ManageEngine
Site24x7

How Site24x7 log management can help

optimize your organization's digital infrastructure

What are we going to cover?

Introduction

Meeting customer expectations on performance and reliability is crucial for the success of any business. In today's digital age, where customers have high expectations for seamless and efficient online experiences, it is essential to ensure that your website or application is running smoothly.

If you consistently receive complaints about slow page loading times and frequent crashes, it is crucial to address these issues promptly. One effective way to do this is by analyzing your system's logs. These logs contain valuable information about your system's functioning, including user activity, server performance, and errors that may be impacting your website or application's performance.

By effectively managing and analyzing your application logs, you can identify and address performance issues, ensuring optimal application reliability and user satisfaction. This is where investing in a centralized log management tool becomes crucial. This tool allows you to collect, store, and analyze all your application logs in one centralized location, making it easier to monitor and troubleshoot any issues that may arise and reducing the mean time to repair (MTTR).

A centralized log management tool also provides observability, giving you a comprehensive view of your application's performance and user behavior trends within your application. This makes it easier to address potential issues before they impact your customers.

This white paper offers insights into the importance of a centralized log management tool and how it can help you easily monitor, troubleshoot, and achieve observability for your application.

The purpose and significance of logs

Logs are vital in IT environments for their significance across key domains:

- **Troubleshooting and debugging:** Contextual correlation across multiple sources to identify errors, warnings, and exceptions to find the root cause of the issue.
- **Performance monitoring and optimization:** Optimizing system performance and pinpointing potential bottlenecks or areas for enhancement.
- **Root cause analysis:** Identifying underlying issues causing failures by correlating events across various log sources, thereby reducing MTTR.
- **Centralized log collection:** Consolidating data from multiple sources into a single location for simplified analysis.

The three most common types of logs are application logs, system logs, and security logs. **Application logs** capture events, exceptions, and warnings from applications. They are frequently used by developers and DevOps teams to identify and troubleshoot issues both during the development and post-release phases. **System logs** monitor a computer system's operations to ensure its health. Site reliability engineers (SREs) and ITOps teams rely heavily on system logs to monitor system health, diagnose issues, and ensure the reliability and availability of services. **Security logs** track security-related events such as login attempts, password changes, and authentication errors to oversee user activities and system access. SecOps teams utilize logs to detect and investigate security incidents, such as unauthorized access attempts and data breaches.

Challenges and solutions

Logs are unstructured and may contain sensitive data. Every application writes logs in different formats, including single-line, multi-line, or any unparsed format. Logs are integral to system observability, and parsing and extracting meaningful data from logs is crucial for detecting issues.

Yet logs are always huge. Excessive, unnecessary log data can inflate log storage requirements and expenses. And, logs are everywhere, whether it's in the cloud, on-premises, or hybrid environments. The complexity of modern distributed systems, like cloud services, Kubernetes, Docker containers, microservices, and network devices, has made log management harder with their fragmented data silos. Examining logs separately from various data silos doesn't provide a comprehensive view of the underlying issue.

Aggregating logs from multiple sources into the same system and correlating them is essential to understanding application behavior properly and tracing issues. Managing the growing volume of logs in distributed and dynamically scaling cloud environments while ensuring performance and reliability makes scalability a critical concern and a significant challenge.

How can Site24x7 help you?

The log management solution from Site24x7, AppLogs, offers a centralized logging architecture for collecting, consolidating, and analyzing structured and unstructured data from various cloud, on-premises, and hybrid services, servers, networks, and applications—all in one central location.

Let's explore how this log management solution improves storage, analysis, and monitoring to boost system efficiency.

- Parse unstructured logs for effective data interpretation
- Interpret structured logs for targeted investigation
- Log forwarding from multiple sources.
- Out-of-the box support for 100+ predefined log formats
- Accommodate diverse log formats with multiple log pattern support
- Accelerate the search process by analyzing fields with derived field support
- Hashing and masking sensitive data for enhanced security and compliance

- Filter unwanted logs for improved log management
- Simplify debugging with query language
- Advanced search options for optimized troubleshooting
- Proactive log alerting for timely action
- Log dashboards for actionable insights
- Streamline reporting with scheduled reports and exports
- Affordable log management

Parse unstructured logs for effective data interpretation

Parsing involves stripping away unnecessary elements and organizing unstructured data into a structured format for efficient storage and analysis.

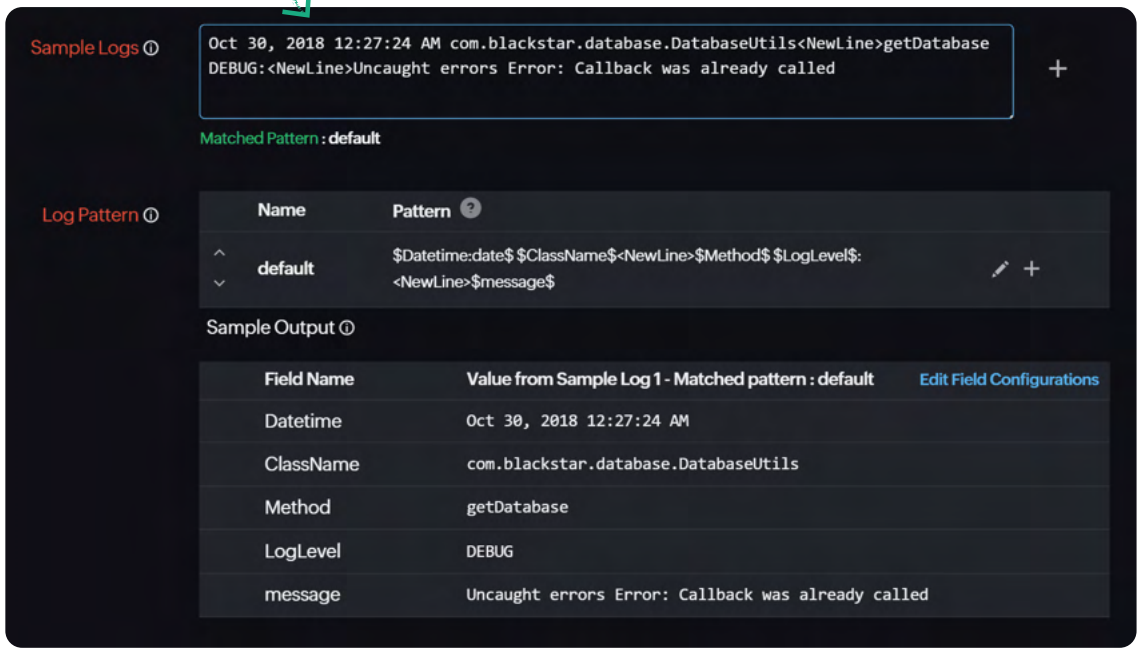
Take this log line:

```
Oct 30, 2018 12:27:24 AM com.blackstar.database.DatabaseUtils
getDatabase DEBUG:
Uncaught errors Error: Callback was already called
```

The log line is part of a multi-line log that provides valuable information for developers and admins to diagnose and troubleshoot issues related to asynchronous database operations. It helps identify instances where callback functions are being invoked multiple times, potentially causing unintended behavior or application instability.

This is one such example of a log line. However, logs may come in a non-standard or custom format. By defining a log pattern, Site24x7 can extract relevant data from each log entry, providing flexibility in parsing various log formats and enabling accurate indexing of the log data. This structured approach to logging enhances readability, facilitates automated processing, and enables integration with logging and monitoring systems for better observability and troubleshooting. Below is the log pattern defined for the aforementioned log line:

```
$Datetime:date$ $ClassName$
<NewLine>$Method$ $LogLevel$:
<NewLine>$message$
```



You can find a simple and multi-line log and the corresponding pattern below:

Type	Sample log format	Sample log pattern
Simple logs	100.10.100.100 - - [07/Jun/2017:19:53:11 +0530] "GET /test.txt HTTP/1.1" 200 12 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36"	$RemoteHost$ $RemoteLogName$ $RemoteUser$ [$DateTime:date$] "$Method$ $RequestURI$ $Protocol$" $Status:number$ $ResponseSize:number$ "$Referer$" "$UserAgent$"
Multi-line logs	Oct 30, 2018 12:27:24 AM com.blackstar.database.DatabaseUtils getDatabase DEBUG: Uncaught errors Error: Callback was already called	$Datetime:date$ $ClassName$ <NewLine>$Method$ $LogLevel$: <NewLine>$message$

Interpret structured logs for targeted investigation

Site24x7 is a versatile platform that can handle various log formats, including JSON, key value, and XML. This makes it easy to monitor and analyze logs from different applications. You also have the option to create and store your own log parsers to collect your complex log formats through custom integration. This allows you to tailor the log data to your unique requirements and easily integrate it into Site24x7.

Structured logs typically have a predefined format or schema, like JSON or XML, which makes parsing relatively straightforward compared to unstructured logs. However, parsing structured logs may still involve tasks such as extracting timestamp information, identifying relevant fields, and converting data types for analysis. You can find a JSON, key value, and XML logs, and the corresponding pattern below:

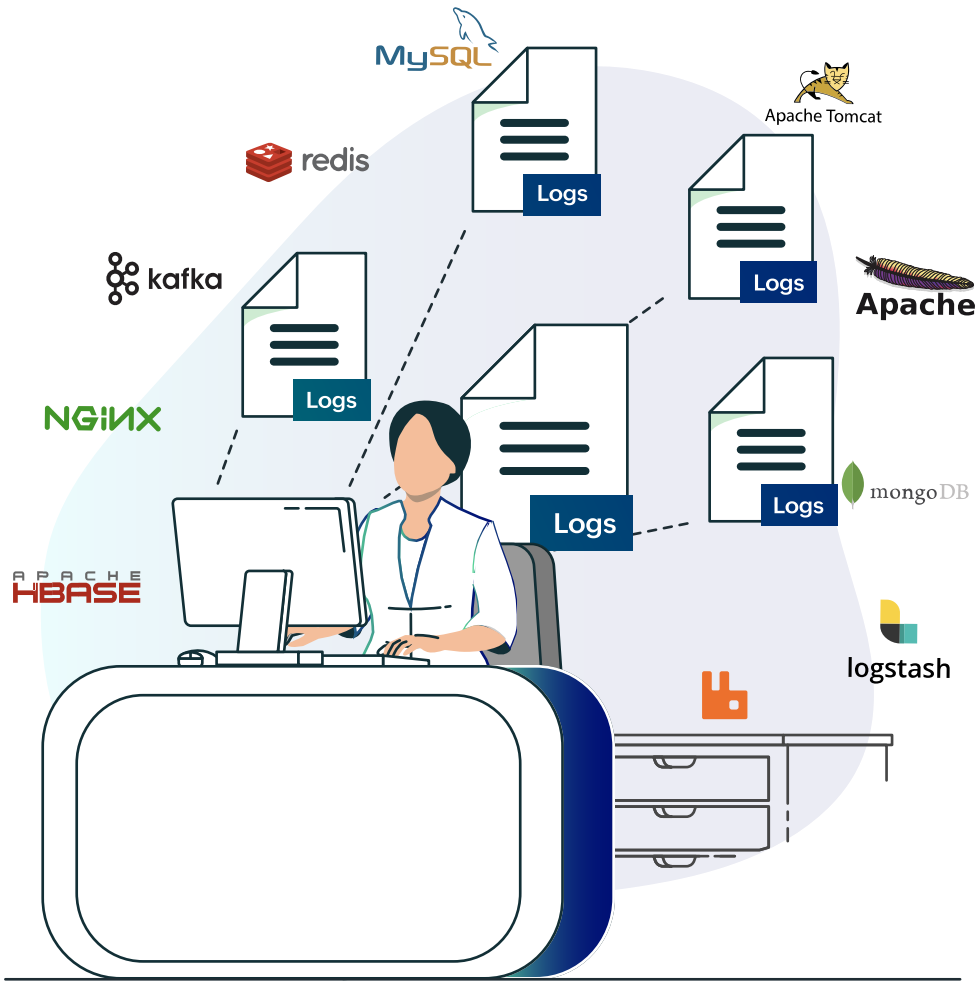
Type	Sample log format	Sample log pattern
JSON logs	{"machineTimeUTC":"2019-01-01T09:27:11.620Z","id":"661dc0ae-404b-4f0d-9579-ceca6e20f22c","callCenterName":"DEV-Station", "name":"A", "value":100,"stack":"dev","machineTime":"2019-01-01T11:27:11.620+02:00"}	json $stack$ $machineTimeUTC:date:yyyy-MM-dd'T'HH:mm:ss.SSS'Z'$ $callCenterName$ $name$ id $machineTime$ $value:number$

Key value logs	date="2014-06-18 11:57:46,719" at=info method=POST path="/en/admin /post/1/edit?k0=v0&k1=v1" host= helloworld-symfony.herokuapp.com request_id=e8843b25-3587-4229 -a430-c93360a0e89f fwd="121.24.53.11" dyno=web.1 connect=1ms service=243ms status=302 bytes=559 protocol=https	keyvalue $date:date:yyyy-MM-dd HH: mm:ss,SSS$ at $method$ $path$ $host$ $request_id as requestid$ fwd $dyno$ $connect$ $service$ $status: number$ $bytes:number$ $protocol$
XML logs	<Log><Time>2022-04-02T19:07:37. 5111809-04:00</Time><Task>System Backup(1)</Task><Operation>Partition Full Backup</Operation><Result-Code>0</ResultCode><Result>Success </Result><Detail>The operation has been completed successfully.</Detail></Log>	xml $Task$ $Time:date:yyyy-MM-dd 'T'HH:mm:ss.SSSX$ $Operation$ $ResultCode:number$ $Detail$ $Result$

Log forwarding from multiple sources

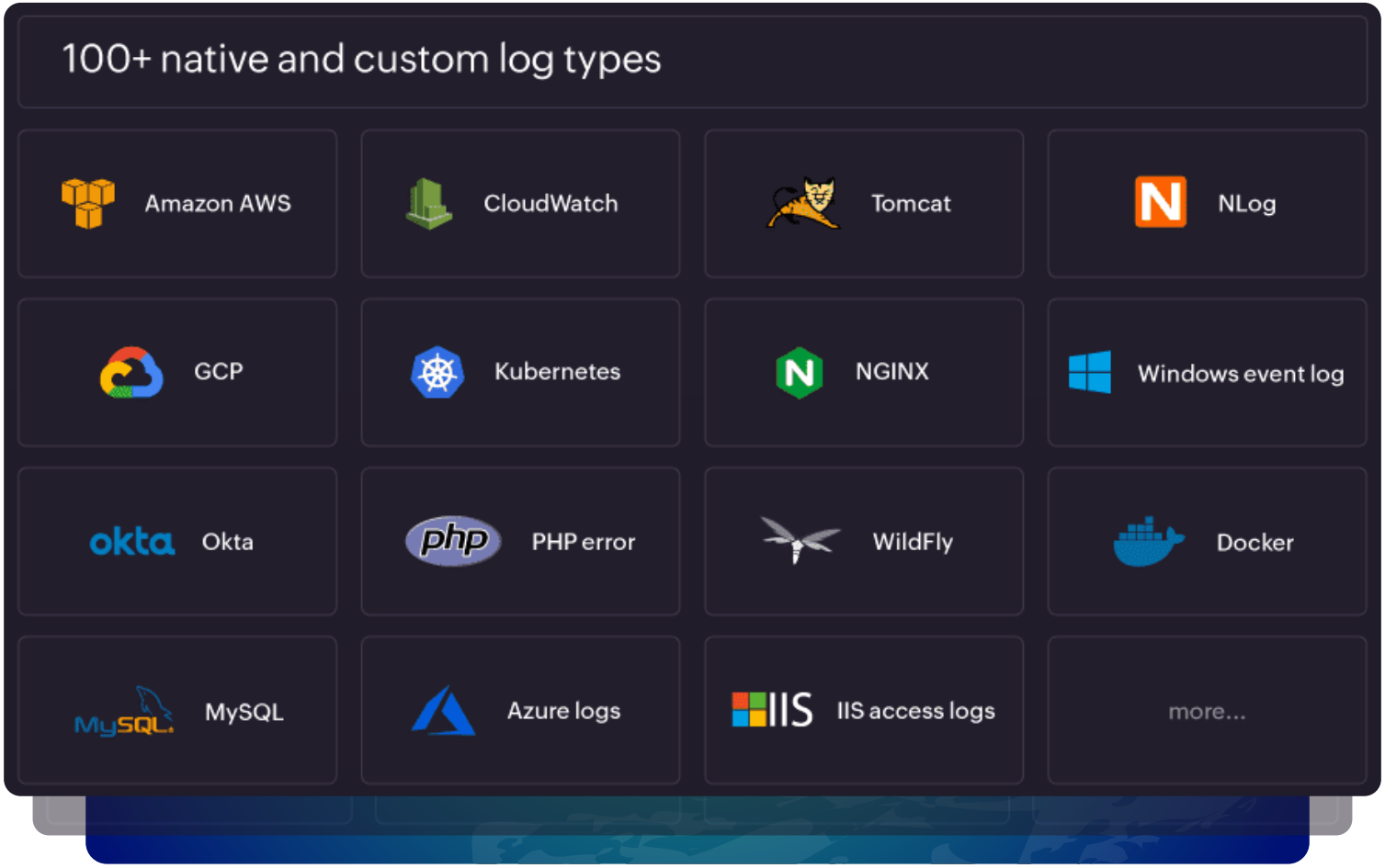
Imagine you are tasked with enhancing an e-commerce website's performance. The e-commerce platform comprises multiple microservices operating on various servers and cloud instances, including web servers, databases, payment gateways, and inventory management systems. Each component generates its own logs, and collecting logs from across the platform is essential for monitoring, analysis, and troubleshooting.

You can easily manage logs by sending them from different sources to Site24x7. You can integrate logs from various places using agent, agentless methods, local or remote files, Windows Event logs, Amazon, Azure, or GCP serverless logs via their respective services, network devices, or open-source log collectors like Logstash or FluentD. Additionally, logs can be sent through an HTTPS endpoint API.



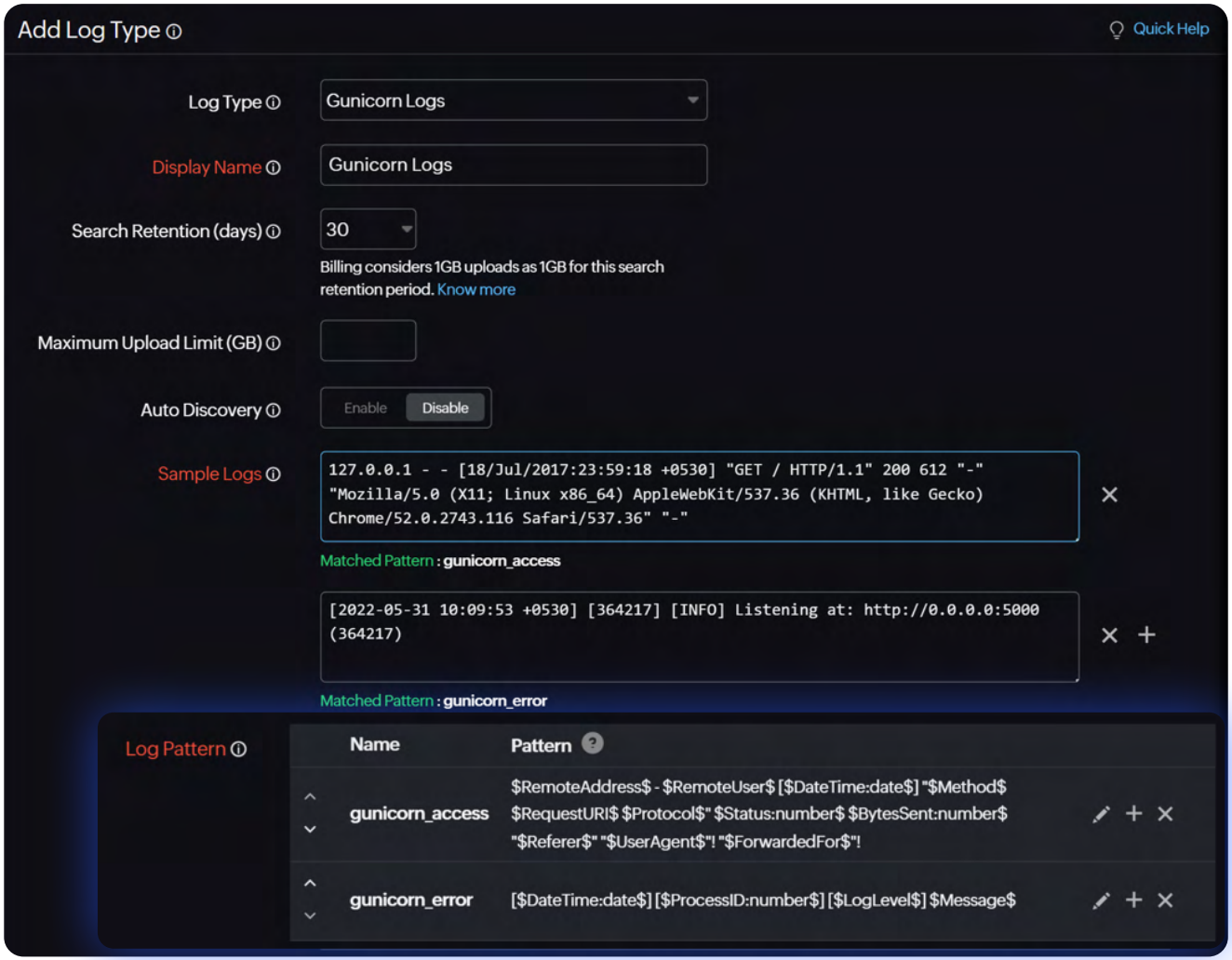
Support for 100+ predefined log formats

Providing out-of-the-box support for over 100 applications, systems, and cloud logs, our platform also automatically discovers the logs from your infrastructure.



Accommodate diverse log formats with multiple log pattern support

Site24x7's multiple log pattern feature allows you to centralize logs from various applications, even if they're in different formats. For instance, if you're interested in monitoring Gunicorn (the Python web server) access and error logs on a single query, our platform provides a solution for centralizing logs to combine different log patterns under a single log type.



Plus, visualize all your data conveniently on a unified dashboard.

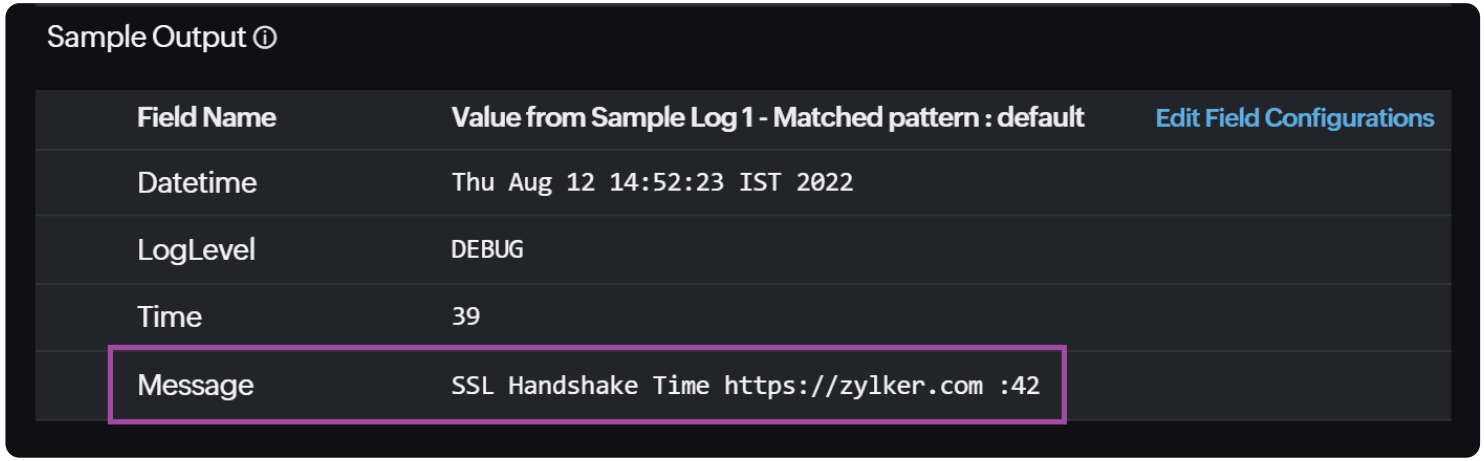
Accelerate the search process by analyzing fields with derived field support

Even within the segmented log data, a single attribute may hold valuable information that shouldn't be overlooked.

For example, consider this log line:

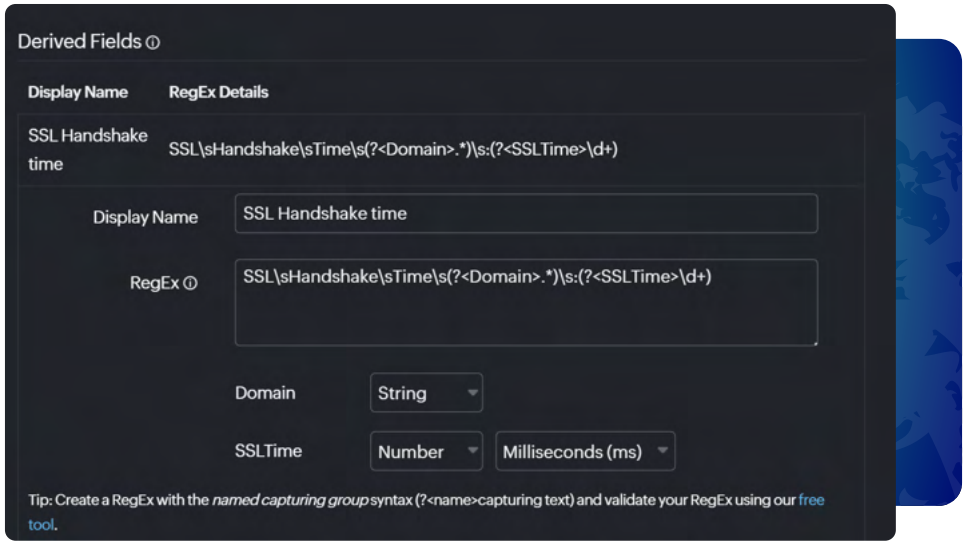
```
[Thu Aug 12 14:52:23 IST 2022|DEBUG|39]: SSL Handshake Time https://zylker.com :42
```

This log line captures a debug-level message about the SSL handshake time for https://zylker.com, along with a timestamp, identifier, and more details.

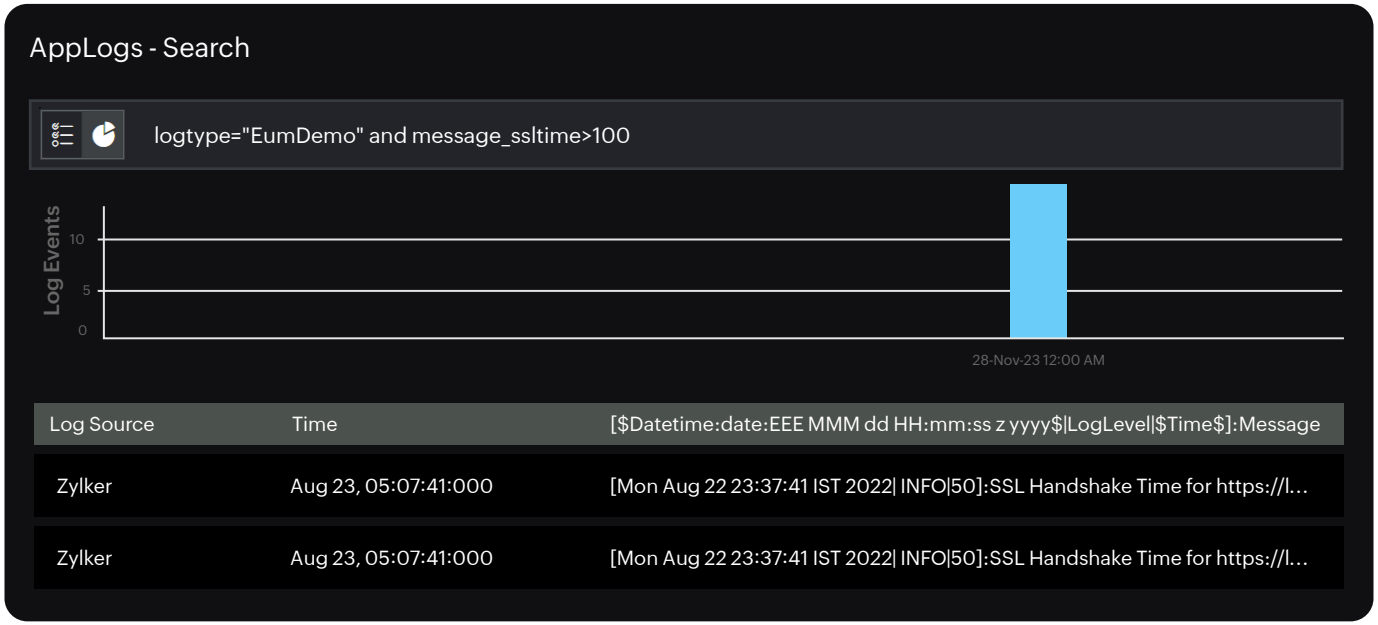


The SSL handshake time provides insights into the speed at which secure connections are established between clients and servers. Extracting and monitoring SSL handshake time and domain name from the message field helps identify potential performance bottlenecks and optimize the SSL/TLS configuration for faster handshakes, thereby improving overall application performance.

To do this, you can add a RegEx rule to create derived fields for the domain and SSL time to extract the data from the message field. Derived field support enables you to create custom parsing rules for log fields that help you extract valuable information.



You can visualize the data on a single dashboard for a better understanding and get alerted for abnormal values.



Hashing and masking sensitive data for enhanced security and compliance

Logs may encompass sensitive data, including personally identifiable information, credit card numbers, passwords, and other confidential information.

For example, take the log line below where API keys are present. API keys are considered sensitive information because they grant access to certain functionalities or resources within an application or system.

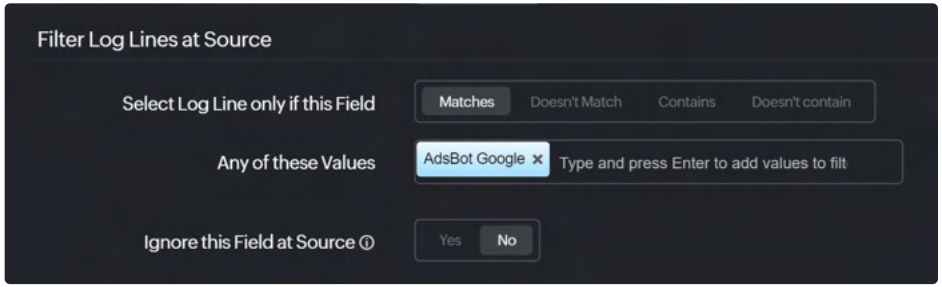
```
209.85.238.199 - - [26/Oct/2021:10:05:15 +0000] "GET presentations/details?apiKey=
877avjkj329082j30sf83s1&type=ppt HTTP/1.1" 200 1370 "-" "Feedfetcher-Google;
(+http://www.google.com/feedfetcher.html; 1 subscribers; feed-id=11390274670024826467)"
```

Before sending your logs to Site24x7, you can hide sensitive data using hashing and masking rules. Configure the expressions you want to mask using regex capture groups. You can customize the masking with a specific string or use the default (***). Hashing replaces expressions with hash codes, securely hiding the data before it's sent to Site24x7. Below is the same log line with the API masked.

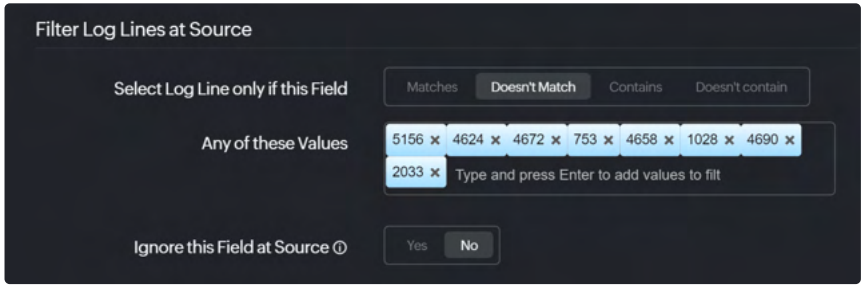
```
209.85.238.199 - - [26/Oct/2021:10:05:15 +0000] "GET presentations/details?apiKey=
***&type=ppt HTTP/1.1" 200 1370 "-" "Feedfetcher-Google; (+http://www.google.com/feedfetcher
.html; 1 subscribers; feed-id=11390274670024826467)"
```

Filter unwanted logs for improved log management

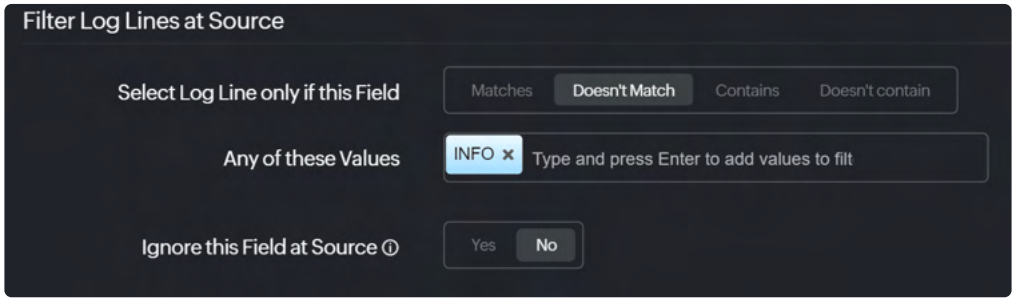
Unnecessary or redundant log lines will increase the volume of data transmitted and stored, thereby increasing storage costs and reducing resource efficiency. By configuring filters at the source, unnecessary or redundant log lines can be excluded. Filters can be set up based on various criteria, such as log severity levels, specific keywords, or originating sources. For example, you might configure filters to ignore bot requests, filter unwanted event IDs, or exclude information logs that are not relevant to your current monitoring or analysis objectives.



1. Ignoring bot requests



2. Filtering unwanted events



3. Ignoring information logs

Simplify debugging with query language

Whether you're a DevOps engineer troubleshooting application issues or a business leader seeking insights into user behavior, our query language provides the essential tools to convert raw log data into actionable intelligence. This powerful language allows you to apply advanced filters and conditions to search for specific log entries.

For example, you can easily filter logs based on timestamps, log levels, error codes, or custom attributes. Additionally, you can perform custom analysis on log data, including aggregating, grouping, and histograms, as well as perform calculations to identify patterns or anomalies. This could include tracking error rates over time, identifying user activity trends, or monitoring system performance metrics.

Below are a few supported operators from a comprehensive list.

Or	and	contains	notcontains
isempty	isnotempty	in	notin
like	startswith	>, <, >=, <=, =, !=	not
count	min	max	sum
avg	sd	percentile	ratio

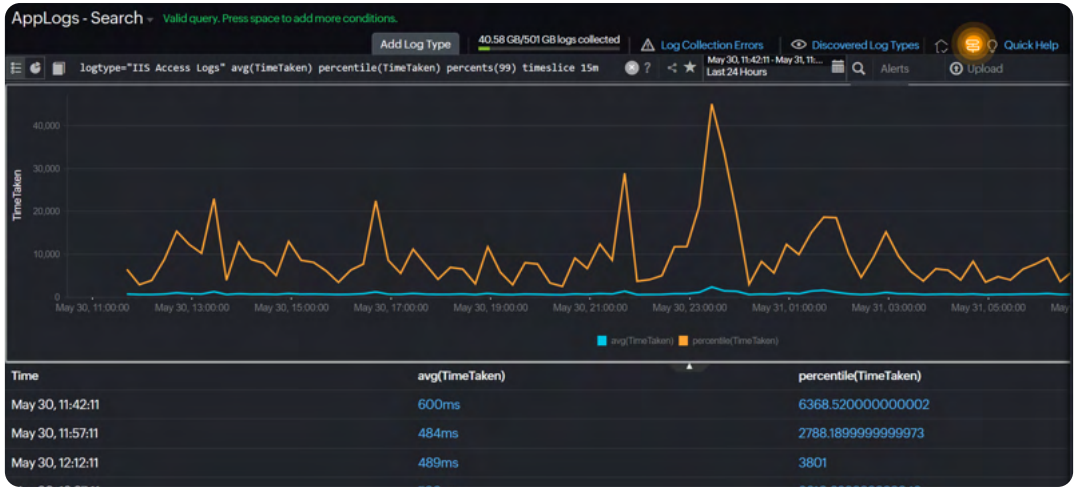
count_distinct	groupby, having	histo, range	tophits
tophits	sort	before	include | exclude
monitor_name	monitor_group	tag	

Here are some scenarios that a query language will help you with:

1. You can evaluate the performance of the IIS web server in terms of response time using this query.

```
logtype="IIS Access Logs" avg(TimeTaken) percentile(TimeTaken) percents(99) timeslice 15m
```

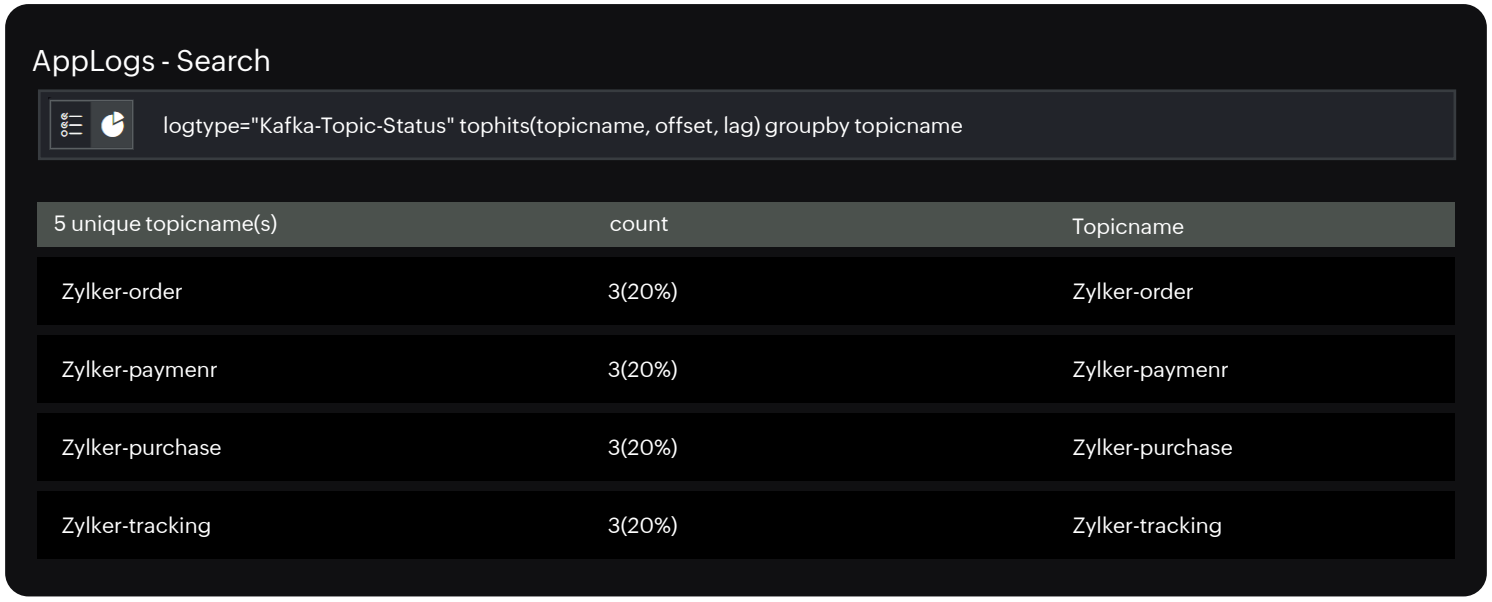
Meeting SLAs often involves ensuring that the response time for web requests falls within certain thresholds, and analyzing metrics such as average and percentile response times can help assess whether those thresholds are being met.



2. You can use the below query to retrieve the latest status information for each Kafka topic, including the offset and lag count. This allows you to monitor the processing status of each topic in real time.

```
logtype="Kafka-Topic-Status" tophits(topicname,offset,lag) groupby topicname
```

Monitoring Kafka topics, offsets, and lag is crucial for several reasons. If you notice a sudden increase in lag counts for a particular topic, it could indicate that the consumers are falling behind in processing messages. With this insight, you can take proactive measures to investigate and address the issue, ensuring smooth data processing and maintaining system performance.



3. The query below is important because it helps filter out bot requests from Apache access logs and then counts the distinct remote hosts accessing the server within each hour's time.

```
logtype="Apache Access Logs" and useragent !="bot" COUNT_DISTINCT(remotehost) timeslice 1h
```

By excluding bot requests (user agents identified as bots), the query focuses on real user interactions with the server, providing insights into genuine traffic patterns. This information is valuable for understanding user engagement, identifying potential security threats, and optimizing server performance based on legitimate user behavior.

Advanced search options for optimized troubleshooting

Let's explore a few search options provided below to navigate and analyze log data efficiently.

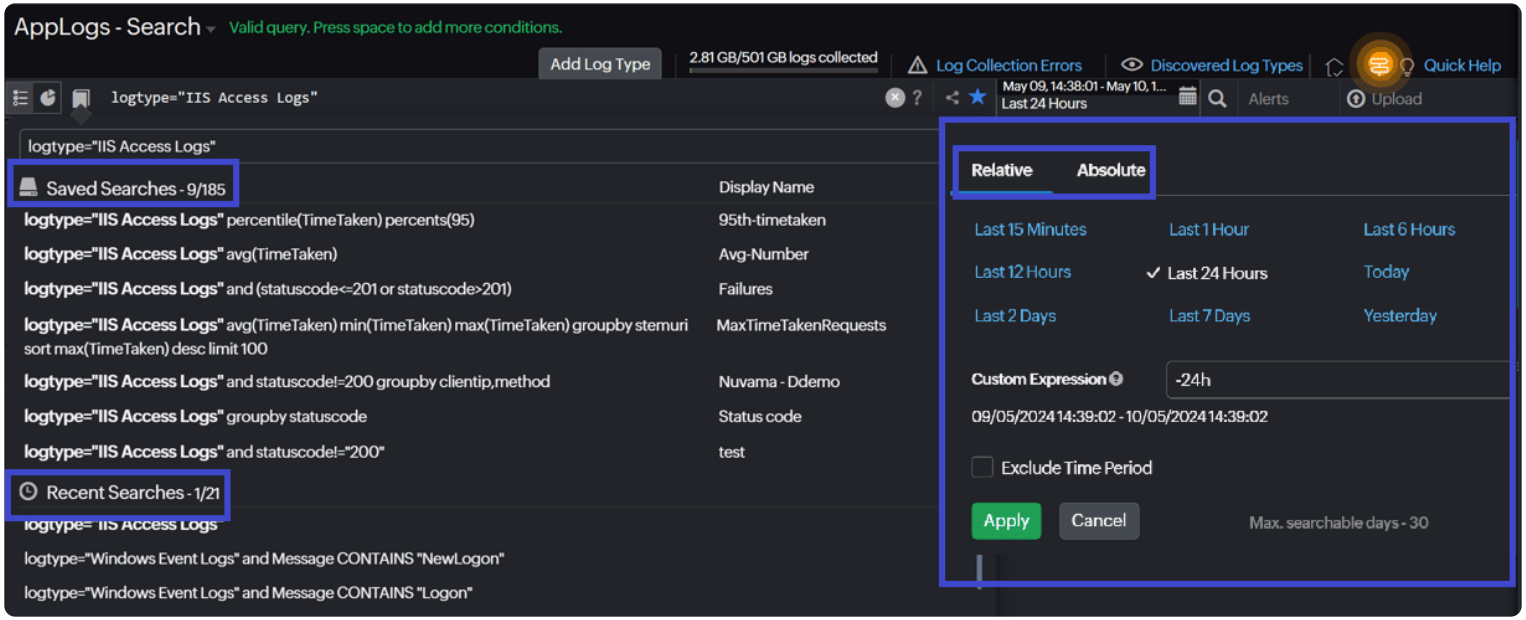
- Saved and recent searches
- Relative time search
- Custom views
- Related log templates

Saved and recent searches

You can save your search queries with Site24x7 so they'll be readily available the next time you need to perform that search. By default, all saved searches are added to the dashboard for quick and easy access. Plus, you have the flexibility to add them to your own custom dashboard. This way, you can stay on top of your most important searches and get the information you need at a glance.

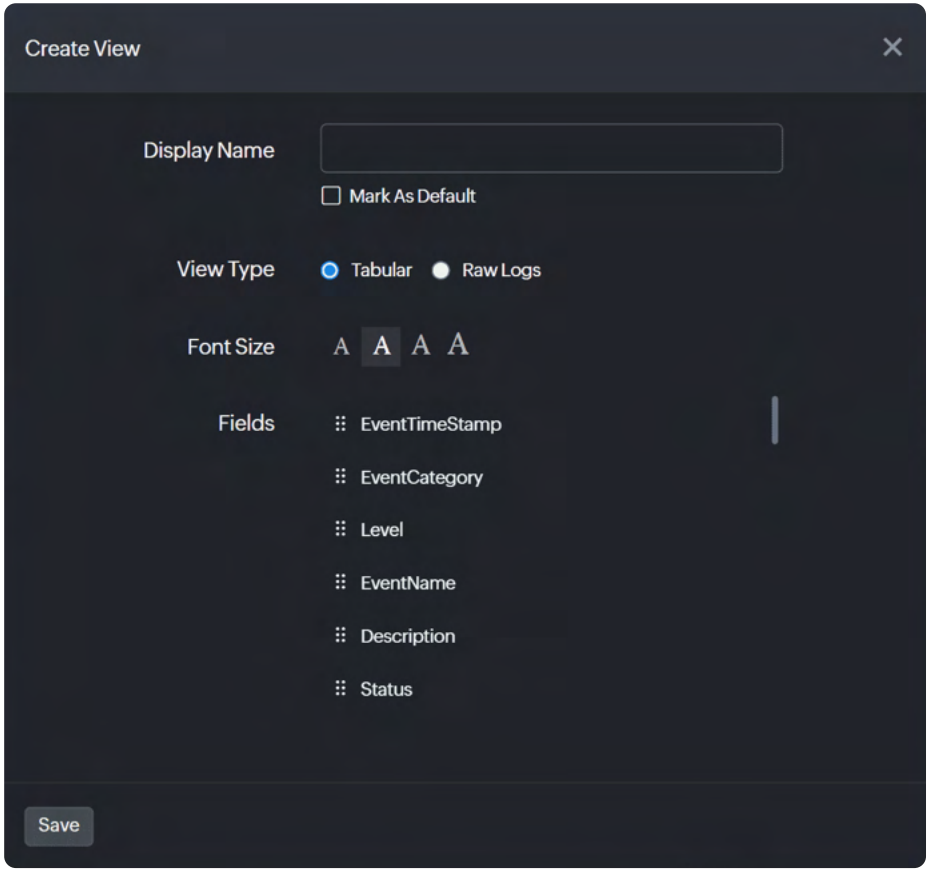
Relative time search

You can use a relative time search to precisely filter your logs based on time intervals such as minutes, hours, or days. This granularity helps you focus on specific events or periods of interest within your log data. It allows you to quickly narrow down your search to specific timeframes without needing to calculate exact timestamps manually. This saves time and effort, especially when investigating recent incidents or trends.



Custom views

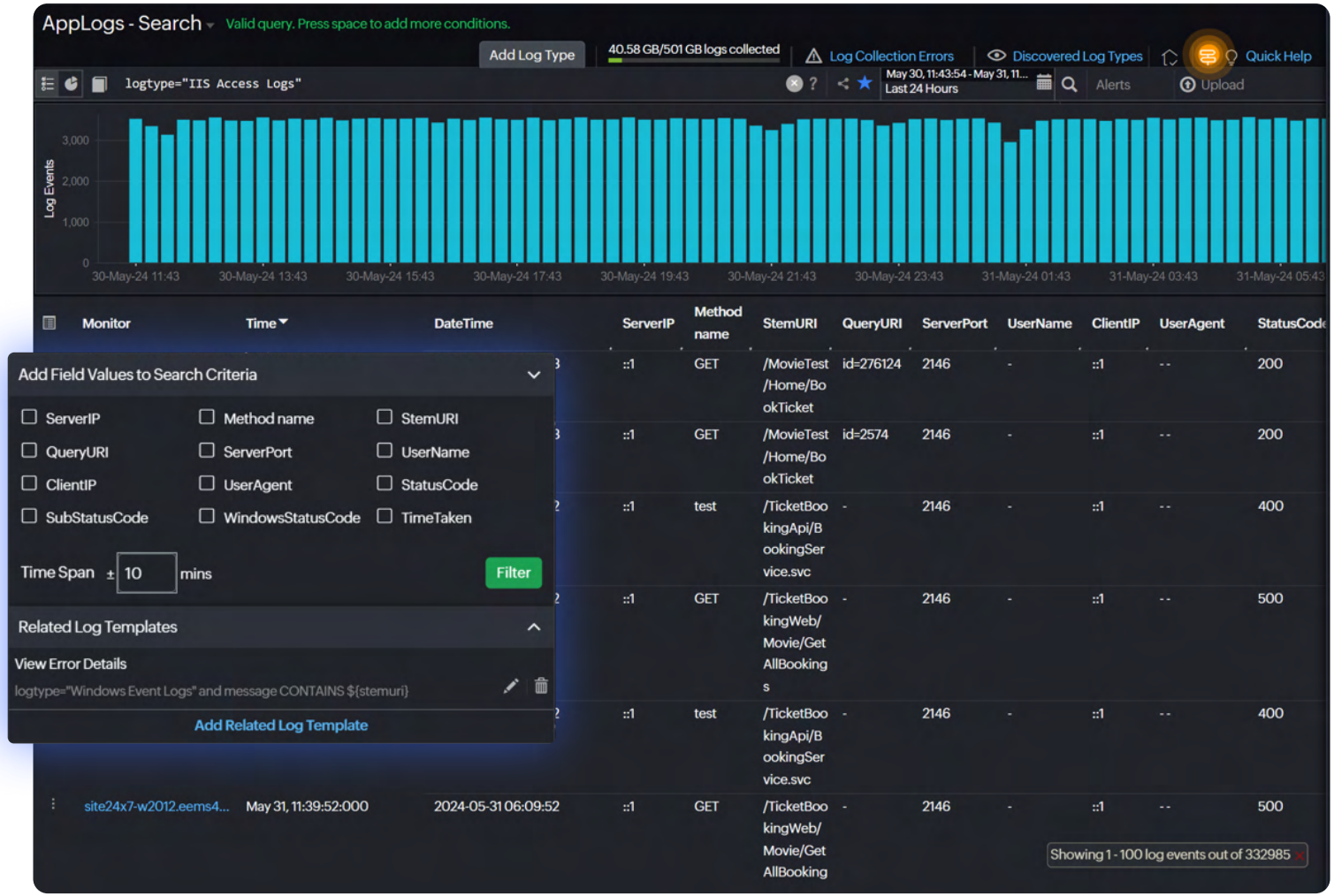
Custom views allow users to tailor the log display to their specific requirements, making it easier to identify patterns, anomalies, and critical events within the log data. Let's take the use case of monitoring Azure activity logs for a cloud-based application. These logs show changes in an Azure subscription, like resource creations or deletions. However, the standard format may have too much information that is not always relevant to a specific monitoring or troubleshooting scenario.



For example, if you are optimizing the performance of a web application running on Azure, you can create a custom view in Site24x7 that focuses on performance-related attributes like event timestamp, event category (operational changes), event name (resource utilization or performance metrics), level (severity level), and status (success or failure).

Related log templates

When troubleshooting application issues, comparing logs from different sources is common. But it's tedious to switch between tabs and find the exact line causing the problem—that's where related log templates help. With this feature, you create a template with fields you want to compare, like ThreadId, and set a time frame. This opens a new tab showing the application logs with the same conditions.



Proactive log alerting for timely action

Real-time event tracking and proactive log analysis play a crucial role in enhancing monitoring efforts and in the optimization of system performance and security. Through customized strategies and a proactive approach to issue detection and preventive maintenance, organizations can improve their user experience and strengthen infrastructure resilience.

Count-based and trend-based alerts

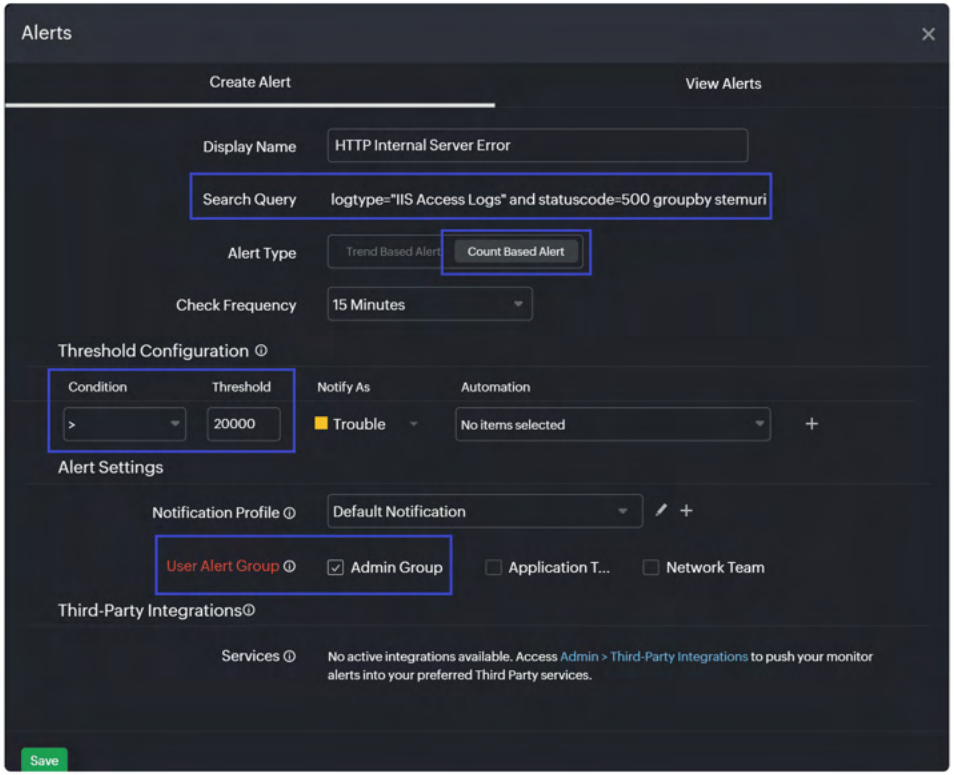
Site24x7 uses count-based and trend-based alerts to monitor log data and notify users about specific conditions or patterns. For example, you can create a count-based alert for IIS access logs with status code 500 and a trend-based alert for the same logs with a 200% increase in failure.

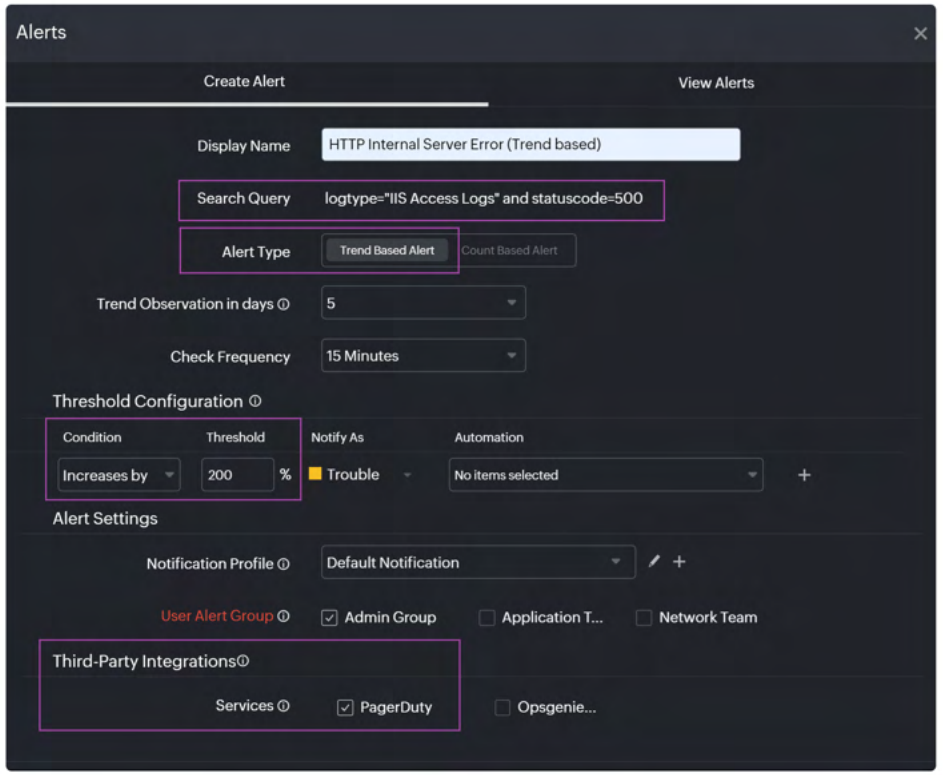
- Setting a count-based alert for logtype="IIS Access Logs" and statuscode=500 groupby stemuri triggers an alert when the failure URI exceeds the configured threshold.

- Setting up a trend-based alert for the same query, logtype="IIS Access Logs" and statuscode=500, and activating it when the 500 failure count for the particular request URI increases by 200% over a regular failure, you can detect significant shifts in error rates.

Count-based alerts offer granular visibility into specific error-prone URLs, while trend-based alerts offer a broader perspective on evolving error trends.

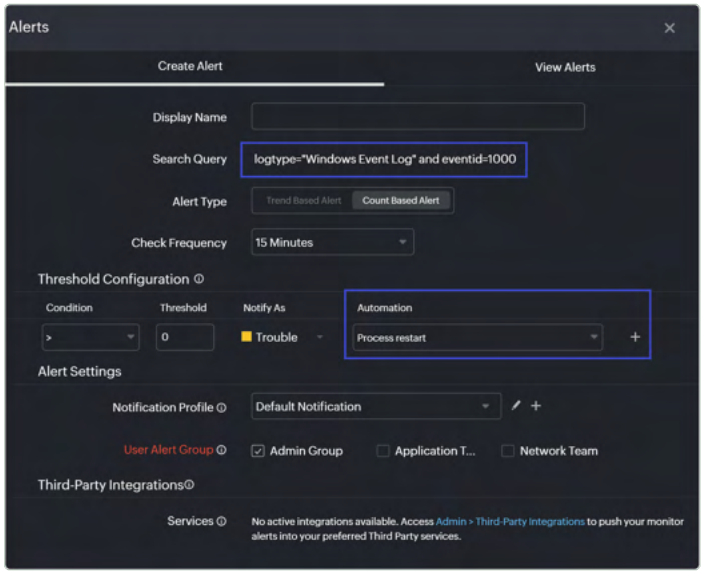
You can choose your preferred alert medium, such as email or SMS, or integrate with various third-party ITSM and collaboration tools, like PagerDuty, Opsgenie, and ServiceNow.





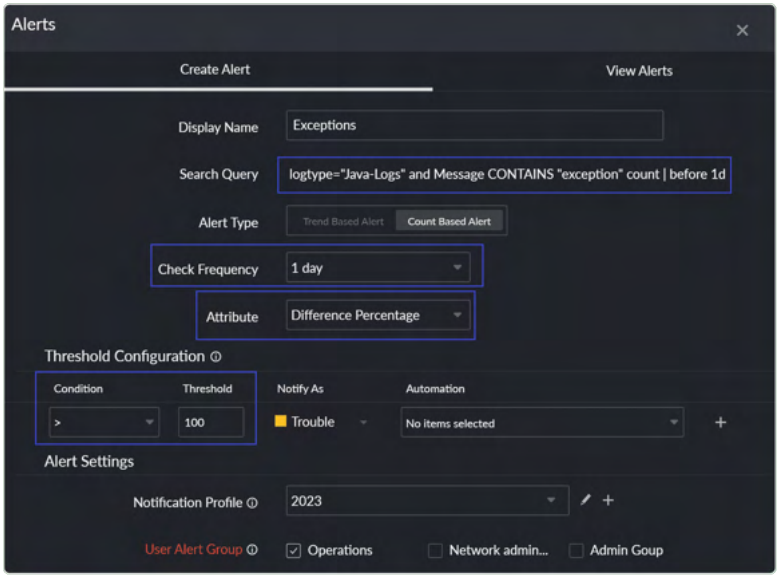
Auto-resolve issues with IT automation

Event-driven IT automation enhances productivity by auto-resolving log event alerts in seconds without manual intervention. For example, you can monitor critical system errors or application crashes in Windows event logs by creating an alert for event ID 1000 and enabling IT automation to restart services. This reduces resolution time and maintains application stability and performance.



Monitor key performance indicators

You can monitor key performance indicators to ensure application stability and get notified when they suddenly increase or decrease over a specific period. For example, when monitoring the frequency of exceptions logged within a Java application, you can create a count-based alert to track exceptions. If the count increases by more than 100% compared to the previous check, users belonging to the specified alert group will be notified.



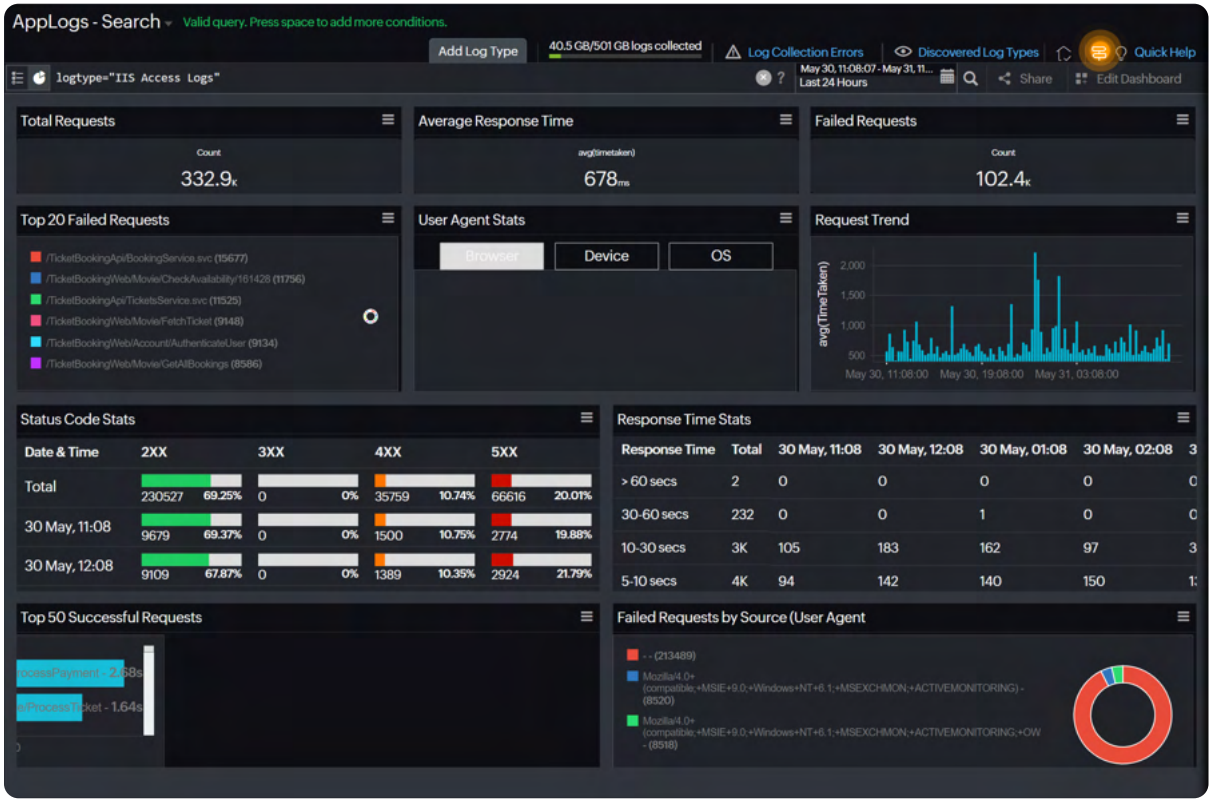
You can also use persistent alerts in Site24x7 to ensure that you receive continuous notifications until you acknowledge the Down, Critical, or Trouble alarms.

Log dashboards for actionable insights

The log dashboard in Site24x7 empowers both business users and SREs, DevOps, and IT admins to monitor, analyze, and optimize system performance and application reliability from different perspectives.

As a business user, you can use the dashboard to understand customer behavior, application usage patterns, and system health. You can also track user activity trends, performance of key features, and identify issues impacting user experience. This information helps you make informed decisions, prioritize feature enhancements, and allocate resources effectively.

If you're a SRE, DevOps engineer, or IT admin, you can monitor server uptime, response times, error rates, and resource utilization in real time. You can also set up custom dashboards, configure alerts, and troubleshoot issues by correlating log data with performance metrics. This proactive approach helps you identify and resolve potential issues before impacting users, ensuring optimal system reliability and uptime.

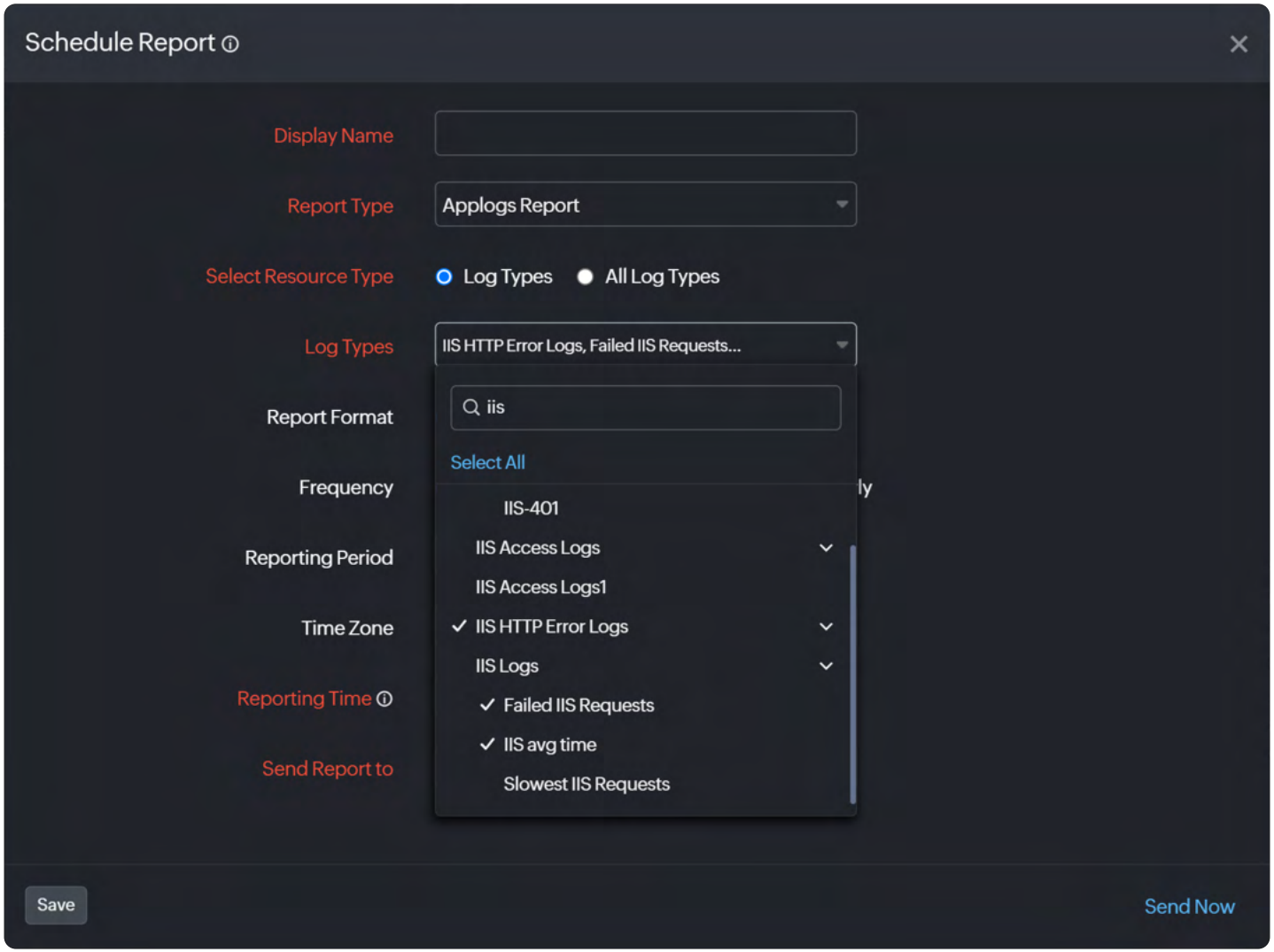


Site24x7 creates an exclusive dashboard for every log type, and you can also build a custom dashboard across log types.

Streamline reporting with scheduled reports and exports

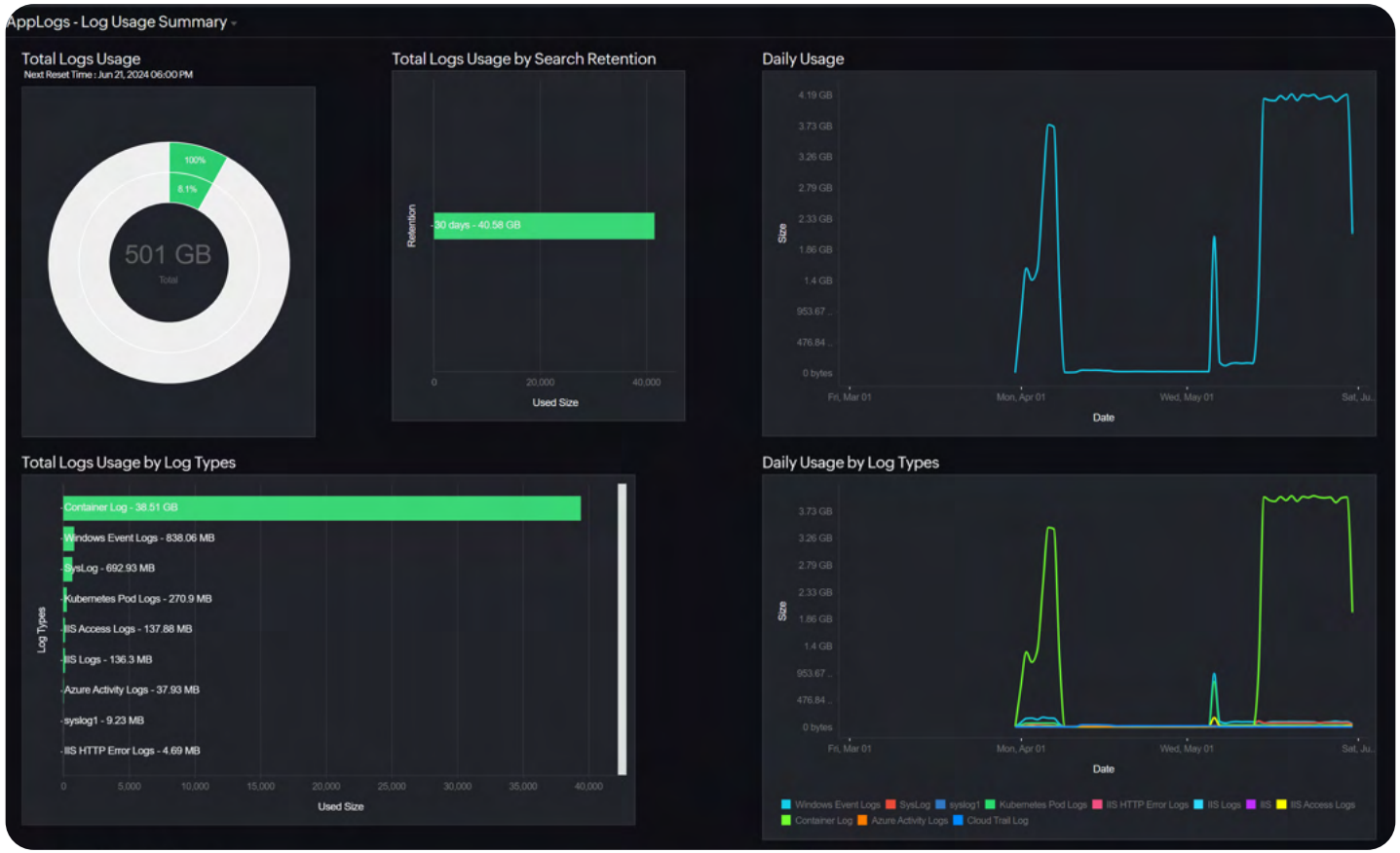
Scheduled reports provide comprehensive insights into configured widgets for your log types. You can schedule reports for both your log types and saved searches.

These reports offer detailed insights into default log type widgets and saved searches of your chosen log types. You can select specific saved search queries to generate customized reports, ensuring you receive the most relevant information in a conve nient format. You can also export the search result to PDF format.



Affordable log management

The log usage summary offers an extensive view of your log usage, including overall and daily consumption, for your monthly AppLogs subscription with Site24x7. This feature helps you analyze usage patterns, manage log consumption effectively, configure alerts for log usage spike, and make informed decisions about log add-ons. Site24x7's cloud-based scalability makes it simple and cost-effective to store essential logs, allowing you to scale to handle logs up to several terabytes. You can search through up to 90 days' worth of logs simultaneously and utilize re-indexing to access archived logs.



Try Site24x7 today

Boost your application and infrastructure performance using our single-console log management tool. Site24x7 allows you to save queries, create alerts based on them, and visualize your results on intuitive dashboards while also receiving real-time notifications on crucial log events.

On our centralized platform, you can combine metrics, events, traces, and logs for enhanced observability and valuable insights. In addition to log management, you can comprehensively monitor your application and server metrics. Our application performance monitoring (APM) tool provides continuous monitoring, enabling instant identification of issues and their root causes. Seamless integration between log management and APM greatly enhances DevOps productivity, providing comprehensive visibility into your IT environment. Try Site24x7 for log management today!

About ManageEngine Site24x7

ManageEngine Site24x7 is an AI-powered observability platform for DevOps and IT operations. The cloud-based platform's broad capabilities help predict, analyze, and troubleshoot problems with end-user experience, applications, microservices, servers, containers, multi-cloud, and network infrastructure, all from a single console. For more information about Site24x7, please visit www.site24x7.com.

Get Quote | Request Demo