



Server to Data Center: The tipping point



Atlassian unleashes the potential in every team.

Our products help teams collaborate, build software and serve their customers better. Teams at more than 54,000 large and small organizations – including BMW, Citigroup, eBay, Coca-Cola and NASA – use Atlassian’s tracking, collaboration, communication, service management and development products to work smarter and deliver quality results on time.

Our product portfolio includes JIRA Software, JIRA Core, JIRA Service Desk, Confluence, HipChat, Bitbucket and Bamboo.



Server to Data Center: The tipping point

1. Scaling Atlassian applications	4
2. The tipping point	10
3. What is Data Center?	15
4. Prepare, plan, execute	24
5. Refine	33
6. Hear it from our customers	37
7. What's new in Data Center	42



1. Scaling Atlassian applications

At some point in your Atlassian journey you'll likely find yourself needing to scale your Atlassian applications to a wider group in your organization. This section provides some high-level guidelines for determining growth and creating internal governance before taking the final step in scaling Atlassian: providing true high availability and building a more robust infrastructure to support uptime and performance needs. Keep in mind that there is no 'one-size-fits-all' approach to scaling Atlassian applications. Each organization, and team, has different processes, workflows and needs.

Determining scale

When considering scaling a single instance of an application like JIRA Software, many customers first think about the number of issues their instance can support, ignoring some of the other factors that can contribute to poor performance or increased administration. The chart to the right is a list of factors that we recommend considering when scaling a single instance of JIRA Software. Keep in mind that one characteristic may not grow at the same rate as others. For example, just because the number of issues are increasing doesn't mean that projects will follow too. More information on scaling JIRA Software and other Atlassian applications can be found in our Enterprise documentation.

Governance

With widespread adoption of highly customizable applications, like JIRA Software, administrators are often left figuring out how to manage the application. One of the most effective ways to scale products is to create and apply rules within the application to limit the number of modifications that users can make.



SCALING ATlassian APPLICATIONS

We've built some sizing guidelines for how we recommend customers think about their JIRA Software instance based on the number of licensed users. Keep in mind that the recommendations for your company depend on the specific growth type, intensity and use cases.

Sizing legend	Small-scale	Mid-scale	Large-scale	Enterprise-scale
Application usage				
Users	100	500	2000	10000
Active (concurrent) users	25	200	600	2000
Issues	15000	60000	200000	1000000
Issues/month	200	1000	4000	20000
Custom fields	50	150	300	600
Permission schemes	3	15	25	100
Projects	20	80	200	300
Parent issue types	10	20	50	160
Resolutions	10	20	30	40
Priorities	10	15	25	40
Workflows	5	20	35	100
System level	Small-scale	Medium-scale	Large-scale	Enterprise-scale

More JIRA Software sizing guidelines for things like system requirements can be found in our documentation.



Federating

If you're running more than one server to support an Atlassian application, let's say Confluence, then you're likely running Confluence in a federated environment. Larger customers choose to federate their Atlassian environment for a variety of reasons but most commonly due to growth. Some examples include:

- **Bottom-up growth:** Due to Atlassian's low price point and practical value, our products often start in a single team and then spread throughout an organization with new teams spinning up their own server instances.
- **Mergers & acquisitions:** Through acquiring or merging with other organizations, a company can find itself managing several application servers.
- **Autonomous IT organizations:** Different departments within an organization may run their own IT organizations. This leads to parallel systems with possible integration at a later stage once cross-divisional processes and collaboration are encouraged.
- **Intentional federation set-up:** There are several reasons why starting out with multiple instances is the right approach for your organization. This could include standing up servers for standalone temporary teams or projects, projects requiring specific levels of permissions or security, or even location-based projects.

If you're interested in learning more about setting up or maintaining a federated environment for JIRA Software, check out our guide [here](#).



Warm failover

A warm failover is an active-passive setup in a single server environment. This means that in the event that your main server fails, users can be directed to a standby server with a complete backup of the main server. This configuration is a great way to provide an additional level of assurance, but you're still left with a single point of failure since the backup server is only available if another server fails. So while there are measures that you can take to keep your teams online in a single server setup, you're not technically providing true high availability.

Unlike an active-passive configuration, an active-active setup gives you true high availability (HA) by allowing you to configure your setup so that you can direct traffic to specific nodes based on activity, or add more active nodes to offset demand during peak times. We'll touch more on how to provide HA in Section 4: What is Data Center.



High availability — a strategy to provide a specific level of availability, access to the application, and an acceptable response time. Automated correction and failover (within the same location) are usually part of high availability planning.



Disaster Recovery

A disaster recovery (DR) strategy is a crucial aspect of any enterprise grade deployment. In the event of a complete system-wide outage, a geographically separate system will resume application functionality and minimize disruption to your team's productivity.

Setting up a DR strategy for a single server allows you to have copies of the database and file system so that your standby site has the same data as your primary site. However, unless you're using Atlassian's Data Center offering, you will not be able to copy indexes of your database from your primary servers to your backup servers. This has an impact on how quickly your team can get back up and running as the application will need to rebuild indexes upon startup and for a large installation this could take hours.

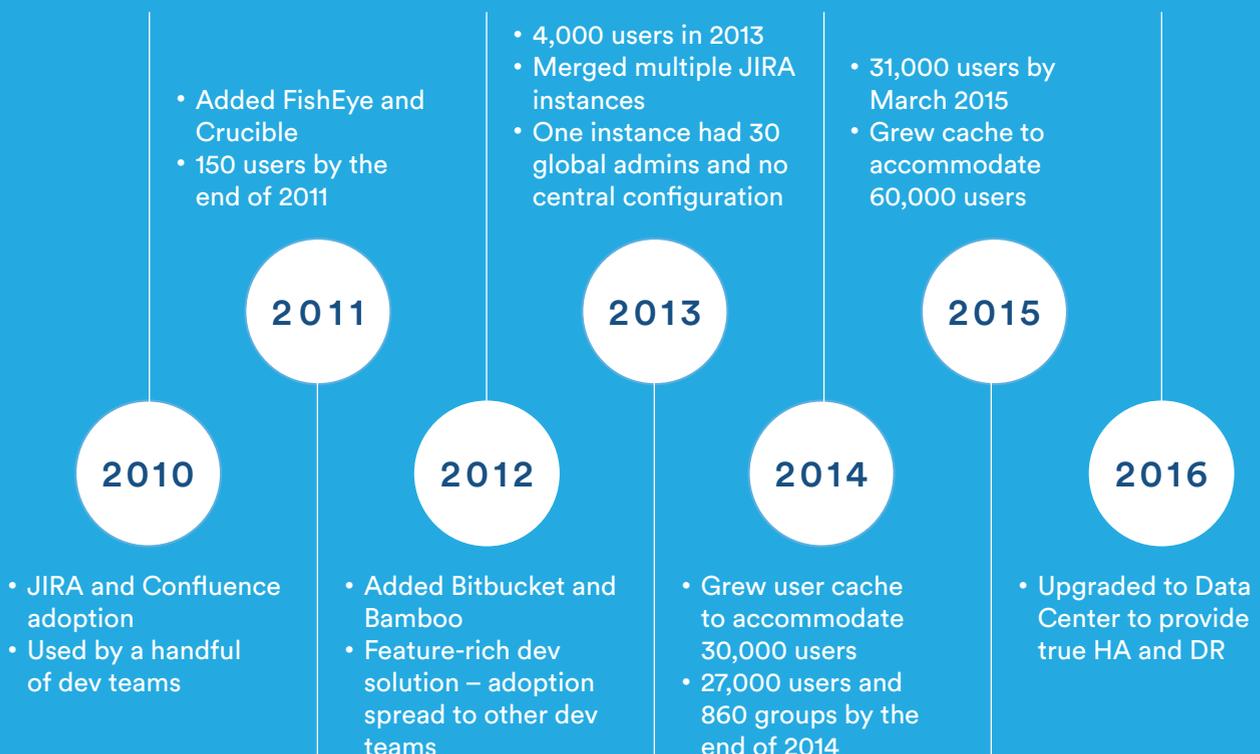
While the only Atlassian supported disaster recovery solution is available with Data Center, there are ways for you to set up an unsupported DR plan on your own. We provide some documentation on how to set that up for JIRA Software here: [Alternative Disaster Recovery Guide for JIRA - Atlassian Documentation](#).

Customer story: from 0 - 60,000 users in 5 years

The Department of Defense grows with Atlassian applications

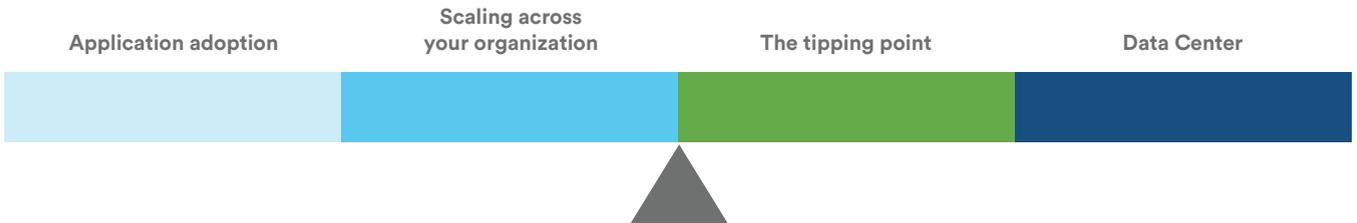
We often hear about the Atlassian journey as a gradual adoption over time: a developer starts up an instance of one of our products and starts using it with his or her team. Other teams find out about the new tool, try it out and a “grassroots” adoption is born. For the Department of Defense (DoD), this journey was no different.

What started as a couple of teams using JIRA and Confluence to document and plan their development work, very quickly grew into 60,000 people using nearly the entire Atlassian stack to manage their software development process. Despite consolidating instances and increasing user cache, DoD administrators needed a more effective way to manage applications at scale. They recently made the decision to upgrade to Data Center for JIRA Software, JIRA Service Desk, Bitbucket, and Confluence so they can ensure they’re providing the highest quality of service and true high availability.





2: The tipping point



For many of our customers there comes a time in their Atlassian journey when they need more than what a single server or federated environment can provide in terms of availability and performance. As applications grow across an organization they become mission-critical to every team's success. We call this the "tipping point" for moving to an active-active clustered environment that provides high availability and supports performance at scale.

Here are some criteria for customers considering the move to Data Center. Note: you may not meet all of these criteria today, but if you have growth plans in your future, think about preparing now.

Users

Consider how many users you have accessing your Atlassian applications each day. Are you at or approaching 500? We've found the tipping point for JIRA Software, Confluence, and Bitbucket customers who need more stability tends to be between the 500-1000 user mark. In fact, roughly 45% of Data Center customers upgrade to this offering at the 500 or 1,000 user tier. When it comes to JIRA Service Desk, 50% of Data Center customers make the move at the 50 agent tier.



As development teams grow, their repos grow alongside them. For distributed teams, this can mean slower clone times between the main instance and remote team. To reduce this pain, Bitbucket Data Center allows for Smart Mirroring which makes read-only copies of repos available on a nearby mirror in a remote location. Mirrors can cut clone and fetch times from hours to minutes, letting users get what they need faster.

Performance

For customers on large instances, performance degradation usually happens under high load or at peak times. Meaning, as more and more users access the application at the same time, response times increase, users get frustrated, and system administrators look for solutions to minimize pain (for users, and themselves). Many global companies experience this when multiple geographic locations come online at the same time. At Atlassian, we experienced this first-hand when our Sydney teams started their day – we had hundreds of concurrent users logging on to a system that already had hundreds online. This usually caused our San Francisco and Austin offices, in addition to our Sydney office, to struggle with slow page load times or brief periods offline.

In addition to concurrent usage, other running jobs, like API calls and queries, can impact performance issues. Adding these on top of your users' traffic only exasperates the problem.

Data Center provides the ability to use a load balancer to direct certain types of traffic to certain nodes in your cluster. This allows you to compartmentalize resources to ensure all of your requests maintain the best performance possible. For example, you could direct all of your API traffic to a specific node (or number of nodes). This way, your normal user traffic is never slowed down by ongoing API jobs.



Downtime

There are typically two primary causes of downtime: application and server-side issues. When it comes to the application side, issues are often a result of JVM errors, the most common of which is the heap being overloaded. That is, the memory dedicated on the server for running the application gets too full and causes the application to fail. Another common application side issue is the database's connection being overloaded with requests and causing the application to fail. Server-side issues can range from planned maintenance to unplanned upgrades/installations to resources like CPU, RAM, or storage on the server being overwhelmed and causing an outage.

Whatever the source of the outage, the result is lost productivity from hundreds or thousands of employees being unable to work. Those costs can quickly add up. How many people in your organization depend on JIRA Software, Bitbucket, Confluence, or JIRA Service Desk to get their jobs done? What does an hour of downtime potentially equate to in lost opportunity cost per hour?

Data Center significantly reduces this risk. If one server in your cluster goes down, the others take on the load. Instead of productivity grinding to a halt until the server gets back up and running, traffic is redirected to an active server and business continues as usual. More on this can be found in Section 4: What is Data Center.

“In 2014, we had 55 outages and in 2015, we reduced that number to 7 by applying scaling lessons. Now with JIRA Software Data Center, we haven't experienced a single unplanned outage in 2016, all while our usage has continue to grow.”

- Mike Damman, Knowledge Architect, Cerner

How much does it cost for your system to go down?

{ Let's assume that you have 500 engineers working in Bitbucket at one time and the system goes down for an hour – no more commits or pull requests, all work and productivity has come to a halt.

Assuming the average salary for a software engineer in the United States, you're looking at a cost of nearly \$22,000 for salaries alone. Not to mention, additional soft costs like productivity and the overall impact to business.



Administration

Providing access to mission-critical applications is no easy task. We know you take this seriously and we want to help you maximize your efforts. Sure, using a single server or federated environment might work for you today, but think about the added complexity when the single server becomes overloaded or the federated servers don't work together the way you're hoping they might. With Data Center, we're aiming to make your job more efficient and stress-free by giving you all the tools you need to maintain the performance and up-time you're expecting while managing growth.



3. What is Data Center?

Data Center is a self-hosted / on premise deployment option designed to help Atlassian customers effectively scale their mission-critical Atlassian applications. Through active-active clustering, teams can work around-the-clock with uninterrupted access to the tools they need to get their jobs done.

Some of the key benefits of Data Center are:

- **High availability:** Active clustering ensures users have uninterrupted access to critical applications.
- **Performance at scale:** Each node added to your Data Center cluster increases concurrent user capacity and improves response time as user activity grows.
- **Instant scalability:** Add new nodes to your Data Center cluster without downtime or additional licensing fees. Indexes and plugins are automatically synced.
- **Disaster recovery:** Deploy an offsite Disaster Recovery system for business continuity even in the event of a complete system outage. Shared application indexes get you back up and running quickly.

Data Center is available for the following Atlassian products:

JIRA Software

Get uninterrupted access to JIRA Software and maintain performance at scale.

Confluence

Collaborate around the clock with Confluence Data Center.

JIRA Service Desk

Serve your enterprise at scale – no request goes unresolved with JIRA Service Desk Data Center's high availability and disaster recovery.

Bitbucket

Bitbucket Data Center provides source code collaboration for professional teams of any size, across any distance.



High availability

Data Center helps you achieve high availability through active clustering and automatic hot failover within your data center. Cluster multiple active servers to ensure uninterrupted access to critical Atlassian applications in the event of a hardware failure. If a node fails, the load balancer will automatically redirect users to another active node in the pool, or cluster. Most users won't notice any downtime since they're automatically directed from the failed node to an active node. Use industry standard load-balancing, database clustering and shared files systems to minimize single points of failure.

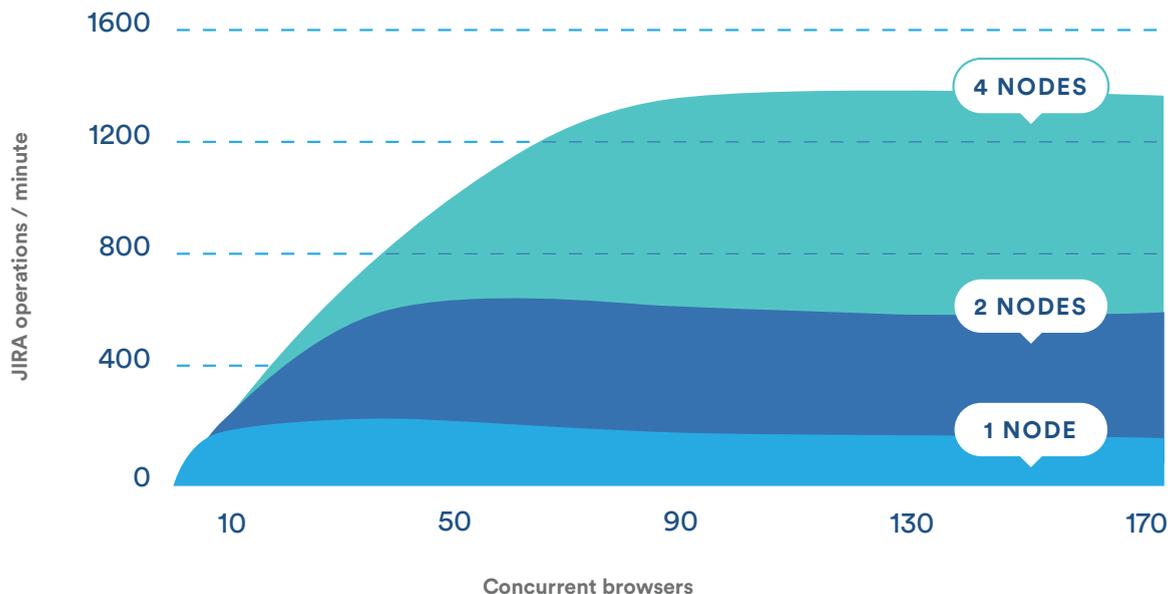




Performance at scale

Each node in a Data Center cluster increases capacity for concurrent users so that you can scale your Atlassian application without sacrificing performance. Dedicate nodes in your cluster for automated tasks or route certain types of traffic – particular teams, or API traffic – to specific nodes while driving the remaining traffic to others to provide the highest quality of service.

Avoid performance degradation during load spikes by increasing application throughput.

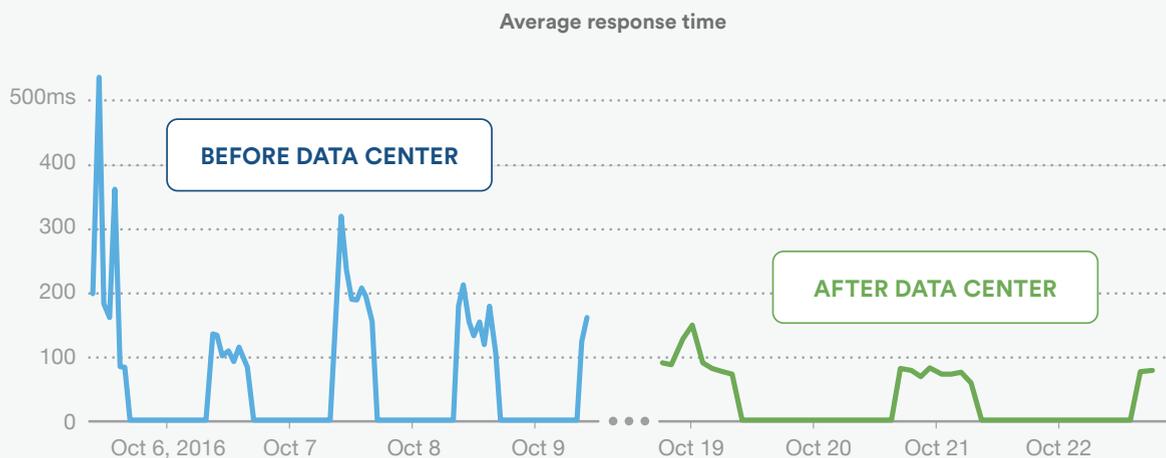


In controlled load testing, we found 2 and 4 node JIRA Software Data Center clusters had near linear capacity increases with the same response time. *Atlassian Performance Labs*

Customer story: Cerner Corporation

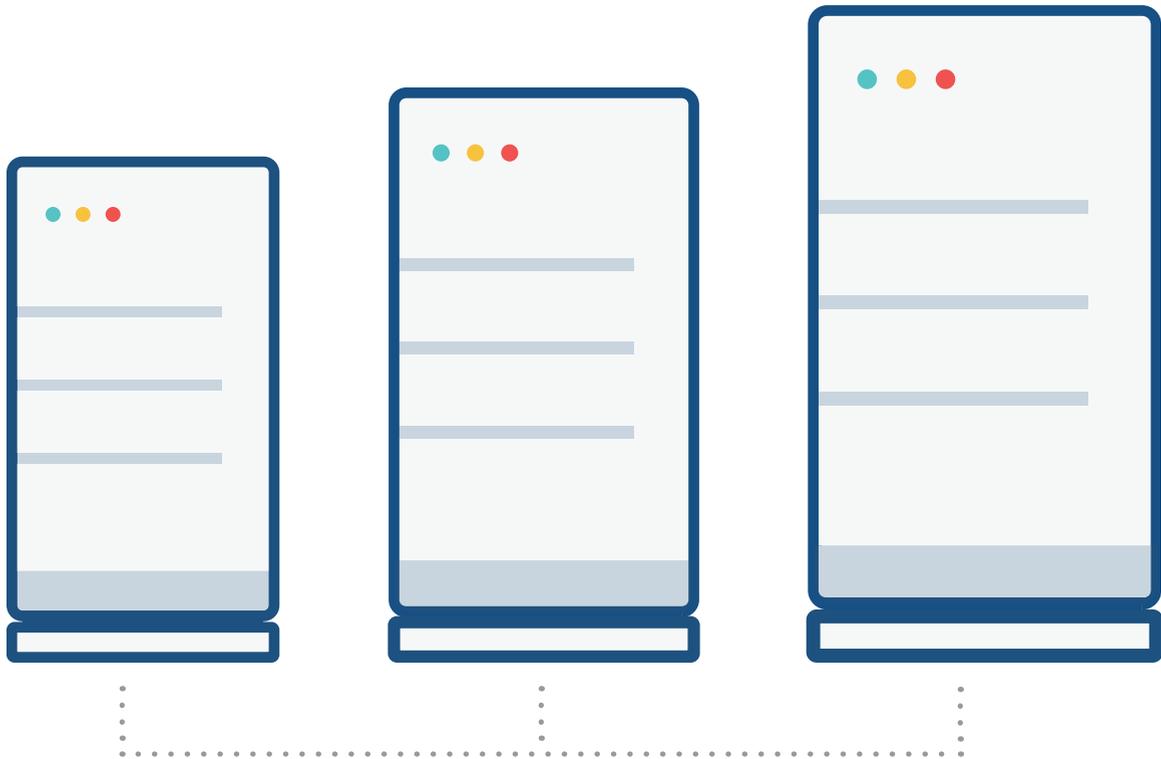
Cerner Corporation drastically reduced application response times by intelligently distributing external REST API requests to a dedicated node. Within their first week of implementing this Data Center configuration, Cerner was seeing 4 times the amount of traffic on the REST API node as on the other two nodes. Response times are faster, CPU utilization has decreased across their non-admin nodes compared to a single server instance, and they haven't seen a single unplanned outage in 2016, all while scaling JIRA Software to thousands of new users.

Cerner needed to make sure that as they continued to add users that application responses times maintained or improved. This architecting proved that Cerner was able to reduce their response times by nearly half, from 150ms to 80ms. Even at peak traffic times – looking at page loads specifically – response times remained steady.





WHAT IS DATA CENTER?



Instant Scalability

Easily add new nodes to your Data Center cluster without any downtime or interruption to services. Existing nodes in the cluster will automatically sync indexes and plugins with each new member giving you a hassle-free deployment and ensuring users have maximum uptime.

And since Data Center is licensed by user count, you can better predict costs and scale your environment without additional licensing fees for new servers or CPU.



Disaster recovery

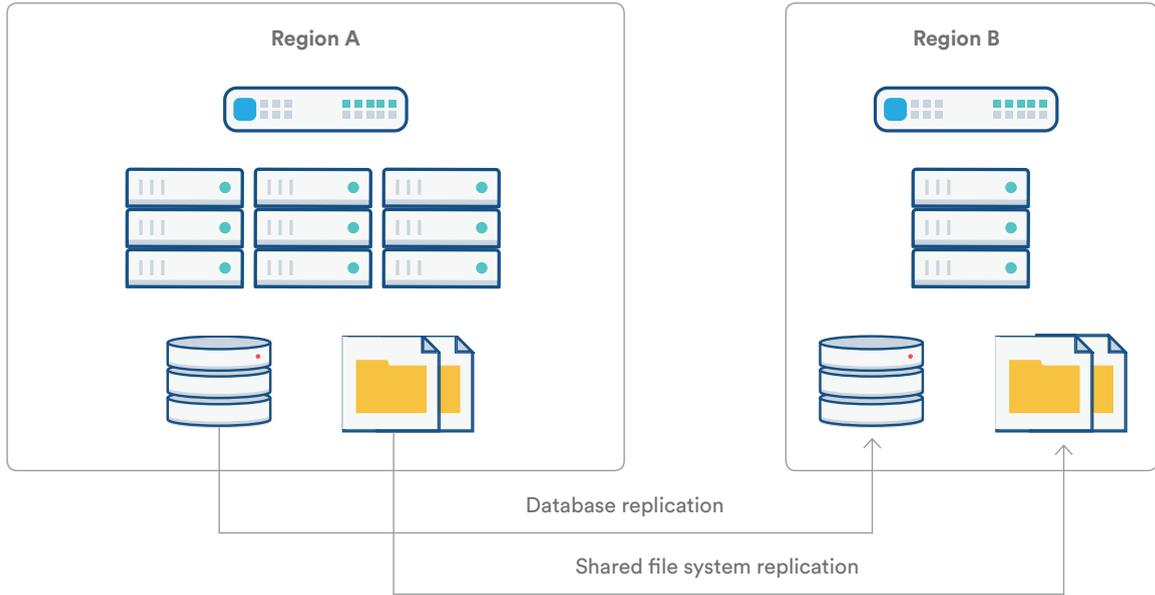
Atlassian's Data Center gives you the ability to have a “cold standby” disaster recovery site for use in a complete system outage. This means the disaster recovery site is located separately from production, and is only on when it's needed. You can implement whichever processes work best for you in terms of replicating your database and shared file system from production to the disaster recovery site.

While you might have a disaster recovery (DR) strategy in place for your current Atlassian environment, with Data Center you're able to share application indexes from your production instance with your DR backup in addition to copying the database and shared storage folders. So in case you need to do a failover, these indexes dramatically reduce the startup time of your DR backup. In the event of a disaster, you can redirect users to the DR system and back online in no time. It's recovery at the flick of a switch.

For more details on how to setup your disaster recovery system for Data Center check out the Resources section.



WHAT IS DATA CENTER?





Smart Mirroring for Bitbucket Data Center

Bitbucket Data Center's Smart Mirroring drastically reduces clone times by making repos available at a nearby server, cutting clone time from hours to a few minutes.

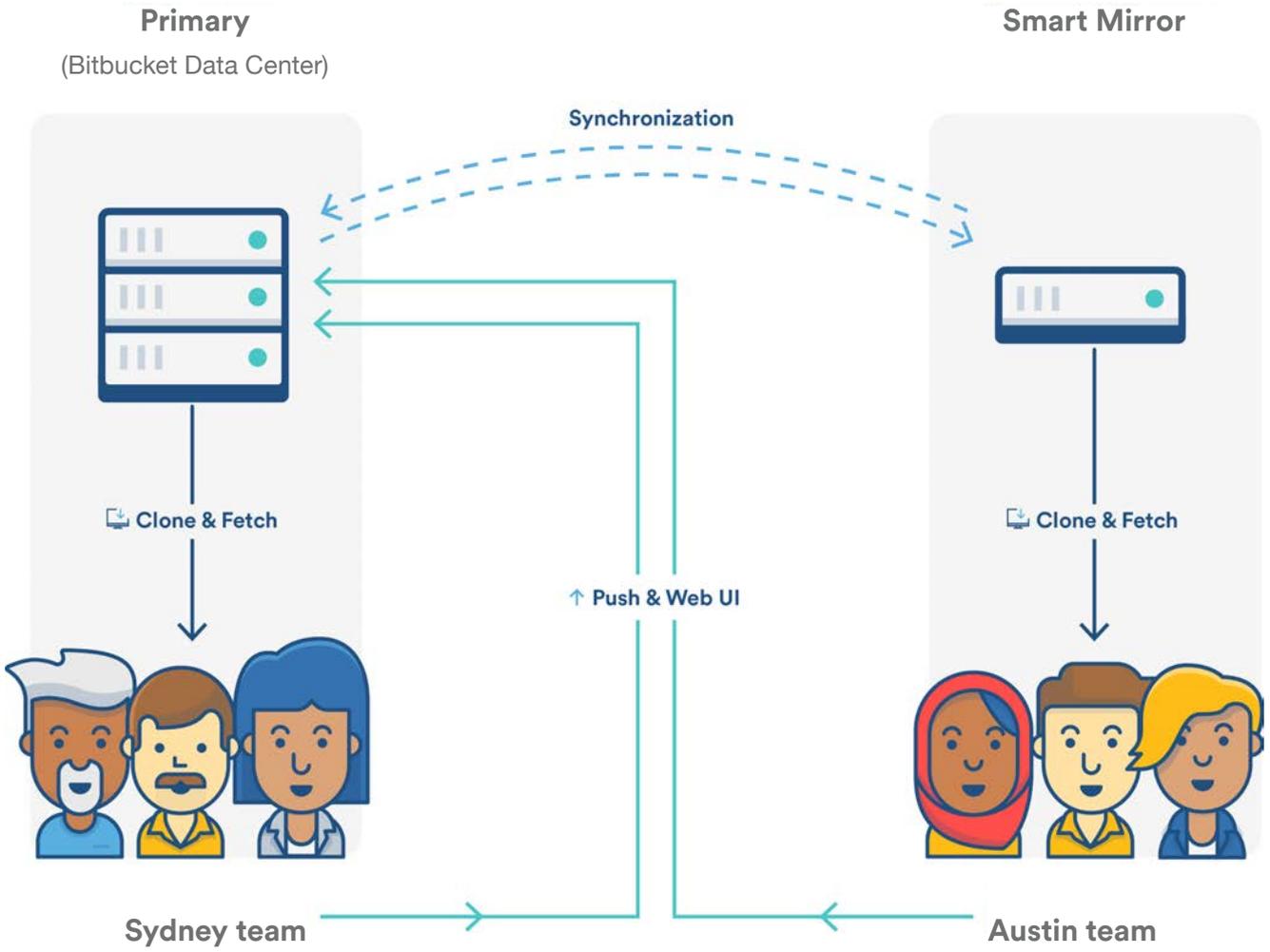
Many software development teams using Git have large repositories as a result of storing a lot of historical information, using monolithic repositories, or storing large binary files (or all three). Companies with distributed software development teams often have little control over the network performance available to them between sites. In combination, this leads to lost development time when developers have to wait long periods, often hours, to clone a large repository from across the world.

Smart Mirroring gives you back that lost development time by allowing you to set up live mirror nodes with read-only copies of repositories in remote locations. The mirrors automatically keep all repositories hosted on them in sync with the primary Bitbucket Data Center instance. Users in those remote locations may clone and fetch repos from the mirror and get identical content, faster. Mirrors can be configured to mirror all repositories in all projects from their primary Bitbucket instance, or a selection of projects configured by an administrator.

Here at Atlassian, we've seen 25x faster clone times for 5GB repos between San Francisco and Sydney using Smart Mirroring.



WHAT IS DATA CENTER?

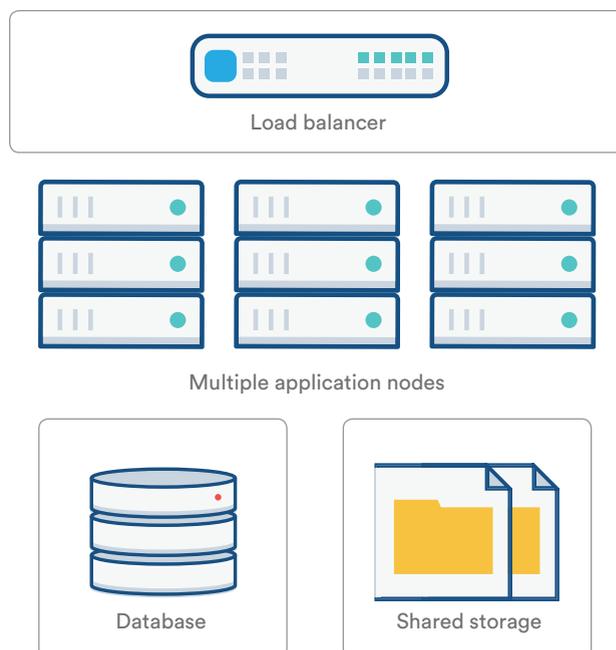




4. Plan, prepare, execute

Before you begin implementing Data Center, it is important to recognize that there is further infrastructure than the multiple application nodes. These elements include a load balancer to distribute traffic between nodes, a shared file system for effective attachment and artifact management and a database to manage metadata. These are the individual elements you will need to build the Data Center deployment.

Throughout this section we reference a recent survey that we conducted with a small sampling of Data Center customers to better understand how customers have set up their Data Center infrastructure. As previously noted, Atlassian does not provide recommendations for hardware or machines but we hope the survey results help inform how you can set up Data Center to best meet your needs.

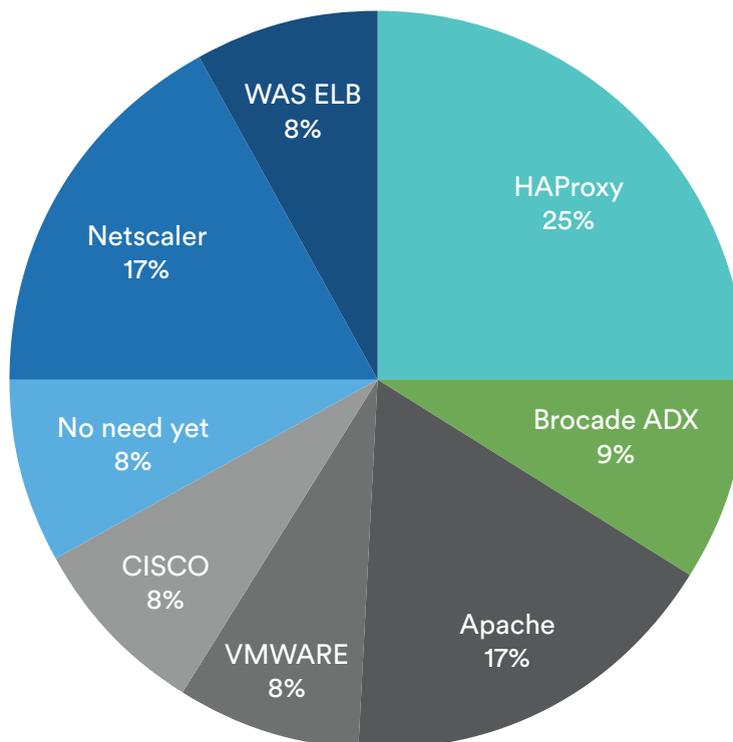




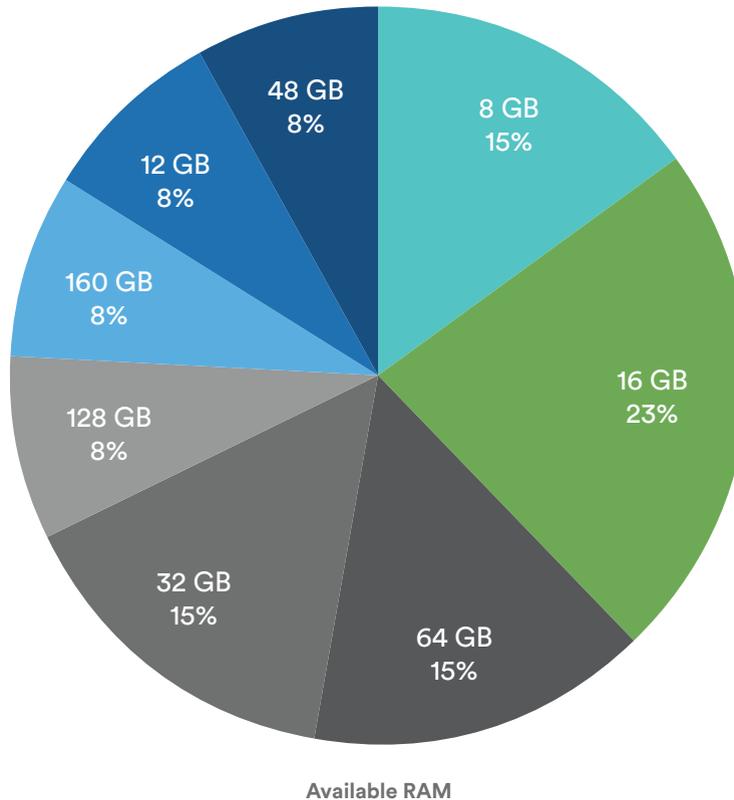
Plan and Prepare

Load Balancer

The load balancer is actually the first stop your requests will make as they come in. The purpose of the load balancer is to direct your incoming traffic to the various application nodes in the cluster. You can set this up to work pretty much however you'd like. You could configure it such that certain types of traffic are sent to particular nodes, or that certain teams have their own nodes. In addition, both hardware (i.e. F5, Cisco, etc.) and software (i.e. Apache) based load balancers are supported. The only specification required is that the load balancer is configured for cookie-based session affinity (also known as sticky sessions.) This means when a user enters the application, they remain on a single node for the entirety of their session. As you can see in the survey results below, there is no one-size, or one-machine, that fits all.



Load balancers used in Data Center deployments

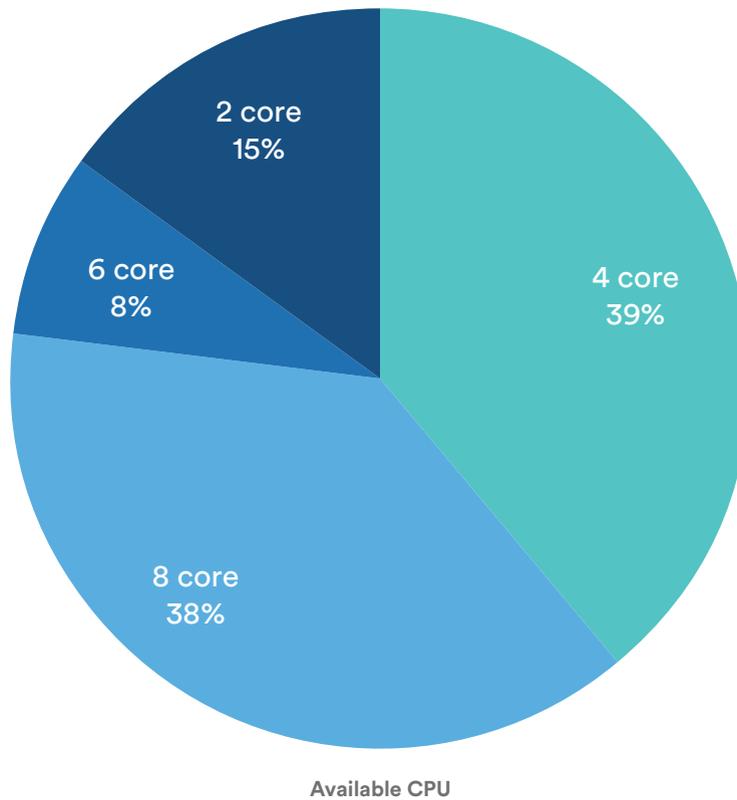


Application Nodes

The application nodes are where the actual Atlassian application lives. Each node will have its own install of the software (JIRA Software, JIRA Service Desk, Confluence, or Bitbucket). These nodes will be configured in a cluster, acting as one, serving the application to your users.

Each node in your Data Center cluster must run the same version of the application and located in the same location, or data center.

Data Center pricing is not dependent on the number of nodes you have, meaning you can have as many as you'd like. We have found that typically between two and four nodes is sufficient for nearly all organizations. In general we recommend starting small and growing as needed.



A Note on High Availability

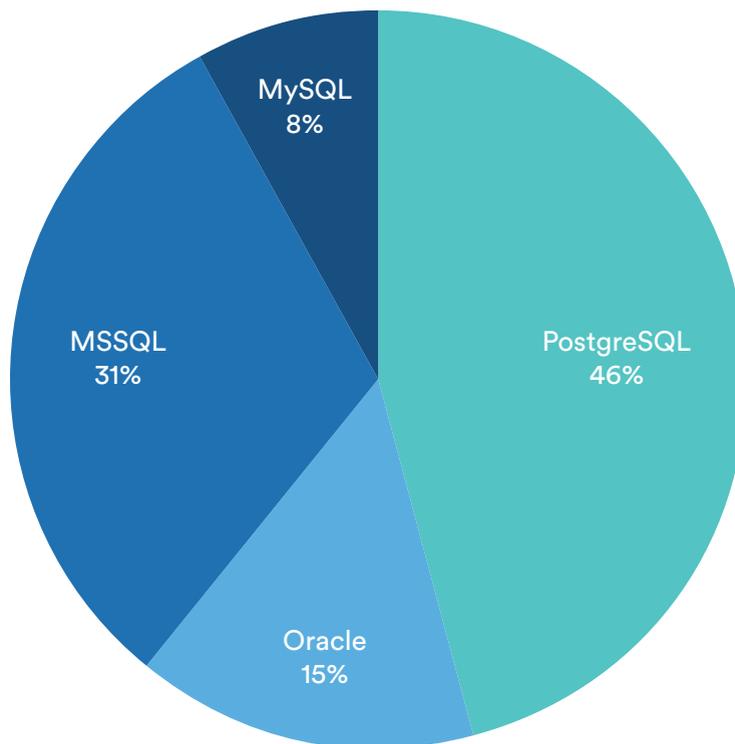
In the installation process, we recommend starting with one application node to ensure that the application is working as it should. When testing has confirmed proper functionality, add another application node to the Data Center cluster. At this point test that the load balancer is directing traffic between the nodes properly; if so, the Data Center now has HA. From here, more nodes can be added at any time if necessary.



Database

In Data Center, it is required that the database be installed on its own node. If you so choose, clustered database technology is supported and recommended as it provides further resiliency to your system; although, a clustered database is not required. Data Center supports the same databases as our server offering but be sure to consult the supported platforms page to ensure that your preferred database technology and version are supported.

For customers moving to Bitbucket Data Center, please note that Bitbucket Data Center does not support MySQL.



Databases used in Data Center clusters



Shared File System

The shared file system is used by the Data Center deployment to store plugins, attachments, icons, and user profiles and avatars. This has to be set up as its own node to be used by the Data Center deployment. You can use SAN, NFS, or NAS file sharing protocols for your shared file system. Just be sure to stay away from distributed protocols like DFS as these are not supported and will result in malfunction.

Execute

Before beginning the installation, make sure you have the latest version of each application you plan to upgrade. You can use the following pages to help you find the most recent version of each product:

Bitbucket Server

JIRA Software Server

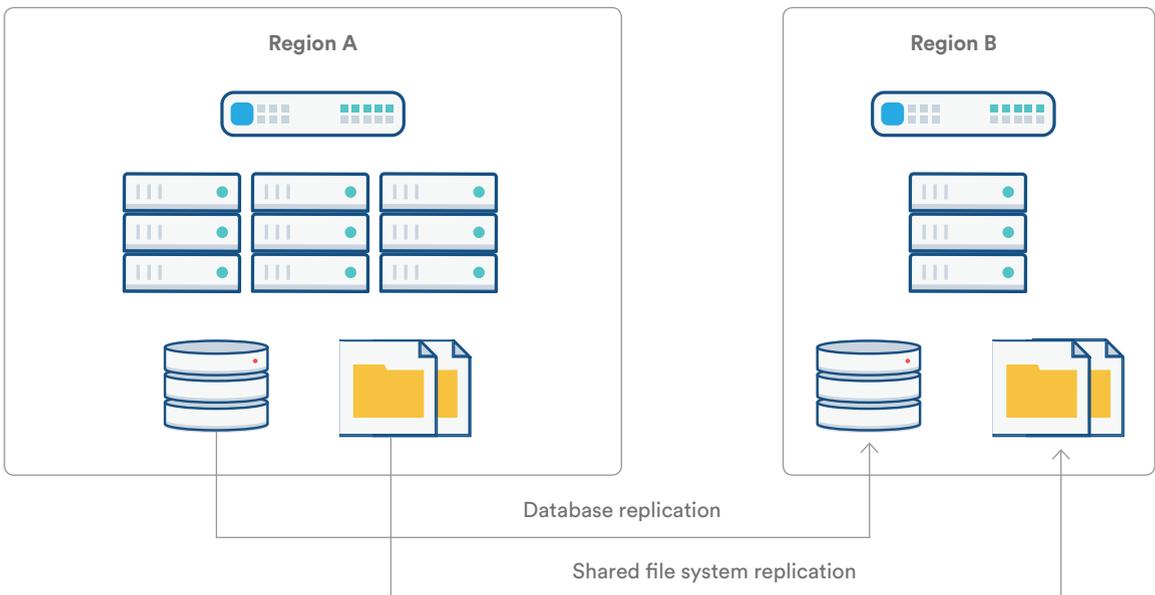
Confluence Server

JIRA Service Desk Server

Infrastructure

We leave it up to you to choose which infrastructure to host your deployment on. Whether it's bare metal servers, virtual machines, or a hosted environment, Data Center runs in whatever environment you prefer. It may be worth noting that in a recent survey of Data Center customers 85% of installations were at least partially virtualized.

Infrastructure as a service is becoming more and more popular amongst advanced IT teams and is compatible with the Data Center deployment option. If you choose IaaS, however, ensure that all instances and services used by Data Center are as collocated as possible. This means that, to the best of your ability, all nodes are located in the same geographical location. For example, in AWS, you can ensure that all nodes are in the same region and subnet. This ensures Data Center will function properly.



Disaster Recovery

To achieve DR with Data Center begin by deploying an offsite DR system. This system will largely resemble the production system limited to one application node. Once the DR system is up and running more application nodes can be added. Next, implement a database replication strategy according to the database technology you have implemented, to replicate your database from production to DR. Lastly ensure that the shared file system is also being replicated from production to DR. There are two ways to do this, first would be a standard replication process in which the whole shared file system is replicated by a process you put in place. The second option is to create a shared file system in DR and mount it to your production system, the application can then be configured to automatically replicate the production file system to this mount.

For more information on implementing a DR strategy, see Disaster Recovery Guide for JIRA - Atlassian Documentation.

Customer Snapshots



JIRA Software Data Center: Industry: Energy

Projects: 52

Environment: Virtual (hosted)

Database: MSSQL

Issues: 12,800

Nodes in the cluster: 2

Load balancer: Apache

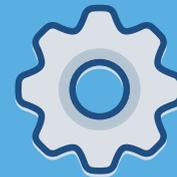
Workflows: 57

CPU on each server: 4 core

DR in place: Yes

Attachments: 4,800

RAM on each server: 8 GB



Bitbucket Data Center: Industry: Software / Technology

Users: 2,000

Environment: Virtual (hosted)

Database: Oracle

Repositories: 3,000

Nodes in the cluster: 3

Load balancer: HA Proxy

Pull requests /day: 200 - 300

CPU on each server: 4 core

DR in place: N/A Atlassian supported DR was not available for Bitbucket Data Center at the time of this survey

Attachments: 4,800

RAM on each server: 48 GB

Confluence Data Center: Industry: Software / Technology

Pages: 2.1M

Nodes in the cluster: 2

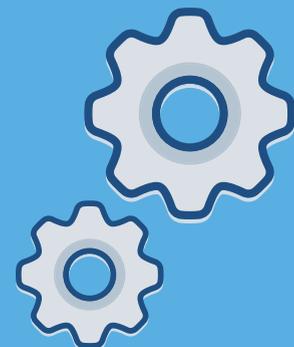
Spaces: 1,400

CPU on each server: 8 core

Active users: 43,000

RAM on each server: 48 GB

Environment: Some combination of virtual (AWS, Azure, etc.) and physical servers





Enterprise Services

To ensure that our customers have every possible opportunity to succeed with complex deployments such as Data Center, Atlassian offers two enterprise services and access to Enterprise Partners.

Technical Account Manager

A Technical Account Manager (TAM) provides proactive and strategic guidance to help you maximize your Atlassian investment. Your TAM will provide guidance on upgrades, be your internal Atlassian champion, give you early access to special alpha/beta/ pioneer programs and much more. Find out more about how a TAM can help you get the most out of your Atlassian tools.

Premier Support

Work with a dedicated team of senior support engineers to diagnose any issues in your Atlassian environment. This team works closely with three primary contacts from your company to learn your network and environment to reduce SLA times and get issues resolved faster. Learn more about our Premier Support offering.

Enterprise Partners

Work with trusted Atlassian partners who specialize in enterprise and complex deployments of Atlassian products. Many of our Data Center customers work closely with our Enterprise Partners during the upgrade, installation and configuration of Data Center. [Click here to find one in your area.](#)

Additional Installation Resources: While we've summarized the key points to consider when planning and preparing to move to Data Center, we have in-depth step-by-step guides with more details. [JIRA Software and JIRA Service Desk Data Center installation guide – click here](#)

[Confluence Data Center installation guide – click here](#)

[Bitbucket Data Center installation guide – click here](#)



5. Refine

Now that you have seen what it's like to implement Data Center, let's examine the tools at your disposal to take Data Center to the next level. Getting the most out of your investment in Data Center starts with knowing the techniques you can benefit from to make Data Center as reliable and scalable as it can get. This starts with monitoring your implementation to understand usage, performance, and what changes (if any) need to be made.

Monitoring

Monitoring should be a cornerstone of any organization's Data Center administration. This is the only way to diagnose performance issues or degradation but more importantly it is crucial to getting ahead of issues in the first place. Different organizations have different requirements around monitoring and there are several approaches you can take. The following are a few ways to break down these options.

Good

At the most basic, you have the option to perform OS level monitoring on your active nodes to get a rough idea of usage. The value of this level of monitoring will be limited to essentially RAM and CPU usage, but will still paint a rudimentary picture of load. Many organizations require that hardware perform within certain OS level thresholds (for example, maintain load of < 60% CPU usage) this is a great place to start in understanding the health of your system.



Better

In addition to OS level monitoring, you should look into monitoring JVM and Tomcat applications and there are a number of tools designed just for that. JMeter, for example, is an industry standard for JVM monitoring that can help you understand not only the load on your system but help you differentiate which processes are having what effect. Enterprise grade implementations of Data Center run all sorts of processes in addition to general user traffic – API calls, automatic queries, dashboards, and plugins – all tax your system in different ways. Using a JVM monitoring tool will help you start to understand the lifecycle of your load much more specifically. It's at this point where you can really start to understand when it is appropriate to add hardware to the implementation based on application usage.

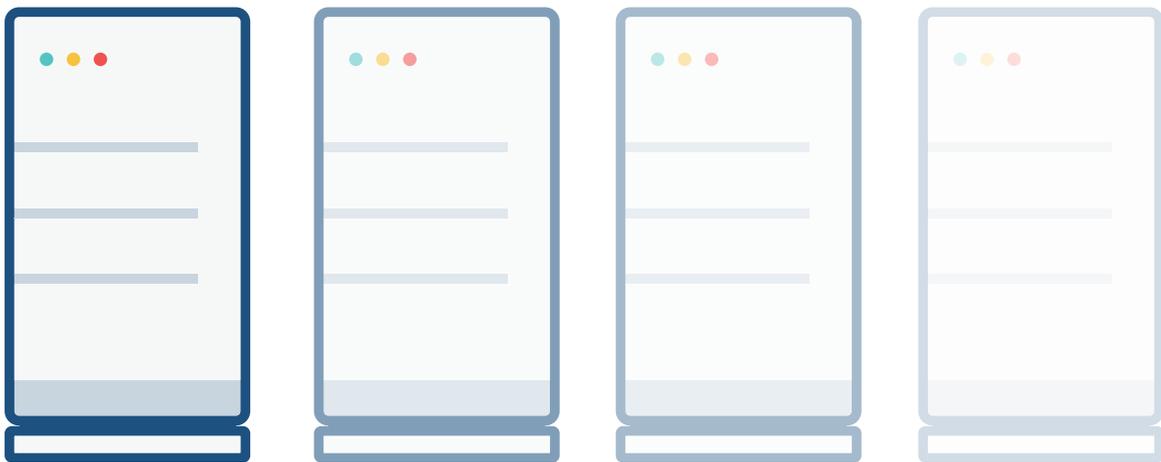
Best

Even monitoring tools won't necessarily identify nuances of your usage like, who is making requests and the result that has on the quality of service. This is where log monitoring comes into play, both at the application and JVM level. Proactively consulting logs for data will help you understand what you are serving, to who, and to what quality. A combination of these three techniques will result in the most comprehensive view possible of the health and performance of your system.

Whether you are a team of 25 or 25,000, growth happens to teams that do great work and you can trust that Atlassian's Data Center offering will be there to grow with you. The easiest and most effective way to scale Data Center is to add new nodes to your cluster to accomodate additional load or increase concurrent usage.

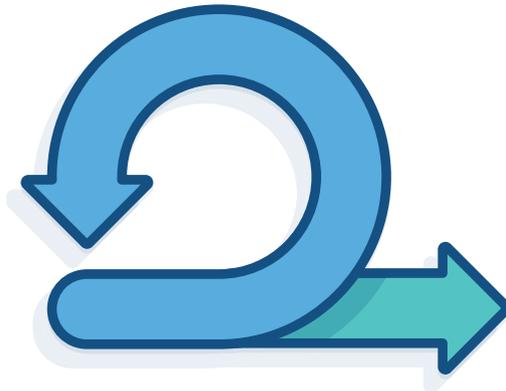


Several modern hosting technologies include autoscaling options for adding nodes automatically based on usage. If your system reaches certain levels of load, new nodes can be spun up automatically. This removes all manual steps from the process and also reduces costs by only bringing on additional nodes when they are needed.



Scale: Grow your Cluster

As discussed in Section 4, we've found there to be a near linear correlation between capacity and the number of nodes in your cluster; in other words, 4 nodes have nearly double the capacity for concurrent usage as 2 nodes. Nodes can be added to your cluster at any time, with no need for downtime. Simply provision a new machine with the application installed and add it to the cluster using the administrative controls. When a new node is added, indexes and plugins will be shared with it to ensure that the new node performs just as the existing nodes do. To make this process even easier, take advantage of server images to provision a new node using an image of an existing node with the application already installed. By using images in this way you can provision new nodes for your cluster almost instantaneously.



Constantly Inspect and Adapt

Monitoring and scaling are not just processes that occur during configuration but rather you should continue to refine your Data Center deployment through iterations of data-informed changes. This is key to the prolonged success of your installation. The tools afforded to you with the Data Center deployment option allow you to iterate endlessly as you see fit without having to worry about changing costs.

Plus, our team of Technical Account Managers can provide guidance and recommendations for continued improvement.



6. Hear it from our customers

“Splunk relies on Confluence Data Center for several aspects of our engineering process including internal team and project websites, automated release note generation, and numerous project and policy documents. Confluence is also tightly integrated with the rest of the Atlassian stack, including JIRA and HipChat, so we can more efficiently produce world-class software with a complete end-to-end solution.”

– Kurt Chase, Director of Release engineering

“Atlassian tools are used by 5000+ employees worldwide and have become key to our teams. We can't afford downtime or latency so we are moving to Data Center solutions to provide around the clock access without compromising performance.”

– Frederick Ros, Head of Quality and Lifecycle Management, Amadeus



How Cerner scaled to 10,000+ users

What do you do when your JIRA Software instance grows from 10 to 10,000+ users? This was what Brian Wallace, Vice President, and Mike Damman, Knowledge Architect, at Cerner, the leading U.S. supplier of healthcare information technology, needed to answer in order to meet the needs of their growing teams. How do you guarantee reliability across such a large instance? What can be done to mitigate the effects of downtime? These were some of the questions tackled with the Cerner team while we identified challenges and came up with some great solutions.

Challenge 1: scale JIRA Software globally

Cerner had three federated instances of JIRA Software Server with thousands of developers using each instance every hour of every day around the globe. JIRA Software quickly became mission-critical and every minute of downtime or performance degradation made it more difficult for Cerner team members to support their customers. They needed a solution that provided high availability.

In the fall of 2015, Cerner chose to upgrade from Server to Data Center so that they could cluster multiple active servers and provide users with uninterrupted access to JIRA Software. This wasn't just critical at the time, but knowing that they were going to add several thousand more users in the coming year, they needed a solution that would scale with them.

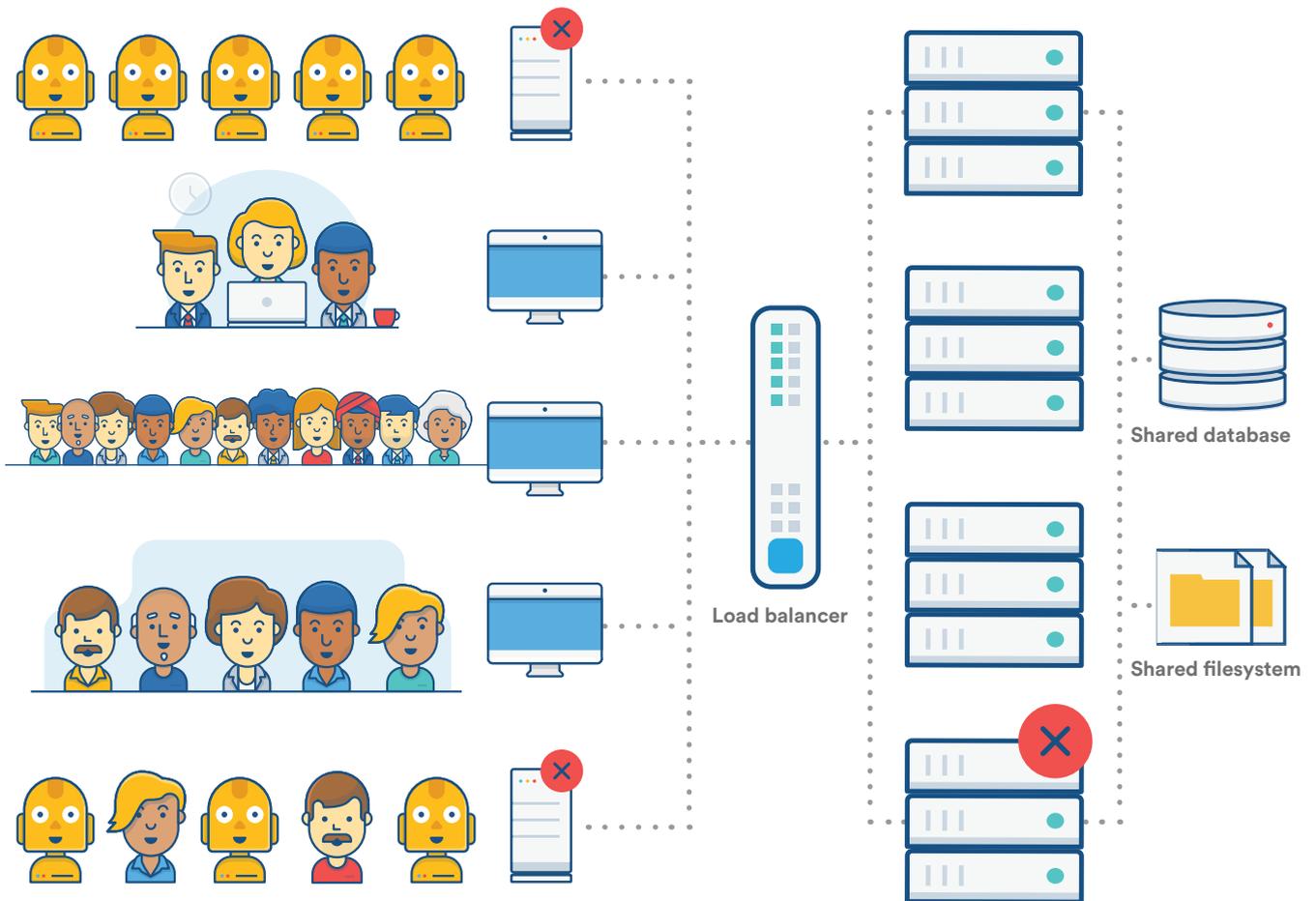
Challenge 2: risks to high availability

Using Zabbix and Splunk to monitor their JIRA instances, Cerner was able to identify one area that needed to be addressed immediately if they wanted to provide true high availability: REST API abuse. Their log analysis showed that team members were using the REST API to get real-time status updates – so whether teams knew it or not, they were pinging JIRA Software instances every single second. Cerner didn't want to restrict users from creating custom dashboards or self-serving the data they needed, but it was obvious that they had to do something different.



HEAR IT FROM OUR CUSTOMERS

“We wanted to be able to isolate REST calls to a single server so that it didn’t have an impact on other users,” Damman noted. With a multi-node cluster they could intelligently distribute traffic by dedicating one node solely to external REST API requests. Cerner also wanted to guarantee that all external requests went to this dedicated node because having users manually change the domain to an IP address, or another domain, wasn’t reliable. That’s when they reached out to their Technical Account Manager (TAM) to help them come up with a better solution.



The impact and rate of growth for external integrations (robot) differ from human interactions and can add stress to individual nodes in the cluster, bringing down everyone else using that node.

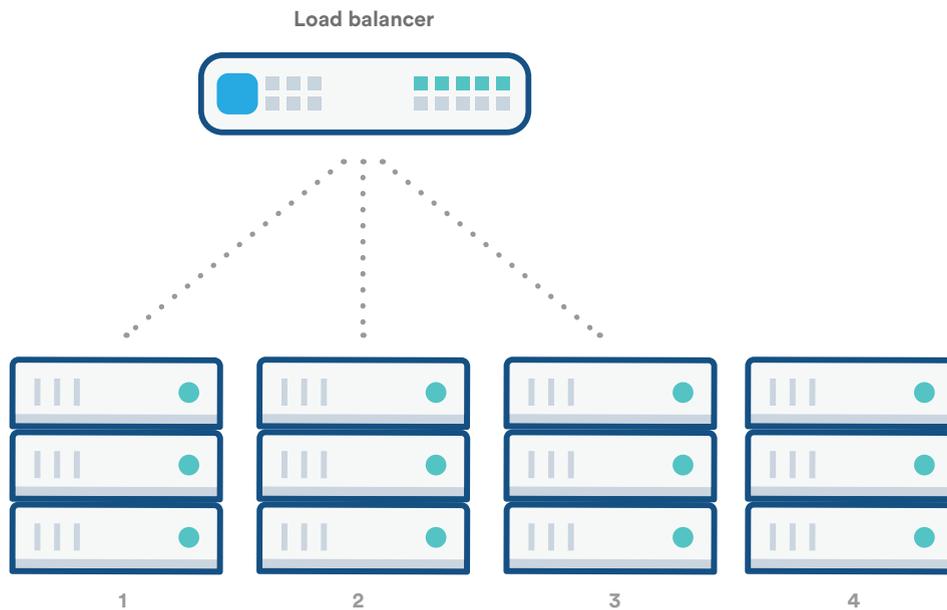
**Solution: intelligently route traffic**

Cerner needed the Data Center configuration to ensure all external REST API requests were routed away from other traffic. They planned to have four nodes in their cluster behind a load balancer with each node performing the following services:

Node 1 – External REST API node

Nodes 2 & 3 – Normal usage nodes

Node 4 – Admin and power user node; not in the load balancer and only accessible by IP address



Their TAM originally thought the best option would be to use the load balancer to route all requests with '/rest' to the REST API node. However, after some testing they found the REST API was also being used throughout JIRA Software, including the login page, so leaving it to '/rest' would mean they'd still be mixing REST API traffic with normal usage.

Working with some of the other Technical Account Managers they found that they could isolate REST API requests by looking for '/rest' in each request AND by looking at where the request



HEAR IT FROM OUR CUSTOMERS

originated using the HTTP referer header. If a person was trying to login to JIRA Software or was already using JIRA Software, they would get directed to or remain on a Normal Usage node. Otherwise, if the person or bot was requesting the REST API, they would get directed to the REST API node.

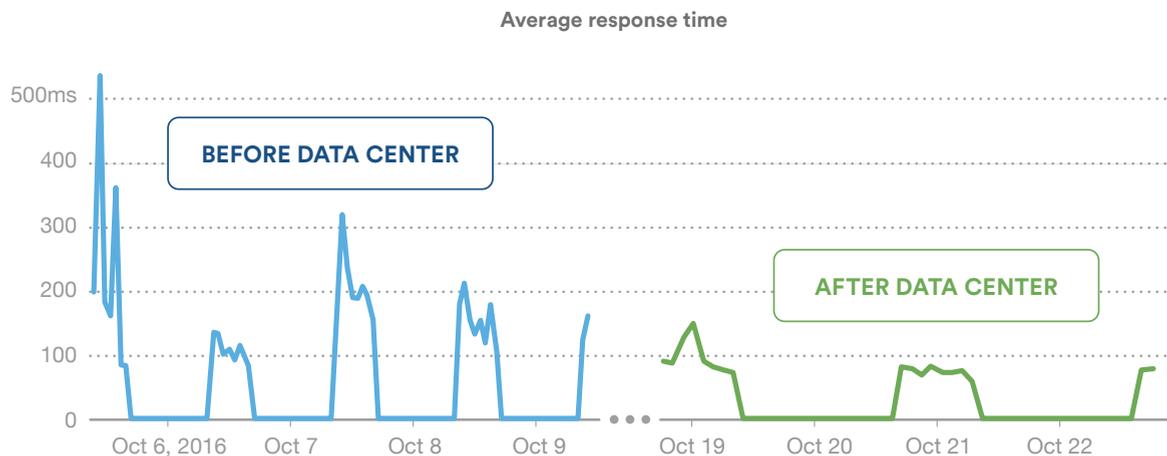
After a few rounds of testing, Cerner went live with JIRA Software Data Center in October 2015.

The results: performance at scale

Within their first week of implementing the proposed Data Center configuration, Cerner was seeing 4 times the amount of traffic on the REST API node as on the other two nodes. Response times are faster, CPU utilization has decreased across their non-admin nodes compared to a single server instance, and they haven't seen a single unplanned outage in 2016, all while scaling JIRA Software to thousands of new users.

Response time

Cerner needed to make sure that as they continued to add users that application responses times maintained or improved. This re-architecting proved that Cerner was able to reduce their response time by nearly half, from 150ms to 80ms. Even at peak traffic times – looking at page loads specifically – response times remained steady.





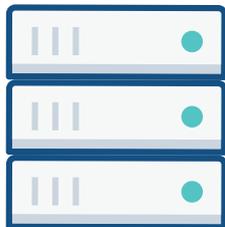
7. What's new in Data Center

Zero downtime upgrades

By clustering multiple servers, we've solved the problem of unplanned downtime but what about planned downtime for things like upgrading software? We know many administrators schedule upgrades over the weekends to avoid productivity disruptions but we know there's a better way – a better way to give administrators their weekends back.

Soon, you'll be able to manage the entire upgrade process without disrupting your end users. This also allows you to upgrade more often so that you get the latest and greatest features (and fixes) to your users faster.

Available for: JIRA Software Data Center





Amazon Web Services support

We want to make managing and deploying Data Center easy, in the environment of your choice. More and more organizations are choosing to host their applications in a virtual environment because it's typically more cost effective and offers a flexible alternative to a traditional data center. Many are leveraging infrastructure service providers like Amazon Web Services to do just that.

With official support of AWS you can now deploy your Data Center cluster on AWS in minutes, including multiple server nodes, databases and a load balancer. Not only do we support AWS, but we worked with Amazon to develop cloud formation templates and Quickstart Guides for each available Data Center product so that you can get up and running in no time. With capabilities like auto-scaling and instant provisioning of nodes, you can much more easily scale Data Center to meet growing demand.

Available for: Bitbucket Data Center and JIRA Software Data Center – coming soon for other Data Center offerings



SAML

As organizations standardize on Atlassian products to support growing business needs, they become central to how your teams work. Making access to these, and the other tools your teams rely on, is a priority for many of our customers.

With SAML 2.0 support for Atlassian Data Center products you can give admins a central location for enabling and managing user authentication while making access to Atlassian Data Center applications easier and faster for users. If a user is already authenticated by your SAML provider, they can quickly jump into JIRA Software Data Center and Bitbucket Data Center without needing to login again.

Available for: JIRA Software Data Center and Bitbucket Data Center



HipChat Data Center

Collaboration is the heart of any well-functioning team. With HipChat, your teams can collaborate in real-time around the globe through 1:1 and group messaging and as more and more teams adopt HipChat within an organization, it quickly becomes mission-critical to how teams function. We're excited to introduce HipChat Data Center, the only on-premise messaging application to work at scale. Soon you'll be able to extend HipChat to every user in your business and ensure uptime and performance. The HipChat Data Center Beta program starts November 2016.



Sources and resources

You can find more information on the topics covered in this paper in our Atlassian Documentation (www.confluence.atlassian.com).

Broad, Stefan. "JIRA Sizing Guide." Atlassian Documentation, <https://confluence.atlassian.com/enterprise/jira-sizing-guide-461504623.html>.

Burwinkle, Christine. "High Availability Guide for JIRA." Atlassian Documentation, <https://confluence.atlassian.com/enterprise/high-availability-guide-for-jira-288657149.html>.

Burwinkle, Christine. "Node Sizing in a Clustered Environment." Atlassian Documentation, <https://confluence.atlassian.com/display/ENTERPRISE/Node+Sizing+in+a+Clustered+JIRA+Environment>.

Burwinkle, Christine. "JIRA Data Center Performance." Atlassian Documentation, <https://confluence.atlassian.com/display/ENTERPRISE/JIRA+Data+Center+Performance>.

Burwinkle, Christine. "JIRA Data Center Health Check Tools." Atlassian Documentation, <https://confluence.atlassian.com/display/ENTERPRISE/JIRA+Data+Center+Health+Check+Tools>.

Mahoney, Claire. "JIRA Data Center Load Balancer Examples." Atlassian Documentation, <https://confluence.atlassian.com/display/ENTERPRISE/JIRA+Data+Center+Load+Balancer+examples>.

Ryall, Matt. "Moving to Confluence Data Center." Atlassian Documentation, <https://confluence.atlassian.com/display/DOC/Moving+to+Confluence+Data+Center>.

Watson, Paul. "Clustering with Bitbucket Data Center." Atlassian Documentation, <https://confluence.atlassian.com/display/BitbucketServer/Clustering+with+Bitbucket+Data+Center>.

Lui, Andrew. "Disaster Recovery Guide for JIRA." Atlassian Documentation, <https://confluence.atlassian.com/enterprise/disaster-recovery-guide-for-jira-692782022.html>.

Brunning, Giles. "Confluence Data Center disaster recovery." Atlassian Documentation, <https://confluence.atlassian.com/enterprise/confluence-data-center-disaster-recovery-790795927.html>.



Paz, John. "Disaster Recovery Guide for Bitbucket Data Center." Atlassian Documentation, <https://confluence.atlassian.com/display/BitbucketServer/Disaster+recovery+guide+for+Bitbucket+Data+Center>.

Paz, John. "Smart Mirroring." Atlassian Documentation, <https://confluence.atlassian.com/display/BitbucketServer/Smart+Mirroring>.

Burwinkle, Christine. "JIRA Data Center Health Check Tools." Atlassian Documentation, <https://confluence.atlassian.com/enterprise/jira-data-center-health-check-tools-644580752.html>.

Paz, John. "Adding Cluster nodes to Bitbucket Data Center." Atlassian Documentation, <https://confluence.atlassian.com/display/BitbucketServer/Adding+cluster+nodes+to+Bitbucket+Data+Center>.

Burwinkle, Christine. "Federating JIRA - Managing Multiple Instances." Atlassian Documentation, <https://confluence.atlassian.com/display/ENTERPRISE/Federating+JIRA+-+Managing+Multiple+Instances>.

Burwinkle, Christine. "Failover for JIRA Data Center." Atlassian Documentation, <https://confluence.atlassian.com/display/ENTERPRISE/Failover+for+JIRA+Data+Center>.

Barnes, Roger. "Distributed teams can now build faster with Bitbucket." Bitbucket Blog, <https://blog.bitbucket.org/2016/01/21/distributed-teams-can-now-build-faster-with-bitbucket>.

Hernandez, Javier. "How to scale JIRA Software to 10,000+ users." Atlassian Blog, <http://blogs.atlassian.com/2016/08/scale-jira-software-10000-users>.

