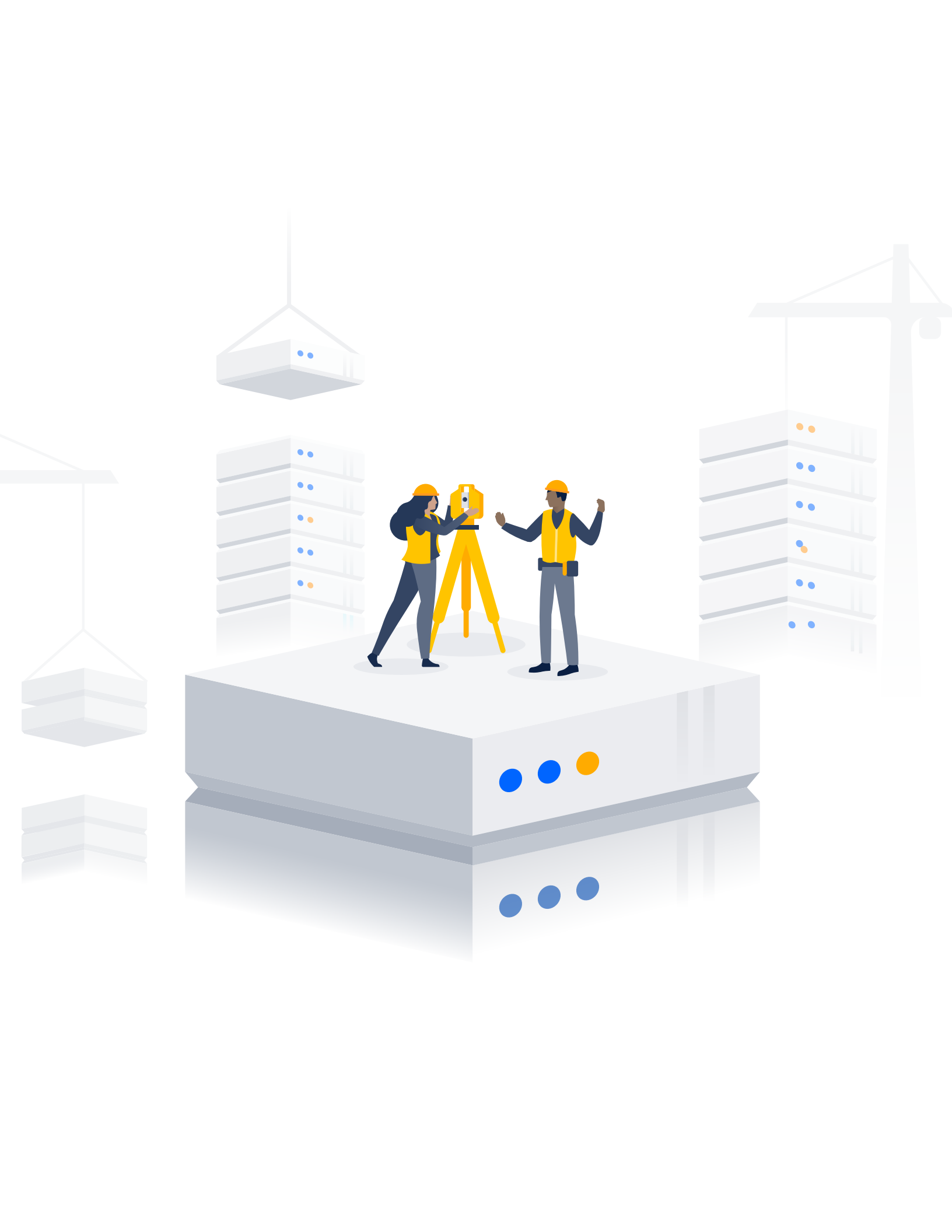
ATLASSIAN

Server to Data Center

The Tipping Point

Atlassian unleashes the potential of every team. Our collaboration software helps teams organize, discuss and complete shared work. Teams at more than 119,000 customers, across large and small organizations—including Citigroup, eBay, Coca-Cola, Visa, BMW and NASA—use Atlassian's project tracking, content creation and sharing, real-time communication and service management products to work better together and deliver quality results on time. Our products include Jira Software, Confluence, Stride, Trello, Bitbucket, Jira Service Desk and Bamboo.



Server to Data Center: The Tipping Point

4 Section I: Scaling Atlassian Applications

11 Section II: The Tipping Point

17 Section III: What is Data Center?

32 Section IV: Prepare, Plan, Execute

43 Section V: Refine

47 Section VI: Hear It From Our Customers

58 Sources and Resources

SECTION I

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, work flows 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. Below is a list of factors that we recommend considering when scaling a single instance of Jira Software. Keep mind that one characteristic may not grow at the same rate as others. Many people consider the amount of users that will have access to the system, but do not take into account the comparative activity of those users. Or conversely, just because the number of issues is increasing doesn't mean that projects will 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 for administrators to limit the number of configurations made to the application.

We've built some sizing guidelines to help you determine how large (or small) your Jira Software instance should be to meet your requirements. 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	100,000
Active (Concurrent Users)	25	200	600	2000
Issues	15,000	60,000	200,000	1,000,000
Issues/month	200	1000	4000	200,000
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
Work flows	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 instances.
- **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 is encouraged.
- **Intentional Federation Setup:** 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 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 III: 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 strategy (DR) is a crucial aspect of any enterprise grade deployment. In the event of a complete site-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. Unlike a single or federated server environment, with Atlassian's Data Center offerings, you gain the ability to copy indexes from primary servers to remote DR servers and assure consistency should the need for failover arise. Index replication has an impact on how quickly your team can get back up and running. Without copies on the backup servers, the application will need to rebuild indexes upon startup and for a large installation this could take several 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

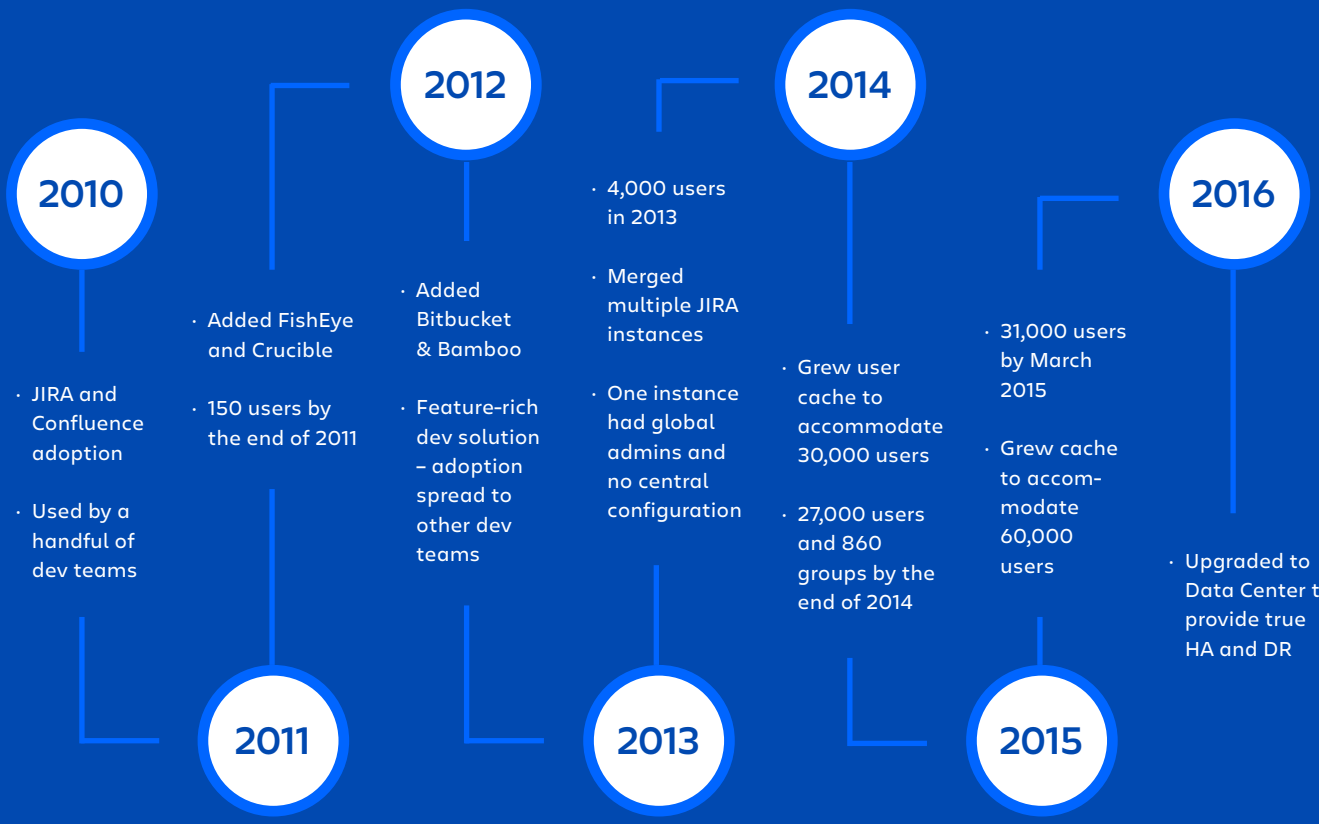
CUSTOMER STORY:

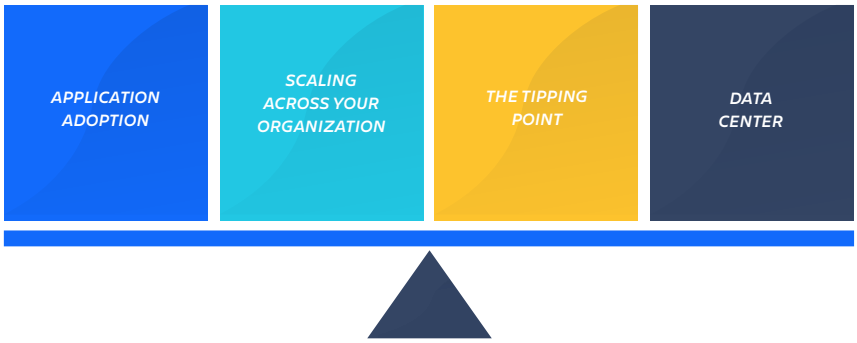
From 0-60,000 users in 5 years

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.





SECTION II

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. As concurrent usage increases, so do response times which leads to user frustration. System administrators are then looking for solutions to minimize pain for users, and themselves. Many global companies experience this when multiple geographic locations comes online at the same time. For us at Atlassian, we experienced this firsthand 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 resulted in our San Francisco and Austin offices, in addition to Sydney, struggling with slow page loads or even taking us briefly offline.

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

> "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 you have 500 engineers working in Bitbucket at any one time and the system goes down for an hour, no more commits or pull requests. All work and productivity comes to a halt. If 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.

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.

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.

SECTION III

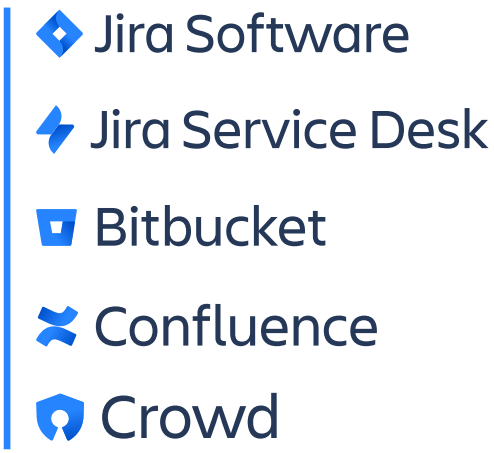
What is Data Center?

Data Center is 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 plug-ins are automatically synced.
- **Deployment Flexibility:** Deploy Data Center behind your firewall using a variety of technologies like VMWare or bare metal, or you can run it on IaaS like AWS or Azure.
- **Authentication and Control:** Standardize and control how end users access and use Atlassian tools with SAML 2.0 support.
- **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:



High Availability

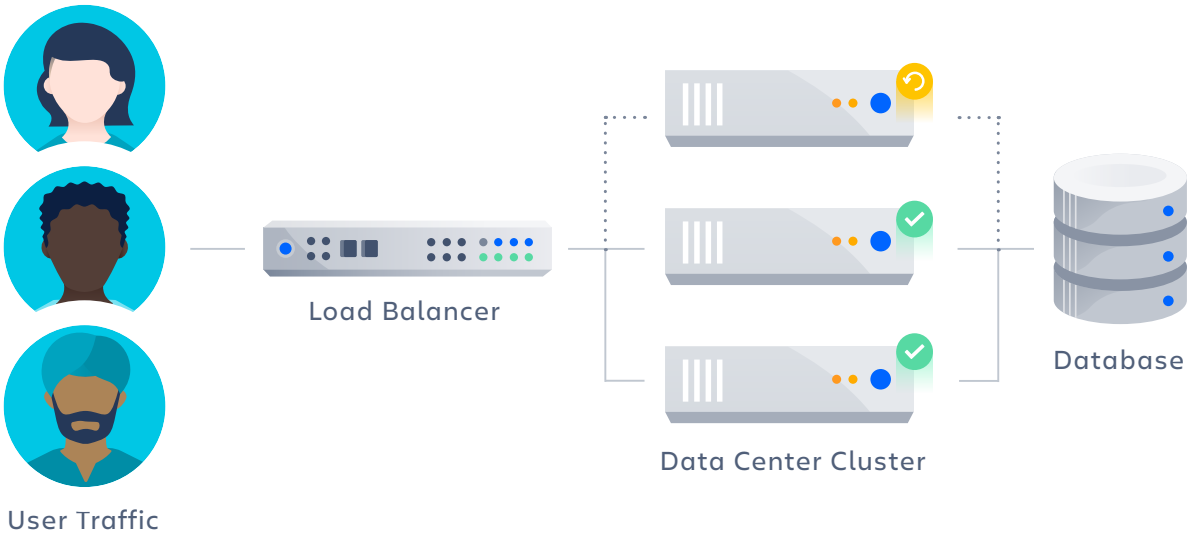
Data Center helps you achieve high availability through active clustering and automatic hot failover within your data center. Cluster multiple active servers to ensure uninterrupted access to critical Atlassian applications in the event of a hardware failure. If a node fails, the load balancer will automatically 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.

We know that work can't stop for maintenance windows, and for many organizations, it can be difficult for admins to find time for planned maintenance. After all, downtime on mission-critical applications can affect productivity and team output.

Zero Downtime Upgrades (Jira Software and Jira Service Desk)

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 believe there's a better way, a way to give administrators their nights and weekends back.

> Let's assume you have a three node Data Center cluster running on version Jira Software and are ready to upgrade to the latest version. To get each node to the latest version, you'll shut down and upgrade one node at a time. All user traffic will be redirected to the two remaining online nodes in your cluster so users can continue with business as usual. Once each node in your cluster has been upgraded, you're ready to apply any outstanding database schema changes.



With zero downtime upgrades, you'll be able to manage the entire upgrade process without disrupting your end users. Confidently upgrade more often so that you can get the latest and greatest features (and fixes) to your users faster.

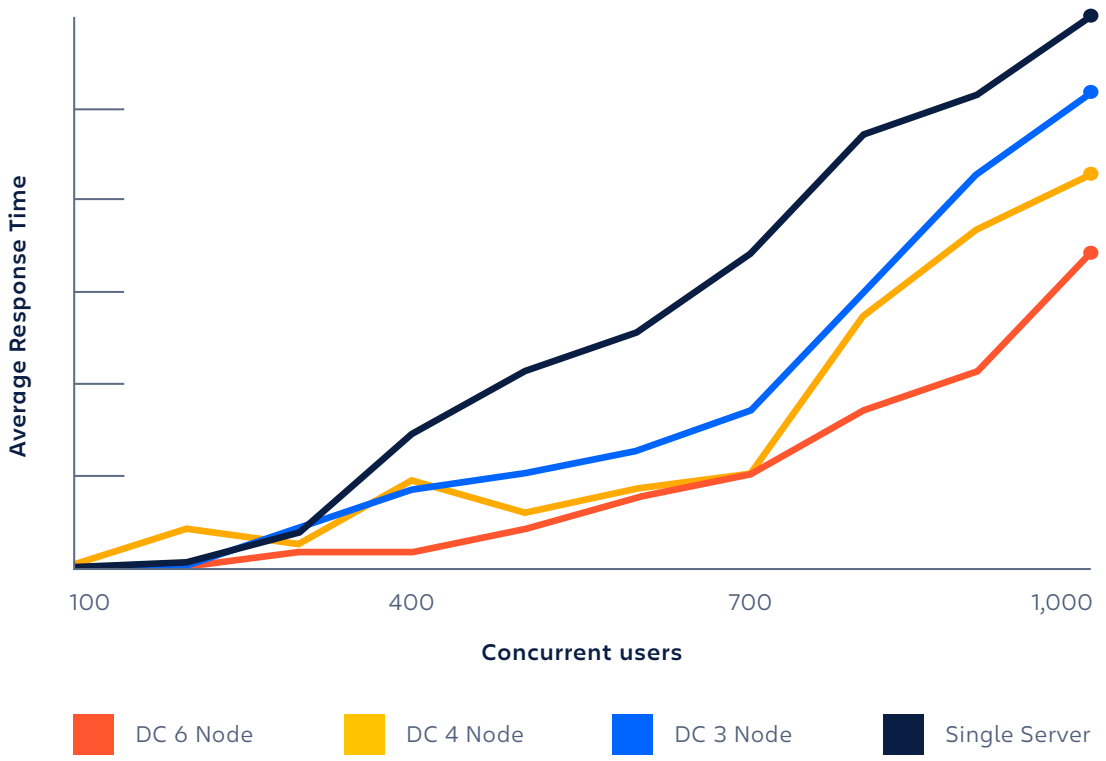
Read-Only Mode (Confluence)

Read-only mode for Confluence Data Center keeps users working while admins perform a range of maintenance related activities. Turning on read-only mode allows users to continue to consume content while you perform a range of maintenance related activities. Users will be able to view, search and navigate pages, blogs and attachments, but they won't be able to edit them. Administration actions however, such as managing apps or changing site configuration, are not restricted in read-only mode. This means you can turn read-only mode on before upgrading your Confluence Data Center instance, consolidating multiple instances, migrating to a new platform, or performing other maintenance work, while ensuring that users still have access to their Confluence content.

As an admin, you'll be able to turn read-only mode on and off via the Maintenance page in the admin console. On this page, we also indicate which user-installed apps have been marked as compatible with read-only mode. If an app is not compatible with read-only mode, you may want to temporarily disable it to prevent users from creating or modifying content.

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.



Project Archiving (Jira Data Center)

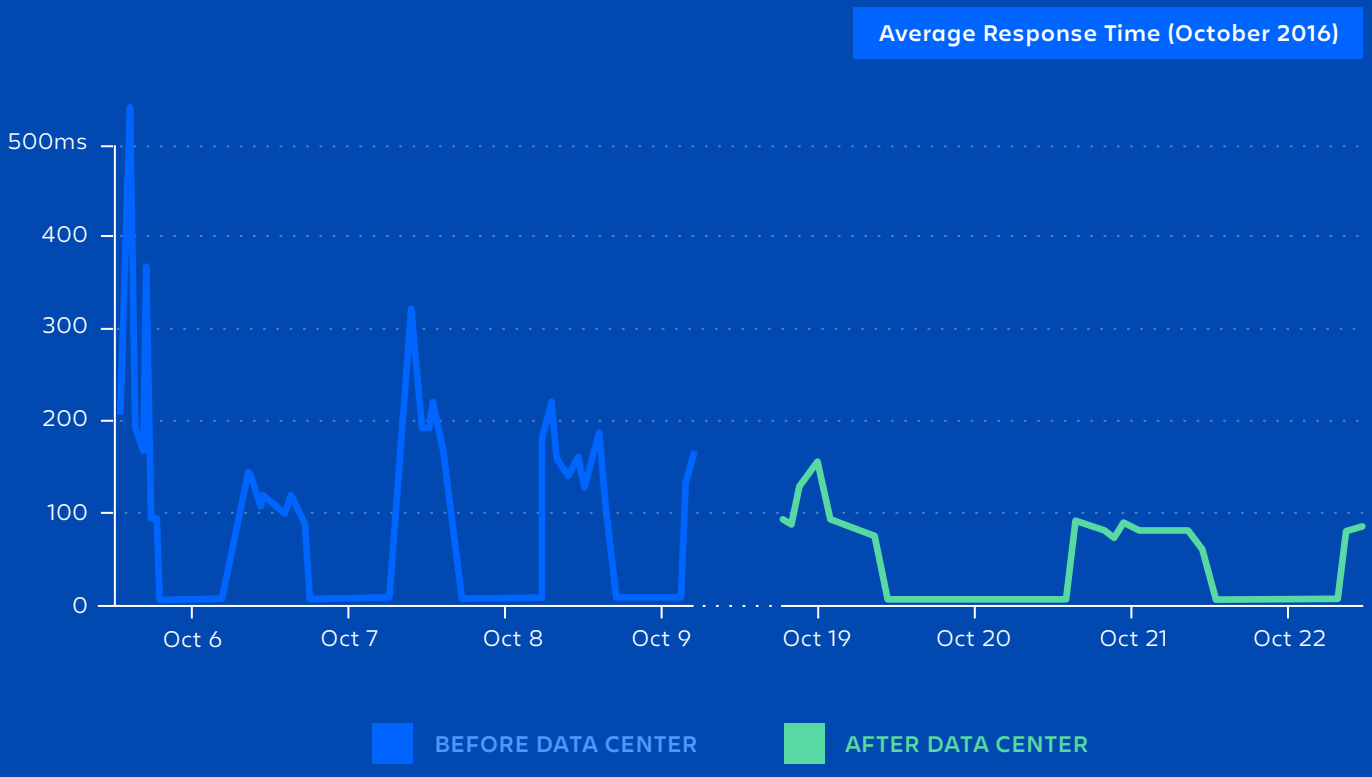
Teams rely on Jira Software every day to get their jobs done, so maintaining Jira's performance and readability is essential. As the scope of Jira grows in your organization, having a plan for how to remove outdated information becomes more important. That's why we introduced project archiving for Jira Data Center. With project archiving, you can create more space and unlock resources for data that's still relevant, making Jira more performant and readable. In our performance testing on a Jira Software instance with 1 million issues where 50% of projects and issues are archived, we observed improvements of 11% faster loading of Jira boards and 25% faster JQL searches. Overall, we see that Jira's performance improves linearly with the amount of information that is archived.

CUSTOMER STORY:

Cerner Corporation

Cerner Corporation drastically reduced application responses 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 architecture 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.



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 using 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.

Deployment Flexibility

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. In fact, 62% of Atlassian's self-hosted customers are choosing to deploy their applications on a virtual architecture. And many are leveraging infrastructure service providers like Amazon Web Services (AWS) or Microsoft Azure to do just that.

AWS

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 Data Center product so that you can get up and running in no time. With instant provisioning of nodes, you can much more easily scale Data Center to meet growing demand.

Azure

Easily get started using a jointly-built Atlassian template for Azure Resource Manager, accessible in the Microsoft Azure Marketplace. Build a Jira Software or Confluence Data Center instance with configurable nodes in a few guided clicks, or simply spin up a test instance with dummy data. Extended services like automatic backups, geo-replication and encryption are supported with Azure SQL Database.

Authentication and Control

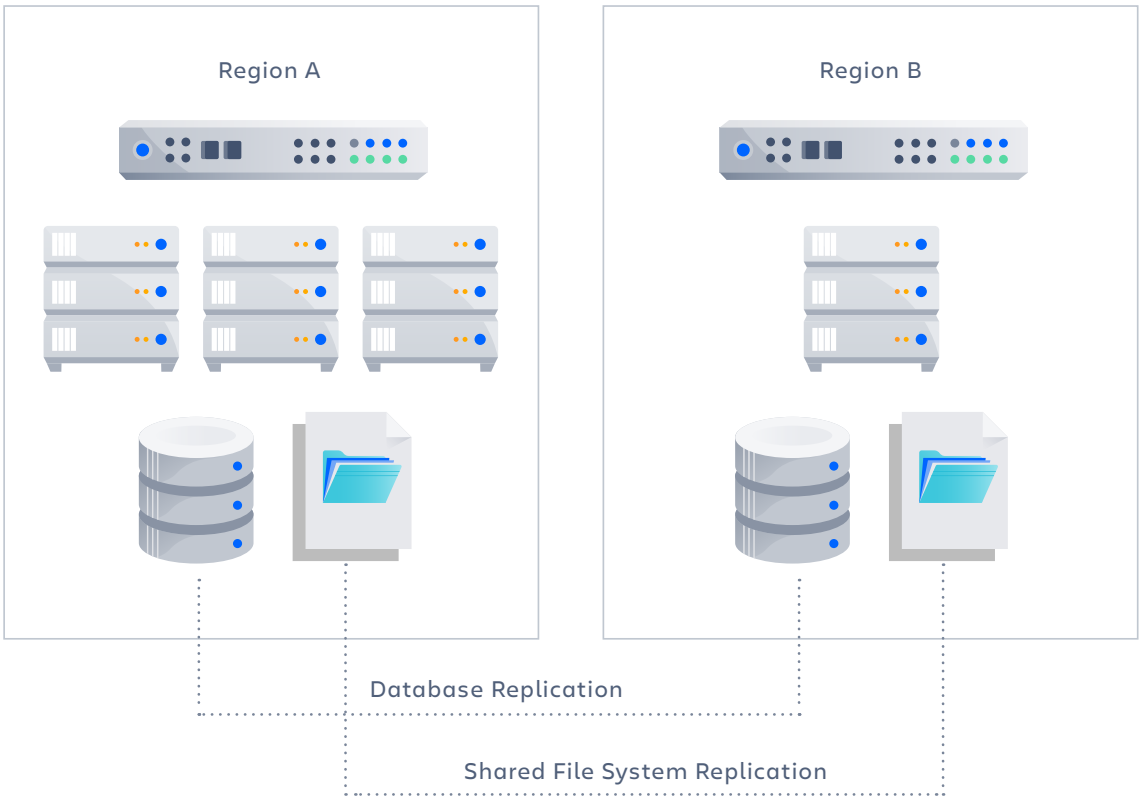
As our products become more mission-critical to your users, it's important that you focus on standardizing and controlling how your end users access and use these systems. For access control, Data Center offers SAML 2.0 support, which allows you to use your existing identity provider for authentication. Not only does it simplify and help you ensure compliance to your company's security policies, but your users don't have to remember (or forget) multiple passwords.

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 plan 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.



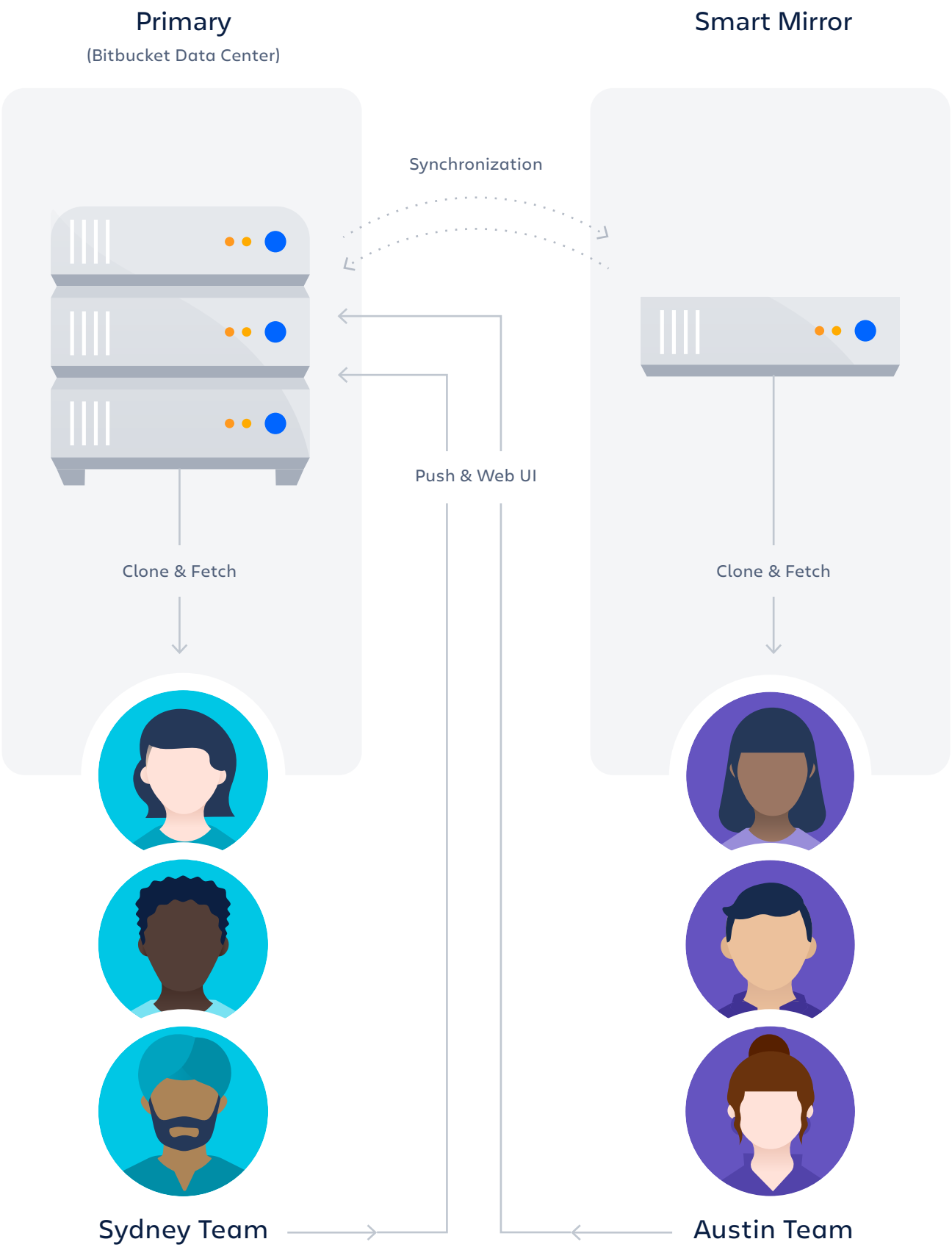
Smart Mirroring for Bitbucket Data Center

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.

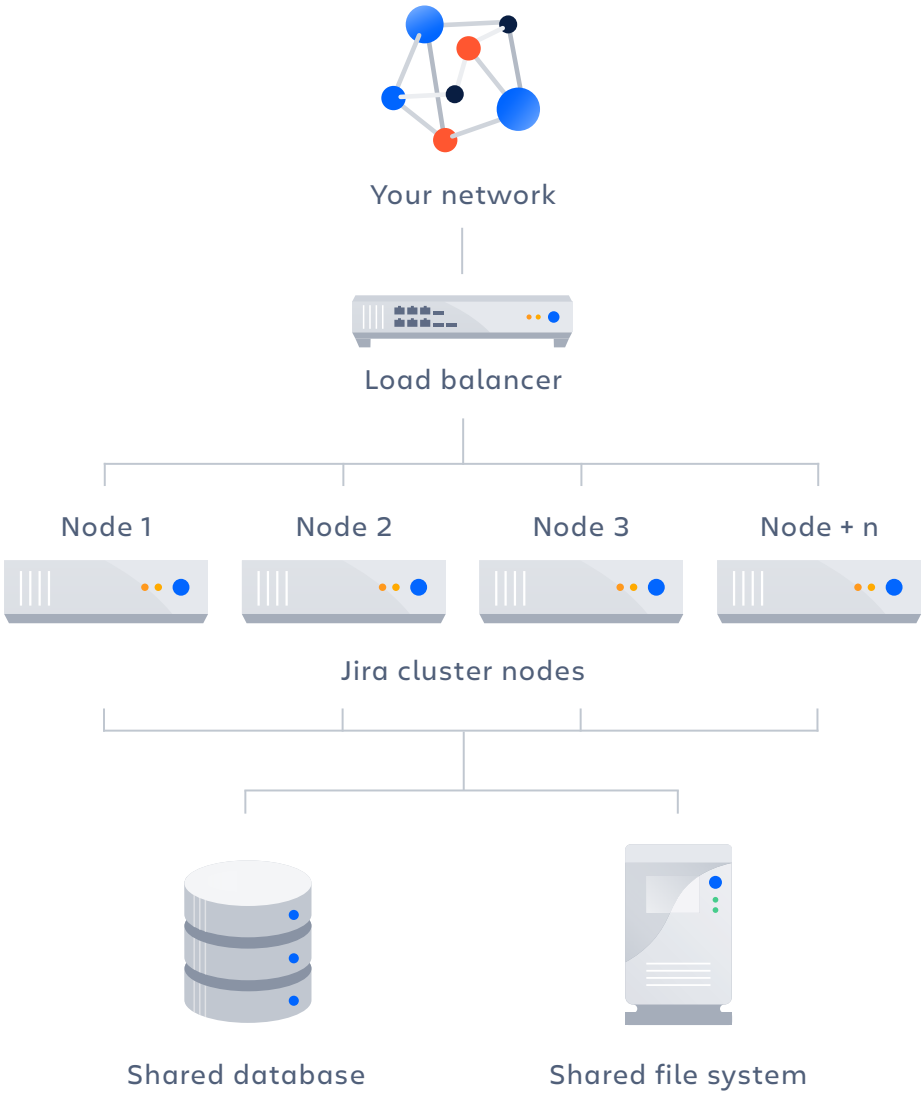
Here at Atlassian, we've seen 25x faster clone times for 5GB repos between San Francisco and Sydney using Smart Mirroring.


Primary
(Bitbucket Data Center)
Smart Mirror
Synchronization
Push & Web UI
Clone & Fetch
Clone & Fetch
Sydney Team
Austin Team

SECTION IV

Plan, Prepare, Execute

Before you begin installing Data Center, let's first understand the pieces of the infrastructure architecture.



In addition to the single server and database that you'd use in a single server deployment, Data Center adds additional application nodes which form a cluster, as well as a load balancer to distribute traffic between nodes and a shared file system for effective attachment and artifact management. 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 setup 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 setup Data Center to best meet your needs.

Plan and Prepare

Load Balancer

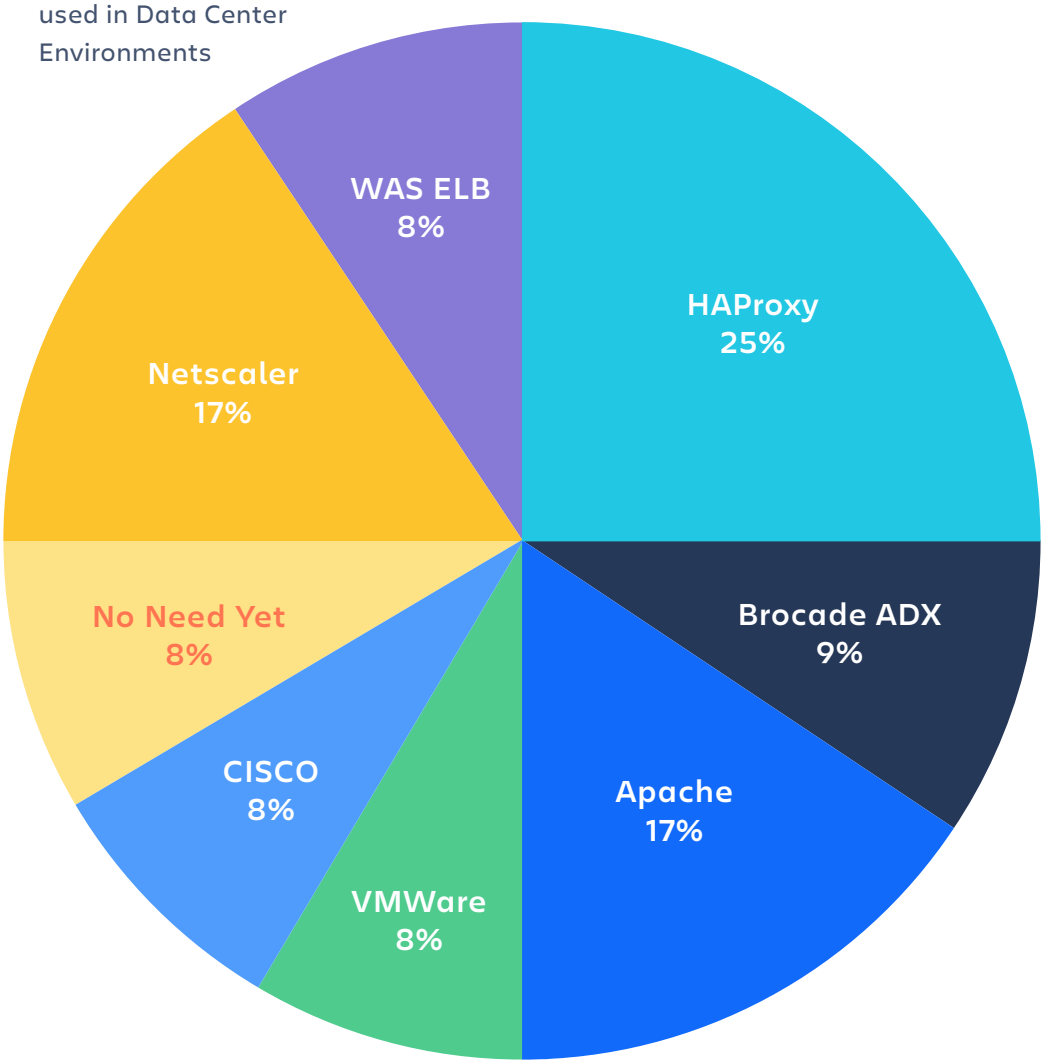
The load balancer is actually the first stop your requests will make as they come in. The purpose of then load balancer is to direct your incoming traffic to the various application nodes in the cluster. You can set this up to work pretty much however you'd like. For example, you could configure it so that certain types of traffic are sent to particular nodes, or that certain teams have their own nodes. Some customers have had

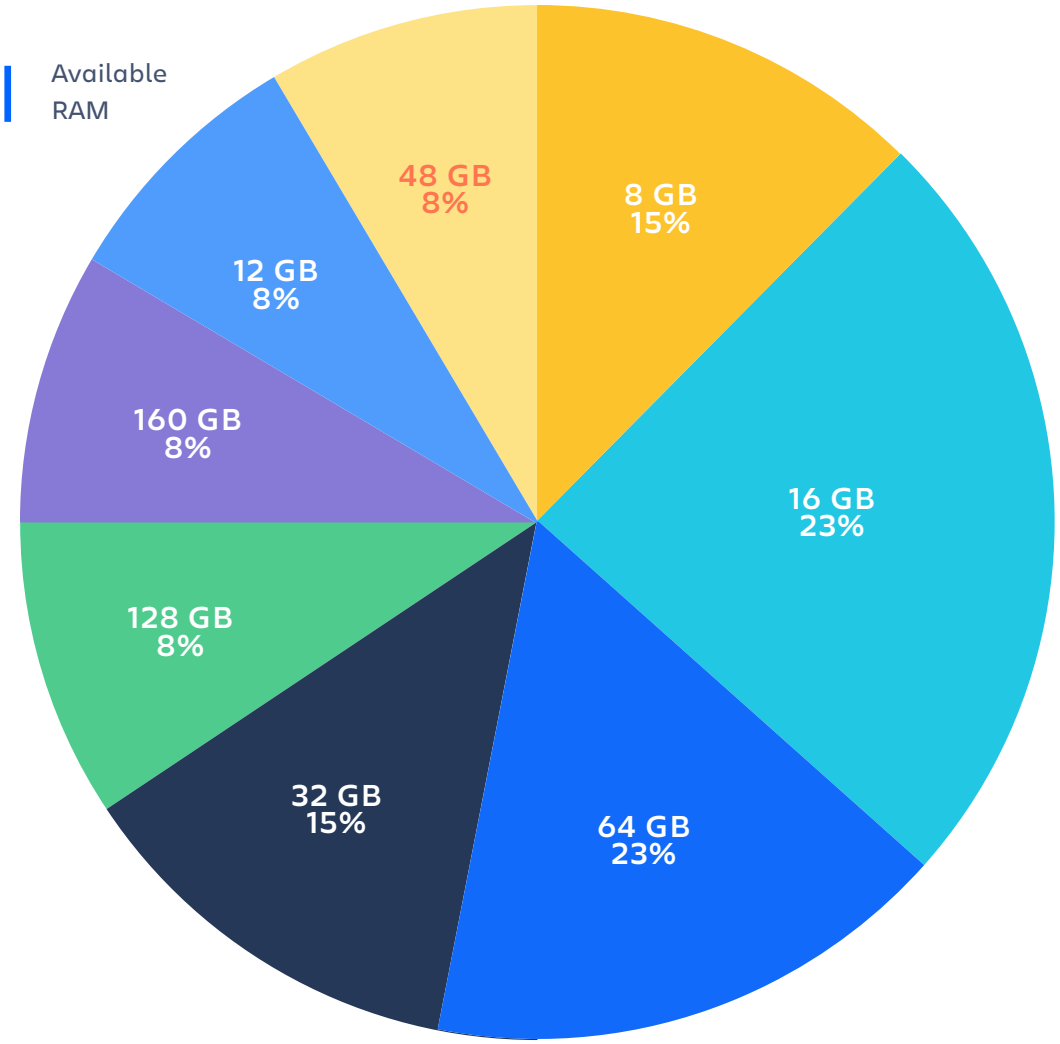
Both hardware (i.e. F5, Cisco, etc.) and software (i.e. Apache) based load balancers are supported but the only specification

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 in the survey results below, there is no one-size, or one-machine, that fits all.

Load balancers used in Data Center Environments





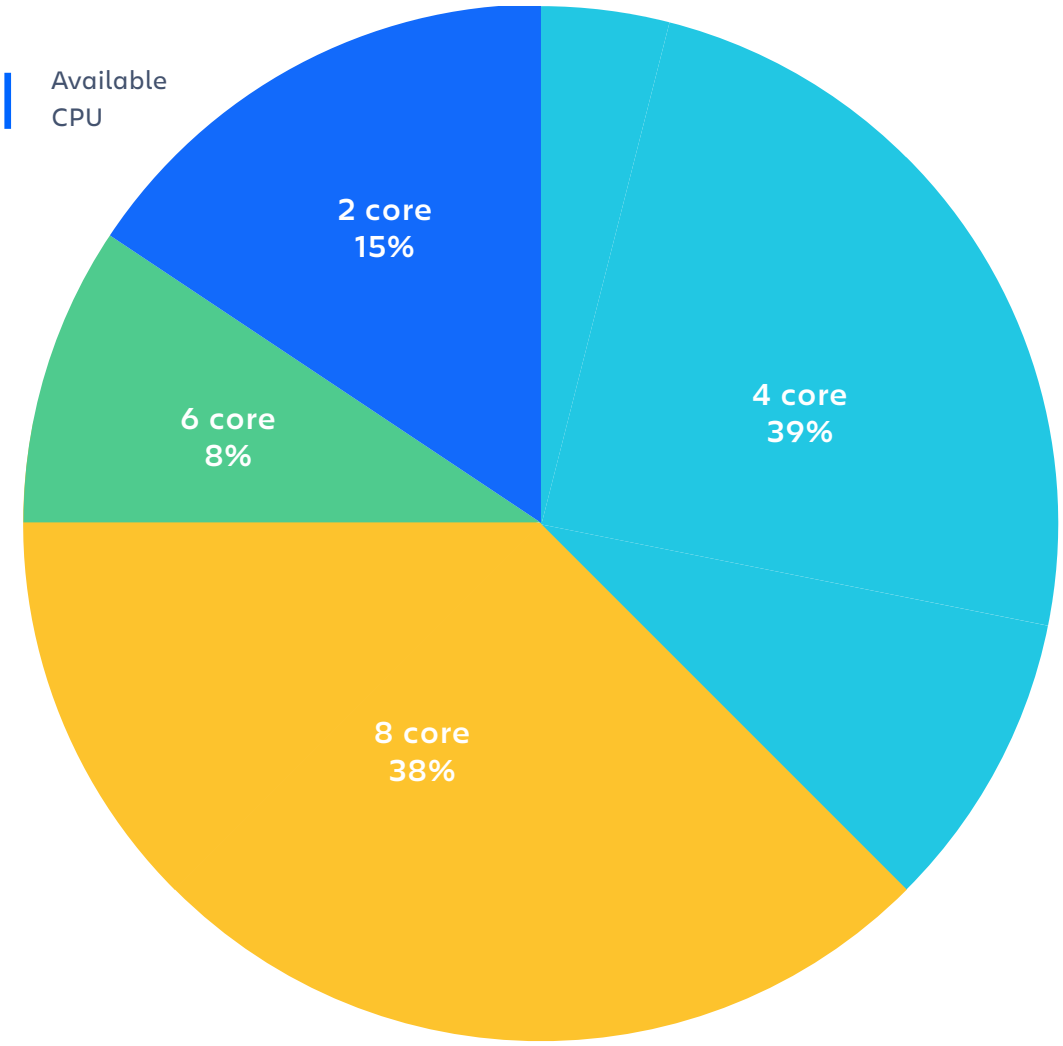
Application Nodes

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, Bitbucket, or Crowd). These nodes will be configured in a cluster, acting as one, serving the application to your users. For Bitbucket in particular, you'll need a separate node dedicated to Elastic Search.

Each node in your Data Center cluster must run the same version of the application and be located in the same physical location.

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.

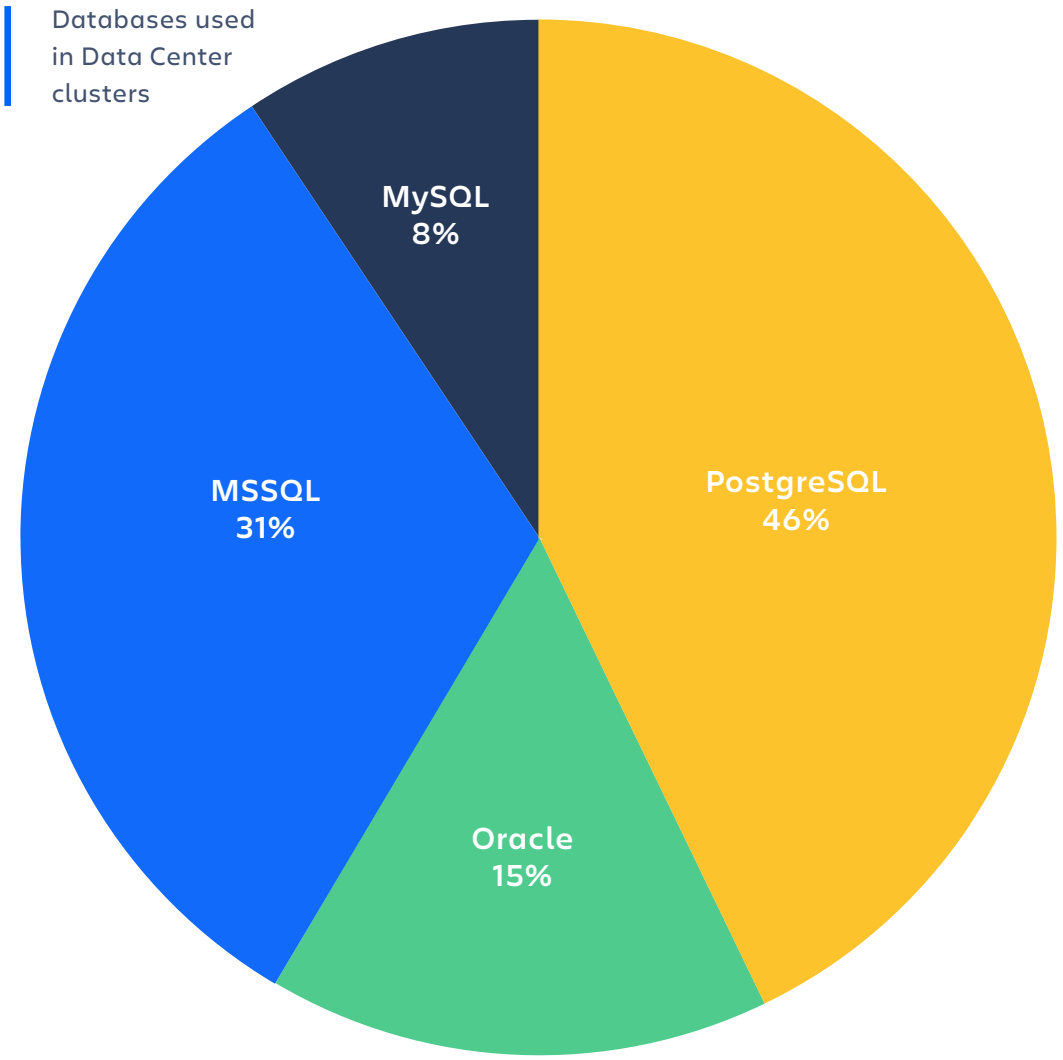
In a recent survey, we asked Data Center customers to share RAM and CPU on the nodes in their cluster. Here's what they reported:



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; having said this, 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.

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

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.

Infrastructure

We leave it up to you to choose which infrastructure to host your deployment on. Whether its 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 property.

CUSTOMER SNAPSHOT: Jira Software Data Center	CUSTOMER SNAPSHOT: Bitbucket Data Center	CUSTOMER SNAPSHOT: Confluence Data Center
Industry: Energy	**Industry: Software/ Technology**	**Industry: Software/ Technology**
Jira Software Instance: · Projects: 52 · Issues: 12,800 · Workflows: 57 · Attachments: 4,800	**Bitbucket Instance:** · Users: 2,000 · Repositories: 3,000 · Pull Requests/Day: 200-300 · Attachments: 4,800	**Confluence Instance:** · Pages: 2.1M · Spaces: 1,400 · Active Users: 43,000
Environment: Virtual (hosted)	**Environment:** Virtual (hosted)	**Environment:** Some combination of virtual (AWS, Azure, etc.) and physical servers
Nodes in the Cluster: 2	**Nodes in the Cluster:** 3	**Nodes in the Cluster:** 2
CPU on Each Server: 4 Core	**CPU on Each Server:** 4 Core	**CPU on Each Server:** 8 Core
RAM on Each Server: 8GB	**RAM on Each Server:** 48GB	**RAM on Each Server:** 48GB
Load Balancer: Apache	**Load Balancer:** HA Proxy	**Load Balancer:** VMWare
DR in Place: Yes	**DR in Place:** N/A – Atlassian Supported DR was not available for Bitbucket Data Center at the time of this survey.	**DR in Place:** Yes

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 guide 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 gets to know 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.

SECTION V

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.

Making 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 affect. 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. Pro actively 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 accommodate additional load or increase concurrent usage.

Scale: Grow Your Cluster

As discussed in Section IV, 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 has 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 nearly instantaneously.

> Several modern hosting technologies include auto scaling 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.

Constantly Inspect and Adapt

Monitoring and scaling are not processes that occur once during configuration without revisiting again. 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.

SECTION VI

Hear It From Our Customers

Splunk

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

Amadeus

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 & LIFECYCLE MANAGEMENT, AMADEUS

Mitchell International

When Mitchell International needed content collaboration software for a new platform initiative, they tried Confluence. Through word of mouth, interest in Confluence began sprouting up in other areas of the business with more and more teams adopting Confluence as their single source of truth. Confluence quickly became essential to the way their teams worked and they needed a way to scale to meet the growth, so they turned to Data Center.

In 2016, Mitchell kicked off the Enterprise Platform Initiative, which supports the development of web-based work flow solutions between insurance companies and collision repair facilities. The Enterprise Platform team were developing new solutions and knew they needed a centralized software to retain team knowledge and information. As content circulated beyond the Enterprise Platform team, other teams began experiencing the simplified site navigation and more effective search functionality of Confluence, as opposed to their existing solution which was described as “a place where documents go to die.”

After seeing so many other teams with interest in Confluence, the Product Manager of the Enterprise Platform team, Michela Baca, decided to lead an internal roadshow for various business units to promote the benefits of its use. Throughout the roadshow, awareness and interest in Confluence continued to grow, and the eventual need for high availability was becoming increasingly apparent. That’s when they decided to fully invest in Confluence Data Center to provide the always-on capability that their organization came to expect.

NOVOMATIC Group

The NOVOMATIC Group is comprised of many subsidiaries, and each subsidiary was operating and managing their Atlassian tools separately using different user management methods. This created a lack of visibility and standardization. As they grew and became even more decentralized, they needed a way to not only standardize the operations and tools used by their subsidiaries but a way to centralize user management for their Atlassian tools. They needed to ensure consistency and gain control over the operations and tools of their decentralized organization.

This journey began three years ago, led by NOVOMATIC team members Christian Wolf, Project and Application Manager, and Georg Aggermann, IT Service and Application Manager. Each subsidiary owned multiple instances of various Atlassian products, and each used a different solution to manage their users, ranging from different Active Directories, LDAP, and other alternatives. They needed a way to create a Group Wide Active Directory where they could completely consolidate user management. For their Atlassian solutions, they wanted to create a single user base across all of their subsidiaries along with a single sign-on experience for every Atlassian user. Adding an extra layer of complexity, many of these teams in each subsidiary need to work together. They don't work in isolation. For example, a few development teams in Austria at one subsidiary often work with a separate subsidiary's development team located in Poland. The NOVOMATIC Group needed to ensure they could give the appropriate level of access to each team, protecting data considered confidential to each subsidiary, without hindering the ability to collaborate.

The NOVOMATIC Group determined that Crowd would be the simplest way to centralize user management for all of their subsidiaries' Atlassian products. They worked with their Atlassian Solution Partner, Celix, on setting up one central Crowd instance along with three Crowd instances for their subsidiaries. These Crowd instances connected to several Jira Software, Confluence, Bitbucket, Fisheye and Crucible instances. Across everything, they established one single user base that synced with each of these four Crowd instances, giving them a single source of truth.

After completing the project of consolidating user management and balancing the right level of access each user should have, they began to realize how critical Crowd was to their users' day to day operations. Since all of their development is done using Atlassian, these tools are critical to their business. Downtime of a Crowd instance meant that their users could be unable to access some or all of Atlassian tools connected to that Crowd instance, which completely stops work. In order to solve for this, The NOVOMATIC Group engaged their Technical Account Manager (TAM) to investigate how they could bring high availability to Crowd.

With the help of their TAM, NOVOMATIC became one of the private beta customers for Crowd Data Center in May 2017. Their TAM also helped NOVOMATIC evaluate prerequisites and ensure that NOVOMATIC was able to meet these prior to beginning the beta program. In less than two months, NOVOMATIC set up and tested Crowd Data Center initially in a staging environment. The Crowd team and their TAM had multiple check-in's throughout the process to ensure

NOVOMATIC's success. After completing installation and finding no issues during their testing, they decided to roll the Crowd Data Center beta version into production. Since rolling out the beta version into their test environment and then into product, they have not experienced any downtime. "With Crowd Data Center, we no longer have to worry about network outages or any impact to our development teams," says Aggermann. In addition to reducing the risk of an outage, upgrades and changes to the system could take place within a full downtime for the users.

> **"With Crowd Data Center, we no longer have to worry about network outages or any impact to our development teams.**
>
> **GEORG AGGERMANN, IT SERVICE & APPLICATION MANAGER**

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

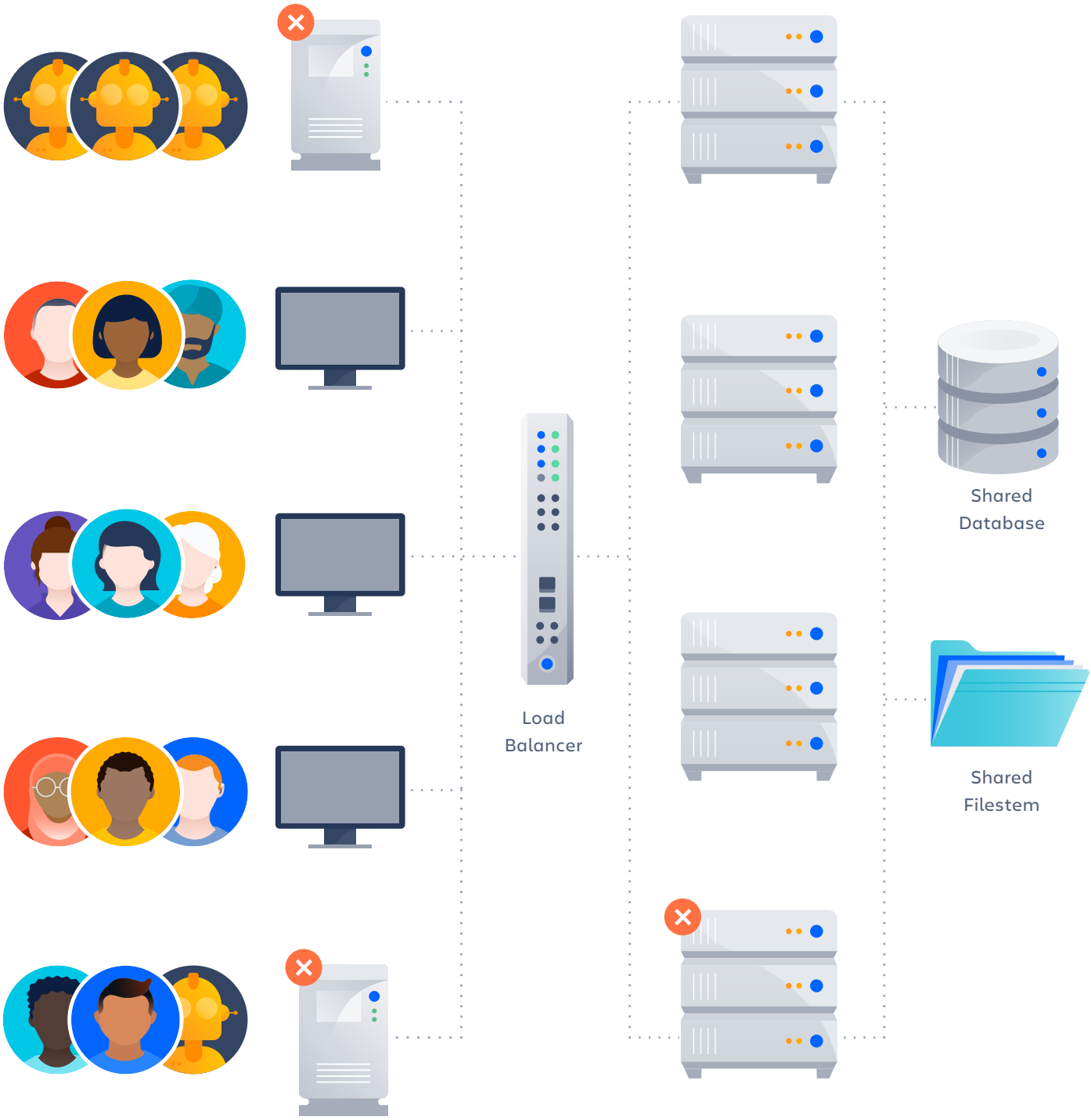
Cerner had three federated instances of JIRA Software Server with thousands of developers using each instance every hour of every day around the globe. JIRA Software quickly became mission-critical and every minute of downtime or performance degradation made it more difficult for Cerner team members 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.

"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.

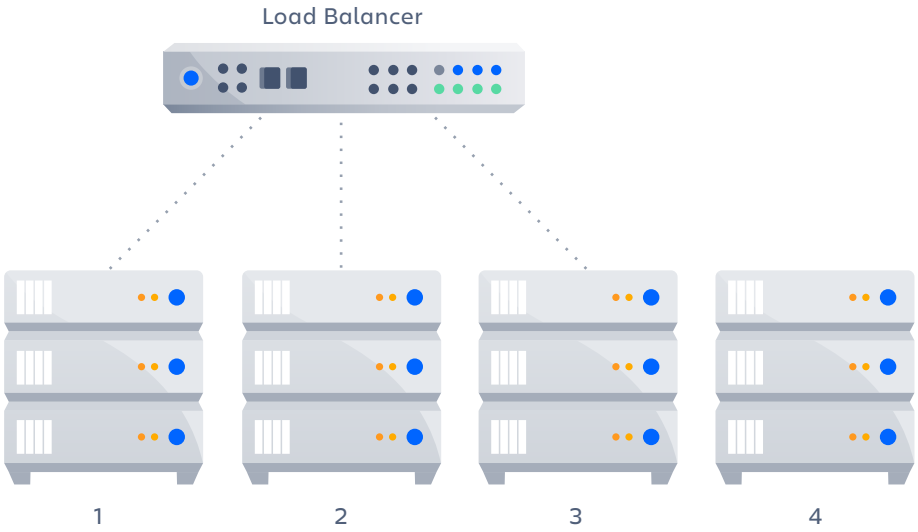


The impact and rate of growth for external integrations (robot) differ from human interactions and can add stress to individual nodes in the cluster, bringing down everyone else using that node.

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



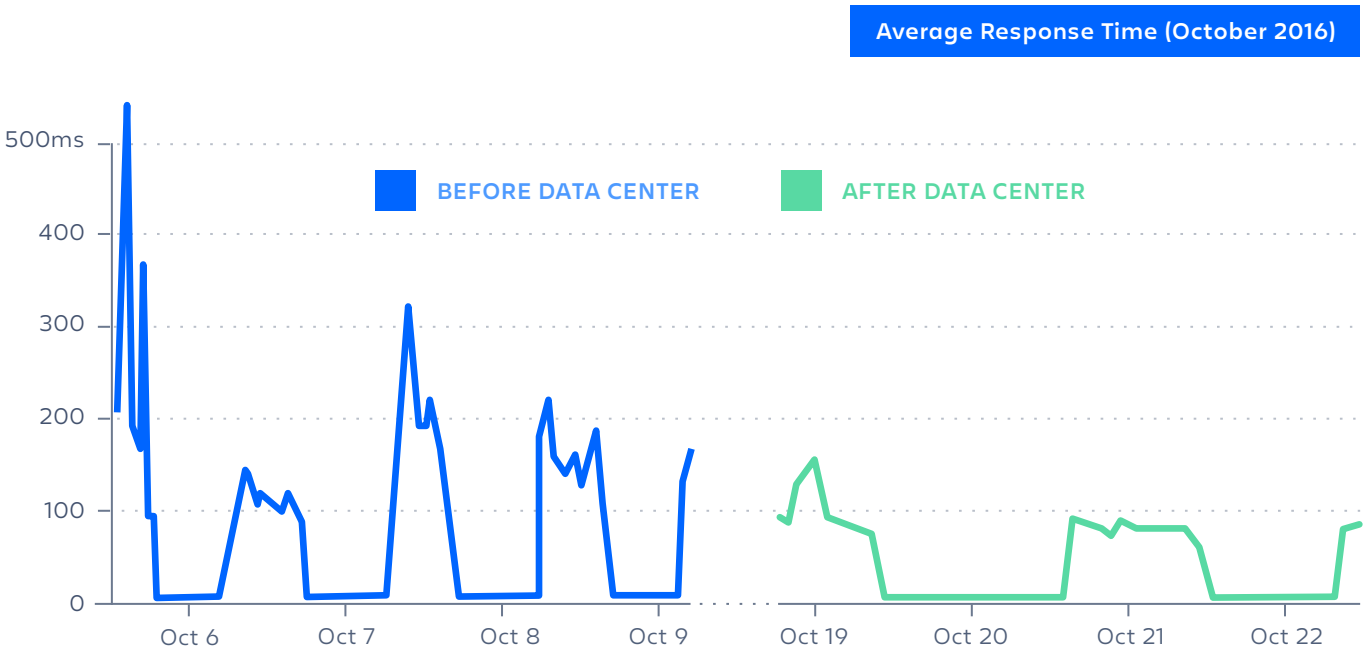
Their TAM originally thought the best option would be to use the load balancer to route all requests with '/rest' to the REST API node. However, after some testing they found the REST API was also being used throughout JIRA Software, including the login page, so leaving it to '/rest' would mean they'd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

Sources and Resources

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

Broda, 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.

Gessow, Danielle. “Jira Software Server 7.5: introducing 5 new features for flexibility at scale.” Atlassian Blog, https://www.atlassian.com/blog/jira-software/jira-software-server-7-5.

Keough, Brian. “Mitchell International’s Grassroots Path to Confluence Data Center.” Atlassian Blog, https://www.atlassian.com/blog/confluence/mitchell-internationals-grassroots-path-to-confluence-data-center.

King, Ben. “Performance at Scale: 10 million issues and beyond with Jira Data Center.” Atlassian White Paper, https://www.atlassian.com/whitepapers/performance-at-scale.

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

Roney, Maggie. “Four Reasons it’s time to try Data Center.” Atlassian Blog, https://www.atlassian.com/blog/jira-software/4-reasons-time-try-data-center.

Roney, Maggie. “How the NOVOMATIC Group Manages their Atlassian Users at scale using Crowd Data Center.” Atlassian Blog, https://www.atlassian.com/blog/crowd/novomatic-user-management-at-scale-with-crowd-data-center.

Rusonis, Shana. “Your Atlassian products and IaaS: What you should know.” Atlassian Blog, https://www.atlassian.com/blog/enterprise/atlassian-data-center-iaas.

Rusonis, Shana. “Introducing Project Archiving for Jira Software Data Center.” Atlassian Blog, https://www.atlassian.com/blog/jira-software/project-archiving-jira-software-enterprise.

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.

ATLASSIAN