

Self-Healing DevOps

(x) matters[®]  dynatrace

Innovation or Uptime: Why Choose?



By: Erin Jones,
Product Marketing
Manager at xMatters

TABLE OF CONTENTS

3 | Process Crash Autoremediation

6 | Full Disk Prevention

9 | Automated Blue-Green Deployment Rollback

When the concept of incident management was born, it was within IT organizations charged with keeping the lights on. IT managers wore pagers in case there was a system outage and they needed to be notified. It wasn't unusual to drive miles in the middle of the night to restart a server. The processes were necessary, but manual. Over time as IT undergoes a digital transformation and leans into efficiency and automation to more quickly troubleshoot incidents, new standards for support and response have emerged and made their way into other parts of these organizations - including DevOps.

While DevOps is a relatively new concept (in the grand scheme of technology practices), developers were the early adopters of best practices, such as testing automation and responsive collaboration through ChatOps before other departments caught on.

As IT functions move more and more into the digital side of the house and require teams to not only build, but support releases, development teams are now charged with thinking about how they do incident management. But for teams that support hybrid infrastructures, a multitude of microservices, as well as pushing regular releases (often more than one per day) - incidents aren't just uptime vs. downtime.

The name of the game is now customer experience - and while downtime is still a major incident, any impact on services can be detrimental to the end user. Teams are now tasked with not only providing innovative (and frequent) new services to customers, but also ensuring that every little thing that could derail a release is minimized (if not prevented). This can make releases feel like a landmine with the risk of performance degradation, process crashes, and errors at every part of the continuous deployment process.

AI monitoring is a must-have for digital teams to give real-time insights across a growing tech stack and increasingly complex environments. Being able to make this data actionable across your DevOps processes is what closes the performance gap between innovation and risk.

In this eBook, we'll look at how Gartner application performance monitoring leader, Dynatrace, and enterprise incident management platform, xMatters integrate to provide the safety net to close the performance gap and embed automated self-healing into the DevOps process (and tech stack). You'll see, unpacked in three use cases, examples of an automated toolchain built in xMatters' Flow Designer, a codeless visual workflow builder. And while each part may be a several minute-long read, the actual creation and execution of these self-healing steps actually occurs in a matter of seconds - getting your services back up and running before customers even notice.

Automated Self-Healing Part I: Process Crash Remediation with xMatters and Dynatrace

In DevOps, incidents aren't always a lights-out experience. Incidents more often start with one service crashing, then a domino effect that takes other services out with it. These system dependencies occur at every layer of the application stack. With [84% percent of organizations already using a hybrid cloud infrastructure and another 58% planning to move in 2019](#), the battle between separate but interrelated systems is the new reality of DevOps teams.

According to a recent report, [organizations report using up to 124 different SaaS-based apps to keep their business running](#).

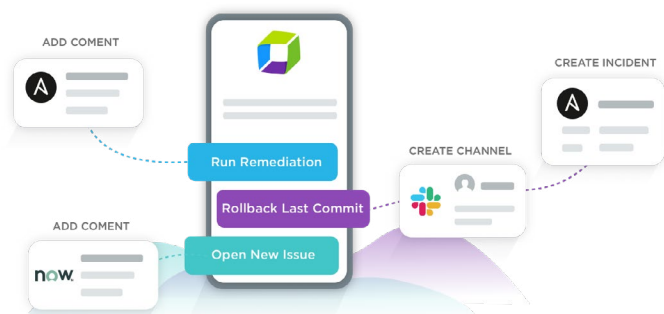
While you might be tempted to decouple dependencies to keep one failure from taking out the service of other teams – and seemingly isolating issues so you can more quickly identify the problem – siloing your services and teams is counterproductive to continuous improvement.

In this blog series, we'll dive into how DevOps teams can take preventative (and not just reactive) action in 3 different scenarios by leveraging xMatters high performance collaboration platform with Dynatrace's unique AI-powered full stack monitoring.

1 | Process Crash

2 | Full Disk:

3 | Slow Microservices



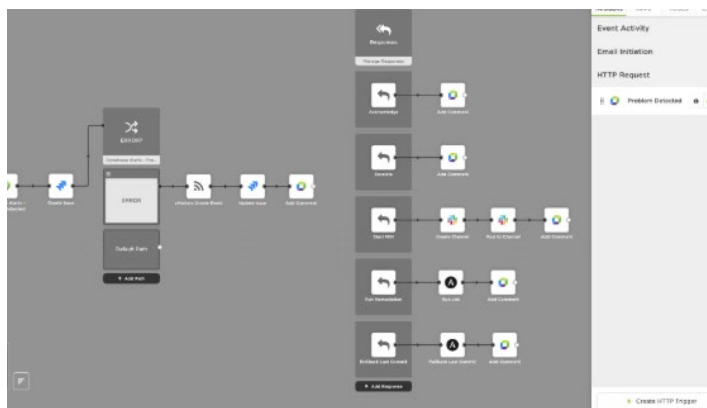
Roll back a release or apply a hotfix with xMatters and Dynatrace.

Scenario #1 | Process Crash

Let's look at how a DevOps team can leverage integrations through [xMatters Flow Designer](#) to prevent a process crash from taking down services across the organization.

The workflow in xMatters Flow Designer (below) shows the toolchain we will use to fix a process crash before it impacts other services or customers.

Tools across the workflow are dragged onto the Flow Designer palette in series of steps to fully automate the remediation process.



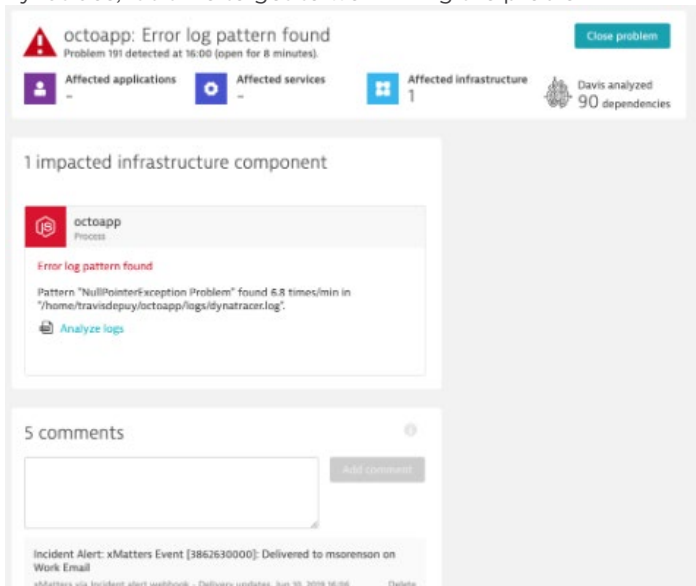
Drag tools into Flow Designer to automate remediation.

What are the key applications to have in your DevOps stack?

Step 1 | Intelligent Application Performance Monitoring:

The ability to not only monitor, but intelligently identify root cause of incidents is key. Your application performance monitoring tool is the first line of defense for detecting performance degradation or anomalies that indicate an incident is brewing.

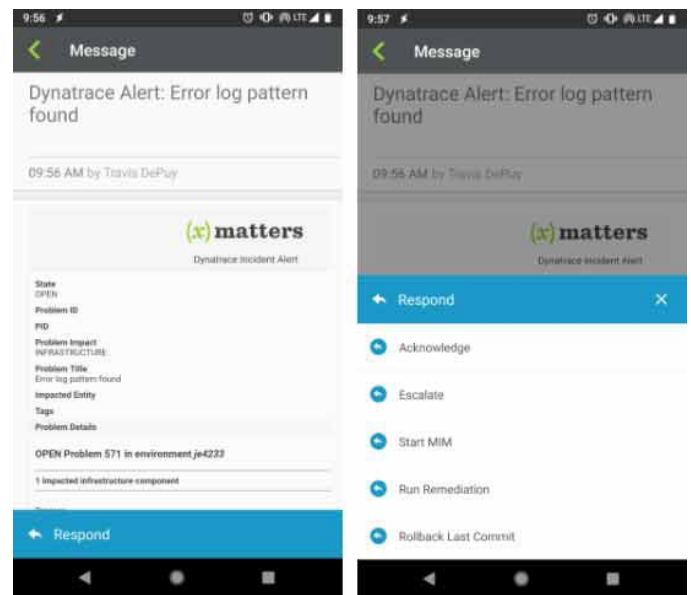
Dynatrace is built to understand how dependencies impact other areas across your infrastructure; should an issue arise, Dynatrace's [AI-driven monitoring](#) automatically relates these to pinpoint root cause and provide context around other impacted services. Once the root cause is identified in Dynatrace, it's time to get to work fixing the problem.



Dynatrace identifies the root cause so you can fix the problem.

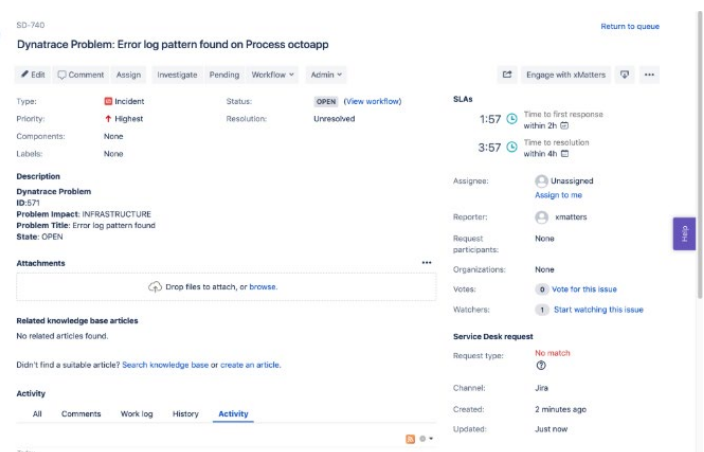
Step 2 | Incident Management: To fix the issue before the next domino falls, you can't wait on someone to log into their ticketing system or email inbox to see the notification. You need to immediately engage the right people with the issue information from Dynatrace to kick off the resolution process. xMatters, a Digital Service Availability Platform and official Dynatrace partner, uses your on-call schedules, groups, and teams to significantly reduce the time to notify the right responders.

But it's not enough to just page someone and have them then log into disparate systems to investigate the issue. That's why xMatters passes the Dynatrace data through alerts, giving you full incident context to inform your remediation path. Depending on the type of Dynatrace issue, xMatters prompts on-call resources with response option buttons that launch workflows across your systems to automatically run your remediation process – and to keep stakeholders and customers updated!



xMatters passes Dynatrace data into alerts (left) with actionable responses (right).

Step 3 | Service Desk: Ticketing systems like Jira Service Desk and Zendesk have become the gold standard for DevOps source of truth. Teams rely heavily on tickets during post mortems so they can identify repeatable processes that can be put in place to help prevent similar incidents in the future. That means tickets should ideally include all incident details, as well as the remediation steps and timeline.



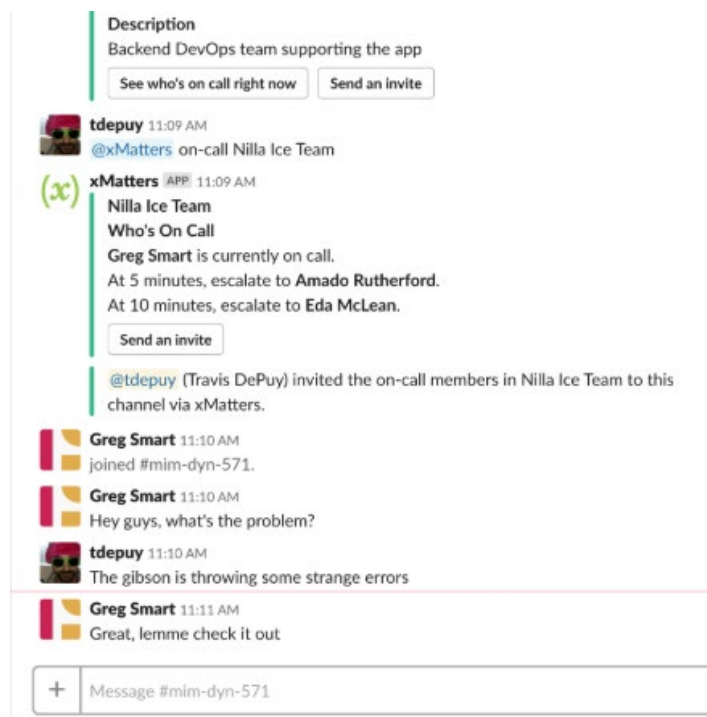
xMatters creates and updates Jira issues with incident information from Dynatrace.

But the reality is that while they're in the middle trying to resolve an incident, few people take the time to create and update tickets. Instead, they often spend extra time post-resolution to manually transfer incident details from multiple systems, which is both inefficient and prone to error.

That's why bi-directional integration is paramount. When integrated with xMatters, Jira steps become part of an automated toolchain process, allowing xMatters to create tickets (with full Dynatrace incident data included), assign and update them, and append them with incident resolution information from your other systems (like Slack and Ansible).

xMatters creates and updates Jira issues with incident information from Dynatrace

Step 4 | ChatOps: During an incident, DevOps on-call resources typically turn to a chat platform like Slack (rather than comment threads on tickets). The automated workflow launched from xMatters provides a step that spins up a dedicated Slack channel populated with your critical Dynatrace incident data. Slackbot then references your on-call schedule and groups to invite your teammates to join so you can discuss and execute remediation actions in your other tools (like Ansible) without ever leaving



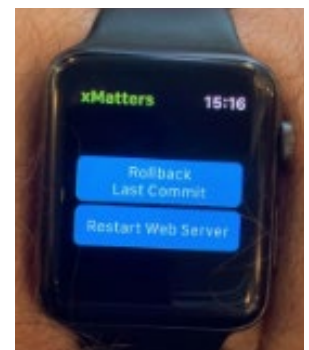
xMatters creates and updates Jira issues with incident information from Dynatrace.

Slack. Once the incident is resolved, your Slack channel transcript is automatically attached to the related Jira issue, ready for your post mortem review.

xMatters creates a dedicated Slack channel wherein users leverage the Slackbot to find and invite the right team to join. (Fun fact: This entire Slack transcript will be auto-appended to the related Jira issue without any manual work!)

Step 5 | Configuration Management:

When it's time to push your fix, a configuration management tool such as Ansible allows you to quickly control and execute jobs in your impacted systems. Dynatrace ensures you know what services are affected so you can launch the proper workflow automation from xMatters to Ansible (while simultaneously updating Jira, Dynatrace, and your other integrated systems).



One-click rollbacks - even from your wrist - with the xMatters Mobile App!

Execute rollbacks with one button - even from your wrist - with the xMatters Mobile App!

Preventing your next process crash

A great customer experience is no longer just about uptime - it's about performance. Progressive organizations are keenly aware that functional degradation and slow service impacts user experience, so they take preventative measures and more quickly fix issues.

By integrating your systems and putting in place a remediation toolchain to bring a service back up, you drastically reduce incident time and limit impact to customers, other services - and your company's bottom line.

**Ready to get proactive?
xMatters can help.**

Self-Healing DevOps with xMatters and Dynatrace Part II: Full Disk Prevention

Anatomy of a Full Disk

The full disk error strikes fear in the hearts of DevOps teams. A full disk prevents incoming requests from being processed, bringing services to a screeching halt. However, teams using [Dynatrace](#) for application performance monitoring can quickly pick up on the warning signs before it gets to this level of severity [blog](#)

Dynatrace monitors your disk health to pick up on performance metrics that fall below your team's Service Level Indicators (SLIs) so you catch issues before they snowball into a full, inoperative disk. Measuring key performance indicators allows you to kick off remediation workflows the moment any degradation is detected. [Disk health](#) KPIs include:

Throughput: The total number of bytes read and written to disk per second

Disk latency: time from I/O request submission to I/O request completion. The average delay of disk read and write operations in milliseconds. This metric is used to detect host slow disk incidents

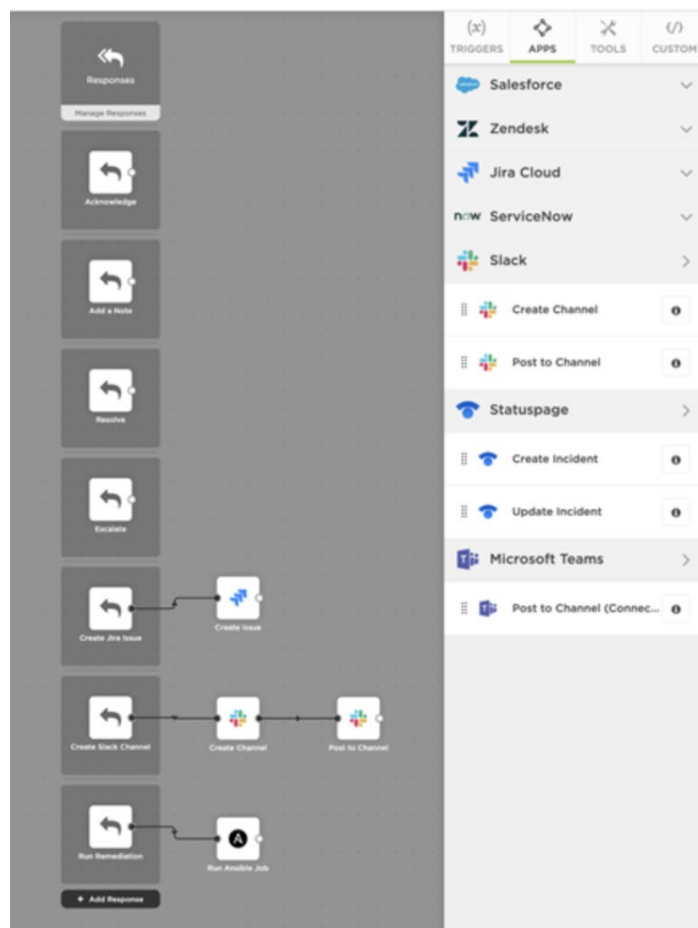
Disk space usage: The amount of disk space that's been used

Idle time: Amount of time the disk has been idle

For the second scenario in our self-healing DevOps series with Dynatrace, we'll look at how early detection helps teams respond to prevent a full-stop full disk.

Scenario #2 | Full Disk Prevention

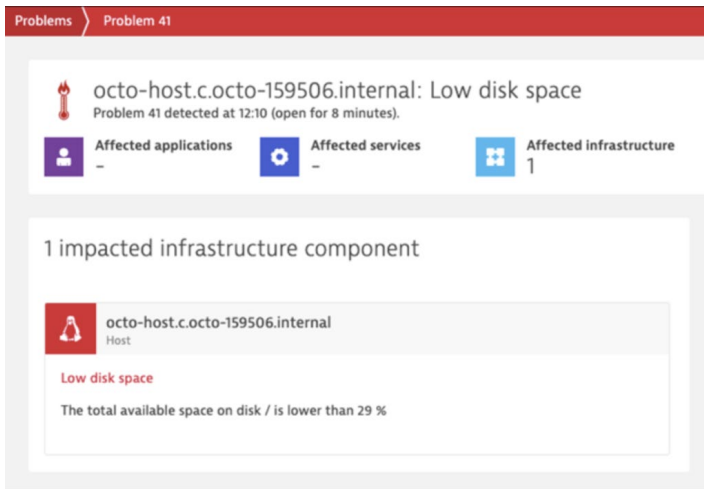
Let's start by looking at the process flow across systems, built in [xMatters Flow Designer](#), that makes your Dynatrace data actionable and automates remediation steps at the touch of a button. Not only does this provide responders the ability to immediately take action on alerts, but it helps standardize processes while removing manual work.



Drag tools onto the Flow Designer palette to automate remediation.

Step 1 | Application Performance Monitoring

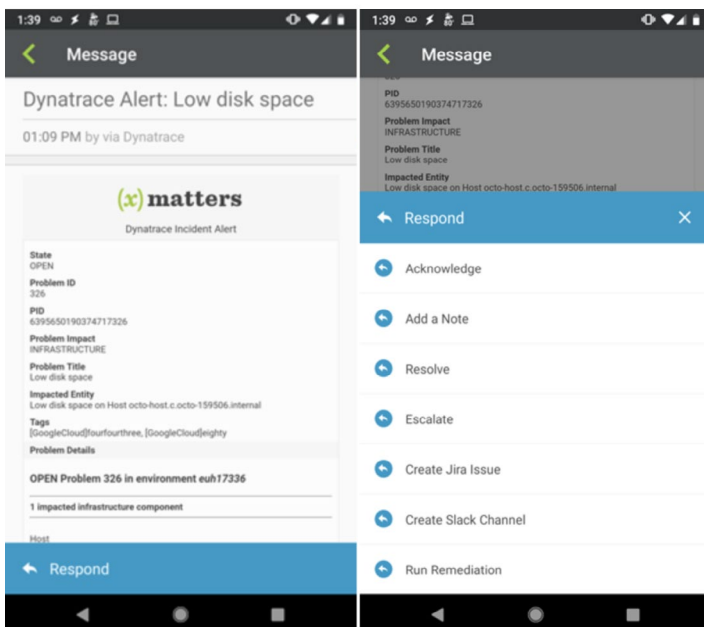
Because Dynatrace's AI-driven monitoring keeps a real-time pulse of your disk health, it surfaces an event when disk latency is first detected. With automated root cause analysis, Dynatrace also correlates related performance errors, such as throughput and IOPS, giving you the full picture of the issue and its impact across service level indicators. When you see the performance issue in time, you can take immediate action before you get a full disk error.



Dynatrace identifies the root cause so you can fix the problem.

Step 2 | Incident Management

Leveraging the workflow you created in Flow Designer, xMatters triggers an alert based on the Dynatrace data. Targeting the proper on-call resources, xMatters gives respondents the [incident information](#) from Dynatrace within the alert (no need to click through to additional system dashboards). Based on this contextual data, resources are prompted with their pre-configured response options, each of which kicks off a workflow across systems (based on the severity of the issue). As xMatters

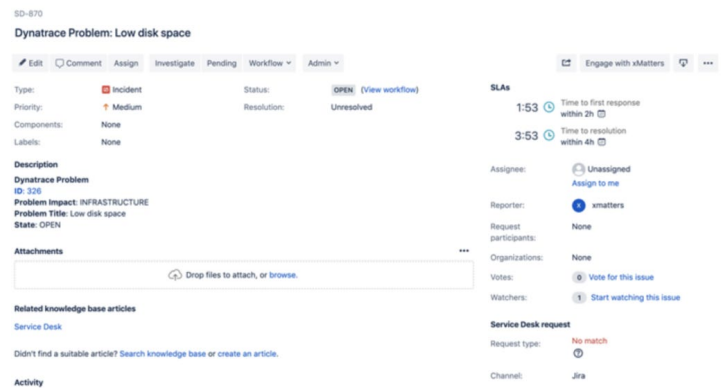


xMatters passes Dynatrace data into alerts (left) with actionable responses (right).

decreases time to notify and respond, on-call resources simply select the right response and launch a workflow that restores disk health while simultaneously documenting the issue in their chat and service desk.

Step 3 | Service Desk

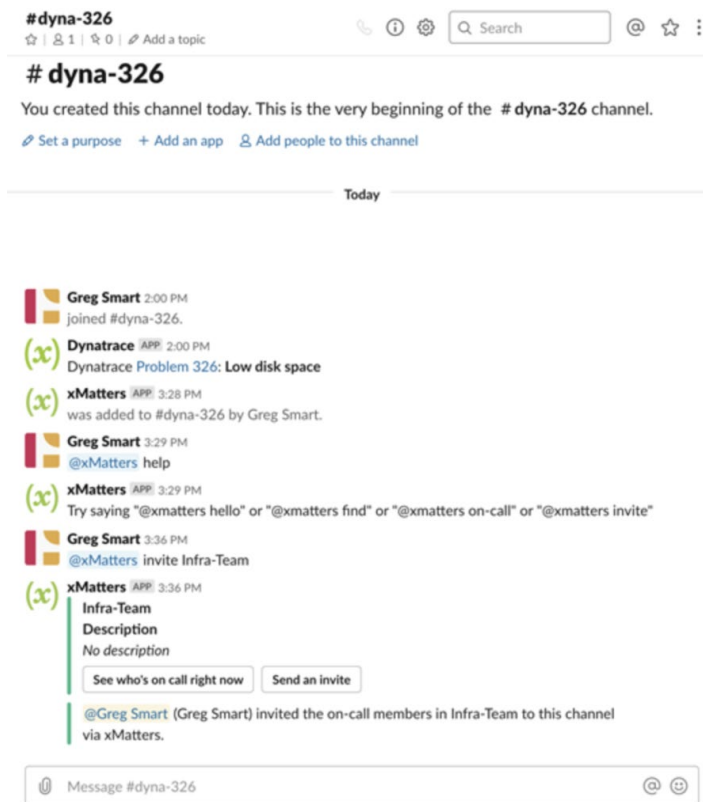
Whether your team uses [Jira Service Desk](#), [Zendesk](#), or [ServiceNow](#) (or any combination therein), your workflow in Flow Designer triggers a ticket to be created and automatically updates it with the incident data from Dynatrace. On-call resources select this response and the job runs. In the background, all integrated systems (Dynatrace, Jira Service Desk, and Slack) are updated with details of the action taken and the newly improved performance data.



xMatters creates and updates Jira issues with incident information from Dynatrace.

Step 4 | ChatOps

Similarly, your Flow Designer workflow automatically creates a chat channel (our [Slack](#) and [Microsoft Teams](#) integrations are the most popular). With the use of the respective chat bots, the Dynatrace incident information is incorporated in chats. You can reference [on-call schedules](#) and teams to invite the right resources to join the conversation. Once the incident is resolved, your chat transcript will also be automatically attached to the respective service desk ticket to give you and your team a full picture of what occurred for easier post-mortems.



xMatters creates a dedicated Slack channel where users use the Slackbot to find and invite the right team to join. Click for a bigger image.

No more full disk errors

A full disk can be catastrophic for your services (and your end users). But because we can pinpoint symptoms of an impending full disk before it strikes, you can take proper action and automate workflows across systems to fix it fast.

Imagine what a rock star you'll be (and how much better you'll sleep at night) knowing that you have Dynatrace and xMatters as your first line of defense against threats to your disk health lurking around the corner. To learn more, read the recent [Dynatrace blog](#) and visit the [Dynatrace website](#) for more information.

**Ready to kiss full disks goodbye?
xMatters can help.**

Step 5 | Configuration Management

As the above steps are being executed, xMatters triggers a runbook in Ansible to push the disk latency fix. In this case, your team determines that deletion of temporary files from the disk to free up processing power will return the disk to its required performance health. The job is run as on-call resources select this response option and, in the background, all integrated systems (Dynatrace, Jira Service Desk, and Slack) are updated with details of the action taken and the newly improved performance data. With a full disk alert successfully averted, resources can now resolve the issue from xMatters, across all your systems.

Self-Healing DevOps Part III: Automated Blue-Green Deployment Remediation

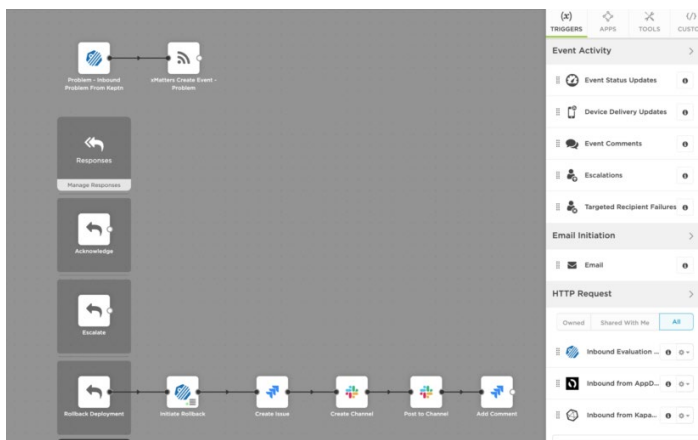
Balancing innovation & risk

According to Dynatrace's 2018 Global CIO Report, [73% of CIOs say the need for speed in digital innovation is putting customer experience at risk](#). The same report noted the organizations polled pushed three releases an hour.

To meet the high volume of new releases while accounting for regular risks like performance degradation and process crashes, many DevOps teams use blue-green deployments as part of their [continuous deployment practices](#).

To quote [DevOps thought leader Martin Fowler](#), "The blue-green deployment approach... [ensures] you have two production environments, as identical as possible. At any time one of them, let's say blue for the example, is live. As you prepare a new release of your software you do your final stage of testing in the green environment. Once the software is working in the green environment, you switch the router so that all incoming requests go to the green environment - the blue one is now idle."

So how do you automate remediation across two production environments to ensure you take automated actions?



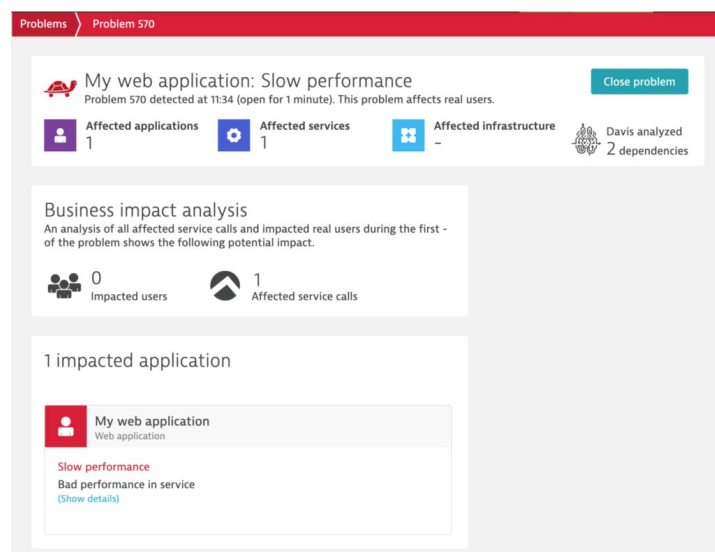
Dynatrace and xMatters remove the stress from blue-green deployments.

Scenario #3 | Autoremediation for Blue-Green Deployments

Step 1 | Application Performance Monitoring

In an era of increasingly complex environments, Dynatrace monitors across your entire application infrastructure to keep proverbial eyes on everything at once.

With Dynatrace in place, resources quickly and easily see when the new release pushed to the blue environment is having a detrimental effect on the performance of interrelated microservices across this environment, slowing applications to a near-grinding halt. Dynatrace surfaces this issue, including pertinent data about the buggy release itself as well as critical information on how it's impacting other services - and where they live.



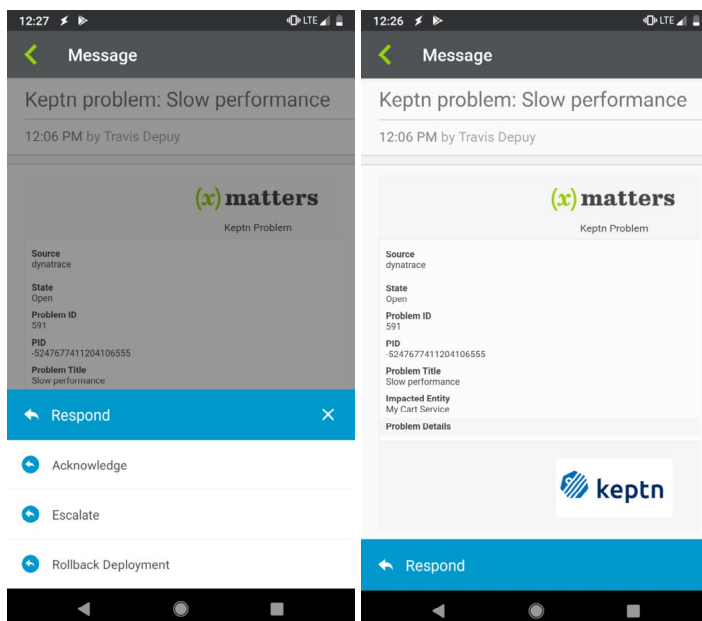
Dynatrace and xMatters remove the stress from blue-green deployments.

Step 2 | Control Plane

The code that was shipped may have passed pre-deployment rounds of testing, but it's now impacting other services running. The team must take immediate action to revert to the previous version in the parallel, green environment with Dynatrace's keptn. In the words of Andreas Grabner of Dynatrace, this new product "...not only orchestrates Continuous Deployment, but it also orchestrates Continuous or Automated Operation." So, as Dynatrace surfaces the incident, keptn simultaneously captures the issue data and triggers an xMatters workflow. This allows the developer on-call to take swift action and fix the problem.

Step 3 | Incident Response

Triggered by keptn, xMatters references the [developer on-call schedule](#) and, in moments, alerts the proper resource. The [notification automatically includes the full context of the issue](#) even on a mobile device, including errors caused, impacted systems, and level of severity. Now the resource can quickly investigate the situation and determine the best response action. With the push of a button, the resource launches a remediation workflow across the prescriptive DevOps toolchain to automate rollback to the green environment through keptn. The same workflow also embeds critical information into the proper channels like [Slack](#), [Jira Service Desk](#), and [Dynatrace](#).



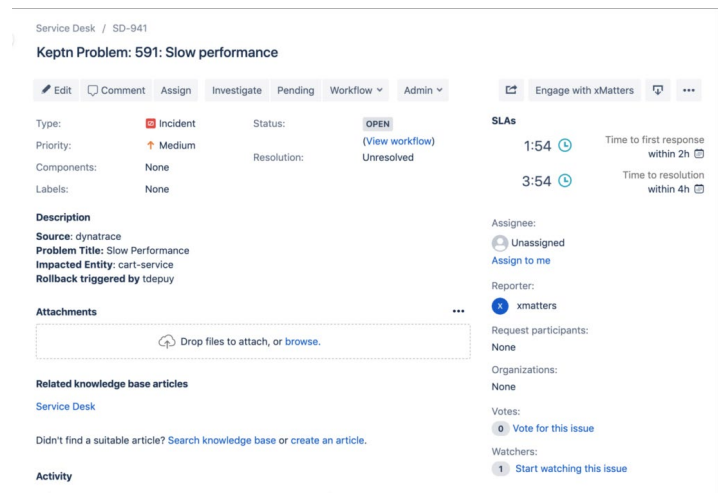
Keptn triggers the xMatters alert with contextual incident information.

xMatters provides on-call resources actionable responses to launch the proper response in the click of a button.

Step 4 | Service Desk

Fully automated and without any manual invention, xMatters creates a ticket in the service desk (or multiple service desks, if your organization uses more than one), embedding incident information into the ticket and automatically updating the status and comments therein so it can be easily referenced in team post-mortems. This takes place at the same time xMatters is triggering the rollback, so responders don't have to choose between taking immediate remediation action or starting full timeline documentation in your service desk.

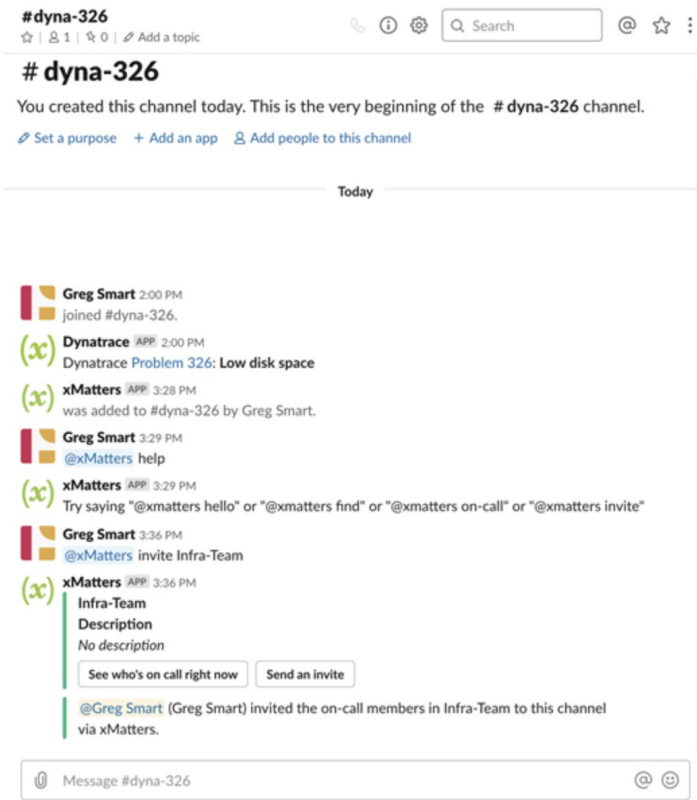
xMatters creates a Jira Service Desk ticket and passes through related information to capture data across tools and automatically update based on actions taken.



xMatters creates a Jira Service Desk ticket and passes through related information to capture data across tools and automatically update based on actions taken.

Step 5 | ChatOps

Because the release error has impacted services owned by other developers, xMatters spins up a dedicated Slack channel where you select the relevant teams to invite to the channel. All of the incident information is pushed into the channel through xMatters, so those who join get immediate context. As you collaborate in chat, on-call resources can also use the xMatters bot to update the related service desk ticket. The full chat transcript from this channel will also be automatically appended to the service desk ticket once the issue is resolved, giving you even more detail for your post-mortem exercise.



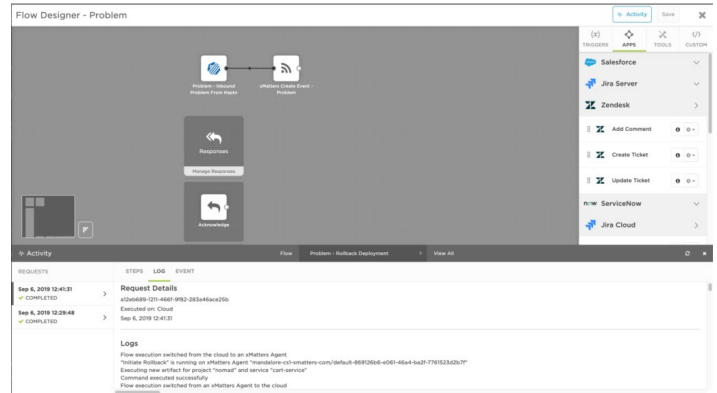
After starting a dedicated Slack channel, the xMatters Slackbot pulls the on-call database to invite the right resources to the channel,

Step 6 | Rollback Execution

Within moments of pushing the xMatters response button to launch the remediation workflow, the rollback to the previous, stable version in the green environment is executed through keptn, fixing the issue just minutes from detection. With the incident resolved, the related chat channels and service desk issues are automatically closed.

Step 7 | Post-Mortem and Reporting

Once the incident is resolved and versioning rolled back to the green environment, it's time to look at what happened, how you responded, and how to prevent things like this in the future. Because xMatters automatically appended and updated your incident management systems - from chat to service desk to monitoring - with the incident information and steps taken to resolve, the development team has easy reference to this during the post-mortem meeting. Furthermore, as xMatters retains historical incident data, you



Per the remediation workflow built in xMatters Flow Designer, responders have the option to execute a rollback through keptn - all through their mobile devices.

can quickly cross-reference similar incidents to identify any patterns of issues to better prevent future incidents. [User-level analytics](#) in xMatters help you know which team members broke their personal record for mean time to respond.

Continuous Deployment, Continuous Self-Healing

Pushing releases at break-neck speeds to keep pace with innovation is not for the faint of heart. The possibility of what could go wrong is just as present as the excitement around shipping the code in the first place. By creating parallel environments, developers get the benefit of being able to roll back to the previous, stable version of any release. The ability to do this in an instant, when customers are counting, is critical. Giving on-call developers the ability to execute a rollback with the push of a button provides a safety net for teams that need to quickly and frequently release new services. No need to break into a cold sweat with your next release; if anything goes wrong, you can now push a button and get everything back to working order so quickly, your customers won't even notice anything happened.

Ready to eliminate risk and enhance your innovation? xMatters can help.

(x) matters®

xMatters is a digital service availability platform that helps enterprises prevent, manage, and resolve IT incidents. From the Global 2000 to small workgroups and innovative DevOps teams, organizations around the world rely on xMatters to maintain operational visibility and control in highly complex IT environments. xMatters is trusted by leading global companies including BMC Software, Credit Suisse, Danske Bank, DXC technology, Experian, NVIDIA, ViaSat, and Vodafone. xMatters is headquartered in San Ramon, California and has offices worldwide.



Dynatrace provides software intelligence to simplify enterprise cloud complexity and accelerate digital transformation. With AI and complete automation, our all-in-one platform provides answers, not just data, about the performance of applications, the underlying infrastructure and the experience of all users. That's why many of the world's largest enterprises trust Dynatrace to modernize and automate enterprise cloud operations, release better software faster, and deliver unrivalled digital experiences.

xMatters.com