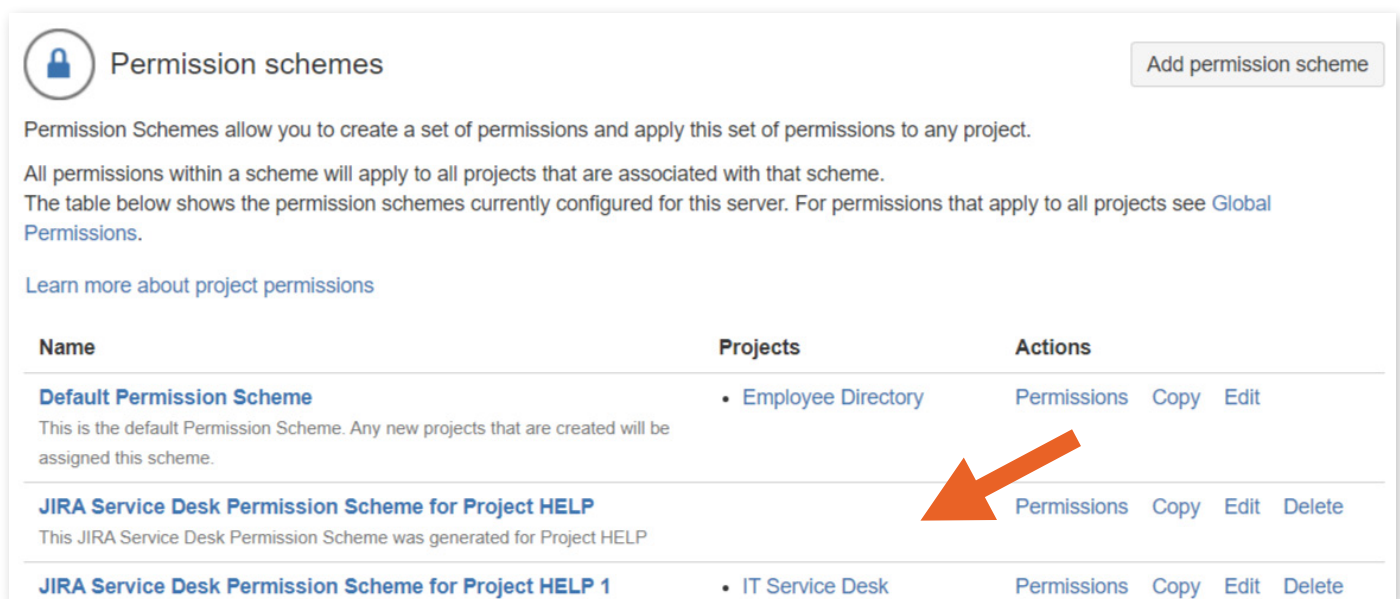
- Backing up data - botron.io/atl-data-backup
- Restoring data - botron.io/atl-restore-data
- Restoring a project from backup - botron.io/atl-project-backup-restore

Clean Up

Any clean-up you can do now on the source application will make the migration easier. Did you know that the number of custom fields, email notifications, apps, issue security levels, and workflow step executions impact performance? Decrease the migration scope by consolidating duplicate settings, consolidating custom fields that serve a similar purpose, removing unused schemes, and shortening long workflows. You'll speed up the application, too!

For example, if there are multiple final workflow statuses (e.g., "Done", "Closed", "Completed", etc.), can they be reduced to one final status?

At a minimum, visit all the administration areas and look for unneeded or unassociated schemes to remove. Here's an easy example using the *Permission schemes* administration page.



Permission schemes Add permission scheme

Permission Schemes allow you to create a set of permissions and apply this set of permissions to any project.

All permissions within a scheme will apply to all projects that are associated with that scheme.

The table below shows the permission schemes currently configured for this server. For permissions that apply to all projects see [Global Permissions](#).

[Learn more about project permissions](#)

Name	Projects	Actions
Default Permission Scheme This is the default Permission Scheme. Any new projects that are created will be assigned this scheme.	• Employee Directory	Permissions Copy Edit
JIRA Service Desk Permission Scheme for Project HELP This JIRA Service Desk Permission Scheme was generated for Project HELP		Permissions Copy Edit Delete
JIRA Service Desk Permission Scheme for Project HELP 1	• IT Service Desk	Permissions Copy Edit Delete

Image 4: Permission Schemes Admin Page

In the screenshot, the second scheme is used by no projects. Remove it so you don't have to migrate it!

Clean up is a mostly manual and time-consuming process, but it's worth it for migration preparation and for a healthy application. Use the Power Admin app, by Botron, to gauge usage of other settings like custom fields and workflows. For example, Jira's *Custom fields* administration page shows which screens fields are used on, but not whether the screen is used. The Power Admin app, however, shows all the projects that use the field.

See botron.io/amp-pa for more about this app.

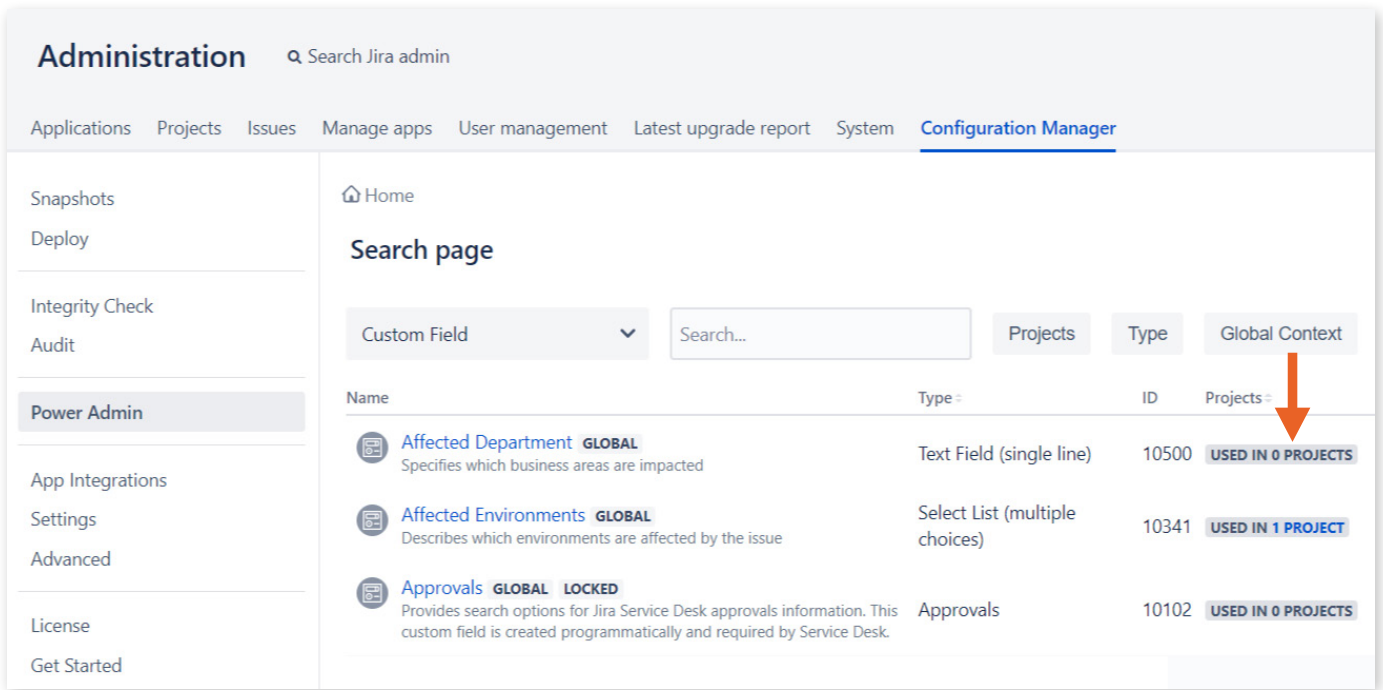


Image 5: Power Admin App

In the screenshot, the “Affected Department” field is not used by any projects. If the field is unused, and doesn’t contain any needed data, remove it! If the field contains needed data, add it to the decisions list.

Archiving Projects

If there are projects you no longer need, then archive, hide, or remove them before the migration.

Jira Cloud Premium and Jira Data Center

Jira Cloud Premium and Jira Data Center have a native project archival function. Use it to remove projects from view, remove issues from the index, and preserve the data in case you need it later.

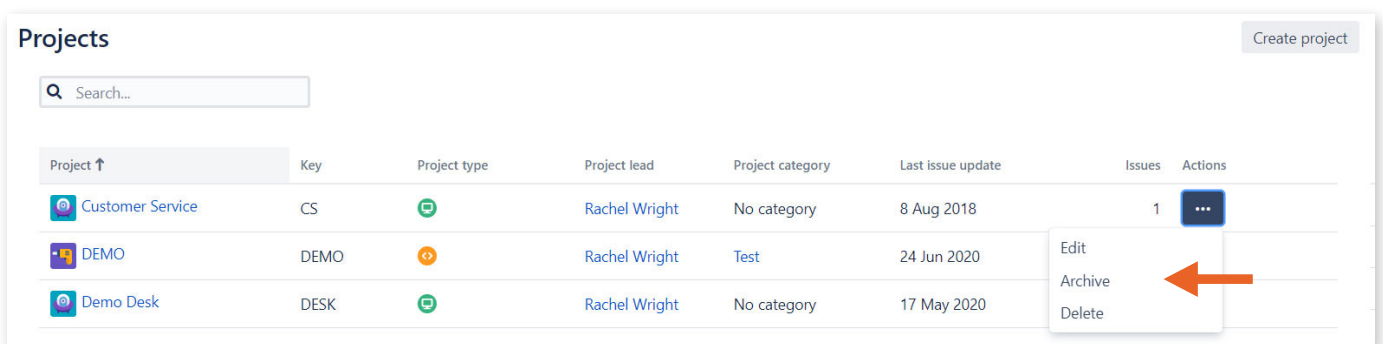


Image 6: Jira Archive Feature

Access the archive feature from the *Projects* administration page. Click the ellipses icon and select the “Archive” option shown in the screenshot.

The project will immediately be hidden from view and moved to a new page called *Archived projects*. Afterwards, do a re-index to remove the issues from the index. Re-indexes can impact performance, so always do them outside of peak hours.

Archived projects are not visible to users. Archived issues are not returned in searches or JQL queries. Other than hiding projects, the archive feature doesn't change a project's configuration. If a project's schemes are no longer needed, remove them manually.



Resources

- Archiving a project - botron.io/atlassian-project-archive

Jira Server and Jira Cloud

There's no archive feature in Jira Server or Jira Cloud, but you can still remove data or make it less visible. Develop your own “archive” procedure for any unneeded projects. Here's what I do:

- Prepend “x ARCHIVED: ” to the project's name. The user sees this information in the project list and in their queries. For application administrators, all the archived projects are listed together, alphabetically, in selection lists.
- Create a category called “x ARCHIVED”. Add all unneeded projects to this category. Users can include or exclude projects with a JQL query like `category = “x ARCHIVED”`.
- Use a unique project icon for all “archived” projects.
- Add an explanation to the project's description field (e.g., “This project is archived. Project data is still available for legal and historical reasons, but no issues can be created or edited. Please create new issues in the X project instead.”)
- Restrict the issue edit and transition ability.
- Restrict the move ability, if appropriate.
- Remove the unneeded notification scheme.

None of these steps improve performance, but they are good maintenance procedures and can improve usability.

If there's data that's no longer needed in Jira, back it up, export it (if desired), and delete it from the application.

Health and Performance Tests

Before moving data, make sure both the source and target applications are healthy and error free. Run the Jira Server and Jira Data Center tools below and fix any problems reported.

Instance Health

Use the instance health check function, which is part of the built-in Troubleshooting and Support Tools app. This checks for common problems with the file system, attachments, index, database, etc. To access this tool, visit: Admin > System > Troubleshooting and support tools.

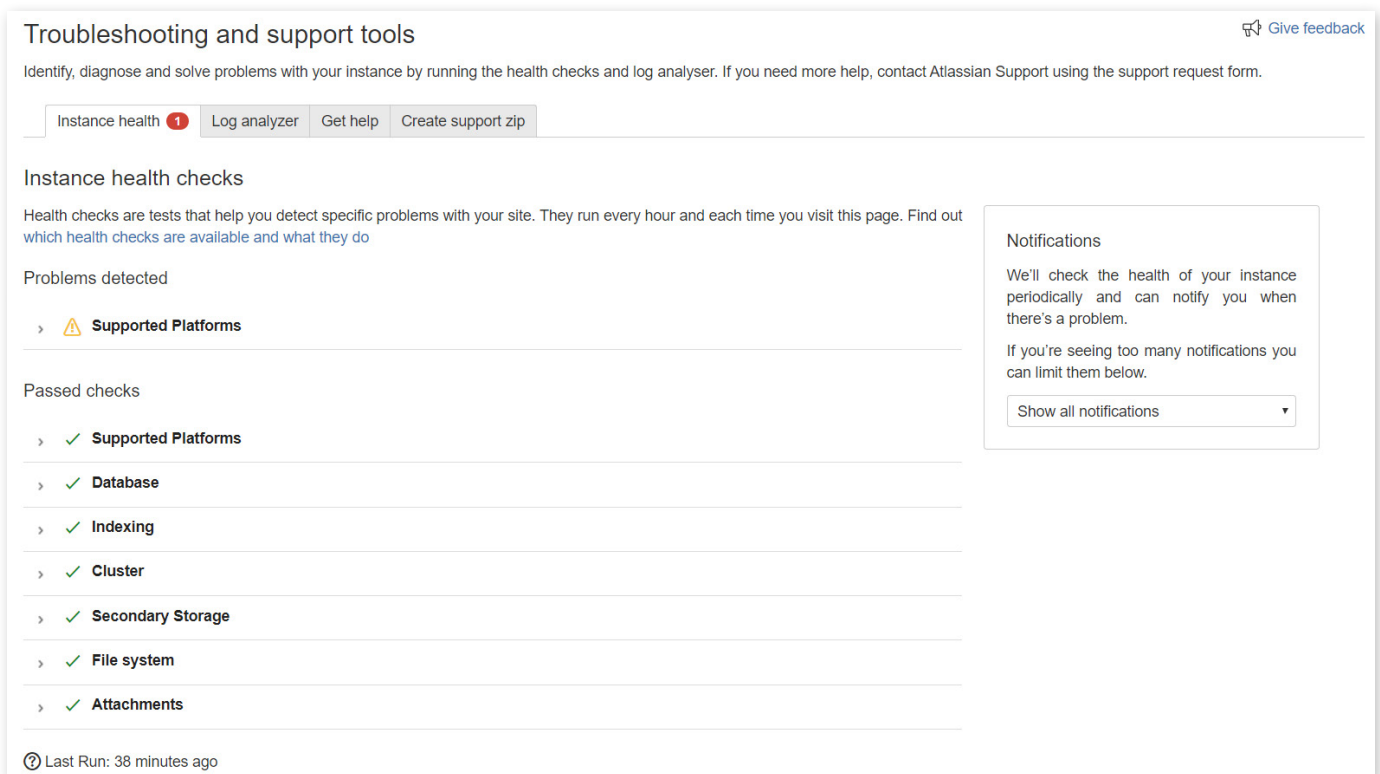


Image 7: Instance Health Checks

The screenshot above shows that a problem was detected. Click on the problem to see additional details and clues for resolving the issue.

Integrity Checker

Next, run Jira's built-in Integrity Checker feature. This checks for a variety of issue, search, permission, and workflow problems. The tests can take a while to complete, but you can exclude some of them or run them in batches when appropriate. To access this tool, visit: Admin > System > Integrity checker.

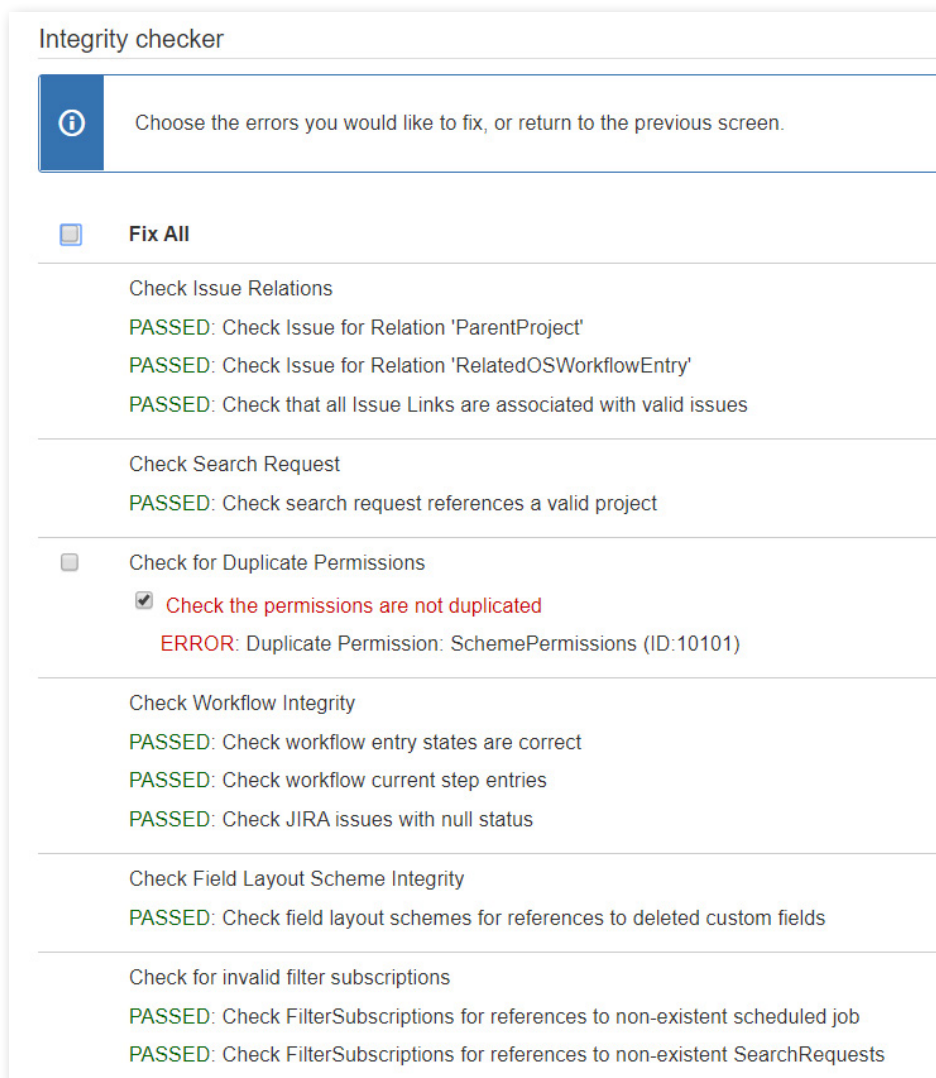


Image 8: Integrity Checker

The screenshot above shows that a problem was detected. This specific error was caused by a missing permission scheme. You can either correct the problem manually or have Jira try to fix it for you.

Integrity Check for Jira App

Finally, download the free Integrity Check for Jira app from the Atlassian Marketplace. The Integrity Check function is also included with the Configuration Manager for Jira app. After installing the app, access it from: Admin > Configuration Manager > Integrity Check.

See botron.io/amp-ic and botron.io/amp-cmj for more about these apps.

This app helps you detect, locate, and fix errors in user objects like filters, boards, and dashboards. Fix problems now so you don't migrate broken objects!

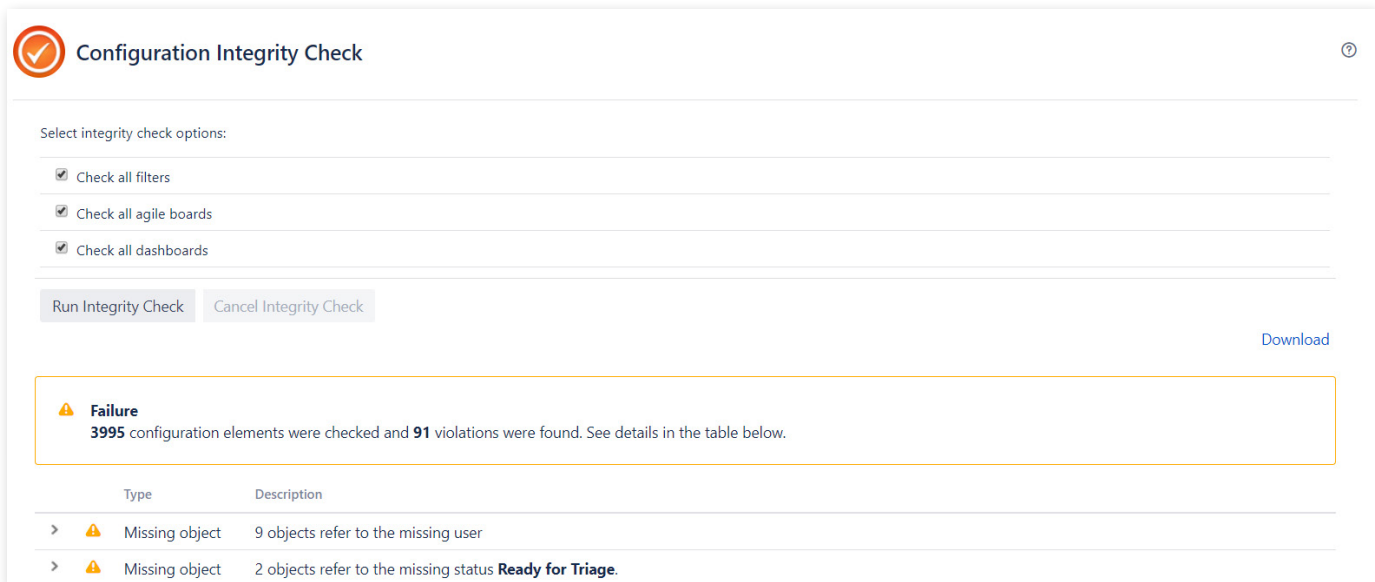


Image 9: Integrity Check for Jira App

The screenshot above shows 91 instances of missing or duplicate settings. I love this tool because it tells you exactly *what* is wrong and *where* to address the problem. There's no guessing or hunting around.

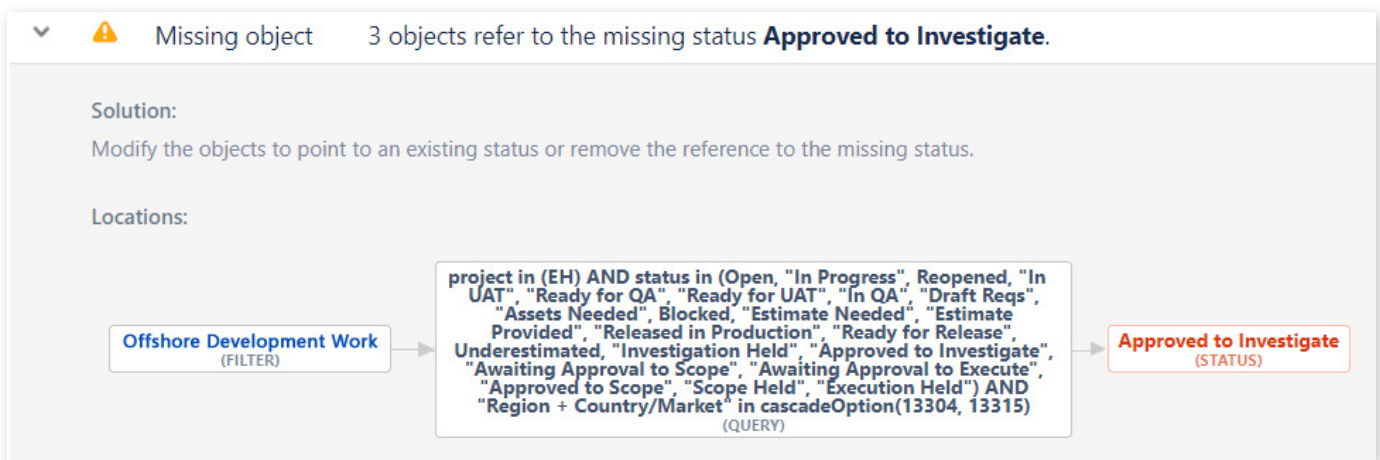


Image 10: Missing Object Detected by Integrity Check for Jira App

For example, when there's a broken JQL filter, the tool shows: the name of the filter, the name of the filter's owner, the specific JQL, and the name of the missing status, field, or selection. This makes it easy to fix the filter yourself or alert the filter's owner.

Additional health checks are available for Atlassian Premier Support customers. Read more: botron.io/atl-premier-support

Performance Tests

Next, make sure both applications have the needed processing power for large data imports and exports. Read more: botron.io/atl-jira-performance-test-tools

Also look for potentially problematic behaviors. For example, is another application heavily using the REST API? Do many users try to run reports at the same time? Are scripts constantly writing to the database? Are there intensive background processes (e.g., backups) or frequent services (e.g., email notifications) to consider or work around?

If you're experiencing any performance issues in the source application, address them now, before migration.

Re-index Test

How long does it take to re-index the source data? Test this now and record the result as a baseline. Also consider how long a re-index might take when combining source data with target data. Factor the duration into the migration schedule. If the re-index takes many hours (or days) consider archival and/or clean-up to reduce the index's size. To access this tool, visit: Admin > System > Indexing.

Note: Re-indexes can impact performance. Always do them outside of peak hours.

Re-indexing

✓

Re-indexing was **successful** and took **262 seconds**.

The index is an essential part of Jira and affects most of its functions. Re-indexing keeps the index in sync with any configuration changes, making new information or data fields searchable. [Learn more](#)

Options

☒ Background re-index

Keep your current index, and update it in the background.

☐ Full re-index

Delete your index, and recreate it from scratch. This way of re-indexing is faster and more thorough, but Jira will be unavailable to your users until re-indexing is complete.

Index Path

`/var/atlassian/application-data/jira/caches/indexesV1`

Re-index

Image 11: Re-index Result

5.3 Helper Apps

The Atlassian Marketplace contains apps to help plan and execute a migration. Apps are safer than scripting and easier to use than manually moving data. Consider the following options:

Jira Cloud Migration Assistant

The Jira Cloud Migration Assistant is a free Atlassian app to move projects, users, and groups from Jira Server or Data Center to Jira Cloud. It's a wizard used to connect to the Cloud target application, assess apps, select projects to migrate, and check for errors. After the initial steps are complete, you can execute the migration immediately or save the selections for later. The Jira Cloud Migration Assistant works well for partial or complete migrations and *will not* overwrite Cloud target application data.

Install the Jira Cloud Migration Assistant from the *Find new apps* administration page in Jira. The “Migrate to cloud” link is on the *System* administration page, under the “Import and Export” sidebar heading.

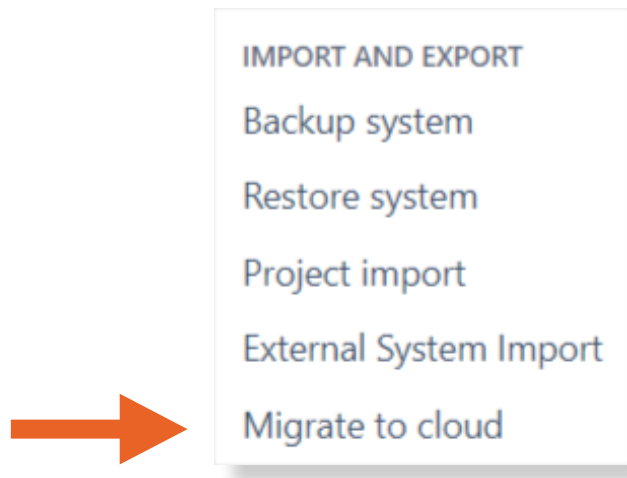


Image 12: Admin > System > Import and Export Menu

This tool includes a handy app assessment feature. It lists apps installed, their Cloud compatibility, vendor documentation links, and records decisions and notes for each.



Tip: If an app you use is listed as unavailable for Jira Cloud, don't stop there. Ask the app vendor for more information. The Cloud version could be listed under a different name or it may currently be in development.

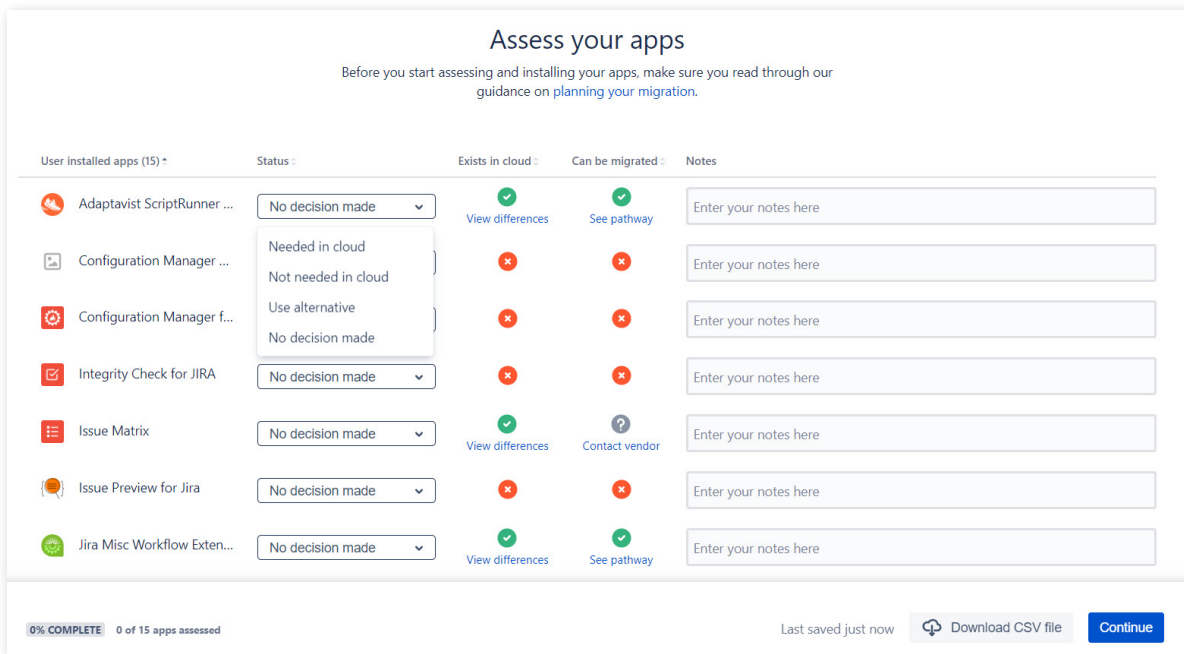


Image 13: Jira Cloud Migration Assistant

The Jira Cloud Migration Assistant is still in development. Jira Service Management projects and some custom fields are not supported yet. For Cloud to Cloud migrations, Next-gen projects are not supported either. See the list of supported items at: botron.io/atl-jira-cloud-migration-assistant-what

For usage instructions see: botron.io/atl-jira-cloud-migration-assistant

Jira Site Import

The Jira Site Import tool supports all Jira project types, including Service Management projects. This method is best for moving all data (you can't choose specific projects) into a brand-new Cloud target application. The import process *will overwrite* existing Cloud data.

Start by exporting the data into a portable XML format. Use the "Backup system" link on the *System* administration page, under the "Import and Export" sidebar heading.

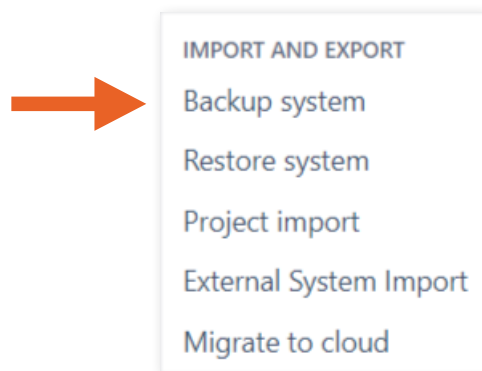


Image 14: Admin > System > Import and Export Menu

You'll need to manually export attachments, project avatars, and logos. Create a ZIP file using these instructions: botron.io/atl-jira-site-import



Tips:

- The export process may take a long time. It may appear to freeze in the browser but continue to run in the background. Allow ample time for the entire process to complete.
- The export file is not downloadable from the browser. Access the server's file system to retrieve it.

Once all XML and ZIP files are generated, log in to the Cloud target application. Use the "Restore system" link on the *System* administration page to import data and media files.

CSV Import

The CSV import method is the easiest but also the least robust. This option only imports selected issue and field data. It does not include project settings, users, attachments, issue history, issue ranking, sprints, versions, or Confluence links.

Jira Data Center Migration App

Atlassian created a way to migrate safely and conveniently, from Server or Service Management to Data Center, for specific scenarios. You can use their free migration tool if your Jira version is supported, your database is PostgreSQL, your server type is Linux, and the target environment is Amazon Web Services (AWS). Of course, you'll also need a Data Center license and an AWS account.

The Jira Data Center Migration app deploys the Data Center infrastructure on AWS. It uses Amazon's "Quick Start" template system. The recommended "Quick Start" architecture includes one initial application instance, a load balancer, the file system, the database, and application monitoring.

Next, the Jira Data Center Migration app migrates the source application's home directory and database records. It does not migrate app data, however. You'll need a different strategy for apps and integrations.

After installing the Jira Data Center Migration app, access it from: Admin > System > Data Center Migration.

Using Azure? There's no migration tool, but there is a similar "Quick Start" deployment template. See: botron.io/atl-dc-azure and botron.io/atl-aws-azure



Resources

- Jira Data Center Migration App - botron.io/amp-dc-migration-app
- How to use the Data Center Migration app to migrate Jira to an AWS cluster - botron.io/atl-dc-migration-app-aws-cluster
- Getting started with Jira Data Center on AWS - botron.io/atl-jira-dc-aws-start

Configuration Manager for Jira

What if you're not moving to AWS, there's app data to migrate, or you need to transform data to handle migration conflicts? What if you need to consolidate multiple applications or simply move data between environments? Then you need the Configuration Manager for Jira app by Botron. See botron.io/amp-cmj.

Configuration Manager for Jira can be used to migrate single projects, multiple projects, or an entire application. This tool addresses the major migration pain points, supports more configuration elements than any other tool, and can migrate data types that other apps cannot. For example, use Configuration Manager for Jira to migrate issue ranking in boards, Confluence links, and even app data from some of the most popular apps in the Atlassian Marketplace. It also supports Jira Service Management.

See the full list of supported elements at: botron.io/CMJ-config-objects

See the full list of integrated apps at: botron.io/CMJ-apps-list

Configuration Manager for Jira also moves data faster, which is especially important for large instances. While other tools rely on Atlassian's XML export feature, which takes a long time to get and parse data, Botron wrote their own extraction and transformation process. As a result, you can move millions of issues in hours instead of days. Additionally, Jira's built-in XML export cannot export partial data and may overwrite existing data in the target application. Configuration Manager for Jira migrates all (or selected) projects and never overwrites existing data.

Botron created this tool in 2013 to help them with their own migration tasks. Since then, it's evolved and improved to cover more use cases, migrate more types of data, handle more issues, provide more visibility, and detect and manage conflicts. For example, Configuration Manager for Jira shows exactly what conflicts exist, so you can make decisions and assess their impact. You can also transform data to avoid collisions.

Configuration Manager for Jira improvements are added constantly as top Atlassian Solution Partners use the tool and provide feedback for additional app features and support. They are currently working on new enhancements like merging custom fields and merging user accounts. You cannot do any of these things today in Jira, but soon, you'll be able to do them using Configuration Manager for Jira!

One of the top reasons I use this tool for migrations is that it greatly reduces the risk of data loss. The app detects and prevents configuration problems and helps you fix them, so you can migrate from one application to another with zero data loss. It shows exactly what you've modified with a detailed change analysis feature. You can be confident that, once migrated to the target application, the configuration will work exactly the way it does in the source application.

Configuration Manager for Jira was built to minimize the need to roll back, but has its own roll back mechanism in case deployment errors are encountered. If the migration fails, the app fully preserves the target application's integrity and leaves the configuration in its pre-migration state. Read more: botron.io/CMJ-rollback-strategy

Finally, for the security conscious, you'll be glad to know that any data migrated with Configuration Manager for Jira stays within the application. The app does not send any data to the Internet.

Key Migration Features

Here are some ways the Configuration Manager for Jira app helps with migrations and merges:

- **Configuration Integrity**

The Integrity Check feature helps find and fix configuration errors. Read more: botron.io/CMJ-integrity-check

- **Dependency Analysis**

The app performs a detailed analysis of all third-party app dependencies and versions. This way you'll know exactly what it can automatically migrate.

- **Change Analysis**

The app conducts a deep change and impact analysis of all added, updated, or removed objects. You can be sure that only expected and desired changes are made to the target application. Read more: botron.io/CMJ-snapshot-change-analysis

- **Impact Analysis**

Configuration Manager for Jira helps you understand how configuration changes impact projects and configurations in the target application.

- **Deployment Modes**

The tool can migrate or merge single projects, multiple projects, or entire applications.

- **Data Loss Prevention**

If errors are encountered during the deployment, all previously applied changes are rolled back, and the Jira configuration is restored to its initial state.

- **Audit Log**

All changes made during the deployment process are logged so you can see exactly what actions occurred and when. Read more: botron.io/CMJ-audit-logging

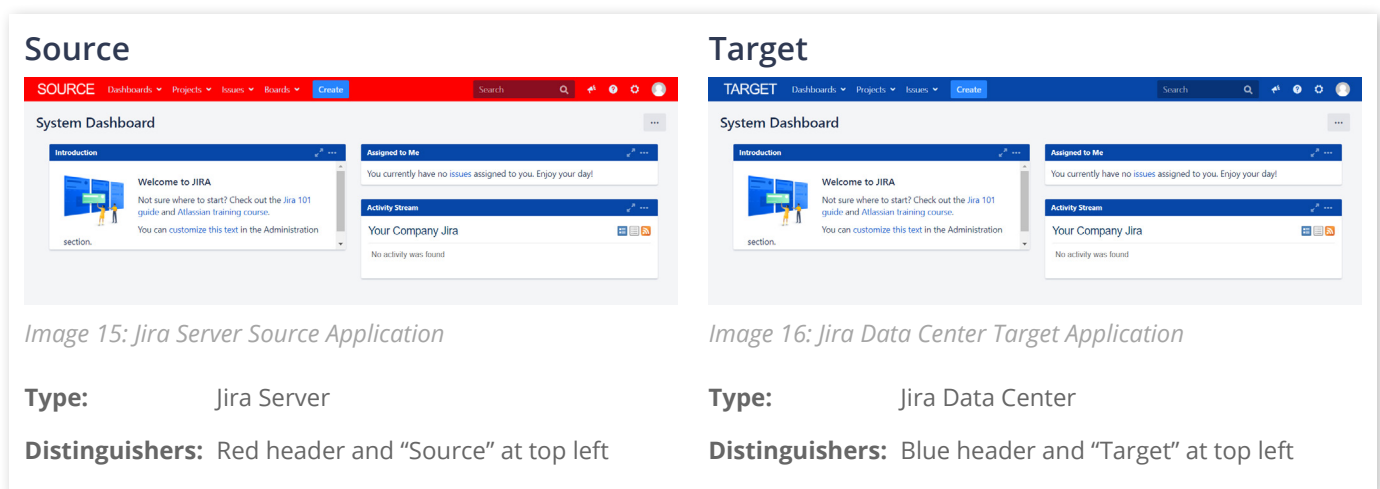
- **Third-party App Support**

Many of the most popular apps are integrated with Configuration Manager for Jira, so app settings and data can be migrated quickly and easily.

5.4 Example Migration Walkthrough using Configuration Manager for Jira

By now you've determined your migration path, assembled a migration team, obtained approval, set up a staging application, and completed most planning steps. You may be all the way to item #54 in **Worksheet 5: Jira Migration Checklist**. It's a good time to start testing the migration process.

As an example project, we'll migrate an entire application from Jira Server (source) to Jira Data Center (target) using Configuration Manager for Jira. The two applications use different colors and logo images to distinguish between them.



Here are the migration steps:

Step 1: Install App and Run Initial Checks

Install Configuration Manager for Jira in Source Application

First, install the Configuration Manager for Jira app from botron.io/amp-cmj. You'll need the following:

- Access: The "Jira System Administrators" global permission (to install the app)

- Environment: A supported version of Jira Server or Jira Data Center. Read more: botron.io/amp-cmj-versions

Log in to Jira as a system administrator, click the “cog” or “gear” icon on the top right, and click the “Manage apps” selection. On the *Find new apps* administration page, search for “Configuration Manager for Jira”. Install the app from the list below the search box. There’s a Server app version and a Data Center app version. By installing the app within Jira, the correct version is automatically selected.

Apply a free trial or paid license on the *Manage apps* administration page. See step-by-step installation instructions at: botron.io/CMJ-install-upgrade

See licensing information at: botron.io/CMJ-license



Note: Configuration Manager for Jira also includes the Integrity Check for Jira and Power Admin apps. You’ll see those installed on the Admin > Manage apps > Manage apps page.

Locate Configuration Manager for Jira

Access the Configuration Manager for Jira app from the administration area’s header or the administration “cog” or “gear” menu.

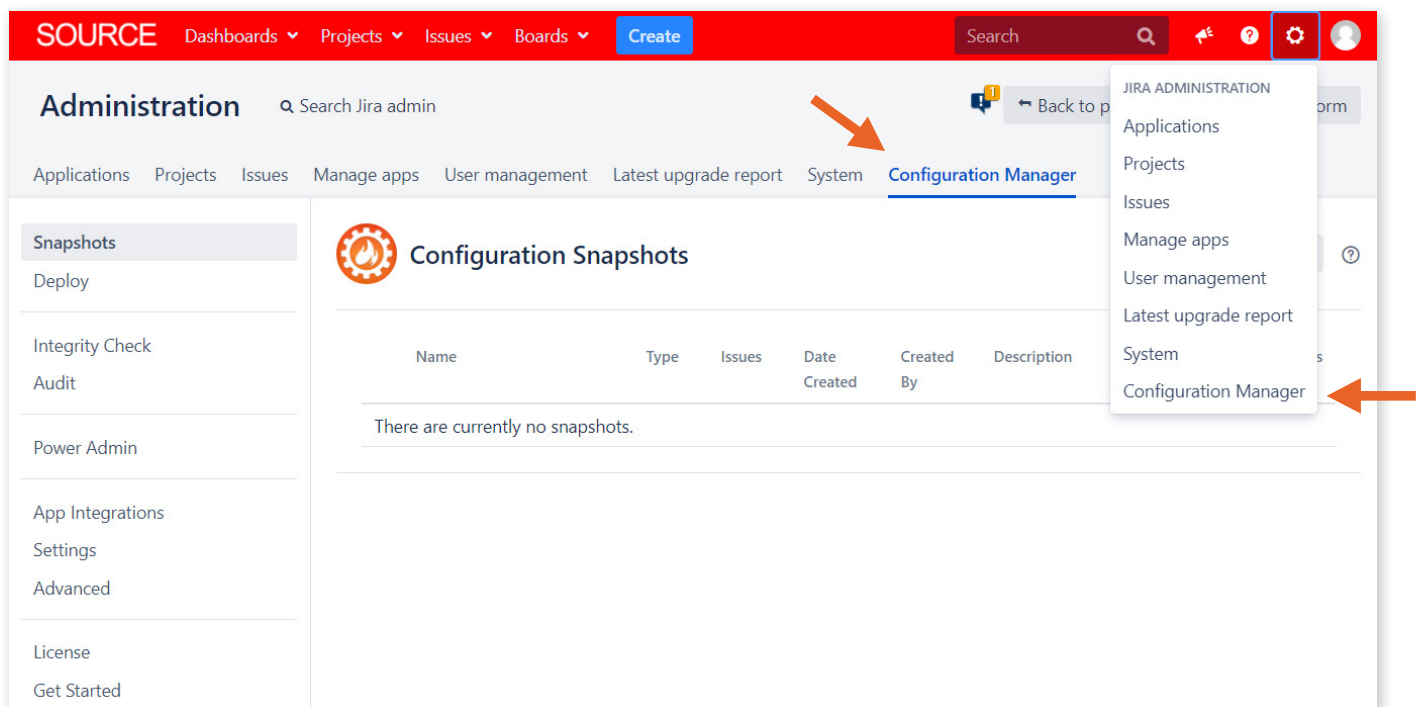


Image 17: Ways to Access Configuration Manager for Jira



Check Integrity

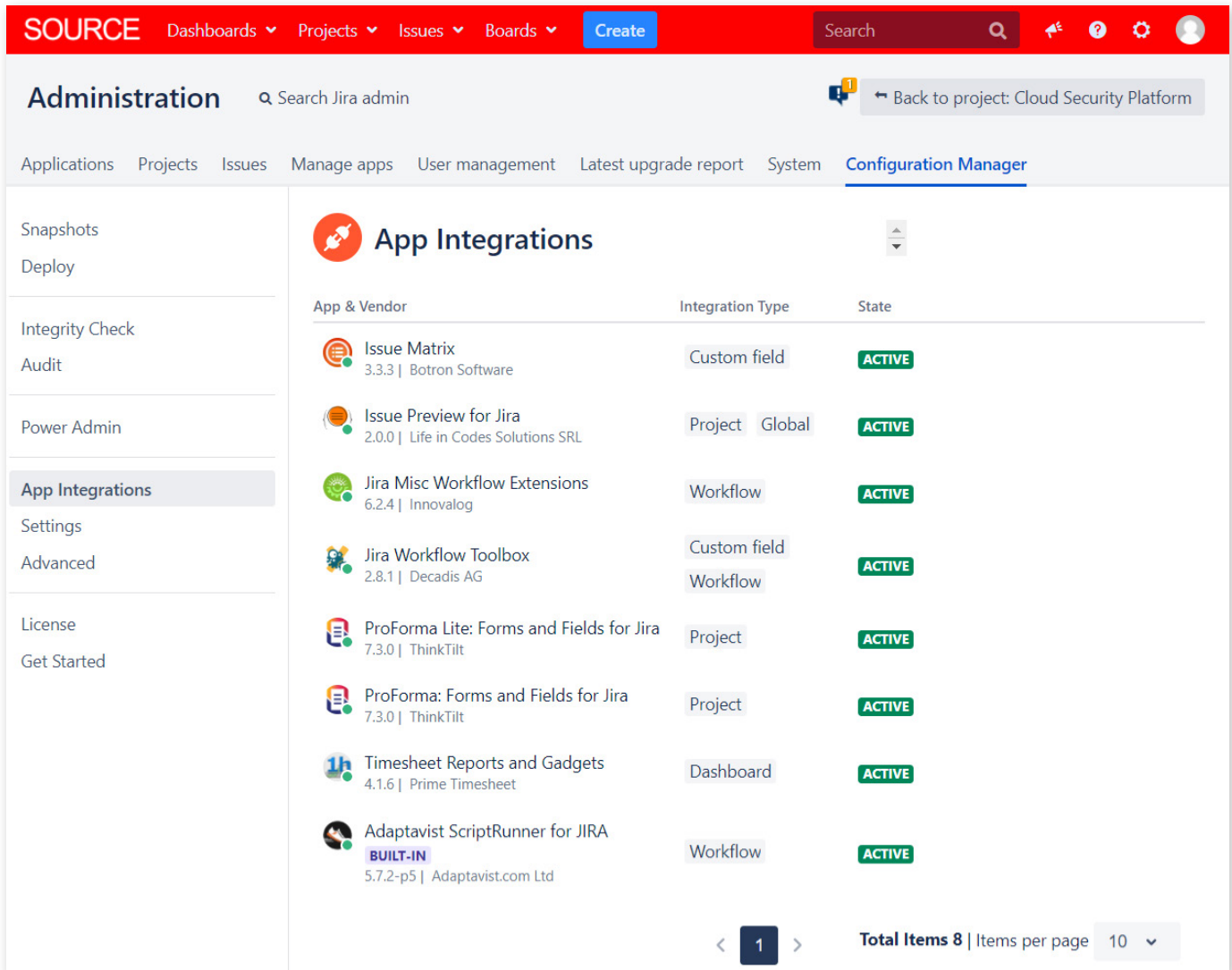
Hopefully, you've already used Jira's built-in Instance Health and Integrity Checker tools and addressed any errors. Now run Configuration Manager for Jira's Integrity Check function as well. Click the "Integrity Check" link in the left sidebar and fix any errors before moving on to the next step. See the previous [Health and Performance Tests](#) section for additional details.

The screenshot shows the Jira Administration interface. At the top, there's a red header with "SOURCE" and navigation links like "Dashboards", "Projects", "Issues", "Boards", and a "Create" button. Below this is a search bar and a "Back to project: Cloud Security Platform" link. The main navigation bar includes "Applications", "Projects", "Issues", "Manage apps", "User management", "Latest upgrade report", "System", and "Configuration Manager". The left sidebar lists various administration options: "Snapshots", "Deploy", "Integrity Check" (highlighted), "Audit", "Power Admin", "App Integrations", "Settings", "Advanced", "License", and "Get Started". The main content area is titled "Configuration Integrity Check" with a success icon. It shows "Select integrity check options:" with three checked items: "Check all filters", "Check all agile boards", and "Check all dashboards". Below these are buttons for "Run Integrity Check" and "Cancel Integrity Check". A green success message states: "Success 375 configuration elements were checked and no violations were found."









Image 18: Integrity Check Results

Review App Integrations

Next, click the “App Integrations” link on the sidebar to see the state of installed apps.



The screenshot shows the Jira Administration interface. The top navigation bar includes 'SOURCE', 'Dashboards', 'Projects', 'Issues', 'Boards', and a 'Create' button. The 'Administration' section is active, with a search bar and a 'Back to project: Cloud Security Platform' link. The left sidebar contains links for 'Snapshots', 'Deploy', 'Integrity Check', 'Audit', 'Power Admin', 'App Integrations' (highlighted), 'Settings', 'Advanced', 'License', and 'Get Started'. The main content area displays the 'App Integrations' page with a table of installed apps.

App & Vendor	Integration Type	State
 Issue Matrix 3.3.3 Botron Software	Custom field	ACTIVE
 Issue Preview for Jira 2.0.0 Life in Codes Solutions SRL	Project Global	ACTIVE
 Jira Misc Workflow Extensions 6.2.4 Innovalog	Workflow	ACTIVE
 Jira Workflow Toolbox 2.8.1 Decadis AG	Custom field Workflow	ACTIVE
 ProForma Lite: Forms and Fields for Jira 7.3.0 ThinkTilt	Project	ACTIVE
 ProForma: Forms and Fields for Jira 7.3.0 ThinkTilt	Project	ACTIVE
 Timesheet Reports and Gadgets 4.1.6 Prime Timesheet	Dashboard	ACTIVE
 Adaptavist ScriptRunner for JIRA BUILT-IN 5.7.2-p5 Adaptavist.com Ltd	Workflow	ACTIVE

At the bottom of the table, there is a pagination control showing '< 1 >' and a summary 'Total Items 8 | Items per page 10'.

Image 19: Configuration Manager for Jira App Integration Result

In the screenshot above, all installed apps are “Active”, meaning the app configuration will be automatically migrated. See the [Apps Migration](#) section below for additional information.

Apps shown as “Inactive” are either disabled or require a newer version of the Configuration Manager for Jira app.






App Integrations		
App & Vendor	Integration Type	State
 Issue Matrix 3.3.3 Botron Software	Custom field	ACTIVE
 Issue Preview for Jira 2.0.0 Life in Codes Solutions SRL	Project Global	ACTIVE
 Jira Misc Workflow Extensions 6.2.4 Innovalog	Workflow	ACTIVE
 Jira Workflow Toolbox 2.8.1 Decadis AG	N/A	INACTIVE  Jira Workflow Toolbox is disabled. Enable the app and check the integration again. Enable

Image 20: Configuration Manager for Jira App Integration Results

In the screenshot above, the Jira Workflow Toolbox app is disabled in Jira. Enabling it will change its “App Integration” status from “Inactive” to “Active”. Apps that are not integrated with Configuration Manager for Jira are unlisted, and the data and settings for those apps will not be migrated.

During the deployment process, if a source app has a different version than a target app, Configuration Manager for Jira will display a warning. If an app is missing on the target application, the deployment will stop. Simply install the missing app and continue the deployment process.

Step 2: Create Snapshot

After you’ve run Configuration Manager for Jira’s Integrity Check and know which app configurations can migrate, it’s time to create a snapshot. A snapshot is a package of configuration information.

Snapshot Types

- **System Configuration** – Includes all projects and all project elements, but no issues. This is useful for quickly matching staging application settings to production, for example.
- **Project Configuration** – Includes a single project or multiple projects and their configuration elements like schemes, workflows, and settings. Does not contain issues.
- **Project with Issues** – Includes a single project or multiple projects and their configuration elements like schemes, workflows, and settings. Also contains issues.

Click the “Create Snapshot” button on the top right of the *Configuration Snapshots* page.

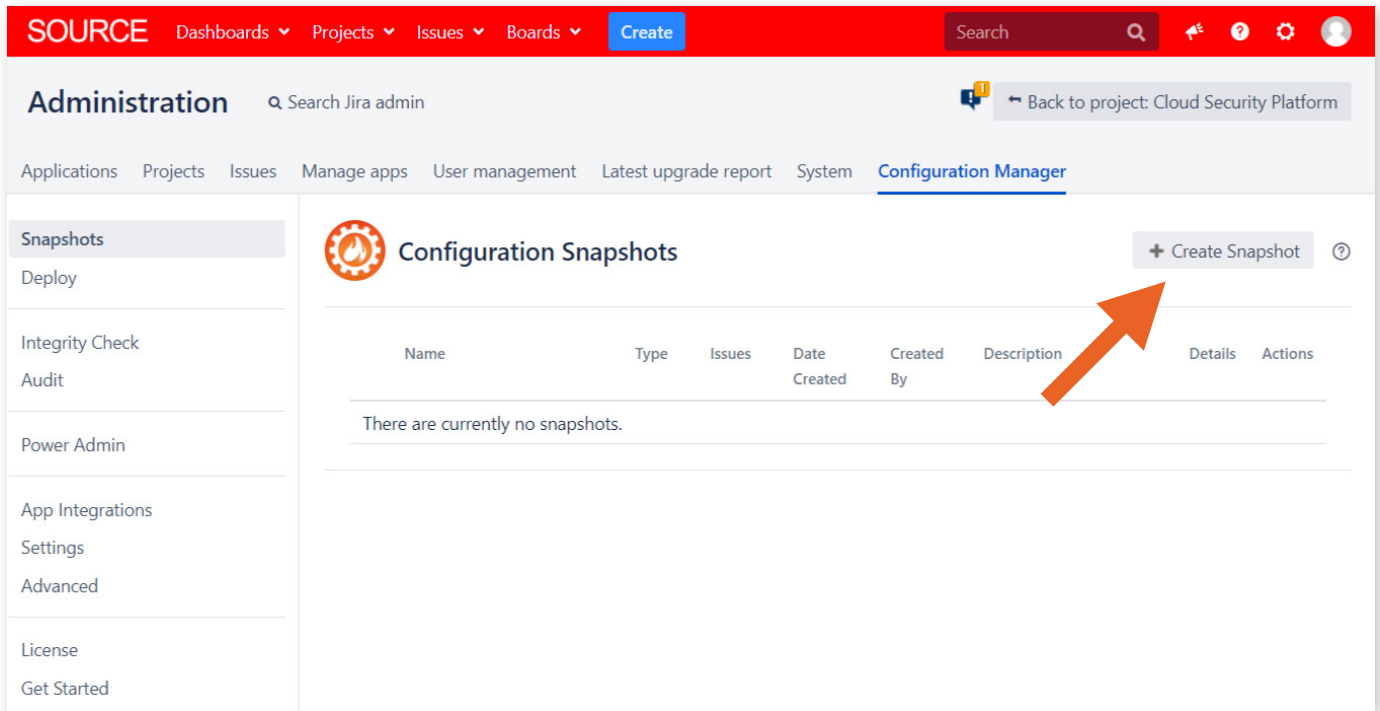


Image 21: Configuration Manager for Jira Snapshots Page

Here’s how to create a “Project with Issues” snapshot for this book’s migration use case.

In the “Create Configuration Snapshot” wizard:

1. Select “Project with Issues” as the snapshot type.
2. Enter a snapshot name.
3. Select all projects.
4. Enter a snapshot description.
5. Select any objects to include and click the form submission button.

SOURCE Dashboards ▾ Projects ▾ Issues ▾ Boards ▾ **Create** Search 🔍 🔊 ? ⚙️ 👤

Create Configuration Snapshot

Select Filters Boards Dashboards Apps Preview Created

Select Snapshot Type System Configuration Project Configuration **Project with Issues** ⓘ

i Package the **Configuration and Issues** of the selected Project(s) into a **Project snapshot with Issues**.

Name* Source

Project(s)* IT Requests × Marketing × Cloud Security Platform × Issue Matrix Demo × ▾

☐ Include attachment files

Description This snapshot is for all projects and issues in the source application at: https://jira.domain.com.

☒ Include Project Filters

☒ Include Project Boards

* This snapshot is used to move Project Configuration and Issues to a target system.

Next **Create** Cancel

Atlassian Jira [Project Management Software](#) (v8.5.1#805001-sha1:c5b54a7) · [About Jira](#) · [Report a problem](#)

ATLASSIAN

Image 22: Configuration Manager for Jira Create Configuration Snapshot

In the screenshot above, I entered “Source” for the snapshot name, selected four projects, entered a description that includes the source Jira URL, and checked the “filter” and “boards” checkboxes.

Click the “Next” button at the bottom of the page to select *specific* filters, boards, and dashboards, or simply click the “Create” button to include all objects in the snapshot.

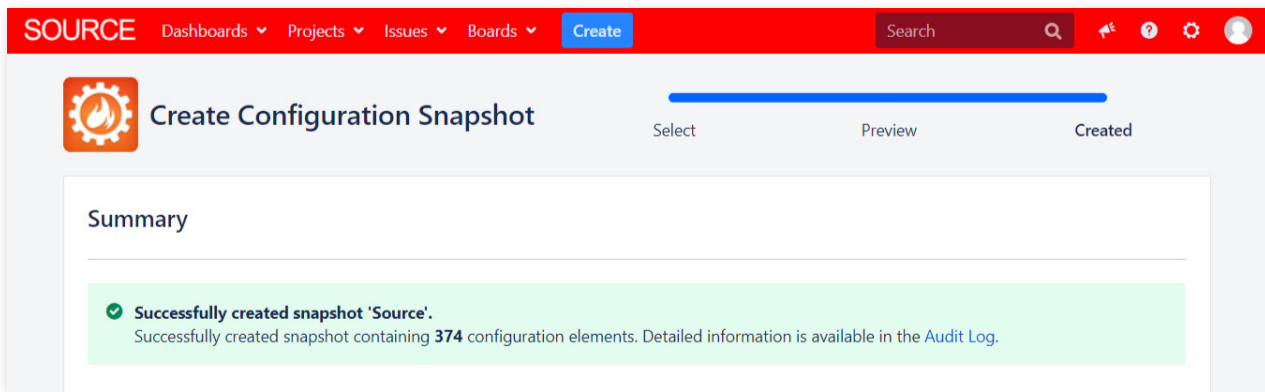


Image 23: Configuration Manager for Jira Snapshot Created

The snapshot was successfully created and added to the audit log. Click the “Done” button at the bottom of the page (not pictured) to return to the *Configuration Snapshots* page.

The *Configuration Snapshots* page now shows the snapshot just created. Helpful information like the snapshot type and date created are shown. Click on the elements link to see a count of included custom fields, groups, workflows, and other settings.

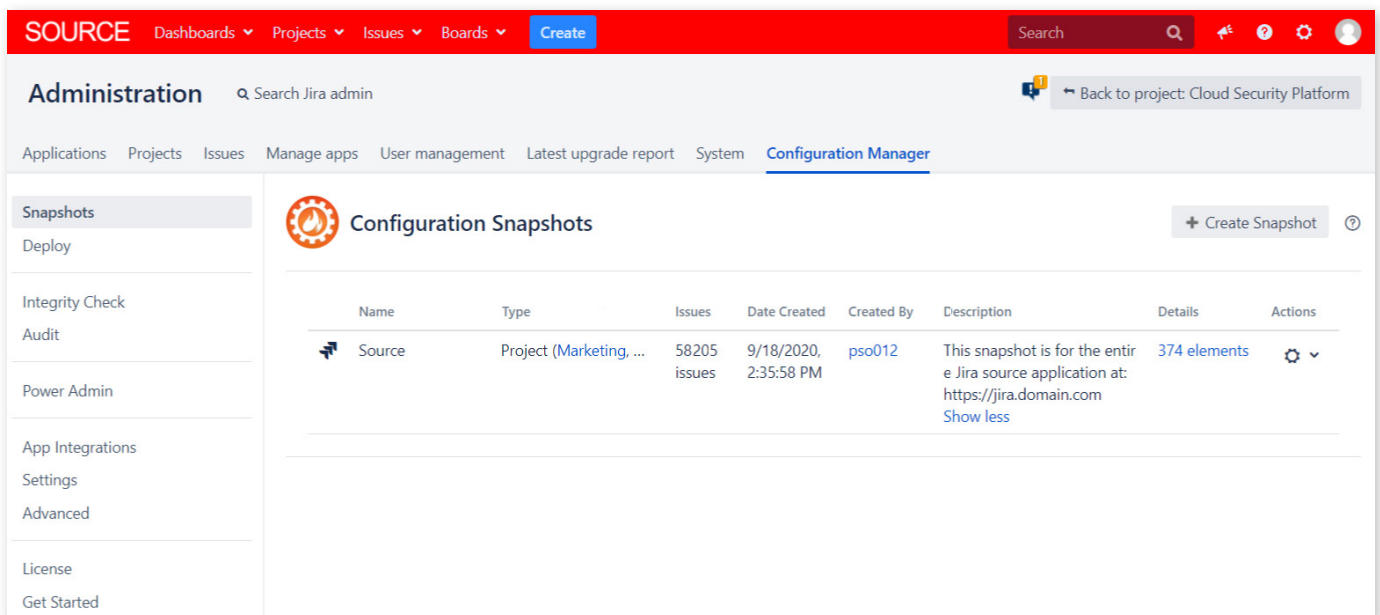


Image 24: Configuration Manager for Jira Snapshot Created

Click the “cog” or “gear” icon in the “Actions” column to download the snapshot in .zip format. The file contains all the .xml and .json information needed to deploy the data in the target application.

Install Configuration Manager for Jira in Staging Application

Install the Configuration Manager for Jira app in the staging environment. Use the same install process previously used in the source environment. You only need a commercial license for the

production instance. Once you purchase a Configuration Manager for Jira license, simply generate development licenses for any test instances.

Step 3: Deploy Snapshot

Deploy Snapshot in Staging Application



Tip: Always back up an application before deploying changes. For more information, see the [Back Up](#) section above.

Now it's time to deploy the source snapshot to a staging application. Log in to the staging application and go to the Configuration Manager for Jira *Deploy Configuration Snapshot* page at Admin > Configuration Manager for Jira > Deploy.



Tip: Configuration Manager for Jira deploys the snapshot to a single application node. For Jira Data Center, you may want to isolate this node by temporarily taking it out of the load balancer.

Snapshot Locations

There are three ways to access snapshot files:

- **From Snapshot File** – For deploying a snapshot downloaded from another application
- **From Current Instance** – For deploying a snapshot taken from the same application
- **From Linked Jira Instance** – For deploying a snapshot stored on a linked application



Tip: For large snapshot files, use the “From Linked Jira Instance” option to copy the file directly to the target application. This is faster than downloading and uploading it in the browser.

Read more about choosing a snapshot at: botron.io/CMJ-choose-snapshot

For the example in this book, click the “From Snapshot File” button on the top right of the *Deploy Configuration Snapshot* page.

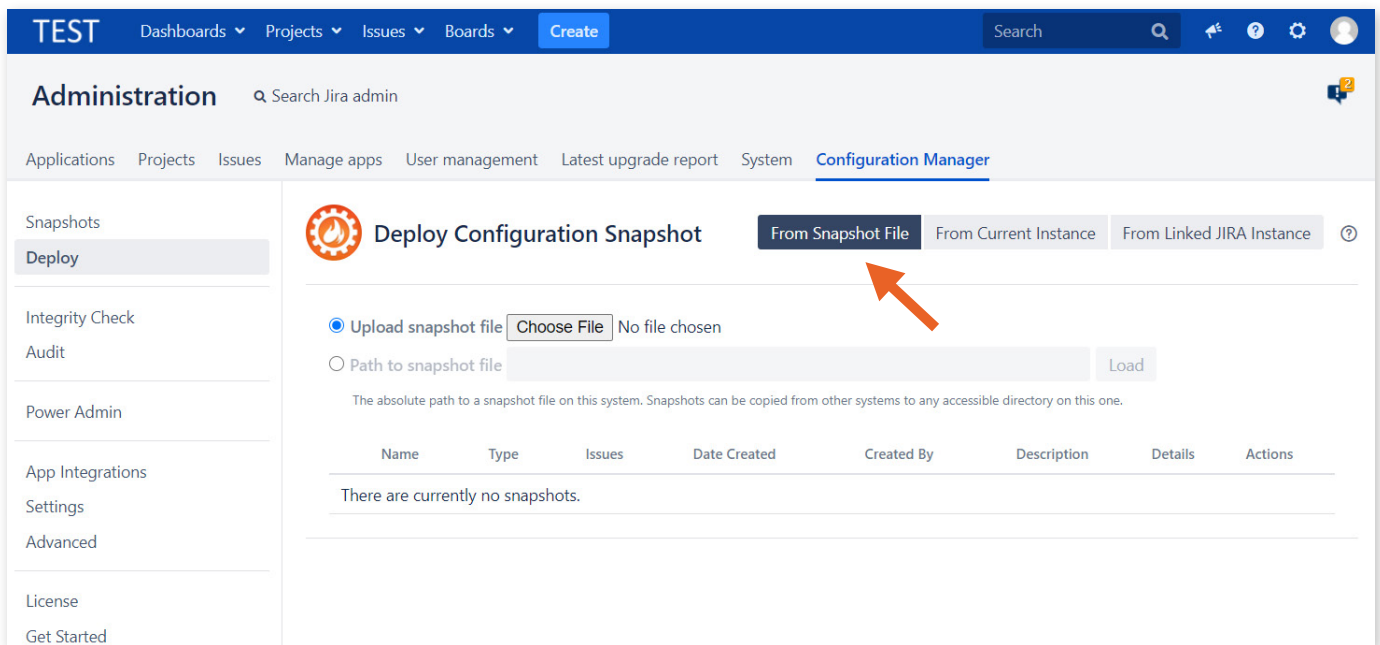


Image 25: Configuration Manager for Jira Deploy Page

Next, upload the snapshot .zip file from your computer.

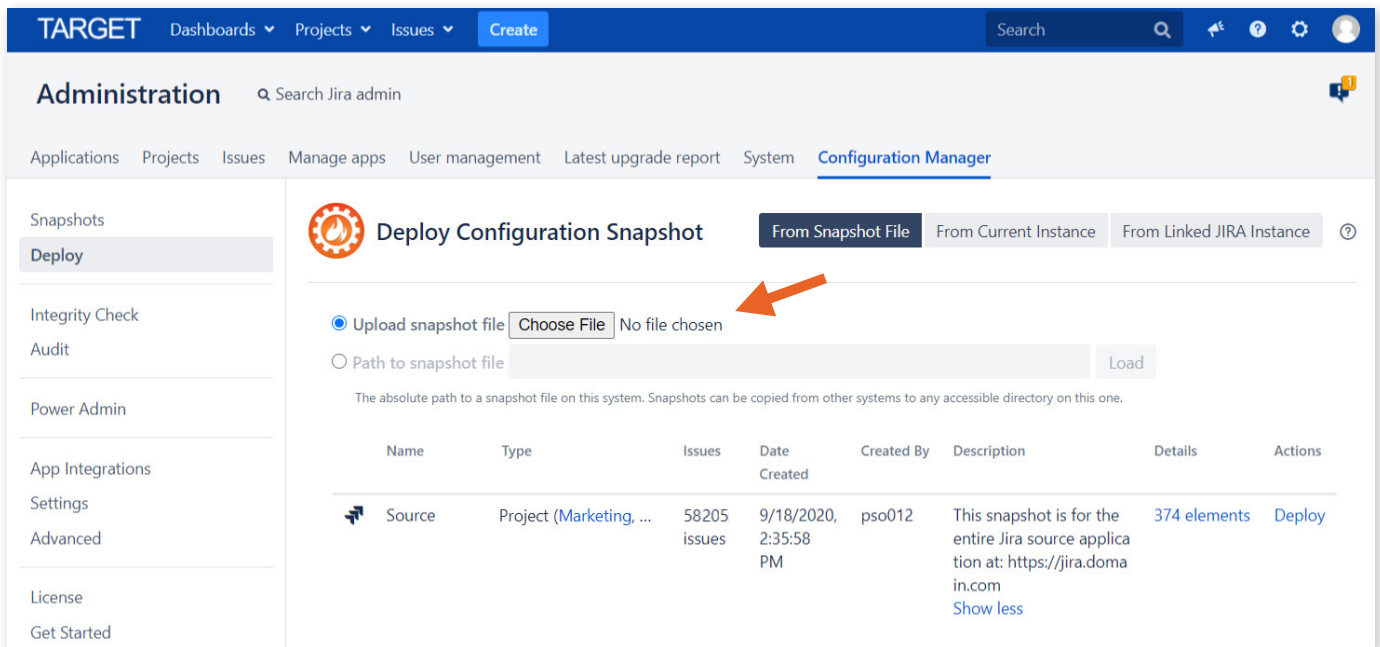


Image 26: Configuration Manager for Jira Deployment Source

The *Deploy Configuration Snapshot* page shows the snapshot just uploaded and includes helpful information, like the snapshot type and date created. Next, click the “Deploy” link in the “Actions” column.

The following screenshot shows information and options for a multi-project deployment.

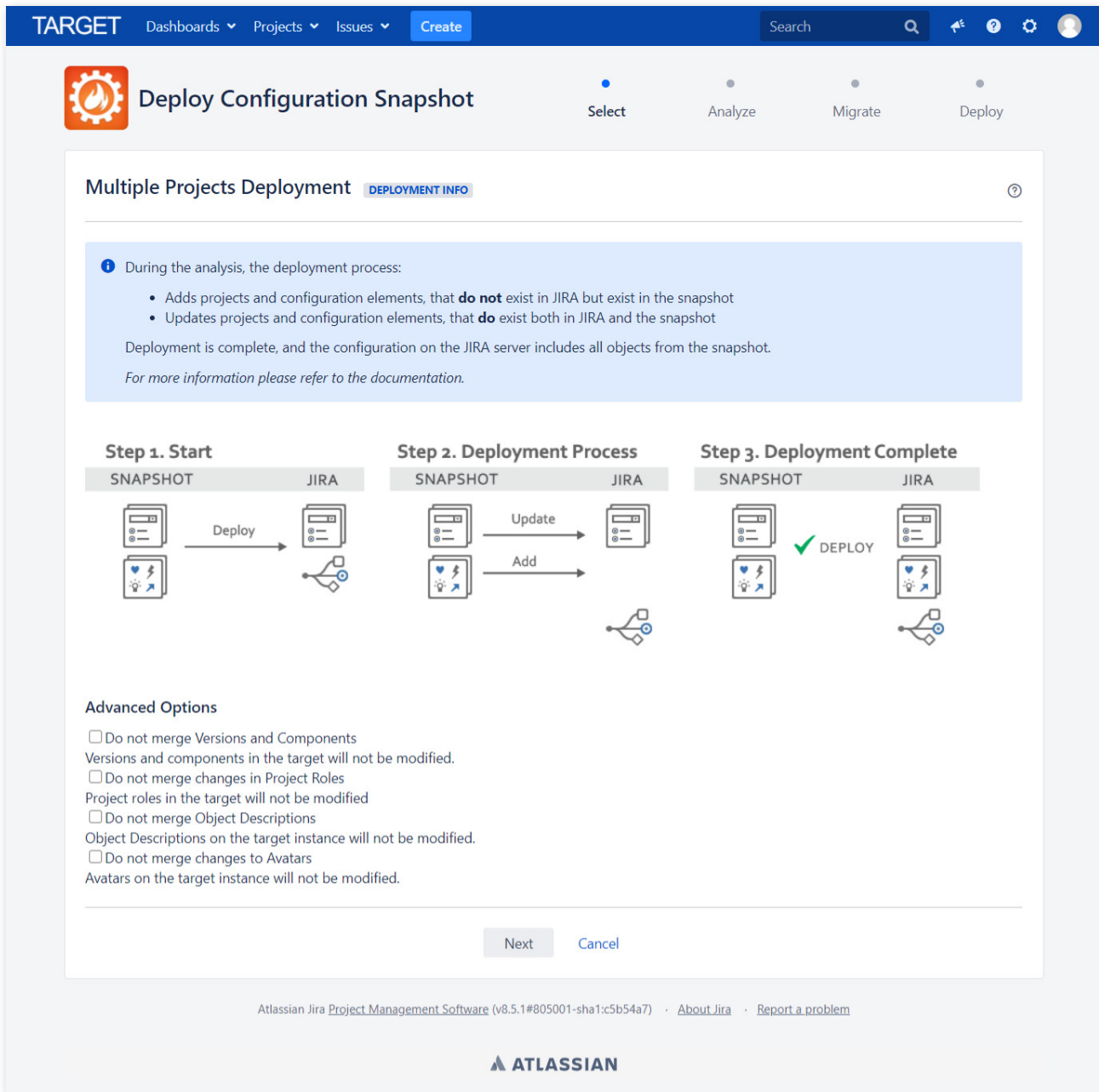



Image 27: Configuration Manager for Jira Deployment Info

Choose any desired options at the bottom of the page and click the form submission button to continue.



Tip: When deploying a “Project with issues” snapshot, there is a re-index option at the bottom of the deployment mode page. For large migrations, check the box to suppress the re-index and perform it manually later. This will be faster. See the “Do not perform re-indexing during the deployment” option in the screenshot.

 **Deploy Configuration Snapshot**

Select●Analyze●Migrate●Deploy●

Select Deployment Mode [DEPLOYMENT INFO](#) Merge Project **New Project** ?

i The snapshot contains issues and related data

New project mode: create a new project, apply the project configuration from the snapshot, import all project issues

Merge mode: merge the snapshot project configuration into the target project, then import all project issues that are not already present in the target project.

New Project

Name*

Max. undefined characters.

Key* ?

Max. 10 characters.

Advanced Options

☐ Do not merge changes to Project Roles

Project Roles on the target instance will not be modified.

☐ Do not merge Object Descriptions

Object Descriptions on the target instance will not be modified.

☐ Do not merge changes to Avatars

Avatars on the target instance will not be modified.

☒ Do not perform re-indexing during the deployment

Re-indexing will not be performed after issue data is imported. The deployment time will be considerably reduced but re-indexing must be manually triggered afterwards.

Attachments*

The path where the attachment files reside on this system.

Next

Cancel

Image 28: Configuration Manager for Jira Deployment Re-index Option

Next, Configuration Manager for Jira analyzes the snapshot contents, compares it to target application settings, and reports any conflicts. Here are some example conflicts and how to solve them:

Analysis Example 1: Missing or Disabled Apps

In the screenshot below, there's a problem with missing or disabled apps. The deployment cannot continue until the issue is resolved.

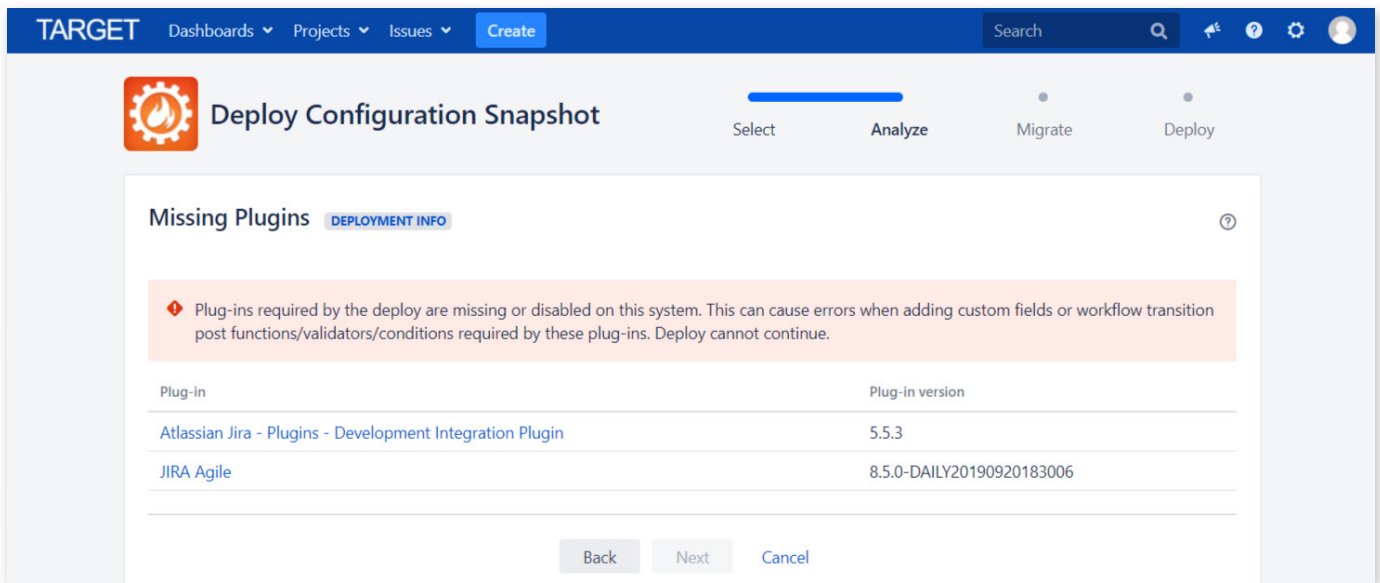


Image 29: Configuration Manager for Jira Deployment Analysis

App details:

- “Jira Agile” refers to the Jira Software application type. In this example, the source application and snapshot contain Jira Software projects and Jira Service Management projects. Jira Software is unlicensed in the target application, however. Adding a free trial or license on the Admin > Applications > Versions & licenses page solves the problem.
- “Atlassian Jira – Plugins – Development Integration Plugin” is a system app related to Jira Software. Licensing Jira Software also resolves this conflict.

To address this specific scenario, there are two options:

1. Add or enable the apps in the target system and restart the deployment (recommended).
2. Remove or disable the apps in the source system, create a new snapshot, and restart the deployment.



Tip: Don't want to disable an app? Use the Power Admin app to assess which elements the app contributes (e.g., custom fields, workflow behaviors, etc.) and remove those items from the configuration. Read more: botron.io/PA-apps

Address any issues encountered and then resume the analysis.

Analysis Example 2: Project Discrepancies

In the screenshot below, the wizard shows a modification to expect.

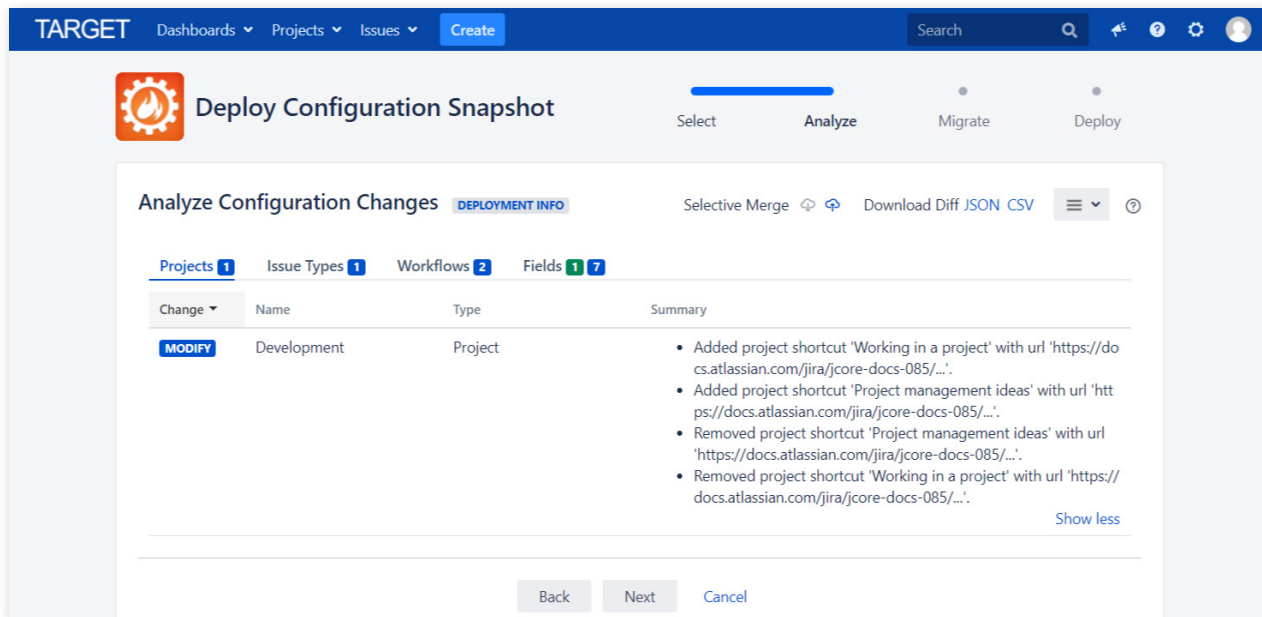


Image 30: Configuration Manager for Jira Deployment Analysis

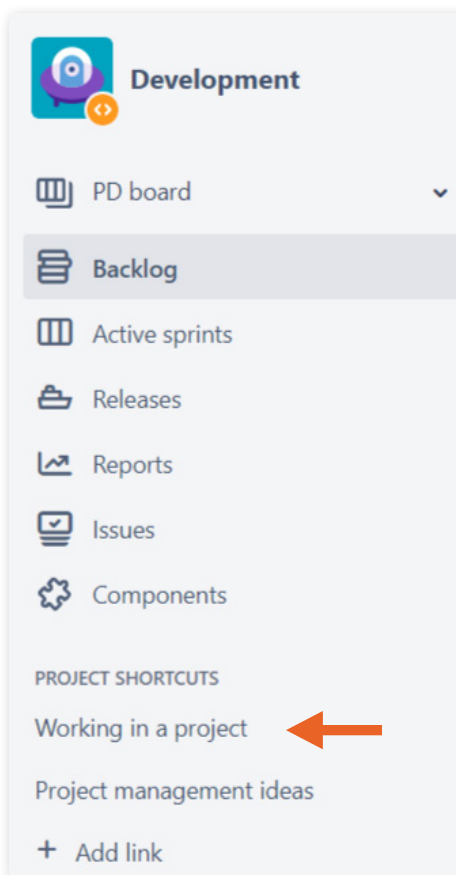


Image 31: Project Shortcuts

In this scenario, both applications have a project named “Development”; however, there is one slight difference with the projects’ sidebar shortcuts.

The first shortcut in the source project (pictured) is named “Working in a project” and the second shortcut is named “Project management ideas”. In the target project, the order is *reversed*.

Even though this is a tiny discrepancy, Configuration Manager for Jira detects it. Because the “Merge Configuration” deployment mode is selected, the project shortcuts in the target application are replaced with settings from the snapshot.

In the screenshot below, the analysis also shows other changes for issue types, workflows, and custom fields. Click on any setting type to see the details.

Deploy Configuration Snapshot

Select Analyze Migrate Deploy

Analyze Configuration Changes **DEPLOYMENT INFO** Selective Merge Download Diff JSON CSV

Projects 1 Issue Types 1 Workflows 2 **Fields 17**

Change	Name	Type	Summary
ADD	→ M Affected Department	Custom Field	• Field context 'Default Configuration Scheme for Affected Department' was added.
MODIFY	→ A → M Epic Colour	Custom Field	• Default value changed from '[ghx-label-2]' to '[ghx-label-8]' for context 'Default Configuration Scheme for Epic Colour'. Show less
MODIFY	→ A → M Other requests by same reporter	Custom Field	• Custom configuration data changed.
MODIFY	→ A → M Request related tasks	Custom Field	• Custom configuration data changed.
MODIFY	→ A → M Default Field Configuration	Field Configuration	• Field Affected Department added to 'Default Field Configuration'.
MODIFY	→ A → M Jira Service Desk Field Configuration for Project CSI	Field Configuration	• Field Affected Department added to 'Jira Service Desk Field Configuration for Project CSI'.
MODIFY	→ A → M JIRA Service Desk Field Configuration for Project IR	Field Configuration	• Field Affected Department added to 'JIRA Service Desk Field Configuration for Project IR'.
MODIFY	→ A → M Marketing field configuration	Field Configuration	• Field Affected Department added to 'Marketing field configuration'.

Back Next Cancel

Image 32: Configuration Manager for Jira Deployment Analysis

One custom field called “Affected Department” exists only in the source application. It will be added to the target application. There are also seven custom field settings that both exist in the source and target applications. The table shows how the target application settings will be modified to match the snapshot.

For help analyzing changes, see: botron.io/CMJ-snapshot-change-analysis

Selective Merge

What if you don't want to perform changes shown in the analysis? Simply use the new “Selective Merge” feature to avoid a change or modify mapping in the target application. Read more: botron.io/CMJ-selective-merge

Use this feature to:

- Rename or add configuration elements without overwriting target elements.
For example, rename “Default Permission Scheme” to “Default Permission Scheme – Source”. The source permission scheme is added to the target application and the target application’s default permission scheme is unchanged.
- Change the mapping (“matching”) between snapshot and target configuration elements.
For example, map the source field called “Team Impacted” to the target field called “Team”.

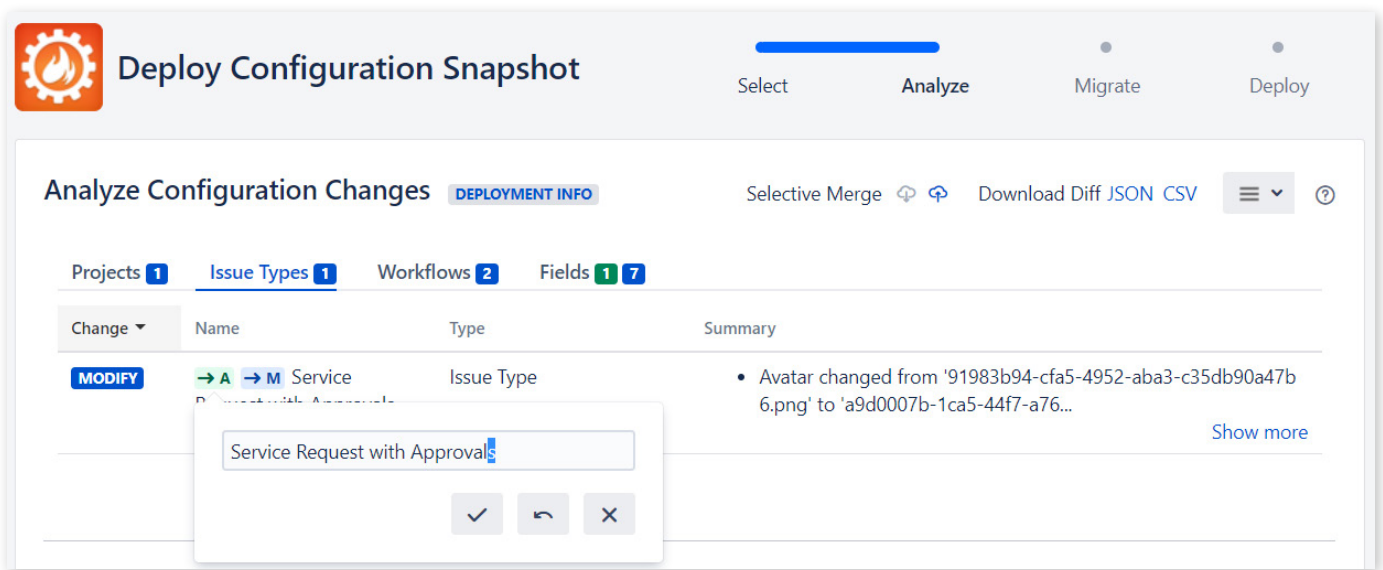


Image 33: Configuration Manager for Jira Selective Merge Rename Feature

In the screenshot above, click the green “A” icon to create a new configuration element instead of overwriting an existing element. For example, change the issue type’s name from “Service Request with Approvals” (plural) to “Service Request with Approval” (singular).



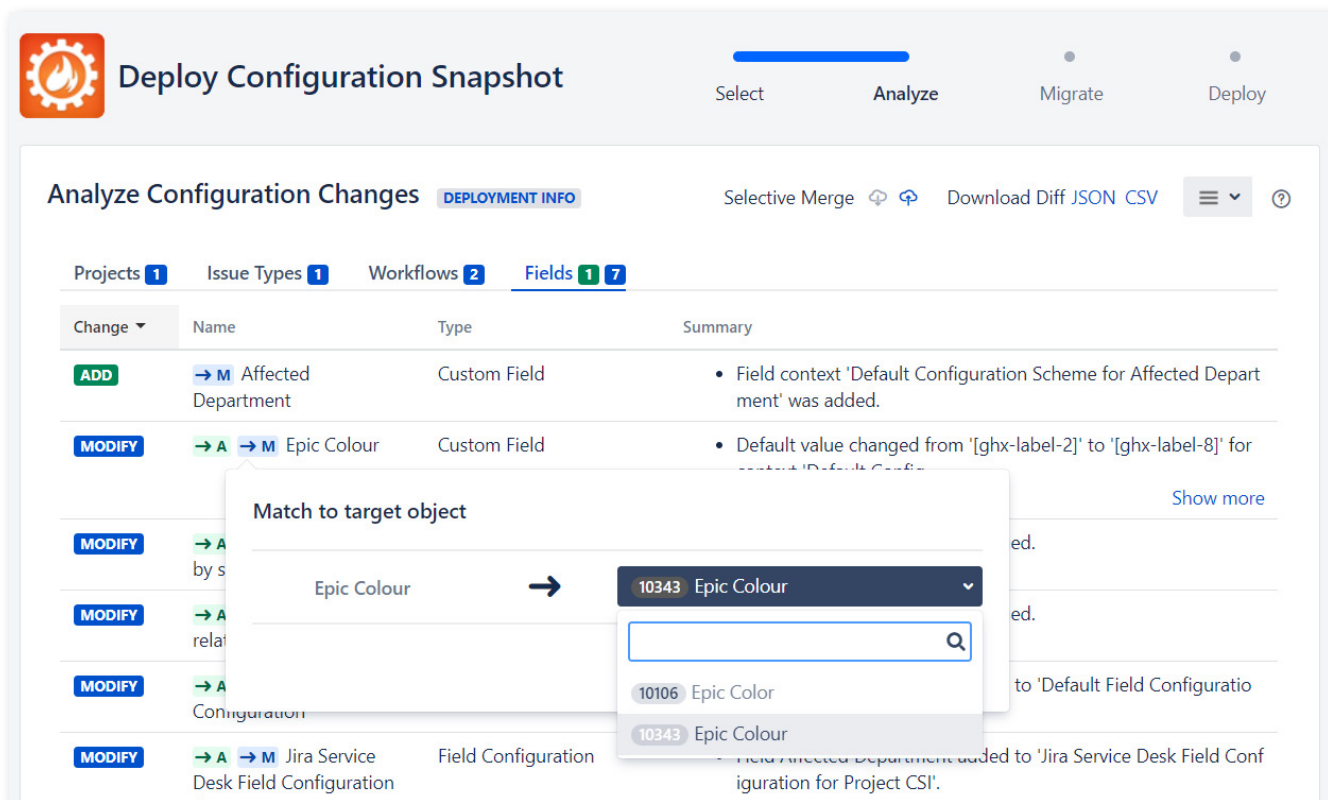


Image 34: Configuration Manager for Jira Selective Merge Matching Feature

In the screenshot above, click the blue “M” icon to change an element’s mapping. For example, the source application contains one field called “Epic Colour”. The target application contains two fields called “Epic Color” and “Epic Colour”. Use this feature to select which target field to map to.

Any selective modifications are noted in the audit log and as a “diff” (difference) analysis. Transforming data with the Selective Merge feature is faster than migrating configurations “as is” and cleaning them up later.



Tip: Click the cloud icon (see screenshot below) to download any selective merge changes. Use this feature to create a template for the next migration. Save time and eliminate user error by using the same settings when testing or replaying the migration process.

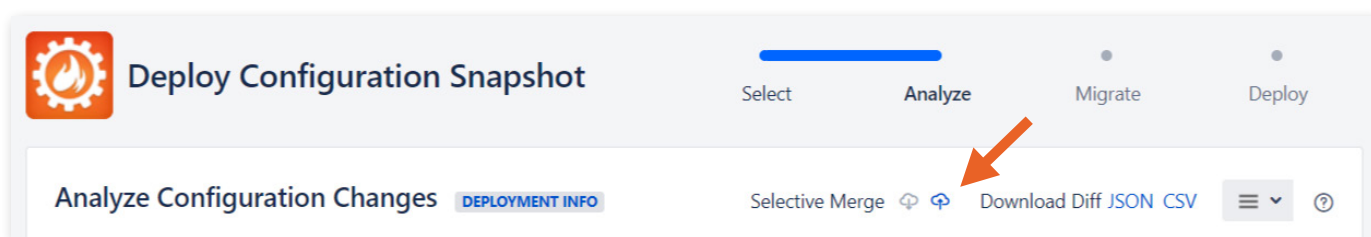


Image 35: Configuration Manager for Jira Selective Merge



Tip: Add any selective merge changes to documentation for testing and post-migration training purposes.

After any selective modifications are made and analysis is complete, click the form submission button to proceed to the next step.

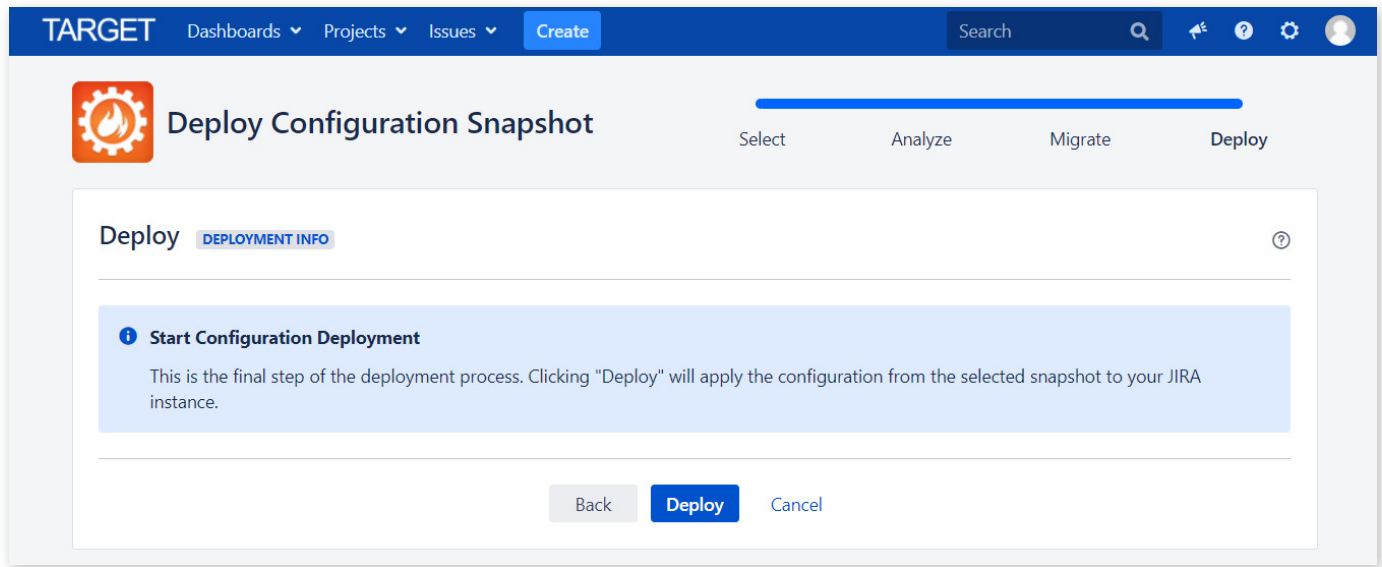


Image 36: Configuration Manager for Jira Start Configuration Deployment

The screenshot above is the last screen before the final deployment step. Click the form submission button to execute the migration.

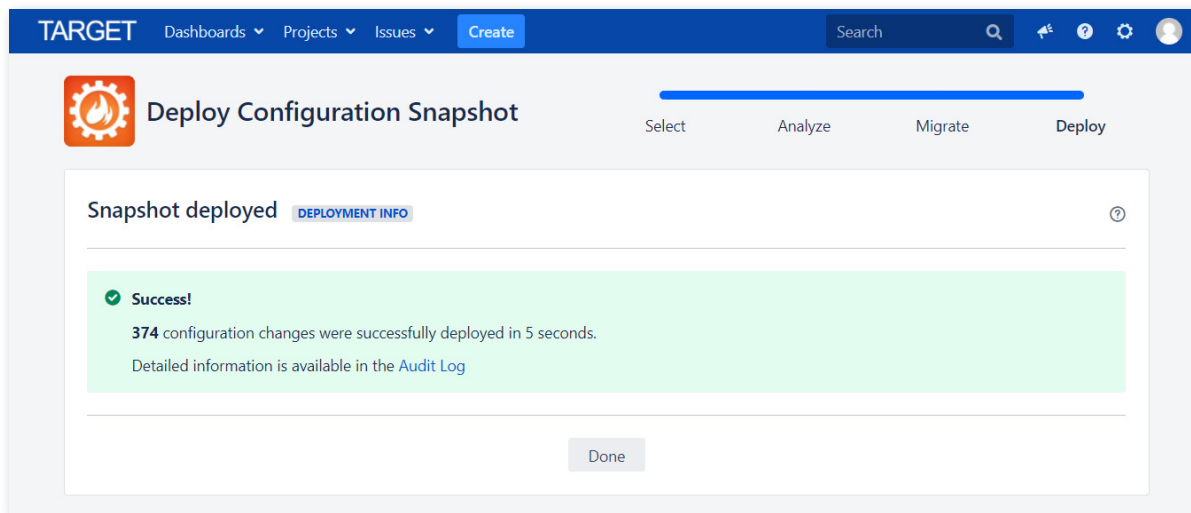


Image 37: Configuration Manager for Jira Deployment Success

The screenshot above shows the deployment was successfully completed.

After the migration, verify that the results are as expected and match the data in the source application. See the [Migration Testing](#) section for tips and test cases.

Install Configuration Manager for Jira in Target Application

Once the staging environment is verified and any issues are remedied, install the Configuration Manager for Jira app in the target environment. Use the same install and license process previously used in the source and staging environments.

Deploy Snapshot in Target Application

Use the exact same process tested in the staging environment to deploy the source snapshot into the target application. Verify that the results are as expected and match the data in the source and staging applications.

User and Group Migration

The Configuration Manager for Jira app can migrate user accounts and groups “as is” or transform them using the “Selective Merge” feature. Here are some specifics:

User Accounts

- Existing user accounts are never deleted or changed.
- User accounts are only exported from the source application if they are referenced by other configuration options.
 - A user with no issues reported, assigned, or specifically named in a scheme or workflow transition will not be included in the snapshot.
 - If unused accounts are desired in the target application, simply create them manually or use an app like Jira Command Line Interface (botron.io/amp-cli) to automate creation.
- New user accounts are created on the target application if they don't already exist.
 - New user accounts are created in the first writable user directory. Note: A warning is added to the Audit Log.
 - Manually move new user accounts to a different directory in the target application, if desired.
- Passwords are automatically generated for new user accounts.
- Existing user passwords are not included in (or exposed by) a configuration snapshot.
- User settings like avatars, favorites, and properties are not included.

Groups

- Existing groups are never deleted or changed.
- Groups are only exported from the source application if they are referenced by other

configuration options.

- If unused groups are desired in the target application, simply create them manually or use an app like Jira Command Line Interface (botron.io/amp-cli) to automate creation.
- New groups are created on the target application if they don't already exist.
 - Exported groups are empty and contain no members. Note: A warning is added to the Audit Log.
 - Simply add users manually, in the target application, as desired.

For merging or mapping user accounts, see the [Conflict Resolution](#) section in the “Migration Process” chapter.

Attachments

Configuration Manager for Jira provides two ways to handle attachment files. When creating a “Project with issues” snapshot, include or exclude them using the wizard’s “Include attachment files” checkbox.

Create Configuration Snapshot

Select Filters Boards Dashboards Apps Preview Created

Select Snapshot Type

System Configuration Project Configuration **Project with Issues**

⚠ Including attachment files in the snapshot may significantly increase its size. If you don't include the files now, on deployment, you will be presented with an option to provide the path where they reside on the target system. Consult with [this documentation article](#) for more information.

ℹ Package the **Configuration and Issues** of the selected Project(s) into a **Project snapshot with Issues**.

Name* Source

Project(s)* IT Requests × Marketing × Cloud Security Platform × Issue Matrix Demo ×

☒ Include attachment files

Description

☐ Include Project Filters

☐ Include Project Boards

* This snapshot is used to move Project Configuration and Issues to a target system.

Next Create Cancel

Image 38: Configuration Manager for Jira Snapshot Attachments

Including attachments significantly increases a snapshot's file size. Depending on web server and bandwidth limitations, the snapshot file may be too big to upload over HTTPS. Botron recommends excluding attachments when they are more than 2GB total in size.

Read more about attachments here: botron.io/CMJ-move-attachments



Tip: 8GB is the minimum recommended heap size but large snapshots may require more memory. Reference Atlassian's recommendations at: botron.io/atl-infrastructure-recs-aws and botron.io/atl-jira-size-guide.

If there are a lot of attachments, exclude them from the snapshot and copy them separately. Start a full directory copy before the migration and use an incremental sync utility (e.g., rsync) to periodically update the attachments. Do a final sync again, right before the migration event.

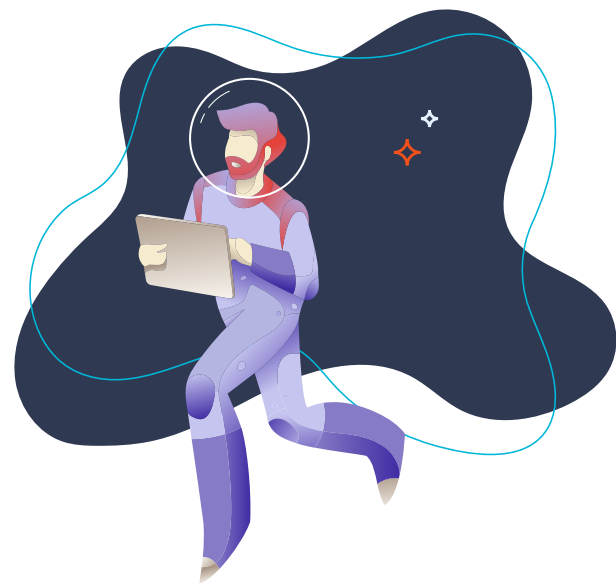


Tip: Temporarily dedicate a faster drive to speed up the copy process.



Note: Attachment metadata (e.g., filename, creator, create date, etc.) is small and is always included in a snapshot.

Jira attachments are stored on the server in separate folders, organized by project key, in the `<JIRA_HOME>/data/attachments/` directory. Copy the source application attachments and deploy them to a different directory on the target server. Example: `<JIRA_HOME>/migration-data/source-attachments/`. Then specify the path in the "Attachments" field on the *Select Deployment Mode* page.



Deploy Configuration Snapshot

Select Analyze Migrate Deploy

Select Deployment Mode **DEPLOYMENT INFO** Merge Project New Project ?

i The snapshot contains issues and related data

New project mode: create a new project, apply the project configuration from the snapshot, import all project issues

Merge mode: merge the snapshot project configuration into the target project, then import all project issues that are not already present in the target project.

Select Target Project

Project(s)

Advanced Options

- ☐ Do not merge changes to Project Roles
Project Roles on the target instance will not be modified.
- ☐ Do not merge Object Descriptions
Object Descriptions on the target instance will not be modified.
- ☐ Do not merge changes to Avatars
Avatars on the target instance will not be modified.
- ☐ Do not perform re-indexing during the deployment
Re-indexing will not be performed after issue data is imported. The deployment time will be considerably reduced but re-indexing must be manually triggered afterwards.
- ☐ Do not import issue attachments
Issue attachments will not be imported on the target instance.

Attachments*

The path where the attachment files reside on this system.

Next Cancel

Image 39: Configuration Manager for Jira Deploy Attachments

Configuration Manager for Jira copies the attachments to the correct location in the target application, even if paths and project keys have changed.

Apps Migration

Until now, there was no easy way to handle data from third-party apps and integrations. Botron changed that by integrating Configuration Manager for Jira with many top vendors in the Atlassian Marketplace. For example, seamlessly migrate workflow conditions, validators, and post functions from JSU Automation Suite for Jira Workflows, Jira Misc Workflow Extensions (JMWE), or ScriptRunner for Jira. Or easily move ProForma forms from one Jira instance to another. The possibilities are endless. See the list of integrated apps at: botron.io/CMJ-apps-list

Some integrations are natively built-in and others are enabled by Botron's Service Provider Interface (SPI). The SPI was designed to make it simple for vendors to implement an integration with Configuration Manager for Jira. The interface provides the tools for moving app configuration and data between Jira applications. It transforms the data so it's relevant to the target location and ports it to the right place. It includes backward compatibility, so integrations don't break in the future. It's a true API with its own versioning. Without the SPI, each vendor would need to deeply understand migration challenges and build their own separate migration utilities.

Need to migrate an app not listed? Please contact the app vendor and request an integration with Configuration Manager for Jira's Service Provider Interface.



Tip: Always verify scripts, workflow behaviors, JQL functions, and sub-queries provided by apps. For example, a filter is migrated but its unsupported JQL function is not. Prepare for known breakage and update these items manually as needed.

Support

Need help with Configuration Manager for Jira? Create a Botron support ticket at: botron.io/support and provide screenshots, steps to reproduce, and any other helpful information from your Jira logs.



Resources

- Configuration Manager for Jira: App Integrations - botron.io/CMJ-apps-integration
- Configuration Manager for Jira: Service Provider Interface - botron.io/CMJ-SPI

Case Study

kreuzwerker is an Atlassian Solution Partner whose client had a common problem. The client had two Jira applications that supported different areas of the business. Managing and licensing two applications was expensive, restricted collaboration, and created two support entry points for end users. kreuzwerker used Botron's Configuration Manager for Jira to move 60 projects, 100 agile boards, 75,000 issues, and 250,000 comments for their client with zero data loss. Read the case study at: botron.io/kreuz-case-study

Now you've tested the migration process in a staging environment with Configuration Manager for Jira. Next, complete the additional steps in the "Migration Testing" and additional sections of **Worksheet 5: Jira Migration Checklist**. Remember: You may need to do multiple test migrations. Don't progress from the staging environment until everything is perfect.

5.5 Migration Execution Strategy

Now that you've done some initial research, considered all that's required for a successful migration, and explored some useful migration tools, it's time to determine whether you'll engage a migration partner or migrate on your own.

Choosing the Right Migration Partner

Atlassian offers some migration support. There's a dedicated Migration Support Team to assist with product, pricing, billing, licensing, and support questions. The Premium Support team can help determine whether a Server application is ready to move to Data Center or Cloud. If you have a Technical Account Manager (TAM) ask them about migration support options as well.

Additionally, Atlassian has a directory of solution partners to help with all types of Jira support needs, including migration assistance. Here are some tips for deciding whether you need a partner and choosing the right one.

Consider a solution partner for your migration when:

- You require more assistance than Atlassian provides.
- There's not enough internal Jira knowledge or migration staff.
- The source application is large or complex.
- There are many users or user management challenges.
- Many apps, integrations, or customizations exist.
- There are complicated migration or merge scenarios.
- There are specific security, compliance, or regulatory concerns.

Choose a solution partner who:

- Is a migration expert, not just a Jira expert.
- Has experience with different types of migrations and merges.
- Is comfortable describing their approach, execution methods, and prior results.
- Can customize a migration strategy to your needs.
- Can help craft and execute a comprehensive plan.
- Is willing to conduct a thorough analysis of the source application's configuration.
- Understands your organization's security and compliance needs.
- Provides backup, roll back, disaster recovery, and escalation plans.
- Follows Jira changes and is active in the Atlassian Community.
- Is responsive, proactive, and nice to work with.



Resources

- Atlassian Partner Directory - botron.io/atl-partners
- Atlassian Migration Services Referral - botron.io/atl-dc-partners

Migrating on Your Own

Experienced Jira administrators may be able to complete migrations on their own. Do you feel comfortable completing the following tasks?

- Merging custom field data
- Completing large workflow changes
- Cleaning up unneeded settings and schemes
- Writing and executing scripts
- Managing user accounts and troubleshooting permissions
- Hunting around the Jira database for information not in the UI
- Finding and fixing errors in the application logs

If you've performed these types of tasks, a migration may be your next interesting challenge and learning opportunity!

Must-have Apps

Regardless of whether you migrate yourself or get additional help, here are some helpful apps to use along the way. Use these to analyze the Jira configuration, fix errors, and migrate data.

Name	Configuration Manager for Jira
Marketplace URL	botron.io/amp-cmj
Description	The most trusted solution for change management, governance & scaling, and data migration in Jira & Jira Service Management. (Includes the Power Admin and Integrity Check for Jira apps.)
Deployment Type	Jira Server, Jira Data Center
Documentation	botron.io/CMJ-overview

Name	Integrity Check for Jira
Marketplace URL	botron.io/amp-ic
Description	Detect, locate, and fix errors in the configuration of Jira
Deployment Type	Jira Server, Jira Data Center
Documentation	botron.io/IC-overview

Name	Power Admin
Marketplace URL	botron.io/amp-pa
Description	The future of Jira administration: Empowering administrators to configure Jira in a better, faster & smarter way
Deployment Type	Jira Server, Jira Data Center
Documentation	botron.io/PA-overview

Additional Worksheets

Use these additional worksheets to manage your migration project.

Worksheet 8: Jira Migration Timeline



Use this worksheet to craft a high-level timeline for a migration project.

ID	Event	Duration	Participants
Ex:	Research and analysis	Start date - End date (or hours)	<ul style="list-style-type: none"> Project Manager Infrastructure Lead Jira Administrator
1	Research <ul style="list-style-type: none"> Source application analysis Target application analysis Initial project approval Etc. 		



ID	Event	Duration	Participants
2	Preparation <ul style="list-style-type: none">• Upgrade source application and apps (if applicable)• Source application clean-up, backup, archive• Health and performance testing• Etc.		
3	Testing <ul style="list-style-type: none">• Create staging environment• Staging migration• Staging testing• Etc.		
4	Migration <ul style="list-style-type: none">• Target migration• Target testing• Retire source application• Etc.		
5	Support <ul style="list-style-type: none">• Post-migration support• User communication• Training• Etc.		



Worksheet 9:

Jira Migration Communication Plan



Use this worksheet to communicate migration milestones and plans to stakeholders and users.

Tip: Also set notification banners in the Jira source and target applications! (Jira Server and Data Center only)

Before Migration

Recommended communications:

- ☐ **Detailed migration announcement for stakeholders**
Include: Purpose, path, scope, expected impacts, team contacts, project tracking details, high-level timeline, etc.
- ☐ **General migration announcement for users**
Include: Purpose, benefits, what to expect, single contact point, high-level timeline, etc.

ID	<i>Ex:</i>
Subject	<i>General migration announcement for users</i>
Audience	<i>End users</i>
Send Date	<i>December 1, 2020</i>
Sender	<i>Rachel Wright</i>
Content	<p><i>Dear users,</i></p> <p><i>The merger with company X is complete and we are ready to start consolidating user accounts and applications. Our goal is to merge Atlassian applications so we have just one set for the entire organization. This will help us collaborate, store information, and benefit from a standard set of tools, processes, and terminology.</i></p> <p><i>A migration team has formed to tackle this project and we've started an initial analysis. Additional details, like expected impacts and a timeline will be provided in the coming weeks. In the meantime, if you have any questions, please contact [contact name] at [contact details].</i></p>



ID	1
Subject	
Audience	
Send Date	
Sender	
Content	

During Migration

Recommended communications:

- ☐ **Progress update for stakeholders**
Include: Progress details, items completed, decisions, timeline modifications, etc.
- ☐ **Configuration freeze reminder**
Include: Freeze duration, impactful changes, known issues, etc.
- ☐ **Test team instructions**
Include: URLs, expectations, test period start and end dates, where to log issues, etc.
- ☐ **Migration reminder**
Include: Planned completion date, URL changes, training plans, single point of contact, etc.

ID	1
Subject	
Audience	
Send Date	
Sender	
Content	

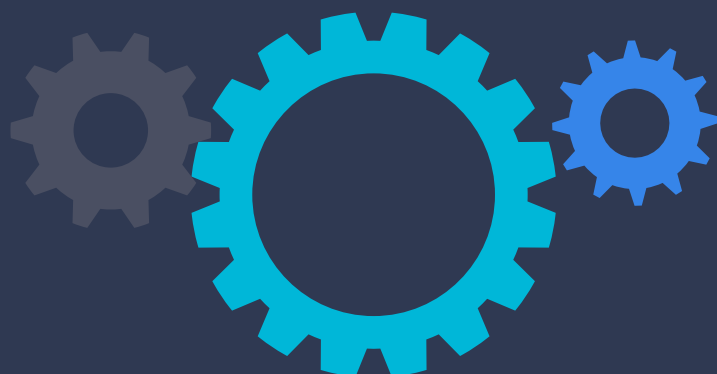


After Migration

Recommended communications:

- ☐ **Completion announcement for stakeholders**
Include: URL changes, known differences, remaining challenges, where to log issues, etc.
- ☐ **Completion announcement for users**
Include: URL changes, process changes, known differences, where to log issues, etc.
- ☐ **Training plans**
Include: How to attend training or access materials, etc.

ID	1
Subject	
Audience	
Send Date	
Sender	
Content	



Worksheet 10: Jira Migration App Decisions



Use this worksheet to record decisions for Jira apps, connections, integrations, and customizations. Use the initial information already compiled in **Worksheet 2: Jira Apps and Customizations**.

Apps

See the apps installed by visiting Admin > Manage Apps.

Tip: Use the Power Admin app by Botron to see how apps are currently used. Read more: botron.io/amp-pa

ID	Ex:
Name & Marketplace URL	Configuration Manager for Jira (botron.io/amp-cmj)
Version	6.6.8
Expiration	19/Apr/22
Cost	\$400
Use	Test environment sync, quarterly maintenance, etc.
Notes	Also needed for June migration
Compatible	Yes
Update	No
Migrate	Yes



ID	1
Name & Marketplace URL	
Version	
Expiration	
Cost	
Use	
Notes	
Compatible	
Update	
Migrate	

Connections & Integrations

Tip: Check for connections and integrations in the following Jira administration areas: Admin > Manage Apps, Admin > Applications, and Admin > System > WebHooks.

ID	Name & URL	Type	Expiration	Cost	Use	Notes	Compatible	Migrate
Ex:	Confluence (botron.io/atl-confluence)	Application link	10/Jun/21	\$1,250	Document repository, collaboration portal, etc.	Same expiration as Jira	Yes	Yes
Ex:	Slack (slack.com)	WebHook	15/Dec/20	\$2,001	Chat	n/a	Yes	Yes
1								
2								
3								
...								



Customizations

ID	Name & URL	Type	Use	Notes	Compatible	Migrate
Ex:	Outgoing mail disabled	Properties file	Disable email notifications in staging environment.	Set "datlassian.mail.senddisabled" to "true" in setenv.sh file	n/a	n/a
Ex:	Groovy script	Workflow validator	If the "Component" selected is "MS Office" require the user to select an application name (e.g. Word, Excel, etc.) in the field "Application Name" in the "Request Details" transition.	If a value is not provided during the transition for field(s) Application Name, show the following error: Please select the application. This validation only applies if the following condition is true: <code>issue.get("components")?.any{it.name == "MS Office"}</code>	No	No
1						
2						
3						
...						

5.6 Migration Considerations

Here are a few additional factors to consider when crafting a migration plan.

Refresh Environment

When testing a migration, make sure the test data is similar in size, content, and configuration to real data. Testing a smaller data set won't provide an accurate duration estimate for important activities like backup, data copy, and re-index. Testing with a different scheme configuration won't surface collisions to handle. You may need to periodically refresh, sync, or otherwise update staging environment data with production data.

Freeze Configuration

Changing Jira's configuration during a migration test makes the whole process harder. Changes may create new conflicts, invalidate test cases, or cause unintended results. You want the final migration to run exactly as it did during the test phase.

Use **Worksheet 11: Jira Migration Configuration Freeze** to help the Jira administration team know when to refrain from making configuration changes. Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 11: Jira Migration Configuration Freeze



Use this worksheet to communicate periods when the Jira configuration should not be changed.

Staging Application

The application for testing the migration procedure and migrated data.

URL:			
Deployment Type:	<input type="checkbox"/> Cloud	<input type="checkbox"/> Server	<input type="checkbox"/> Data Center
Freeze Start Date:		Freeze End Date:	
Dependencies:			
Notes:			



Target Application

The destination for migrated data.

URL:			
Deployment Type:	<input type="checkbox"/> Cloud	<input type="checkbox"/> Server	<input type="checkbox"/> Data Center
Freeze Start Date:		Freeze End Date:	
Dependencies:			
Notes:			

Impactful Changes

Certain changes, like creating new schemes, will impact the migration process. Other general Jira use changes, like creating new issues, won't impact the migration process. Here are some example changes to do and to avoid. Be sure to communicate this list to the Jira administration team.

During the freeze period:

Do	Do Not
<ul style="list-style-type: none">• Add new users (using the correct username format)• Create issues and otherwise use end user functions	<ul style="list-style-type: none">• Update existing user accounts• Create new schemes<ul style="list-style-type: none">◦ Includes: Issue type schemes, workflow schemes, screen schemes, permission schemes, etc.• Create new settings<ul style="list-style-type: none">◦ Includes: Issue types, workflows, custom fields, statuses, resolutions, priorities, projects, project roles, etc.• Change schemes or settings



Tips:

- Document the list of application administrators. If configuration changes are made during the freeze window, you'll want to quickly be able to contact the responsible party. Either ask administrators to wait or collect the change details so you can check for migration impacts.
- For change details, check the application audit log at Admin > System > Audit Log and the app audit log using the link at the bottom of the Admin > Manage apps > Manage apps page.
- If an administrator continues to make changes during the freeze period, consider temporarily reducing or revoking their administration abilities. Also, check the Atlassian Marketplace for apps that may help limit permissions and track or audit changes. Consider disabling the "Extended project administration" ability in permission schemes.

Known Issues

The staging application may behave differently during migration testing. During the freeze period, communicate the following to users:

- The application(s) may be unavailable or periodically restarted.
- The application(s) may be available, but you may not be able to log in.
- The application(s) may report errors.

Include any other problems or abnormalities. Tell users where to report other issues encountered (if reports are desired).

Launch

What is the final migration launch plan? Choose a start and end date that impacts users the least, doesn't conflict with other important events, and ensures key team members are available to execute the needed steps.

Selecting a Launch Window

Before selecting a migration start and end date, consider the following:

- What are the peak Jira use hours to avoid?
- Are there time zone differences to accommodate?
- Are there upcoming events to avoid? Examples: An internal software release, completion of a strategic initiative, holidays, vacation days, etc.

- Are key migration team members available for the entire launch period?
- Are additional emergency resources available, if needed?
- Is enough time available? (Starting a migration on Saturday morning is safer than Sunday afternoon.) Plan extra time in case a step runs longer than expected.

Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 12: Jira Migration Sample Launch Plan



Use this worksheet to create a migration launch plan that fits your needs.

ID	Status	Task	Owner	Start (Duration)	Notes
Ex:	Done	Post "migration started" banner	Rachel Wright	12/12/20 9:00 AM EST (5m)	Add announcement banner to source
1		Send "migration started" communication			Send pre-written notification to all users
2		Prevent source login			Remove application access for jira-software-users at: Admin > Applications > Application access
3		Backup			Back up file system and database
4		Copy attachments to target			Perform final incremental file sync
5		Take source snapshot			
6		Deploy snapshot in target			
7		Re-index target			
8		Perform manual changes			Update settings, properties, scripts, connections, etc.
9		Test and verify target			
10		Allow target login			
11		Send "migration complete" communication			Send pre-written notification to all users
12		Perform post-migration support and activities			Verify target backup, periodic clean-up, maintain staging environment, archive, etc.
...					

Also see the **Jira Migration Checklist** in the [Migration Process](#) section for additional steps to complete.

Launch Communication

Before and after the migration, communicate the following to stakeholders and users. Reference **Worksheet 9: Jira Migration Communication Plan** in the [Additional Worksheets](#) section.

- When will the application be unavailable?
- When will the production migration start and end?
- When can users log into the target application?
- What has changed? Include URLs, process changes, UI differences, and anything expected to be different.
- Where should problems be reported? Provide multiple contact methods, like an email address or chat room, in case issues cannot be reported in Jira.

Pre-Migration Training

If migrating between Jira Server and Jira Data Center, the user interface will look the same for the end user. The application administrator will see a few capability differences, however, like project archival, content delivery network capabilities, and file replication settings. If migrating to or from Jira Cloud, the user interface and capabilities are quite different.

Before the migration, give users a high-level demo of the differences they can expect. Open the source and target applications in two browser windows and show a side-by-side comparison. Here are sample pre-migration demo agendas for each audience:

Demo Agenda: Jira Server and Jira Data Center for Application Administrators

- Introduction
 - Include: Migration purpose, path, scope, expected impacts, team contacts, project tracking details, high-level timeline, etc.
- Data Center Differences
 - Include: Whether you'll have a clustered or non-clustered application
 - Use the "Benefits of Clustering" and "Benefits of One Instance" information in the [Migration Options](#) chapter for details.
- Data Center Features
 - Show: Project Archival (Admin > Projects), application settings like Replication, CDN, and LexoRank Management (Admin > System > General configuration), etc.
 - Use the "Server vs. Data Center vs. Cloud" section in the [Jira Deployment Differences](#) chapter for details.
- Questions

Demo Agenda: Jira Cloud for Users

- Introduction
 - Include: Migration purpose, benefits, what to expect, single contact point, high-level timeline, etc.
- Cloud Differences
 - Include: A side-by-side comparison of the navigation:
 - » For users: login, system dashboard, projects page, sample issue, workflow transition menu, user profile page, etc.
 - » For application and project administrators: user management page, project administration page, etc.
 - For more, see my Jira Server vs. Jira Cloud Interface Comparison article at: botron.io/rw-server-cloud-compare
- Cloud Features
 - Show: People menu (top navigation), collapsible menus for custom fields (right issue sidebar), Roadmap feature, Next-gen project type (if enabled), etc.
 - Use the “Server vs. Data Center vs. Cloud” section in the [Jira Deployment Differences](#) chapter for details.
- Questions

5.7 Merge Considerations

Merging applications is considerably more complex than simply porting data from one application to another. Here are a few additional factors to consider for merges.

Application Comparison

In the “Migration Process” chapter, you used **Worksheet 7: Jira Migration Configuration Assessment** to audit and assess application configuration elements. Complete the assessment worksheet again for any other applications to merge. Then, compare the differences to identify conflicts and any decisions that are needed.

See also the “Consolidation” section in the [Migration Options](#) chapter for additional details.

Conflict Resolution

Conflicts are inevitable when merging multiple applications. It's important to identify them and make decisions that support how you want the target application to function in the long term. Some conflicts impact end users (e.g., custom field changes) and others only impact administrators (e.g., screen schemes, permission schemes, etc.)



Tip: For objects only visible to administrators, prefix the scheme name with a unique value (e.g., "source_Default Permission Scheme") so you can tell which application it came from.

Simple Conflict Example 1: Priority Schemes

An organization is merging an old Jira Server version 7.5 application with a Jira Server version 8.12 application. One of the feature differences between the applications is the addition of priority schemes. In Jira 7.5 and below, priorities were global. Issues and projects all used the same list of selections. The original priority names were "Blocker", "Critical", "Major", "Minor", and "Trivial". More recent versions of Jira allow project-specific priority lists and use new terminology like "Highest", "High", "Medium", "Low", and "Lowest". As such, the configuration assessment uncovered the following settings:

Application 1: Jira Server 7.5	Application 2: Jira Server 8.12
<p>Priorities – All Projects:</p> <ul style="list-style-type: none">• Blocker• Critical• Major• Minor• Trivial	<p>Priorities – Project 1:</p> <ul style="list-style-type: none">• Highest• High• Medium• Low• Lowest <p>Priorities – Project 2:</p> <ul style="list-style-type: none">• High• Medium• Low

How should this conflict be resolved? The best options are to either transfer the version 7.5 selections to the target application or map the version 7.5 selections to their closest equivalents in the target application. You may also need to choose a new default selection.

Option 1: Transfer selections	Option 2: Map selections
<p>Outcome:</p> <ul style="list-style-type: none">• Blocker• Critical• Major• Minor• Trivial• Highest• High• Medium• Low• Lowest	<p>Mapping:</p> <ul style="list-style-type: none">• Blocker → Highest• Critical → High• Major → Medium• Minor → Low• Trivial → Lowest <p>Outcome:</p> <ul style="list-style-type: none">• Highest• High• Medium• Low• Lowest

Option 1, on the left, means all 10 selections are available. Any projects without a specifically associated priority scheme will see the default, which includes all selections. It might be confusing for users and reports to have both “Major” and “Medium” in the same project.

Option 2, on the right, means a name change for previous values. For example, “Blocker” issues are now classified as “Highest” priority. This difference will require query updates and end user notification.

Regardless of which method you choose, always consider the end goal, and keep the long-term health of Jira in mind.



Tip: The Configuration Manager for Jira app can be used with any supported Jira version, and it handles cross version deployments, too. You don't *have to* match versions or upgrade, but it may make migration easier.

Simple Conflict Example 2: Permission Scheme Settings

Both applications have a default permission scheme, but they have different settings.

Application 1: Default Permission Scheme	Application 2: Default Permission Scheme
Administer Projects <ul style="list-style-type: none">Project role: AdministratorsSingle user: Chris Jones Manage Sprints <ul style="list-style-type: none">Single user: Chris JonesSingle user: Kelly Smith	Administer Projects <ul style="list-style-type: none">Project role: AdministratorsGroup: jira-administrators Manage Sprints <ul style="list-style-type: none">Project role: AdministratorsGroup: jira-administratorsGroup: team-leads

How should this conflict be resolved? If you choose to overwrite one scheme with the other, multiple users will lose access to administer projects and manage sprints. A better option would be to change the name of one of the schemes and migrate it as is, or to encompass all the settings in one scheme. For example, adding single user “Chris Jones” to the “jira-administrators” group and adding single user “Kelly Smith” to the “team-leads” group negates the need for two separate schemes.



Tip: Use the “Selective Merge” feature, in the Configuration Manager for Jira app to change the name of the first scheme.

Simple Conflict Example 3: Conflicting Custom Fields

Both applications have the same custom field. The Configuration Manager for Jira app detects this conflict and helps resolve it. If the custom field IDs are the same, Configuration Manager for Jira will automatically match the source field to the correct target field. You’ll be able to confirm the suggested matching in the wizard. If the IDs are different, you can map the source field to a different target field or create a new target field.



Tip: Even if custom fields have the same name, their type and use may be different.

For example, there are two custom select fields named “Country”. One field’s selections are “Bulgaria”, “India”, “Switzerland”, and “United States”. The other field’s selections are in country code format instead (e.g., “BG”, “IN”, “CH”, and “US”). Choosing one format over the other will break queries. It’s important to understand each field’s popularity before deciding between formats. Or, if both fields are widely used, it may be safer to create a new field context instead.



Note: If there’s a custom field context conflict, the Configuration Manager for Jira app will migrate it as “disabled” instead of removing it.



Tips:

- Watch out for changes to a field’s context. Changing the scope from project to global (or vice versa) may have unintended consequences.
- Renaming custom fields breaks filters. Use the Integrity Check feature of Configuration Manager for Jira to understand the impact of any custom field changes. Then, update any impacted user objects like filters, dashboards, or boards.
- Renaming custom fields with the Power Admin app automatically updates impacted user objects.

Complex Conflict Example: Username Format Differences

One potentially complex conflict example is managing user account differences. For example:

- Username format differs
 - Common when changing application types
 - e.g., “first.last” in Jira Server and “first.last@domain.com” in Jira Cloud
- Multiple Jira accounts
 - Common in merge scenarios
 - e.g., “first.last” in application 1 and “flast” in application 2
- Multiple directory accounts
 - Common in acquisition scenarios
 - e.g., “rachel.wright@domain1.com” and “rachel.wright@domain2.com”

Just like in the scenarios above, you need to decide what to do. If you migrate user accounts “as is” the users will end up with multiple accounts, which is usually not desirable. And of course, Jira provides no native way to merge accounts. The other option is to map user accounts from old to new.

There are few tools to migrate users “as is” or to help with mapping, merging, or transforming accounts. Atlassian’s Cloud migration tools use their own internal script. For Server or Data Center, use the “Selective Merge” feature in the Configuration Manager for Jira app. “Selective Merge” allows you to choose how to deploy users and groups in the target application.

Otherwise, the only other option is scripting. I recommend the ScriptRunner for Jira app, which works in all Jira deployment types. Read more: botron.io/amp-scriptrunner

Here are some user migration recommendations:

- Start by identifying differences. Use **Worksheet 7: Jira Migration Configuration Assessment** above and **Worksheet 13: Jira Migration User Account Strategy** below.
 - Don’t forget to consider service accounts and REST API users. Review how these accounts are used and if additional migration considerations exist or test cases are needed.
- Create users and groups in the target application before migration.
- If using the same external directory, verify that usernames in the source application match usernames in the target application. Read more about username normalization at: botron.io/CMJ-merge-servers
- Communicate to users that they will need to set any personal preferences (e.g., their time zone or email notification format) in the target application.



Tip: There’s only one chance to get user management right! Regardless of the strategy, handle it before or during the migration, not after. After migration, Jira provides no way to clean up users or their change history.

You previously compiled a list of users in **Worksheet 7: Jira Migration Configuration Assessment** in the [Configuration Assessment](#) section. Use **Worksheet 13: Jira Migration User Account Strategy** to consider how to migrate users and if any usernames need to be changed. Also list any service or REST API accounts to make sure they and their integrations are accounted for.

Worksheet 13:

Jira Migration User Account Strategy



Use this worksheet to record and determine user management strategies.

Staging Application

The application for testing the migration procedure and migrated data.

URL:			
Deployment Type:	<input type="checkbox"/> Cloud	<input type="checkbox"/> Server	<input type="checkbox"/> Data Center
Active User Count:		Inactive User Count:	
Username Format:	(e.g. first.last)		

Service or REST API Accounts

ID	Username	Directory	Use
Ex:	reporting_bot	Jira Internal Directory	To generate the monthly time report for client billing
1			
2			
3			
...			

Target Application

The destination for migrated data.

URL:			
Deployment Type:	<input type="checkbox"/> Cloud	<input type="checkbox"/> Server	<input type="checkbox"/> Data Center
Active User Count:		Inactive User Count:	
Username Format:	(e.g. first.last)		



Service or REST API Accounts

ID	Username	Directory	Use
Ex:	reporting_bot	Jira Internal Directory	To generate the monthly time report for client billing
1			
2			
3			
...			

Migration Strategy

- ☐ Migrate accounts “as is”
- ☐ Transform username format
- ☐ Create new accounts (not recommended)
- ☐ Other: _____

Details

e.g., Migrate all users including service accounts. Prefix all inactive usernames with “source_”.

5.8 Migration Testing

After proper planning, testing is the next most important part of any migration event. Make sure enough time and effort is devoted to testing. Remember that testers often have “regular work” demands competing for their time. Here are some worksheets and sample test cases to help in the testing phases.

Test Team

Use **Worksheet 14: Jira Migration Test Team** to identify test team members and share important details, like application URLs and test case due dates. Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 14: Jira Migration Test Team



Use this worksheet to create teams to test the sandbox and production migrations. List each team member, their contact details, and what they will test.

Staging Application

The application for testing the migration procedure and migrated data.

URL:			
Deployment Type:	<input type="checkbox"/> Cloud	<input type="checkbox"/> Server	<input type="checkbox"/> Data Center
Testing Start Date:		Testing End Date:	

Team Members

ID	Name	Contact Details	Time Zone	Role	Scope
Ex:	Rachel Wright	rwright@jirastrategy.com	EST	Jira Admin	<ul style="list-style-type: none">• Main development workflow• Default system dashboard• Personal boards and dashboards
1					
2					
3					
...					



Target Application

The destination for migrated data.

URL:			
Deployment Type:	<input type="checkbox"/> Cloud	<input type="checkbox"/> Server	<input type="checkbox"/> Data Center
Testing Start Date:		Testing End Date:	

Team Members

ID	Name	Contact Details	Time Zone	Role	Scope
Ex:	Rachel Wright	rwright@jirastrategy.com	EST	Jira Admin	<ul style="list-style-type: none">Issue count in all projectsShared filters
1					
2					
3					
...					

Test Cases

There are two types of testing to complete. The migration team first validates the application's data and then, along with the end users, they validate the application's functionality.

Also check the application logs for any unexpected issues.

Data Tests

The first tests are for the migration team to simply make sure the migrated data is exactly what was expected. This testing phase answers questions like:

- Did all the schemes migrate as desired?
- Are the expected issue counts in each project accurate?
- Are all the needed users and groups present?

- Was all “do not migrate” data excluded?
- Were any special cases handled as desired?
- Etc.



Tip: To automatically count elements, use the Command Line Interface (CLI) app, SQL, the REST API, or the “Database Statistics” section of the Admin > System > System Info page. Verifying specific elements should be done manually, however.

Use the information you previously compiled, in **Worksheet 7: Jira Migration Configuration Assessment**, in the [Configuration Assessment](#) section, to verify all configuration elements are as expected.

Worksheet 15: Jira Migration Configuration Testing is the same as **Worksheet 7: Jira Migration Configuration Assessment** except it includes a “result” column to record test findings.

Download this worksheet! Get your customizable copy at botron.io/worksheets

Worksheet 15: Jira Migration Configuration Testing



Use this worksheet to verify the expected source configuration elements appear in the sandbox and target applications. Use the “Results” column to note items that are present, missing, or unchanged. Use the information previously gathered in **Worksheet 7: Jira Migration Configuration Assessment**.

Application Details

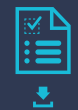
URL:

Projects

Projects

Verify all items exist, or are omitted, as expected. Visit: Admin > Projects

ID	Key	Project Name	Project Lead	Migrate	Scope	Result
Ex:	DEV	Development	Kelly Smith	Yes	All issues	Present (as expected)
Ex:	FIN	Finance	Chris Jones	Yes	JQL: project = FIN and (created > startOfYear() or updated > startOfYear())	Present - more issues than expected



ID	Key	Project Name	Project Lead	Migrate	Scope	Result
Ex:	DEMO	Demo	Kelly Smith	No	n/a	Missing (as expected)
Ex:	TEST	Test	Kelly Smith	Exists	n/a	Unchanged (as expected)
1						
2						
3						
...						

Project Categories

Verify all items exist, or are omitted, as expected. Visit: Admin > Project categories

ID	Project Category Name	Projects Used By	Migrate	Result
Ex:	Development	DEV, QA, APPS	Yes	Present (as expected)
Ex:	Test	TEST, TASKS	No	Missing (as expected)
Ex:	Demo	DEMO	Exists	Unchanged (as expected)
1				
2				
3				
...				

Project Roles

Verify all items exist, or are omitted, as expected. Visit: Admin > System > Project roles

ID	Project Role Name	Projects Used By	Migrate	Result
Ex:	Administrators	n/a	Exists	Unchanged (as expected)
1				
2				
3				
...				



Issue Types

Issue Types

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Issue types and Admin > Issues > Sub-tasks

ID	Issue Type Name	Notes	Migrate	Result
Ex:	Bug	n/a	Exists	Unchanged (as expected)
1				
2				
3				
...				

Issue Type Schemes

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Issue type schemes

ID	Issue Type Scheme Name	Issue Types	Notes	Migrate	Result
Ex:	Default Issue Type Scheme	<ul style="list-style-type: none">• Task• Sub-task• Story (Default)• Bug• Epic	n/a	Exists	Unchanged (as expected)
1					
2					
3					
...					

Workflows

Workflows

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Workflows

Note: Do not migrate inactive workflows

ID	Workflow Name	Notes	Migrate	Result
1				
2				
3				
...				



Workflows Schemes

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Workflow schemes

Note: Do not migrate inactive workflow schemes

ID	Workflow Scheme Name	Workflow > Issue Type	Usage	Migrate	Result
Ex:	Software Simplified Workflow Scheme	<ul style="list-style-type: none">• Approval workflow for Bug > Bug• Simple workflow for Task > Task	<ul style="list-style-type: none">• Dev• QA• Apps	Exists	Unchanged (as expected)
1					
2					
3					
...					

Screens

Screens

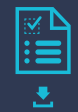
Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Screens

ID	Screen Name	Notes	Migrate	Result
1				
2				
3				
...				

Screen Schemes

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Screen schemes

ID	Screen Scheme Name	Notes	Migrate	Result
1				
2				
3				
...				



Issue Type Screen Schemes

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Issue security schemes

ID	Issue Type Screen Scheme Name	Notes	Migrate	Result
1				
2				
3				
...				

Fields

Fields

Verify all items exist, or are omitted, as expected. Note any conflicting fields which also exist in the target application. Visit: Admin > Issues > Custom fields

ID	Field Name	Notes	Migrate	Result
1				
2				
3				
...				

Field Configurations

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Field configurations

ID	Field Configuration Name	Notes	Migrate	Result
1				
2				
3				
...				



Field Configuration Schemes

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Field configuration schemes

ID	Field Configuration Scheme Name	Notes	Migrate	Result
1				
2				
3				
...				

Priorities

Priorities

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Priorities

ID	Priority Name	Notes	Migrate	Result
Ex:	Highest	n/a	Exists	Unchanged (as expected)
1				
2				
3				
...				

Priority Schemes

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Priority Schemes

ID	Priority Scheme Name	Priorities	Notes	Migrate	Result
1					
2					
3					
...					



Issue Links

Issue Links

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Issue linking

ID	Issues Link Name	Notes	Migrate	Result
Ex:	<i>Relates</i>	<i>n/a</i>	<i>Exists</i>	<i>Unchanged (as expected)</i>
1				
2				
3				
...				

Issue Attributes

Statuses

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Statuses

ID	Status Name	Notes	Migrate	Result
Ex:	<i>Closed</i>	<i>n/a</i>	<i>Exists</i>	<i>Unchanged (as expected)</i>
1				
2				
3				
...				

Resolutions

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Resolutions

ID	Resolution Name	Notes	Migrate	Result
Ex:	<i>Duplicate</i>	<i>n/a</i>	<i>Exists</i>	<i>Unchanged (as expected)</i>
1				
2				
3				
...				



Issue Security Schemes

Issue Security Schemes

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Issue security schemes

ID	Issue Security Scheme Name	Projects	Notes	Migrate	Result
1					
2					
3					
...					

Notifications

Notification Schemes

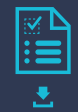
Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Notification schemes

ID	Notification Scheme Name	Projects	Notes	Migrate	Result
1					
2					
3					
...					

Default Notifications

Verify all items exist as expected.

ID	Default Notification Name	Granted To	Notes	Result
Ex:	Issue created	<ul style="list-style-type: none">Current AssigneeReporterAll Watchers	n/a	Unchanged (as expected)
1	Issue Created			
2	Issue Updated			
3	Issue Assigned			
4	Issue Resolved			
5	Issue Closed			
6	Issue Commented			
7	Issue Comment Edited			



10	Issue Deleted			
11	Issue Moved			
12	Work Logged On Issue			
13	Work Started On Issue			
14	Work Stopped On Issue			
15	Issue Worklog Updated			
16	Issue Worklog Deleted			
17	Generic Event			

Permission

Permission Schemes

Verify all items exist, or are omitted, as expected. Visit: Admin > Issues > Permission schemes

ID	Permission Scheme Name	Projects	Notes	Migrate	Result
1					
2					
3					
...					

Default Permissions

Verify all items exist as expected.

ID	Default Permission Name	Granted To	Notes	Result
Ex:	Administer Projects	Project role <ul style="list-style-type: none">Administrators Groupjira-administrators	n/a	Unchanged (as expected)
1	Administer Projects			
2	Browse Projects			
3	Manage Sprints			
4	Service Desk Agent			
5	View Development Tools			
6	View Read-Only Workflow			
7	Assignable User			
8	Assign Issues			



9	Close Issues			
10	Create Issues			
11	Delete Issues			
12	Edit Issues			
13	Link Issues			
14	Modify Reporter			
15	Move Issues			
16	Resolve Issues			
17	Schedule Issues			
18	Set Issue Security			
19	Transition Issues			
20	Manage Watchers			
21	View Voters and Watchers			
22	Add Comments			
23	Delete All Comments			
24	Delete Own Comments			
25	Edit All Comments			
26	Edit Own Comments			
27	Create Attachments			
28	Delete All Attachments			
29	Delete Own Attachments			
30	Delete All Worklogs			
31	Delete Own Worklogs			
32	Edit All Worklogs			
33	Edit Own Worklogs			
34	Work On Issues			



User Objects

Note: Only migrate objects associated with migrated projects listed above. Do not migrate objects with errors (e.g. boards with deleted filters or filters with invalid JQL). Use the Integrity Check for Jira app to detect and address errors. Read more: botron.io/amp-ic

Filters

Verify all items exist, or are omitted, as expected. Visit: Admin > System > Shared filters

ID	Filter Name	Notes	Migrate	Result
1				
2				
3				
...				

Dashboards

Verify all items exist, or are omitted, as expected. Visit: Admin > System > Shared dashboards

ID	Dashboard Name	Filters	Notes	Migrate	Result
Ex:	Development Dashboard	<ul style="list-style-type: none">Filter=12234Filter=10587	n/a	Exists	Unchanged (as expected)
1					
2					
3					
...					

Boards

Verify all items exist, or are omitted, as expected. Visit: Boards > View all boards

ID	Board Name	Filter	Notes	Migrate	Result
1					
2					
3					
...					



Customization

Apps

Verify all items exist, or are omitted, as expected. Also see **Worksheet 2: Jira Apps and Customizations** and **Worksheet 10: Jira App Decisions**. Visit: Admin > Manage apps > Manage apps

ID	App Name	Notes	Migrate	Result
1				
2				
3				
...				

Application Links

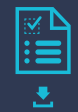
Verify all items exist, or are omitted, as expected. Visit: Admin > Applications > Application links

ID	Application Link Name	Notes	Migrate	Result
Ex:	Confluence	https://confluence.domain.com	Yes	Manually added
1				
2				
3				
...				

Connections, Integrations, WebHooks, Listeners, and other customizations

Verify all items exist, or are omitted, as expected. Visit: Admin > Applications, Admin > System > WebHooks, Admin > System > Listeners, etc.

ID	Customization Name	Notes	Migrate	Result
1				
2				
3				
...				



User Management

Users

Verify all items exist, or are omitted, as expected. Visit: Admin > User management

ID	User Name	Notes	Migrate	Result
1				
2				
3				
...				

Groups

Verify all items exist, or are omitted, as expected. Visit: Admin > User management > Groups

ID	Group Name	Members	Notes	Migrate	Result
Ex:	jira-administrators	Kelly Smith	n/a	Exists	Unchanged (as expected)
1					
2					
3					
...					

Global Permissions

Verify all items exist, or are omitted, as expected. Visit: Admin > System > Global permissions

ID	Global Permission Name	Granted To	Notes	Migrate	Result
Ex:	Jira System Administrators	Group: • jira-administrators	n/a	Exists	Unchanged (as expected)
1	Jira System Administrators				
2	Jira Administrators				
3	Browse Users				
4	Create Shared Objects				
5	Manage Group Filter Subscriptions				
6	Bulk Change				
7	Browse Archive				

Functional Tests

After the migration team has verified that the expected data exists, the next test type is a functional check. Does Jira behave like it did before and/or as expected? Are users able to access their issues and objects?



Tip: Ask end users help complete functional testing. Ask them to create additional test cases for their specific important scenarios. Encourage them to go through their daily work routines, which will help detect additional problems. For example, instead of simply spot checking a board, use it by ranking issues, transitioning issues, etc.

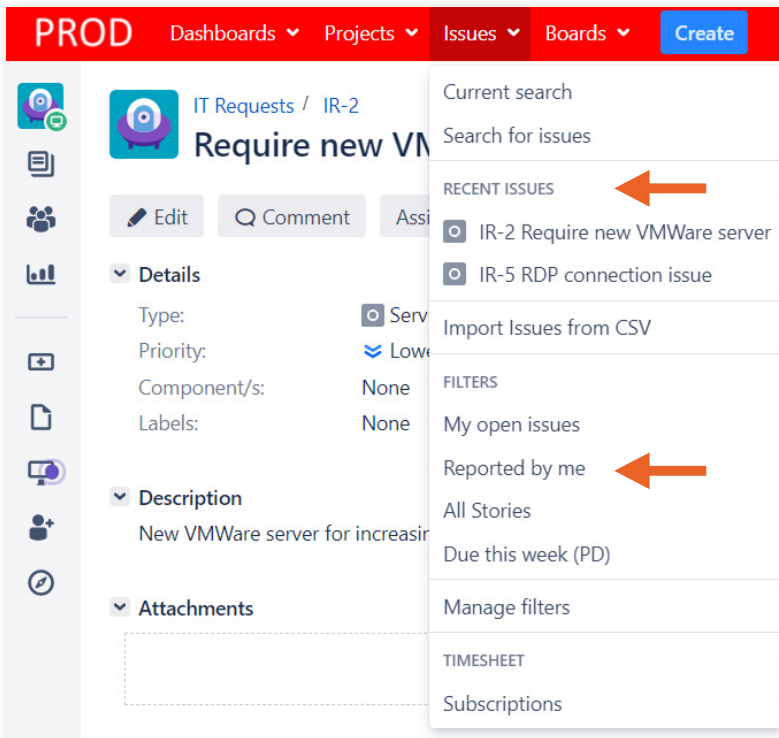


Image 40: Recent Issues and Filters



Tips:

- Ask end users to bookmark their most frequently accessed areas, boards, and favorite objects. Any “recently accessed” history, like in navigation menus, for example, is not migrated. The screenshot shows shortcuts to recently visited issues and filters.
- Create test accounts and set up specific sample scenarios, so users know exactly what to test and look for. Consider creating an account to test as an application administrator, a project administrator, and a project role or group member with specific abilities (e.g., developers).

Worksheet 16: Jira Migration Test Cases



Use this worksheet to craft test cases to verify migrated Jira data. Get end users to help! Create test accounts and sample scenarios as needed. For each test case, record any specific details needed to perform the test. Fill in any variables noted with "<>" symbols. Perform these tests after both the test and final migrations. The results should be the same.

Application Details

URL:			
Environment:	<input type="checkbox"/> Test	<input type="checkbox"/> Staging	<input type="checkbox"/> Target
Testing Start Date:		Testing End Date:	

Test Accounts

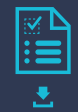
ID	Username	Password	Role	Notes
1	test-regular	<password>	Regular user	General project access
2	test-dev	<password>	Regular user	"jira-developers" group
3	test-project-admin	<password>	Project-level admin	DEV, QA, TEST projects
...				

Test Cases

ID	Test	Details	Tester	Results
Ex:	Is the application accessible to non-logged in users?	<ul style="list-style-type: none">Visit: <URL> before logging in	Rachel Wright	Inaccessible (as expected)
1	Is the application accessible to non-logged in users?	<ul style="list-style-type: none">Visit: <https://your-jira-url.com> before logging in		



ID	Test	Details	Tester	Results
2	Does the System Dashboard contain expected information?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/secure/Dashboard.jspa>• Gadget count: <number>• Issues in <gadget name> gadget: <number>		
3	Do all expected projects exist?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/secure/BrowseProjects.jspa>• Project count: <number>		
4	Do the expected issues exist for each project?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/secure/BrowseProjects.jspa>• Project <project-name> issue count: <number>• Project <project-name> issue count: <number>• Project <project-name> issue count: <number>• ...		
5	Does an existing issue contain the expected data?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/browse/ISSUE-1>• Summary: <summary>• Component: <component>• Reporter: <reporter>• Attachment: <filename>• Link: <issue-id> linked as <link-type>• Status: <status-name>• Comments: <number>• Work logged: <number> hours		
6	Can a new issue be created or cloned?	<ul style="list-style-type: none">• Project: <project-name>• Issue type: <issue-type>		
7	Can an issue's Description be changed?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/browse/ISSUE-1>• Existing description: <description>• New description: <description>		
8	Can an issue's assignee be changed?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/browse/ISSUE-1>• Existing assignee: <assignee>• New assignee: <assignee>		
9	Can an issue's status be changed?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/browse/ISSUE-1>• Existing status: <status-name>• New status: <status-name>		



ID	Test	Details	Tester	Results
10	Can an issue be transitioned from its first to its final status?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/browse/ISSUE-1>• Workflow: <status-name> → <status-name> → <status-name>• Validator: The <transition-name> transition, in the <status-name> status, is restricted to members of the <group-name> group• Post Function: The <script-name> script is run on the <transition-name> transition in the <status-name> status		
11	Do all expected custom fields exist?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/browse/ISSUE-1>• Field: <field-list>• ...		
12	Do all personal filters exist?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/secure/ManageFilters.jspa>• Filter count: <number>		
13	Do filters return the expected results?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/issues/?filter=12345>• Issue count: <number>• Viewers: <user-list>• Editors: <user-list>		
14	Do all personal dashboards exist?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/secure/ConfigurePortalPages.jspa>• Dashboard count: <number>		
15	Do dashboards show the expected results?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/secure/Dashboard.jspa?selectPagelId=12345>• Gadget count: <number>• Issues in <gadget name> gadget: <number>• Issues in <gadget name> gadget: <number>• Issues in <gadget name> gadget: <number>• ...• Viewers: <user-list>• Editors: <user-list>		
16	Do all personal boards exist?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/secure/ManageRapidViews.jspa>• Board count: <number>		



ID	Test	Details	Tester	Results
17	Do boards show the expected results?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/secure/RapidBoard.jspa?rapidView=12345>• Filter: <https://your-jira-url.com/issues/?filter=12345>• Issue count: <number>• Columns: <column-names>• Swimlanes: <swimlane-names>• Estimates: <number> hours on <issue-id>• Ranking: <issue-id> is above <issue-id>• Admins: <user-list>		
18	Are all group permissions as expected?	<ul style="list-style-type: none">• Group: <group-name>• Permissions: <permissions-list>		
19	Are all role permissions as expected?	<ul style="list-style-type: none">• Role: <role-name>• Project: <project-name>• Permissions: <permissions-list>		
20	Are all application links functioning as expected?	<ul style="list-style-type: none">• Visit: <https://your-jira-url.com/browse/ISSUE-1>• Confluence link: <https://your-confluence-url.com/spaces/KEY/pages/page-name>		
21	Are all apps functioning as expected?	<ul style="list-style-type: none">• <enter-details-for-each-app>		
22	Are all integrations functioning as expected?	<ul style="list-style-type: none">• <enter-details-for-each-integration>		
23	Are all customizations functioning as expected?	<ul style="list-style-type: none">• <enter-details-for-each-customization>		
...				



Tips:

- Always test and fix connections with other tools. When Jira can't connect to Bitbucket, for example, users cannot see code commit data.
- Use the Configuration Manager for Jira app to create a list of all source and target IDs. This is useful for reconnecting external integrations (e.g., Tableau) and updating scripts. Check the "Persist deployed snapshot to current system native ID mappings" checkbox on the app's settings page at: Admin > Configuration Manager > Settings. A downloadable .json file with the mapping information is created during the snapshot deployment step.

6.0 After Migration

Congratulations on completing the migration! That was likely a large effort. Take a moment to celebrate and thank the migration team. Hopefully, there were few support requests to address and it's time to work on post-migration duties!

Post-migration Stats

Ideally, you did any needed clean-up activities before the migration, but if you didn't, now is the time to address them.

Here's a real example of the before and after impact of merging a clean target application with a messy source application. Which Jira area would you clean up first?

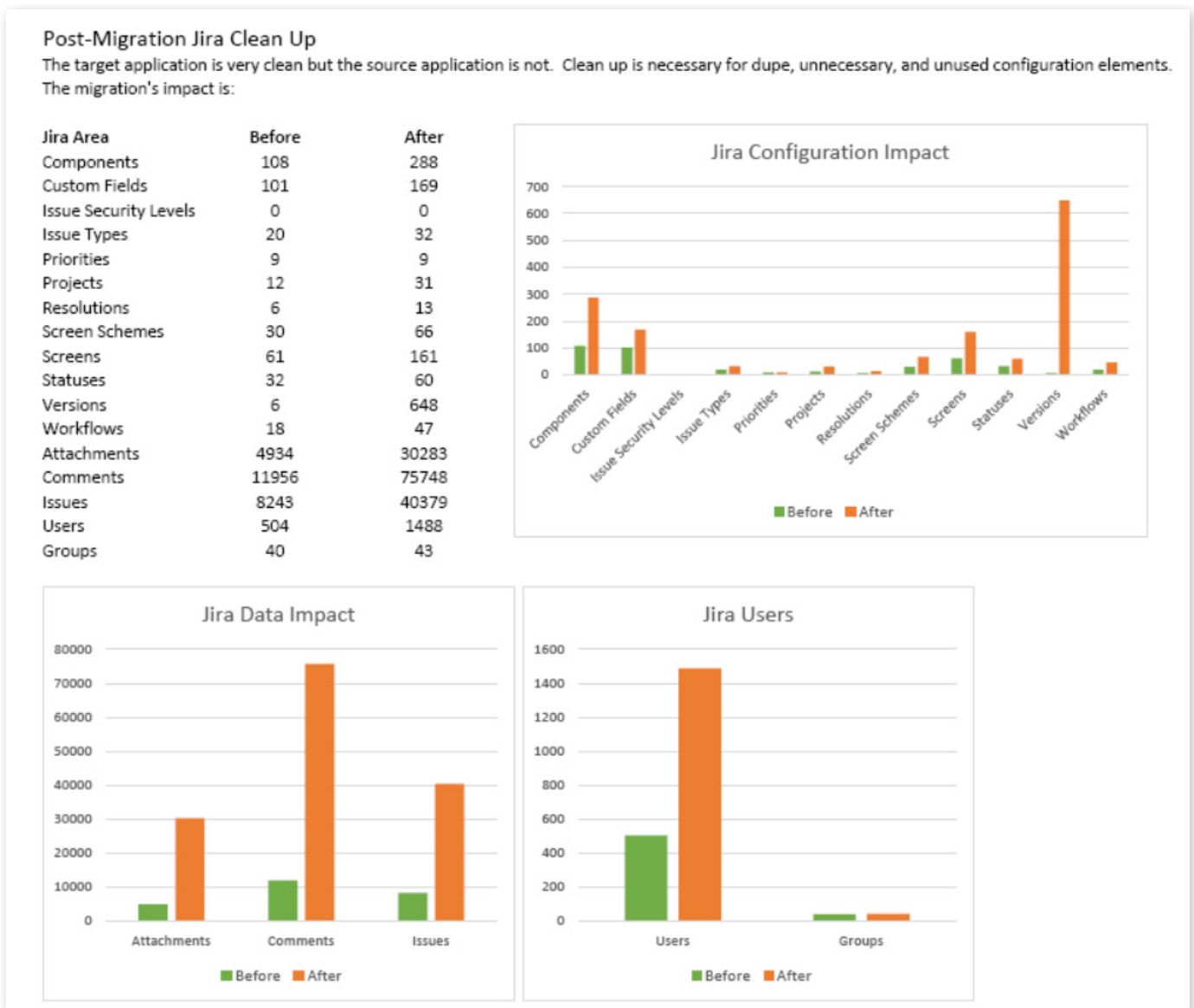


Image 41: Before and After Migration Stats

Regardless of when and how often you clean up, it's useful to compare the growth or reduction of key application settings throughout the years. By regularly compiling stats, you can predict growth and future needs.

Source Application Decommission

Now that the target application is ready for users, it's time to slowly decommission the source application. Do the following as appropriate:

- Prevent source application changes and make all source data “read only”.
- Add an announcement banner directing users from the source application to the target application (Check out Announcer for Jira by Wittified: botron.io/amp-announcer).
- Create a URL redirect from the source application to the target application.
- Disable source login ability.
- Take the source application offline.

A Jira administrator's job is never done! After the source application is decommissioned, it's time to focus on ways to keep the target application healthy now and in the future. Here are some areas to address. The Configuration Manager for Jira app can help with all of them and other long-term maintenance needs. Read more: botron.io/CMJ-use-cases

6.1 Test Environment

The test or staging environment served you well during the migration. Keep it up and running for the long term. Use it to test future impactful efforts like vetting and installing third-party apps, major configuration changes, or periodic clean-up efforts. Always make sure there's a place to experiment and test changes without impacting real data and users. Verify changes in a staging environment before promoting them to production.

For Jira Cloud, you can sync issues and settings using an app from the Atlassian Marketplace.

For Jira Server and Data Center, use Configuration Manager for Jira to automate the promotion process and keep changes in sync between environments. Simply take a “Project Configuration” snapshot to deploy small configuration changes or a “System Configuration” snapshot to sync an entire application configuration. Here's an example:

System Configuration Snapshot

Use the “System Configuration” snapshot type in the Configuration Manager for Jira app to quickly refresh the configuration of a staging environment.

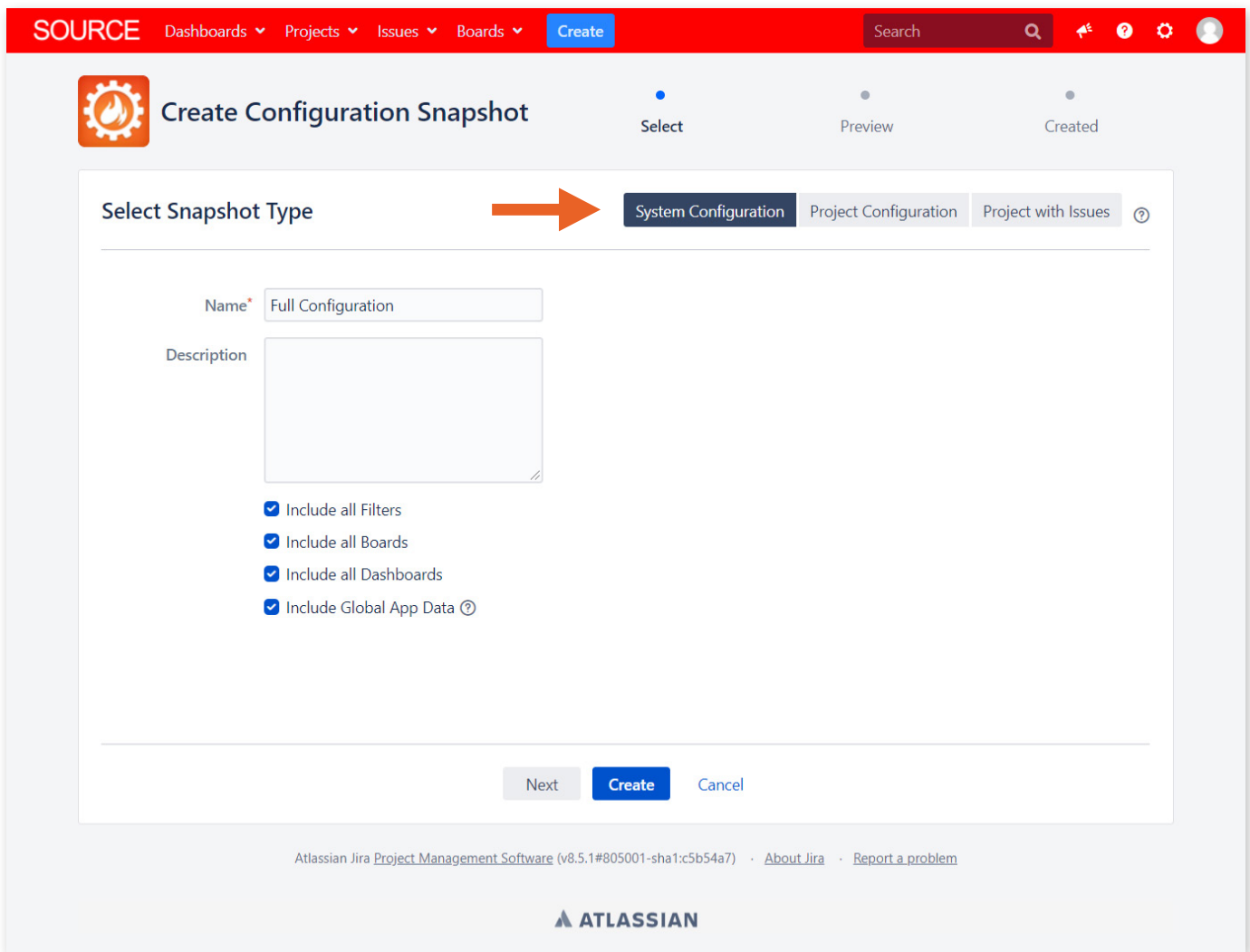


Image 42: Configuration Manager for Jira System Configuration Snapshot

In the screenshot above, I've selected the "System Configuration" snapshot type, entered a name, and checked all the object selection boxes.

"Global App Data" is information from installed apps. Botron has partnered with some of the most popular Atlassian Marketplace app vendors to migrate app data using Configuration Manager for Jira. Read more about global app data at: botron.io/CMJ-migrate-global-app

Learn about integration partners in the [Apps Migration](#) section.

Deploy System Configuration

In Configuration Manager for Jira, click the "Deploy" link in the sidebar. On the *Deploy Configuration Snapshot* page, choose a snapshot and click the "Deploy" link in the "Actions" column on the right.

The next page shows two deployment modes: "Merge Configuration" and "Restore Configuration".

When needed data exists in the target application, use the **“Merge Configuration”** mode. This is the safest mode.

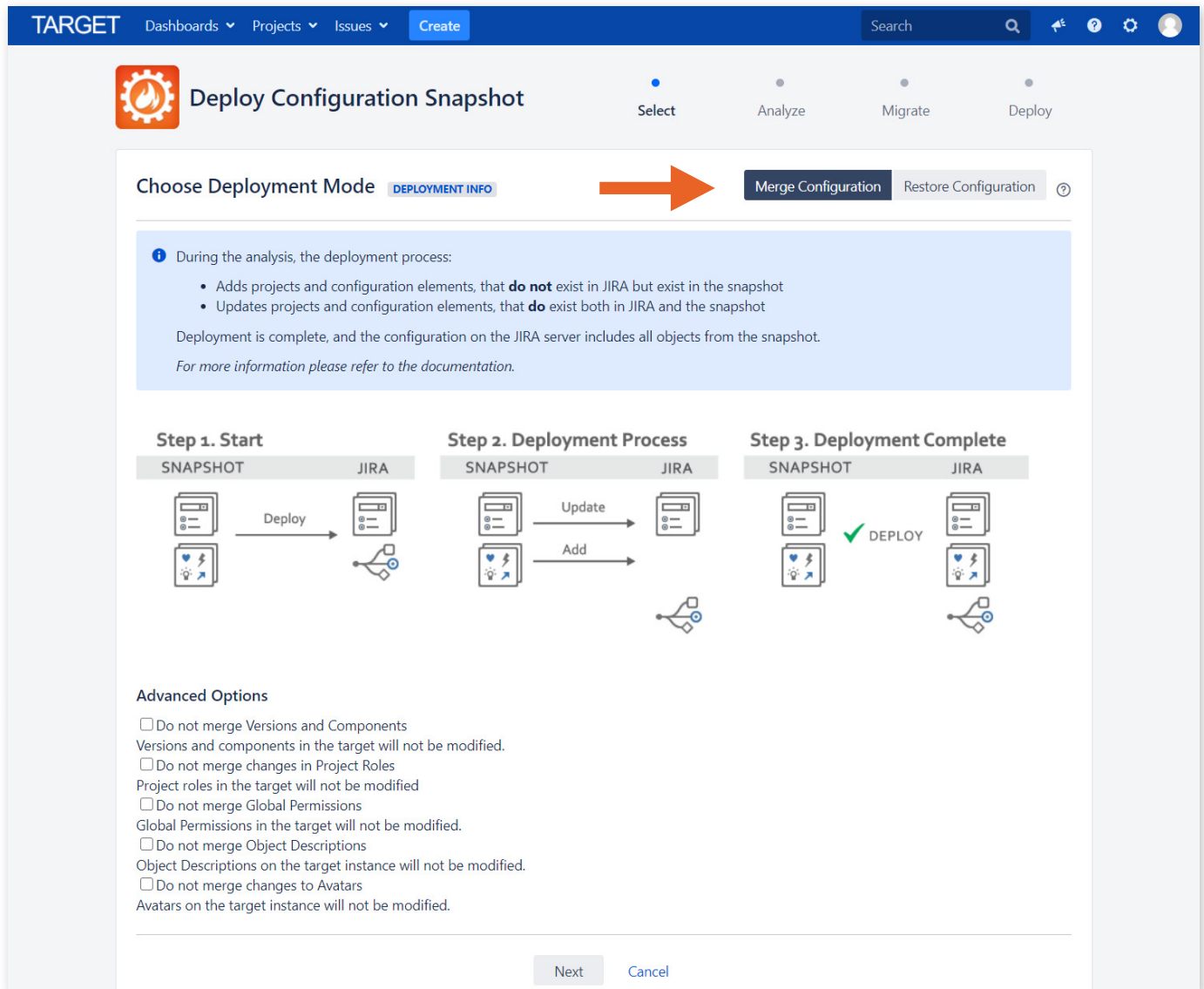


Image 43: Configuration Manager for Jira Merge Deployment Mode

This means:

- All objects from the snapshot that are not present in the target application are added.
- Objects present in both applications are modified by the objects in the snapshot.

For example:

- A project called “Library” is present in the snapshot *but not* in the target application
Result: The “Library” project will be created (added) using the configuration in the snapshot.
- A project called “Marketing” is present in the snapshot *and* in the target application.
Result: The configuration of the “Marketing” project is replaced by the snapshot configuration.

- A project called “Human Resources” is *not* present in the snapshot but *is* present in the target application.

Result: The “Human Resources” project will not be changed in any way.

To remove data, restore data, or when no data exists in the target application, use the **“Restore Configuration”** mode. This mode is not recommended in a production environment unless you specifically intend to delete data.

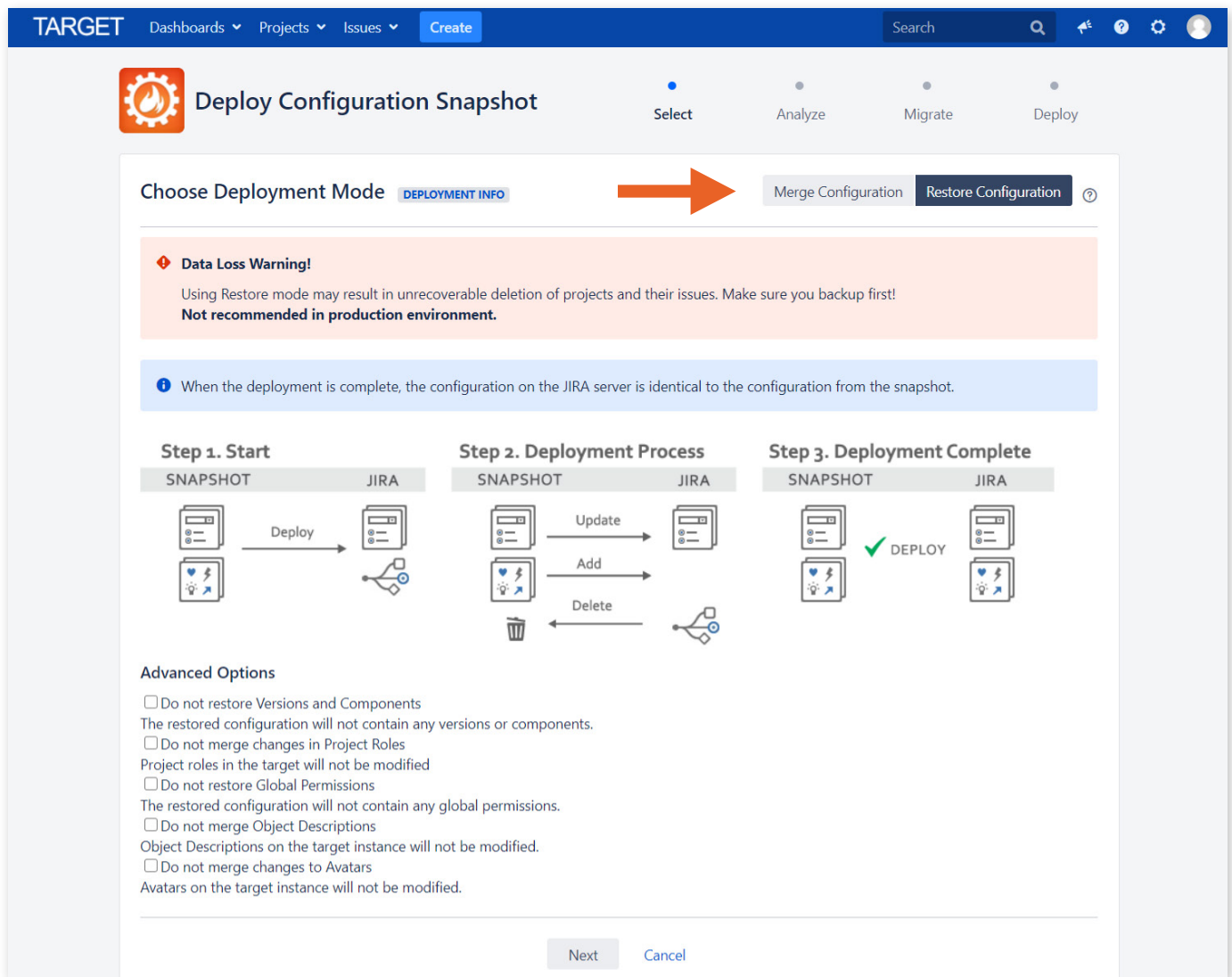


Image 44: Configuration Manager for Jira Restore Deployment Mode

This means:

- The configuration of the target Jira instance is *replaced* by the configuration in the snapshot.
- Objects present in the snapshot are added.
- Objects present in both applications are *replaced* by the objects in the snapshot.

- Objects not in the snapshot are *deleted* from the target application.
- After the deployment, the target application's configuration is identical to the snapshot's configuration.

For example:

- A project called "Library" is present in the snapshot *but not* in the target application.
Result: The "Library" project will be created (added) using the configuration in the snapshot.
- A project called "Marketing" is present in the snapshot *and* in the target application.
Result: The configuration of the "Marketing" project is replaced by the snapshot configuration.
- A project called "Human Resources" is *not* present in the snapshot but *is* present in the target application.
Result: The "Human Resources" project is *deleted*.



Tip: The "Restore Configuration" mode deletes projects and issues not included in the snapshot. **Deletions cannot be reversed.** Always back up the target application prior to deployment.

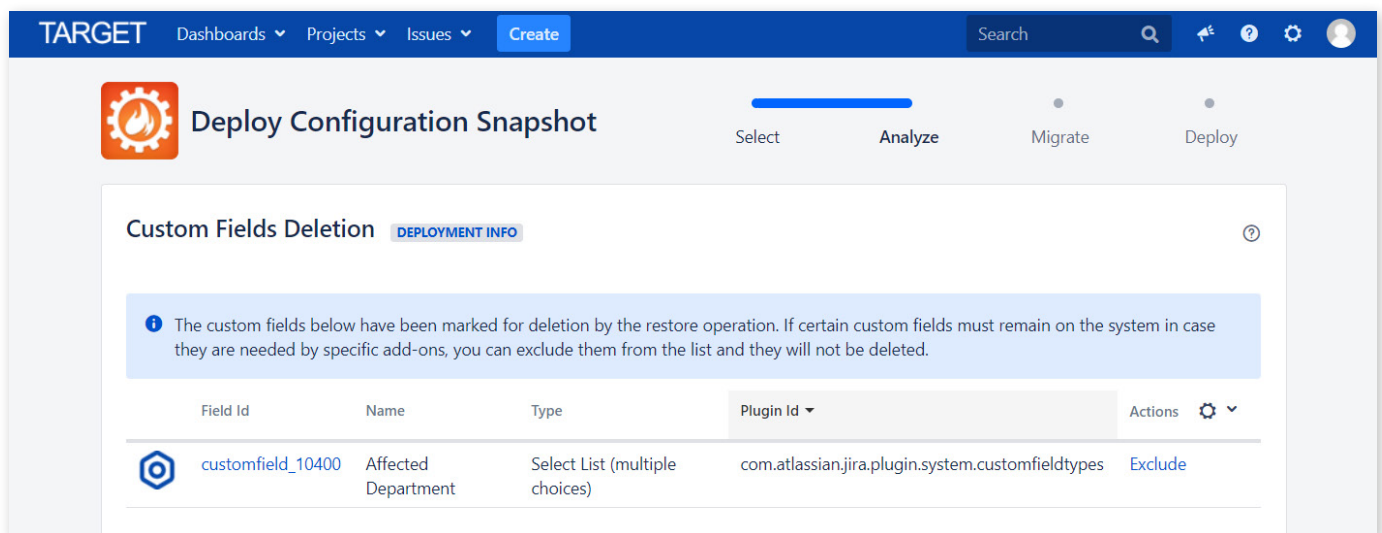


Image 45: Configuration Manager for Jira Field Deletion Warning

The screenshot above shows the "Affected Department" custom field, of type "Select List (multiple choices)" will be deleted because this exact field does not exist in the source application.

Next, handle any conflicts and complete the remaining steps in the deployment wizard.

Read more about managing multiple environments here: botron.io/CMJ-test-stage-prod

Additionally, read about cloning here: botron.io/CMJ-jira-clones



Tip: Have a lot of projects? Automate the snapshot and deployment process using Jira's REST API to stagger snapshot creation and deployment. Automation also ensures the proper settings are consistently selected and eliminates user error. Read more: botron.io/CMJ-REST-API

6.2 Project Templates

If you just did a clean-up, the last thing you want to do is mess the configuration up again! An easy way to prevent over-customization is to create project templates. I suggest starting with three project templates: one for development projects, one for support projects, and one for task-based projects.

Let's use a common development configuration as an example. Here's what the settings might look like:

- An Issue Type Scheme that includes the Epic, Story, Bug, Task, and Sub-task issue types.
- Screens that contain dev-specific fields like Fix Version, Affects Version, Story Points, and Sprint.
- A special workflow for Bugs with a validation step.

Once you've created project standards, document the settings so users know what features are included. Apply the standard configuration when a new project is requested. Don't create custom configuration elements until there's a real business need.

The Configuration Manager for Jira app makes templating easy. Simply take a "Project Configuration" snapshot of an existing project using the desired standard settings. Then use the "New Project" deployment mode to create a new project with the exact same configuration. The snapshot includes all project configuration elements, plus filters, dashboards, and boards, if desired.

Create Configuration Snapshot

Select Filters Boards Dashboards Apps Preview Created

Select Snapshot Type

System Configuration **Project Configuration** Project with Issues ?

Name* Development

Project(s)* Development x v

☒ Include custom fields with value in at least one issue

Description

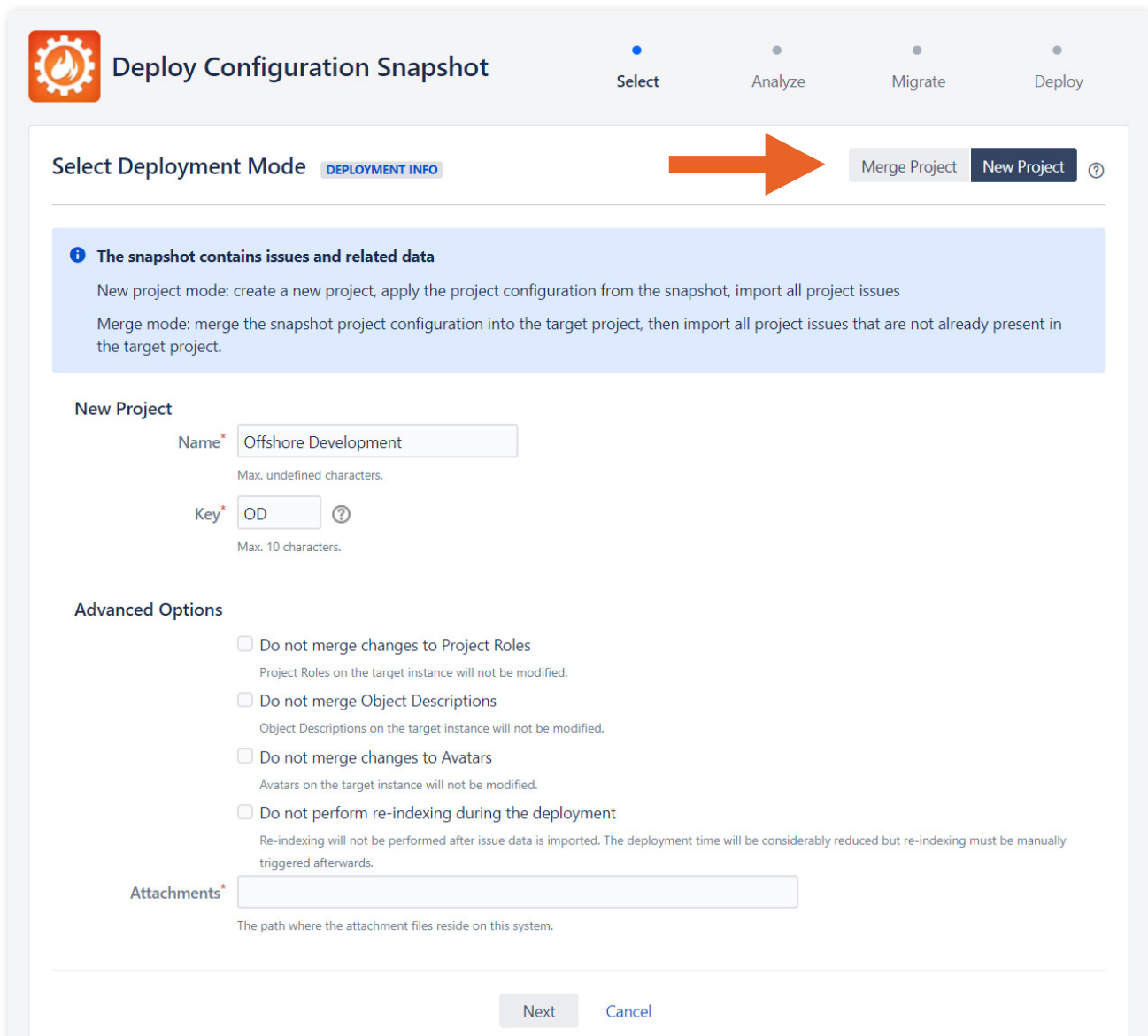
☐ Include Project Filters

☐ Include Project Boards

Next Create Cancel

Image 46: Configuration Manager for Jira "Project Configuration" Snapshot Mode

Create a snapshot of the "Development" project, as shown in the screenshot above.



Deploy Configuration Snapshot

Select Analyze Migrate Deploy

Select Deployment Mode **DEPLOYMENT INFO** Merge Project **New Project** ?

The snapshot contains issues and related data

New project mode: create a new project, apply the project configuration from the snapshot, import all project issues

Merge mode: merge the snapshot project configuration into the target project, then import all project issues that are not already present in the target project.

New Project

Name* Offshore Development
Max. undefined characters.

Key* OD ?
Max. 10 characters.

Advanced Options

- ☐ Do not merge changes to Project Roles
Project Roles on the target instance will not be modified.
- ☐ Do not merge Object Descriptions
Object Descriptions on the target instance will not be modified.
- ☐ Do not merge changes to Avatars
Avatars on the target instance will not be modified.
- ☐ Do not perform re-indexing during the deployment
Re-indexing will not be performed after issue data is imported. The deployment time will be considerably reduced but re-indexing must be manually triggered afterwards.

Attachments*
The path where the attachment files reside on this system.

Next Cancel

Image 47: Configuration Manager for Jira “New Project” Deployment Mode

Deploy the “Development” snapshot as a new project called “Offshore Development”. The new project will have the exact same settings as the existing project.

Read more about project templating with Configuration Manager for Jira at: botron.io/CMJ-project-template



Tip: Appfire (Botron’s parent company) has two additional apps that help delegate project administration and creation to trusted users. Check out Wittified’s Delegated Project Admin Pro for Jira at botron.io/amp-admin and Delegated Project Creator for Jira at botron.io/amp-dpc.

6.3 Disaster Recovery and Backup

What is the disaster recovery plan? How quickly can you access backups or stand up a new application in an emergency? Use Configuration Manager for Jira, in addition to the regular database backup procedure, to quickly deploy the “last known good” configuration. Take regular snapshots for safekeeping. Automate the snapshot process with the REST API. Read more:

botron.io/CMJ-project-backup-CMDB



Tip: Store snapshots in a source control application.

See also the [Back Up](#) section in the “Migration Process” chapter.

6.4 Archive Strategy

Jira Data Center and Jira Cloud Premium have internal archive capabilities. But what if you have a different Jira type or need to move data to a separate or “archival” application? Configuration Manager for Jira handles this case and incremental archival. Read more about archive options at: botron.io/CMJ-project-archive and botron.io/CMJ-incremental-archiving

See also the [Archiving Projects](#) section in the “Migration Process” chapter.

6.5 Post-Migration Training

After the migration, give the side-by-side comparison demos you previously gave again, and this time, also include any additional changes users should expect.

Start with the sample **Jira Server and Jira Data Center for Application Administrators** and **Jira Cloud for Users** “Demo Agendas” in the [Migration Considerations](#) section and add any of the following additional items needed:

- URL changes
- Process differences:
 - e.g., the final status for all workflows changed from “Closed” to “Done”
 - e.g., the “Prioritization Board” is now called “Strategic Company Initiatives”
- Login differences:
 - e.g., the username format changed from “first.last” to “first.last@domain.com”
- Data differences:
 - e.g., the following projects were archived: <project-list>
 - e.g., the following projects were not migrated: <project-list>



Tip: Remind users to check their filters, dashboards, and boards. Specify where to report any questions or problems encountered.



7.0 Q&A

Botron Atlassian Apps (an Appfire company) is the creator of Configuration Manager for Jira and other helpful apps for Jira administrators. They understand the intricacies of Jira configuration and have completed many migrations for organizations of all sizes. Their migration tips and stories are shared below.

7.1 What are the worst migration mistakes?

Poor Planning

Organizations often rush to migrate without performing enough pre-migration analysis. They don't take the necessary time for due diligence. This often creates sub-optimal results. When proper analysis isn't done, impactful setting differences are missed, data is mistakenly omitted, and Jira doesn't function the way users expect.

When security and permission settings aren't handled properly, users don't see data they expect. Users can't find their issues or objects, panic, and file lots of emergency support requests. Sometimes the reverse is true — users have access to data they *shouldn't* see. These scenarios are easily avoided with a proper security analysis.

At one company, the attachment size limit was considerably smaller in the target application. Users were not able to attach their files and quickly found alternate ways to store data. This led to important information stored in multiple locations, which was undesirable.

If email settings aren't considered, end users may receive more or fewer notifications than expected. Always check notification schemes, workflows that send email, and email addresses embedded in Jira or Jira Service Management projects.

Organizations don't always have a plan to handle data from third-party apps. This needs to be part of the up-front planning process, not considered at the end of the migration.

Poor Communication

Stakeholders don't always do a good job communicating and managing expectations. Make sure the migration process isn't a surprise to users! At a minimum, users need to verify that their data is as expected in the staging and production environments. Involve users early and often.

Poor Wind-down

When the source application is not locked down after the migration, users can log in to the old location and modify data, and there is no longer a single source of truth.

Other Challenges

Sometimes organizations use Jira in ways that create additional migration challenges. For example, one company used hard-coded URLs instead of the built-in issue linking function. All the issue links broke and had to be updated in the database.

Another company had over a million diagrams statically linked to Jira issues. Issue keys and issue IDs changed in the target application, so scripting a way to update the links was challenging.

In one application, the re-index process took two days to complete! Luckily, this was discovered during testing. The migration planning stopped so the application could be cleaned up to reduce the index size. The migration testing continued after a manageable re-index duration was obtained.

7.2 What is the biggest migration misconception?

No Strategy Needed

You need to determine the overall goals before you start. Is the goal to:

- Have the target application behave exactly the way it does in the source application?
- Migrate all data “as is”?
- Only migrate development project data?
- Give users different abilities than they had before?
- Clean up the configuration before the migration or clean up after?

Determine goals in the beginning and make sure the strategy supports them. Without goals, there’s no way to measure success.

Migration is Simple

The biggest migration misconception is that migration or consolidation is an easy process. You simply pick up your data and move it, right? It’s rarely that simple. Application settings are usually more complex than you think. There are probably settings you forgot about or don’t even know exist.

Ideally, you’ll predict conflicts before they arise, quickly mitigate issues when they occur, and be ready to compromise when needed.

Migration is Quick

Organizations often underestimate time and complexity. Having many users, schemes, and apps always equals more migration time. Allow more time than you think you’ll need, especially in the planning and testing phases.

Be sure to take your time, do a trial run (or many runs), and don't migrate to the production application until everything is perfect.

7.3 How can you manage risk during the migration process?

Focus on Security

Security and permissions are especially important but sometimes overlooked. For example, the source application is *private*, but the target application is mistakenly *public*! Or the source and target applications have the same group name, but their usage or abilities differ. Without careful analysis, security differences can have undesired consequences. Review the security model and global permissions in depth and look for differences and potential conflicts. Use the “Power Admin” feature in the Configuration Manager for Jira app to analyze group usage across multiple Jira locations at the same time.

The screenshot shows the 'jira-administrators' group details in the Jira Power Admin interface. The group is of type 'Group' and has 3 members. The summary section shows the group is configured in 7 projects, subscribed to 0 filters, accessing 4 filters, and accessing 0 dashboards. The security section shows 32 permissions, 6 global permissions, and 0 issue security schemes. The notifications section shows 0 events. The workflows section shows 0 workflows. The JQL section shows 0 agile boards and 0 filters. The details section shows a table of permissions.

Name	Type	ID	Projects	Locations	Actions
Delete Own Worklogs Ability to delete own worklogs made on issues.	Permission	DELETE_OWN_WORKLOGS	CONFIGURED IN 2 PROJECTS		
Create Issues Ability to create issues.	Permission	CREATE_ISSUES	CONFIGURED IN 2 PROJECTS		
Delete Own Comments Ability to delete own comments made on issues.	Permission	DELETE_OWN_COMMENTS	CONFIGURED IN 2 PROJECTS		
Work On Issues Ability to log work done against an issue. Only useful if Time Tracking is turned on.	Permission	WORK_ON_ISSUES	CONFIGURED IN 2 PROJECTS		
Modify Reporter Ability to modify the reporter when creating or editing an issue.	Permission	MODIFY_REPORTER	CONFIGURED IN 7 PROJECTS		
Manage Watchers Ability to manage the watchers of an issue.	Permission	MANAGE_WATCHERS	CONFIGURED IN 7 PROJECTS		
Edit Issues Ability to edit issues.	Permission	EDIT_ISSUES	CONFIGURED IN 3 PROJECTS		
View Voters and Watchers Ability to view the voters and watchers of an issue.	Permission	VIEW_VOTERS_AND_WATCHERS	CONFIGURED IN 2 PROJECTS		
Add Comments Ability to comment on issues.	Permission	ADD_COMMENTS	CONFIGURED IN 2 PROJECTS		
Edit Own Comments Ability to edit own comments made on issues.	Permission	EDIT_OWN_COMMENTS	CONFIGURED IN 2 PROJECTS		

Page 1 of 10. Total Items 38 | Items per page 10

Image 48: Power Admin Group Details Page

The screenshot shows the “jira-administrators” group is used in seven projects and four filters. The group has 32 permissions and six are global. The group is not used in filter subscriptions, dashboards, notification schemes, or workflows.

Alternatively, you can gather group usage information manually, but it’s tedious and error prone.

Involve Users

Make sure end users are aware of the migration project and have them participate in the testing process. Have a good communication plan that sets expectations, deadlines, and lists everything users should check. For example, users should always verify their most important boards and dashboards work as expected.

Test and Test Again

Test the migration process multiple times in a staging environment. Know exactly how long each process takes to complete. Pay special attention to lengthy steps like re-indexing, copying data, and copying attachments.

Use Tools

Tools are always safer than custom scripts, direct database changes, or the native Jira backup/restore utility. These options are error prone and don’t provide roll back capabilities.

Use the “Integrity Check” feature in the Configuration Manager for Jira app to detect and address problems. Don’t migrate broken objects!

A common example is a broken board. If the filter powering a board is deleted, the board is no longer usable. The board loads strangely and shows the error in the screenshot below. The board needs to be deleted, but there’s no way to do it in the Jira UI.

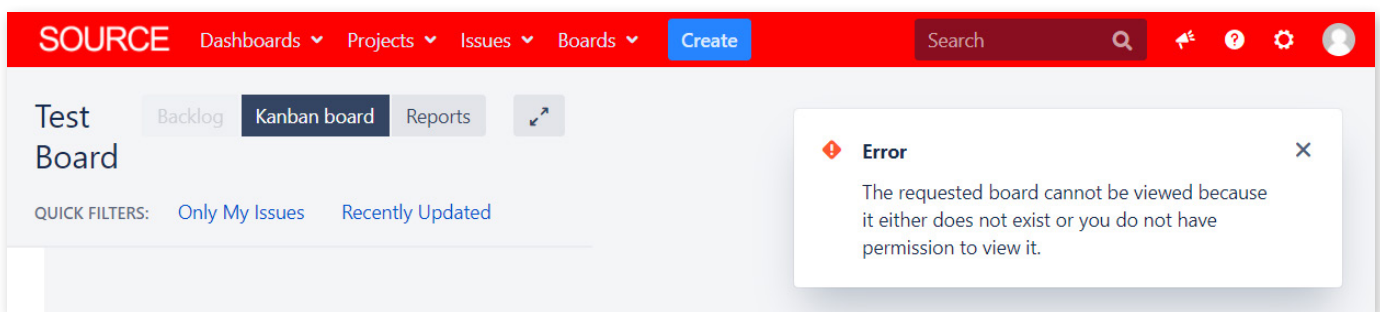


Image 49: Broken Board

Until recently, the only way to delete the board was directly in the database. Making database changes is always risky, however. Instead, use the new “Quick fix” function in the Integrity Check for Jira app to solve the problem. See other common “Quick fix” actions at:

botron.io/CMJ-quick-fix



Configuration Integrity Check



Select integrity check options:

- ☒ Check all filters
- ☒ Check all agile boards
- ☒ Check all dashboards

Run Integrity Check

Cancel Integrity Check

[Download CSV](#) [Download JSON](#)



Failure

377 configuration elements were checked and **1** violations were found. See details in the table below.

	Type	Description	
▼	Missing object	Agile Board 'Test Board (pso012)' refers to the missing filter 10200 .	Quick fix available

Details:

Agile Board 'Test Board (pso012)' refers to the missing filter **10200**.

Solution:

Modify agile Board 'Test Board (pso012)' to point to an existing filter or remove the reference to the missing filter. This error may be caused by an inconsistency in the Jira database and may not be evident in the Jira UI.

Location:

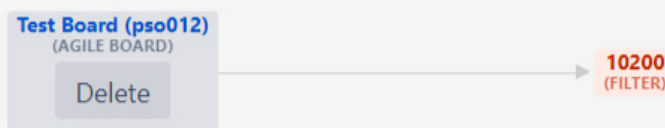


Image 50: Integrity Check for Jira Results

In the screenshot above, click the “Delete” button to remove the board and solve the problem.

Conclusion

You've reached the end of the book, which means you should have a good understanding of the scope of a migration. Was it more involved than you thought? It always is for me. If you take nothing else away from this book, I hope it's to spend as much time as possible in the *planning* and *testing* phases. Your upfront effort will determine the success of your migration. I know there's a lot to do and many decisions to make. Don't be overwhelmed; just complete one step at a time. I know you can do it!

There's a lot of great migration information available, but until now, it was all scattered in different locations. I wrote this book to create one master resource to answer all your major migration questions and provide a comprehensive plan to follow. I hope I've accomplished that and put you on the right path. If you have Jira questions, I'd love to hear from you.

My contact information is below. Good luck with your Jira journey, and I'll see you online.

- Rachel

Email: rwright@jirastrategy.com

Linkedin: [linkedin.com/in/rlwww](https://www.linkedin.com/in/rlwww)

Twitter: [@rlw_www](https://twitter.com/rlw_www)

Worksheets

Use the following worksheets to help you plan and execute a migration. Worksheets are grouped by chapter and listed in order of appearance.



Note: All the worksheets in this book are compiled in separate, editable files for you to download, customize, or print. Feel free to add your branding, change the information, or remove sections that don't apply. Use them all or just the ones you need. Store this information in Google Docs, Confluence, or wherever your other Jira documentation is located. Download the worksheets at botron.io/worksheets

1.0 Introduction

n/a

2.0 Jira Deployment Differences

2.3 [Worksheet 1: Jira Migration Survey](#)

2.4 [Worksheet 2: Jira Apps and Customizations](#)

3.0 Migration Options

n/a

4.0 Governance, Approval and Change Management

4.1 [Worksheet 3: Jira Migration Stakeholders](#)

4.1 [Worksheet 4: Jira Migration Team](#)

5.0 Migration Process

5.1 [Worksheet 5: Jira Migration Checklist](#)

5.2 [Worksheet 6: Jira Migration Decisions List](#)

5.2 [Worksheet 7: Jira Migration Configuration Assessment](#)

5.5 [Worksheet 8: Jira Migration Timeline](#)

5.5 [Worksheet 9: Jira Migration Communication Plan](#)

5.5 [Worksheet 10: Jira Migration App Decisions](#)

5.6 [Worksheet 11: Jira Migration Configuration Freeze](#)

5.6 [Worksheet 12: Jira Migration Sample Launch Plan](#)

5.7 [Worksheet 13: Jira Migration User Account Strategy](#)

5.8 [Worksheet 14: Jira Migration Test Team](#)

5.8 [Worksheet 15: Jira Migration Configuration Testing](#)

5.8 [Worksheet 16: Jira Migration Test Cases](#)

6.0 After Migration

n/a

7.0 Q&A

n/a

Sources and Documentation

Use the following links for further information. Links are grouped by chapter and listed in order of appearance.

1.0 Introduction

- Accelerating our journey to the cloud, together – botron.io/atl-cloud-journey
- Atlassian Marketplace – botron.io/amp

2.0 Jira Deployment Differences

- Compare Atlassian cloud vs. server – botron.io/atl-cloud-vs-server
- Functional differences in Jira Cloud product – botron.io/atl-jira-cloud-vs-server
- User management differences in cloud and server – botron.io/atl-cloud-server-user-management
- Server vs. Data Center – what's right for you? – botron.io/atl-server-v-dc
- Jira Server and Data Center feature comparison – botron.io/atl-jira-server-dc-compare
- Jira Software Plans and pricing – botron.io/atl-jira-pricing
- Security at Atlassian – botron.io/atl-security
- Compliance at Atlassian – botron.io/atl-compliance
- Privacy at Atlassian – botron.io/atl-privacy
- Free trials
 - Download Jira Data Center – botron.io/atl-jira-dc
 - Cloud migration free trial – botron.io/atl-explore-cloud
- Worksheets for this book – botron.io/worksheets

3.0 Migration Options

- Jira Cloud focused
 - Atlassian cloud products – botron.io/atl-explore-cloud
 - Compare cloud migration methods – botron.io/atl-cloud-migrate-methods
 - Atlassian cloud migration support scope – botron.io/atl-cloud-migrate-methods
 - Shared migration roles and responsibilities – botron.io/atl-shared-roles
 - Use the Jira Cloud Migration Assistant to migrate from server to cloud – botron.io/atl-explore-cloud
 - Use Jira Site Import to migrate from server to cloud – botron.io/atl-jira-site-import

- What gets migrated with the Jira Cloud Migration Assistant – botron.io/atl-jira-cloud-migration-assistant-what
 - FAQs about migrating to Jira or Confluence Cloud – botron.io/atl-cloud-migrate-FAQ
- Jira Server and Data Center focused
 - Jira Data Center sample deployment and monitoring strategy – botron.io/atl-jira-dc-sample-deploy
 - Jira Data Center size profiles – botron.io/atl-jira-dc-size-profiles
 - Jira Data Center FAQs – botron.io/atl-jira-dc-faq
 - Moving to Jira Data Center – botron.io/atl-migrate-server-to-dc
 - Running Jira Data Center in a cluster – botron.io/atl-jira-dc-cluster
 - Set up a Jira Data Center cluster – botron.io/atl-jira-dc-cluster-setup
 - Upgrading Jira Data Center with zero downtime – botron.io/atl-jira-upgrade-dc
 - Running Jira Data Center on a single node – botron.io/atl-jira-dc-single
 - Evaluate apps for Data Center migration – botron.io/atl-dc-apps-eval
- Exporting issues – botron.io/atl-cloud-export-issues
- Export Content to Word, PDF, HTML and XML – botron.io/atl-export-content

4.0 Governance, Approval, and Change Management

- n/a

5.0 Migration Process

- Integrity Check for Jira – botron.io/amp-ic
- Configuration Manager for Jira – botron.io/amp-cmj
- Atlassian Data Center migration checklist – botron.io/atl-dc-migrate-checklist
- Backing up data – botron.io/atl-data-backup
- Restoring data – botron.io/atl-restore-data
- Restoring a project from backup – botron.io/atl-project-backup-restore
- Archiving a project – botron.io/atl-project-archive
- Premier Support Offering Details – botron.io/atl-premier-support
- Jira Performance Testing – Available Tools – botron.io/atl-jira-performance-test-tools
- Jira Cloud Migration Assistant – botron.io/amp-cloud-migration-asst

- What gets migrated with the Jira Cloud Migration Assistant – botron.io/atl-jira-cloud-migration-assistant-what
- Use the Jira Cloud Migration Assistant to migrate from server to cloud – botron.io/atl-jira-cloud-migration-assistant
- Use Jira Site Import to migrate from server to cloud – botron.io/atl-jira-site-import
- Data Center on Azure – botron.io/atl-dc-azure
- Deploying Data Center on AWS and Azure just got a lot easier – botron.io/atl-aws-azure
- Jira Data Center Migration App – botron.io/amp-dc-migration-app
- How to use the Data Center Migration app to migrate Jira to an AWS cluster – botron.io/atl-dc-migration-app-aws-cluster
- Getting started with Jira Data Center on AWS – botron.io/atl-jira-dc-aws-start
- Configuration Manager for Jira
 - Configuration Object Support Matrix – botron.io/CMJ-config-objects
 - List of Integrated Apps – botron.io/CMJ-apps-list
 - Data Loss Prevention – botron.io/CMJ-prevent-data-loss
 - Transactional Deployment & Roll back Strategy – botron.io/CMJ-rollback-strategy
 - Integrity Check – botron.io/CMJ-integrity-check
 - Audit Logging – botron.io/CMJ-audit-logging
 - Supported Versions – botron.io/amp-cmj-versions
 - Installation and Upgrade Guide – botron.io/CMJ-install-upgrade
 - Licensing – botron.io/CMJ-license
 - Choose a Snapshot – botron.io/CMJ-choose-snapshot
 - Apps – botron.io/PA-apps
 - Analyzing Changes – botron.io/CMJ-snapshot-change-analysis
 - Selective Merge – botron.io/CMJ-selective-merge
 - Move Attachments – botron.io/CMJ-move-attachments
 - Support – botron.io/support
 - App Integrations – botron.io/CMJ-apps-integration
 - Service Provider Interface – botron.io/CMJ-SPI
 - kreuzwerker Case Study – botron.io/kreuz-case-study

- Jira Command Line Interface (CLI) – botron.io/amp-cli
- Infrastructure recommendations for enterprise Jira instances on AWS – botron.io/atl-infrastructure-recs-aws
- Jira Sizing Guide – botron.io/atl-jira-size-guide
- Atlassian Partner Directory – botron.io/atl-partners
- Atlassian Migration Services Referral – botron.io/atl-dc-partners
- Configuration Manager for Jira Documentation – botron.io/CMJ-overview
- Power Admin: Documentation – botron.io/PA-overview
- Integrity Check for Jira: Documentation – botron.io/IC-overview
- Rachel Wright's Jira Server vs. Jira Cloud Interface Comparison – botron.io/rw-server-cloud-compare
- ScriptRunner for Jira app – botron.io/amp-scriptrunner

6.0 After Migration

- Configuration Manager for Jira
 - Use Cases – botron.io/CMJ-use-cases
 - Migrate Global App Data – botron.io/CMJ-migrate-global-app
 - Test – Staging – Production – botron.io/CMJ-test-stage-prod
 - Jira Clones – botron.io/CMJ-jira-clones
 - REST API: Configuration Manager – botron.io/CMJ-REST-API
 - Project Templates – botron.io/CMJ-project-template
 - Project Backup and CMDB – botron.io/CMJ-project-backup-CMDB
 - Project Archiving – botron.io/CMJ-project-archive
 - Incremental Archiving – botron.io/CMJ-incremental-archiving
 - Quick Fix – botron.io/CMJ-quick-fix
- Delegated Project Admin Pro for Jira – botron.io/amp-admin
- Delegated Project Creator for Jira – botron.io/amp-dpc

7.0 Q&A

- n/a

Index

A

Approval 8, 44, 48, 104, 172, 174

App(s) 10, 36, 78, 81, 82, 86, 87, 89, 92, 93, 101, 111, 157, 172, 175, 176

- Command Line Interface (CLI) 108, 137, 176
- Configuration Manager for Jira 11, 32, 33, 41, 81, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 110, 111, 113, 114, 129, 130, 131, 132, 154, 156, 157, 158, 159, 160, 161, 162, 163, 164, 166, 168, 169, 174, 175, 176
- Delegated Project Admin Pro for Jira 163, 176
- Delegated Project Creator for Jira 163
- Integrity Check for Jira 81, 82, 90, 113, 114, 169, 170, 174, 176
- Power Admin 77, 78, 90, 101, 113, 131, 168, 176
- ScriptRunner for Jira 41, 110, 176

Archive 37, 50, 78, 79, 164, 174, 176

Atlassian Solution Partner 87

Attachment(s) 43, 76, 108, 166

Audit log 50, 96, 105, 124

B

Backup 37, 38, 76, 112, 122, 164, 169, 174, 176

Buy-in 7

C

Change Management 8, 44, 172, 174

Classic Project 10, 15

Clean-up 11, 34, 42, 50, 77, 83, 155, 156, 161

Cloud 8, 10, 11, 12, 13, 15, 25, 32, 33, 37, 38, 39, 40, 41, 42, 78, 79, 84, 85, 86, 112, 126, 127, 131, 132, 156, 164, 173, 174, 175, 176

Cluster 10, 11, 13

Configuration Assessment 61, 127, 132, 137, 172

Configuration Manager for Jira 11, 32, 33, 41, 81, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 110, 111, 113, 114, 129, 130, 131, 132, 154, 156, 157, 158, 159, 160, 161, 162, 163, 164, 166, 168, 169, 174, 175, 176

- Deploy Configuration Snapshot 97, 98, 157
- Project Configuration Snapshot 93, 156, 161, 162
- Project with Issues Snapshot 93, 94
- Snapshot 11, 93, 94, 95, 96, 97, 98, 107, 108, 156, 157, 162, 175
- System Configuration Snapshot 93, 156, 157

Conflict(s) 101, 124, 129, 130, 131

Consolidation 39, 127

CSV Import 86

Customization(s) 25, 161

D

Database Administrator 13

Data Center 3, 7, 8, 10, 11, 12, 13, 15, 25, 34, 35, 36, 37, 39, 40, 47, 59, 78, 80, 84, 86, 87, 89, 90, 97, 112, 113, 114, 126, 127, 132, 156, 164, 173, 174, 175

Data Loss 88, 111

Disaster 10, 25, 112, 164

Disaster Recovery 10, 25, 112, 164

F

Freeze Configuration 122

G

Goal(s) 37, 129, 167

Governance 8, 44, 174

Group(s) 2, 50, 130, 168, 169, 179

I

Instance Health Check 80

Integrity Checker 80, 81, 91

J

Jira Administrator(s) 7, 11, 113, 166, 179

Jira Cloud. *See* Cloud

Jira Cloud Migration Assistant 32, 33, 84, 85, 173, 174, 175

Jira Core 10

Jira Data Center. *See* Data Center

Jira Data Center Migration App 86, 87, 175

Jira Server. *See* Server

Jira Service Management 10, 49, 85, 87, 101, 113, 166

Jira Site Import 32, 33, 85, 173, 175

Jira Software 10, 101, 173

L

Launch 124, 125

Lead 130

Load Balancer 34, 86, 97

M

Merge 9, 39, 88, 103, 105, 106, 112, 127, 131, 132, 175

Migration Checklist 51, 89, 111, 126, 172

Migration Process 2, 7, 8, 39, 51, 59, 76, 89, 105, 111, 123, 166, 168, 169

N

Next-gen 10, 11, 15, 85, 127

Next-gen project 127

Node 10, 11, 36, 97, 174

P

Performance 13, 80, 83, 91, 174

Plan(s) 30, 36, 39, 41, 43, 44, 47, 49, 59, 76, 84, 112, 122, 124, 164, 166, 169, 171, 172

Production Application 11, 168

Project Manager(s) 8

Project Templates 161, 176

R

Refresh Environment 122

Risk(s) 88, 168

S

Server 2, 3, 7, 8, 10, 11, 12, 13, 14, 15, 25, 32, 33, 34, 35, 37, 38, 39, 40, 41, 42, 47, 79, 80, 84, 86, 89, 90, 112, 113, 114, 126, 127, 128, 131, 132, 156, 164, 173, 174, 176

Source Application 11, 26, 37, 42, 60, 61, 75, 77, 83, 86, 88, 101, 103, 105, 107, 109, 112, 132, 155, 156, 160, 167, 168

Staging Application 11, 89, 93, 96, 97, 124

Stakeholders 2, 44, 46, 51, 60, 126, 166, 172

T

Target Application 11, 36, 39, 60, 75, 84, 85, 86, 87, 88, 93, 96, 97, 100, 101, 102, 103, 104, 105, 107, 108, 110, 126, 128, 129, 132, 155, 156, 158, 159, 160, 166, 167, 168

Tester(s) 135

Testing 8, 107, 111, 122, 135, 137, 172, 174

Test Cases 8, 48, 61, 107, 122, 132, 135, 150

Test Environment 156

Test Team 135, 172

Training 2, 34, 43, 106

Post-Migration Training 106

Pre-Migration Training 126

U

User(s) 2, 9, 12, 13, 14, 15, 33, 34, 37, 38, 41, 43, 50, 79, 81, 87, 105, 107, 108, 112, 113, 123, 126, 127, 129, 130, 131, 132, 161, 173, 179

W

Worksheet 11, 46, 47, 51, 75, 89, 111, 116, 126, 127, 132, 133, 135, 137, 151, 172

About the Author

Rachel Wright

Rachel has a unique set of skills and experience you won't find elsewhere. As the founder of a Web Development company for 13 years, she understands business and software development tools, and has lived the software development life cycle. As a prioritization specialist for five years, she helped business and IT teams analyze, prioritize, schedule, and track strategic company projects. And as a certified Jira Administrator, Rachel thoroughly understands Jira configuration best practices and how decisions today affect the application in the future.

She literally wrote the book on Jira strategy and effective administration. Her consulting experience is diverse — helping development, support, and business teams in all different industries and environments.

Finally, Rachel is well known in the Atlassian community as a frequent speaker, trainer, user group leader, and community champion. This uncommon combination of skills and experience creates a deep understanding and holistic, unbiased, and informed approach for setting up, cleaning up, migrating, and maintaining Atlassian applications.



Connect with Rachel at:

rwright@jirastrategy.com

[linkedin.com/in/rlwww](https://www.linkedin.com/in/rlwww)

[@rlw_www](https://twitter.com/rlw_www)

Special Thanks

Thank you to the wonderful reviewers, editors, and designers who vetted my recommendations, clarified my thoughts, and formatted this book to make it a useful resource. They are:

Abby Nitta	Maria Timova
Betsy Walker	Piers Morgan
Boris Georgiev	Rossen Marinkov
Edward Grosse	Saran Taharnkleaw
George Dinkov	Shankar Asam
Holly Wright	Teodora Vasileva
Kelsey LaMastres	ThaoLinh Judin
Kiran Panduga	Vladimir Yakimov



Configuration Manager for Jira

More than 4,000 companies worldwide and over half of the Fortune 100 have trusted Configuration Manager for Jira to manage their Jira configurations at scale and solve their most complex migration challenges.

Try it free today!



"A must-have for any enterprise maintaining the Jira instance health, applying incremental configuration changes to projects or system, great for project templates and moving filters, agile boards, and even dashboards."

- Charles C.



"Configuration Manager has been a vital Jira App assisting with both the daily management of our Jira Data Center Instance as well as the migration and consolidation of multiple Instances."

- Alex J.