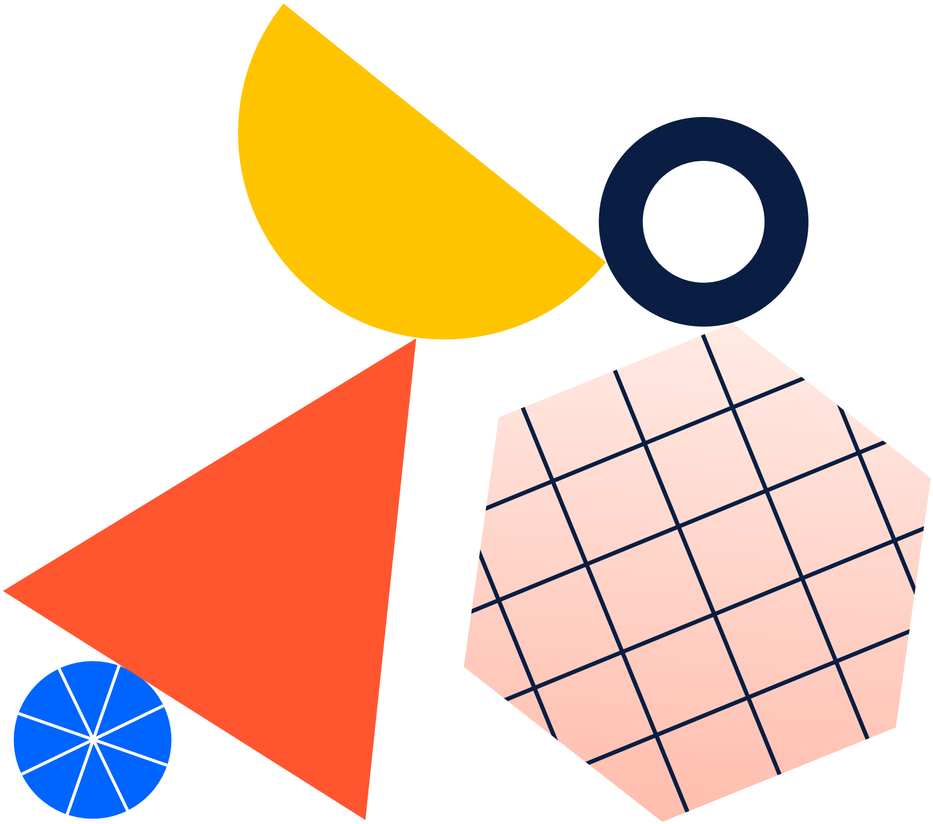
ATLASSIAN + airbnb



5 core responsibilities every admin should master

A guide to scale your Atlassian products with finesse

Alex Gallien & Isha Metha

Contents

Introduction

Let's face it, being an admin isn't a low-pressure job.

Most of you are responsible for managing hundreds or even thousands of workflows, knowledge articles, and data repositories that allow your organization to get work done, and it can often feel like you're putting out fires just to keep work running smoothly.

As your organization continues to grow your role becomes increasingly crucial, and that feeling can quickly multiply.

We call admins experiencing organizational growth, "admins at scale." Let's take a look at some differences between an admin and an admin at scale.

Admin	Admin at Scale
Manages 1 instance	Manages multiple instances
Serves one team using Atlassian products	Serves multiple teams using Atlassian products
Co-located with their team in one location	Part of a multi-location organization
Atlassian products are a “nice to have” for specific teams	Atlasian products are mission critical for most, if not all teams

With an increase in users, teams, instances, or complexity, it’s common for admins at scale to experience operational disconnect across their organization, not to mention significant reporting challenges and a lack of proper controls. Decentralization will also result in multiple backup solutions (or even worse - none), ongoing infrastructure provisions, unnecessary overhead, and a plethora of versions with no standard methodologies across teams.

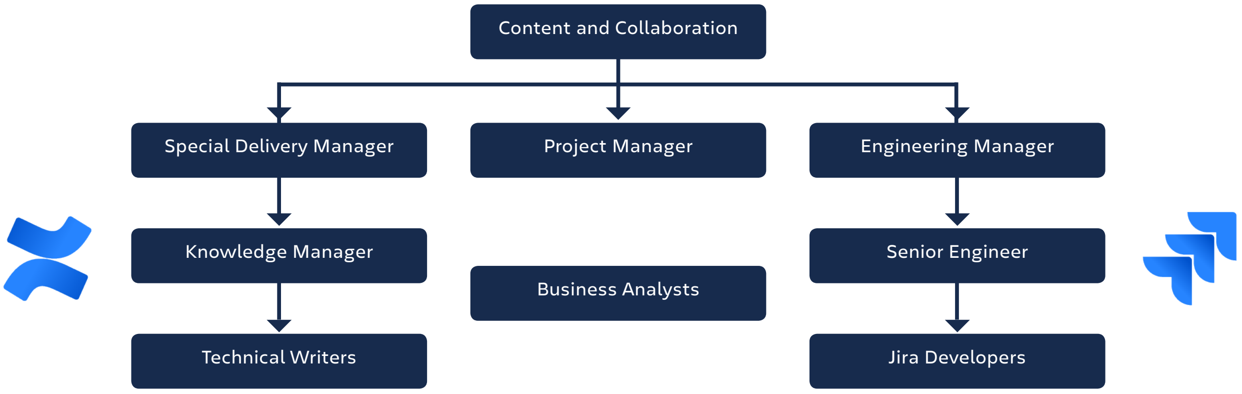
Today, let’s dive into five core topics that will help you avoid these challenges and master administration at scale, beginning with governance.

01. Establish Governance

Centralization is an excellent place to start when establishing governance because it often leads to a simplified workflow for you, as an admin, and a more reliable experience for your users. And who doesn’t want that?

3 Ways to Centralize

1. **Centralize your setup:** Centralizing your instances allows you to put the necessary controls in place and enforce standardization for reporting and management purposes, while still allowing teams to work in their preferred ways. You can achieve this by either scaling a Server instance or adopting Data Center. The main distinction between these two is that Server runs on a single node, while Data Center allows you to run on multiple nodes.
2. **Centralize your team:** Establish one team that is responsible for the administration of Atlassian products across the entire organization.
3. **Centralize your processes:** Set up processes across the full Atlassian suite for all users and make them simple and consistent. Look for common recurring themes across the organization to ensure you don’t over-optimize for every specific use case.



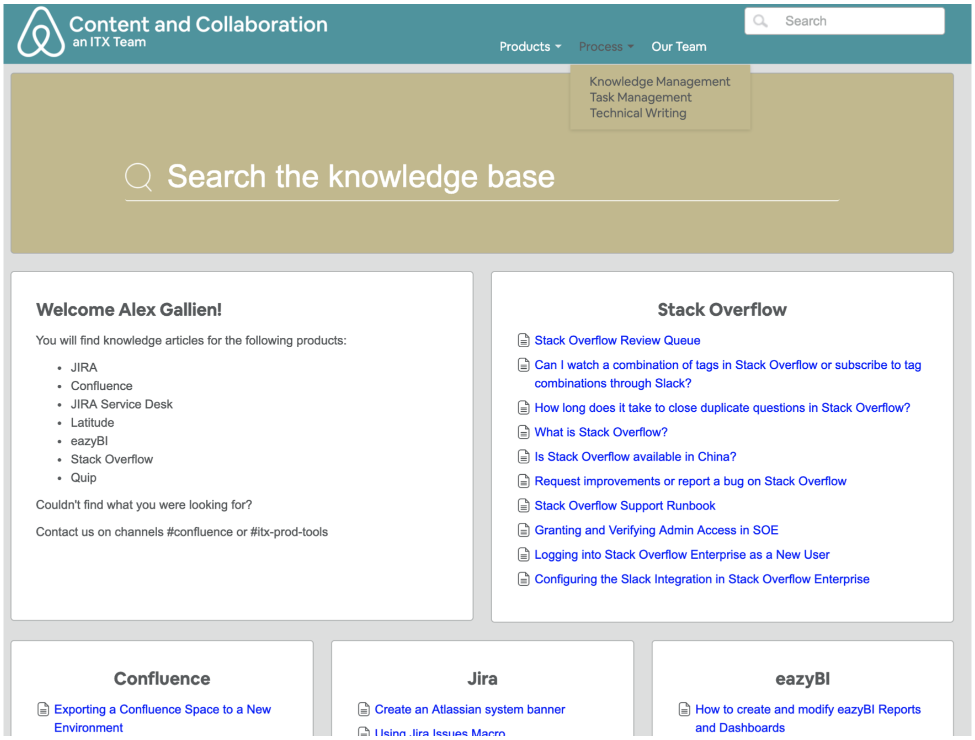
Airbnb’s centralized team structure, with content and collaboration at their core

Intentional Access

Granting users the appropriate level of access can feel more like an art than a science, but understanding the types of roles will help this. For example, “System Administrators” can perform all Jira administration functions. However, “Jira Administrators” can only conduct administrator duties such as creating new users or projects, without the ability to make changes that could affect the application environment or network. Knowing these differences will prevent you from granting unnecessary permissions up front, but it’s important to audit access levels on a monthly or quarterly basis as circumstances change. If a user doesn’t actively depend on access to do the work they’re doing, then remove it.

Empowering Users

Once your users have the right permissions to get their job done, empower them by offering support and guidance at every turn. Providing a single source of truth for user questions in products like Confluence not only removes ambiguity, but provides a reliable way to get answers quickly. Atlassian products are built to be flexible and they allow users to solve many problems in many different ways. To maintain consistent workflows in your organization, clearly outline the intended use for each product to avoid any confusion.



Welcome Alex Gallien!

You will find knowledge articles for the following products:

- JIRA
- Confluence
- JIRA Service Desk
- Latitude
- eazyBI
- Stack Overflow
- Quip

Couldn't find what you were looking for?

Contact us on channels #confluence or #itx-prod-tools

Stack Overflow

- Stack Overflow Review Queue
- Can I watch a combination of tags in Stack Overflow or subscribe to tag combinations through Slack?
- How long does it take to close duplicate questions in Stack Overflow?
- What is Stack Overflow?
- Is Stack Overflow available in China?
- Request improvements or report a bug on Stack Overflow
- Stack Overflow Support Runbook
- Granting and Verifying Admin Access in SOE
- Logging into Stack Overflow Enterprise as a New User
- Configuring the Slack Integration in Stack Overflow Enterprise

Confluence

- Exporting a Confluence Space to a New Environment

Jira

- Create an Atlassian system banner
- Using Jira Issues Macro

eazyBI

- How to create and modify eazyBI Reports and Dashboards

airbnb | Airbnb set up their Confluence instance as a knowledge base to empower users to self-serve.

Balancing Flexibility and Consistency

While maintaining a completely centralized and consistent governance system is appealing, the reality remains that customization is still needed. Different departments often need their own unique workflows to meet their specific needs. Providing precise documentation does help, but some questions simply can't be anticipated, or are specific enough to require human intervention. Providing a channel of communication ensures that your users aren't blocked from doing their jobs unnecessarily, and allows your organization to quickly pivot as new user needs arise.

airbnb | Airbnb set up a dedicated Slack channel for users to get on the fly help for potentially blocking issues.

02. Improve Performance

What Impacts Performance?

There are many factors that can affect performance, which makes performance issues so difficult to solve. For Jira, it’s common to measure performance by how many issues you have.

> Don’t default to only using “number of issues” as an indicator of performance, but also consider other factors that have an impact, including the usage patterns of your instance, the way it’s configured, and the hardware it’s deployed on.

While issues are a contributing factor, you can’t have a full understanding of how to improve your performance without first understanding your environment. Because everyone’s environment is different, there is no ‘one size fits all’ solution, which makes learning your unique set-up even more crucial.

Know Your Limits

Once you’re aware of all the factors that can impact your performance, learn your limits. There are three limits that you need to be familiar with:

Hardware limits

Our sizing guide provides guidance on how to approach sizing your application node(s) based on usage level factors like instance size and complexity. Additionally, Atlassian provides hardware recommendations for organizations of all sizes.

Usage limits

Understand your usage patterns, especially when it comes to integrations. In some cases, a single bloated metric could be defining your overall size profile; once identified, you can easily find ways to manage it.

Configuration complexity

Atlassian products are flexible and can be customized based on needs, but products do not scale infinitely. Be sure that your environment can handle your level of customization, and be prepared to adjust and enforce that level of customization company-wide.

Seek First to Understand

Interpreting what a user means when they say “Jira is slow” is a critical component of an admin’s job. Many times the problem isn’t with the product itself, but how it is being used.

airbnb | An example of effective administration in a real-life scenario that occurred at Airbnb

User reported that “Jira is unusably slow.”

Alex dug into the issue, and found that the user’s Agile board had a complex filter pulling over 4,000 issues, five different card color rules, and swimlanes based on JQL filters.

Alex was able to greatly simplify the board, giving their team a much better experience.

Help determine what your users’ goals are and what they need to be successful. Teams use Atlassian products for a variety of use cases, so translating user needs and setting expectations is key. Oftentimes as an admin, you’re forced to make changes that users disagree with, such as locking down administrator permissions. Preparing a good explanation of “why” will make these conversations much easier.

2 Must-dos for Quick Performance Wins

1. **Set custom field context:** Get in the habit of setting custom field context for all new custom fields! Period. This can greatly reduce the size of your index and consequently, the speed of anything in Jira involving searches. If you already have a ton of custom fields without context, there are plugins such as “Optimizer for Jira” in the Atlassian Marketplace that can help clean up your instances as you establish better habits.

   → Use Jira Software Data Center’s custom fields optimizer to speed up your Jira instance. The custom field optimizer allows you to easily identify and remove unused custom fields in just one click. Upgrade to Jira Software Data Center 7.13 or later to get access to this feature.

airbnb

Using custom fields optimizer, Airbnb set custom field context and subsequently ran a reindex, taking their search time from 14 seconds down to one. At the time, they had about 800,000 issues and 900 custom fields.

2. **Turn off empty JQL Search:** Disabling empty Jira Query Language (JQL) search is an easy fix that significantly improves performance by relieving pressure on the index. Disabling empty JQL search was introduced in Jira 7.9, so if you’ve upgraded from an earlier version you just have to switch it on! It has been backported to almost all branches of Jira at this point.

   → You can turn off empty JQL search in your “Global Configuration” settings. This changes the behavior when a user selects “Issues” --> “Search for Issues” to return an empty search instead of searching the entire index, and provides users a much better experience.

03. Develop a Monitoring Strategy

As a vital part of ensuring the integrity and continued optimization of your application, regularly monitoring performance helps you prepare for usage growth or re-configuration.

Know What to Monitor

First, you need to identify key performance indicators (KPIs) that will tell you if your instance is healthy and performing to your desired expectations. The two types of metrics you should be measuring are usage metrics and intrastructure metrics.

Here are examples of common KPIs to monitor for each one:

Usage	Infrastructure
• Number of issues • Number of custom fields • Number of concurrent users • Activity of major integrations • Available license seats	• Hosting options • CPU/Disk Speed • Avilable memory • JVM/JMX statistics • Database performance • Network speed

Establish Alert Levels & Actions

Not all metrics are of the same importance, so consider the appropriate alert level for each metric. And what will you do once an alert is triggered? Whoever receives an alert should know exactly what the intended course of action is. Define Service Level Agreements (SLAs) to help you react in a timely fashion and ensure you're handling these requests in the most appropriate way.

Measure Your Metrics

It’s unrealistic to assume your performance metrics will always be perfect. If you don’t already have baselines in place, know that it takes time to understand your instance before you can set the right targets for your organization. When deciding how to best optimize your application over time, focus on building enough data to identify trends rather than studying performance snapshots.

Not all metrics are required to have alerts. While it’s beneficial to keep an eye on active HTTP threads, the number of users and fields growing in Jira, and the amount of email notifications, these are examples of metrics that may not require active alerts. The metrics that do need more attention include: the health of your Java Virtual Machine (JVM), application response times, thread profiling, and end-to-end tracing. Consider adding tools to help watch these more closely.

airbnb | Airbnb uses Sumologic, Logic Monitor, and Cloud Watch to make up their monitoring suite.

Other monitoring tools we see customers use include Opsgenie, Java Melody, New Relic, Dynatrace, and more. Ultimately, to choose the right monitoring tools you need to consider your organization’s instance and needs, available technical expertise or resources, budget, and time. And if you’re curious, you can see how Atlassian monitors their enterprise deployments here.

04. Streamline Your Upgrades

Upgrading to modern versions has a ton of advantages like access to more features, bug fixes, and improvements. But upgrading isn't the easiest task, especially at scale, which makes having an upgrade strategy essential for admins.

Types of Releases

When considering your upgrade strategy, there are four types of releases to be aware of:

Platform releases contain significant or breaking changes. This might include changes to (or the removal of) existing APIs, significant changes to the user experience, or a new major feature.

Bug fix releases contain bug fixes, stability, and performance improvements. They may introduce minor changes to existing features, but do not include new features or high-risk changes. We recommend regularly upgrading to the latest bugfix release for your current version.

Feature releases contain new features, changes to existing features, changes to supported platforms or removal of features.

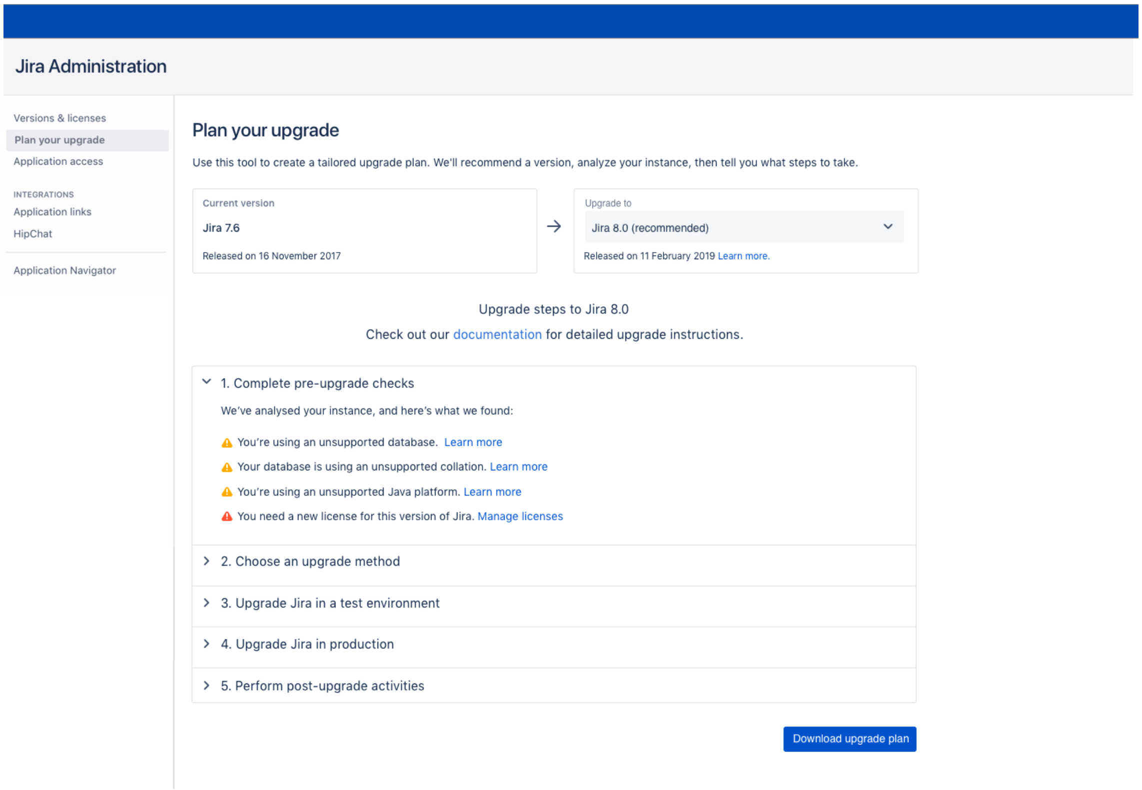
Enterprise releases (ER) are designated feature releases that receive bug fixes for a longer period of time than a standard feature release. If you're only upgrading once or twice a year, you should aim to upgrade to an enterprise release.

Consider your organization's needs present and future, as well as how frequently you plan to upgrade, to determine which release is right for you.

Plan your Upgrade

To effectively plan your upgrade, you'll first need to identify any changes in supported platforms. For each Atlassian product, there's documentation for each release to view changes in supported platforms from version to version.

After identifying any changes, run instance health checks to make sure your instance is healthy before an upgrade. There's nothing worse than running an upgrade and finding out that something was wrong with your instance before the upgrade. Atlassian provides resources such as the instance health page and pre-upgrade planning page to help you create a tailored upgrade plan for your specific instance.



Plan your upgrade

Use this tool to create a tailored upgrade plan. We'll recommend a version, analyze your instance, then tell you what steps to take.

Current version

Jira 7.6

Released on 16 November 2017

Upgrade to

Jira 8.0 (recommended)

Released on 11 February 2019 Learn more.

Upgrade steps to Jira 8.0

Check out our documentation for detailed upgrade instructions.

1. Complete pre-upgrade checks

We've analysed your instance, and here's what we found:

- ⚠ You're using an unsupported database. Learn more
- ⚠ Your database is using an unsupported collation. Learn more
- ⚠ You're using an unsupported Java platform. Learn more
- ⚠ You need a new license for this version of Jira. Manage licenses

2. Choose an upgrade method

3. Upgrade Jira in a test environment

4. Upgrade Jira in production

5. Perform post-upgrade activities

An in-product view of the the Pre-upgrade Planning page in Jira.

Upgrading your Instance

Before you upgrade, it's essential to always test your upgrade beforehand. This will help iron out any issues in advance. Be sure to document all your learnings in the test environment so that you and your team can use them during the actual upgrade.

If you're on a Data Center instance, there are specific features to help with your upgrade process and give users access to the system during upgrades. For Jira Data Center, there's zero downtime upgrades, which allows you to upgrade node by node to avoid any downtime - and hopefully it saves you from performing an upgrade over a weekend! For Confluence Data Center, there's read-only mode, allowing you to set your instance to read-only mode and back up all the data required before you upgrade, resulting in much less downtime.

airbnb | Airbnb uses tools like Chef and Terraform to make upgrades as painless as possible by automating parts of the upgrade process.

Lean on Experts

If possible, it's always a good practice to lean on experts while planning or performing upgrades.

Technical Account Managers (TAMs) can help you develop the right strategy and practices to manage and improve the usage of your Atlassian products.

Priority and Premier support provide weekend support coverage and faster response times, which is extremely handy when performing upgrades!

Solution Partners can also help with solutions, training, and strategic services.

05. Master the Human Side of Scaling

We find that the human element of scaling is perhaps the most important part, but it's often the easiest to neglect. Understanding how your teams work together and developing trust before mapping process to your tools will better set you up for success.

Help Users Help Themselves

- **Make it easy to get help:** Establish a standard process for users to get answers.

> airbnb
>
> Airbnb uses a central Jira project that all requests come through and leverages canned responses as templates for common requests, such as project creation requests.

- **Be available:** Making yourself available is crucial, especially when you've locked down administrator permissions like you should. Small requests, such as modifications to a workflow, don't take much time. Providing a quick turnaround will raise satisfaction and help you gain your team's trust.
- **Learn to say no:** Atlassian products are heavily customizable tools that can do just about anything you want them too, but just because you can doesn't mean you should. You have to be able to say no. Understanding how customizations will affect performance, and providing supporting data will help to make this conversation go as smoothly as possible. Remember, you're the ultimate gatekeeper of your Atlassian products.

Share the Knowledge

Ensure the knowledge of Atlassian products is not just within the admin team, but shared throughout the organization. It should be easy for anyone to find.

airbnb

> Airbnb established a “power users group” meetup for all Jira enthusiasts in the company to share tips and best practices, which then get disseminated to the rest of the organization.

Having trusted subject matter experts (SMEs) champion your products internally greatly helps with company-wide adoption.

Training & Certification

Lastly, Atlassian University provides training and certification programs to help you and your teams gain expertise and get the most out of your products. Courses are divided into learning paths for Jira admins, Jira team leads or Jira Service Desk admins, and more, to make sure you are reading up on the most revelant content based for your role. In addition to online courses and official Atlassian certification exams, you’ll find a ton of free learning material on Atlassian University to share with your team as well.

Conclusion

Although the responsibilities of admin can be overwhelming at times, with these 5 skills you'll be in a much better place to master administration at scale.

1. **Governance:** Establish governance early on to build a unified, well-running, organization-wide system that's consistent.
2. **Performance:** Learn how your instance performs and set guidelines so you can plan for growth while ensuring performance is maintained.
3. **Monitoring:** Monitor changes in your instance and set alerts for major issues as soon as they arise.
4. **Upgrades:** Develop an upgrade strategy and update regularly to take advantage of the newest features and fixes.
5. **Human side of scaling:** Recognize that users are only human, and being an admin means you sometimes have to say "no" for the greater good of the instance in order to scale in the future.

At the end of the day, successful administration comes down to your people, tools, and practices. Keep all of this in mind as you continue to grow your Atlassian products with finesse.

See our infographic for more tips to improve the performance of your Server products.

Learn more

Planning an upgrade to Data Center? Download this guide to prepare.

Download

About the authors

Alex Gallien

Senior Atlassian Engineer, Airbnb

Alex Gallien is the Senior Engineer for the Productivity Tools team at Airbnb, responsible for the Atlassian Data Center stack. When not tending to his Atlassian products, he enjoys cooking, teaching meditation, playing jazz bass, and spinning fire.

Isha Mehta

Product Manager, Atlassian

Isha is an Enterprise Product Manager at Atlassian currently based in Sydney. She works with cross-functional teams to deliver solutions that solve problems for customers at scale. She is an ardent advocate for the customer and is passionate about building productivity through better practices.