



Ze

Z E B R I U M

# Autonomous Monitoring

*USING MACHINE LEARNING*

Larry Lancaster  
Founder and CTO  
Zebrium



# Machine data is my life

- NetApp - *Engineering Informatics*
- EMC / Data Domain - *Product Analytics\**
- Glassbeam - *Chief Technology Officer\**
- Nimble Storage - *Chief Data Scientist\**
- Zebrium - *Founder and CTO\**

\* Companies where I got to use Vertica :)





ZEBRIUM

# Our vision

## Characterize incidents before I notice

**Automatically Detect Incidents**  
Without Setting Up Manual Alert Rules



**Automatically Find Root Cause**  
Without Manually Searching Across  
GBs of Logs

# Log monitoring today



SLOW (MTTR)  
FRAGILE (FORMATS CHG)  
ANNOYING (ALERT FATIGUE)

HUMAN-DRIVEN





# Logs are used for RC

---

...so why aren't they better at helping us monitor?



# What keeps logs "dumb"?

---

Logs are stuck in "index + search".





20 YEARS AGO

TODAY

Shrink-Wrap:

*1 incident 1 user*

*1 incident 1 monolith*

*1 incident 10 logfiles*

Log use for root-cause:

*index and search*

SaaS:

*1 incident 100K users*

*1 incident 100 services*

*1 incident 1K logstreams*

Log use for root-cause:

***still index and search(!)***



We believe

The future cannot be "index + search".  
It simply cannot scale.





ZEBRIUM

# Autonomous Monitoring

Quickstart Incidents Browse Alert Rules -08:00 (browser) ⚙ Sign Out Give Feedback

← JUN 25 2019 04:17:37 Jun 25 04:17:37 mars systemd[1]: Stopped PostgreSQL RDDBMS. 📌 📄

**Trigger** Show Me

Automatically discovered by anomaly detection.

**Probable Root Cause**

Jun 25 04:17:37 mars systemd[1]: Stopped PostgreSQL RDDBMS.

etype postgresql logtype syslog container\_image zebrium/psql:rel\_20191025123422 deployment\_name atlassian115 host host008

namespace\_name default pod\_name postgres\_master\_67d32e34-bfe3

Jun 25 04:17:37 mars systemd[1]: Stopping PostgreSQL Cluster 9.5-main...

etype postgresql\_cluster logtype syslog container\_image zebrium/psql:rel\_20191025123422 deployment\_name atlassian115

host host008 namespace\_name default pod\_name postgres\_master\_67d32e34-bfe3

**Symptoms Detected**

2019-06-25 04:17:39,129 commons-pool-EvictionTimer WARN [o.a.commons.dbcp2.BasicDataSource] An internal object pool swallowed an Exception.

etype internal\_object\_pool\_swallowed logtype jira container\_image zebrium/jira:rel\_20191025123422 deployment\_name atlassian115

host host007 namespace\_name default pod\_name jira\_master\_7fed321a-bfe3



ZEBRIUM

# Autonomous Monitoring

## Probable Root Cause

[Drilldown to Incident Events →](#)

Seen In: Pod: pod-delete-ugi8e7-89jmd

Deployment Name: zebrium-k8s-demo

- 2020-01-27 20:36:54.626394 Step: Get a list of all pods from given namespace
- 2020-01-27 20:36:54.694865 Step: Initialize deletion list
- 2020-01-27 20:36:54.773292 Step: Select a random pod to kill
- 2020-01-27 20:36:54.851302 Step: Construct the deletion list with single random pod
- 2020-01-27 20:36:56.482794 Step: Kill application pod

## Symptoms

All 6

Other 2

Pod: pod-delete-ugi8e7-89jmd 1

Pod: carts-745cc4588d-4zrxf 3

- 2020-01-27 20:37:01.787 WARN [carts,b3376c3cc0068994,f68b1336165f2b5a,true] 6 --- [p-nio-80-exec-5] org.mongodb.driver.connection : Got socket exception on connection [connectionId(localValue:6, serverValue:2)] to carts-db:27017. All connections to carts-db:27017 will be closed.
- 2020-01-27 20:37:01.815 ERROR [carts,,] 6 --- [p-nio-80-exec-5] o.a.c.c.C.[.][.][dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed; nested exception is org.springframework.data.mongodb.UncategorizedMongoDbException: Prematurely reached end of stream; nested exception is com.mongodb.MongoSocketReadException: Prematurely reached end of stream] with root cause
- 2020-01-27 20:37:06.556128 Step: Wait for the interval timer
- 2018-1-27T20:37:09.916697 WARN [1206 docker\_container.go:216] Cannot create symbolic link because container log file doesn't exist!
- 2018-1-27T20:37:09.916855 ERROR [1206 remote\_runtime.go:213] StartContainer "754e35b492032e6582282405f1669d09d15617ee544d8549bdfbc7c64841ef17" from runtime service failed: rpc error: code = Unknown desc = failed to start container "754e35b492032e6582282405f1669d09d15617ee544d8549bdfbc7c64841ef17": Error response from daemon: cannot join network of a non running container: ecff3eebaa9ae00e476408baa4b2a181c7e411f8cfff4fe46c648c3c7ed5c8ac
- 2020-01-27 20:37:22.061 ERROR [carts,,] 6 --- [p-nio-80-exec-2] o.a.c.c.C.[.][.][dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed; nested exception is org.springframework.data.mongodb.UncategorizedMongoDbException: Query failed with error code 11600 and error message 'interrupted at shutdown' on server carts-db:27017; nested exception is com.mongodb.MongoQueryException: Query failed with error code 11600 and error message 'interrupted at shutdown' on server carts-db:27017] with root cause





ZEBRIUM

# Autonomous Monitoring

## Possible Root Cause

[Drilldown to Incident Events →](#)

Seen In: Pod: disk-fill-hwizb7-cb62p      Deployment Name: zebrium-k8s-demo

2020-03-03 20:32:11.604478 Step: Wait for the specified ramp time after injecting chaos

2020-03-03 20:32:21.748439 Step: Update the chaos result CR to reflect EOT (End of Test)

details = > {"changed": true, "checksum": "860684b046d8d5c70600cd392ced04f6c66e0143", "dest": "./chaos-result.yml", "gid": 0, "mode": "0644", "owner": "root", "size": 315, "src": "/root/.ansible/tmp/ansible-tmp-1583267540.15-161408044609654/source"}  
127.0.0.1 : ok = 54    changed = 27    unreachable = 0    failed = 0

2020-03-03 20:32:23.543246 Step: Apply the chaos result CR

details = > {"changed": true, "cmd": "kubectl apply -f chaos-result.yml -n sock-shop", "delta": "0:00:01.407630", "end": "2020-03-03 20:32:22.043148", "rc": 0, "start": "2020-03-03 20:32:22.043148", "stderr": "", "stderr\_lines": [], "stdout": "chaosresult.litmuschaos.io/sock-chaos-litmuschaos.io/sock-chaos-disk-fill configured"}  
127.0.0.1 : ok = 54    changed = 27    unreachable = 0    failed = 0

META: ran handlers

2020-03-03 20:32:23.551821 \*\*\*\*\* RELAX, EXPERIMENT ENDS! \*\*\*\*\*

127.0.0.1 : ok = 54    changed = 27    unreachable = 0    failed = 0



## Symptoms

All 334



# Metrics Complete the Picture

- One-Stop Shop for Incident RC
  - Ingest / structure logs
  - Ingest / structure Prometheus metrics
  - Perform A/D on both
  - Cross-correlate metric and log anomalies
  - Create incidents automatically





# Recent validation



Mayadata reproduced a slew of real-world incidents from real Kubernetes clusters using Litmus.

Zebrium immediately detected and root-caused **100%** of these incidents... no training, config, metadata required.





# Ze: How it works

No information included or required about:

- Known prefix formats
- Specific logtype keywords
- Event grammar / syntax

***We embrace free-text logs***





# Ze: How it works

No information included or required about:

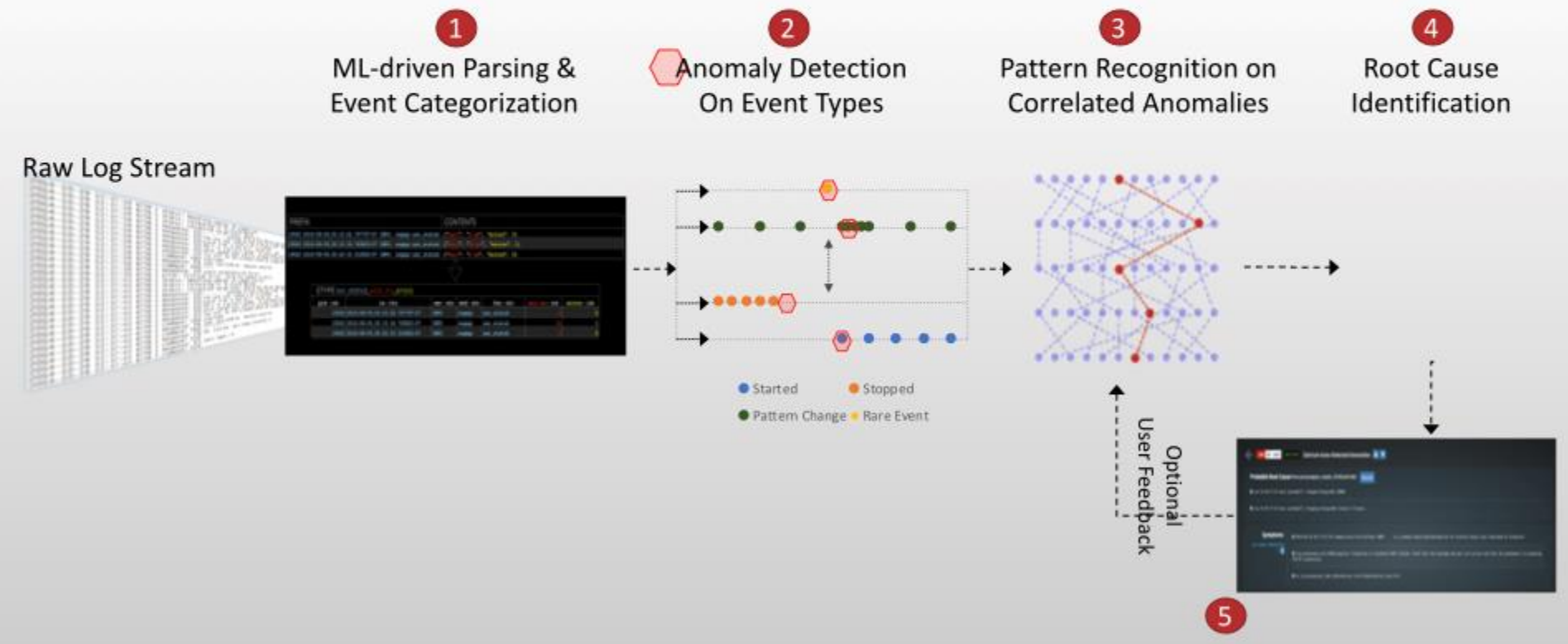
- Connectors, knowledge bases
- Specific application behaviors
- Specific semantic keywords

***Works great on bespoke app or stack***



ZEBRIUM

# Ze: How it works







# Ze: How it works

Incident detection needs relationally structured data, including logs.



ZEBRIUM

# Ze: How it works

```
19563 2016-08-09,00:13:02.797797-07 INFO: regmgr:auto_stop_axr_stats: Out -----
19562 2016-08-09,00:15:34.708807-07 INFO: io:f_MDOP_do_ckpt: group-id 11: ckpt for memory scrubber coff 1194692848128 cycles 21
19562 2016-08-09,00:15:34.708807-07 INFO: config.db:scan_md_chg_log: _my_next_md_entry=1, tail=1 comp=axr
```

```
$ zq "select table_name from information_schema.tables where table_name ilike '%scrubber%';"
```

```
table_name
```

```
=====
t_group_hourly_update_for_disk_scrubber_coff_cycles_progress
t_group_disk_scrubber_cycles_progress_percent_total_user
t_group_ckpt_for_memory_scrubber_coff_cycles
```

```
(3 rows)
```

```
$ zq "select * from t_group_ckpt_for_memory_scrubber_coff_cycles;" | head -3
```

```
sev | fac |          ttz          | us | dte   | pid | fun           | _coff_int | _cycles_int | _group_id_int
=====
INFO | io  | 2016-08-09 00:15:34-07 | 708807 | 2016-08-09 | 19562 | f_MDOP_do_ckpt | 1194692848128 | 21 | 11
```





# VERTICA: MPP Column Store

- Building on top of machine data is hard.
  - No defined schema, semantics, or interfaces
- Structured data can use RDBMS.
  - Ze uses a scale-out MPP RDBMS: VERTICA



# VERTICA: Huge Efficiencies

- Data perfectly suited for columnar encoding
  - Slowly-varying attributes dimensioning high-cardinality events and stats
  - 10X-20X footprint redux on disk is common
- Late materialization propagates these benefits
- Scale-out, MPP arch suits petascale workloads
- Business-level expectations: SQL, ODBC, ACID





# VERTICA: Cloud-Native SaaS

- Vertica in Eon Mode = shared object storage
  - Scale storage and compute separately
  - Load and query from different subclusters(!!!)
- Analytics: Vertica In-Database ML
- Sequence analytics: Perfectly suited to logs
- Timeseries analytics: Perfectly suited to stats



Try it out!

email: `larry@zebrium.com`

twitter: `stochastimus@twitter.com`

**TRY IT OUT!**

`www.zebrium.com`