DATA WORKS SUMMIT

#DSW2018

10 Steps to Make Your Cluster Hum

Alejandro Fernandez

San Jose
June 17-21, 2018

Speaker

Alejandro Fernandez

airbnb

Infrastructure Engineer

’17 San Jose

’16 Melbourne

’16 San Jose

PMC

Apache Ambari

Contributed

HIVE

Superset

Airflow

Poll

Managing your Hadoop cluster?

Easy

Medium

Difficult

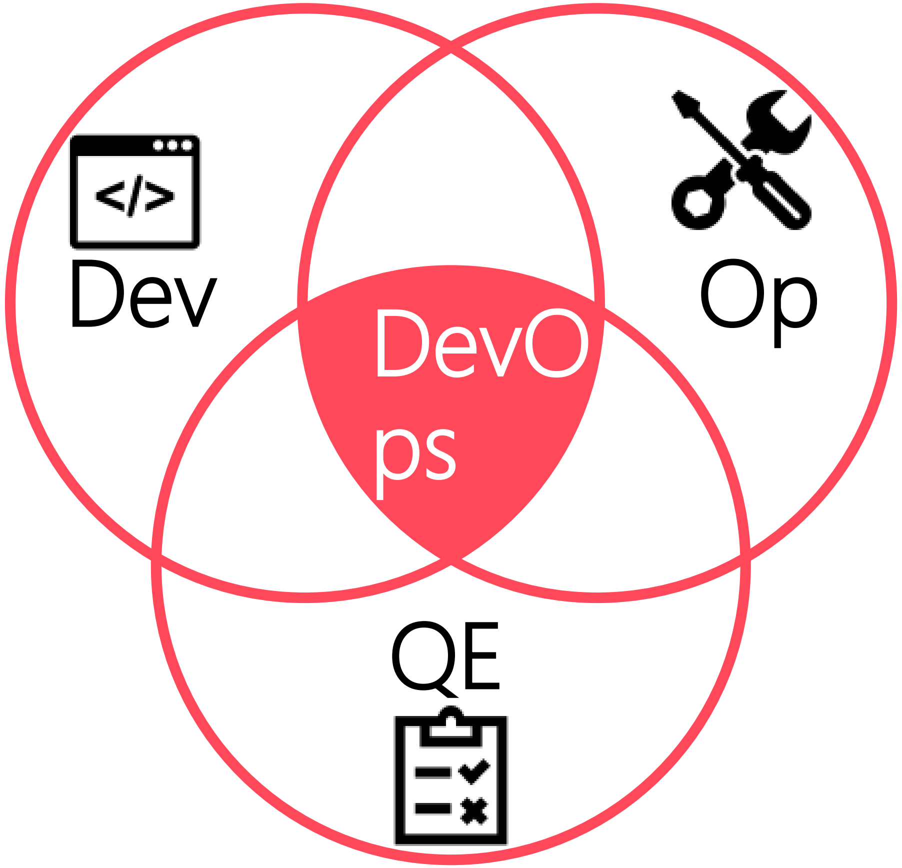
Who is this for?

DevOps & Infra Teams
Technical audience

Dev: Add features

QE: Test for security & scale

Op: Deployment + business continuity



Steps To a Smoother Cluster

1. Automate adding hosts
2. Host Name discovery
3. Artifact management
4. Deployments and upgrades
5. Log searching with ELK

1 2 >

Steps To a Smoother Cluster

6. Monitor Hadoop metrics with Ambari Metrics System
7. Data-driven (metrics + alerts)
8. Cron-jobs, ETL with Apache Airflow (incubating)
9. Visualize trends with Apache Superset (incubating)
10. Automated maintenance

1 2 Q A

1. Deploy New Hosts

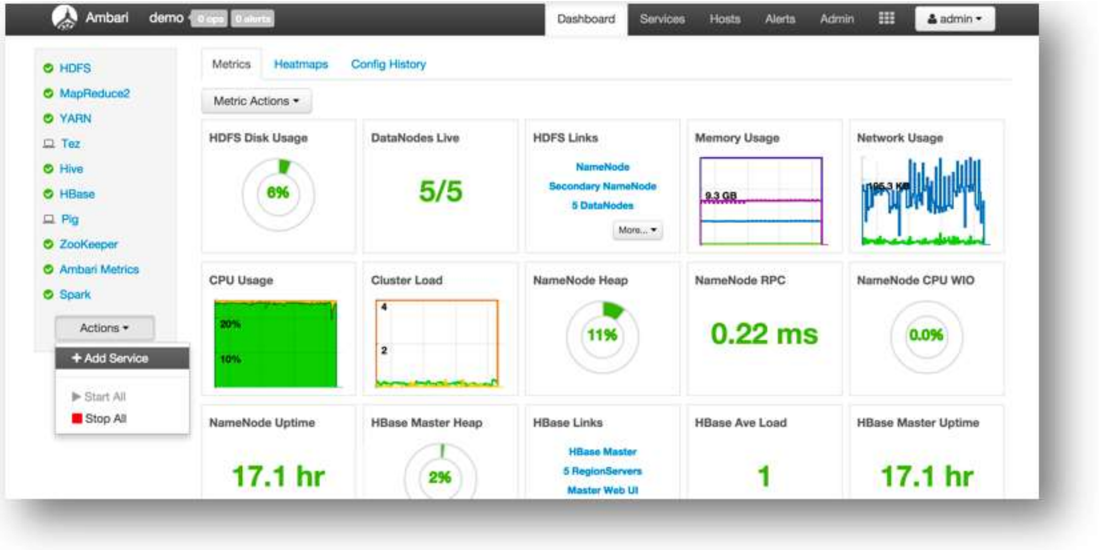
- Using Chef
  - Cookbooks (collection of templates, attributes, and recipes)
  - Recipes (patterns + blocks)
  - Roles (NM, DN, NM, etc.)
- Repeatable
- Idempotent + fixable

Automatically Add Hosts to Cluster

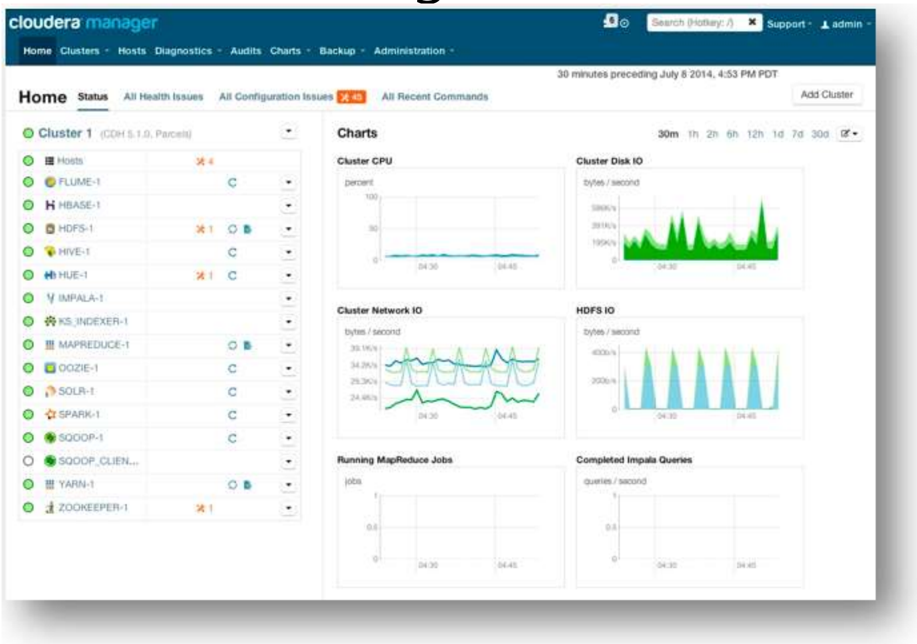
- Add hosts to cluster
- Assign roles

- Rack awareness
- Host discovery

Apache Ambari curl http://



Cloudera Manager API



2. Host Name Discovery

Avoid hardcoding hostnames in configs

File	Config
yarn-site.xml	hadoop.registry.zk.quorum
hbase-site.xml	hbase.zookeeper.quorum
hive-site.xml	hive.metastore.uris hive.zookeeper.quorum
storm-site.xml	nimbus.seeds
etc	etc

- Replace bad host
- ~~Change hard-coded configs~~
- ~~Restart services dependent on those configs~~

Name Discovery Solutions

Method	Pros	Cons
! Apache Ambari Stack Advisor	Works out of the box. {{ var }} is defined in params.py	Write your own for custom services
! Chef Attributes	Can handle multiple environments with same recipe	Configs are written to disk, but not using CM or Ambari
✓ Smartstack	Local load-balanced servers behind the same name	

3. Trust The Artifacts

Artifact: build, jars, scripts, resources, build log, and test results

- Reproducible: Generated by build system
- Trackable: Embed version, Git SHA that generated it
- Trustworthy: Hash and Signed

Simple

+ S3

Enterprise

JFrog Artifactory

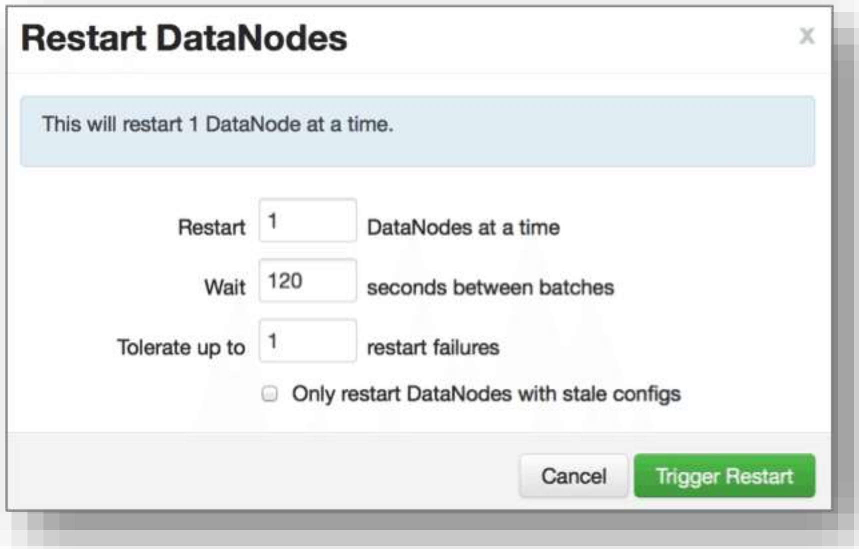
Sonatype
Nexus

Deployments

1. Artifact server

2. Always take backups

3. Deploy parcels or bits
   - Orchestration: rolling restart if possible
   - Verification: analyze alerts on hosts with new changes
   - Downgrade: ready to rollback



Restart DataNodes

This will restart 1 DataNode at a time.

Restart 1 DataNodes at a time

Wait 120 seconds between batches

Tolerate up to 1 restart failures

☐ Only restart DataNodes with stale configs

Automate Calls to Upgrade

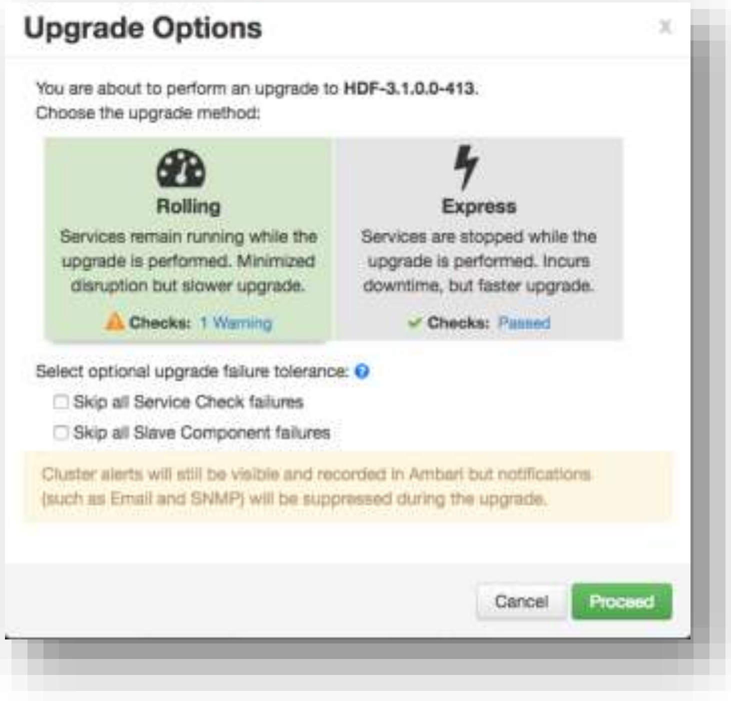
- Prechecks:
  - All services up
  - Ran Service Checks after config changes
  - All hosts heartbeating
  - No hosts in maintenance mode

  …

- Start: Options for failure handling

- Status

Apache Ambari



Upgrade Options

You are about to perform an upgrade to **HDF-3.1.0.0-413**.
Choose the upgrade method:

Rolling
Services remain running while the upgrade is performed. Minimized disruption but slower upgrade.
Checks: 1 Warning

Express
Services are stopped while the upgrade is performed. Incurs downtime, but faster upgrade.
✓ **Checks:** Passed

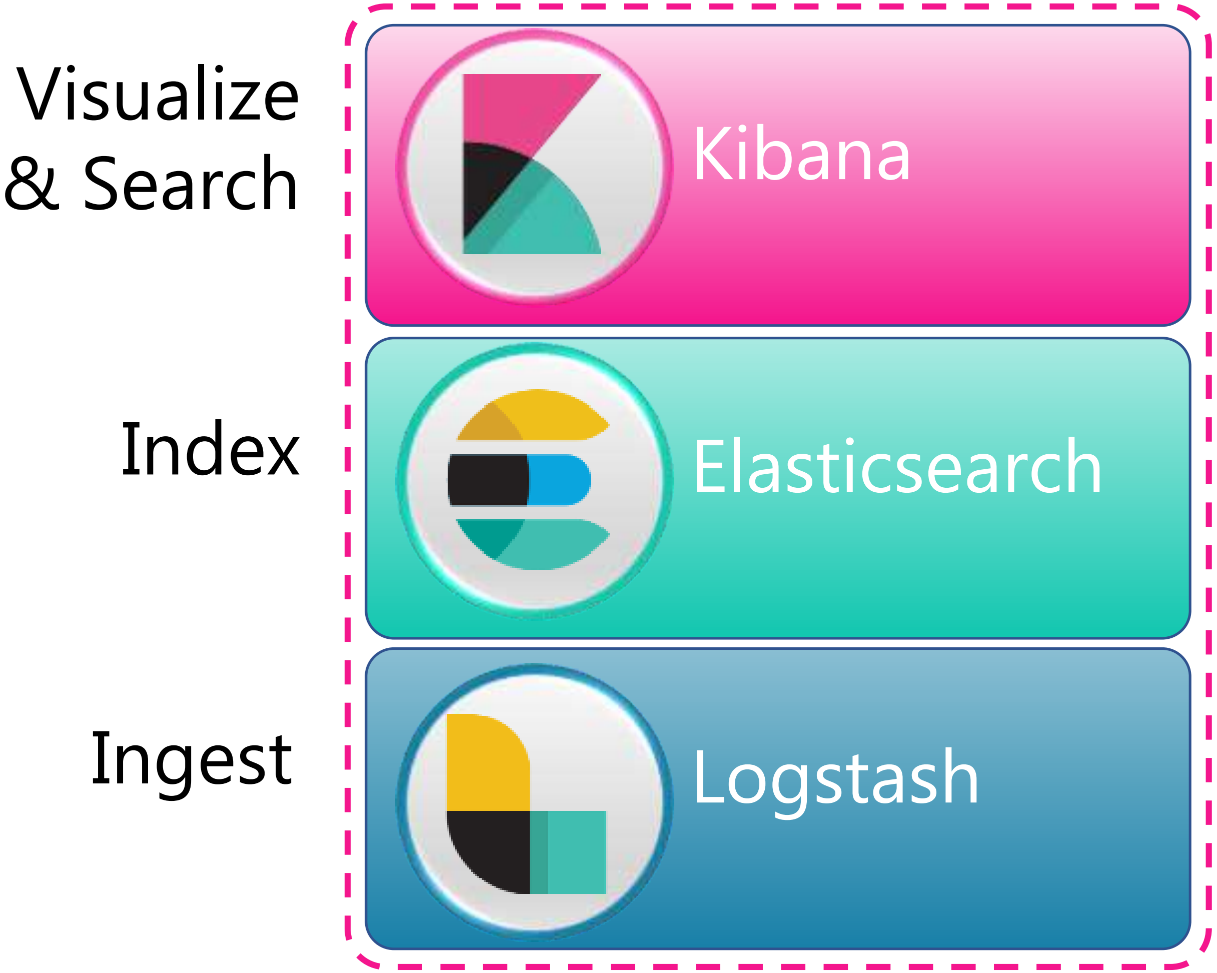
Select optional upgrade failure tolerance:

□ Skip all Service Check failures

□ Skip all Slave Component failures

Cluster alerts will still be visible and recorded in Ambari but notifications (such as Email and SNMP) will be suppressed during the upgrade.

5. ELK Stack for Searching Logs



Visualize & Search

Kibana

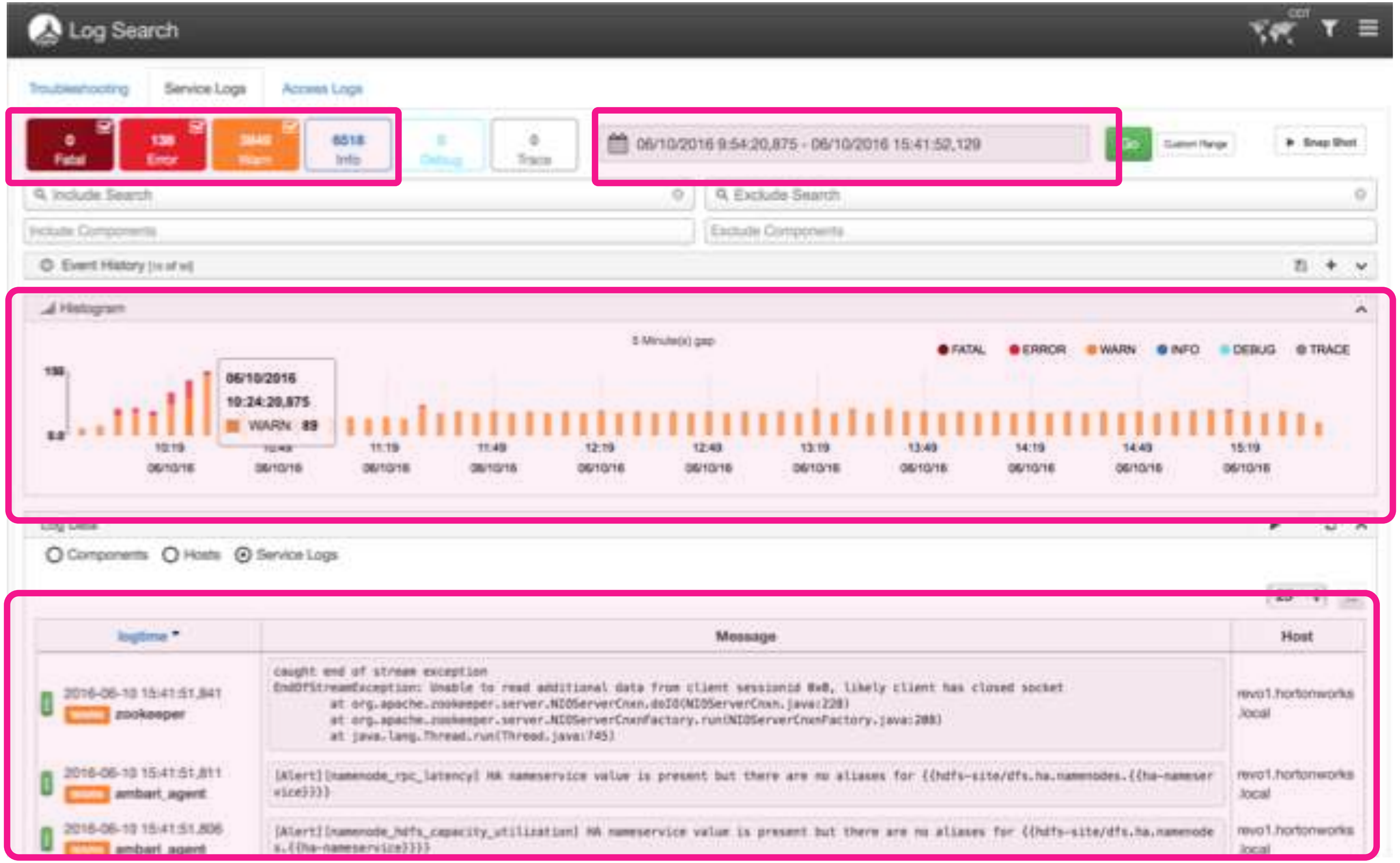
Index

Elasticsearch

Ingest

Logstash

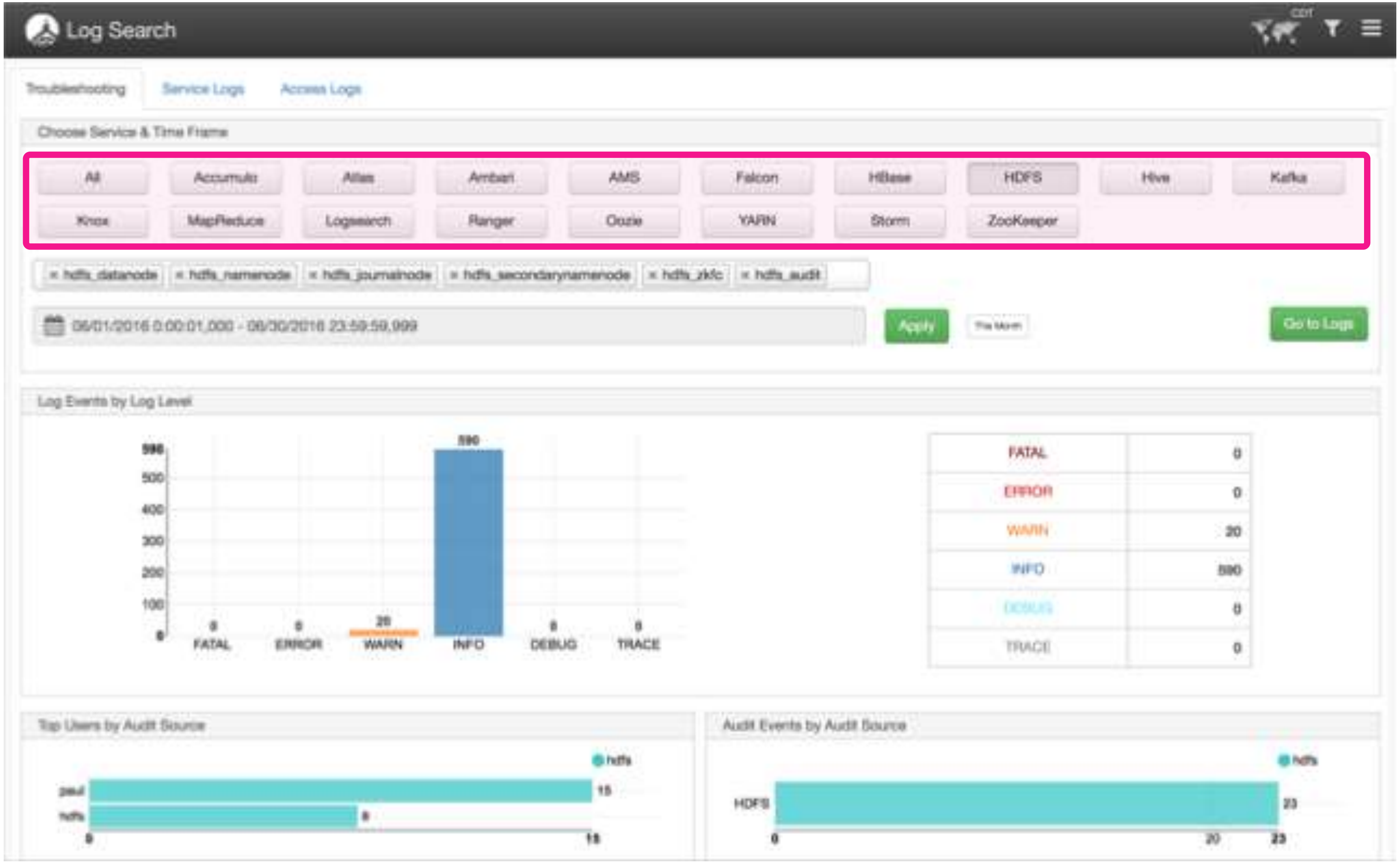
Out-of-the-Box Log Search



06/10/2016 9:54:20,875 - 06/10/2016 15:41:52,129

logtime	Message	Host
2016-06-10 15:41:51,841 WARN zookeeper	caught end of stream exception EndOfStreamException: Unable to read additional data from client sessionid 0x0, likely client has closed socket at org.apache.zookeeper.server.NIOServerCnxn.doIO(NIOServerCnxn.java:228) at org.apache.zookeeper.server.NIOServerCnxnFactory.run(NIOServerCnxnFactory.java:208) at java.lang.Thread.run(Thread.java:745)	revo1.hortonworks.local
2016-06-10 15:41:51,811 WARN ambari_agent	[Alert][namenode_rpc_latency] HA nameservice value is present but there are no aliases for {{hdfs-site/dfs.ha.namenodes.{{ha-nameservice}}}}	revo1.hortonworks.local
2016-06-10 15:41:51,806 WARN ambari_agent	[Alert][namenode_hdfs_capacity_utilization] HA nameservice value is present but there are no aliases for {{hdfs-site/dfs.ha.namenodes.{{ha-nameservice}}}}	revo1.hortonworks.local

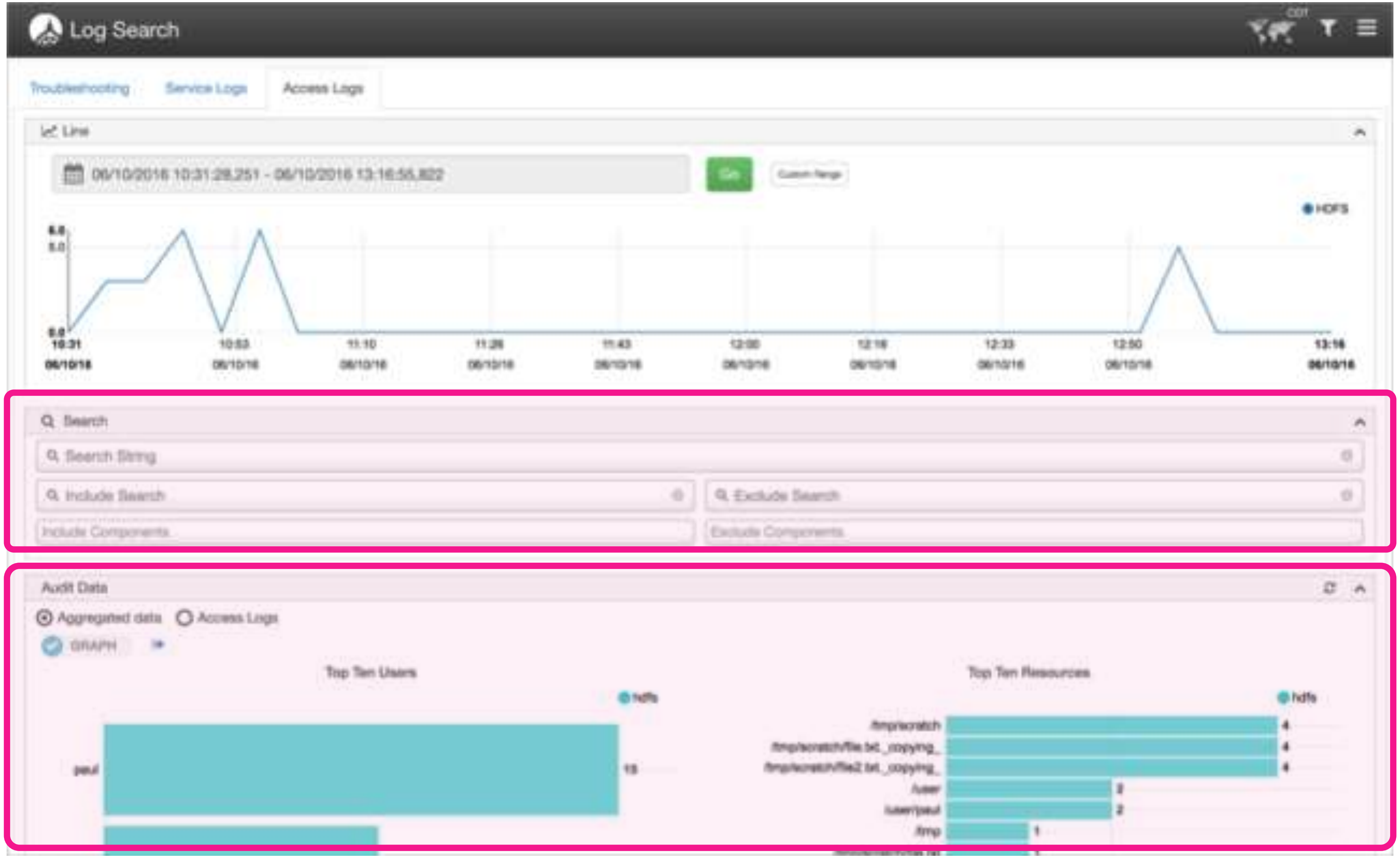
Advanced Log Search



06/01/2016 0:00:01,000 - 06/30/2016 23:59:59,999

FATAL	0
ERROR	0
WARN	20
INFO	590
DEBUG	0
TRACE	0

Advanced Log Search



06/10/2016 10:31:28.251 - 06/10/2016 13:16:55.822

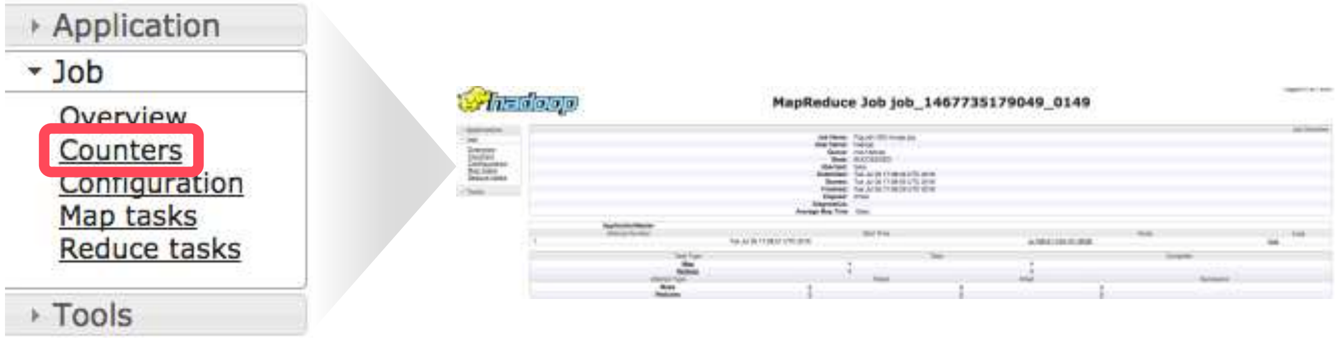
⦿ Aggregated data ○ Access Logs

6. Collecting Metrics

- Apache Zookeeper:
  - Jmxterm
  - Inspect the beans: num connections, latency, timeouts, etc.

- Apache HDFS and YARN:
  - WebHDFS stats on a directory
  - ResourceManager web UI job counters
  - JMX

```
curl http://$HOST:$PORT/webhdfs/v1
```

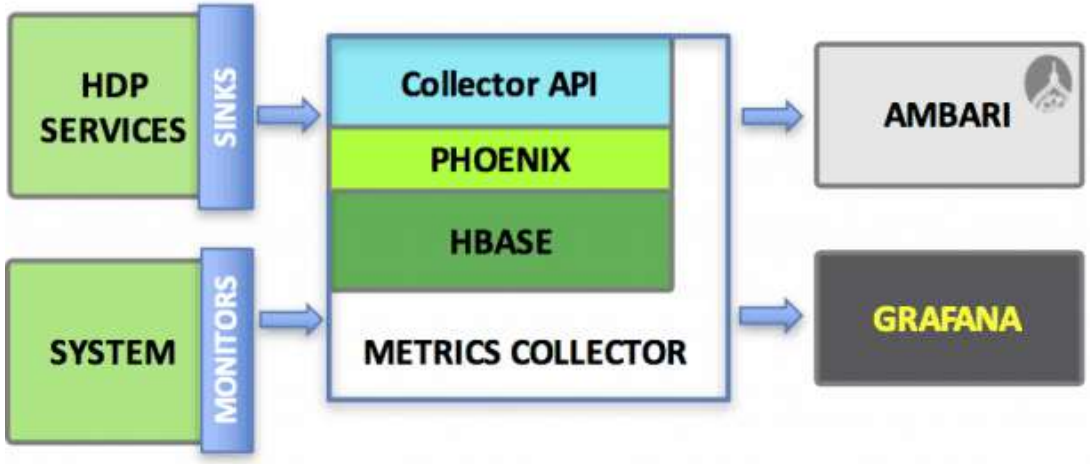


and more…

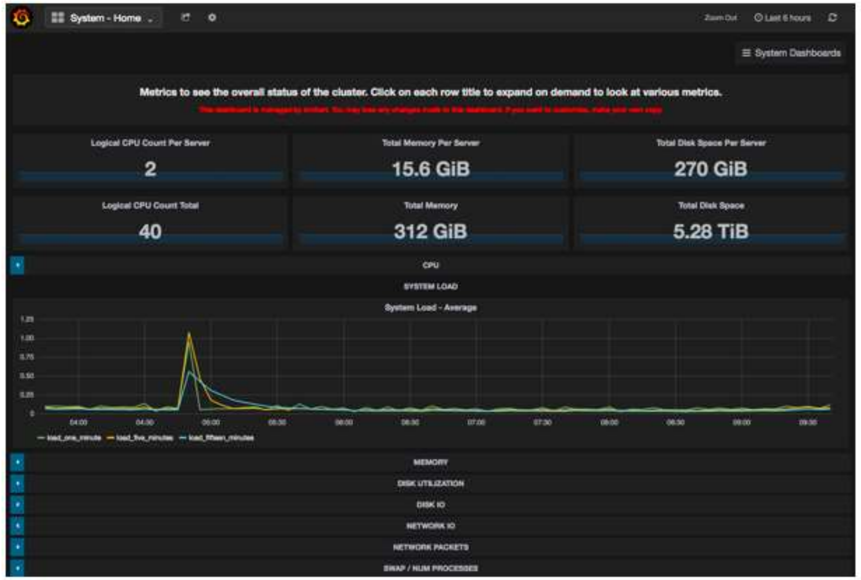
6. Collecting with AMS

Ambari Metrics System for Hadoop metrics:

- Support Hadoop stack using Hadoop syncs interface
- Anomaly Detection
- Query with Phoenix
- High Availability
- Scale horizontally



Grafana UI



Metrics to see the overall status of the cluster. Click on each row title to expand on demand to look at various metrics.

[illegible]

Logical CPU Count Per Server	Total Memory Per Server	Total Disk Space Per Server
2	15.6 GiB	270 GiB

Logical CPU Count Total	Total Memory	Total Disk Space
40	312 GiB	5.28 TiB

CPU

SYSTEM LOAD

System Load - Average

MEMORY

DISK UTILIZATION

DISK IO

NETWORK IO

NETWORK PACKETS

SWAP / NUM PROCESSES

METRICS LOOK GOOD

I THINK IT’S SAFE TO DEPLOY

7. Telemetry

StatsD client on each host

- Emitting metrics: CLI, Python client, Ruby client

GAUGE	**INCREMENT**	**DECREMENT**	**RATE**	**COUNT**	**…**

</> E.g., DataDog Client

```
from checks import AgentCheck

class HelloCheck(AgentCheck):
  def check(self, instance):
    self.gauge('hello.world', 1)
```

DataDog Examples (reference)

</> Python HTTP Check Snippet

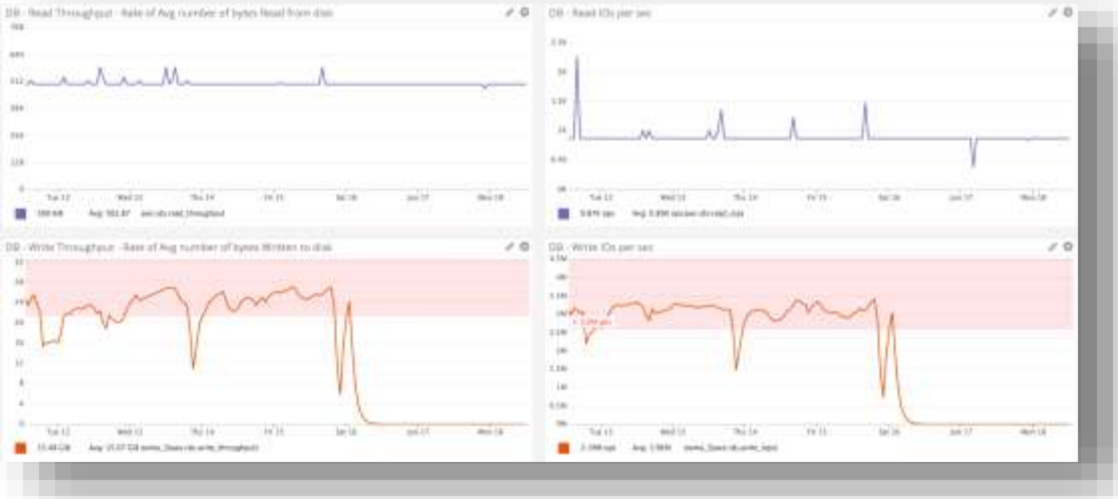
```python
start_time = time.time()
try:
  r = requests.get(url, timeout=timeout)
  end_time = time.time()
except requests.exceptions.Timeout as e:
  self.timeout_event(url, timeout, aggregation_key)
  return

if r.status_code != 200:
  self.status_code_event(url, r, aggregation_key)

timing = end_time - start_time
self.gauge('http.response_time', timing, tags=['http_check'])
```

Telemetry

- Functions: avg, min, max, rolling window, stddev
- Thresholds
- Alerts


DB - Read Throughput - Rate of Avg number of bytes Read from disk
DB - Read IOs per sec
DB - Write Throughput - Rate of Avg number of bytes Written to disk
DB - Write IOs per sec
Tue 12
Wed 13
Thu 14
Fri 15
Sat 16
Sun 17
Mon 18

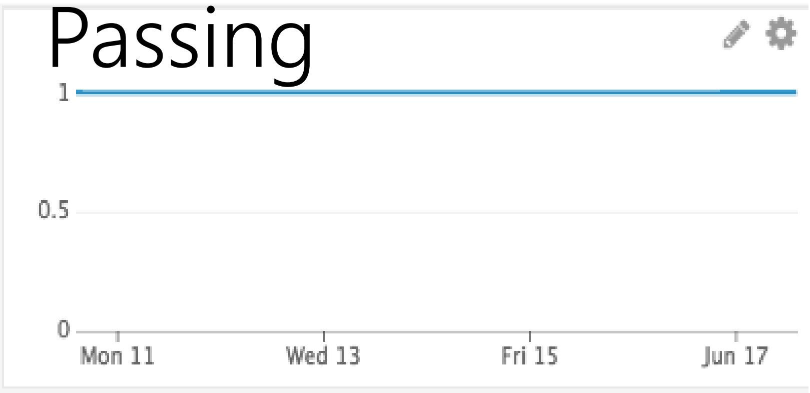
Simple Metric

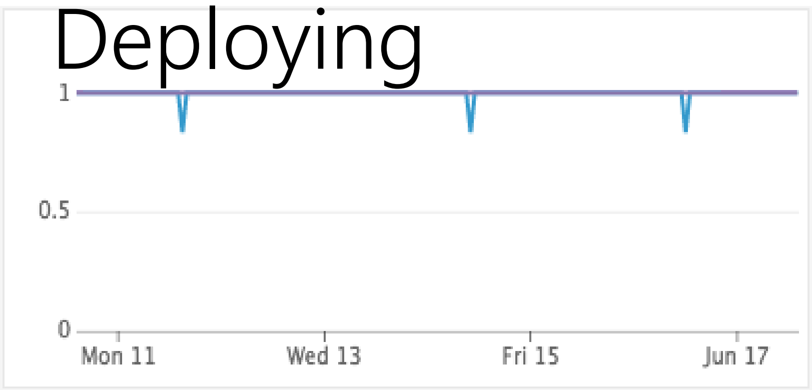
</>

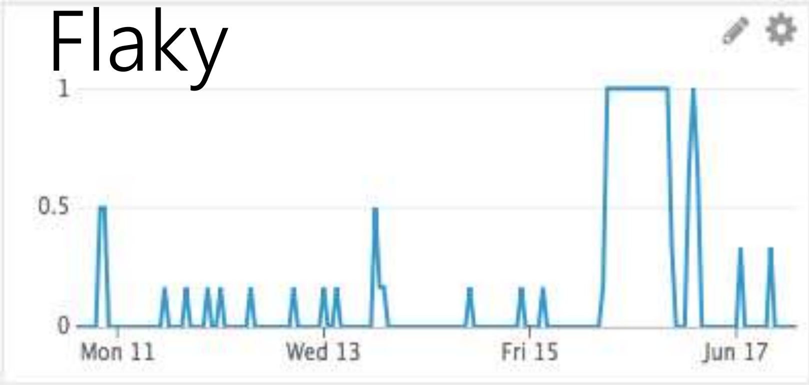
```
metric = 1

try:
  hive -e "SELECT COUNT(*) FROM canary"
except SomeExceptionType:
  metric = 0

dd-report-metric -m "hive" -v $metric
```


Passing


Deploying


Flaky

8. ETL with Apache Airflow (incubating)

Airflow

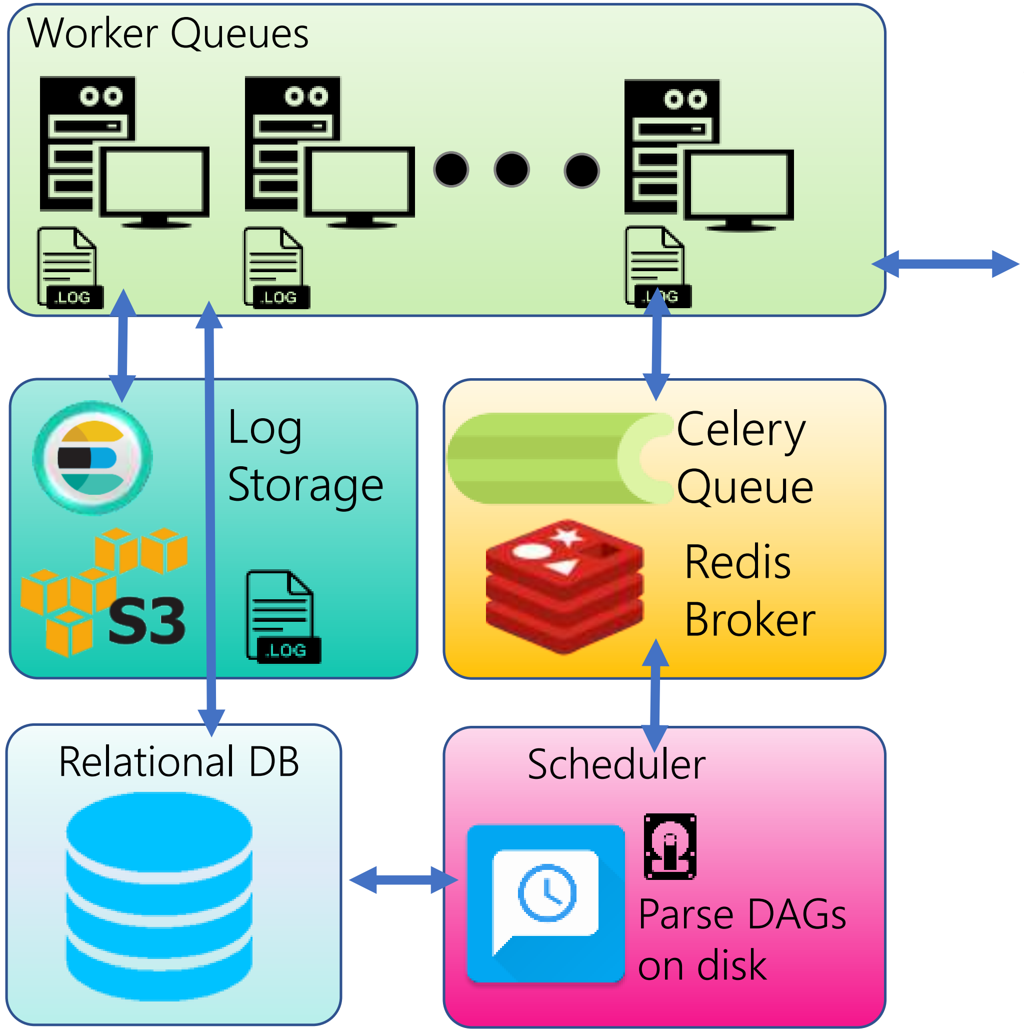
Open-sourced at Airbnb by Max Beauchemin

ETL – Develop your DAGs using Python + Jinja

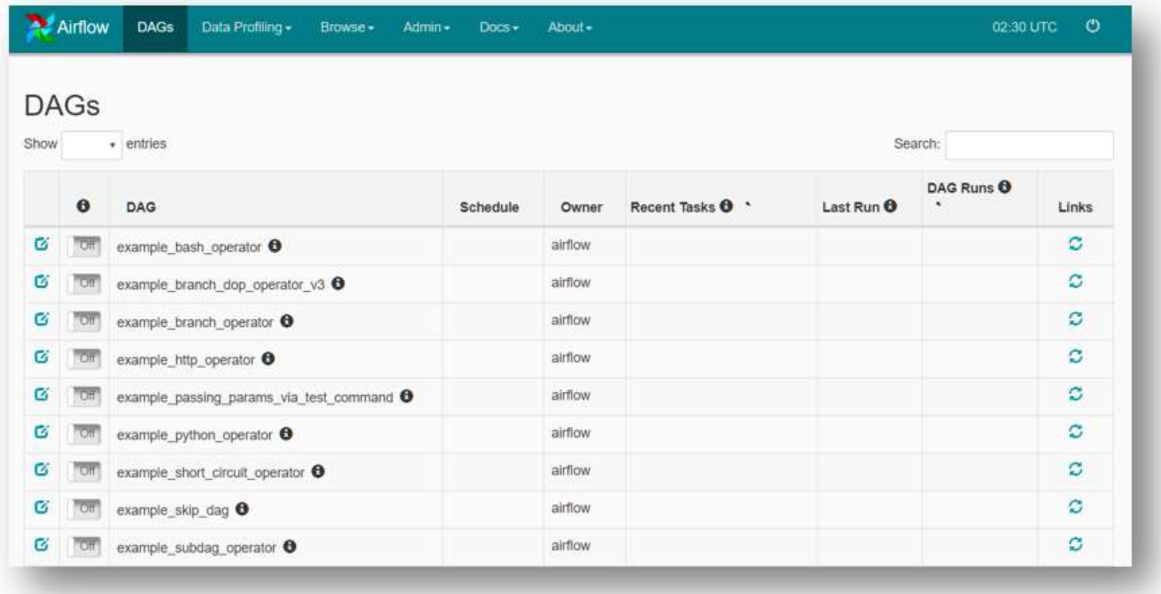
- Sensors: wait for data to land
- Operators galore: Bash, Hive, Presto, MySQL, etc.
- Failure tolerance, retries, notifications
- Backfill
- GUI

8. ETL with Apache Airflow (incubating)

Airflow



Web UI



DAGs

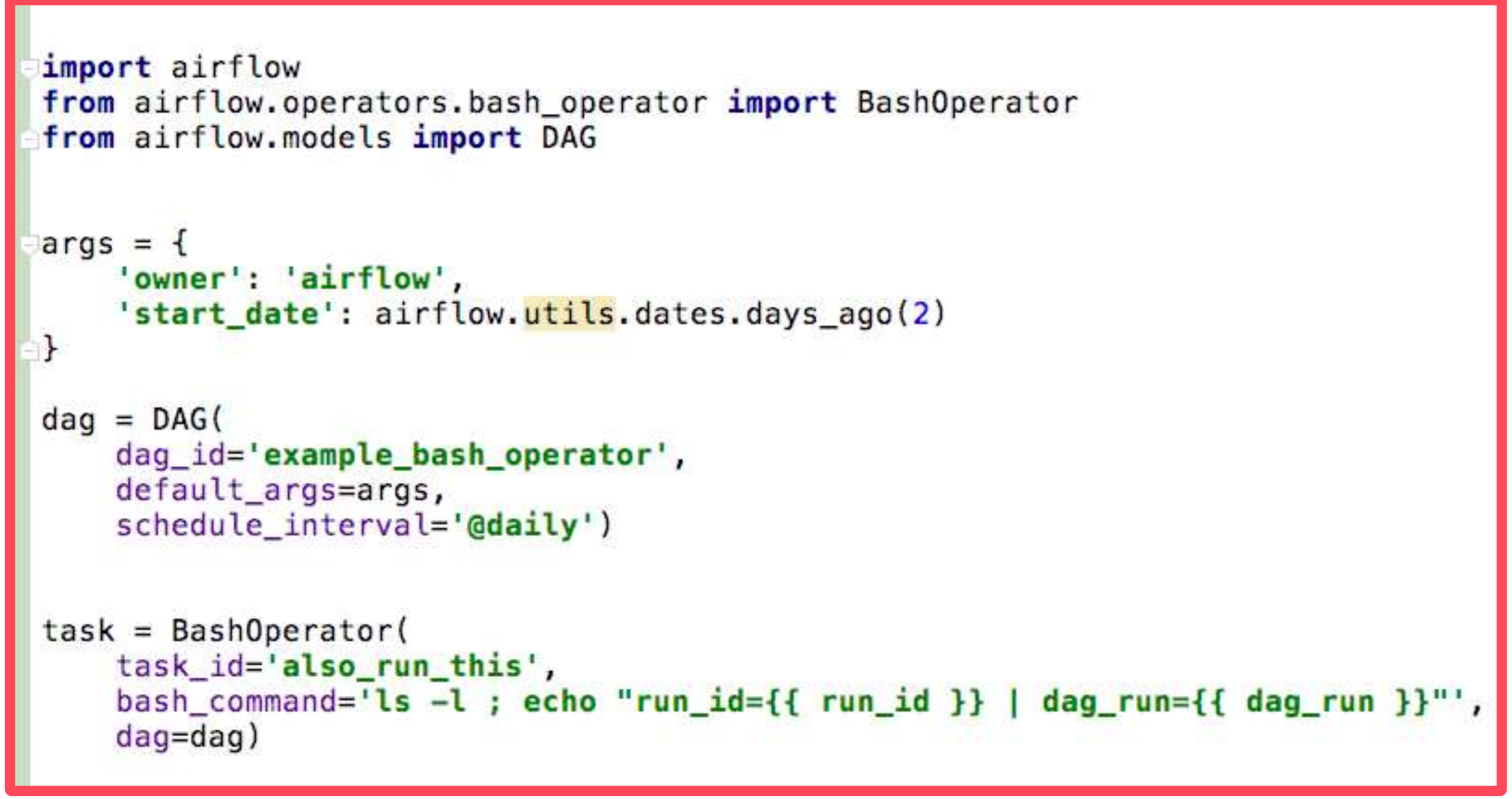
Show [▾] entries

Search:

	ℹ	DAG	Schedule	Owner	Recent Tasks ℹ	Last Run ℹ	DAG Runs ℹ	Links
✎	Off	example_bash_operator ℹ		airflow				⟳
✎	Off	example_branch_dop_operator_v3 ℹ		airflow				⟳
✎	Off	example_branch_operator ℹ		airflow				⟳
✎	Off	example_http_operator ℹ		airflow				⟳
✎	Off	example_passing_params_via_test_command ℹ		airflow				⟳
✎	Off	example_python_operator ℹ		airflow				⟳
✎	Off	example_short_circuit_operator ℹ		airflow				⟳
✎	Off	example_skip_dag ℹ		airflow				⟳
✎	Off	example_subdag_operator ℹ		airflow				⟳

Example Bash Operator (reference)

</> DAG - Bash Operator



```
import airflow
from airflow.operators.bash_operator import BashOperator
from airflow.models import DAG


args = {
    'owner': 'airflow',
    'start_date': airflow.utils.dates.days_ago(2)
}

dag = DAG(
    dag_id='example_bash_operator',
    default_args=args,
    schedule_interval='@daily')


task = BashOperator(
    task_id='also_run_this',
    bash_command='ls -l ; echo "run_id={{ run_id }} | dag_run={{ dag_run }}"',
    dag=dag)
```

9. Dashboards with Apache Superset (incubating)

Superset

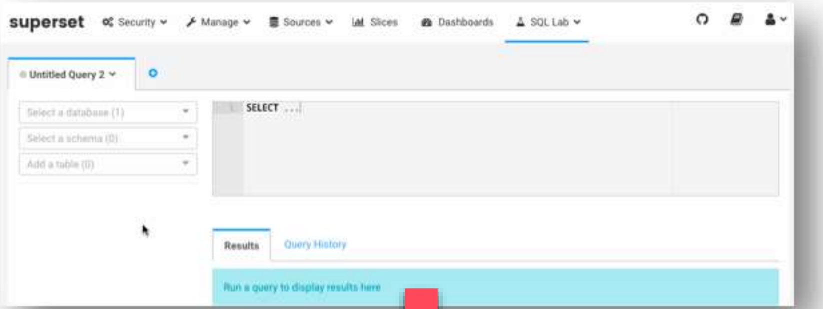
Open-sourced at Airbnb by Max Beauchemin

- Ad-hoc queries
- Connections: MySQL, SqlAlchemy, Presto, Hive, etc.
- Dashboards

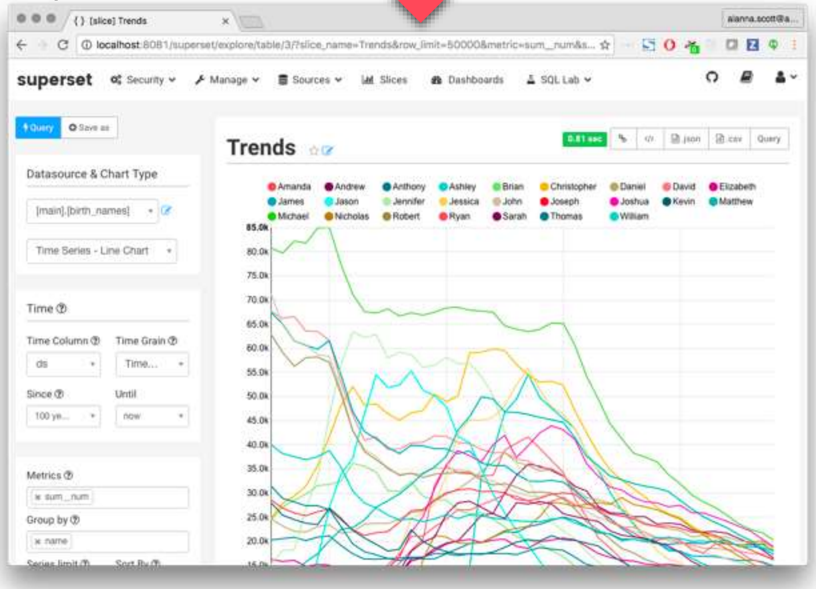
9. Dashboards with Apache Superset (incubating)

Superset

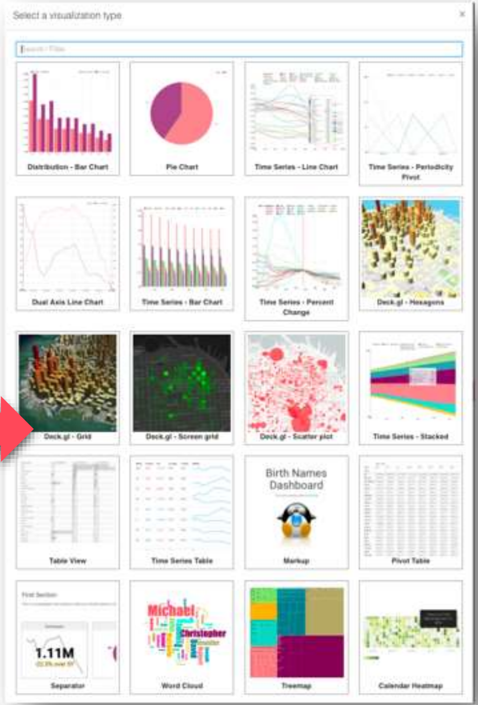
Query Data



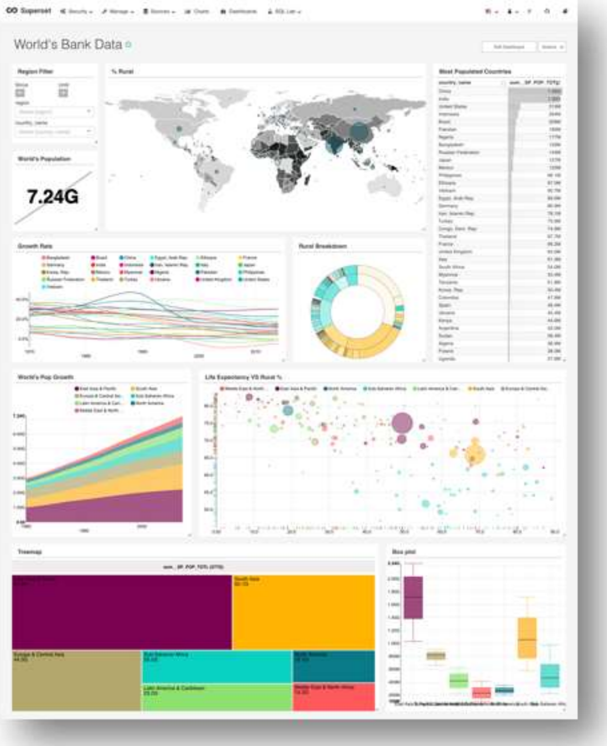
Explore Data



Visualize It



Share It



10. Automated Maintenance

- Rotate unhealthy nodes (too many failed volumes)
    - Decommission
    - Stop role, remove from cluster
    - Replace
- Clean space

Thank you

References

- https://medium.com/airbnb-engineering/superset-scaling-data-access-and-visual-insights-at-airbnb-3ce3e9b88a7f
- https://blog.cloudera.com/blog/2012/09/automating-your-cluster-with-cloudera-manager-api/
- https://hortonworks.com/blog/hood-ambari-metrics-grafana/
- https://community.hortonworks.com/storage/attachments/7372-ru-eu-prechecks-details.pdf
- https://community.hortonworks.com/articles/54944/stack-upgrade-pre-checks-purpose-and-remediation.html
- https://docs.datadoghq.com/developers/agent_checks/
- https://www.datadoghq.com/blog/collecting-hadoop-metrics/#namenode-and-datanode-metrics-via-jmx

Images:

- https://grafana.com/plugins/praj-ams-datasource
- https://www.estabil.is/post/o-que-e-devops
- http://warriorfitness.org/tag/top-10-list/
- https://superset.apache.org/