NGINX Integration

Introduction

The Nginx integration allows you to monitor Nginx servers. Use the Nginx integration to collect metrics and logs from your server then visualize that data.

For example, if you wanted to be notified if a certain number of client requests failed in a given time period, you could install the Nginx integration to send logs to Elastic. Then, you could view the logs stream into Elastic in real time in the Observability Logs app. You could also set up a new log threshold rule in the Logs app to alert you when there are more than a certain number of events with a failing status in a given time period.

Data Collection Types

It collects two main types of data:

- **Logs** — Capture and record events occurring within the Nginx server. These include access logs (client requests) and error logs (issues encountered during request handling). Log data helps in auditing activities, identifying issues, and analyzing request patterns.
- **Metrics** — Provide real-time performance insights into Nginx server operations. Metrics include details such as the total number of active client connections, connection states, request counts, and other performance indicators essential for capacity planning and system optimization.

By utilizing this integration, administrators gain visibility into both operational and performance aspects of Nginx, enabling effective monitoring, troubleshooting, and optimization of web infrastructure.

Prerequisites

Before setting up the **Nginx Integration**, ensure that the following requirements are met:

1. **Nginx Server Installed and Running**
   - A functioning **Nginx server** must be installed on your host system.
   - Verify that the Nginx service is active and accessible.

2. **Access Permissions**
   - Administrative or root privileges are required to configure log file paths and enable the Nginx status module.
   - Read permissions must be granted for Nginx log files (e.g., `access.log` and `error.log`).

3. **Nginx Status Module Enabled**
   - The **stub_status** module should be enabled to allow collection of server metrics such as active connections and request rates.
   - Add or verify the following configuration in your Nginx configuration file.

```
location /nginx_status {
    stub_status;
    access_log off;
    allow 127.0.0.1;    # restrict access as needed
    allow <Network_IP>; # e.g. 192.172.10.0/24
    deny all;
}
```

   - Test and Reload Nginx after making changes:

```
sudo nginx -t              # Test first
sudo systemctl reload nginx  # Use reload, not restart
```

   - Verify it works:

```
curl http://127.0.0.1/nginx_status
```

     Linux:
     Ubuntu/Debian:
     - Main config: `/etc/nginx/nginx.conf`
     - Site configs: `/etc/nginx/sites-available/default`
     - Enabled sites: `/etc/nginx/sites-enabled/default` (symlink)

     CentOS/RHEL:
     - Main config: `/etc/nginx/nginx.conf`
     - Site configs: `/etc/nginx/conf.d/default.conf`

     Where to add `stub_status`:
     - Add to your site configuration file (e.g., `/etc/nginx/sites-available/default` or `/etc/nginx/conf.d/default.conf`)
     - Or add directly to `/etc/nginx/nginx.conf` in the `server` block

     Windows:
     Configuration file location:

```
C:\nginx\conf\nginx.conf
```

     Where to add `stub_status`:
     - Edit `C:\nginx\conf\nginx.conf`
     - Add inside the `http { server { } }` block

macOS:
Homebrew installation:

```
/usr/local/etc/nginx/nginx.conf
```

or

```
/opt/homebrew/etc/nginx/nginx.conf
```

MacPorts Installation:

```
/opt/local/etc/nginx/nginx.conf
```

4. **Network Connectivity**
   - Ensure that the system where monitoring is configured can connect to the Nginx host via the appropriate network ports (typically port **80** or **443**).

5. **Log File Availability**
   - Confirm that standard Nginx log files are present in their default or custom locations:
     - Access logs: `/var/log/nginx/access.log`
     - Error logs: `/var/log/nginx/error.log`

On the device where **Nginx Server** is installed, you must also install the **AQUILA Log Collector Agent**. This agent is responsible for collecting the Nginx access and error logs and forwarding them to AQUILA for processing.

Please refer to the official manuals for installing the **AQUILA Log Collector Agent** on different operating systems:

- **Linux:** Log Collector Installation - Linux Manual
- **Windows:** Log Collector Installation - Windows Manual
- **Mac:** Log Collector Installation - Mac Manual

Ensure that after installation, the Log Collector service is running properly.

Required Credentials for Integration

Collect logs from Nginx instances

- Access Logs Path: *eg.* ***/var/log/nginx/access.log****
- Error Logs Path: *eg.* ***/var/log/nginx/error.log****

Collect metrics from Nginx instances

- Host: *eg. http://127.0.0.1:80*

Integrating monitoring for NGINX is straightforward once your setup and agents are in place: simply add the integration to an agent policy, configure log paths (e.g., /var/log/nginx/*.log) and metrics host/path (e.g., http://127.0.0.1/nginx_status), and let the managed agents handle collection. This enables real-time monitoring of access/error logs and server metrics like connections and requests, with pre-built dashboards for visualization, alerting, and anomaly detection.

NGINX integration is really helpful for boosting reliability and performance insights with minimal effort—it provides centralized management, scalable observability, and easy customization for custom log formats. If issues arise, check agent status and permissions first. For production, secure endpoints and rotate credentials. Dive into the docs for advanced tweaks!

If you need further assistance, kindly contact our support at info@cytechint.com for prompt assistance and guidance.

Revision #2
Created 4 February 2026 01:45:46
Updated 4 February 2026 06:47:23