ALGORITHMIA ENTERPRISE

ADMINISTRATION EXPERIENCE

ALGORITHMIA

Algorithmia Enterprise is the foundation layer for intelligent software. It turns complex services and machine learning models into REST APIs, centralizes them for ease of discoverability, and monitors them from a single dashboard. Companies use Algorithmia Enterprise to reduce duplication of effort between siloed teams and accelerate go-to-market for AI-driven products.

An Algorithmia Enterprise cluster is designed to self-heal and auto-scale on demand, supporting thousands of developers and thousands of services in critical production environments. With that in mind, we have designed a rich administrative experience that enables super users to monitor and interfere whenever necessary.

This document gives a brief overview on the following topics:

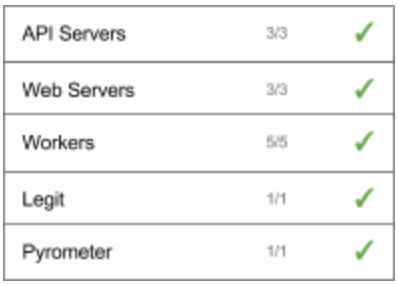
- System Metrics
- API Metrics
- Workers & Scaling
- Errors Logs
- Dashboards & Triggers

System Metrics

This is an Administrator's first stop and best place to identify the current state of the system. It provides a bird's-eye view of overall system metrics and health and highlights any abnormal behavior.

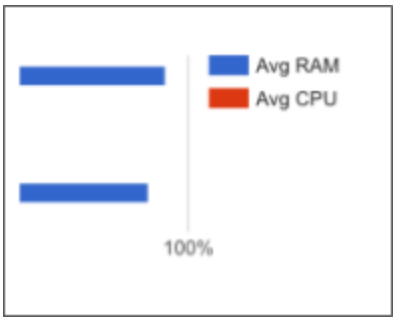
API Calls & Errors Per Minute

Algorithmia Enterprise logs every API call, including errors. A line chart in the admin panel will show how many calls were received in the past duration and the proportion of that to user-errors and system-errors, enabling admins to immediately identify spikes or falls in traffic.



Service Health

The result of service heartbeat is shown on the main page of the admin dashboard. A color-coded notification will appear next to any service group with an unhealthy member.

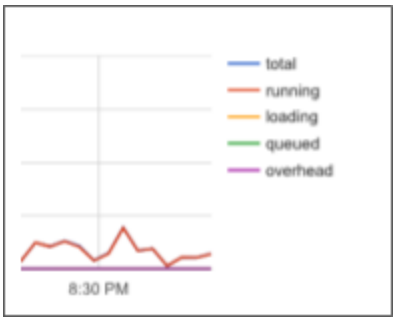


Cluster Utilization

Algorithmia Enterprise shows the cumulative average RAM and CPU for each Worker within your cluster. Those values are also shown over time as a line chart, allowing you to correlate it with API calls over time.

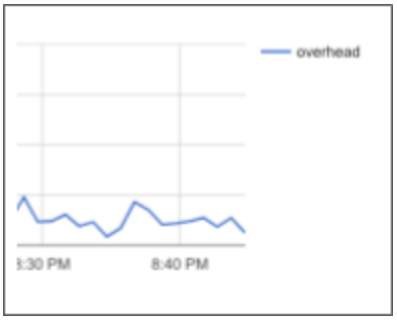
API Metrics

Everything is instrumented. Algorithmia Enterprise keeps detailed metrics on each API call, from hitting the load balancer to processing to pushing back to the client. Administrators are able to spot trends and identify bottlenecks in real-time.



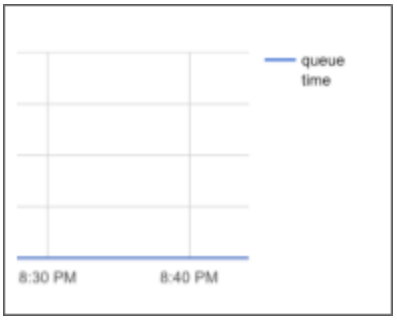
API Runtime

The main API timing metrics chart breaks down the average API call timing into runtime (actual algorithm processing), load time (loading Docker container from cold-start), overhead, and queue time.



API Overhead Timing

Overhead is defined as the latency introduced by the system - in other words, the time it takes for an API call to go from the load balancer until it reaches the entry point of the algorithm. This latency is hardware-dependent and is typically in low milliseconds.



API Queue Timing

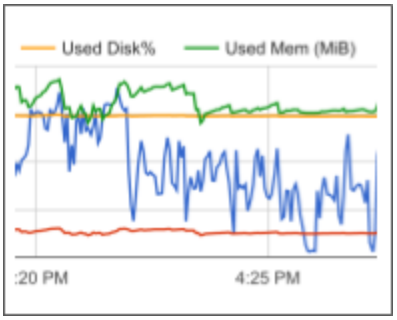
Queue is the average time an API call is queued because the system was unable to immediately allocate resources for that request. This metric is measured in milliseconds and is typically flat - otherwise it will trigger the the auto-scaler depending on sensitivity settings.

Workers & Scaling

Workers make up your cluster. A Worker is a virtual or physical machine instance dedicated to running user code. The Workers tab enables administrators to zoom-in from cluster-level view to worker-level view and drill down to a running algorithm-process-view.

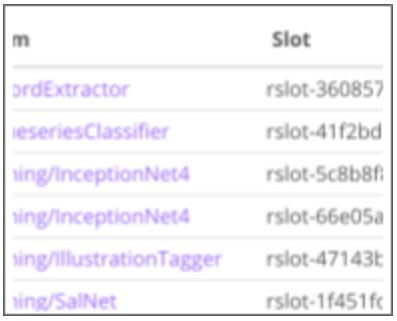
Monitor, Kill, and Add Workers

A mix of CPU and GPU machines on different clouds and zones tagged with color-coded utilization metrics. The list of workers is updated as your cluster automatically adds or kills workers in response to demand, which is an operation that admins can manually adjust as well..



Inspect Worker Resources

Inspecting an individual worker reveals utilization metrics (CPU, RAM, Disk) and currently running algorithms within that node. GPU workers show additional metrics, such as GPU memory utilization and frame buffer levels.

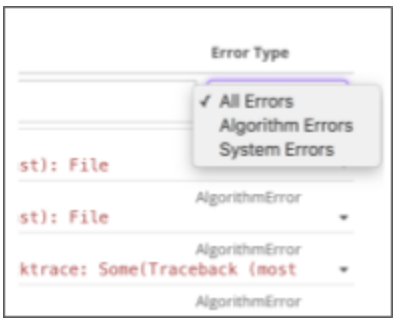


Inspect Running Algorithms

Admins can drill down to individual algorithm processes and examine how much each algorithm is contributing to the overall utilization metrics. In addition to helping admins diagnose their cluster, Algorithmia Enterprise keeps track of these numbers to optimize its orchestrator in a way that maximizes cluster utilization and minimizes cloud bill.

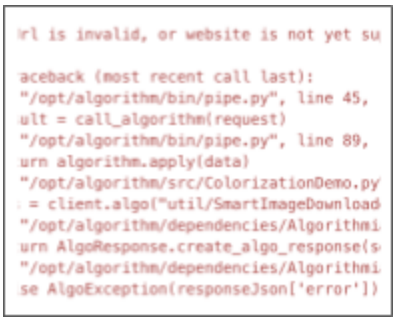
Error Logs

Exception messages occurring on any worker will immediately appear here. Administrators are able to know what user or algorithm is generating most errors and work with them to improve that experience - either from the Admin Panel or using external log-parsing tools like Splunk.



User Versus System Errors

User errors are generated by the algorithm (user code) or as a result of user usage (such as corrupt input, missing file, or short timeout settings). System errors are very rare, capture worker-level errors, and is one of several factors that Algorithmia Enterprise monitors to determine workers' health and abnormal behavior.



Configurable Verbosity

Each error log includes a timestamp and is attributed to a user, worker, algorithm, and algorithm version. Logs can be configured to include all or partial input, and all or partial stacktrace.

Searchable Logs

Admins can list errors generated by a particular user, algorithm, worker, or errors within a specific time range. Filters allow admins to identify the most common error source (i.e. user or algorithm) and take any action to improve that experience.

Dashboards

From dashboards on big screens to single event triggers. Algorithmia Enterprise has the flexibility to provide those metrics and events to your views.

Grafana Built-in

Algorithmia Enterprise comes bundled with Grafana, with all the relevant views and dashboards already populated. Administrators can customize any of those views to match their internal requirements.

Your Own Dashboards

All metrics and events are exposed via REST API endpoints. Admins can utilize those endpoints or ones from Grafana to integrate with their existing dashboarding solutions, including metrics around errors, most active users, and overall cluster health.

Integrations & Event Triggers

Admins can design triggers for specific metrics, such as emails or Slack notifications when error threshold passes a threshold.