TechTarget | Search ITOperations

E-guide

PRO+ Content

Don't panic! The definitive guide to IT troubleshooting

In this e-guide:

IT troubleshooting is a vital part of operations in any organization, whether it hosts applications on the cloud, on premises or across a combination of the two. IT problems arise in monolithic and legacy applications, as well as in distributed microservices, where they take a little extra effort to chase down.

The response process and methodology are at least as important as the tools in use. Users are an important piece of the puzzle and can prove major assets for IT troubleshooting. And there are numerous ways to get the most out of IT monitoring and log management software. Root cause analysis is a matter of logic, tools and knowledge.

From how to engage superusers to the promise of AI-embedded tools, these articles from IT experts in the field and independent analysts have your organization's back.

IT's application support model depends on everyone outside IT

Adam Fowler, IT operations manager

https://searchitoperations.techtarget.com/tip/ITs-application-support-model-depends-on-everyone-outside-IT

In many organizations, IT service management lacks user and business consultation. A successful application support model must go beyond IT.

DevOps has succeeded as an application support model in part because it caters to user experience. The feedback loop between IT operations, developers and users must be as short as possible, but processes must fall below the business's established risk threshold as well.

Application owners have a role to play

Every key business application requires at least one advocate, but the application support model changes from one to another. Some specialty applications need dedicated owners who understand the product well and who know who to contact internally or externally for help. Others, such as those for more common tasks, need superusers who not only know the product well but also have the drive to improve it and will provide the IT team with feedback.

These team members take some application support responsibilities off of IT. They can answer the easier questions from other users, as well as lend their understanding and expertise to troubleshooting and change management. They contribute directly to business performance improvement and outage reduction through involvement in update commits and setting changes. App owners and superusers should test out changes first and have direct access to the IT staff members who manage, change and develop the application, rather than go through the help desk.

IT administrators, app owners and superusers should be in an easy-to-find list. Users can know who to contact about application issues or questions, and the list also aids the help desk staff who have front-line application support roles.

Benefits of feedback

Feedback should be constructive, simple to follow and encouraged in company culture -- easier said than done. To reach that feedback goal, start with key applications or processes, and grow at a rate that users and IT staff can accommodate.

Organizations have numerous methods to collect feedback, such as email, forms and help desk incident requests. Choose a method for your application support model where the user can give feedback easily and recipients can respond just as easily. If users submit feedback and never get a response, they won't be motivated to comment in the future.

Some feedback is impossible to act upon, but most users would rather hear "no" with an explanation -- even if they don't like the answer -- than receive no reply at all. For actionable feedback, improvements demonstrate to the business that IT can adapt and change. Inversely, feedback can lend business justification to IT's change requests that get held up by resource limits.

Don't just make changes indiscriminately when responding to feedback. Just because a few users want Comic Sans as their default font doesn't mean that it should apply to the rest of the business. Where standardization makes sense, enforce it. Where it doesn't, leave flexibility in the app support model to let users customize their experience.

Automation in the right places

Not everything can, or should, be automated, but sloughing off user error and tediousness from processes benefits everyone in app support. For example, barcode readers automate asset management and lead to more accurate records. Staff are more likely to keep the assets in an application stack up to date because a machine now reads the eye-straining ID number and submits it accurately to the designated system.

IT must speak regularly with business leaders to identify which processes and procedures can be automated. IT cannot expect the organization's user base to ask about automating procedures; often, they are unaware of the possibilities. Users might also resist automation out of fear of being replaced by a robot, but as with the barcode reader example above, someone must still oversee and even do a lot of the work.

There are other ways to improve the application support model and IT service management overall. Look for opportunities that fit the business's and application users' needs, and implement ones that have the biggest effect.

Next article

Use an IT incident management system to enable automation

Adam Fowler, IT operations manager

https://searchitoperations.techtarget.com/tip/Use-an-IT-incident-management-system-to-enable-automation

No IT environment is perfect. Issues can range from simple problems, such as a server running low on disk space that causes an application to stop responding, to more complex intermittent issues, such as a finance system that runs poorly at the end of each month when the accounting department prints a year's worth of invoices.

An IT operations administrator might not be able to predict every issue, but an automated IT incident management system might not either.

Systems and operations are hard to look after. They're hard to implement, manage and troubleshoot. Environments change constantly, and ops admins must set up monitoring and change management for all seven layers of Open Systems Interconnection -- eight, if you include users. Every environment is unique and, in turn, imperfect.

Where do you draw the line between employing IT gurus who manually maintain and fix the environment and investing in automated IT incident management systems that report or even remediate issues? Automation and internal IT knowledge must coexist for the best chance at a highly operational environment.

Where to place the line between these two is a problem for each company to work out individually. For the best results, have expert IT staff rely on the systems that show environmental health, rather than act as eternal emergency repair personnel. Humans are fallible, and you also are unlikely to have multiple staff members with the exact same knowledge across all systems inside the company.

Incident management options

A properly configured IT incident management system uses monitoring tools to pick up an issue before a human does. For example, if a remote site's WAN link goes down, it might go undetected until an end user complains. However, a monitoring tool that tracks the availability of any device at the other end of the WAN link -- or even an IP address of the router that provides the WAN link -- will find an anomaly quickly. The IT team can use monitoring settings to trigger an event, such as send an email alert to the whole IT team. The IT experts determine the cause and communicate about the problem to users. To receive an alert from an automated IT incident management system and then act upon it requires less technical and environmental knowledge from the first-line support team than to troubleshoot an issue brought up by a user, especially when the user's understanding of the issue is unclear.

IT incident management systems are becoming more flexible and powerful. For example, Microsoft Azure's Operations Management Suite uses basic functions, such as centralized logging, along with advanced features, such as Service Map, which automatically discovers and builds a dependency reference map of servers, processes and third-party services.

IT incident management systems that incorporate tools like Service Map, application dependency mapping and other features move the effort and work required to address issues out of the hands of the internal IT expert, who must remember every server by IP address, name and disk capacity. Instead, an ops admin can follow standard instructions to set up monitoring and incident management and visualize how specific servers and services interact. If this work is done at the time of system build, complex systems can be self-documenting to record and show connectivity and requirements of all moving parts. The result is that, when something breaks, you can quickly see inside the system and easily discover the point of failure.

Advanced options

Some large companies place heavy emphasis on automation, including for their IT incident management systems. "Automate all the things" is a popular IT catchphrase, but full automation can make little sense for resource management, depending on what the tasks are. Advanced automated IT incident response can lead to a self-healing infrastructure, but that's beyond the reality for most organizations. Automation must start with the most basic processes and build up for any hope of a fluid and functional state, leaving the promise of end-to-end automation as a dream out in the ether.

Another way to approach IT incidents is to create them yourself. Netflix developed a chaos engineering program, Chaos Monkey, and sister tools collectively called the Simian Army, which test system resiliency by purposefully breaking processes or disrupting services. More conservative organizations can experiment with chaos engineering in small doses or in staging environments, rather than take down production systems.

Ultimately, IT operations admins exist to help the rest of the business to do its job. Quick to deploy and easily modifiable automated IT monitoring and incident management systems make this task easier. The right tooling will make a difference. If the effort to set up monitoring and remediation processes seems too cumbersome, maintenance will only become more so when future system changes occur; at the current rate of change in IT, that will be a continual job. Even with a successful IT incident management system with built-in automation, admins still need to understand their environments. Combine that knowledge with automation, and they'll do less manual troubleshooting and have better focus on where to seek out issues.

↘ Next article

Simple steps to improve the IT incident management process

Adam Fowler, IT operations manager

https://searchitoperations.techtarget.com/tip/Simple-steps-to-improve-the-IT-incident-management-process

IT incident management is a telling measure of how well the IT team functions, but generally goes without praise and recognition. Users often don't notice good work, but poor incident responses create outrage, angry email chains and a lot of questions for IT operations.

Most organizations have room for improvement in their IT incident management process. Focus on three stages: troubleshooting sessions, IT incident reports and the postmortem of substantial issues.

Many of these ideas are covered in the IT Infrastructure Library, and each organization should choose the parts that work for them.

What went wrong?

This is where it all starts -- a help desk ticket, phone call or email alert. Review all your methods of incoming communication in a timely manner. Regardless of how many times you tell people to call if it's urgent, they'll still use email instead. A ticketing system that automatically logs incoming email as an incident -- and that responds with a comment to call for urgent issues -- can improve communication.

Set and communicate expectations as a fundamental part of the initial IT incident response.

Incident management should also take the requestor into account. Try to accommodate their communication preferences, whether they demand a rundown of what's going on or simply expect to hear when an issue is fixed. Diagnose an IT incident with as much research and remote management as possible, so long as it doesn't affect users' work.

Some admins are better at the art of troubleshooting root causes than others. Regardless, everyone should follow an IT incident management process to eliminate possible causes more efficiently.

Envision a game in which players guess a number between 1 and 100. With each guess, they're told to aim higher or lower. This information narrows the focus of the next guess. For example, if a computer can't see network drives, check if it has any network connectivity at all. The answer can help you hone in on an understanding of the issue and how to resolve it.

What have we learned?

An IT staff empowered to learn and collaborate with each other inherently develops a better IT incident management process.

The more you know about IT, the more aware you are of what you don't know. A team that shares knowledge and helps each other out is better equipped to solve IT issues more quickly. Make a knowledge base part of your IT incident management toolkit: Wikis are common, and many teams rely on a collaborative chat tool, such as Slack or Microsoft Teams, but even basic email conversations can spread the word and make it easier to fix an issue next time.

Analyze any incidents that have a significant effect on the business. Post-mortem analysis isn't a witch hunt looking to place blame; it's an investigation to see what failed and why, and to discuss what approaches may have worked better. Report findings, including the measurable effect of the IT incident, how it was remediated and how it should be avoided in the future, to the business side.

Systems and staff are ever-changing, so IT incident analysis must be an ongoing process. Think of it as operational maintenance for the service desk.

How can we handle that better?

An ideal IT incident management process includes a way to analyze how you track issues and where to improve response processes. But a system with all the bells and whistles often isn't feasible because of costs and maintenance overhead.

Frequent meetings are a great time to discuss people's thoughts and experiences with incidents. Meet internally with the IT team, and see if users are willing to participate, as well.

At the least, infrequent surveys can gauge the general impression your user base holds regarding IT responses. Surveys enable users to provide constructive criticism of the IT incident response process.

Feed issues back to other IT areas, and other departments, to improve IT operations overall. If the engineering or development teams aren't aware of bugs, technical issues and end-user experience problems, they can't be expected to do things differently next time.

Next article

IT process improvements lurk in every onerous incident

Adam Fowler, IT operations manager

https://searchitoperations.techtarget.com/tip/IT-process-improvements-lurk-in-every-onerous-incident

IT incident management is one of the best ways to pinpoint which processes are in dire need of improvement. So, the next time something goes wrong, jump on the opportunity to do better.

IT process improvements are born out of bottlenecks, errors and user frustration. The root cause of an IT incident often highlights where the IT team should adjust and mature its operations. This correlation, however, is easier to talk about than act upon. It takes time to investigate and determine how to run IT processes more smoothly, especially when any immediate issues take priority over eventual improvement. The best way to get lasting results from the IT incident management process is to go in with a plan.

First things first

Start with the most common incident types, and look for patterns that will guide where to make IT process improvements. The IT help desk and ticketing system should generate reports based on category and incident type, which helps admins identify the most frequent issues. With or without that data, talk to the IT team that works on the front line to discover the areas that need attention. Admins who address the most obvious issues will free up support staff from repetitive work and enable them to focus on other tasks.

The frequency of password reset calls are among the most common complaints from help desk staff. Commonly, a user will change his or her password and forget the replacement or otherwise need help from IT for a reset. Before you commit to any action, consider the ultimate goal. Reduced help desk calls is a good start, but then think about other IT process improvements you could make to password resets, such as embedding security.

Analyze how the IT operations and help desk teams handle the password reset process. Is the user identity verification system satisfactory organization-wide? Does the help desk change passwords for the users, or do users change the password at next login attempt post-reset? These questions will help you identify ways to optimize the process overall.

Solidify a plan

Help desk staff who observe inefficiencies should notify their managers, who usually appreciate suggestions for IT process improvements. However, think through your ideas fully before you present them. For example, if you let users reset their own passwords, it can reduce help desk calls, but to do this securely requires detailed plans. Will you rely on a range of user authentication questions or trust a one-time text or email message to a third-party provider?

IT process improvements require that the team understand the organization's overarching goals. Think through the problem at hand, and develop a concrete suggestion. For further input, convene with fellow admins and any others likely affected by the change, and use these additional ideas to build a mutually acceptable plan.

The more time you put into planning, the more likely you'll address any issues before you actually implement the change.

User experience is another important factor in IT process improvements. Ensure any process changes won't prompt new issues. For example, if you roll out that new password reset process, will it introduce privacy issues around automated text and email messages to services that staff members use?

Communication around a process change is just as important. Determine who on staff needs to know about it, as well as any required documentation or training. It isn't easy to manage users' expectations and needs during transitions. So, don't rush the decision, and learn what has and hasn't worked in previous changes.

Once you have discussed, decided on and put in place IT process improvements, there's still more work to do. The change could create some user frustration in the short term. So, dig into the help desk tickets, and gather operations and support staff to look at the results. Did the process improvement reduce help desk calls and better the user experience? Did the updated process cause any new issues? These answers are fodder for future IT process tweaks, which is how the service management lifecycle works: continual service improvement.

Incident resolution is also a way to make the combined IT operations and help desk staff more efficient when users raise issues. It could even potentially create new ways for the company to be more profitable.

Next article

IT troubleshooting is as much about users as technology

Adam Fowler, IT operations manager

https://searchitoperations.techtarget.com/blog/Modern-Operations-Apps-Stacks/IT-troubleshooting-is-as-much-about-users-as-technology

An IT troubleshooting rule I've held for a long time is to never trust what the user tells you. In my opinion, it's one of the fundamental rules of IT that will come back to bite you when not followed. When you assume the user knows what they're talking about, you'll end up going down the wrong rabbit hole.

Hours can be wasted troubleshooting a problem that doesn't really exist. Alternatively, asking the right questions at the start can turn a complicated sounding problem into a simple one.

There are also times when the user has actually told you the correct information, but it's too hard to believe.

This is one such story of a problem that could have been resolved quickly if a single assumption wasn't made. It is based on real IT troubleshooting experience.

Missing a crucial step

A normal, unexciting day in IT, and the help desk phone rings, breaking the tapping of keys surrounded by silence. A flustered user is at the other end of the line, desperate to have their issue solved. The problem? They urgently need a file off a USB key, but it's "not working."

The help desk staff member's brain starts ticking over the best troubleshooting steps. "Not working" isn't useful at all and could be one of too many problems, time to do some awesome troubleshooting.

The first step they take is to ask if the USB has worked before. The user doesn't know. "I'm following a set of instructions, and I've done everything it says."

A fair question is then asked of the user: "Could you read out the instructions to me?"

The exasperated end user agrees, but highlights that they need to leave soon to catch a plane. "There are a bunch of steps. Step one is to turn the computer on. Step two is to login with username/password." The help desk person sighs internally at the use, and recording on instructions, of a generic account with its password. That's a fight for another day however, as there's a small fire to put out.

The user continues: "Step three then says to open Windows Explorer. Step four is to grab the USB drive, and step five is to click on E with some dots and a slash."

At this stage, the support person at the help desk thinks that's all reasonable. They can't remote onto the computer because it's an off-network PC at a somewhat secure location, so they'll have to rely on the user.

“Can you see Windows Explorer?” asks the hopeful help desker.

The user quickly responds “Yes, I think so. I can see a computer and a letter under it, C dots.”

The help desk person makes a fair assumption that this is the C drive. “But you don’t see any other drives, like the E drive?”

Getting annoyed, the user responds “No, nothing else. Why isn’t this working?” A question often asked during IT troubleshooting.

“OK, let’s try rebooting the computer. Sometimes things go a bit funny and that can help” the support person offers, unsure of what to try next.

“I really don’t have time for this, but fine.” The now disgruntled user goes about finding the power button, too quickly for the IT pro to intervene for them to shut down the correct way, via the operating system.

A minute later, after some slow key presses and sighs, the user gets back on the phone. “There’s STILL no E, this is ridiculous!”

Running out of troubleshooting options, the IT staffer comes to the conclusion that it’s not something that can be fixed remotely. “I think we’re out of options here, it could be a faulty USB stick, or it could be the USB port on the front of the computer.”

After a few seconds, the user responds “That’s strange that the front of the computer would have anything to do with this, is that where the wireless card is?”

Confused by this response, the help desker asks: "Did you try unplugging and plugging in the USB drive?"

"What do you mean? The instructions don't say that," the user responds.

It dawns on the help desk staffer. "When it said to grab the USB, did you plug it in or just hold it?"

The user responds matter of factly that the USB key was in their hand the entire time. "Isn't it wireless?"

Head in hands, and after a moment's silence, the help desk staffer concludes with "I don't think so. Let's try plugging it in."

As you can see, it's easy with IT troubleshooting to head down a path that makes sense with the information you are given, following reasonable assumptions.

Verify those reasonable assumptions throughout IT troubleshooting steps. Start from the absolute basics and work your way through to the more technical troubleshooting. Various problems – a USB stick that's been forced in the wrong way, a faulty USB stick, a faulty USB port, a driver issue, Group Policy restrictions and a myriad of other root causes – show the exact same symptoms as someone simply not plugging in a USB memory stick at all.

Next article

Log management systems benefit greatly from AI

Stuart Burns, Virtualization and Linux expert

https://searchitoperations.techtarget.com/tip/Log-management-systems-benefit-greatly-from-AI

IT administrators struggle to see the overall picture when they're faced with heaps of operational logs and plenty of false alarms.

With a touch of AI, log management systems enter a new realm of possibility for IT infrastructure management and reporting.

Basic log management tools provide enough functionality to reduce the amount of collected operations data into a common, manageable format, and then into a set of useful reports, alarms and warnings. The classic log parsers are now in an evolve-or-die situation, with a wealth of new log management systems and features that include AI and new methods to view and manage large volumes of data.

Bringing AI performance and intelligence gains into log management systems opens up a range of opportunities to data-rich IT operators. Log management isn't only for IT management and reporting: logs inform inventory management, performance analysis, and security and server reporting.

The addition of AI into security allows almost real-time detection of vulnerabilities and attacks. This can effectively turn log analysis into a near-real-time monitoring system. For example, logging systems paired with security information and event management (SIEM) tools make threat detection and analysis -- as well as prioritizing items that need urgent attention -- easier.

To understand why these tools work so well together, consider the many forces at play: humans handle voluminous data inefficiently, threat landscapes evolve rapidly, disconnected information might not clearly show the incoming threat until it's viewed together, different information is important to different people, and losing data is undesirable.

Traditional log management systems don't deal with these sophisticated threats well. Potential attackers are aware of how log management servers modify the baseline over time. Increasing activity slowly gives potential attackers the opportunity to mask their visibility. Determined attackers will usually find a way, but the right log monitoring tools can decelerate or even discourage them enough to cause a change of target.

However, the combination of SIEM tools and deep learning capabilities makes these tools better able to detect these threats. Standard baseline tracking would likely miss a skilled and determined attacker deploying baseline hiding techniques. The new intelligence built into tools from companies such as LogRhythm and Splunk Enterprise examines what is happening from one or more logs, what value is placed upon the event that creates the entry and what these values mean in the context of security.

AI is also more effective at minimizing trivial entries in log management systems. AI can detect suspicious patterns and deviations from norms, such as malware signatures, unscheduled network scanning, and unusual login patterns and times.

Individual entries aren't always reported, but the log management system's AI allows it to correlate apparently unrelated items and provide more confidence in the generated alarms. An operations professional could easily overlook all of these seemingly insignificant events.

Once the tool collects operational data, it allows the user to drill down into information to see why the alarm is important.

AI has changed the way that information is consumed. The sheer volume of data generated in large-scale IT environments is difficult to consume for humans. Threat categorization and prioritization can become impossible. How can an operator differentiate between the importance of several different, yet critical alarms?

One of the first tenets of threat management is a prompt reaction to mitigate the threats. AI can step in and conduct preprocessing for the security operations center and provide a categorized threat assessment. This enables the operator to delve into the threats.

AI can even make better decisions than humans when provided with enough information. AI is not subject to the same bias as humans. AI makes better decisions faster when given more information at a larger scale. Humans can then determine the final triage of an issue and take the appropriate action.

At the most fundamental level, AI shortens reaction times. This can mean the difference between a data breach and pre-emptive detection. Additionally, it presents a trove of information to security personnel regarding the source of the looming breach.

AI-enhanced log management tools can form part of the arsenal of security detection and response tools in enterprise IT departments.

A range of tools previously unavailable or too slow to generate reports can now offer real-time log monitoring and reporting built using AI. The possibilities are limited only by the user's imagination. As AI integration improves, the range of features and functionality available will, as well.

↘ Next article

Enterprise shops look to AIOps for IT root cause analysis

Beth Pariseau, Senior News Writer

https://searchitoperations.techtarget.com/feature/Enterprise-shops-look-to-AIOps-for-IT-root-cause-analysis

Enterprises have automated IT root cause analysis in their sights as AIOps hype reaches a fever pitch, but some IT vendors are reluctant to jump on the AI bandwagon.

Automated IT root cause analysis is central to many AIOps tools, a category of operations tools that incorporates AI to improve the tool's ability to monitor and manage IT deployments. New Relic's error profiles feature, for example, is meant to narrow down the cause of glitches and speed up IT incident response. Enterprise IT ops pros imagine a day where that response is automated through application development tools and IT service ticket systems. AIOps tool vendors even tout a proactive, rather than reactive, IT monitoring approach, which identifies the root cause of potential problems and stops them before they reach the troubleshooting stage.

But despite customers' demand for such features, some IT monitoring vendors won't adopt AIOps.

LogicMonitor, for example, touts its monitoring tools' ability to pinpoint the causes of errors in the IT stack, but its founder and chief evangelist, Steve

Francis, balks at the suggestion that machine learning can automate IT incident response.

"People tend to define machine learning as anything they don't understand," he said. "Anything we do understand is just statistics. Everyone's saying we need to be talking about this, but I don't see the value of it yet."

LogicMonitor customers beg to differ.

"They need to think a little beyond red light/green light and [rather about] how to build innovation around root cause analysis," said Miten Marvania, founder and COO at Agio, a managed IT and cybersecurity firm in Norman, Okla.

Marvania wants LogicMonitor's tool to understand the real effect of events in the IT stack. For example, if a load balancer has five servers attached and one fails, he wants LogicMonitor to know that four out of those five servers must fail before it's a critical event.

"Right now, it's a binary approach where, if a server is down, it's critical," Marvania said. "They need to assess the real impact of that."

AIOps fans flames of IT root cause analysis debate

The most advanced DevOps shops no longer want IT root cause analysis to be the focus of IT incident response anyway, as infrastructures become ephemeral and applications distributed. But in more traditional enterprise IT shops, automatic root cause analysis of ongoing problems is still the holy grail of AIOps.

"If LogicMonitor had a way to understand the least common denominator of problems, then it could directly tell us: 'It's this network switch that's acting up' or 'These database errors seem to be at the end of the chain of dependencies,'" said Andy Domeier, director of technology operations at SPS Commerce, a communications network for supply chain and logistics businesses based in Minneapolis. "It gives an engineer a lot more context about how to approach that problem."

LogicMonitor's Francis is not convinced, however, that the approaches Marvania and Domeier suggest are viable. Users can configure LogicMonitor with thresholds that indicate how many load-balanced servers can fail without a critical alert being triggered, for example, but he doubts the discovery of such configurations could be automated out of the box.

"That's not stuff we can know absent human knowledge about their application," Francis said. "That's human knowledge and configuration."

As for Domeier's common denominator idea, Francis said LogicMonitor plans to help customers narrow down likely root cause culprits by correlating alerts, but he is skeptical such correlations can be reliably precise.

"I'm not sure that's a legitimate thing to say anyone will have in the short term," Francis said. "We can shorten the time it takes you to look for your root cause. But I don't think we'll ever be able to say, 'This is it.'"

LogicMonitor could be bluffing about the workability of AIOps for IT root cause analysis as a response to heavy marketing messages from its competitors. But those who've seen AI deployed at scale in IT operations have said that Francis has legitimate concerns.

"The blessing and the curse of these things is that they often demo incredibly well," said Ben Sigelman, senior staff software engineer for Google from 2003 to 2012 and co-founder of infrastructure monitoring startup LightStep. "In a controlled environment where you know what the inputs are, you can show things that are almost magical -- which means someone's going to buy it and then you have the issue of making it work in production."

LightStep specializes in monitoring cloud-native microservices infrastructures, but Sigelman doesn't plan to use the AI buzzword to market LightStep either.

"I would rather we talk about the benefits or features of products outright and, below the fold, it can say it's because of AI, statistics processing or machine learning," Sigelman said. "Just saying, 'This works because it's AI' is really glib."

Could crowdsourced AIOps boost root cause analysis?

Some industry watchers wonder if proactive IT monitoring would be more realistic with a broader set of data collected from multiple enterprise customers of the particular tool. Such proactive data analysis is already in use in manufacturing and refinery facilities, where equipment vendors can analyze streams of data from their machines to proactively identify potential failures.

"We'll see groups of like-minded companies allowing customers to subscribe to aggregated feeds of cleaned-up data," said Brad Shimmin, an analyst at Current Analysis. "AI processing against not just your data but everyone's can help make more accurate predictions."

Francis said this might enable proactive monitoring on broad terms, such as detecting whether a cloud service provider's data center or an internet service provider's network connection is down in a particular region.

"That is a solvable case of root cause analysis because you have enough data from enough data points, and it's a relatively simple problem," he said.

The word *relatively* is operative there, Francis added.

"Having been a network guy, I know you can have trace routes that work perfectly well for one device that's going over the same network and another one that totally fails" because of EtherChannel routing behind the scenes, he said. "So even that is not going to be a perfect use case."

That's to say nothing of the obvious potential security and compliance snags of aggregate data for automated IT root cause analysis. New Relic, for example, has held off on such a service because of customer worries about sharing IT monitoring data with other companies.

Beth Pariseau is senior news writer for TechTarget's Data Center and Virtualization Media Group. Write to her at bpariseau@techtarget.com *or follow* @PariseauTT *on Twitter.*

Next article

With AI-based cloud management tools, context is king

Kristin Knapp, Senior Site Editor

https://searchcloudcomputing.techtarget.com/news/450431470/With-AI-based-cloud-management-tools-context-is-king

Administrators who struggle to get deeper insight into cloud infrastructure and application performance have a new ally: artificial intelligence.

Some emerging and legacy IT vendors have infused AI technology into their cloud management tools. While their feature sets -- such as the ability to analyze host performance, optimize costs and set up alerts -- sound similar to those found in more traditional third-party management tools, these AI-based platforms reach a new level of sophistication, providing greater granularity and broader context, according to IT pros.

Travis Perkins PLC, a retail provider for the home improvement and construction markets based in the U.K., uses Dynatrace's AI-based performance monitoring platform for its on-premises and Amazon Web Services (AWS) environments. Rather than focus on higher-level metrics related to host servers or instances, the tool reports more granularly on aspects like Java runtime code and errors, said Abdul Rahman Al-Tayib, e-commerce DevOps team leader at the company. This enables his team to perform faster and more precise root-cause analysis when something goes wrong, and better assess the overall impact any issues might have on the business.

"When it comes down to investigating or looking into specific elements of performance where we have had challenges, rather than having to do the investigation manually, [Dynatrace combines] it all into one report," Al-Tayib said. "So, it tells you, 'This service here failed to fire, and, therefore, it caused this series of events, which was then related back to [a disruption at your] customer.' You can immediately see where the challenge is."

To initiate this root-cause analysis, users install a Dynatrace agent on their host machine to identify the various dependencies between resources and help correlate certain events with any issues that arise, explained Alois Reitbauer, chief technology strategist at the company, based in Waltham, Mass.

"If you have a host that is running out of CPU, and the service running on that host has a response-time problem, [the tool can tell] these are related to each other," Reitbauer said.

More sophisticated anomaly detection, or identifying when an IT service is performing in an abnormal way, is another feature that makes AI-based management tools stand out. To do this, the Dynatrace tool performs auto-baselining -- an automatic process that assesses baseline, or standard, system performance by applying different algorithms for metrics such as response time, failure rate and throughput.

After the tool extrapolates what normal performance looks like, it alerts IT teams to any deviations from that behavior. To avoid being bombarded with alerts, users can further specify performance thresholds, and the tool also applies algorithms to assess criticality.

"If I have two hosts that have infrastructure problems ... I obviously care more about the problem that might be with a checkout function for a cart in an e-commerce application than the other one that maybe does some background batch processing," Reitbauer said. "[That] user context, from an infrastructure case, is of main importance."

This ability for AI-powered cloud management tools to weed out noncritical alerts has been a boon to other users, as well. According to a network and infrastructure capacity planner at a cloud storage provider that uses AWS for its back-end infrastructure, that capability was one of the main reasons his company adopted an AI-based cloud management tool called YotaScale.

The capacity planner, who asked to remain anonymous, conducted evaluations on several third-party cloud management tools, but found that YotaScale allowed him to "suppress a lot of the noise" that can come with those tools' alerts and recommendations.

For example, a company might spin up some AWS instances for a new research and development project, and those instances tend to have low utilization as the project ramps up, he said. Third-party cloud management tools might recommend to right-size those instances or reserve them via an AWS Reserved Instance, but in this case, those suggestions are irrelevant.

"That's not how we would really do things in a bootstrapping scenario, where we are trying to bring up a new test or project, and so I'm going to ignore those," he said.

The benefit of the AI layer in tools such as YotaScale is to analyze IT infrastructure through the lens of various business departments or units,

In this e-guide

How to address IT problems

Best monitoring technologies

Ways to mitigate IT incidents

according to the Menlo Park, Calif., company's CEO, Asim Razzaq. In the example above, that's through the lens of a research and development team.

"We map that enterprise, organizational way of looking at things to the infrastructure," Razzaq said. "And then, within that context, deliver optimization [recommendations] and anomaly detection."

The YotaScale tool achieves this business context via user input. Users adjust certain parameters and dismiss recommendations that don't fit, teaching the tool to detect what's most relevant over time.

AI replacing humans? Not so fast

One overarching benefit of these AI-based cloud management tools is they reduce the need for humans to perform a lot of this analysis on their own. But even the most sophisticated tools won't provide the same level of insight -- at least not yet -- as an IT professional with 20 years of industry experience, said Chris Wilder, analyst at Moor Insights & Strategy.

"These algorithms will be smarter and smarter based on the anomalies they find, but they still don't have the experience a person would," Wilder said. "Data, in my opinion, is not a replacement for human expertise. It's just something to augment it."

These AI capabilities are still in their early phases, agreed Jay Lyman, analyst at 451 Research. But they will eventually become a must-have for infrastructure management tool vendors.

"We'll get to a point before too long where every provider is going to have to have some sort of machine learning and AI in their automation," Lyman said. "I think it will become pretty much a check-box item."

Next article

What are the best log consolidation and storage methods?

Stephen Bigelow, Senior Technology Editor

https://searchservervirtualization.techtarget.com/answer/What-are-the-best-log-consolidation-and-storage-methods

Log movement and storage can pose challenges to IT staff, but some basic guidelines can streamline log consolidation and help avoid potential issues.

Logs are key tools that IT staff can use to troubleshoot problems, maintain security and comply with regulations. Logs are often the foundation of many decisions regarding optimization and regulatory compliance. Log management requires logging analysis tools to harvest the data, but, once the data is captured, IT teams must use proper log consolidation procedures for its storage.

Location matters when storing log data

Accelerate log performance by writing first to local buffers or queues. It's generally not a good practice to write logs directly or synchronously to disk because it's easy for the log activity to disrupt application performance. Instead, provision a buffer or scratchpad storage area and queue up log entries that IT administrators can write to the actual log later.

Store logs outside of the production environment in dedicated storage assets. This log consolidation enables many prospective log users, such as software developers, to access logs without the potential security problems of production access. If possible, it's often helpful to parse log files when ingesting the logs for the first time. Parsed log data makes for much faster searching and filtering.

Become an expert in log analysis

Focus your data capture on business objectives with log filters. Plan log management with effective strategies that take into account storage and security needs. Logging standards help ensure proper procedures are followed.

Regardless of how many logs the IT staff consolidates and stores, it's critical to monitor the storage resources provisioned for log consolidation. Monitoring and alerts can prevent log storage volumes from filling unexpectedly and disrupting logging operations. Generally, IT staff will configure a storage limit or implement a log rotation policy that prevents storage exhaustion and unexpected logging errors.

Protect log files with security and backup policies

Apply thorough security policies during log consolidation. Logs can contain a wealth of sensitive data, so you should use encryption, filtering or scrubbing to guard against data loss or theft. This can be particularly important because logs might be accessible to different groups or users within the business. It's also a good practice to maintain encryption when transferring log data over the network.

Finally, consider the appropriate data protection or backup policy for log files. For example, multiple groups within the business often share logs, so replication can be an important tool for data protection and to ensure that each constituent group has its own copies of the log files. This prevents any one group from altering or deleting log data.

Next article

Ensure container monitoring and control via smart tooling

Clive Longbottom, Co-founder, Service Director

https://searchitoperations.techtarget.com/tip/Ensure-container-monitoring-and-control-via-smart-tooling

Applications -- if you can still call them that -- are so dynamic and malleable when composed of microservices in containers on distributed resources that they hardly resemble monolithic designs. Yet, no matter the architecture, IT operations support must quickly identify and rectify any problems in production.

IT organizations tooled up for monolithic app monitoring and support are poorly equipped for microservices and container monitoring.

Container isolation and portability

For IT support professionals experienced on VMs, containers seem to be a backward step. A VM includes everything that the application needs to run: the OS, the software stack and, sometimes, even the data. With VMs' capability to package functions or services, the admin easily captures and stores known-good working systems. If a change causes problems, the admin can revert to an earlier version, maintaining functionality while the team conducts root cause analysis on the issue.

Containers share certain parts of the underlying OS and only include parts of the overall stack that the app function or microservice in the container requires to run. Containers need the same underlying OS as they get moved around infrastructure; VMs do not.

Containers can include enough of an environment to be highly portable yet still maintain an efficiency advantage over VMs because they do not carry around a complete OS copy. Virtuozzo's system container approach is one way to achieve this middle ground. For example, if a function requires a specific patch level of an OS, that patch level is held within the container, whereas the actual OS itself is still shared by multiple containers.

As such, it is possible to replace many VMs with an equivalent, but much more resource-efficient container. The capability to fail over to a previous container, even against a nonidentical base platform, makes containers a viable and valuable deployment method.

Container monitoring and management tools

IT organizations can keep containers healthy without failing over to previous versions thanks to advanced tooling. Container management tools are maturing and can be tied into evolving DevOps processes.

Open source Kubernetes technology -- and the many commercial variations thereof -- introduces capabilities such as container monitoring; the ability to restart ones that are misbehaving in small ways and to shut down and replace ones that are misbehaving badly; and methods to make the containers available to other containers and users in a clear and easy way. A large group of IT product vendors, including Oracle, IBM, Red Hat with OpenShift, CoreOS via Tectonic, and Canonical (particularly in conjunction with its DevOps tooling, Juju), offer Kubernetes.

Also, consider vendors such as Electric Cloud with its ElectricFlow product and CA, which, after its acquisition of Automic, has a range of tools that enable container monitoring and management with automation. Cloudify also offers strong container orchestration. For container deployments already managed with Mesophere DC/OS and the Apache Mesos open source technology, Mesosphere Marathon is a strong option for container monitoring and control. And for Docker users, swarm mode provides capabilities through scripts and command line-based interactions.

Most public cloud platforms have container orchestration services, and here, Kubernetes is also prominent. Amazon Web Services offers Elastic Container Service and Elastic Container Service for Kubernetes, while Microsoft runs Azure Container Service and Google has its Google Kubernetes Engine.

Monitoring, analysis and remediation

For IT operations support, a container orchestration system should offer a range of capabilities:

- **Health monitoring**: The orchestrator should enable users to check on containers to make sure that they operate within agreed policy limits. This kind of container monitoring data should be reported in an easy-to-digest form, preferably via a GUI.
- **Automation**: Orchestrators should consistently and repeatedly provision containers against different base platforms, as required.
- **Remediation**: Remediation is based in the capacity to stop, restart, replace or otherwise work with containers that are operating outside agreed policy limits.
- **Root cause analysis**: Container monitoring identifies a problem -- so, where exactly is it? Container-based systems hosting microservices can be complex, so tools that quickly and effectively identify where the problem lies are in demand.
- **Integration into other systems**: Containerization is the downstream side of a full DevOps process. Therefore, make sure that the chosen orchestration system can operate within the enterprise's DevOps tool set. An orchestrator should complete feedback loops to other systems, such as help desk/trouble ticketing and developer tools.
- **Housekeeping**: While monitoring container health and compliance, an orchestrator should also identify zombies -- unused containers that are still live. It can either alert administrators for intervention or shut down the containers as necessary.

Containers -- alongside related trends, such as cloud computing and distributed application architectures -- change the way that organizations provision and use their IT platforms, so ensure that the orchestrator suits tomorrow's needs, as well as today's. A container monitoring and management system fit for purpose in the future will provide long-term value.

Next article

Self-healing software stacks guard against outages, downtime

Brian Kirsch, IT Architect, Instructor

https://searchitoperations.techtarget.com/tip/Self-healing-software-stacks-guard-against-outages-downtime

When talking about self-healing software and infrastructure, it's important to know what self-healing isn't.

Failover and high levels of equipment redundancy prevent downtime, but do not equal self-healing infrastructure. While redundancy is a preventative measure to ensure uptime from power systems to servers, hardware generally cannot replace its own failed components -- at least not yet. Nor is self-healing relegated to science fiction realms of artificial intelligence or super-computer-worthy processes. While self-healing software and infrastructure may seem out of reach, the reality is many of today's IT monitoring tools incorporate this ability.

How IT stacks heal

Self-healing software relies on the ability to take corrective or preventive action based on a status check. In a truly self-healing stack, the corrective action is an autonomic function done without human interaction.

Self-healing capability is to virtual resources, software and application components close to what failover and redundancy are to hardware. The key difference is the lack of human intervention. While hardware outages and impacts are noticeable and somewhat easy to correct, a self-healing software stack involves fundamental changes to the application architecture. Self-healing simply wasn't feasible for most IT shops until virtualization and microservices application architectures emerged. Once the application stack is ready for granular management, it becomes possible to add autonomic functions that correct issues on small and large scales. Corrective action can be anything from preventative steps that avoid an incident to reactive actions and corrections based on an incident.

To achieve a self-healing software stack, start at monitoring. Self-healing actions are traditionally based on something that occurs to trigger a response. Reactive action -- steps taken to heal a stack after an outage occurs -- is a lot easier to implement because event-based triggers are well understood in IT tooling. Preventative action is quite a bit harder, because the monitoring tool or suite seeks out indicators that a problem might be coming, and must initiate action before the outage occurs -- if it were going to occur. For example, when a monitoring tool identifies excessive memory usage, the cause might be a memory leak or simply a busy period for the application in question. The less expensive and complex reactive method gives IT operations teams the benefit of exact 20/20 hindsight, rather than the sometimes unclear crystal ball of predictive analytics.

Self-healing software stacks take corrective action to fix the underlying cause of an outage. It might be as simple as an application restart or as complex as having to rebuild or create a new instance or virtual server.

These systems need the flexibility to restart something as small as a service or to execute a task as large as deploying a new server. The self-healing system needs to implement progressively escalating corrective steps. For the reactive self-healing process on software and infrastructure, the system must try one action, then if that does not restore operations, move to a larger action, and so on, until the application comes back online. The operations team must decide how long to wait between each step before moving on to the next.

Invest in visionary self-healing

Preventative self-healing infrastructure and software have a place in your data center, with a bit more work and data. For a monitoring system to effectively predict events, it needs a large amount of historical reference data. The obvious choice would be to save up performance data over time, but accurate predictions rely on a static environment. Every system change, upgrade or even patch can influence the software and infrastructure baseline, affecting the baseline of performance and reactions. The other challenge is to set up systems to mine data for possible patterns, while considering how static the system is from change, something that is difficult at best in IT.

All of this doesn't preclude predictive self-healing approaches. Consider focusing the predictive triggers on a much shorter subset of data. It won't be as effective as with a larger set of historical data, but still offers an advantage over the reactive approach alone.

Any software stack you support that incorporates self-healing at any level should be tested -- one prime example of real-world testing is Netflix's Chaos Monkey application. Chaos Monkey's purpose is to randomly terminate instances in the user's microservices architecture in production. The tool set is available on GitHub. While this might seem more like getting hit by a virus rather than deploying something helpful, it proves an IT operations team's ability to provide continuous service even in unpredictable events. While no one wants to do it, the only true way to know if your self-healing software and infrastructure deployment works is to test it out in production. Otherwise it is simply a theory -- perhaps a wish -- of self-healing ability.

↘ Next article

Predictive IT analytics improves distributed application monitoring

Brian Kirsch, IT Architect, Instructor

https://searchitoperations.techtarget.com/tip/Predictive-IT-analytics-improves-distributed-application-monitoring

How do you comprehensively track an application in one place, when the app is anything but?

Organizations run applications on multiple platforms -- on premises, in one or more clouds or all of the above -- and use highly distributed architectures, deploying code as microservices and in containers. Compared to monolithic apps, distributed applications decrease monitoring visibility, as well as the effectiveness and accuracy of analytics. Organizations must choose the right tools and set up a distributed application monitoring process.

Concurrently with the trend of distributed applications and computing, more predictive IT analytics tools cropped up to combat issues of reactive IT. This predictive technology evaluates current stats, trends and historical data and then uses techniques, such as machine learning and data mining, to make suggestions about future or unknown events. While not foolproof, predictive IT analytics that reduce even a few events over an application lifecycle can yield huge cost savings. Predictive monitoring and analysis don't solve every problem, however, because IT organizations still must decide what they monitor and when.

Distributed problems

Modern applications have shifted away from the monolithic, single-server install design to a distributed approach with several moving pieces that reside on even more moving pieces of IT infrastructure. Your organization's approach to overall monitoring can't have a single focus; it must cover a wide net of resources, such as external storage, networking and compute power.

Predictive analytics on a distributed application gets more complicated still. To figure out what application and infrastructure aspects to monitor, start with the top and bottom of the stack, and connect the pieces in the middle. The top end is about the customer experience with the application.

The client-side aspect of the application is critical to the overall performance assessment but is difficult to apply predictive analytics to. Application performance issues -- such as features that often work but then don't -- can occur seemingly at random and are, therefore, difficult to predict. Nevertheless, customer experience is the overall gauge with which to validate analytics information.

Predictive analytics systems' value diminishes if the information can't be used across multiple systems for a complete picture, without which it's challenging to determine the impact of one system's issues on the entire application stack. The days of the siloed application are over, overtaken by interconnected pieces. For operations, this creates a jigsaw puzzle with plenty of missing pieces.

What to track

It would take a small army of IT operations personnel wielding multiple IT analytics tools across cloud and on-premises systems to monitor distributed applications without any gaps in coverage. Unless you have an unlimited budget for monitoring, it isn't feasible. The key to predictive tools' success is the method used to gather, share and use data, more so than raw machine learning capability or trend recognition.

Correlate the predictive IT analytics to customer experience to ensure that the setup reveals the information IT needs to act upon. With customer experience guiding analytics, the organization's operations team can either prevent or minimize the impact of bugs and failures on customers or, at least, understand what the impact will be and provide workarounds.

Predictive IT analytics limitations

The goal of predictive analytics for distributed application monitoring is to detect and prevent issues, but not every failure or incident is preventable. Analytics cannot occur in real time.

A significant and reasonable concern is the turnaround time of predictive IT analytics. Machine learning and data mining do not go hand in hand with real-time reporting. Both management and engineers must understand that predictive IT analytics systems require time to build sufficient data to process and analyze before the investment pays off with useful insights.

Time can vary from hours to days depending on data volume. Admins can reduce the data set, but this endangers the accuracy of the analysis. The goal for predictive IT analytics is issue prevention for better customer experience and IT resource management. Organizations can rely on other methods for immediate alerting and incident response.

Predictive IT analytics can never address every potential random incident that can affect the application stack. Large-scale power outages, cloud vendors going offline and massive hardware failures are events that don't work into average expectations. But the more data your organization has, the better the outcomes will be. The key to success is to understand the technology's limits and benefits and to work within those parameters.

↘ Next article

A good QA team needs a proper software staging environment for testing

Amy Reichert, Software quality and testing veteran

https://searchsoftwarequality.techtarget.com/tip/A-good-QA-team-needs-a-proper-software-staging-environment-for-testing

As a test team, we generally do the best we can with what we have, and what we have never exactly matches production. What we test on and how the software is configured does not match -- cannot match -- every configuration possibility a customer encounters. What a staging environment provides is an exact replica of production for testing efforts. The hardware, servers, platforms, databases, database triggers and anything else match exactly to your production environment.

Amy Reichert

It goes without saying that neither a company, nor a test team, nor any development organization can guarantee their software is defect-free. I can guarantee you that there is never enough time to cover all test scenarios to ensure code is defect-free, whether using automated or manual test efforts. Therefore, as testers, we have to use risk analysis to determine what areas are more problematic and require more focused testing. Validating that risk analysis requires information about the way the application runs in the actual production environment.

What is a staging environment?

A stage or staging environment is an environment for testing that exactly resembles the production environment. In other words, it's a complete but independent copy of the production environment, including the database. Staging provides a true basis for QA testing because it precisely reproduces what is in production. A well-implemented staging environment makes it possible to define the important standards and test those accurately.

Now, consider a test team that executes all of their tests in a nonproduction environment. For example, in most software development organizations, there are multiple environments for development coding and QA testing on the way to a production release. However, neither the development nor the QA test environment has exactly what the production environment does.

The database is usually not the same -- close perhaps, but not the same. The configurations and platforms may be different. The back-end third-party systems used may differ due to the cost of licensing or installation restrictions. The bottom line is that what they're testing is not exactly the production code because they're testing on nonproduction environment with simulated data. They face the distinct possibility of releasing critical defects to customers because they're not testing in a real-world environment.

Developers and QA pros tweak these environments as needed to simulate testing on production, but simulate does not mean create the same thing. The test team won't see the issues when the environment is not the same because the playing field, so to speak, is not even. Sounds like a place for test escapes and defects to thrive and grow. Maybe there's a better way.

Why is a staging environment critical?

Testing on a staging environment provides a more accurate measure of performance capacity and functional correctness. As Web applications become more mission-critical for customers, it becomes increasingly important to test on environments that exactly mimic production because it's production where customers use your application. Any defect found in production is a miss or an escape. Any defects experienced by customers in production negatively impacts your application's and company's reputation.

More on software testing environments

Learn what software testing challenges to look out for when building and maintaining a staging environment

What is the biggest pitfall in cloud database testing?

In the current business climate, as new applications become more widely available in health care, government infrastructure and financial transactions, it's become more critical that your application doesn't fail. It's critical to not offend, disappoint or annoy your customer base, because with so many products and producers, it's much easier for costumers-- even internal customers -- to switch.

Customers prefer not to be surprised. No one wants their system to go down or to go really, really, really slowly. As workers, we don't want to be negatively impacted at all by software upgrades. As working professionals, we want software upgrades to be seamless, unnoticeable -- a non-event. The only way to truly ensure that your software doesn't interrupt or interfere with your professional users is to test on a staging environment.

As a company, it's tempting to bypass creating a staging environment for preproduction testing. However, when producing mission-critical software of any kind, the staging environment is imperative to ongoing success.

↘ Next article

Mimic production with the right staging environment

Mike Pfeiffer, Chief Technologist

https://searchitoperations.techtarget.com/tip/Mimic-production-with-the-right-staging-environment

DevOps isn't just about moving new code to production more quickly -- it also demands no surprises when that code arrives.

A four-tier architecture that includes separate environments for development, testing, staging and production is a common strategy to test and release software. Application changes and new features are tested and validated as they move through each environment before being deployed into production.

Staging environments allow DevOps teams to test changes on a preproduction infrastructure deployment that is essentially an exact mirror of the production setup. A staging environment lets teams execute load and performance tests to ensure that the application and supporting infrastructure can handle expected requests in production.

Building a staging environment in the public cloud

A key principle of the DevOps approach is to automate everything, including infrastructure. While it's possible to build static staging environments that never change, it's much more feasible to spin up environments on demand and simply pay for what you use and tear it down when finished.

Public cloud providers such as Microsoft Azure and Amazon Web Services (AWS) let users define and version infrastructure in declarative JSON-based templates. Engineers can version these templates in source control, just like developers version application code. A single application programming interface call during an automated release will spin up a completely functional staging environment that is an exact clone of production configurations: servers, networking, storage and more.

Cloud platforms provide a mechanism to export production infrastructure definitions to a reusable template. For example, with Microsoft Azure Resource Manager, the staging manager can deploy all production infrastructure into a resource group. This group can include virtual networks, load balancers, servers, storage and anything else the deployment needs. Once the production environment is defined, they simply export all of the resources in the group to a JSON-based template. Teams can then take this template and automatically deploy the staging environment inside a release pipeline. Once the DevOps team is satisfied with the results, they can release the changes into production, and terminate the staging environment.

The next version of the application kicks off another release, which spins up a new staging environment for testing and validation. AWS has a similar model with the CloudFormation service, which allows teams to define infrastructure as code within a JSON template. Teams build a production environment on AWS and export that definition using the CloudFormer tool. This requires the user to spin up an instance running the CloudFormer tool and select the resources that are running in production. The tool will export a CloudFormation template to a Simple Storage Service bucket. Teams can automate staging resources deployment from this template within a stage in the release pipeline.

Limitations of templated infrastructure

Infrastructure templates are great for deploying a staging environment with the exact same resources and architecture that you're using in production. However, you may need to fine-tune the exported versions of the templates to support various input parameters, customize network settings, and apply your organization's own naming conventions for certain resources.

Additionally, exporting infrastructure to a reusable template does not capture the configuration of IT systems from an operating system perspective. It takes some time to craft bootstrapping scripts that can configure the servers in the staging environment when it's launched. Either bootstrap the servers to install prerequisites and applications using standard shell scripts, or deploy a more robust configuration management system, such as Chef, Puppet or Ansible. You can build custom server images with software and configurations baked in and ready to go.

Cloning application environments

If you're not using infrastructure as a service and instead use platform as a service (PaaS) offerings for your application, Azure and AWS cloning options may help with staging.

For example, if you're running web applications on Azure App Service, you can clone your existing Azure Web Apps. The Azure App Service allows you to run web applications as a managed service. However, the cloning functionality in App Service is in preview at the time of this writing and has a few limitations, as it can't clone network or auto scale settings or database content. It is also restricted to premium-tier app service plans.

AWS has several cloning options depending on your managed service. Amazon Elastic Beanstalk, a PaaS product, allows developers to focus on developing code while the service deploys and monitors the infrastructure on their behalf. Applications run in the context of a Beanstalk environment, which can be cloned to a new environment. You can also clone stacks within AWS OpsWorks, which is a configuration management as a service platform that uses Chef. In both these cases, content in managed databases are not cloned, so the operations manager must come up with their own process to replicate the data.

Because managed services can be deployed through Azure Resource Manager and CloudFormation templates, you may not need to clone an environment. You might simply deploy an App Service with a web app, or a Beanstalk environment within your templates.

Staging environments on private cloud and virtualized infrastructure

Cloning your production environment on premises can be a bit more challenging than deployments on public cloud, but as long as you're working with a virtualized environment, it is possible to automate the process. Most virtualization platforms have command-line tools and APIs that can be used to programmatically provision infrastructure.

An on-premises private cloud may have native support for exporting virtualized environment settings, depending on the platform. If not, automate staging environment deployments with scripts that deploy virtual machines, networks, storage and other configurations in ways to match production resources. Popular configuration management tools interact with common virtualization platforms, such as VMware vSphere and Microsoft Hyper-V. In these cases, you can use Chef, Puppet or another tool to automate the infrastructure, along with operating system configurations for VMs.

Some private cloud frameworks also support infrastructure as code. For example, OpenStack's Heat orchestration tool is similar to AWS CloudFormation. You can use Heat templates to define and version infrastructure and automate the deployment of a staging environment within a release pipeline.

The satisfaction of self-reliance

Self-service is a key benefit for automated deployments in the cloud, virtualized on-premises data center or PaaS deployment. Choose a cloud platform with fine-grained access controls. The identity and access management features from companies such as Microsoft and Amazon allow IT managers to be explicit about who can deploy resources and where they can be deployed. With security controls in place, giving team members the ability to spin up staging environments on their own can accelerate the testing process, which ultimately increases the application's stability and gets changes into production at a much faster pace.

Next article

Five approaches to IT resiliency that don't need new hardware

Stephen Bigelow, Senior Technology Editor

https://searchitoperations.techtarget.com/tip/Five-approaches-to-IT-resiliency-that-dont-need-new-hardware

IT resiliency isn't just a concern for top-tier enterprise-class applications -- organizations developing and deploying software for web-based storefronts, mobile users and a myriad of other tasks are increasingly concerned with how resilient the workload is.

Traditional hardware-based clusters are still a powerful choice to ensure IT resiliency for critical applications in the data center, but these five options bolster application resilience without major hardware investments.

1. Fault-tolerant software platforms

The idea behind clusters is to load balance application traffic across multiple duplicate servers. If a server fails, the other servers take up the load and the workload's operations continue unaffected. One alternative to the traditional server cluster is server fault tolerance or high availability. These models typically duplicate and synchronize VMs across multiple physical servers.

In the fault tolerant mode, duplicate VMs share the load in a hot/hot configuration, like a traditional hardware-based cluster. If one VM fails, the other continues without disruption, though some traffic may drop because load balancing is not as comprehensive as hardware clustering techniques. Thus, the application is tolerant of faults not yet immune to them.

In the high availability mode, the duplicate VM is kept idle and synchronized with the working VM in a hot/warm configuration. If the working VM fails, the standby VM becomes active and takes on the traffic load. There may be a small, usually brief, amount of traffic disruption during the switchover.

Fault-tolerant and high availability deployments use software tools, such as Stratus everRun Enterprise and Vision Solutions Double-Take, capable of creating, synchronizing and failing over to redundant workload instances. The workload's importance to the business dictates whether high availability or the more rigorous fault-tolerance configuration is the right resiliency clustering choice.

2. Redundant cloud architectures

Some enterprise applications are developed and deployed in public clouds, such as Amazon Web Services (AWS), Google Cloud Platform and Microsoft Azure. Public clouds allow rapid and scalable VM and storage provisioning. Now, they also offer IT resiliency features for software developers, operations staff and cloud administrators.

AWS, for example, provides clustering with Auto Scaling services, which allow administrators to create groups of Elastic Compute Cloud (EC2) compute instances. EC2 instances increase or decrease manually, or automatically with changes in workload traffic. AWS' Elastic Load Balancer services distribute traffic on cloud instances.

Organizations need no upfront capital hardware or software platform investment to create workload resilience in a public cloud deployment. The public cloud provider handles all of the hardware and management and the business pays for the compute resources that are actually used -- this amount will vary as EC2 instances and other associated cloud services scale up or down over the course of a billing cycle. To increase IT resiliency against regional disruption, consider cloud providers with international installations, across numerous geopolitical regions.

3. Take a snapshot

Almost every enterprise workload needs some level of operational protection. Not all of them require the real-time protection of clusters, fault tolerance and high availability platforms. Secondary applications or applications in test and development can tolerate some amount of downtime and data loss, and those applications may receive adequate levels of resilience with ordinary VM snapshots.

VMs are basically complete OS, driver, application and data instances running within a server's memory space. A snapshot essentially captures the current state of that memory space or the changes to that memory space since the last snapshot, and saves that content to a disk file such as a *.vmdk or *-delta.vmdk file. If the VM fails, administrators restore the snapshot to restart the VM in a matter of minutes. This usually recovers the application to the point of the last snapshot. There may be some data loss and time to recover, so consider the implications of recovery point objective and recovery time objective before choosing snapshot-based resiliency. If the application can tolerate the potential downtime, this option minimizes hardware commitment by using only one server for the VM.

Major virtualization platforms such as VMware vSphere include powerful snapshot tools that can capture, organize, consolidate, manage and restore VM snapshots.

Resiliency mixologists

There are many potential strategies to bolster modern IT resiliency, and those strategies are rarely exclusive. As software developers and operations professionals collaborate more closely to achieve faster application delivery, it is possible to deploy multiple schemes to match resilience goals with the specific application. One size doesn't fit all.

4. Resilience in application designs

IT resiliency isn't just a deployment or operations issue. Resiliency is also a vital development consideration, and a workload's resilience can be profoundly influenced by the integrity of that application's design and implementation. In simplest terms, application resiliency is responding as elegantly as possible to problems or errors in that application's components, rather than creating nonsensical responses or crashing.

Applications with specific hardware dependencies can pose serious failover or restoration problems. Similar issues arise when workloads depend on specific OSes, drivers, database structures and other software components. Complex software with poor security or inadequate vulnerability testing leaves open many possible attack vectors, which also compromises the application's resiliency.

Proper design techniques and comprehensive testing can't prevent every problem, but do help ensure that versions released to production continue service or fail gracefully when they encounter bugs and other errors. Integrating log and data collection capabilities into the application will help record error conditions and pinpoint performance problems.

5. Containers and microservices in application design

Workload resiliency is increasingly affected by scalability. If the traffic demands outstrip the available compute resources of a workload instance, the workload's performance suffers, or it crashes entirely. Virtual machine clusters and load balancing are well-established means to scale an application; modern application design can capitalize on microservices architecture, deployed in virtualized containers. Instead of monolithic workload designs deployed as VMs, functional components communicate through application programming interfaces to enable the application's functions.

The advantage of container-based microservices is that containers share a common OS, allowing faster scaling with much less compute overhead. The containerized workload can scale in an independent fashion, allowing for clustered and load-balanced containers for each functional area rather than full iterations of the entire application. Functional components are updated and upgraded more quickly than monolithic apps, requiring less regression testing and posing less risk of unintended consequences.

It's a more complex deployment scheme, but the reward for that additional work is felt in IT resiliency as applications scale further than traditional physical or virtual machines while using less total compute hardware.