Telemetry & Observability

Diego Pacheco

About Me

- Cat's Father
- Principal Software Architect
- Agile Coach
- SOA/Microservices Expert
- DevOps Practitioner
- Speaker
- Author

diegopacheco

@diego_pacheco

http://diego-pacheco.blogspot.com.br/

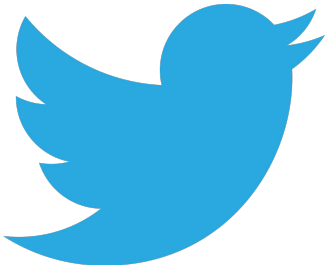
Observability

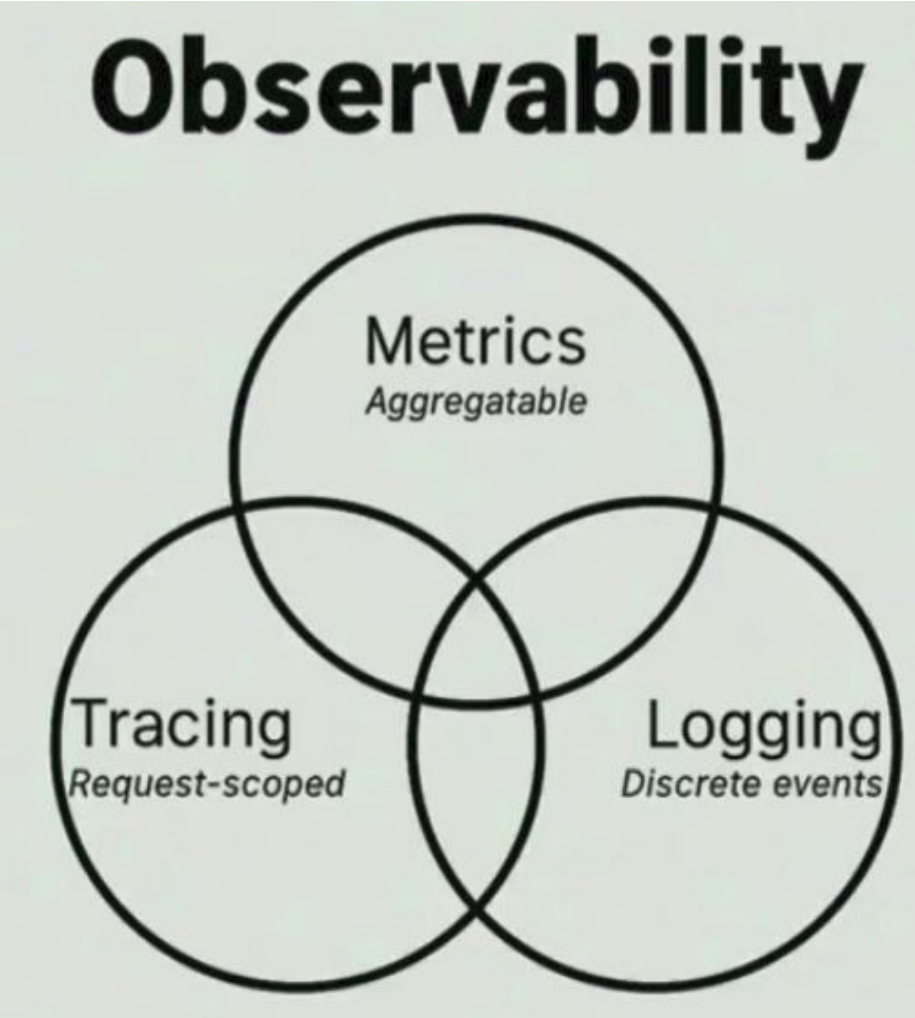
Observability

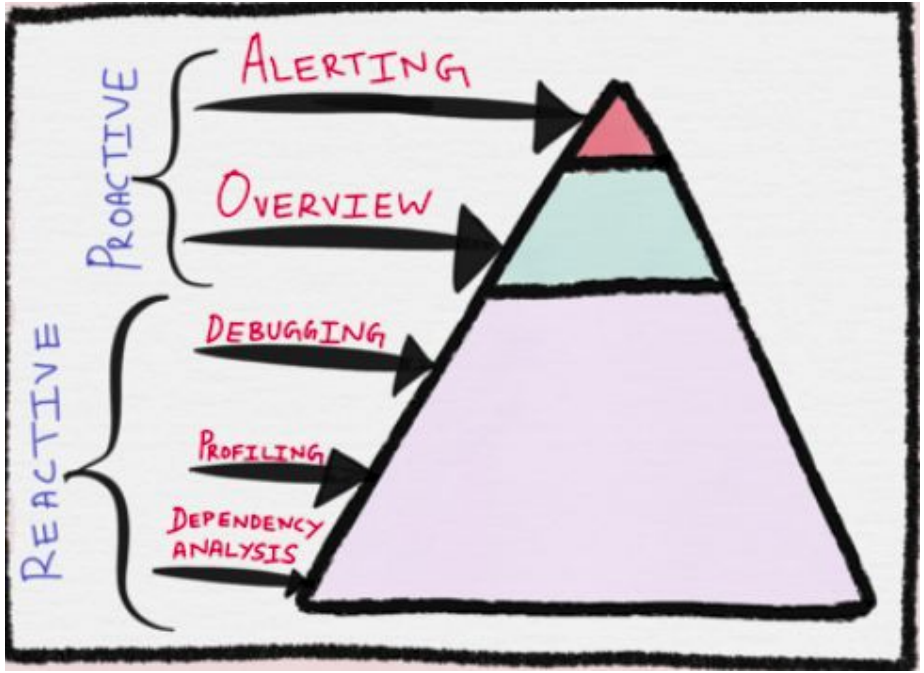
From Wikipedia, the free encyclopedia

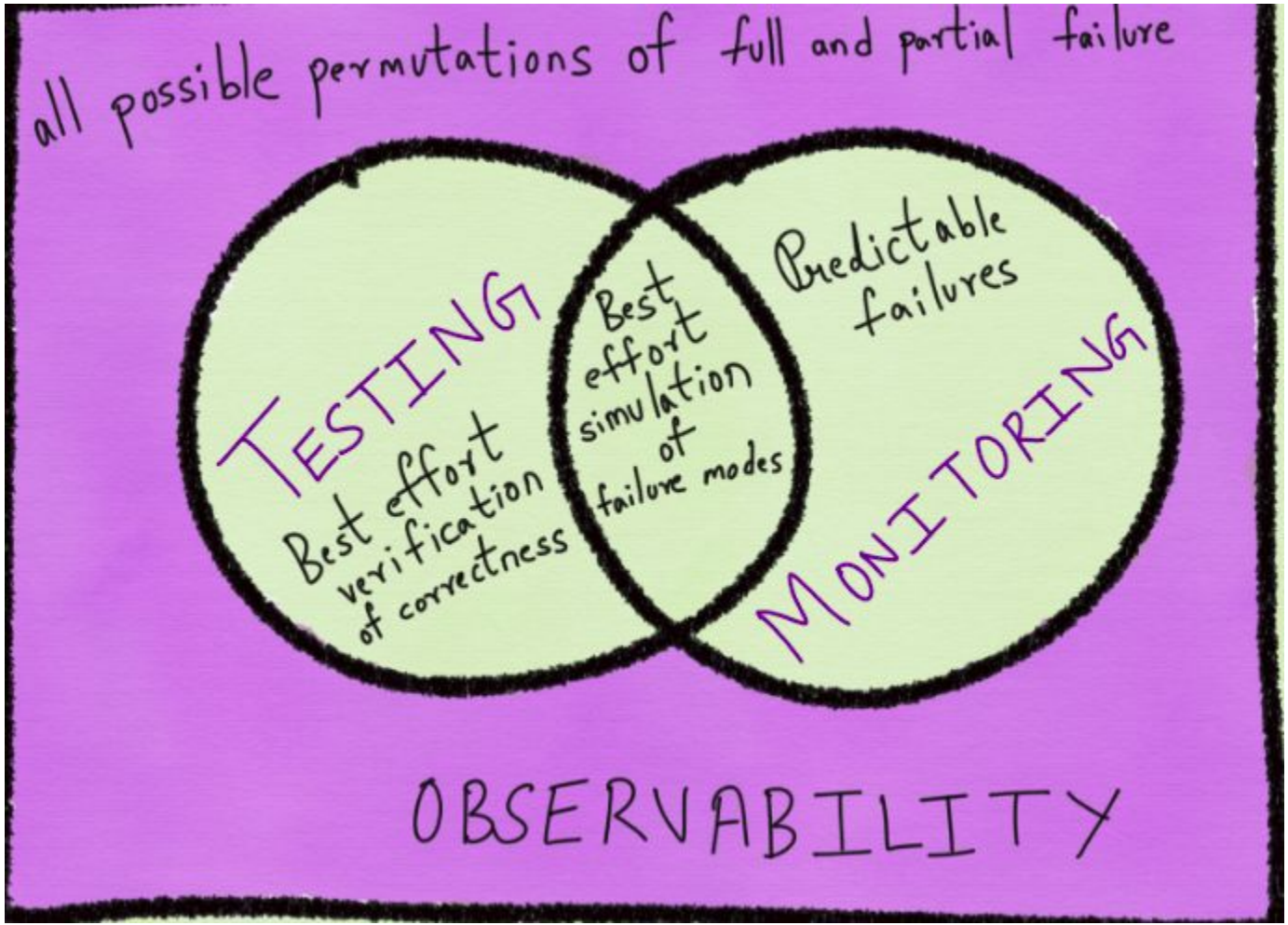
For the concept in quantum mechanics, see observable.

In control theory, **observability** is a measure of how well internal states of a system can be inferred from knowledge of its external outputs. The observability and controllability of a system are mathematical duals. The concept of observability was introduced by Hungarian-American engineer Rudolf E. Kálmán for linear dynamic systems.[1][2]

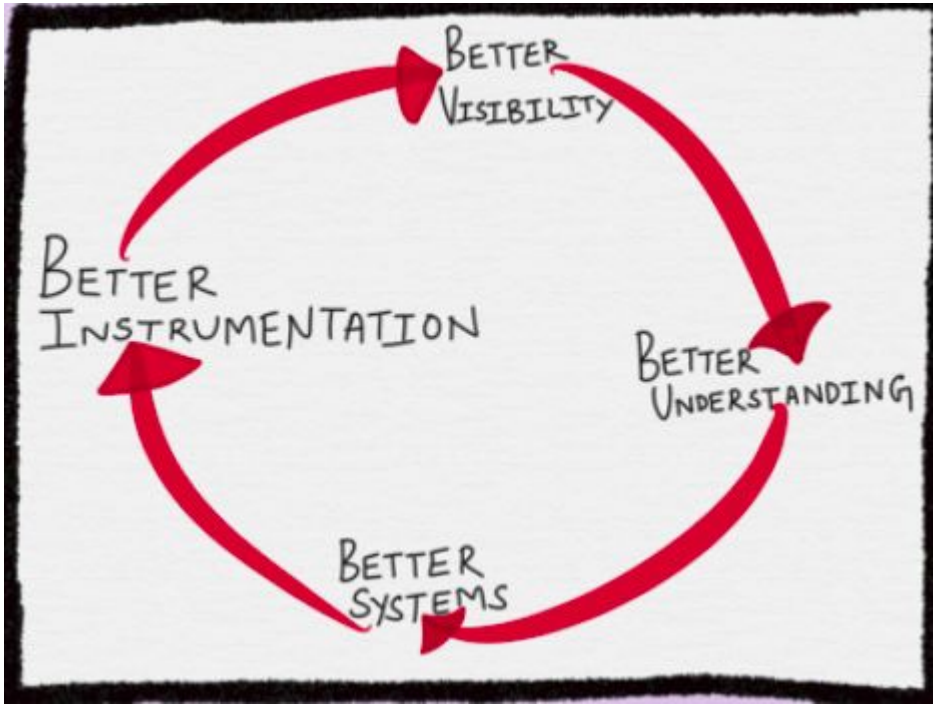


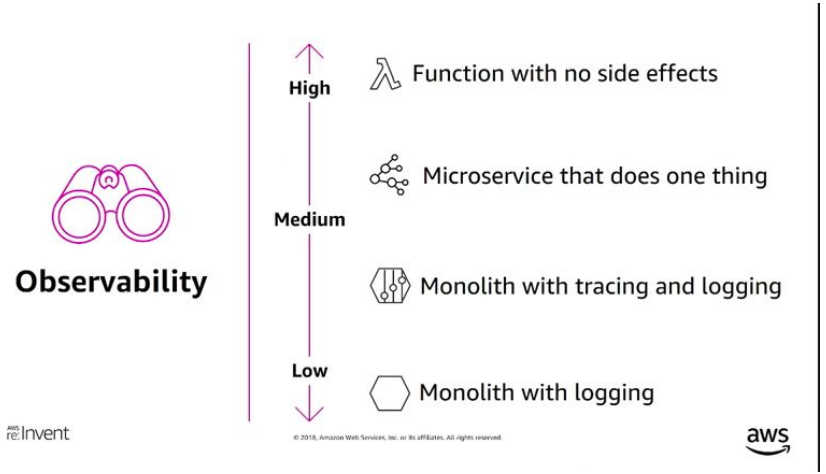

Observability
Metrics
Aggregatable
Tracing
Request-scoped
Logging
Discrete events


Proactive
Alerting
Overview
Reactive
Debugging
Profiling
Dependency analysis


all possible permutations of full and partial failure
TESTING
Best effort verification of correctness
Best effort simulation of failure modes
Predictable failures
MONITORING
OBSERVABILITY

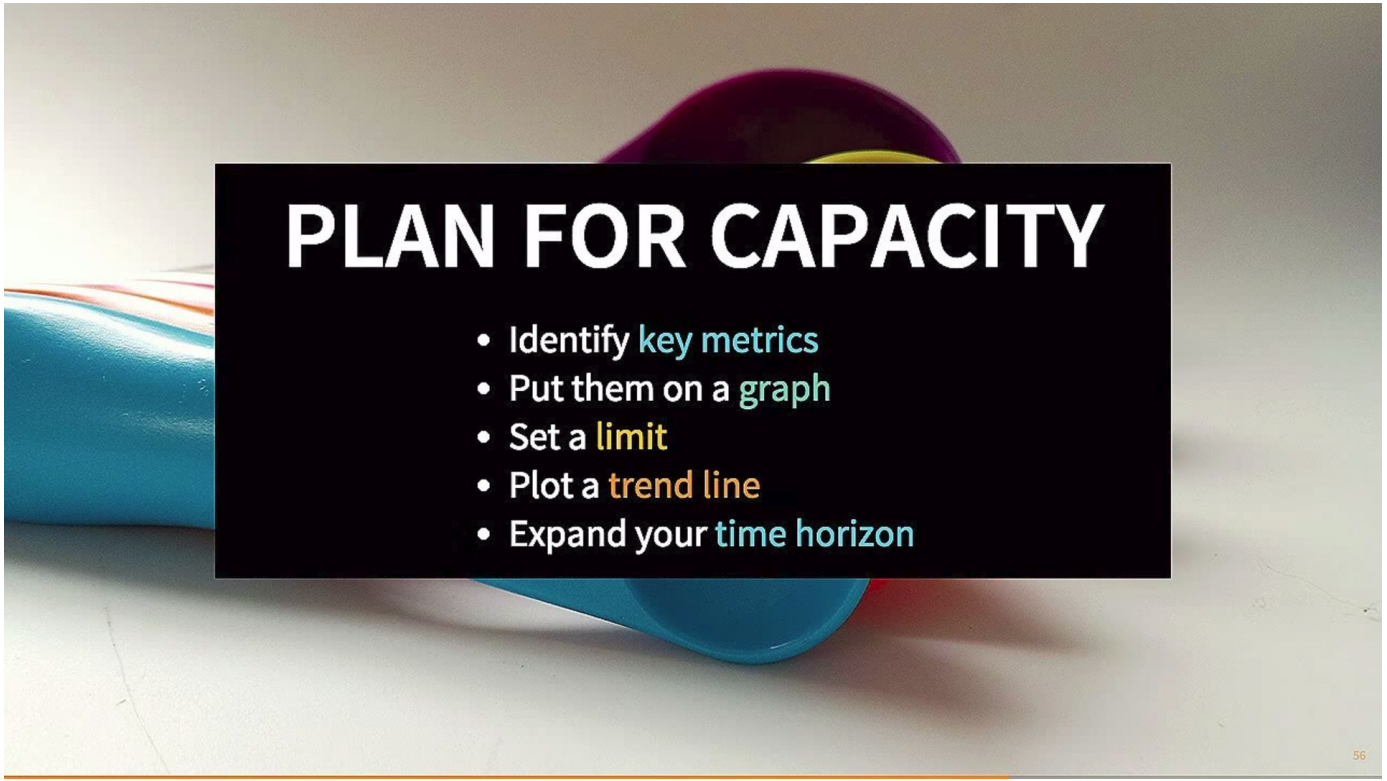
Observability Cycle



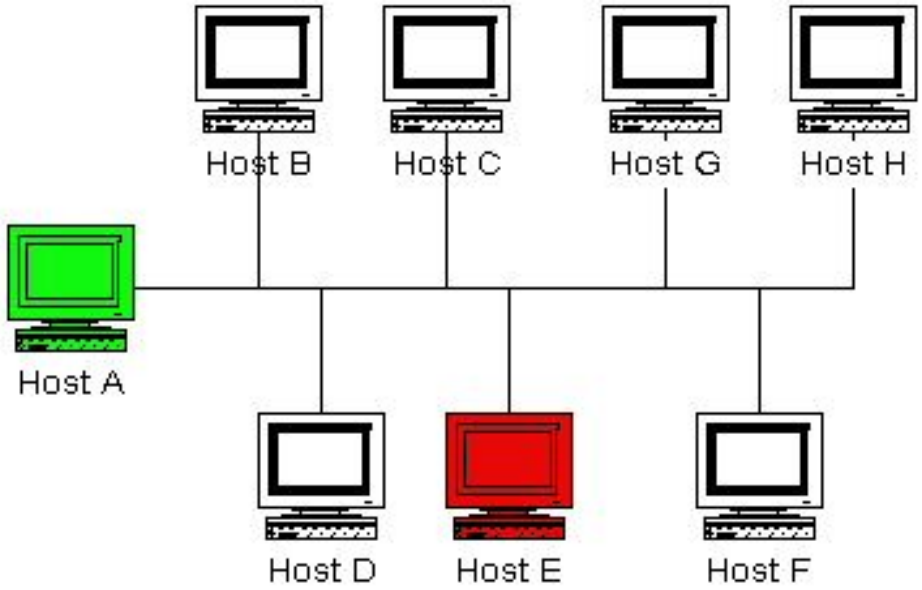


Alerts for Alerts are WRONG!

Capacity Planning



Host Monitoring



sensu

ZABBIX

Remember the Cloud...

Pets

Unique and Indispensable

- GUI Driven
- Ticket Based
- Hand Crafted
- Reserved
- Scale-Up
- Smart Hardware
- Proprietary
- "Waterfall Ops"
- . . .

vs

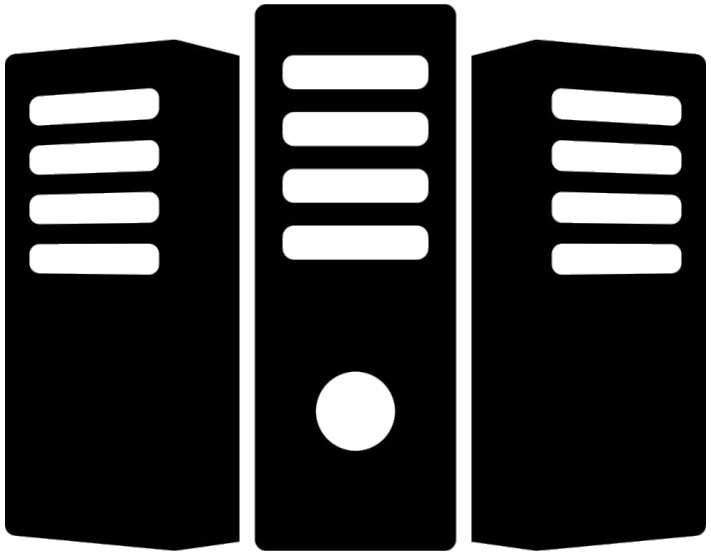
Cattle

Disposable, One of the Herd

- API Driven
- Self Service
- Automated
- On Demand
- Scale-Out
- Smart Apps
- Open Source
- Agile DevOps
- . . .

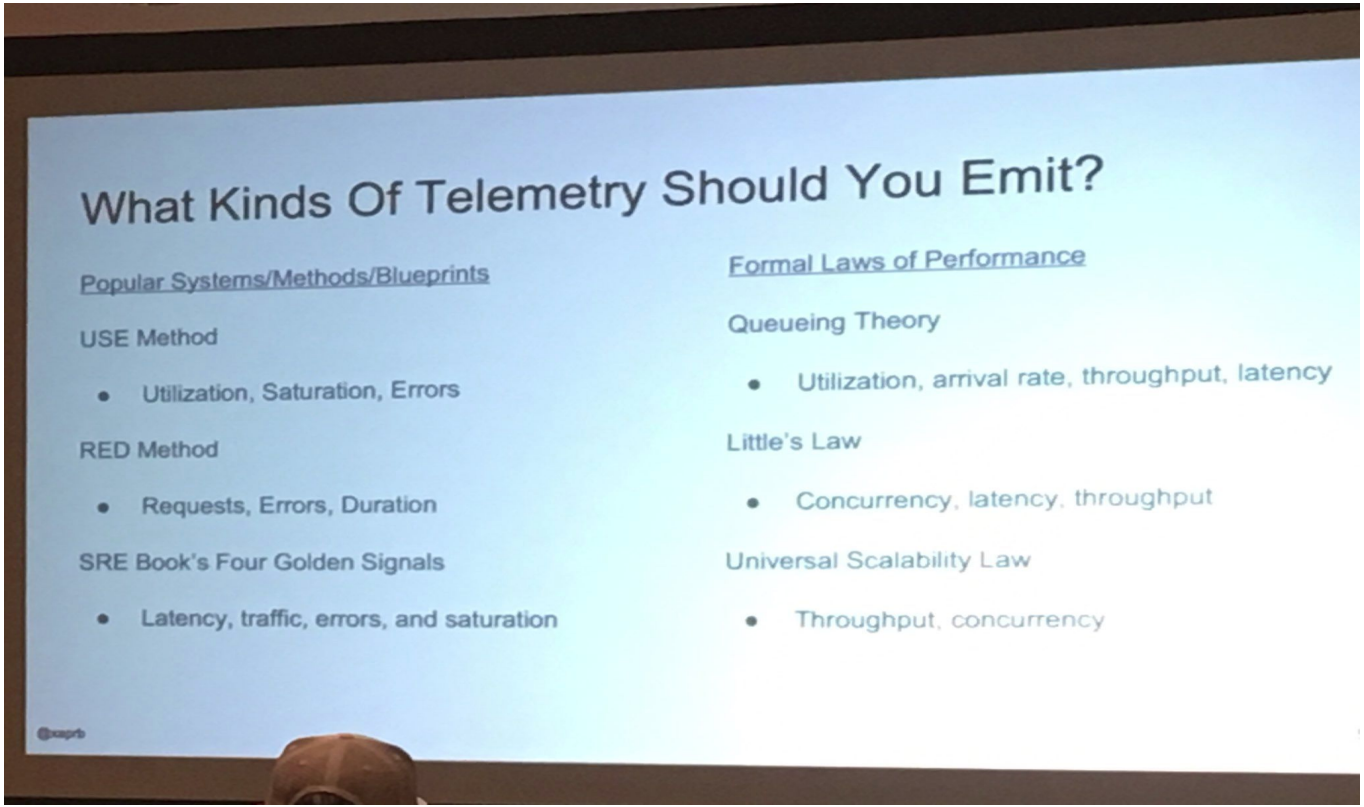
Time Series Databases

Application Expose Metrics

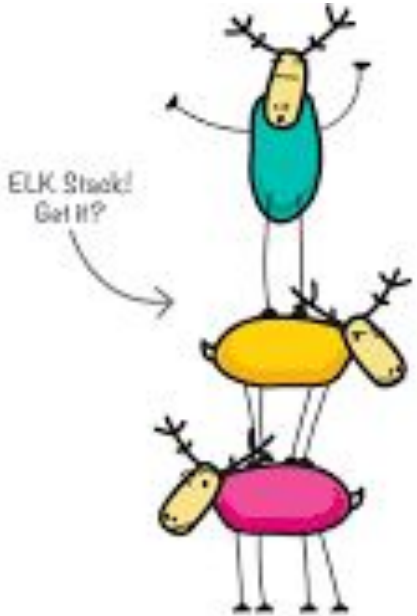


- Black box monitoring dont work any more
- Health/Ping/UP Status are not enough
- Need to send All kinds of metrics like:
  - OS(CPU, Memory, Disk, Network)
  - App(Latency, Requests, Custom)
  - Biz (Transactions, Purchases, etc..)
- This metrics cannot be processed in place
- They need to goto a time series database
- Where we can do Aggregation and Signal Processing and CO RELATE events.

Expose what?


What Kinds Of Telemetry Should You Emit?
Popular Systems/Methods/Blueprints
USE Method
• Utilization, Saturation, Errors
RED Method
• Requests, Errors, Duration
SRE Book's Four Golden Signals
• Latency, traffic, errors, and saturation
Formal Laws of Performance
Queueing Theory
• Utilization, arrival rate, throughput, latency
Little's Law
• Concurrency, latency, throughput
Universal Scalability Law
• Throughput, concurrency

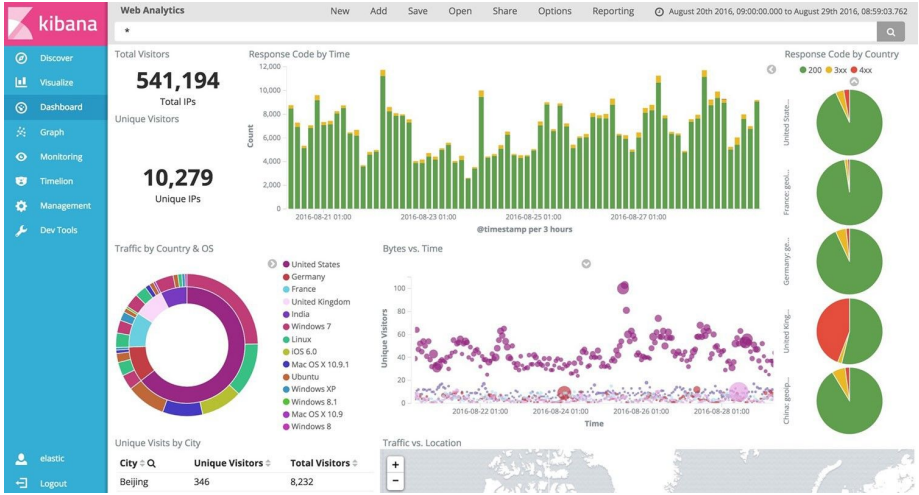
ELK



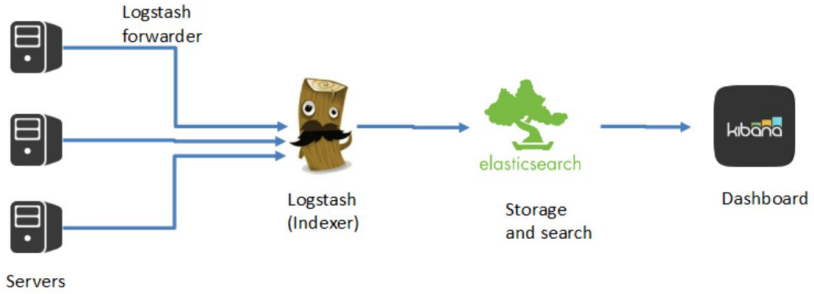
E Elasticsearch

L Logstash

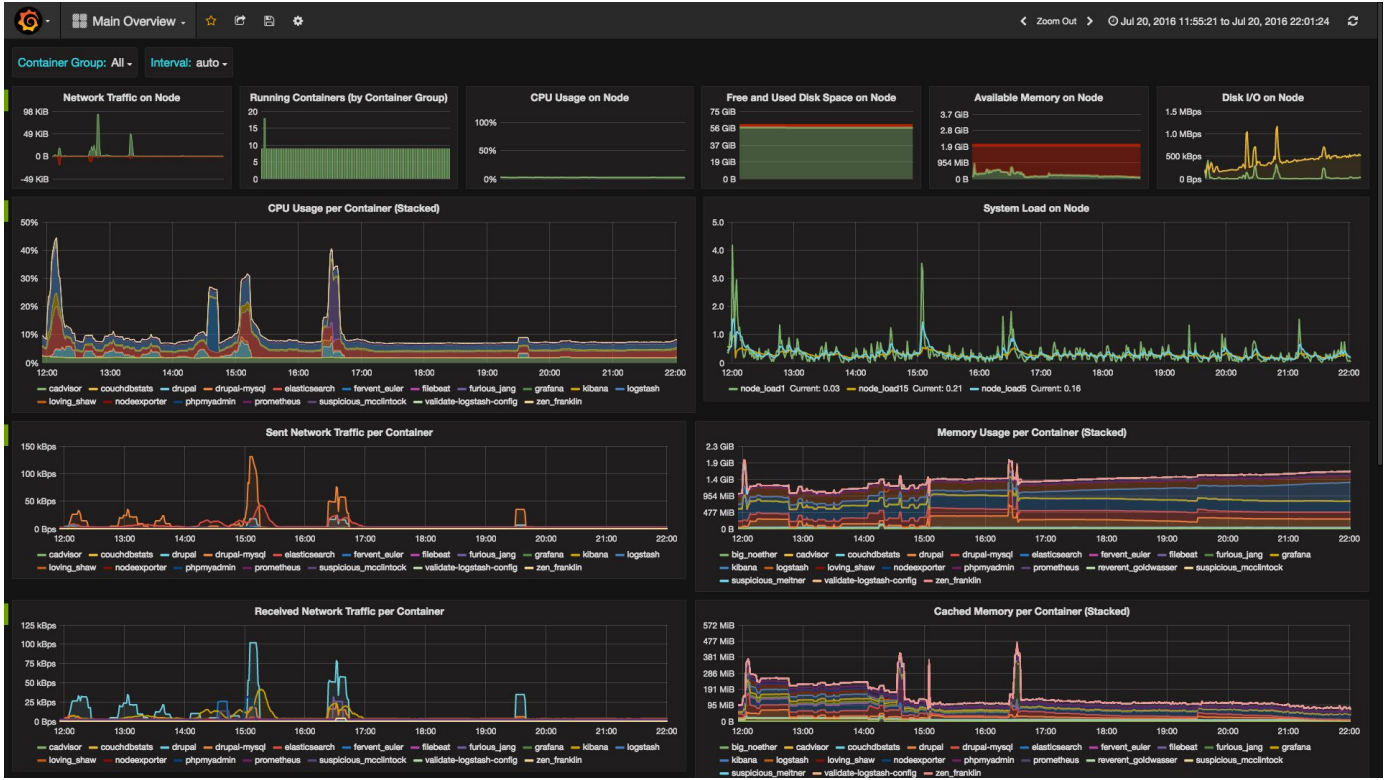
K Kibana



ELK Architecture

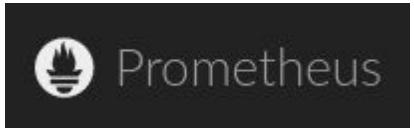


Grafana



Grafana

Prometheus



Dimensional data

Prometheus implements a highly dimensional data model. Time series are identified by a metric name and a set of key-value pairs.

Powerful queries

PromQL allows slicing and dicing of collected time series data in order to generate ad-hoc graphs, tables, and alerts.

Great visualization

Prometheus has multiple modes for visualizing data: a built-in expression browser, Grafana integration, and a console template language.

Efficient storage

Prometheus stores time series in memory and on local disk in an efficient custom format. Scaling is achieved by functional sharding and federation.

Simple operation

Each server is independent for reliability, relying only on local storage. Written in Go, all binaries are statically linked and easy to deploy.

Precise alerting

Alerts are defined based on Prometheus's flexible PromQL and maintain dimensional information. An alertmanager handles notifications and silencing.

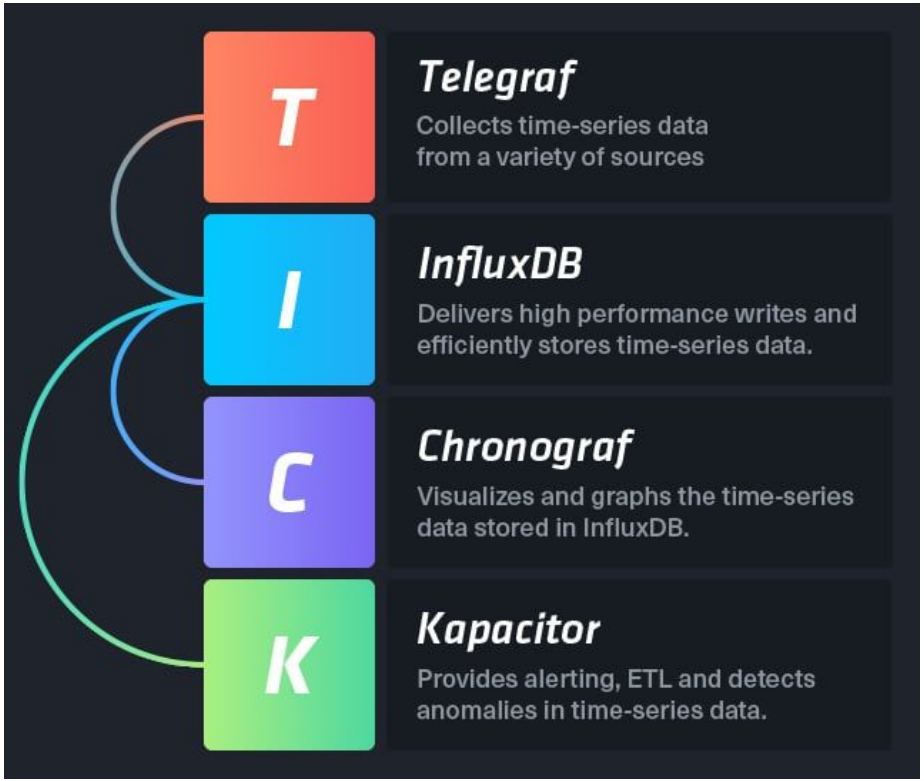
Many client libraries

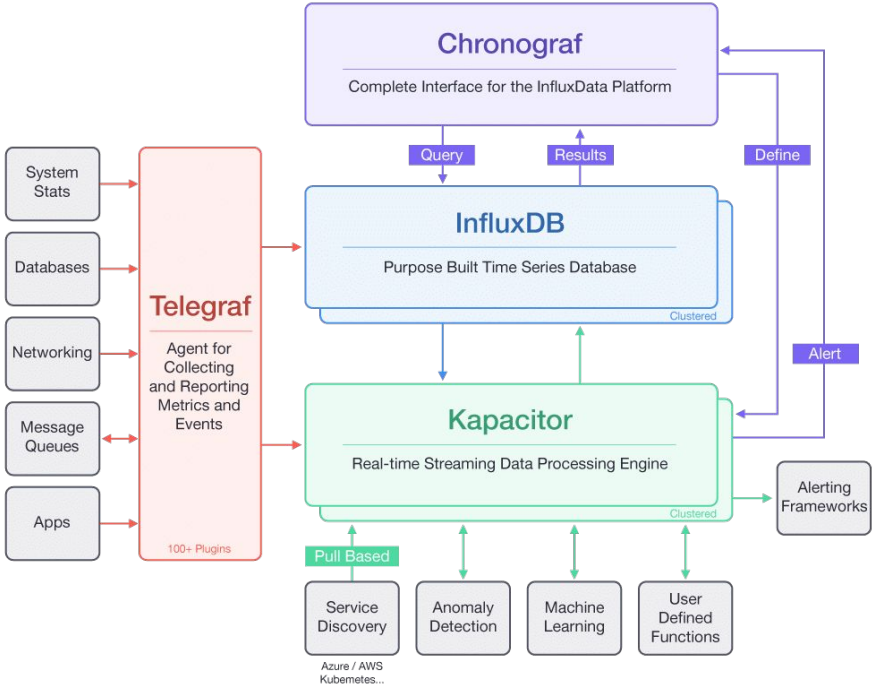
Client libraries allow easy instrumentation of services. Over ten languages are supported already and custom libraries are easy to implement.

Many integrations

Existing exporters allow bridging of third-party data into Prometheus. Examples: system statistics, as well as Docker, HAProxy, StatsD, and JMX metrics.

Influx - TICK

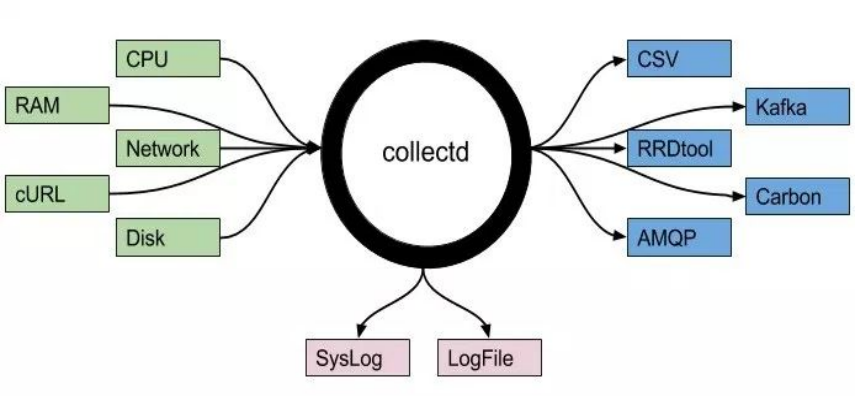

T
Telegraf
Collects time-series data from a variety of sources
I
InfluxDB
Delivers high performance writes and efficiently stores time-series data.
C
Chronograf
Visualizes and graphs the time-series data stored in InfluxDB.
K
Kapacitor
Provides alerting, ETL and detects anomalies in time-series data.


Chronograf
Complete Interface for the InfluxData Platform
Query
Results
Define
System Stats
Databases
Networking
Message Queues
Apps
Telegraf
Agent for Collecting and Reporting Metrics and Events
100+ Plugins
InfluxDB
Purpose Built Time Series Database
Clustered
Alert
Kapacitor
Real-time Streaming Data Processing Engine
Clustered
Alerting Frameworks
Pull Based
Service Discovery
Azure / AWS Kubemetes...
Anomaly Detection
Machine Learning
User Defined Functions

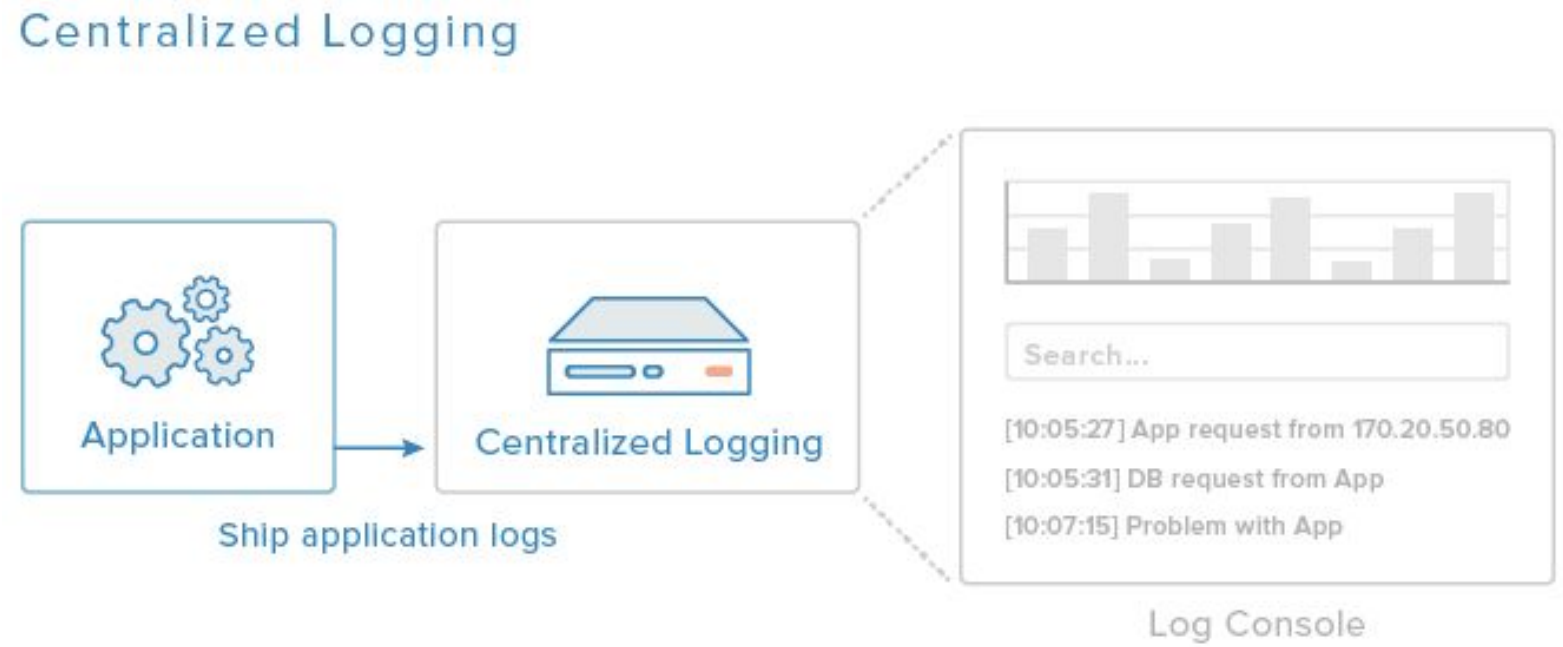
CollectD

```
Hostname    "test.example.com"
LoadPlugin interface
LoadPlugin load
LoadPlugin memory
LoadPlugin network
LoadPlugin logfile

<Plugin logfile>
        LogLevel info
        File "/var/log/collectd.log"
</Plugin>
<Plugin interface>
    Interface "eth0"
    IgnoreSelected false
</Plugin>
<Plugin network>
 Server "172.20.1.10" "7070"
</Plugin>


Logstash Config :

input {
udp {
port => 7070
buffer_size => 1452
codec => collectd { }
type => "collectd"
}
}
output {
stdout { codec => json }
}
```

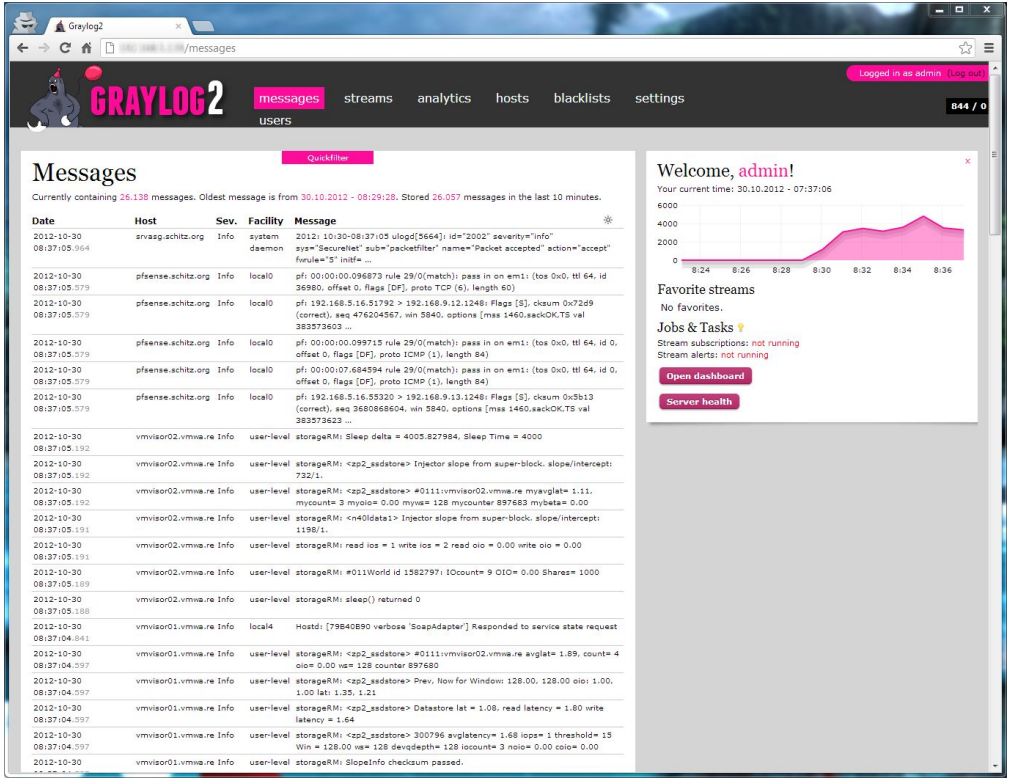
- OLD
- Everybody use it
- Lots of Plugins and plug and play metrics
- HARD TO TEST
- HARD TO TROUBLESHOOT
- Need to RESET in order to change configs(FS)



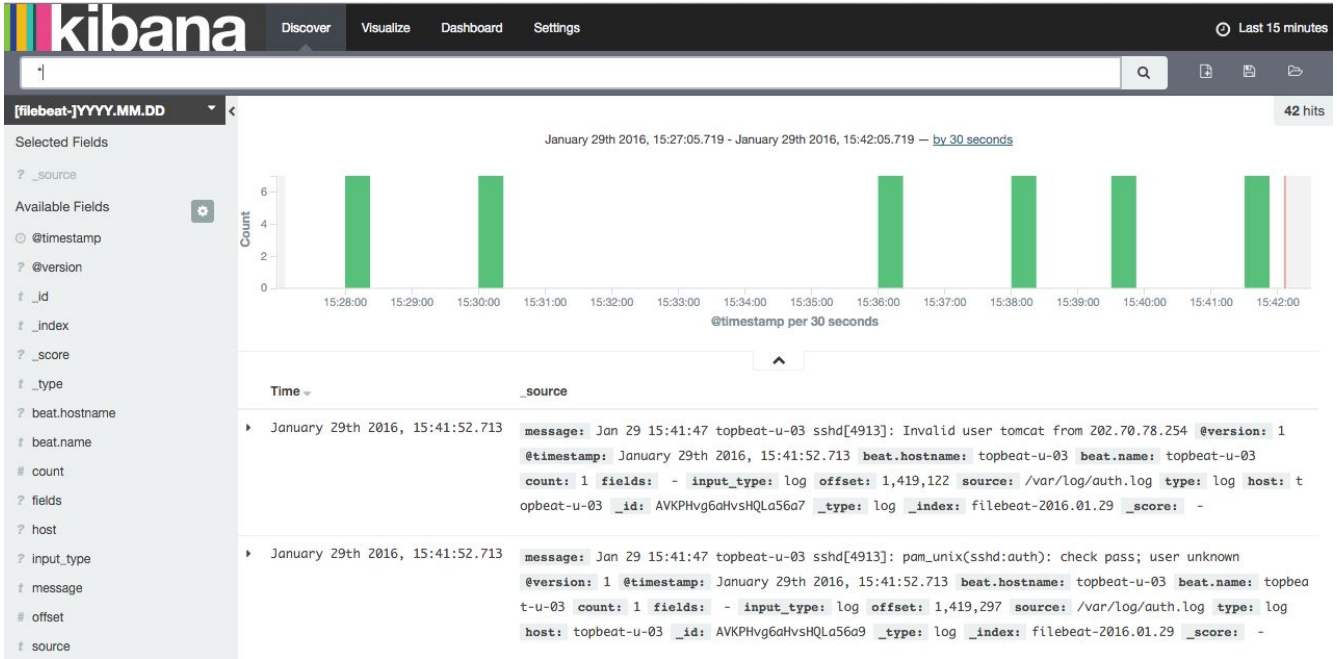
The same for TEXT == Logging



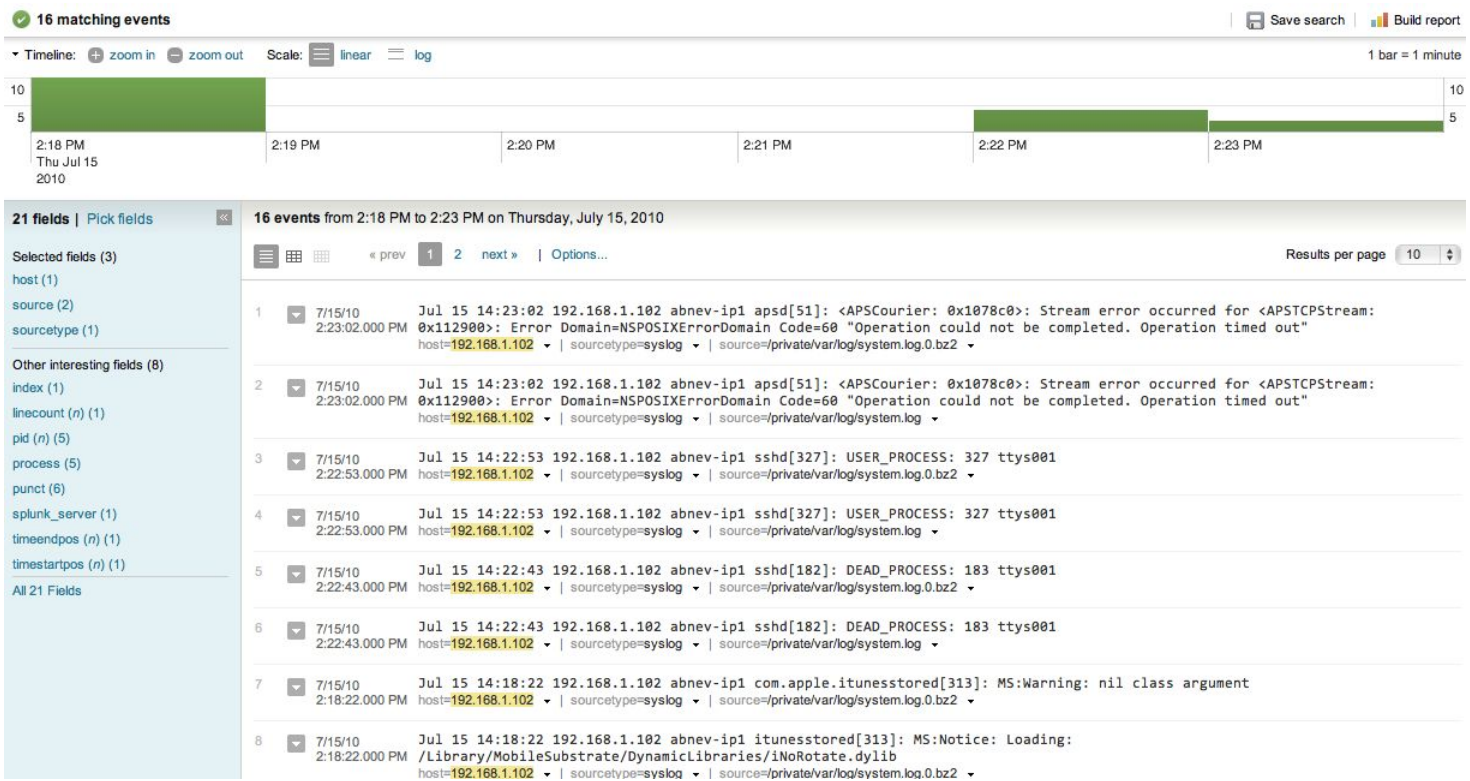
Graylog 2


Graylog2
/messages
GRAYLOG2
messages
streams
analytics
hosts
blacklists
settings
users
Logged in as admin (Log out)
844 / 0
Quickfilter
Messages
Currently containing 26.138 messages. Oldest message is from 30.10.2012 - 08:29:28. Stored 26.057 messages in the last 10 minutes.
Date
Host
Sev.
Facility
Message
Welcome, admin!
Your current time: 30.10.2012 - 07:37:06
6000
4000
2000
0
8:24
8:26
8:28
8:30
8:32
8:34
8:36
Favorite streams
No favorites.
Jobs & Tasks
Stream subscriptions: not running
Stream alerts: not running
Open dashboard
Server health

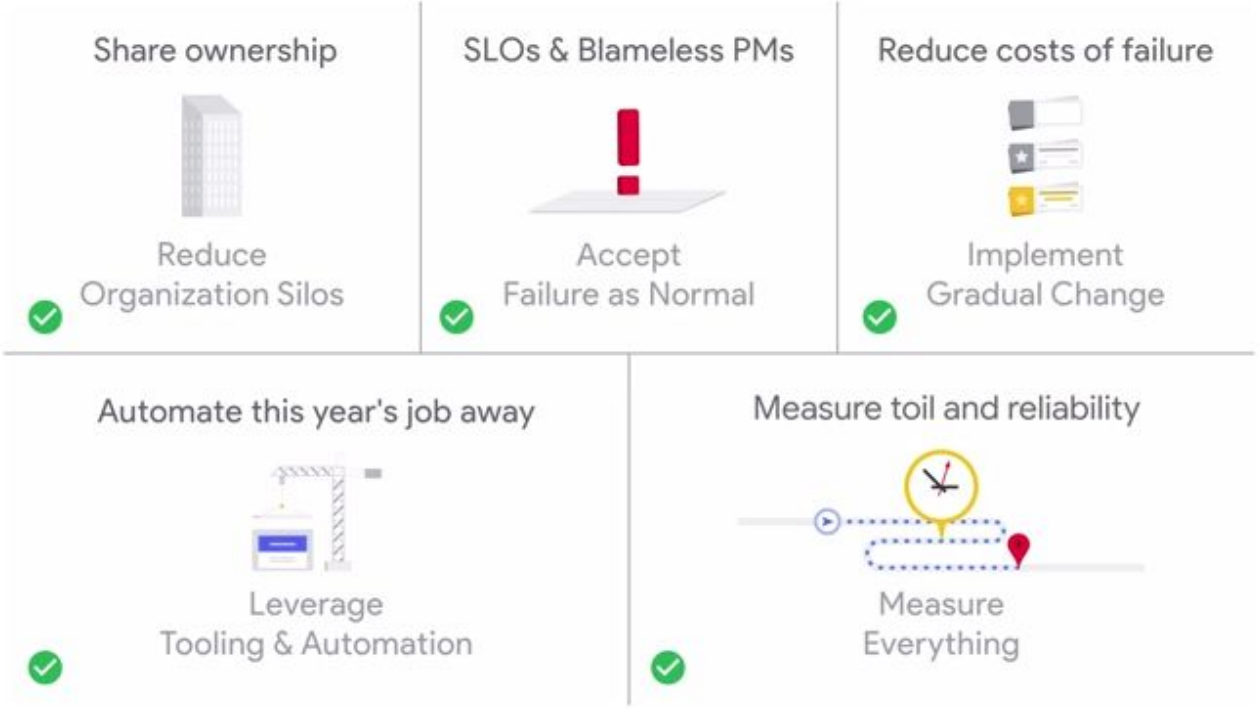
ELK

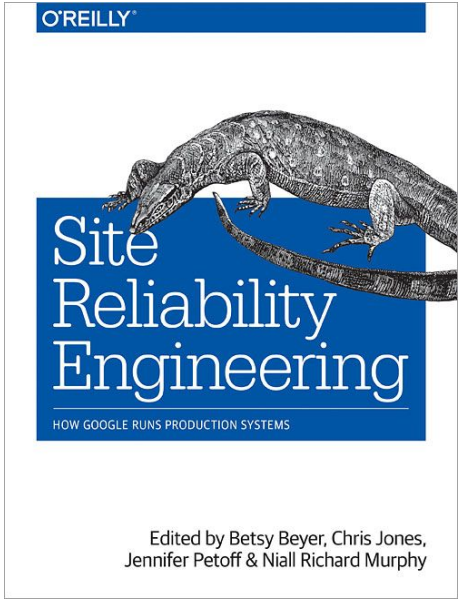

kibana
Discover
Visualize
Dashboard
Settings
Last 15 minutes
*
[filebeat-]YYYY.MM.DD
Selected Fields
? _source
Available Fields
@timestamp
? @version
t _id
t _index
? _score
t _type
? beat.hostname
t beat.name
count
? fields
? host
? input_type
t message
offset
t source
42 hits
January 29th 2016, 15:27:05.719 - January 29th 2016, 15:42:05.719 — by 30 seconds
Count
6
4
2
0
15:28:00
15:29:00
15:30:00
15:31:00
15:32:00
15:33:00
15:34:00
15:35:00
15:36:00
15:37:00
15:38:00
15:39:00
15:40:00
15:41:00
15:42:00
@timestamp per 30 seconds
Time
_source
January 29th 2016, 15:41:52.713
message: Jan 29 15:41:47 topbeat-u-03 sshd[4913]: Invalid user tomcat from 202.70.78.254 @version: 1 @timestamp: January 29th 2016, 15:41:52.713 beat.hostname: topbeat-u-03 beat.name: topbeat-u-03 count: 1 fields: - input_type: log offset: 1,419,122 source: /var/log/auth.log type: log host: topbeat-u-03 _id: AVKPHvg6aHvsHQLa56a7 _type: log _index: filebeat-2016.01.29 _score: -
January 29th 2016, 15:41:52.713
message: Jan 29 15:41:47 topbeat-u-03 sshd[4913]: pam_unix(sshd:auth): check pass; user unknown @version: 1 @timestamp: January 29th 2016, 15:41:52.713 beat.hostname: topbeat-u-03 beat.name: topbeat-u-03 count: 1 fields: - input_type: log offset: 1,419,297 source: /var/log/auth.log type: log host: topbeat-u-03 _id: AVKPHvg6aHvsHQLa56a9 _type: log _index: filebeat-2016.01.29 _score: -

Splunk


16 matching events
Save search
Build report
Timeline: zoom in zoom out Scale: linear log
1 bar = 1 minute
10
5
2:18 PM Thu Jul 15 2010
2:19 PM
2:20 PM
2:21 PM
2:22 PM
2:23 PM
21 fields | Pick fields
Selected fields (3)
host (1)
source (2)
sourcetype (1)
Other interesting fields (8)
index (1)
linecount (n) (1)
pid (n) (5)
process (5)
punct (6)
splunk_server (1)
timeendpos (n) (1)
timestartpos (n) (1)
All 21 Fields
16 events from 2:18 PM to 2:23 PM on Thursday, July 15, 2010
« prev 1 2 next » | Options...
Results per page 10
1 7/15/10 2:23:02.000 PM Jul 15 14:23:02 192.168.1.102 abnev-ip1 apsd[51]: <APSCourier: 0x1078c0>: Stream error occurred for <APSTCPStream: 0x112900>: Error Domain=NSPOSIXErrorDomain Code=60 "Operation could not be completed. Operation timed out"
host=192.168.1.102 | sourcetype=syslog | source=/private/var/log/system.log.0.bz2
2 7/15/10 2:23:02.000 PM Jul 15 14:23:02 192.168.1.102 abnev-ip1 apsd[51]: <APSCourier: 0x1078c0>: Stream error occurred for <APSTCPStream: 0x112900>: Error Domain=NSPOSIXErrorDomain Code=60 "Operation could not be completed. Operation timed out"
host=192.168.1.102 | sourcetype=syslog | source=/private/var/log/system.log
3 7/15/10 2:22:53.000 PM Jul 15 14:22:53 192.168.1.102 abnev-ip1 sshd[327]: USER_PROCESS: 327 ttys001
host=192.168.1.102 | sourcetype=syslog | source=/private/var/log/system.log.0.bz2
4 7/15/10 2:22:53.000 PM Jul 15 14:22:53 192.168.1.102 abnev-ip1 sshd[327]: USER_PROCESS: 327 ttys001
host=192.168.1.102 | sourcetype=syslog | source=/private/var/log/system.log
5 7/15/10 2:22:43.000 PM Jul 15 14:22:43 192.168.1.102 abnev-ip1 sshd[182]: DEAD_PROCESS: 183 ttys001
host=192.168.1.102 | sourcetype=syslog | source=/private/var/log/system.log.0.bz2
6 7/15/10 2:22:43.000 PM Jul 15 14:22:43 192.168.1.102 abnev-ip1 sshd[182]: DEAD_PROCESS: 183 ttys001
host=192.168.1.102 | sourcetype=syslog | source=/private/var/log/system.log
7 7/15/10 2:18:22.000 PM Jul 15 14:18:22 192.168.1.102 abnev-ip1 com.apple.itunesstored[313]: MS:Warning: nil class argument
host=192.168.1.102 | sourcetype=syslog | source=/private/var/log/system.log.0.bz2
8 7/15/10 2:18:22.000 PM Jul 15 14:18:22 192.168.1.102 abnev-ip1 itunesstored[313]: MS:Notice: Loading: /Library/MobileSubstrate/DynamicLibraries/iNoRotate.dylib
host=192.168.1.102 | sourcetype=syslog | source=/private/var/log/system.log.0.bz2

SRE


Share ownership
Reduce Organization Silos
SLOs & Blameless PMs
Accept Failure as Normal
Reduce costs of failure
Implement Gradual Change
Automate this year's job away
Leverage Tooling & Automation
Measure toil and reliability
Measure Everything


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Richard Murphy

Google - How to Develop SRE?

What Makes SRE, SRE?

Simple:

- Hire only coders
- Have an SLA for your service
- Measure and report performance against SLA
- Use Error Budgets and gate launches on them
- Common staffing pool for SRE and DEV
- Excess Ops work overflows to DEV team
- Cap SRE operational load at 50%
- Share 5% of ops work with DEV team
- Oncall teams at least 8 people, or 6x2
- Maximum of 2 events per oncall shift.
- Post mortem for every event
- Post mortems are blameless and focus on process and technology, not people

SRE

Characteristics of Toil

- Manual
- Repetitive
- Automatable
- Tactical
- Devoid of long-term value

Overhead

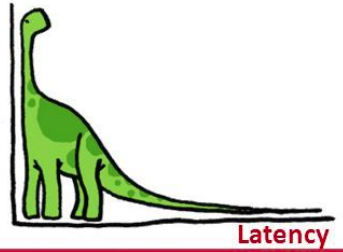
Not toil

- Email
- Expense reports
- Meetings
- Traveling

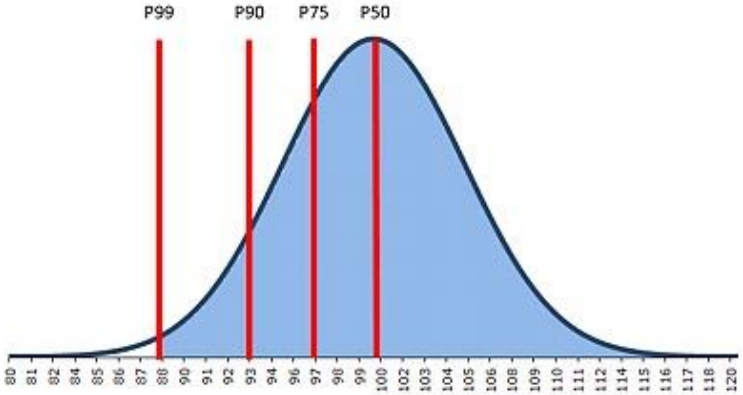
Tail Latency

Low Latency for All Users

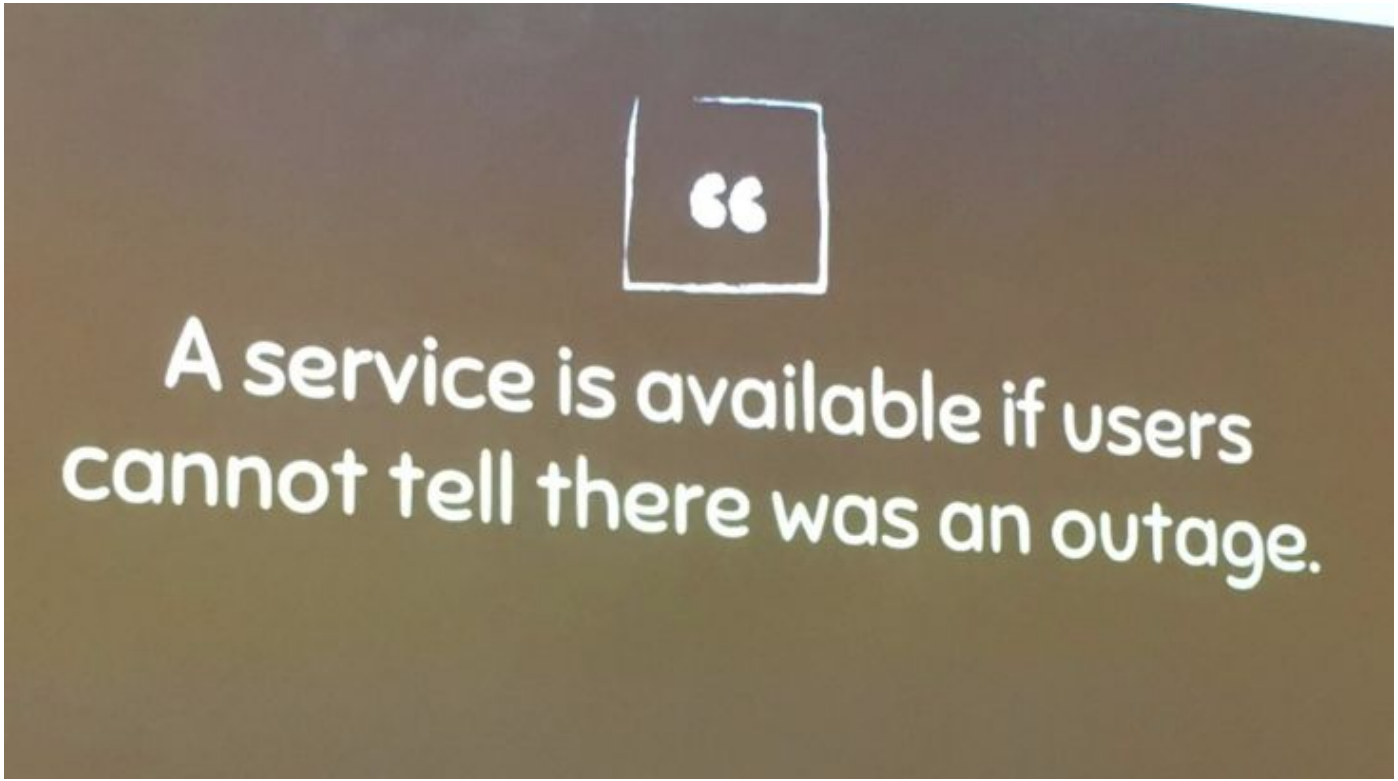
- **Reduce tail latency** (high-percentile response time)
- Reducing average latency is not sufficient



Commercial search engine reduces 99th-percentile latency



Reliability and Availability!



Telemetry & Observability

Diego Pacheco