

MITIGATE OUTAGES AND MAINTAIN PRODUCTIVITY WITH UNIFIED IT ANALYTICS

- Outages are an IT professional's worst nightmare. Discover how to mitigate outages using the three-step framework discussed in our latest e-book.

Introduction

Outages are an IT professional's worst nightmare. The staggering effects of outages can be felt across financial and operational levels, including revenue losses, productivity losses, and reputational damage.

Depending on the size of the organization, financial losses from an outage can vary. Downtime costs between:



\$137 to \$427^[1] per minute
for smaller businesses



\$16,000^[1] per minute
for larger businesses

Productivity losses and reputational damages are not so easy to quantify, as these losses can at times far outweigh direct financial losses.

The key to being proactive against and mitigating outages is to identify indicators of failures early on, and take steps to prevent the outage.

Traditionally, concerned IT leaders have turned towards vertical-specific IT monitoring and management applications to predict or forecast impending outages. However, performing unified analytics facilitates the collation of data from multiple IT monitoring and management applications cutting across IT verticals. This helps you understand the events or changes that preceded an IT outage, allowing you to gain comprehensive insights that can provide 360-degree visibility into outages.

In this e-book, we'll explore a three-step framework that can help you catch early symptoms of failure, as well as how to identify and mitigate outages.

Step 01

Select and track reliable indicators of failure

Outages can bring business operations to a screeching halt.

However, the good news is that outages are rarely a result of an abrupt disruption of services, and mostly follow a specific sequence of events leading up to failure. Tracking these signals or symptoms can help you predict with a fair degree of accuracy when failures and outages are likely to occur.

For instance, an outage due to network congestion could have had several symptoms prior to its manifestation, such as slow internet speed, packet loss, latency, connection time-outs, unresponsive applications, longer round trip time, network jitters, poor network throughput, or longer queue length.

Identifying indicators of failure and tracking the historical performance of these indicators can help you establish normal behavior patterns. Once baselines are established, you'll have an easier time identifying anomalous behavior and setting up a system of alarms and notifications for impending outages. Doing so can go a long way in preventing outages. To help narrow down the indicators, we're going to look into four broad categories of failures resulting in outages:

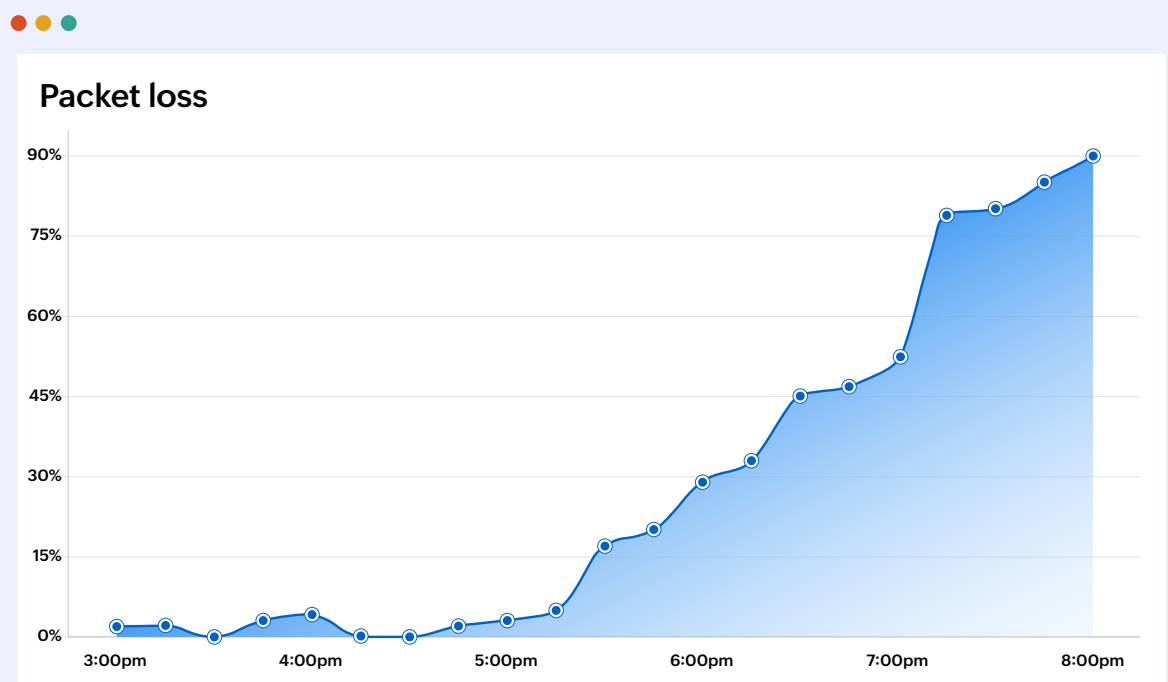
- Outages resulting from component failures
- Outages resulting from capacity constraints
- Outages resulting from human errors
- Outages resulting from natural causes

● Outages resulting from component failures

Hardware and software components often fail, or not function as expected, which leads to outages. Hardware components such as switches, routers, or servers can degrade or become outdated, leading to poor performance and eventual shutdown. Software components can develop bugs or flaws in its code over time or during updates that can lead to malfunctioning of software applications. Tracking availability and performance of hardware and software components, and watching out for changes in performance levels can help predict when these components are likely to fail. Here are a few examples:

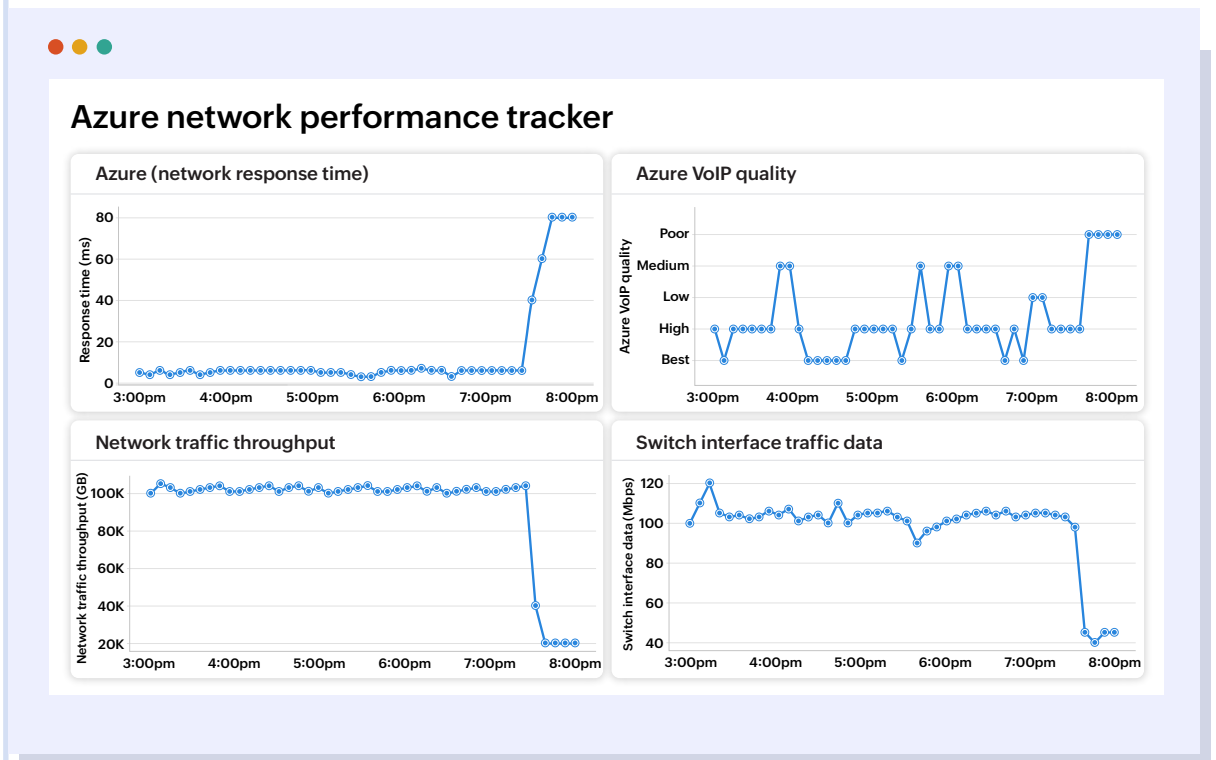
Example A: Consider network outages resulting from failure of one or more network components. Tracking **latency and packet loss** can indicate when network performance has dropped, and serve as an early warning sign that the network is likely to go down.

The sample report below shows the packet loss (in percentage) for a **VoIP^[2]** network within a span of a few hours. Packet loss of 1 to 5% can be acceptable. Anything more than 5% is a cause for concern and can impact uploads, downloads, and the speed at which videos are rendered. Packet loss of 20% and increasing indicates that the network is likely to go down in a few minutes.



While increasing packet loss provides a clue that network performance is degrading, in-depth analysis of **network throughput** or the **volume of traffic** flowing through specific network devices—such as routers, switches, servers, storage devices, and endpoints—related to the concerned network can help identify the network component that is failing.

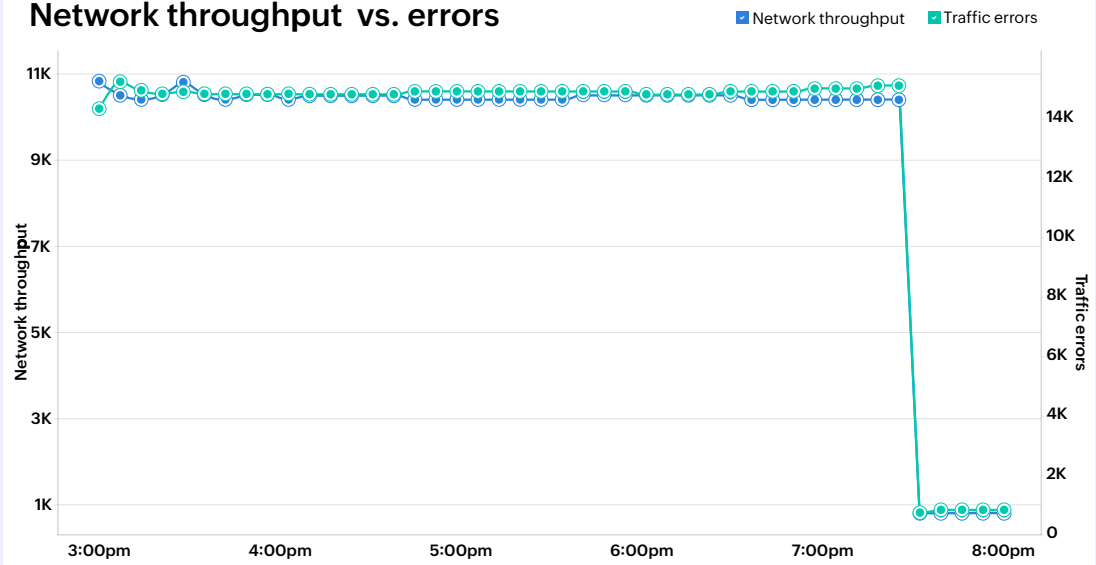
The sample dashboard below provides a real-time picture of traffic passing through network devices such as routers, switches, storage, and endpoint devices.



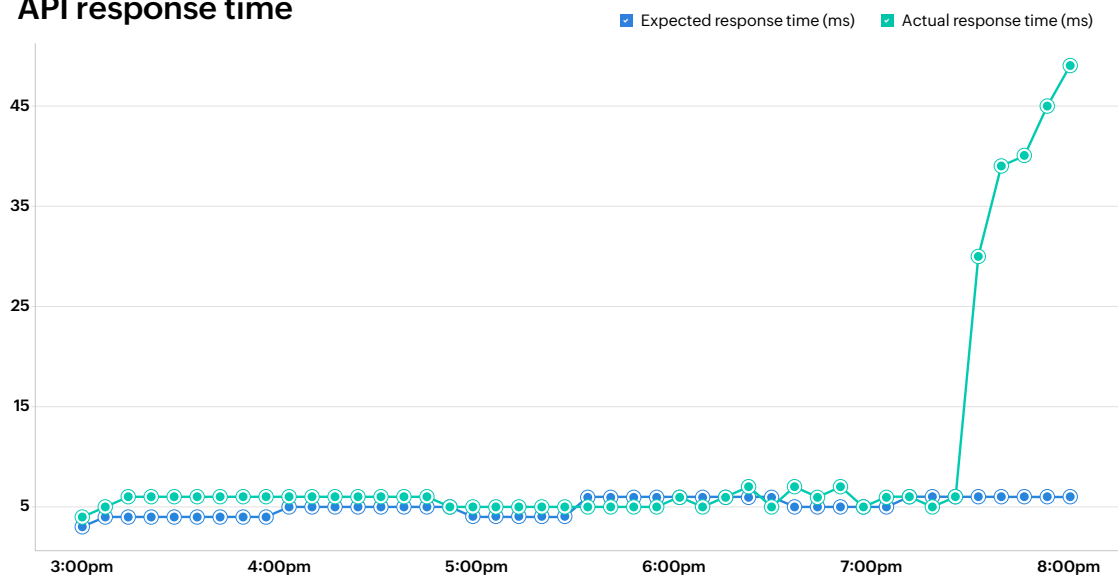
Example B: Consider software-related issues such as API failures. APIs that are essential for communication between two or more applications can fail due to a multitude of reasons, ranging from faulty upgrade patches to unintended configuration changes that are incompatible with the API. Tracking the number of **traffic errors** and the **time taken to get responses** by the target from the client endpoints can reveal when APIs are likely to fail.



Network throughput vs. errors



API response time



API calls taking a protracted time to respond indicates issues with the API call that can impact the performance of services that rely on the API responses.

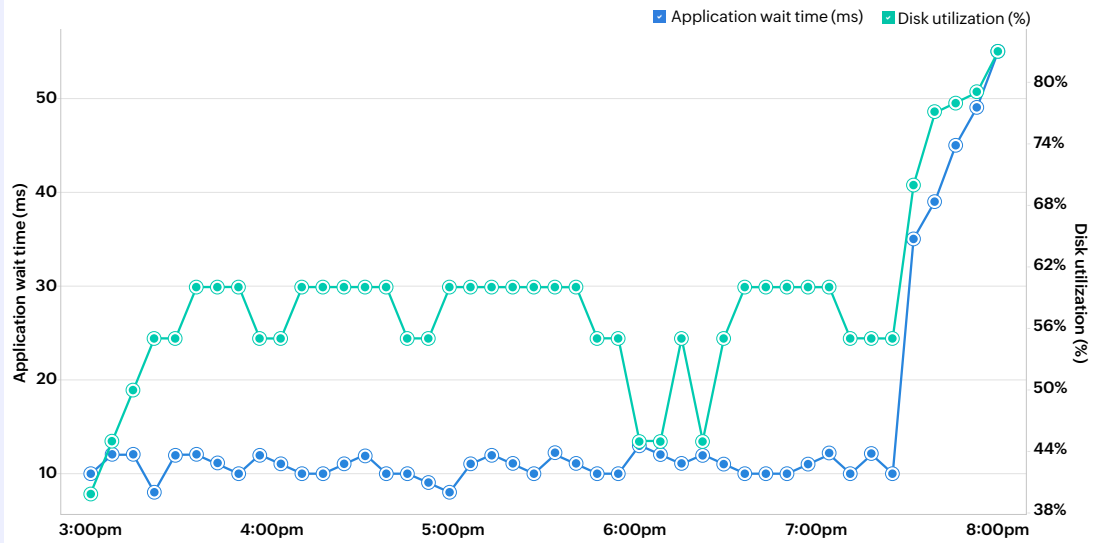
Tracking critical availability and performance metrics such as response time, network traffic, latency, and packet loss are useful to identify hardware or software failures that might result in an outage.

Outages resulting from capacity constraints

When the demand for a resource exceeds the load it can handle, it results in capacity constraints. Further increasing demand for that resource might slow down its performance, and eventually shut down the resource due to overload. A few examples of capacity constraints are servers running out of CPU or storage capacity, networks running out of bandwidth, or applications that have too few threads to process in parallel. The best metrics to track capacity constraints are metrics related to usage. Tracking usage metrics of resources against their available capacity can reveal when resources are heading towards an overload. Here are a few examples:

Example A: Application slowness or unavailability due to capacity constraints can sometimes be traced back its disk utilization. The report here illustrates that as disk utilization increases and nears peak utilization, application wait time increases.

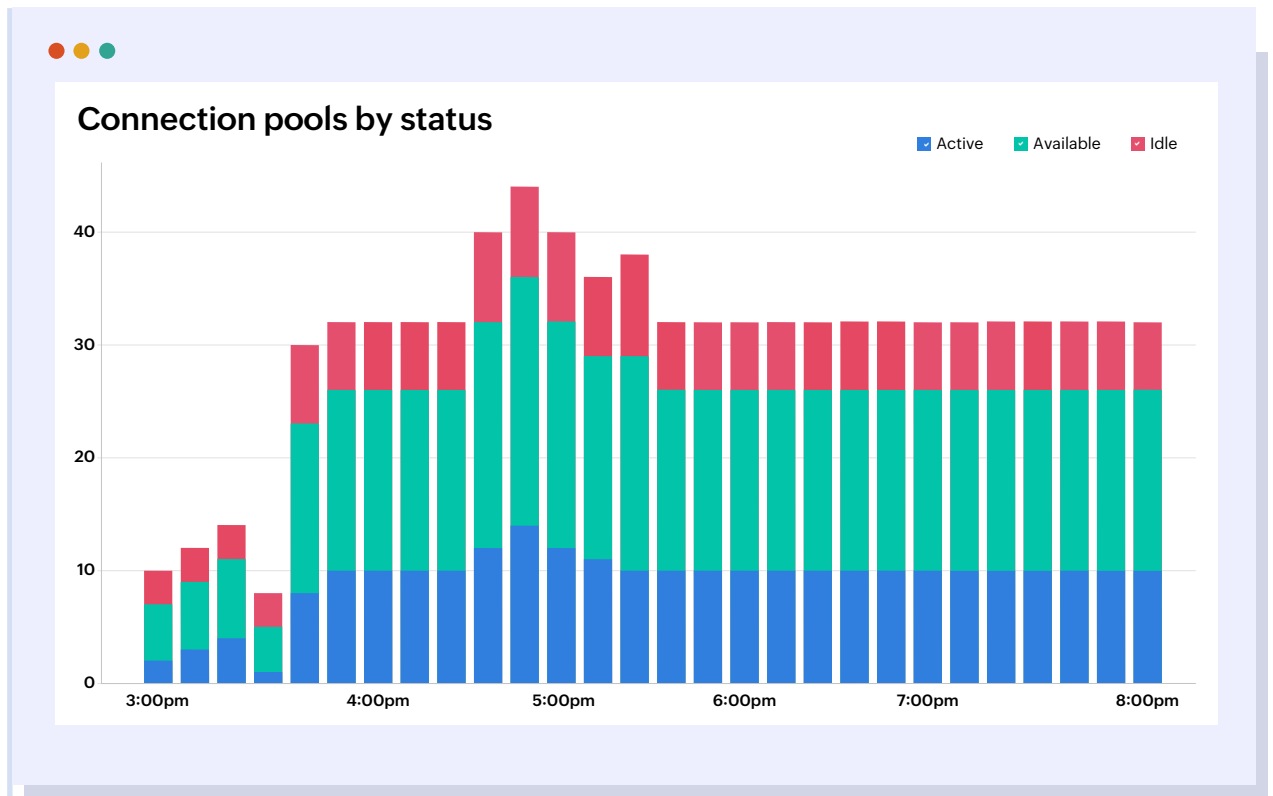
Application wait time vs. Disk utilization



When disk utilization increases beyond its peak, the system will not be able to load additional data or support the application any longer, and the application becomes unavailable.

Example B: Another example of capacity constraints is insufficient connection pools creating a deadlock, rendering applications unavailable to users. When the size of the connection pool is too small, latency creeps in and application wait time increases, eventually leading to unavailability.

Here's a report that shows the number of connection pools for a Tomcat server by their status.



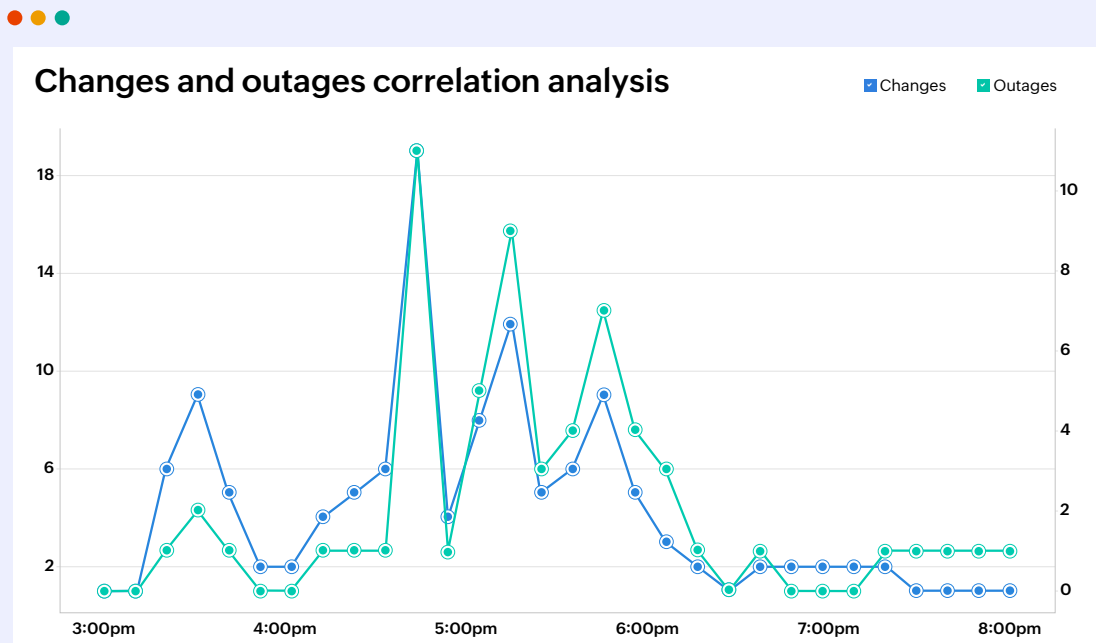
Tracking usage metrics like the ones discussed above are reliable indicators of impending failures from capacity constraints.

Outages resulting from manual errors

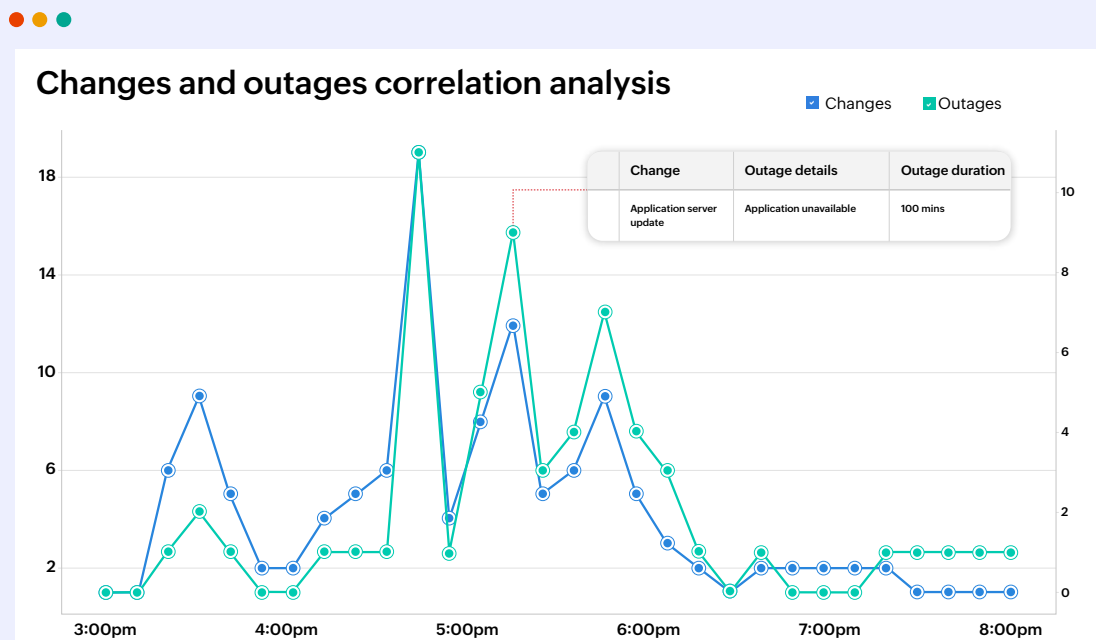
Outages due to human errors—such as implementing unplanned changes, installing untested updates, or changing network configurations without proper planning—can profoundly impact availability of networks and systems. In **January 2023**^[3], Microsoft's cloud services, including Azure, Teams, and Outlook, experienced a global disruption due to a WAN update. This outage affected users worldwide and caused significant disruptions to productivity and communication. Exactly one month later, Amazon AWS suffered a major outage that impacted several websites and applications such as Netflix, Spotify, and Reddit. Around the same time, Google Cloud Platform (GCP) also experienced a partial outage that affected some of its products, including Gmail, YouTube, and Drive. The source of both these outages were traced back to a network configuration error introduced during a routine maintenance.

Considering that human errors are a result of changes implemented or introduced, tracking change volume and correlating failure with changes implemented can serve as reliable indicators of change-related outages.

Here's a report that compares the trend of outages in the past two years and the volume of changes implemented.



Drilling down into one of the outages, it's evident that a strong correlation exists between a change implemented (an application server update) and an outage that lasted 1.5 hours.



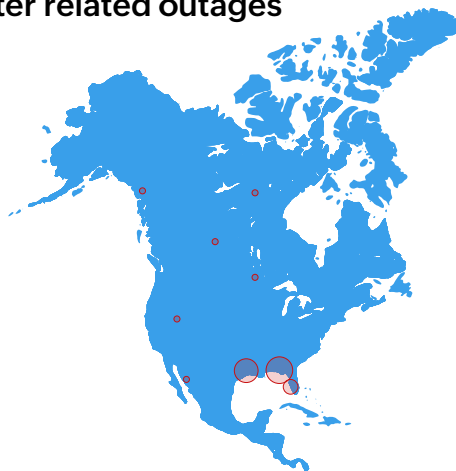
Outage resulting from natural causes

Even when IT departments are run like a tight ship, outages can still occur. Causes that are beyond human control such as power outages, inclement weather, or natural disasters can greatly impact business continuity. The best defense—and possibly, the only defense against natural causes that organizations can adopt—is to understand patterns and prepare. For instance, analyzing the historical outage pattern of an organization can reveal business units or offices that are prone to natural disaster-related outages.

Here's a map chart that illustrates the history of outages for a sample organization. It's clear that Florida has witnessed the highest number of outages in the past. It's common knowledge that coastal regions, particularly along the Gulf of Mexico, are prone to hurricanes and floods. This information can be used to proactively build fail-safes for these regions when natural disasters do occur.



Natural disaster related outages



Natural causes are one of the major reasons for unpredictable outages. While there's no stopping mother nature, there are always steps you can take to protect your servers, and ensure your services are always available.

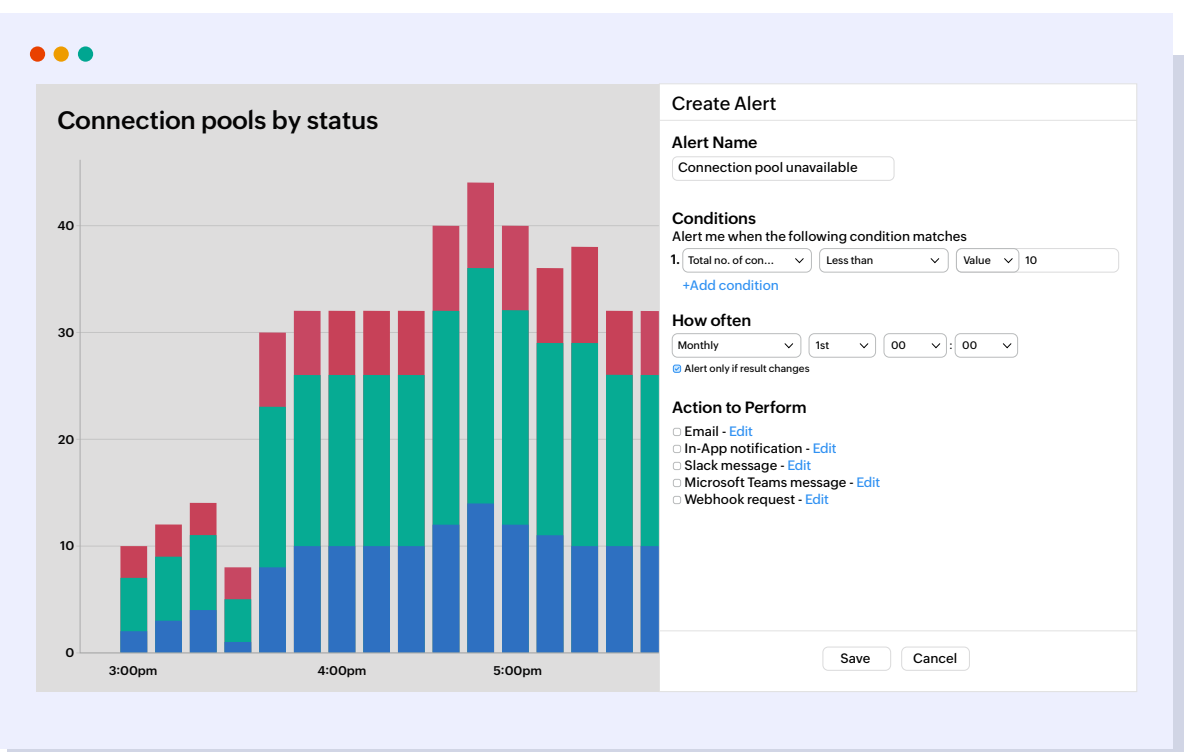
- Strategically map out disaster recovery plans to ensure business continuity. First, analyze the types of disaster that pose the greatest risk to business continuity in specific regions. Then, audit all the IT resources on your networks and servers to understand capacity and usage requirements. Then create plans on how much of data storage, network, and other infrastructure would be needed to continue operations should a disaster strike.
- Set up disaster recovery sites and ensure data is backed up periodically in these sites/servers. Having a backup ensures you can quickly switch your services from your primary servers to your back-up servers almost instantly, while backing-up data ensures that the data available is recent and updated for your employees and end users to operate. Opt for cloud backups and back up data based on geography and operations for seamless disaster recovery.
- Create specific roles and responsibilities for people in your IT team during disasters. Having dedicated resources to tackle special IT requirements during disasters instills order and clarity during chaos.
- Do a dry run once you've planned your infrastructure and the people to handle disasters. Do either a full-recovery test or a simulation test to see if your plan is accurate and holds up your business when needed.

Step 02

Set up rule-based alerts to catch transgressions

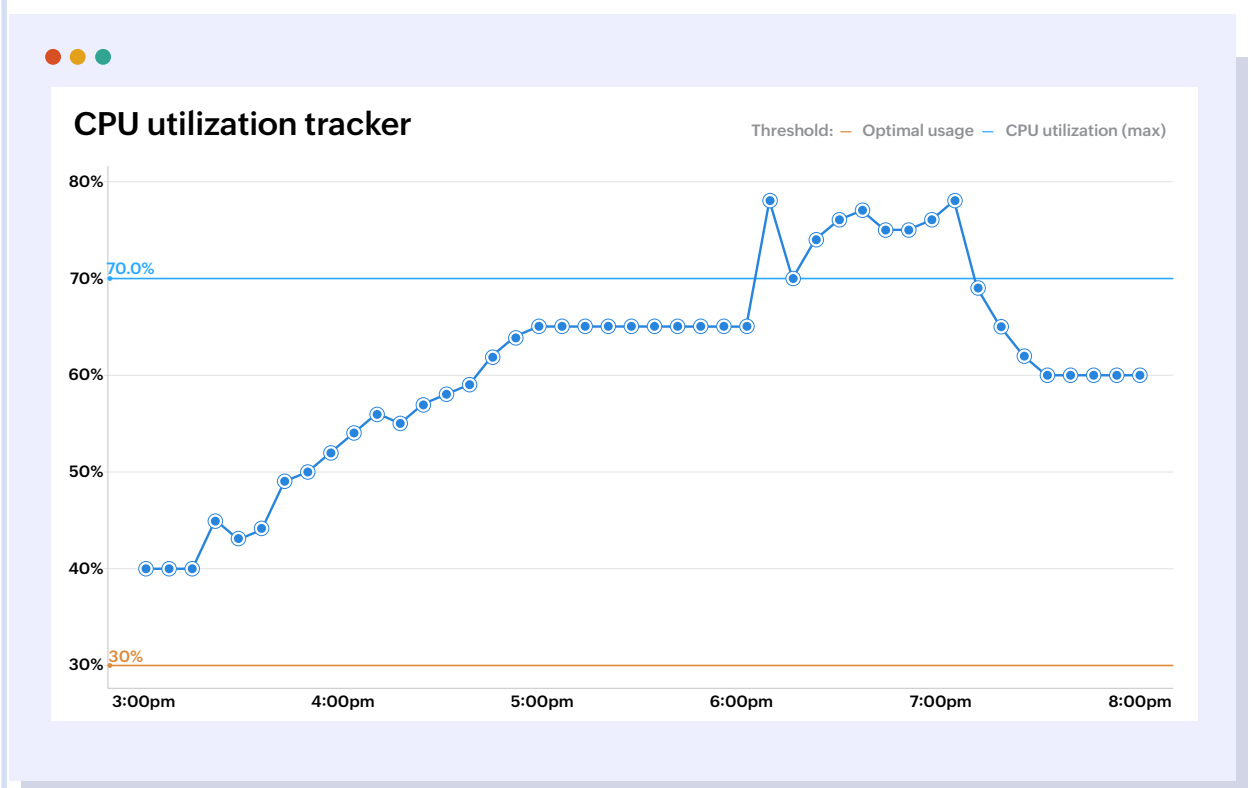
Determining the right metrics and critical indicators is the crucial step in mitigating outages. The rest—that is, setting up thresholds, and monitoring and adjusting thresholds—is easy to achieve once the indicators are decided upon.

In one of our earlier examples, we explored connection pools unavailability. Thresholds can be set up for the minimum and maximum number of connection pools needed for seamless application availability.



Once the number of connection pools needed falls below permissible limits (that is the threshold created), concerned IT heads can be alerted so they can increase the number of connections available.

Alternately, thresholds can be set up to track baseline and maximum performance as well. The report below shows the CPU usage for an Azure instance. Base levels are set at 30%, while max levels are set at 70%.



Tacking disk utilization using base and max levels takes predicting outages to the next level by providing you a window to evaluate and plan disk utilization, with a possibility of avoiding a situation wherein you have high disk utilization. Identifying and understanding occasions where disk utilization drops and peaks will give you enough agency to balance disk capacity requirements between usage and requirements.

Rule-based thresholds can also be set up to catch expiry and end of life for software assets such as SSL certificates or licenses. A lack of a proper alerting system for license and certificate expiry has been known to shut down several major players in the past. **Microsoft**^[4] experienced an embarrassing outage back in 2020 due to an expired SSL certificate. A year later, **Epic Games**^[5]—maker of Fortnite, Rocket League, and Houseparty—experienced a massive outage due to expired SSL certificates. The average annual cost of outages due to certificate expiry is around **\$11.1 million and rising**^[6].

SSL certificate expiry tracker

	SSL certificate name	Vendor	Purchase date	Expiry date	Days to expiry
1.	Aurib-45	Network solutions	03/05/2023	03/04/2024	-255
2.	Auxim-4	Namecheap	04/03/2023	04/02/2024	-284
3.	Comodo-privy 2	Comodo cybersecurity	01/02/2022	01/02/2023	171
4.	DCcentral-45	GeoTrust	01/07/2022	01/07/2023	166
5.	Digital-45	The SSL store	05/07/2023	05/06/2024	-318
6.	GNC-4	GoDaddy Inc	05/07/2022	05/07/2023	46
7.	Jake-43	Entrust	01/09/2022	01/09/2023	164
8.	LE-45	Let's Encrypt	04/06/2023	04/05/2024	-287
9.	Panama-20	GlobalSign	03/07/2022	03/07/2023	107
10.	RM-45	Gen Digital	03/06/2023	03/05/2024	-256
11.	Thawte-40	Thawte	04/07/2022	04/07/2023	76
12.	Webpage-19	DigigCert	06/01/2023	31/05/2024	-343

Create Alert

Alert Name
SSL certificate expiry tracker

Alert based on ☒ Grand summary ☐ Subtotal ☐ Data

Conditions
Alert me when the following condition matches
1. Days of expiry Equal to Value 30
[+Add condition](#)

How often
Monthly 1st 00 00
☒ Alert only if result changes

Action to Perform
☐ Email - [Edit](#)
☐ In-App notification - [Edit](#)
☐ Slack message - [Edit](#)
☐ Microsoft Teams message - [Edit](#)
☐ Webhook request - [Edit](#)

[Save](#) [Cancel](#)

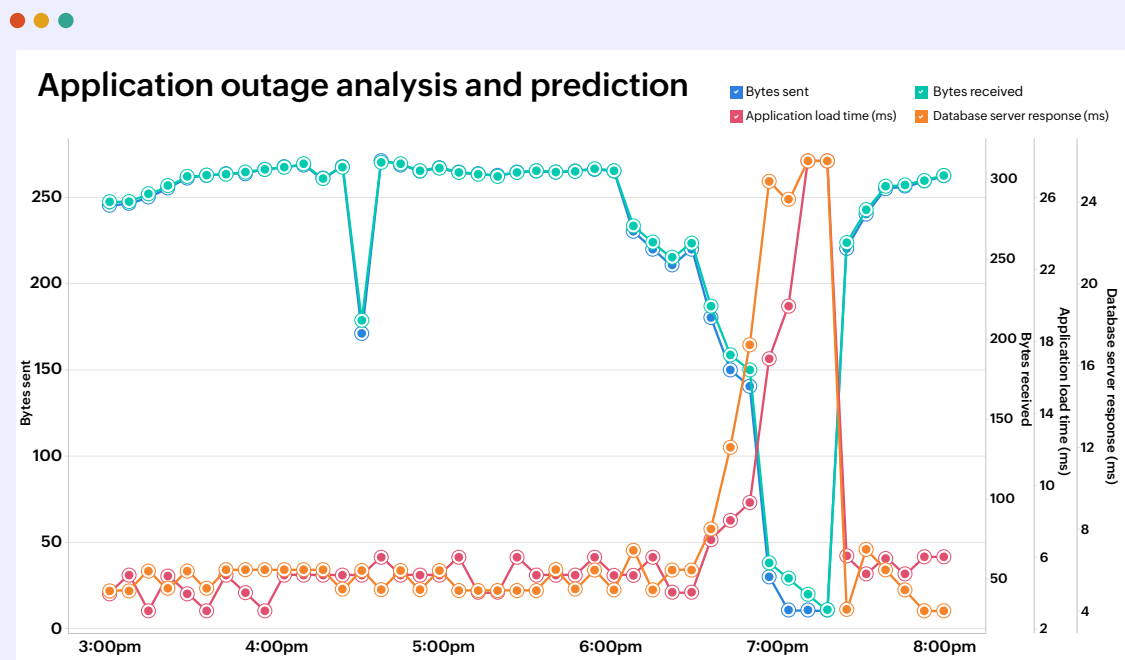
Tracking when certificates expire or licenses expire is important to catch transgressions before they occur. This buys you just enough time and insight to remedy issues and prevent outages.

Step 03

Unify failure indicators

Deconstructing an outage would reveal that failure resulted due to a change in normal functioning of IT systems and components. Identifying these points of failures using indicators, and setting up threshold-based alerting can help you catch these issues as they're progressing towards an outage. However, these thresholds are likely to change as operational processes and the organization matures. Continuously analyzing data and updating these thresholds will ensure that your system stays functional and effective in the long run.

Although setting and adjusting thresholds then watching out for changes helps you catch outage triggers or outage indicators early on, these indicators can also be clustered to catch multi-point failures by unifying information from multiple indicators together. For instance, application failures needn't always be a result of single point failures. It can also be a result of multi-point failures such as network devices, application servers, and database servers; the three baseline components that need to work efficiently in order to allow users to access applications seamlessly. An application can render unresponsive to the end user even if one of these components fails or doesn't function efficiently. Analytics allows users to look at all the underlying components together to allow application teams get a holistic view of application health.



Such overlaying or unification of information enables effortless error logging and alerting, making it easier to notice when failures occur in real time.

Conclusion

While vertical-specific IT monitoring and management applications can provide a fair idea of outages and help forecast them, unifying all IT data relies on real-time indicators and thresholds to predict an outage, using data from a cluster of monitoring and management apps. This increases the accuracy and reliability of outage predictions by providing comprehensive, 360-degree visibility into IT data.

In this e-book, we've applied the three-step outage prediction framework and applied it to a few use cases to understand how to predict, prepare for, and prevent an IT outage.

For more information on how to leverage unified analytics for preventing outages, **check out some of our other resources.**

About

ManageEngine Analytics Plus is a self-service, AI-driven IT analytics solution that helps organizations implement complex initiatives that address requirements of expanding businesses. Available on-premises and on the cloud, Analytics Plus visualizes IT data from several applications and integrates out-of-the-box with several popular IT applications such as ManageEngine ServiceDesk Plus, Jira, Service Now, Zendesk, and ManageEngine Endpoint Central. Analytics Plus features an AI-powered analytics assistant that responds to voice and text prompts to provide meaningful visualizations. This eliminates the need for a data analyst to aid help desk managers and reduces report building time while enabling organizations to make faster, data-driven decisions.

Kickstart your IT analytics journey with a free trial of Analytics Plus.

Want to learn more about the product before giving it a try?

Sign up for a free, virtual tour with one of our solution experts.

280K
customers
across the world

90+
products
and free tools

190+
countries
served

20+
years of IT
management experience

Reference

1. <https://www.pingdom.com/outages/average-cost-of-downtime-per-industry/>
2. <https://www.ir.com/guides/what-is-network-packet-loss>
3. <https://www.linkedin.com/pulse/recent-cloud-platform-outages-2023-pankaj-kumar-mandal/>
4. <https://informationsecuritybuzz.com/experts-reaction-on-microsoft-teams-suffers-major-worldwide-outage-due-to-expire-certificate/>
5. <https://www.keyfactor.com/blog/the-enemy-of-uptime-an-expired-ssl-certificate/>
6. <https://devops.com/5-ways-to-prevent-an-outage/>



© ManageEngine, a division of Zoho Corporation