



Post-Incident Review

April 2022 Outage

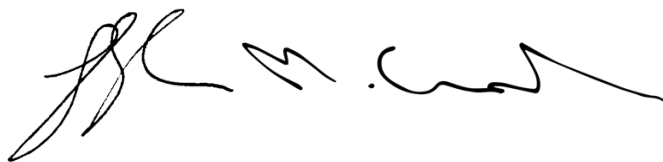
Letter from our co-founders & co-CEOs

We want to acknowledge the outage that disrupted service for customers earlier this month. We understand that our products are mission critical to your business, and we don't take that responsibility lightly. The buck stops with us. Full stop. For those customers affected, we are working to regain your trust.

i At Atlassian, one of our core values is “Open company, no bullshit”. We bring this value to life in part by openly discussing incidents and using them as opportunities to learn. We are publishing this Post-Incident Review for our customers, our Atlassian community, and the broader technical community. Atlassian is proud of our [incident management process](#) which emphasizes that a blameless culture and a focus on identifying ways to improve our technical systems and processes are critical to providing high-scale, trustworthy services. While we do our best to avoid any type of incident, we also embrace the idea that incidents are a powerful way to improve.

Rest assured, Atlassian's cloud platform allows us to meet the diverse needs of our over 200,000 cloud customers of every size and across every industry. Prior to this incident, our cloud has consistently delivered 99.9% uptime and exceeded uptime SLAs. We've made long-term investments in our platform and in a number of centralized platform capabilities, with a scalable infrastructure and a steady cadence of security enhancements.

To our customers and our partners, we thank you for your continued trust and partnership. We hope the details and actions outlined in this document show that Atlassian will continue to provide a world-class cloud platform and a powerful portfolio of products to meet the needs of every team.

Handwritten signatures of Scott and Mike, the co-founders and co-CEOs of Atlassian.

-Scott and Mike

Executive summary

On Tuesday, April 5th, 2022, starting at 7:38 UTC, 775 Atlassian customers lost access to their Atlassian products. The outage spanned up to 14 days for a subset of these customers, with the first set of customers being restored on April 8th and all customer sites progressively restored by April 18th.

This was not a result of a cyberattack and there was no unauthorized access to customer data. Atlassian has a comprehensive [data management](#) program with published SLAs and a history of exceeding those SLAs.

Although this was a major incident, no customer lost more than five minutes of data. In addition, over 99.6% of our customers and users continued to use our cloud products without any disruption during the restoration activities.



Throughout this document, we refer to those customers whose sites were deleted as part of this incident as “affected” or “impacted” customers. This PIR provides the exact details of the incident, outlines the steps we took to recover, and describes how we will prevent situations like this from happening in the future. We provide a high-level summary of the incident in this section, with further detail in the remainder of the document.

What happened?

In 2021, we completed the acquisition and integration of a standalone Atlassian app for Jira Service Management and Jira Software called "Insight – Asset Management". The functionality of this standalone app was then native within Jira Service Management and no longer available for Jira Software. Because of this, we needed to delete the standalone legacy app on customer sites that had it installed. Our engineering teams used an existing script and process to delete instances of this standalone application, but there were two problems:

- **Communication gap.** There was a communication gap between the team that requested the deletion and the team that ran the deletion. Instead of providing the

IDs of the *intended app* being marked for deletion, the team provided the IDs of the *entire cloud site* where the apps were to be deleted.

- **Insufficient system warnings.** The API used to perform the deletion accepted both site and app identifiers and assumed the input was correct - this meant that if a site ID is passed, a site would be deleted; if an app ID was passed, an app would be deleted. There was no warning signal to confirm the type of deletion (site or app) being requested.

The script that was executed followed our standard peer-review process, which focused on which endpoint was being called and how. It did not cross-check the provided cloud site IDs to validate whether they referred to the Insight App or to the entire site, and the problem was that the script contained the ID for a customer's entire site. The result was an immediate deletion of 883 sites (representing 775 customers) between 07:38 UTC and 08:01 UTC on Tuesday, April 5th, 2022. See "*What happened*"

How did we respond?

Once the incident was confirmed on April 5th at 08:17 UTC, we triggered our major incident management process and formed a cross-functional incident management team. The global incident response team worked 24/7 for the duration of the incident until all sites were restored, validated, and returned to customers. In addition, incident management leaders met every three hours to coordinate the workstreams.

Early on, we realized that a number of challenges restoring hundreds of customers with multiple products simultaneously.

At the start of the incident, we knew exactly which sites were affected and our priority was to establish communication with the approved owner for each impacted site to inform them of the outage.

However, some customer contact information was deleted. This meant that customers could not file support tickets as they normally would. This also meant we did not have immediate access to key customer contacts. For more details, see "*High-level overview of recovery workstreams*"

What are we doing to prevent situations like this in the future?

We have taken a number of immediate actions and are committed to making changes to avoid this situation in the future. Here are four specific areas where we have made or will make significant changes:

1. **Establish universal “soft deletes” across all systems.** Overall, a deletion of this type should be prohibited or have multiple layers of protections to avoid errors, including staged rollout and tested rollback plan for “soft deletes”. We will globally prevent the deletion of customer data and metadata that has not gone through a soft-delete process.
2. **Accelerate in our Disaster Recovery (DR) program to automate restoration for the multi-site, multi-product deletion events for a larger set of customers.** We will leverage the automation and learnings from this incident to accelerate the DR program to meet the recovery time objective (RTO) as defined in our policy for this scale of incident. We will regularly run DR exercises that involve restoring all products for a large set of sites.
3. **Revise incident management process for large-scale incidents.** We will improve our standard operating procedure for large-scale incidents and practice it with simulations of this scale of incident. We will update our training and tooling to handle the large number of teams working in parallel.
4. **Create large-scale incident communications playbook.** We will acknowledge incidents early, through multiple channels. We will release public communications on incidents within hours. To better reach impacted customers, we will improve the backup of key contacts and retrofit support tooling to enable customers without a valid URL or Atlassian ID to make direct contact with our technical support team.

Our full list of action items is detailed in the full post-incident review below. See “*How will we improve*”

Table of contents

| | |
|---|---------|
| Overview of Atlassian's cloud architecture | Page 7 |
| <ul style="list-style-type: none">• Atlassian's cloud hosting architecture• Distributed services architecture• Multi-tenant architecture• Tenant provisioning and lifecycle• Disaster Recovery program<ul style="list-style-type: none">○ Resiliency○ Service storage restorability○ Multi-site, multi-product automated restorability | |
| What happened, timeline, and recovery | Page 13 |
| <ul style="list-style-type: none">• What happened• How we coordinated• Timeline of the incident• High-level overview of recovery workstreams<ul style="list-style-type: none">○ Workstream 1: Detection, starting the recovery & identifying our approach○ Workstream 2: Early recovery and the Restoration 1 approach○ Workstream 3: Accelerated recovery and the Restoration 2 approach○ Minimal data loss following the restoration of deleted sites | |
| Incident communications | Page 21 |
| <ul style="list-style-type: none">• What happened | |
| Support experience & customer outreach | Page 23 |
| <ul style="list-style-type: none">• How was support for our customers impacted?• How did we respond? | |
| How will we improve? | Page 25 |
| <ul style="list-style-type: none">• Learning 1: "Soft deletes" should be universal across all systems• Learning 2: As part of the DR program, automate restoration for multi-site, multi-product deletion events for a larger set of customers• Learning 3: Improve incident management process for large scale events• Learning 4: Improve our communications processes | |
| Closing remarks | Page 31 |

Overview of Atlassian's cloud architecture

To understand contributing factors to the incident as discussed throughout this document, it is helpful to first understand the deployment architecture for Atlassian's products, services, and infrastructure.

Atlassian's cloud hosting architecture

Atlassian uses Amazon Web Services (AWS) as a cloud service provider and its highly available data center facilities in [multiple regions worldwide](#). Each AWS region is a separate geographical location with multiple, isolated, and physically-separated groups of data centers known as Availability Zones (AZs).

We leverage AWS' compute, storage, network, and data services to build our products and platform components, which enables us to utilize redundancy capabilities offered by AWS, such as availability zones and regions.

Distributed services architecture

With this AWS architecture, we host a number of platform and product services that are used across our solutions. This includes platform capabilities that are shared and consumed across multiple Atlassian products, such as Media, Identity, Commerce, experiences like our Editor, as well as product-specific capabilities, like Jira Issue service and Confluence Analytics.

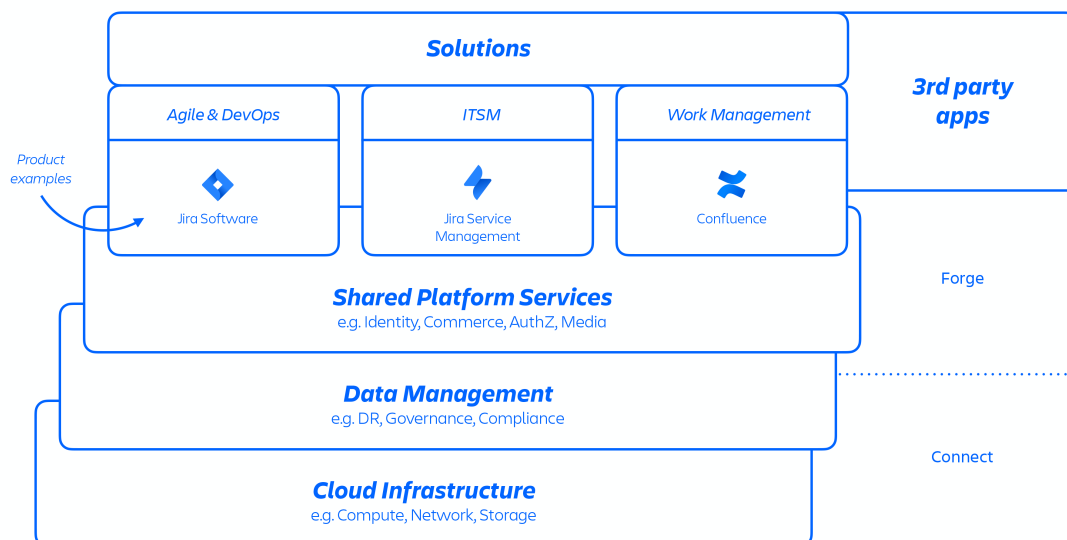


Figure 1: Atlassian platform architecture.

Atlassian developers provision these services through an internally developed platform-as-a-service (PaaS), called Micros, which automatically orchestrates the deployment of shared services, infrastructure, data stores, and their management capabilities, including security and compliance control requirements (see *Figure 1* above). Typically, an Atlassian product consists of multiple “containerized” services that are deployed on AWS using Micros. Atlassian products use core platform capabilities (see *Figure 2* below) that range from request routing to binary object stores, authentication/authorization, transactional user-generated content (UGC) and entity relationships stores, data lakes, common logging, request tracing, observability, and analytical services. These micro-services are built using approved technical stacks standardized at the platform level:

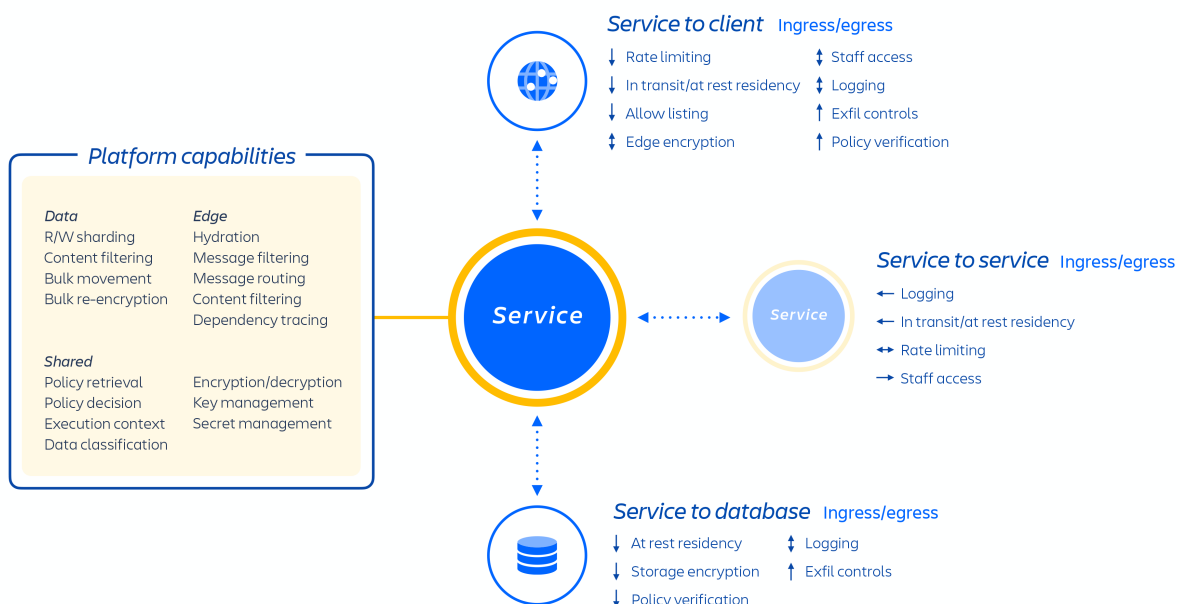


Figure 2: Overview of Atlassian micro-services.

Multi-tenant architecture

On top of our cloud infrastructure, we built and operate a multi-tenant micro-service architecture along with a shared platform that supports our products. In a multi-tenant architecture, a single service serves multiple customers, including databases and compute instances required to run our cloud products. Each shard (essentially a container - see *Figure 3* below) contains the data for multiple tenants, but each tenant’s data is isolated and inaccessible to other tenants.

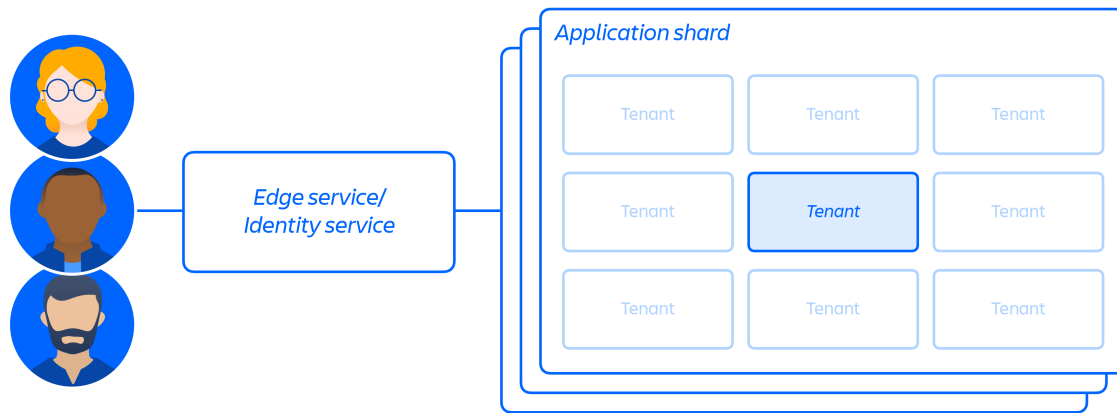


Figure 3: How we store data in a multi-tenant architecture.

Tenant provisioning and lifecycle

When a new customer is provisioned, a series of events trigger the orchestration of distributed services and provisioning of data stores. These events can be generally mapped to one of seven steps in the lifecycle:

- 1 Commerce systems are immediately updated with the latest metadata and access control information for that customer, and then a provisioning orchestration system aligns the “state of the provisioned resources” with the license state through a series of tenant and product events.

Tenant events

These events affect the tenant as a whole and can either be:

- Creation: a tenant is created and used for brand new sites
- Destruction: an entire tenant is deleted

Product events

- Activation: after the activation of licensed products or third-party apps
- Deactivation: after the de-activation of certain products or apps
- Suspension: after the suspension of a given existing product, thus disabling access to a given site that they own
- Un-suspension: after the un-suspension of a given existing product, thus enabling access to a site that they own

License update: contains information regarding the number of license seats for a given product as well as its status (active/inactive)

- 2 Creation of the customer site and activation of the correct set of products for the customer. The concept of a site is the container of multiple products licensed to a particular customer. (e.g. Confluence and Jira Software for `<site-name>.atlassian.net`). This (see *Figure 4* below) is an important point to understand in the context of this report, as the site container is what was deleted in this incident and the concept of a site is discussed throughout the document.

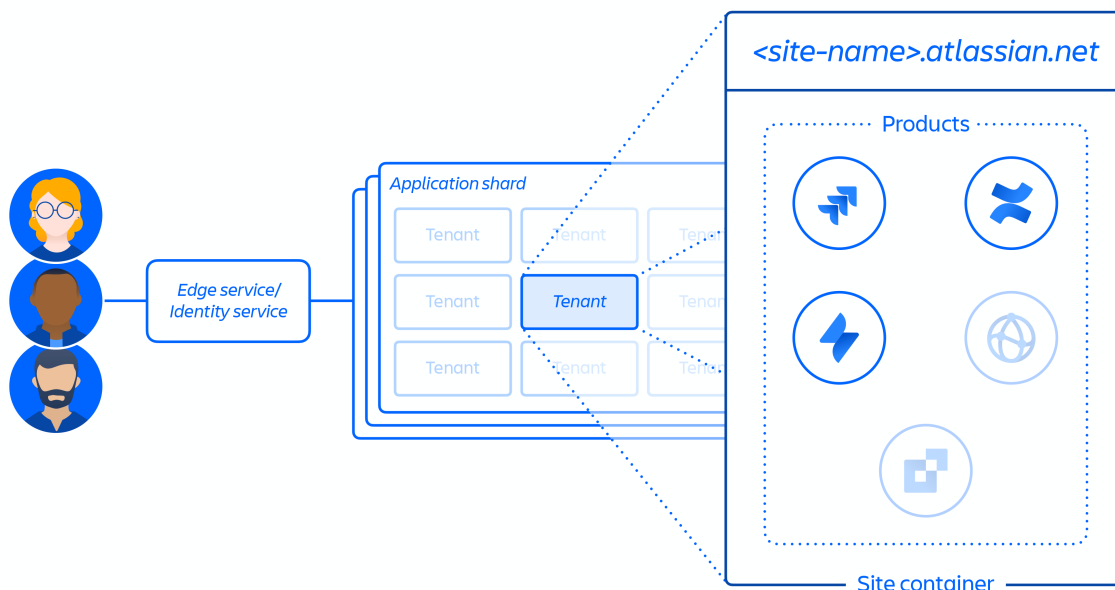


Figure 4: Overview of the site container.

- 3 Provisioning of products within the customer site in the designated region.

When a product is provisioned it will have the majority of its content hosted close to where users are accessing it. To optimize product performance, we don't limit data movement when it's hosted globally and we may move data between regions as needed.

For some of our products, we also offer data residency. Data residency allows customers to choose whether product data is globally distributed or held in place in one of our defined geographic locations.

- 4 Creation and storage of the customer site and product(s) core metadata and configuration.

- 5 Creation and storage of the site and product(s) identity data, such as users, groups, permissions, etc.
- 6 Provisioning of product databases within a site, e.g. Jira family of products, Confluence, Compass, Atlas.
- 7 Provisioning of the product(s) licensed apps.

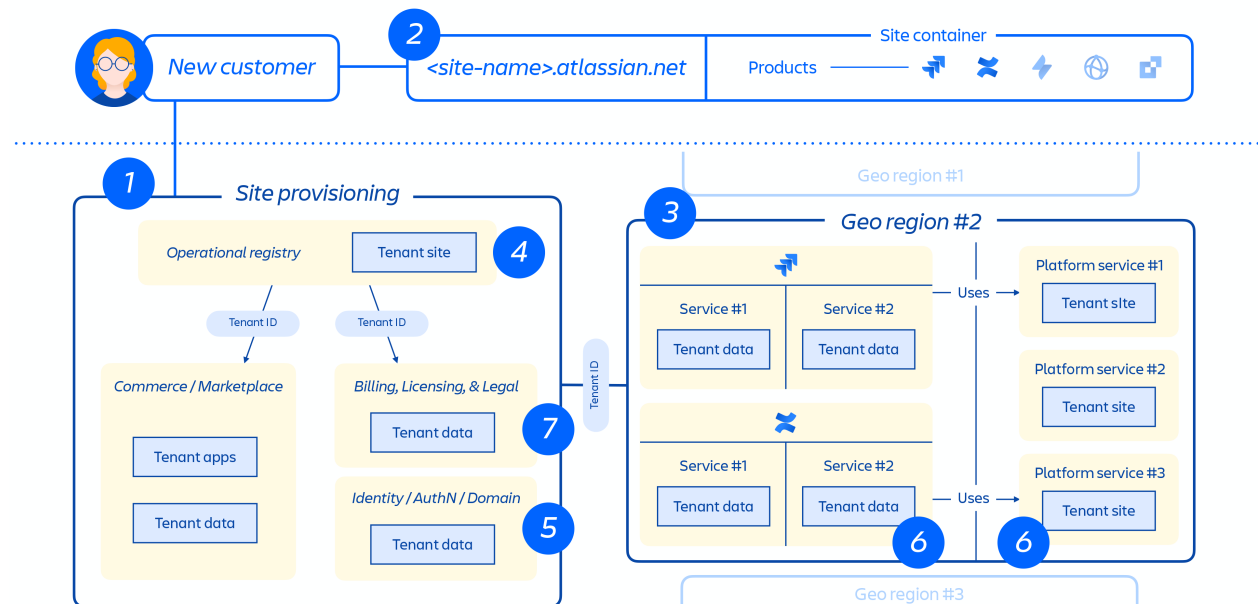


Figure 5: Overview of how customer site is provisioned across our distributed architecture.

Figure 5 above demonstrates how a customer's site is deployed across our distributed architecture, not just in a single database or store. This includes multiple physical and logical locations that store meta-data, configuration data, product data, platform data and other related site info.

Disaster Recovery program

Our [Disaster Recovery](#) (DR) program encompasses all of our efforts to provide resiliency against infrastructure failures and restorability of service storage from backups. Two important concepts to understand disaster recovery programs are:

- **Recovery time objective (RTO):** How quickly can the data be recovered and returned to a customer during a disaster?
- **Recovery point objective (RPO):** How fresh is the recovered data after it is recovered from a backup? How much data will be lost since the last backup?

During this incident, we missed our RTO but met our RPO.

Resiliency

We prepare for infrastructure-level failures; for example, the loss of an entire database, service, or AWS Availability Zones. This preparation includes replication of data and services across multiple availability zones and regular failover testing.

Service storage restorability

We also prepare to recover from data corruption of service storage due to risks such as ransomware, bad actors, software defects, and operational errors. This preparation includes immutable backups and service storage backup restoration testing. We are able to take any individual data store and restore it to a previous point in time.

Multi-site, multi-product automated restorability

At the time of the incident, we did not have the ability to select a large set of customer sites and restore all of their inter-connected products from backups to a previous point in time.

Our capabilities have been focused on infrastructure, data corruption, single service events, or single-site deletions. In the past, we have had to deal with and test these kinds of failures. The site-level deletion did not have runbooks that could be quickly automated for the scale of this event which required tooling and automation across all the products and services to happen in a coordinated way.

The following sections will go into more depth about this complexity and what we are doing at Atlassian to evolve and optimize our abilities to maintain this architecture at scale.

What happened, timeline, and recovery

What happened

In 2021, we completed the integration of a standalone Atlassian app for Jira Service Management and Jira Software, called "Insight – Asset Management". The functionality of this standalone app was then native within Jira Service Management and was no longer available for Jira Software. Because of this, we needed to delete the standalone legacy app on customer sites that had it installed. Our engineering teams used an existing script and process to delete instances of this standalone application.

However, two critical problems ensued:

- **Communication gap.** There was a communication gap between the team that requested the deletion and the team that ran it. Instead of providing the IDs of the intended app being marked for deletion, the team provided the IDs of the entire cloud site where the apps were to be deleted.
- **Insufficient system warnings.** The API used to perform the deletion accepts both site and app identifiers and assumes the input is correct - this means that if a site ID is passed, a site will be deleted; if an app ID is passed, an app will be deleted. There was no warning signal to confirm the type of deletion (site or app) being requested.

The script that was executed followed our standard peer-review process, which focused on which endpoint was being called and how. It did not cross-check the provided cloud site IDs to validate whether they referred to the app or to the entire site. The script was tested in Staging per our standard change management processes, however, it would not have detected that the IDs input were incorrect as the IDs did not exist in the Staging environment.

When run in Production, the script initially ran against 30 sites. The first Production run was successful, and deleted the Insight app for those 30 sites with no other side effects. However, IDs for those 30 sites were sourced prior to the miscommunication event and included the correct Insight app IDs.

The script for the subsequent Production run included site IDs in place of Insight app IDs and executed against a set of 883 sites. The script started running on April 5th at 07:38

UTC and was completed at 08:01 UTC. The script deleted sites sequentially based on the input list, so the first customer's site was deleted shortly after the script started running at 07:38 UTC. The result was an immediate deletion of the 883 sites, with no warning signal to our engineering teams.

The following Atlassian products were unavailable for impacted customers: Jira family of products, Confluence, Atlassian Access, Opsgenie, and Statuspage.

As soon as we learned of the incident, our teams were focused on restoration for all impacted customers. At that time, we estimated the number of impacted sites to be ~700 (883 total sites were impacted, but we subtracted out the Atlassian-owned sites). Of the 700, a significant portion were inactive, free, or small accounts with a low number of active users. Based on this, we initially estimated the approximate number of impacted customers at around 400.

We now have a much more accurate view, and for complete transparency based on Atlassian's official customer definition, 775 customers were affected by the outage. However, the majority of users were represented within the original 400 customer estimate. The outage spanned up to 14 days for a subset of these customers, with the first set of customers being restored on April 8th, and all customers restored as of April 18th.

How we coordinated

The first support ticket was created by an impacted customer at 07:46 UTC on April 5th. Our internal monitoring did not detect an issue because the sites were deleted via a standard workflow. At 08:17 UTC, we triggered our major incident management process, forming a cross-functional incident management team, and in seven minutes, at 08:24 UTC, it had been escalated to Critical. At 08:53 UTC, our team confirmed that the customer support ticket and the script run were related. Once we realized the complexity of restoration, we assigned our highest level of severity to the incident at 12:38 UTC.

The incident management team was composed of individuals from multiple teams across Atlassian, including engineering, customer support, program management, communications, and many more. The core team met every three hours for the duration of the incident until all sites were restored, validated, and returned to customers.

To manage the restoration progress we created a new Jira project, SITE, and a workflow to track restorations on a site-by-site basis across multiple teams (engineering, program management, support, etc). This approach empowered all teams to easily identify and track issues related to any individual site restoration.

We also implemented a code freeze across all of engineering for the duration of the incident on April 8th at 03:30 UTC. This allowed us to focus on customer restoration, eliminate the risk of change causing inconsistencies in customer data, minimize the risk of other outages, and reduce the likelihood of unrelated changes distracting the team from recovery.

Timeline of the incident

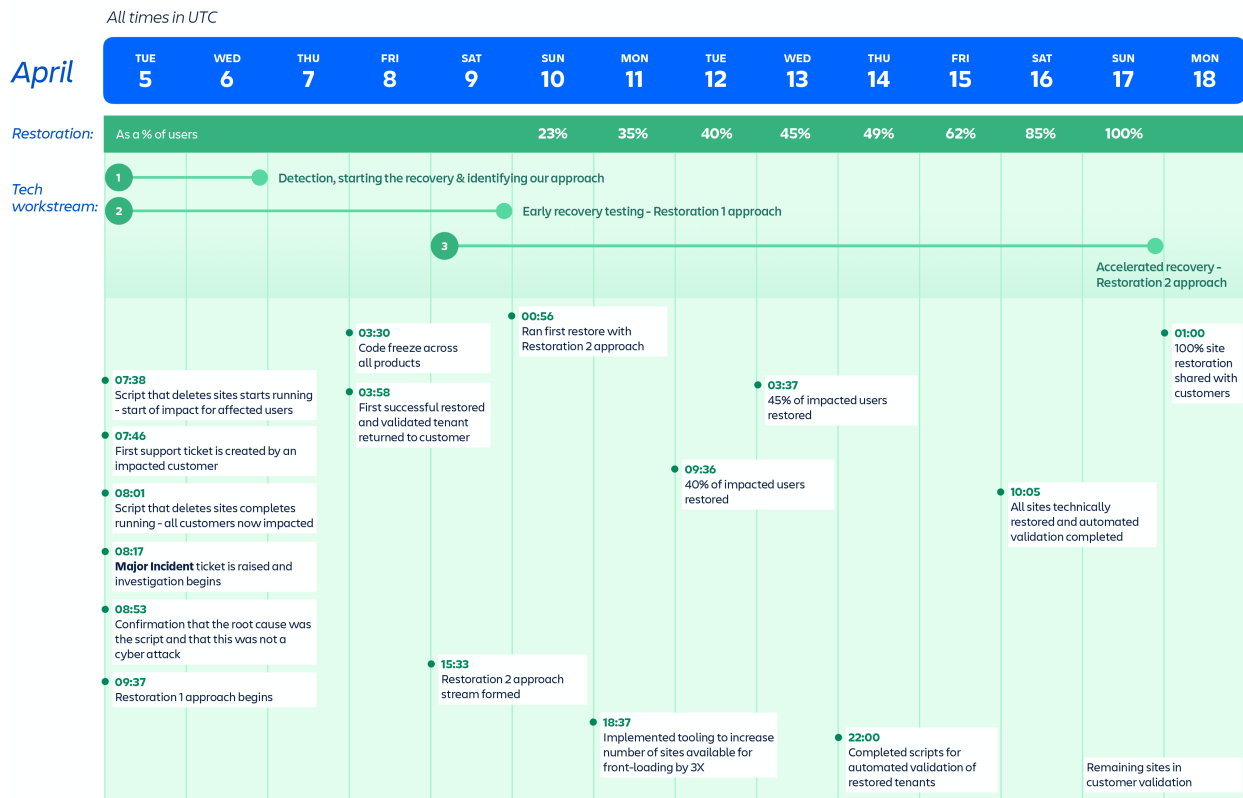


Figure 6: Timeline of the incident and key restoration milestones.

High-level overview of recovery workstreams

Recovery ran as three primary workstreams - detection, early recovery, and acceleration. While we've described each workstream separately below, during recovery there was work happening in parallel across all of the workstreams.

Workstream 1: Detection, starting the recovery & identifying our approach

Timestamp: Days 1-2 (April 5th - 6th)

At 08:53 UTC on April 5th, we identified that the Insight app script caused the deletion of sites. We confirmed that this was not the result of an internal malicious act or cyberattack. Relevant product and platform infrastructure teams were paged and brought into the incident.

At the beginning of the incident, we recognized:

- Restoring hundreds of deleted sites is a complex, multi-step process (detailed in the architecture section above), requiring many teams and multiple days to successfully complete.
- We had the ability to recover a single site, but we had not built capabilities and processes for recovering a large batch of sites.

As a result, we needed to substantially parallelize and automate the restoration process in order to help impacted customers regain access to their Atlassian products as quickly as possible.

Workstream 1 involved large numbers of development teams engaging in the following activities:

- Identifying and executing restoration steps for batches of sites in the pipeline.
- Writing and improving automation to allow the team(s) to execute restoration steps for larger numbers of sites in a batch.

Workstream 2: Early recovery and the Restoration 1 approach

Timestamp: Days 1-4 (April 5th - 9th)

We understood what caused the site deletion on April 5th at 08:53 UTC, within an hour after the script finished its run. We also identified the restoration process that had

previously been used to recover a small number of sites into production. However, the recovery process for restoring deleted sites at such a scale wasn't well defined.

To get moving quickly, the early stages of the incident split into two working groups:

- The manual working group validated the steps required and manually executed the restoration process for a small number of sites.
- The automation working group took the existing restoration process and built automation to safely execute the steps across larger batches of sites.

Overview of the Restoration 1 approach (see *Figure 7* below):

- It required the creation of a new site for each deleted one, followed by every downstream product, service, and data store needing to restore their data.
- The new site would come with new identifiers such as `cloudId`. These identifiers are all considered immutable, meaning that many systems embed these identifiers in data records. As a result, we needed to update large quantities of data if these identifiers changed, which is particularly problematic for third-party ecosystem apps.
- Modifying a new site to replicate the state of the deleted site had complex and often unforeseen dependencies between steps.

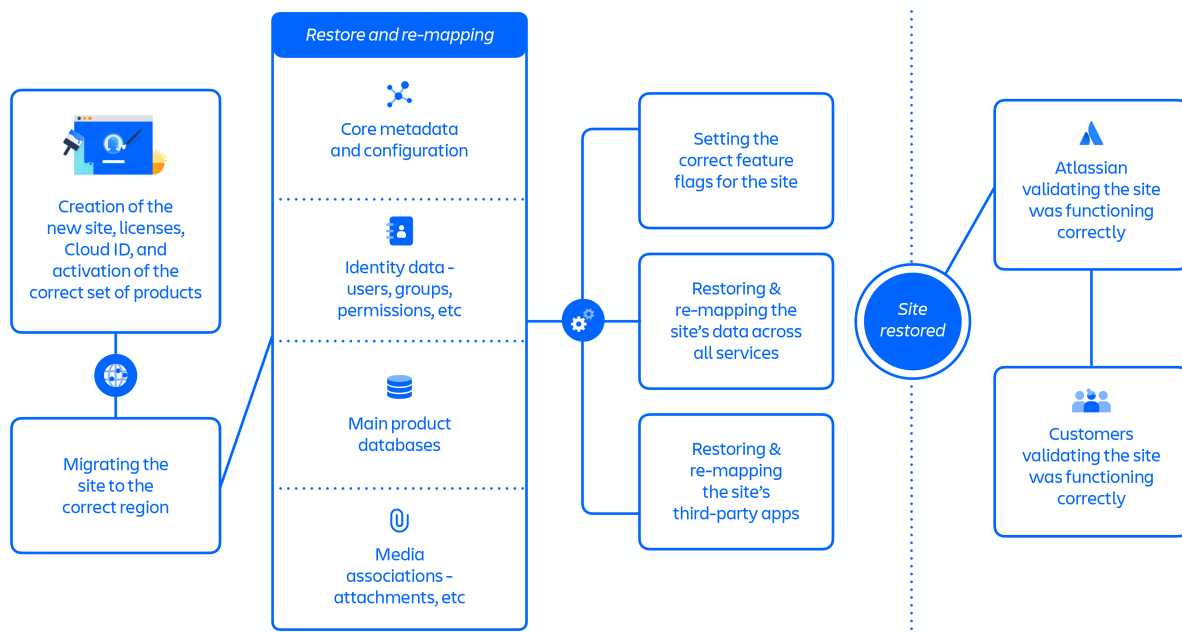


Figure 7: Key steps in the Restoration 1 approach.

The Restoration 1 approach included approximately 70 individual steps that, when aggregated at a high level, followed a largely sequential flow of:

- Creation of the new site, licenses, Cloud ID, and activation of the correct set of products
- Migrating the site to the correct region
- Restoring & re-mapping the site's core metadata and configuration
- Restoring & re-mapping the site's Identity data - users, groups, permissions, etc
- Restoring the site's main product databases
- Restoring & re-mapping the site's media associations - attachments, etc
- Setting the correct feature flags for the site
- Restoring & re-mapping the site's data across all services
- Restoring & re-mapping the site's third-party apps
- Atlassian validating the site was functioning correctly
- Customers validating the site was functioning correctly

Once optimized, the Restoration 1 approach took approximately 48 hours to restore a batch of sites, and was used for the recovery of 53% of impacted users across 112 sites between April 5th and April 14th.

Workstream 3: Accelerated recovery and the Restoration 2 approach

Timestamp: Days 4 - 13 (April 9th - 17th)

With the Restoration 1 approach, it would have taken us three weeks to restore all customers. Therefore, we proposed a new approach on April 9th to speed up the restoration of all sites, Restoration 2 (see *Figure 8* below).

The Restoration 2 approach offered improved parallelism between restoration steps by reducing complexity and the number of dependencies that were present with the Restoration 1 approach.

Restoration 2 involved the re-creation (or un-deletion) of records associated with the site across all respective systems, beginning with the Catalogue Service record. A key element of this new approach was to *re-use all of the old site identifiers*. This removed over half of the steps from the prior process that were used to map the old identifiers to the new identifiers, including the need to coordinate with every third-party app vendor for each site.

However, the move from the Restoration 1 to the Restoration 2 approach added substantial overhead in the incident response:

- Many of the automation scripts and processes established in the Restoration 1 approach had to be modified for Restoration 2.
- Teams performing restorations (including incident coordinators) had to manage parallel batches of restorations in both approaches, while we tested and validated the Restoration 2 process.
- Using a new approach meant that we needed to test and validate the Restoration 2 process before scaling it up, which required duplicating validation work that was previously completed for Restoration 1.

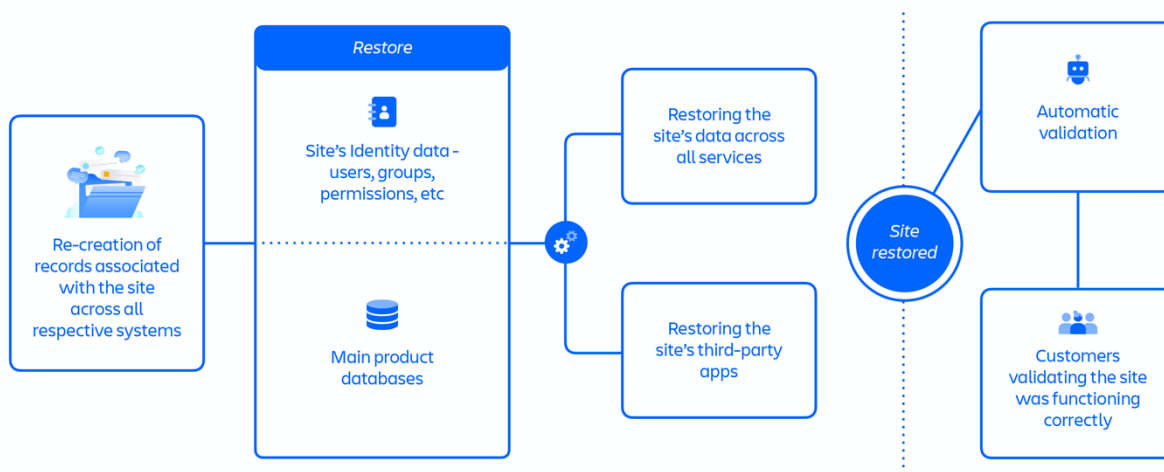


Figure 8: Key steps in the Restoration 2 approach.

The graphic above represents the Restoration 2 approach, which included over 30 steps that followed a largely parallelized flow of:

- Re-creation of records associated with the site across all respective systems
- Restoring the site's Identity data - users, groups, permissions, etc
- Restoring the site's main product databases
- Restoring the site's data across all services
- Restoring the site's third-party apps
- Automatic validation
- Customers validating the site was functioning correctly

As part of the accelerated recovery, we also took steps to front-load and automate site restoration because manual restoration would not scale well for large batches. The sequential nature of the recovery process meant site restoration could be slower for

large database restorations and user base/permissions restorations. Optimizations we implemented included:

- We developed the tooling and guard rails needed to *front-load* and long-running steps like database restorations and Identity synchronizations so that they were completed in advance of other restoration steps.
- Engineering teams built automation for their individual steps that enabled large batches of restorations to be safely executed.
- Automation was built to validate sites were functioning correctly after all restoration steps were completed.

The accelerated Restoration 2 approach took approximately 12 hours to restore a site and was used for the recovery of approximately 47% of impacted users across 771 sites between April 14th and 17th.

Minimal data loss following the restoration of deleted sites

Our databases are backed up using a combination of full backups and incremental backups that allow us to choose any particular “Point in Time” to recover our data stores within the backup retention period (30 days). For most customers during this incident, we identified the main data stores for our products and decided on using a restore point of five minutes prior to the deletion of sites as a safe synchronization point. The non-primary data stores were restored to the same point or by replaying the recorded events. Using a fixed restore point for primary stores enabled us to get consistency of data across all the data stores.

For 57 customers restored early on in our incident response, a lack of consistent policies and manual retrieval of database backup snapshots resulted in some Confluence and Insight databases being restored to a point *more* than five minutes prior to site deletion. The inconsistency was discovered during a post-restoration audit process. We have since recovered the remainder of the data, contacted the customers affected by this, and are helping them apply changes to further restore their data.

In summary:

- We met our Recovery Point Objective (RPO) of one hour during this incident.
- Data loss from the incident is capped at five minutes prior to the deletion of the site.

- A small number of customers had their Confluence or Insight databases restored to a point more than five minutes prior to site deletion, however, we are able to recover the data and are currently working with customers on getting this data restored.

Incident communications

When we talk about incident communications, it encompasses touch-points with customers, partners, the media, industry analysts, investors, and the broader technology community.

What happened

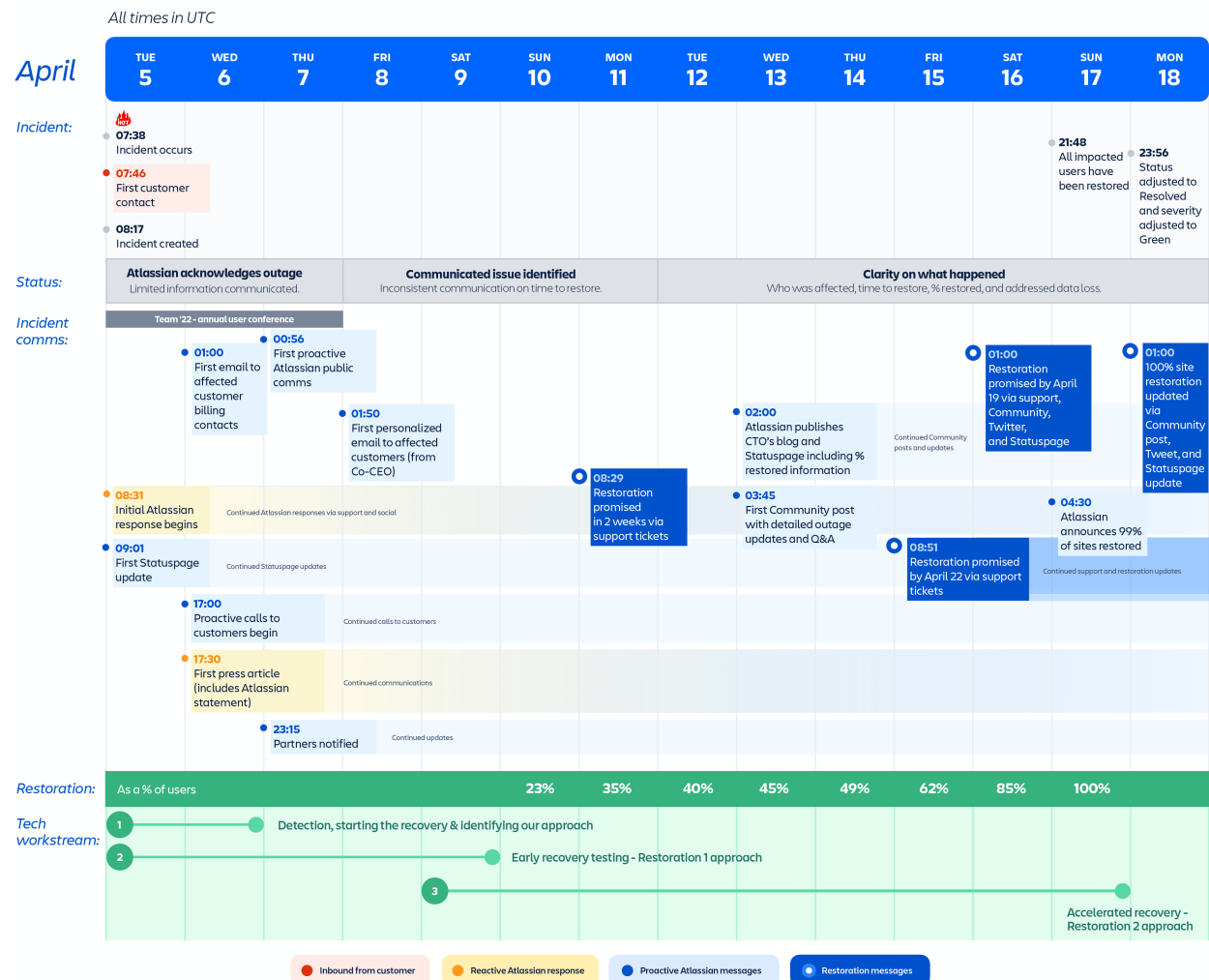


Figure 9: Timeline of the key incident communications milestones.

Timestamp: Days 1 - 3 (April 5th - 7th)

Early response

The first support ticket was created on April 5th at 7:46 UTC and Atlassian support responded acknowledging the incident by 8:31 UTC. At 9:03 UTC, the first Statuspage update was posted letting customers know that we were investigating the incident. And at 11:13 UTC, we confirmed via Statuspage that we had identified the root cause and that we were working on a fix. By 1:00 UTC on April 6th, the initial customer ticket communications stated that the outage was due to a maintenance script, and that we expected minimal data loss. Atlassian responded to media inquiries with a statement on April 6th at 17:30 UTC. Atlassian tweeted its first broad external message acknowledging the incident on April 7th at 00:56 UTC.

Timestamp: Days 4 - 7 (April 8th - 11th)

Broader, personalized outreach begins

On April 8th at 1:50 UTC, Atlassian emailed affected customers an apology from co-founder and co-CEO, Scott Farquhar. In the days that followed, we worked to restore the deleted contact information and create support tickets for all impacted sites that didn't yet have one filed. Our support team then continued to send regular updates about restoration efforts through the support tickets associated with each impacted site.

Timestamp: Days 8 - 14 (April 12th - 18th)

Greater clarity and complete restoration

On April 12th, [Atlassian published an update from CTO, Sri Viswanath](#), providing more technical details of what happened, who was affected, whether there was data loss, our progress on restoration, and that it may take up to two weeks to fully restore all sites. The blog was accompanied by another press statement attributed to Sri. We also referenced Sri's blog in our [first proactive Atlassian Community post from Head of Engineering, Stephen Deasy](#), which subsequently became the dedicated place for additional updates and Q&A with the broader public. An April 18th update to this post announced the full restoration of all affected customer sites.



Why didn't we respond publicly sooner?

1. We prioritized communicating directly with affected customers via Statuspage, email, support tickets, and 1:1 interactions. However, we were unable to reach many customers because we lost their contact information when their sites were deleted. We should have implemented broader communications much earlier, in order to inform affected customers and end-users about our incident response and resolution timeline.
2. While we immediately knew what had caused the incident, the architectural complexity and the unique circumstances of this incident slowed down our ability to quickly scope and accurately estimate time to resolution. Rather than wait until we had a full picture, we should have been transparent about what we did know and what we didn't know. Providing general restoration estimates (even if directional) and being clear about when we expected to have a more complete picture would have allowed our customers to better plan around the incident. This is particularly true for System Admins and technical contacts, who are on the front lines of managing stakeholders and users within their organizations.

Support experience & customer outreach

As previously mentioned, the same script that deleted customer sites also deleted key customer identifiers and contact information (e.g. Cloud URL, site System Admin contacts) from our production environments. This is notable because our core systems (e.g. support, licensing, billing) all leverage the existence of a Cloud URL and site System Admin contacts as primary identifiers for security, routing, and prioritization purposes. When we lost these identifiers, we initially lost our ability to systematically identify and engage with customers.

How was support for our customers impacted?

First, the majority of impacted customers could not reach our support team through the normal [online contact form](#). This form is designed to require a user to log in with their Atlassian ID and to provide a valid Cloud URL. Without a valid URL, the user is prevented from submitting a technical support ticket. In the course of normal business, this verification is intentional for site security and ticket triage. However, this requirement created an unintended outcome for customers impacted by this outage; they were blocked from submitting a high-priority site support ticket.

Second, the deletion of site System Admin data caused by the incident created a gap in our ability to proactively engage with impacted customers. In the first few days of the incident, we sent proactive communications to the impacted customer's billing and technical contacts registered with Atlassian. However, we quickly identified that many billing and technical contacts for the impacted customers were outdated. Without the System Admin information for each site, we did not have a complete list of active and approved contacts through which to engage.

How did we respond?

Our support teams had three equally important priorities to accelerate site restoration and repair the breakage in our communication channels in the first days of the incident.

First, getting a reliable list of validated customer contacts. As our engineering teams worked to restore customer sites, our customer-facing teams focused on restoring validated contact information. We used every mechanism at our disposal (billing systems, prior support tickets, other secured user backups, direct customer outreach, etc) to rebuild our contact list. Our goal was to have one incident-related support ticket for each impacted site to streamline direct outreach and response times.

Second, re-establishing workflows, queues, and SLAs specific to this incident. Deletion of the Cloud ID and the inability to authenticate users correctly also impacted our ability to process incident-related support tickets through our normal systems. Tickets did not appear correctly in relevant priority and escalations queues and dashboards. We quickly created a cross-functional team (support, product, IT) to design and add additional logic,

SLAs, workflow states, and dashboards. Because this had to be done within our production system, it took several days to fully develop, test, and deploy.

Third, massively scaling manual validations to accelerate site restorations. As engineering made progress through initial restores it became clear that the capacity of our global support teams would be required to help accelerate site recovery via manual testing and validation checks. This validation process would become a critical path to getting restored sites to our customers, once our engineering team accelerated data restores. We had to create an independent stream of standard operating procedures (SOPs), workflows, handoffs, and staffing rosters to mobilize 450+ support engineers to run validation checks, with shifts providing 24/7 coverage, to accelerate restores into the hands of customers.

Even with these key priorities well established by the end of the first week, we were limited in our ability to provide *meaningful* updates due to the lack of clarity around the incident resolution timelines due to the complexity of the restoration processes. We should have acknowledged our uncertainty in providing a site restoration date sooner and made ourselves available earlier for in-person discussions so that our customers could make plans accordingly.

How will we improve?

We have immediately blocked bulk site deletes until appropriate changes can be made.

As we move forward from this incident and re-evaluate our internal processes, we want to recognize that people don't cause incidents. Rather, systems allow for mistakes to be made. This section summarizes the factors that contributed to this incident. We also discuss our plans to accelerate how we will fix these weaknesses and problems.

Learning 1: “Soft deletes” should be universal across all systems

Overall, deletion of this type should be prohibited or have multiple layers of protection to avoid errors. The primary improvement we are making is to globally prevent the deletion of customer data and metadata that has not gone through a soft-delete process.

a) Data deletion should only happen as a soft-delete

Deletion of an entire site should be prohibited; and, soft-delete should require multi-level protections to prevent error. We will implement a “soft delete” policy, preventing external scripts or systems from deleting customer data in a Production environment. Our “soft delete” policy will allow for sufficient data retention so that data recovery can be executed quickly and safely. The data will only be deleted from the Production environment after a retention period has expired.

Actions:

- ✓ **Implement a “soft delete” in the provisioning workflows and all relevant data stores:** Additionally, the Tenant Platform team will verify that data deletions can only happen after deactivations, as well as other safeguards in this space. In the longer term, Tenant Platform will take a leading role to further develop correct state management of tenant data.

b) Soft-delete should have a standardized and verified review process

Soft-delete actions are high-risk operations. As such, we should have standardized or automated review processes that include defined rollbacks and testing procedures to address these operations.

Actions:

- ✓ **Enforced staged rollout of any soft-delete actions:** All new operations that require deletion will first be tested within our own sites to validate our approach and verify automation. Once we’ve completed that validation, we will progressively move customers through the same process and continue to test for irregularities before applying the automation to the entire selected user base.
- ✓ **Soft-delete actions must have a tested rollback plan:** Any activity to soft-delete data must test restoration of the deleted data prior to running in production and have a tested rollback plan.

Learning 2: As part of the DR program, automate restoration for multi-site, multi-product deletion events for a larger set of customers

[Atlassian Data Management](#) describes our data management processes in detail. To provide high availability, we provision and maintain a synchronous standby replica in multiple AWS Availability Zones (AZ). The AZ failover is automated and typically takes 60-120 seconds, and we regularly handle data center outages and other common disruptions with no customer impact.

We also maintain immutable backups that are designed to be resilient against data corruption events, which enable recovery to a previous point in time. Backups are retained for 30 days, and Atlassian continuously tests and audits storage backups for restoration. If required, we can restore all customers to a new environment.

Using these backups, we regularly roll back individual customers or a small set of customers who accidentally delete their own data. However, the site-level deletion did not have runbooks that could be quickly automated for the scale of this event which required tooling and automation across all the products and services to happen in a coordinated way.

What we have not (yet) automated is restoring a large subset of customers into our existing (and currently in use) environment without affecting any of our other customers.

Within our cloud environment, each data store contains data from multiple customers. Because the data deleted in this incident was only a portion of data stores that are continuing to be used by other customers, we have to manually extract and restore individual pieces from our backups. Each customer site recovery is a lengthy and complex process, requiring internal validation and final customer verification when the site is restored.

Actions:



Accelerate multi-product, multi-site restorations for a larger set of

customers: DR program meets our current RPO standards of one hour. We will leverage the automation and learnings from this incident to accelerate the DR program to meet the RTO as defined in our policy for this scale of incident.



Automate and add the verification of this case to the DR testing: We will regularly run DR exercises that involve restoring all products for large set of sites. These DR tests will verify that runbooks are up to date as our architecture evolves and any new edge cases are encountered. We will continuously improve our restoration approach, automate more of the restoration process, and reduce recovery time.

Learning 3: Improve incident management process for large-scale events

Our incident management program is well-suited for managing the major and minor incidents that have occurred over the years. We frequently simulate incident response for smaller-scale, shorter-duration incidents, that typically involve fewer people and teams.

However, at its peak, this incident had hundreds of engineers and customer support employees working simultaneously to restore customer sites. Our incident management program and teams were not designed to handle the depth, expansiveness, and duration of this type of incident (see *Figure 10* below).

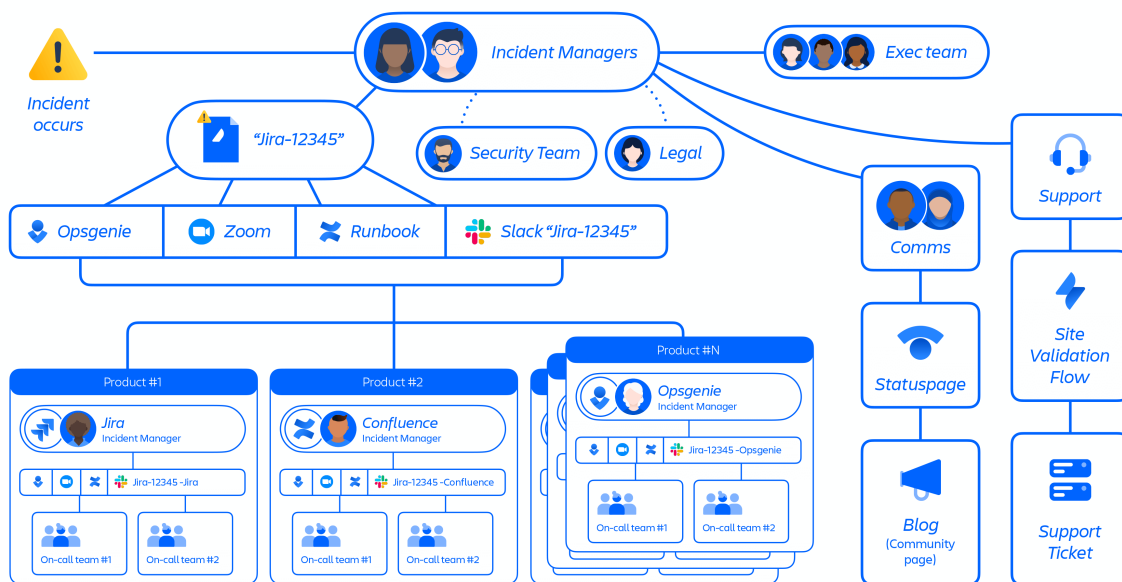


Figure 10: Overview of large-scale incident management process.

Our large-scale incident management process will be better defined and practiced often

We have playbooks for product-level incidents, but not for the events of this scale, with hundreds of people working simultaneously across the company. In Incident Management tooling we have automation that creates communication streams like Slack, Zoom, and Confluence doc but it lacks creating sub-streams that are required for large-scale incidents to isolate restoration streams.

Actions:



Define a playbook and tooling for large-scale incidents and conduct simulated exercises: Define and document the types of incidents that may be considered large-scale and require this level of response. Outline key coordination steps and build tooling to help Incident Managers and other business functions streamline the response and start recovery. Incident Managers with teams will regularly run simulations, trainings, and refinement of tooling and documents to continually improve.

Learning 4: Improve our communications processes

a) We deleted critical customer identifiers, impacting communications and actions to those affected

The same script which deleted customer sites also deleted key customer identifiers (e.g. site URL, site System Admin contacts) from our Production environments. As a result, (1) customers were blocked from filing technical support tickets via our normal support channel; (2) it took days for us to get a reliable list of key customer contacts (such as site System Admins) impacted by the outage for proactive engagement; and (3) support workflows, SLAs, dashboards, and escalation processes did not properly function initially because of the unique nature of the incident.

During the outage, customer escalations also came through multiple channels (email, phone calls, CEO tickets, LinkedIn and other social channels, and support tickets). Disparate tools and processes across our customer-facing teams slowed our response and made holistic tracking and reporting of these escalations more difficult.

b) We did not have an incident communications playbook thorough enough to deal with this level of complexity

We did not have an incident communications playbook that outlined principles as well as roles and responsibilities to mobilize a unified, cross-functional incident communications team quickly enough. We did not provide acknowledgment of the incident quickly and consistently through multiple channels, especially on social media. More broad, public communications surrounding the outage, along with the repetition of the critical message that there was no data loss and this was not the result of a cyberattack, would have been the correct approach.

Actions:

- ✓ **Improve the backup of key contacts:** Backup authorized account contact information outside of the product instance.
- ✓ **Retrofit support tooling:** Create mechanisms for customers without a valid site URL or Atlassian ID to make direct contact with our technical support team.
- ✓ **Customer escalation system and processes:** Invest in a unified, account-based, escalation system and workflows that allow for multiple work objects (tickets, tasks, etc) to be stored underneath a single customer account object, for improved coordination and visibility across all of our customer-facing teams.
- ✓ **Expedite 24/7 Escalation Management coverage:** Execute against global footprint expansion plans for the Escalation Management function to allow for consistent 24/7 coverage with designated staff based in each major geographic region along with support roles to assist with required product and sales subject-matter experts and leadership.
- ✓ **Update our incident communications playbook with new learnings and revisit it regularly:** Revisit the playbook to define clear roles and lines of communications internally. Use the [DACI](#) framework for incidents and have 24/7 back-ups for each role in case of sickness, holidays, or other unforeseen events. Conduct a quarterly audit to verify readiness at all times.

Actions (cont.)

Follow the incident communications template in all communications: address what happened, who was impacted, timeline to restoration, site restoration percentages, expected data loss, with the associated confidence levels, along with clear guidance on how to contact support.

Closing remarks

While the outage is resolved and customers are fully restored, our work continues. At this stage, we are implementing the changes outlined above to improve our processes, increase our resiliency, and prevent a situation like this from happening again.

Atlassian is a learning organization, and our teams have certainly learned a lot of hard lessons from this experience. We are putting these lessons to work in order to make lasting changes to our business. Ultimately, we will emerge stronger and provide you with better service because of this experience.

We hope that the learnings from this incident will be helpful to other teams who are working diligently to provide reliable services to their customers.

Lastly, I want to thank those who are reading this and learning with us and those who are part of our extended Atlassian community and team.

-Sri Viswanath, CTO