

LMAO Helps During Outages

Richard Lewis

www.gogorichie.com

- Sr. DevOps Consultant
- 20+ years of working with Operations and Software Development Team
- Organizer of Chicago Monitoring Enthusiast meetup community
- Diehard Chicago White Sox fan!



3Cloud is the largest pureplay Azure services partner in the world.

UNRIVALED IMPACT

No other Microsoft partner invest more resources to create the ultimate Azure experience.

100% AZURE FOCUSED

Our expertise in Data & Analytics, App Innovation, and Cloud Platform position us as the most comprehensive pure play Azure partner in the world.

PROVEN EXPERTISE

750+ Azure projects delivered to 350+ unique U.S. managed clients in 2021 alone!

\$117M

FY22 YTD
ACR Influenced

\$38M

FY22 YTD
ADS Influenced

\$24M

FY22 YTD
APP Influence

> 100%

Annual Growth
over 5 years

650+

FY22 YTD
Co-sell Wins

600

full-time, U.S.
based
Azure experts

600+

Full time U.S.
based Azure
architects

15+

average years consulting
for each technical staff
member



Why LMAO?



Things Needed For A Strategy

Logs
Metrics
Alerts
Observability Tool

```
2020-03-21 05:24:26 - WARNING :: CP Server Thread-11
2020-03-21 05:24:26 - WARNING :: CP Server Thread-11
2020-03-21 05:24:26 - WARNING :: CP Server Thread-11
2020-03-21 05:24:32 - WARNING :: Thread-31 : Failed to
2020-03-21 05:24:32 - WARNING :: Thread-31 : Tautulli
2020-03-21 05:24:32 - DEBUG :: Thread-31 : Tautulli
2020-03-21 05:24:32 - DEBUG :: Thread-31 : Tautulli
2020-03-21 05:24:33 - INFO :: Thread-396
2020-03-21 05:24:36 - ERROR :: Thread-396
2020-03-21 05:24:36 - INFO :: Thread-396
2020-03-21 05:24:36 - DEBUG :: Thread-396
2020-03-21 05:24:56 - WARNING :: CP Server
2020-03-21 05:24:56 - WARNING :: CP Server
2020-03-21 05:24:56 - WARNING :: CP Server
2020-03-21 05:25:26 - WARNING :: CP Server
2020-03-21 05:25:26 - WARNING :: CP Server
2020-03-21 05:25:26 - WARNING :: CP Server
2020-03-21 05:25:26 - WARNING :: Thread-19
2020-03-21 05:25:48 - WARNING :: Thread-19
```



What Are We Talking About?

Platform Support Strategies

Creating a standard for knowledge sharing

Reduction in mean time to resolution (MTTR)

Building Psychological confidence within your team

Logs & Metrics

- Provide insight into the what and when
- What happened?
 - How many errors
 - How many request
- When did it happen?
 - How long was the duration



Alerts

Pages



Or

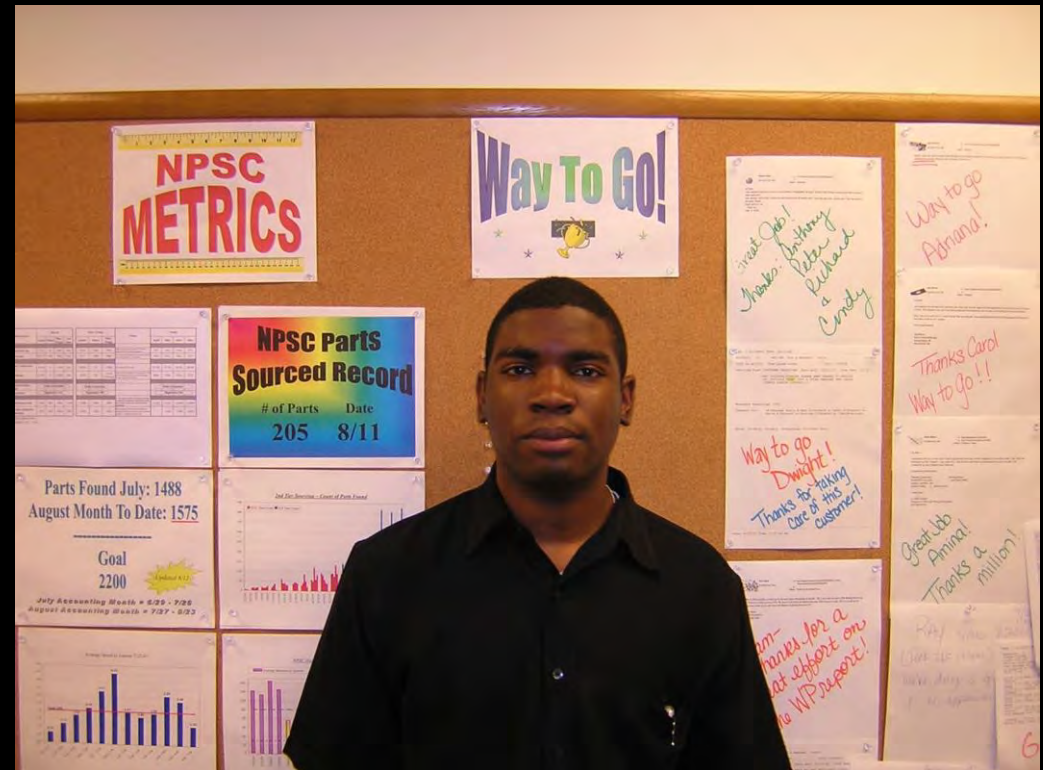
Tickets

ID	SATISFACTION	SUBJECT
#100004	Unoffered	Phone Call to Caller: +1 (602) 908-2849
#100044	Unoffered	LANGUAGE test (dutch) spoke ticket
#100070	Unoffered	Phone Call to Caller: +1 (480) 580-4689
#100072	Unoffered	signature test - spoke

Alert Trauma

Little Richard

- On-Call 365 16hrs a day
- No way to remote connect to office
- No logging framework
- Burned out within 6 months



Managing Alerts Effectively

Scheduling team members appropriately.

Avoid alert fatigue where possible.

Collect data on the alerts and look for ways to reduce them regularly.

Be cautious when introducing new alerts

Observability

- Keep track of KPI's and SLA's
- Monitor usage
- Helps with spotting trends



Monitoring

 Prometheus CNCF Graduated	 cortex CNCF Incubating	 OPENMETRICS CNCF Incubating	 Thanos CNCF Incubating												
															
															
															
															
															

13

All violations - All o...

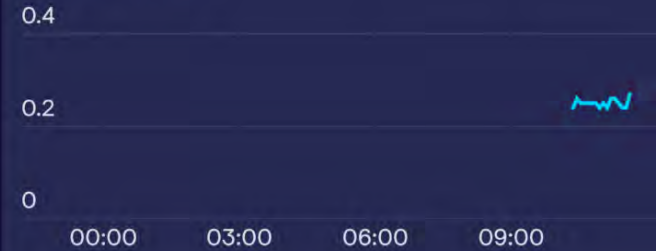
13

Synthetic Failures

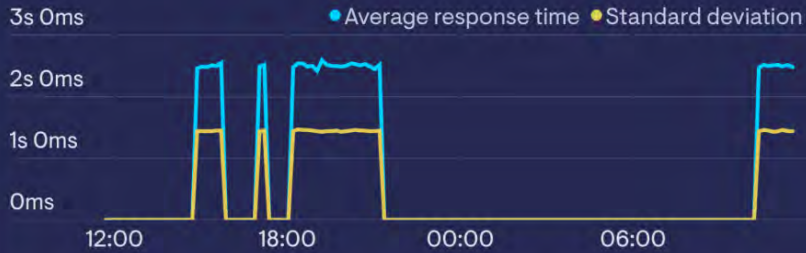
Recent deployments

Re...	Deplo...	Description	Time
1.5	tom	Update README with endpoints	14:39, 25 Jun
1.4	matt	Add instrumentation throughout app	14:39, 25 Jun
1.3	jon	Add endpoint for users to manage account	14:39, 25 Jun
1.2	tomr	Add ping endpoint	14:38, 25 Jun

Apdex score



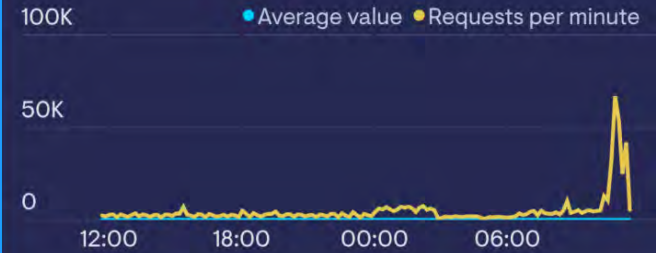
WebTransactions



Errors per minute



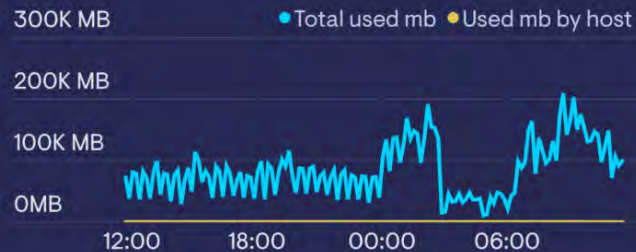
Datastore/all



CPU User time



Physical memory



CPU Time (Percent)



13

All violations - All o...

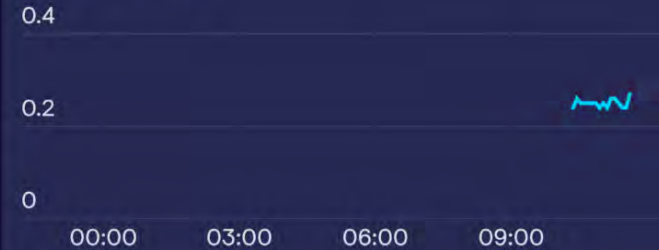
13

Synthetic Failures

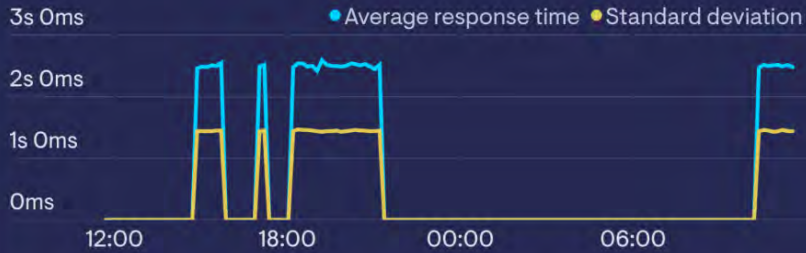
Recent deployments

Re...	Deplo...	Description	Time
1.5	tom	Update README with endpoints	14:39, 25 Jun
1.4	matt	Add instrumentation throughout app	14:39, 25 Jun
1.3	jon	Add endpoint for users to manage account	14:39, 25 Jun
1.2	tomr	Add ping endpoint	14:38, 25 Jun

Apdex score



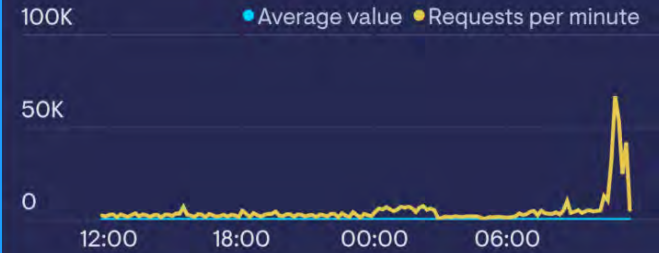
WebTransactions



Errors per minute



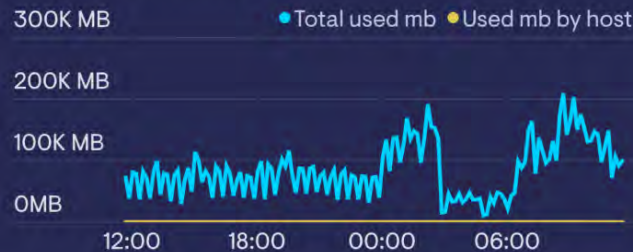
Datastore/all



CPU User time



Physical memory



CPU Time (Percent)



13

All violations - All o...

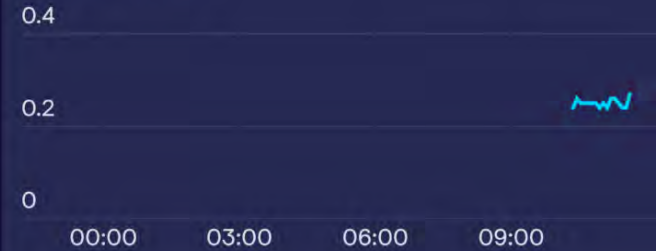
13

Synthetic Failures

Recent deployments

Re...	Deplo...	Description	Time
1.5	tom	Update README with endpoints	14:39, 25 Jun
1.4	matt	Add instrumentation throughout app	14:39, 25 Jun
1.3	jon	Add endpoint for users to manage account	14:39, 25 Jun
1.2	tomr	Add ping endpoint	14:38, 25 Jun

Apdex score



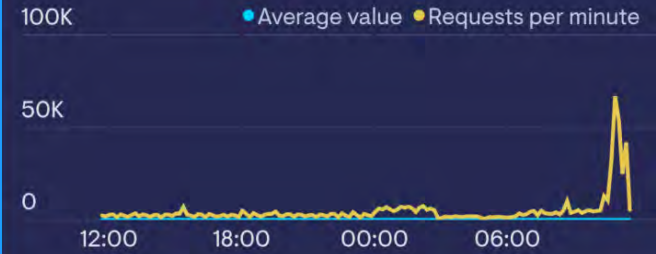
WebTransactions



Errors per minute



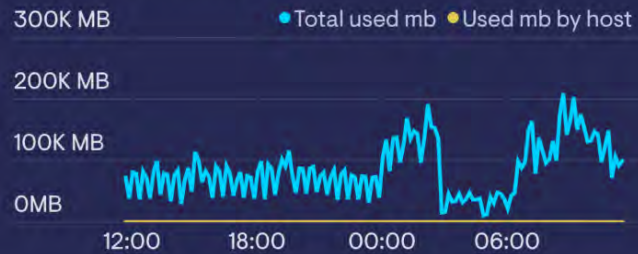
Datastore/all



CPU User time



Physical memory



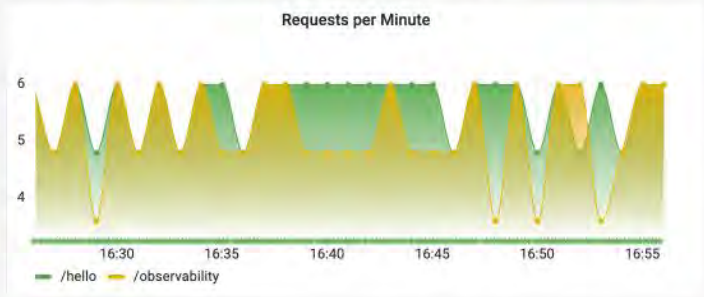
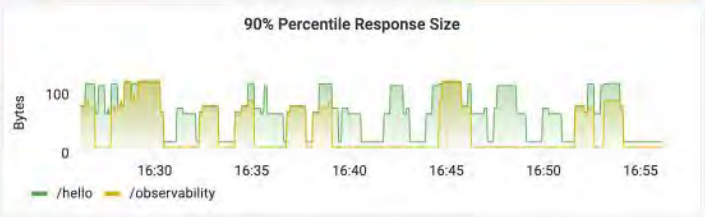
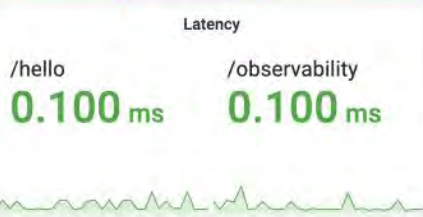
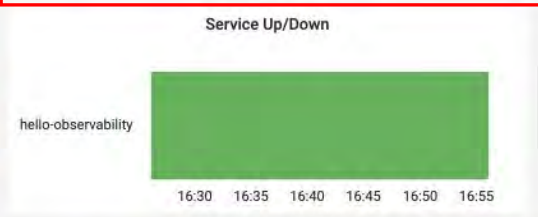
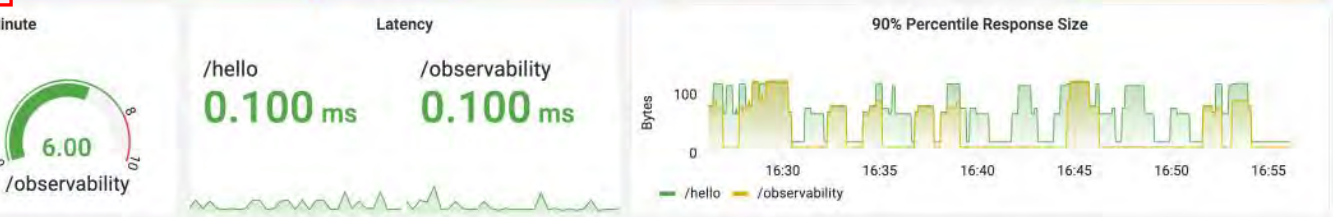
CPU Time (Percent)



Hello Observability

Dashboard for a simple [Spring Boot](#) application, [Hello Observability](#), to demonstrate observability by collecting and correlating logs, metrics and traces with:

- Prometheus
- OpenTelemetry
- Grafana Agent
- Grafana Cloud

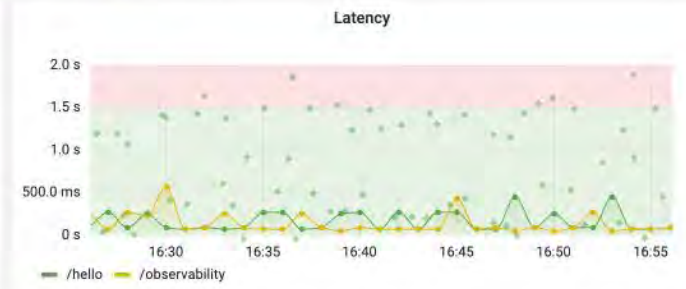
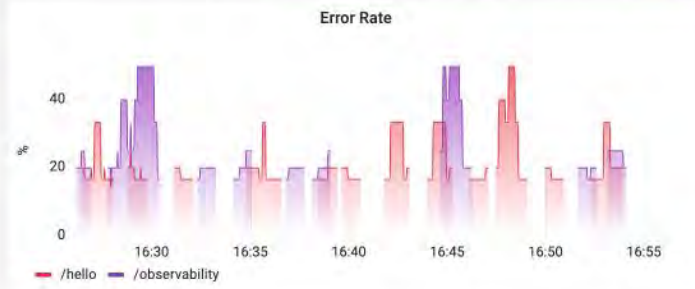
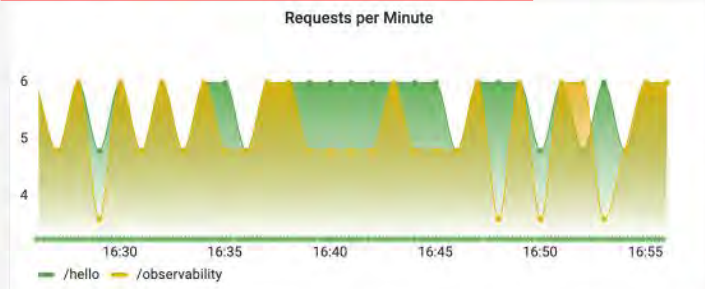
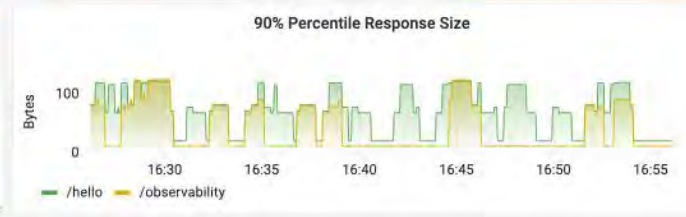
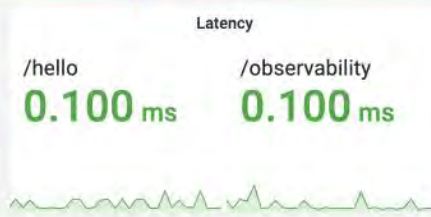
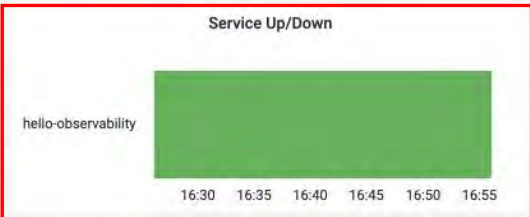
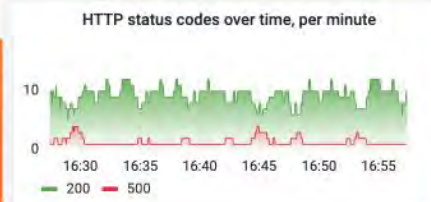


```

> 2022-03-11 22:56:05 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1,text/p
> 2022-03-11 22:56:05 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1,text/
> 2022-03-11 22:56:03 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /hello, client=192.168.80.2, headers=[host:"hello-observability:8080", user-agent:"curl/7.81.0-DEV", accept:"*/*"] traceID=a0edf0cd52c6b757bac0f0ae9c3d5934
> 2022-03-11 22:56:03 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /observability, client=127.0.0.1, headers=[traceparent:"00-a0edf0cd52c6b757bac0f0ae9c3d5934-3d8bbc119703716f-01", host:"localhost:8080", connection:"Keep-Alive", acce
> 2022-03-11 22:56:02 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /observability, client=127.0.0.1, headers=[traceparent:"00-a0edf0cd52c6b757bac0f0ae9c3d5934-3d8bbc119703716f-01", host:"localhost:8080", connection:"Keep-Alive", acc
> 2022-03-11 22:56:02 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /hello, client=192.168.80.2, headers=[host:"hello-observability:8080", user-agent:"curl/7.81.0-DEV", accept:"*/*"] traceID=a0edf0cd52c6b757bac0f0ae9c3d5934
> 2022-03-11 22:55:55 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1;text/p
  
```

Dashboard for a simple Spring Boot application, Hello Observability, to demonstrate observability by collecting and correlating logs, metrics and traces with:

- Prometheus
- OpenTelemetry
- Grafana Agent
- Grafana Cloud



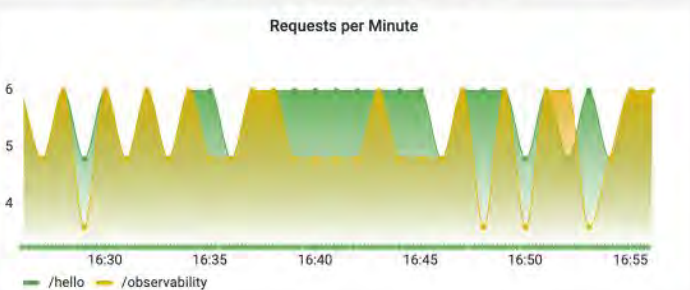
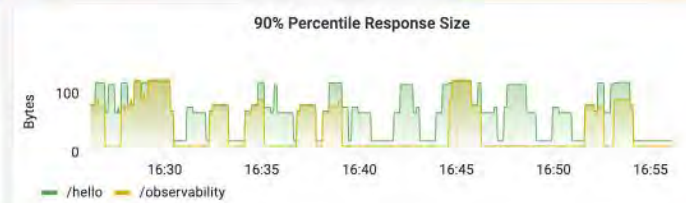
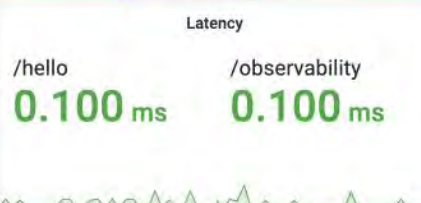
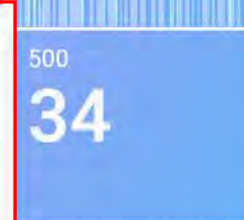
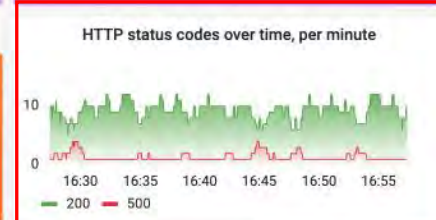
Logs

```

> 2022-03-11 22:56:05 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1,text/p
> 2022-03-11 22:56:05 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1,text/
> 2022-03-11 22:56:03 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /hello, client=192.168.80.2, headers=[host:"hello-observability:8080", user-agent:"curl/7.81.0-DEV", accept:"*/*"]] traceID=a0edf0cd52c6b757bac0f0ae9c3d5934
> 2022-03-11 22:56:03 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /observability, client=127.0.0.1, headers=[traceparent:"00-a0edf0cd52c6b757bac0f0ae9c3d5934-3d8bbc119703716f-01", host:"localhost:8080", connection:"Keep-Alive", acce
> 2022-03-11 22:56:02 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /observability, client=127.0.0.1, headers=[traceparent:"00-a0edf0cd52c6b757bac0f0ae9c3d5934-3d8bbc119703716f-01", host:"localhost:8080", connection:"Keep-Alive", acc
> 2022-03-11 22:56:02 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /hello, client=192.168.80.2, headers=[host:"hello-observability:8080", user-agent:"curl/7.81.0-DEV", accept:"*/*"]] traceID=a0edf0cd52c6b757bac0f0ae9c3d5934
> 2022-03-11 22:55:55 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1;text/p
  
```


Dashboard for a simple Spring Boot application, Hello Observability, to demonstrate observability by collecting and correlating logs, metrics and traces with:

- Prometheus
- OpenTelemetry
- Grafana Agent
- Grafana Cloud



Logs

```

> 2022-03-11 22:56:05 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1,text/p
> 2022-03-11 22:56:05 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1,text/
> 2022-03-11 22:56:03 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /hello, client=192.168.80.2, headers=[host:"hello-observability:8080", user-agent:"curl/7.81.0-DEV", accept:"*/*"]] traceID=a0edf0cd52c6b757bac0f0ae9c3d5934
> 2022-03-11 22:56:03 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /observability, client=127.0.0.1, headers=[traceparent:"00-a0edf0cd52c6b757bac0f0ae9c3d5934-3d8bbc119703716f-01", host:"localhost:8080", connection:"Keep-Alive", acce
> 2022-03-11 22:56:02 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /observability, client=127.0.0.1, headers=[traceparent:"00-a0edf0cd52c6b757bac0f0ae9c3d5934-3d8bbc119703716f-01", host:"localhost:8080", connection:"Keep-Alive", acc
> 2022-03-11 22:56:02 - o.s.w.f.CommonsRequestLoggingFilter - Before request [GET /hello, client=192.168.80.2, headers=[host:"hello-observability:8080", user-agent:"curl/7.81.0-DEV", accept:"*/*"]] traceID=a0edf0cd52c6b757bac0f0ae9c3d5934
> 2022-03-11 22:55:55 - o.s.w.f.CommonsRequestLoggingFilter - After request [GET /metrics, client=192.168.80.3, headers=[host:"hello-observability:8080", user-agent:"GrafanaAgent/v0.23.0", accept:"application/openmetrics-text; version=0.0.1;text/p
  
```


Preparing
The Team

Have a playbook

Practice for outages

What Should Be In It?

- Playbook should be located where the on-call team will be able to access it quickly.
- Links to application related observability tools
- Details about the golden signals for application.
- Relevant information from previous outages.
- Application owner contact details
- Anything else you might find important.



Preparation and Training

Everybody during a
tornado drill:



Practice Chaos Engineering

Concept created by Netflix.

The goal Increase resiliency

Identify and address single points of failure early.

Gives you the ability to test your documentation and processes.



Postmortems



Recap of the Outage

- Should be done within a day or so post incident
- What went right?
- What went wrong?
- Where do we get lucky?

“ *The cost of failure is education.* ”

Devin Carraway

Take Aways

- Have a LMAO strategy in place.
- Have the appropriate documentation ready to go.
- Update documentation regularly.
- Avoid alert fatigue.
- Run readiness preparation drills.



Thank You

Richard Lewis

Twitter: @gogorichie

Email: Richard@gogorichie.com