Logging & Monitoring

2 imp concepts that help debug issues, find improvement areas easily as your sys. keeps growin

LOGGING:

The act of collectin and storin logs (useful info abt. events in your sys).

Typically, prog.s output log msgs to its STDOUT (regular logs) or STDERR (for error logs) pipes.
eg:- console.log for JS, cout for C++, print() for Python.

The outputs recd at the end of these pipes automatically get aggregated into a 'centralized logging solⁿ' → like a DB we store all logs in for visibility into sys's f"ing and debug issues
eg:- Google Stackdriver.

Use case :- someone purchasin a product gets charged but doesn't gain prod. access
Logs related to that user's opn./behaviour, shows you what exactly happened durin this transacⁿ opn.

Gotta format these logs usin a special lib to ensure logs r hr useful info and are easy to read

2 popular opⁿ's → Syslog, JSON

Date:
M T W T F S S

MONITORING :

The process of givin visibility into a sys's key metrics. Creatin sys-s to monitor imp metrics on overall sys
Usually implemented by collectin imp events in a sys, and aggregatin em in human-readable statistic representa"s (eg:- charts)

Makes managin a sys, lookin at its performance stats, stats related to which part of sys is most imp to users (eg:- Google Sign In > sign in by email), stats related to gen health, status of sys. ⟶ a.k.a. gathering metrics

As sys grows, we wanna hv more and more visibility into a lotta opns. in it

eg: of metrics that are useful →
- e-commerce related (how many sales per week?)
- health, performance, gen. status of sys
- stats on what features are used most
- login (eg:- logins/day), authentica" stats (eg:- Google auth vs Github auth)

→ M-1 : Build / use a tool that scrapes logs and extracts data to create metrics

Problems → ltd. by data present in logs.
↳ if you ever decide to change your logs, you risk breakin the monitorin sys.

M-2 → Using a time series dB (TSDB) i.e. a specialised dB specifically tailored for data measured of time
eg of TSDB → InfluxDB, Prometheus, Graphite

Servers periodically send data to TSDB

Querying this dB (use query lang) for data

Use tools to graph values stored in dB
(eg of tool: Grafana)
↳ creates graphs out TSDBs

Can build robust monitoring sys.s to reduce dependency on just logs

ALERTING:

The process thru which sys. admins get notified when critical sys. issues occur.

Alerting can be set up by defining specific thresholds on monitoring charts, past which alerts are sent to a comm" channel (eg of channel → Slack)

eg:- alerts on abnormal ↑ in latency of sys.

Logging And Monitoring

CREATE METRIC | CREATE SINK | SAVE SEARCH | SHOW LIBRARY

Filter by label or text search

GKE Container, main-cluster, algoexpert | All logs | Any log level | Last hour | Jump to now

Showing logs from the last hour ending at 5:36 PM (PST) — Download logs | View Options

```
2020-01-13 17:35:53.473 PST {"service_name":"auth","response_code":200,"ts":1578965753.4734147,"elapsed_time_ms":0.165404…
2020-01-13 17:35:53.473 PST {"path":"/api/auth/describe","ts":1578965753.4737482,"level":"info","elapsed_time_ms":1.09223…
2020-01-13 17:35:53.481 PST {"msg":"Running code in csharp for user ","ts":1578965753…
2020-01-13 17:35:53.481 PST {"msg":"Prepping user code on worker 0","ts":1578965753.4810894,"level":"info"}
2020-01-13 17:35:53.488 PST {"ts":1578965753.488276,"level":"info","msg":"running code"}
2020-01-13 17:35:53.526 PST {"msg":"Request completed in 153.196916ms (retry: 0)","ts":1578965753.526525,"level":"info"}
2020-01-13 17:35:53.526 PST {"ts":1578965753.5267758,"request_id":"86232398","elapsed_time_ms":875.431942,"request_path":…
2020-01-13 17:35:53.532 PST level=warn ts=2020-01-14T01:35:53.532Z caller=scrape.go:930 component="scrape manager" scrape…
2020-01-13 17:35:53.577 PST {"request_path":"/api/users/v1/meta","msg":"Request Served","level":"info","service_name":"us…
2020-01-13 17:35:53.582 PST {"request_path":"/api/users/v1/screen_lease/acquire","msg":"Request Served","level":"info","s…
2020-01-13 17:35:53.641 PST {"service_name":"auth","response_code":200,"ts":1578965753.6414318,"elapsed_time_ms":0.170159…
2020-01-13 17:35:53.642 PST {"msg":"Internal Request","path":"/api/auth/describe","ts":1578965753.641831,"level":"info","…
2020-01-13 17:35:53.667 PST {"ts":1578965753.6672525,"elapsed_time_ms":0.838036,"request_id":"36655995","request_path":"/…
2020-01-13 17:35:53.722 PST {"level":"info","service_name":"users","response_code":200,"ts":1578965753.7219486,"elapsed_t…
2020-01-13 17:35:53.728 PST {"service_name":"auth","response_code":200,"ts":1578965753.7285347,"request_id":"86939580","e…
2020-01-13 17:35:53.729 PST {"msg":"Internal Request","path":"/api/auth/describe","ts":1578965753.7290704,"level":"info",…
2020-01-13 17:35:53.734 PST {"request_path":"/api/events/v1/event","msg":"Request Served","level":"info","service_name":"…
2020-01-13 17:35:53.741 PST {"ts":1578965753.7413568,"elapsed_time_ms":100.670154,"request_id":"2719906","request_path":"…
2020-01-13 17:35:53.746 PST {"request_path":"/api/auth/describe","msg":"Request Served","level":"info","service_name":"au…
2020-01-13 17:35:53.747 PST {"path":"/api/auth/describe","ts":1578965753.7470586,"level":"info","elapsed_time_ms":1.13663…
2020-01-13 17:35:53.748 PST {"ts":1578965753.7483938,"level":"info","msg":"Running code in python for user "
2020-01-13 17:35:53.748 PST {"ts":1578965753.7484279,"level":"info","msg":"Prepping user code on worker 1"}
2020-01-13 17:35:53.749 PST {"ts":1578965753.748842,"request_id":"46706538","elapsed_time_ms":0.158472,"request_path":"/a…
2020-01-13 17:35:53.749 PST {"path":"/api/auth/describe","ts":1578965753.7492566,"level":"info","elapsed_time_ms":1.03406…
2020-01-13 17:35:53.754 PST {"ts":1578965753.7547185,"level":"info","msg":"running code"}
2020-01-13 17:35:53.755 PST {"request_path":"/api/auth/describe","msg":"Request Served","level":"info","service_name":"au…
2020-01-13 17:35:53.756 PST {"msg":"Internal Request","path":"/api/auth/describe","ts":1578965753.7563138,"level":"info",…
2020-01-13 17:35:53.756 PST {"ts":1578965753.7565453,"level":"info","msg":"Running code in go for user testing|caee8e08-7…
2020-01-13 17:35:53.756 PST {"msg":"Prepping user code on worker 0","ts":1578965753.7565784,"level":"info"}
2020-01-13 17:35:53.763 PST {"ts":1578965753.7634645,"level":"info","msg":"running code"}
2020-01-13 17:35:53.885 PST level=warn ts=2020-01-14T01:35:53.884Z caller=scrape.go:930 component="scrape manager" scrape…
2020-01-13 17:35:53.902 PST level=warn ts=2020-01-14T01:35:53.902Z caller=scrape.go:930 component="scrape manager" scrape…
2020-01-13 17:35:53.951 PST {"request_path":"/api/users/v1/screen_lease/acquire","msg":"Request Served","level":"info","s…
2020-01-13 17:35:53.990 PST {"ts":1578965753.9900293,"request_id":"34391740","elapsed_time_ms":0.51834,"request_path":"/a…
2020-01-13 17:35:53.992 PST {"request_path":"/api/users/v1/screen_lease/acquire","msg":"Request Served","level":"info","s…
2020-01-13 17:35:54.021 PST level=warn ts=2020-01-14T01:35:54.020Z caller=scrape.go:930 component="scrape manager" scrape…
2020-01-13 17:35:54.030 PST {"ts":1578965754.0304697,"level":"error","msg":"run failed: exit status 1"}
2020-01-13 17:35:54.030 PST {"msg":"Done running user code on worker 1 at /shared/worker-1: [timeout 3 do-test /mnt/progr…
2020-01-13 17:35:54.031 PST {"msg":"worker ready check: true","ts":1578965754.031149,"level":"info"}
2020-01-13 17:35:54.031 PST {"service_name":"rce-python","response_code":200,"ts":1578965754.031206,"request_id":"6086324…
2020-01-13 17:35:54.031 PST {"msg":"worker 1 ready","ts":1578965754.0314157,"level":"info"}
2020-01-13 17:35:54.071 PST {"request_path":"/api/users/v1/screen_lease/acquire","msg":"Request Served","level":"info","s…
2020-01-13 17:35:54.085 PST level=warn ts=2020-01-14T01:35:54.085Z caller=scrape.go:930 component="scrape manager" scrape…
2020-01-13 17:35:54.286 PST level=warn ts=2020-01-14T01:35:54.286Z caller=scrape.go:930 component="scrape manager" scrape…
```