Building a logging pipeline with Open Source tools

Iñigo Ortiz de Urbina Cazenave
• Iñigo Ortiz de Urbina Cazenave
• Systems Engineer
whoami; groups;

- Iñigo Ortiz de Urbina Cazenave
- Systems Engineer @ RIPE NCC
- RIPE NCC
  - RIR for *Europe, the Middle East, parts of Central Asia*
  - IP and ASN allocation, registration
  - RIPE DB
  - DNS
  - Routing Information Service
  - RIPE Stat
  - RIPE Atlas
What is RIPE Atlas?

v1 & v2: Lantronix XPort Pro

v3: TP-Link TL-MR3020

RIPE Atlas anchor: Soekris net6501-70
What is RIPE Atlas?

Largest active measurement network

https://atlas.ripe.net
RIPE Atlas in numbers

- ~8200 probes online
- ~21000 users
- ~7800 ongoing measurements
  - ~250 built-ins
  - ~7550 user defined measurements
- ~2300 results per second
- ~40 servers
Infrastructure Overview
Log sources

- RIPE Atlas httpd access logs
- RIPE Atlas Software (warning, error, critical)
- Hadoop, HBase, zookeeper, Thrift
- Other:
  - Syslog
  - Custom scripts
The problem

- Production
- Collection
- Transport
- Queueing, buffering
- Massaging
- Storage
- Search and analysis
The problem
The target

- Timestamps: ns since epoch, RFC3339
- Structured information
- Common representation and semantics across the board
- Robust, scalable, stateless pipeline for events and metrics
- *One stop shop* for logs and metrics
The prototype

- Servers publish events to message brokers
- Workers consume events from queues
  - Perform arbitrary data transformation on raw data
  - Store events
- Logging backend supports:
  - Log search
  - Log analysis and visualisation
  - Monitoring dashboards
The toolset

- **Heka**
  - Collect, transport, enhance, output events

- **RabbitMQ**
  - Decouple producers from consumers

- **Elasticsearch**
  - Distributed event store and search

- **Kibana**
  - UI for search, analysis, dashboards

- **Ansible and Git**
  - Version control and configuration management
Heka

- Written in go
- Small footprint
- Performant
- Uses protobuf
- Sandboxed execution of custom LUA scripts
- TOML config files
- Sensible internal pipeline
RabbitMQ

- Written in erlang
- Dedicated vhost/exchange/queue per user
- Acknowledged messages, persistent when required
- Standalone instances behind LB pool
- Nice flow control features
- ~250 concurrent connections
- ~550 channels
- 8 exchanges
- 12 queues
Elasticsearch

- Distributed free text search engine
- Apache Lucene
- Scalable, fast
- Aggregations (*facets*)
- Battle-tested at GitHub, ebay, The Guardian, bol.com…
- Powerful Query DSL
- Backup and restore (NFS, HDFS)
- Sensible defaults
Cluster description

- ~1B docs, 90+ indices, 550+ shards, ~1K events/sec
- 3x dedicated servers (Dell PowerEdge C5000 chassis)
  - 4x 1TB disks (7.2K RPM), 32GB of RAM
- No dedicated master node
- Dedicated standalone cluster for monitoring and index management
- PXE booted, RAM based root FS
Cluster monitoring

![Cluster Monitoring Dashboard](image)

### Cluster Summary
- Name: [Name]
- Status: [Status]
- Nodes: [Nodes]
- Indices: [Indices]
- Shards: [Shards]
- Data: [Data]
- CPU: [CPU]
- Memory: [Memory]
- Up time: [Up time]
- Version: [Version]

### Document Count

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Search Request Rate

<table>
<thead>
<tr>
<th>Time</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Indexing Request Rate

<table>
<thead>
<tr>
<th>Time</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Nodes

**3 of 3 nodes** / 0 selected / Last 10m

<table>
<thead>
<tr>
<th>Node</th>
<th>OS CPU (%)</th>
<th>Load (m)</th>
<th>JVM Mem (%)</th>
<th>Disk Free Space</th>
<th>IOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>data01</td>
<td>4.0 min: 0.0 max: 7.0</td>
<td>0.0 min: 0.0 max: 0.1</td>
<td>7.0 min: 3.0 max: 19.0</td>
<td>90.6 GB min: 90.6 GB max: 90.6 GB</td>
<td>18.3 min: 3.5 max: 31.7</td>
</tr>
<tr>
<td>data02</td>
<td>4.0 min: 0.0 max: 15.0</td>
<td>0.0 min: 0.0 max: 0.1</td>
<td>5.7 min: 2.0 max: 18.0</td>
<td>90.6 GB min: 90.6 GB max: 90.6 GB</td>
<td>18.0 min: 3.3 max: 30.0</td>
</tr>
<tr>
<td>search01</td>
<td>1.0 min: 0.0 max: 1.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
<td>3.5 min: 2.0 max: 8.0</td>
<td>90.6 GB min: 90.6 GB max: 90.6 GB</td>
<td>2.2 min: 0.3 max: 2.2</td>
</tr>
</tbody>
</table>

### Indices

**3 of 3 indices** / 0 selected / Last 10m

<table>
<thead>
<tr>
<th>Index</th>
<th>Documents</th>
<th>Index Rate</th>
<th>Search Rate</th>
<th>Merge Rate</th>
<th>Field Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>marvel-2014.11.14</td>
<td>1.8 K min: 1.8 K max: 1.8 K</td>
<td>0.8 min: 0.8 max: 0.8</td>
<td>0.5 min: 0.5 max: 0.5</td>
<td>288.2 KB min: 288.2 KB max: 288.2 KB</td>
<td>228.6 KB min: 228.6 KB max: 228.6 KB</td>
</tr>
<tr>
<td>test_index-2014.11.14</td>
<td>0.0 min: 0.0 max: 0.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
</tr>
<tr>
<td>marvel-kibana</td>
<td>1.0 min: 1.0 max: 1.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
<td>0.0 min: 0.0 max: 0.0</td>
</tr>
</tbody>
</table>
Index management

Flow delete-indices

<table>
<thead>
<tr>
<th>Execution Id</th>
<th>User</th>
<th>Start Time</th>
<th>End Time</th>
<th>Elapsed</th>
<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1155</td>
<td>azkaban</td>
<td>2014-11-16 02:00 0ts</td>
<td>2014-11-16 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1151</td>
<td>azkaban</td>
<td>2014-11-15 02:00 0ts</td>
<td>2014-11-15 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1147</td>
<td>azkaban</td>
<td>2014-11-14 02:00 0ts</td>
<td>2014-11-14 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1143</td>
<td>azkaban</td>
<td>2014-11-13 02:00 0ts</td>
<td>2014-11-13 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1139</td>
<td>azkaban</td>
<td>2014-11-12 02:00 0ts</td>
<td>2014-11-12 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1135</td>
<td>azkaban</td>
<td>2014-11-11 02:00 0ts</td>
<td>2014-11-11 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1131</td>
<td>azkaban</td>
<td>2014-11-10 02:00 0ts</td>
<td>2014-11-10 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1126</td>
<td>azkaban</td>
<td>2014-11-09 02:00 0ts</td>
<td>2014-11-09 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1121</td>
<td>azkaban</td>
<td>2014-11-08 02:00 0ts</td>
<td>2014-11-08 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>1117</td>
<td>azkaban</td>
<td>2014-11-07 02:00 0ts</td>
<td>2014-11-07 02:00 0ts</td>
<td>0 sec</td>
<td>Success</td>
<td></td>
</tr>
</tbody>
</table>

Flow Execution 686 SUCCEEDED

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Timeline</th>
<th>Start Time</th>
<th>End Time</th>
<th>Elapsed</th>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>create-hdp-canis</td>
<td>command</td>
<td></td>
<td>2014-11-03 01:00 02s</td>
<td>2014-11-03 01:00 07s</td>
<td>5 sec</td>
<td>Success</td>
<td>Details</td>
</tr>
<tr>
<td>update-hdp-canis</td>
<td>command</td>
<td></td>
<td>2014-11-03 01:00 07s</td>
<td>2014-11-03 01:00 13s</td>
<td>5 sec</td>
<td>Success</td>
<td>Details</td>
</tr>
</tbody>
</table>
Kibana

- Webapp for analytics and visualisation
- Intuitive for most
- Easy sharing capabilities
- Pretty graphs!
  - Which *may* melt your cluster :-)

Iñigo Ortiz de Urbina Cazenave - NLUUG - 28/05/2015
Kibana

Test - RIPEStat API logs

BY DATA CALL

BY IP

4XX-5XX

2XX-3XX

HTTP STATUS CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Count</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>1124258</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>6222243</td>
<td></td>
</tr>
<tr>
<td>301</td>
<td>9894</td>
<td></td>
</tr>
<tr>
<td>302</td>
<td>7648</td>
<td></td>
</tr>
<tr>
<td>304</td>
<td>3677</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>709</td>
<td></td>
</tr>
<tr>
<td>401</td>
<td>1127</td>
<td></td>
</tr>
<tr>
<td>501</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>414</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Missing field: 0

Other values: 0

Iñigo Ortiz de Urbina Cazenave - NLUUG - 28/05/2015
Recap

- Production
- Collection
- Transport
- Queueing, buffering
- Massaging
- Storage
- Search and analysis
Future plans

• Kibana4
  - Operational simplicity
  - Superior capabilities and UX
  - Migrate all dashboards
  - Encourage data exploration

• Kafka
  - “Stateless” broker
  - Superior performance (backlog ingestion)
  - Unified pipeline

• Data quality checks
Thanks!

- Mail, XMPP: iortiz@ripe.net
- Twitter: @ioc32