Apache Aurora
an Introduction

10.11.2015
@ErbStephan

Apache Aurora

Mesos framework for the deployment and scaling of stateless and fault tolerant services in a datacenter

Apache Mesos

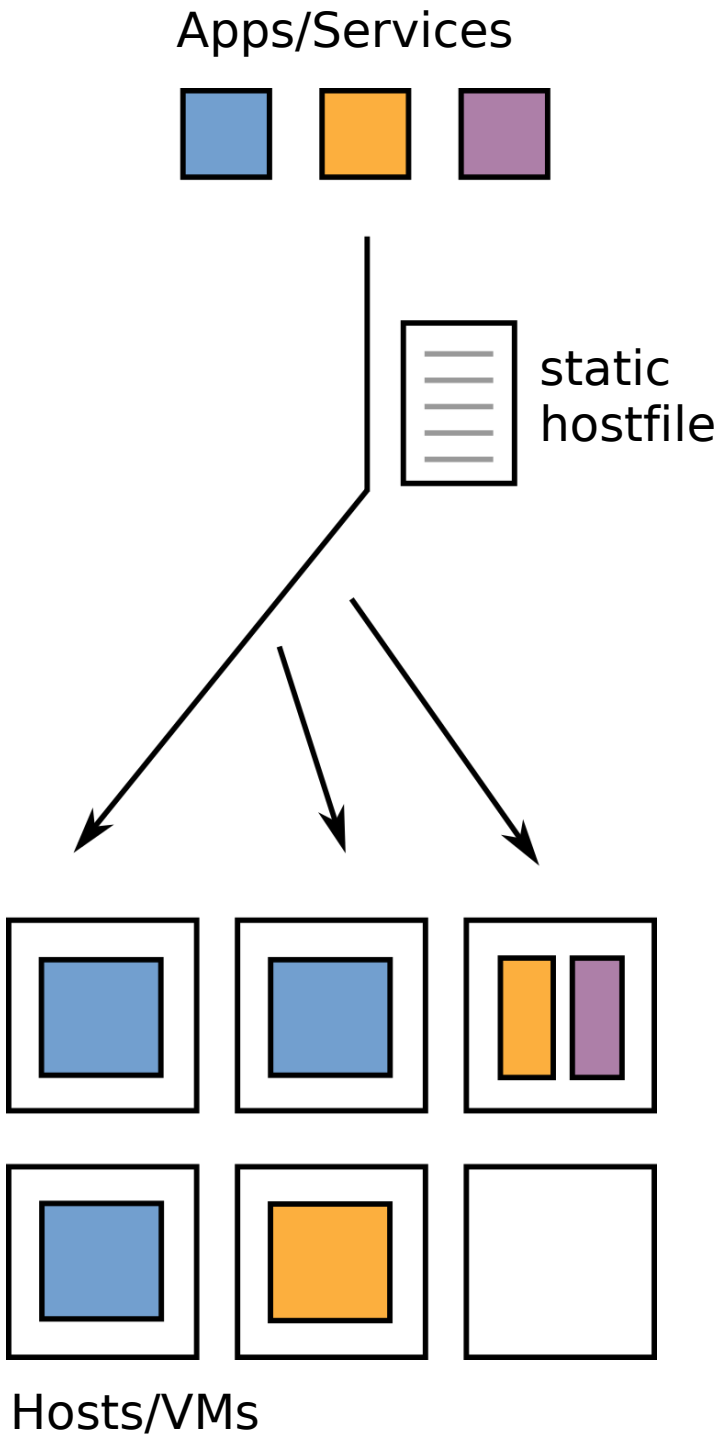
Cluster manager providing fault-tolerant, fine-grained multitenancy via containers

Apache Aurora

„distributed supervisord“

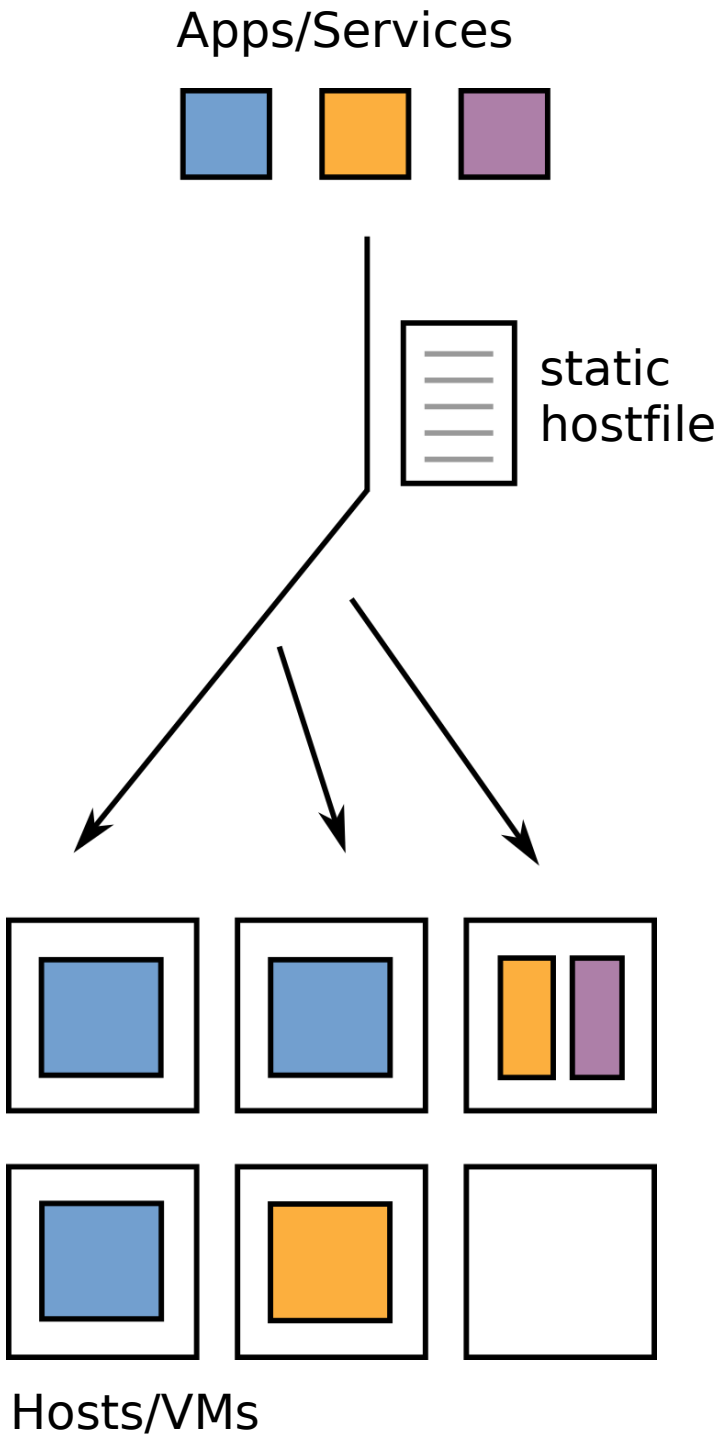
I speak ansible and deploy VMs for breakfast...

...why should I care?



Aspects

- Configuration/package management
- Deployment
- Naming/discovery
- Monitoring

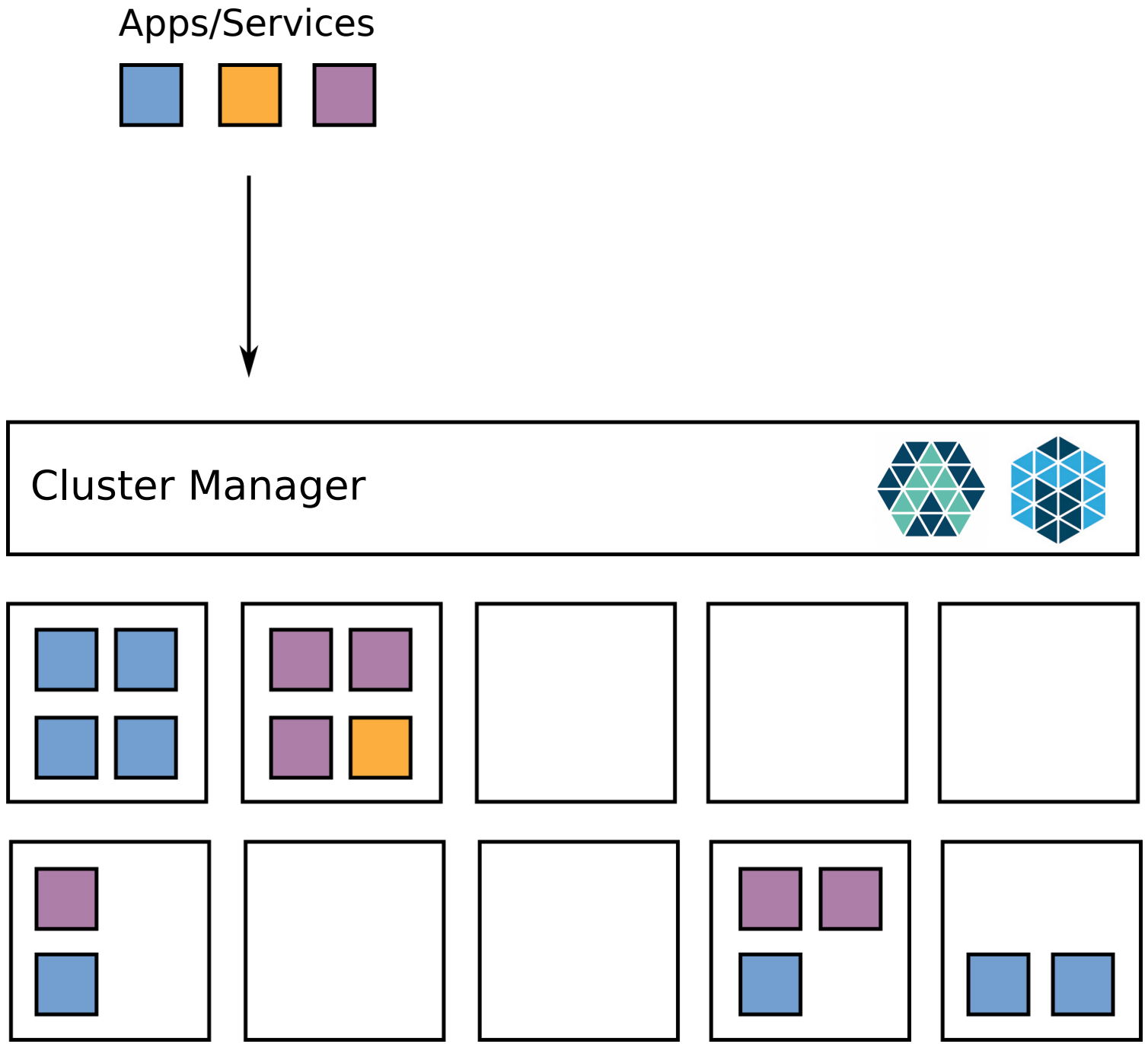


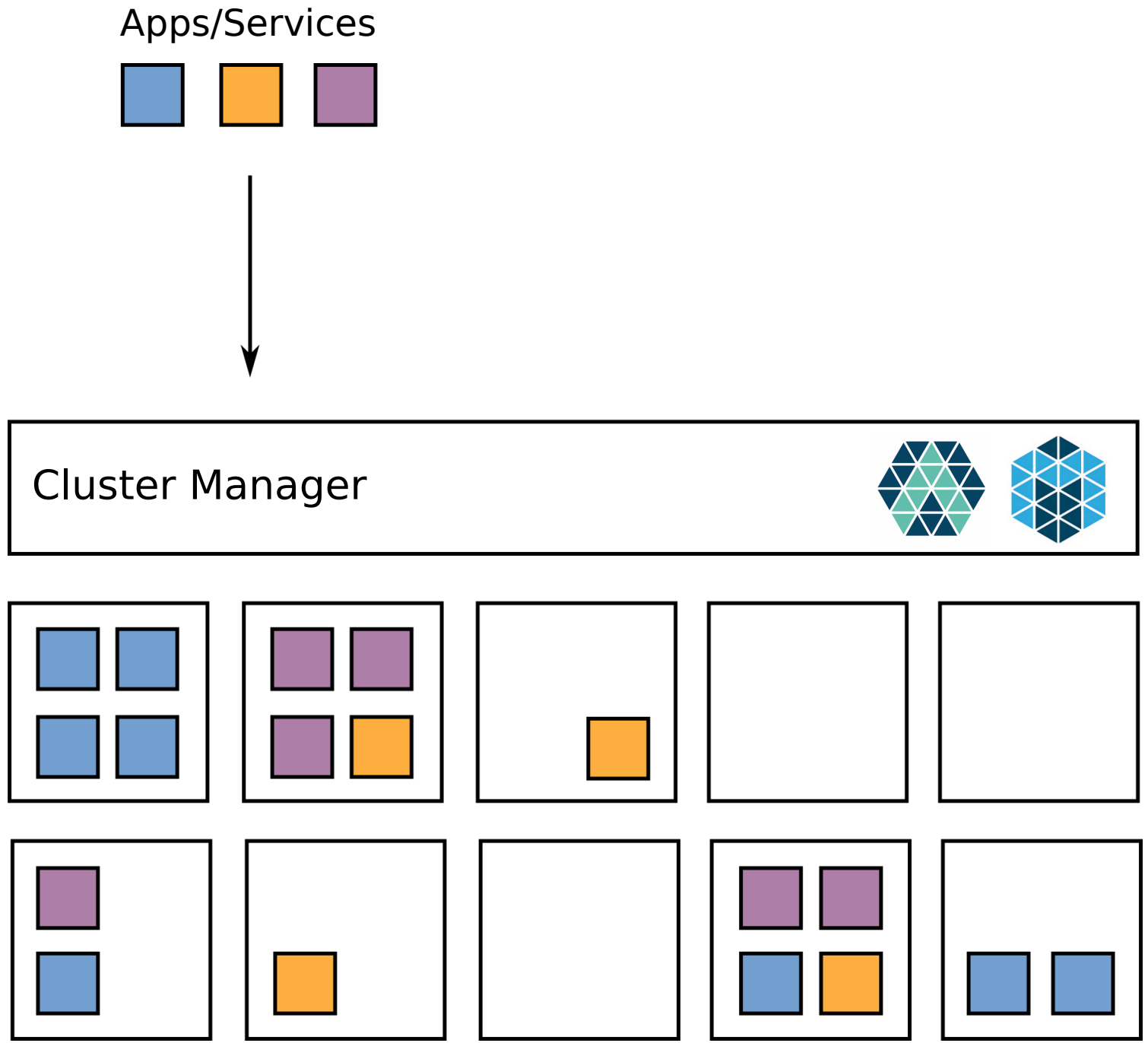
Aspects

- Configuration/package management
- Deployment
- Naming/discovery
- Monitoring

Challenges

- Failures
- Maintenance
- Utilization
- Scaling (instances + teams)


Apps/Services
Cluster Manager


Apps/Services
Cluster Manager

User perspective
(above the fold)

File webserver.aurora:

```
server = Process(
  name = 'simplehttp',
  cmdline = 'python -m SimpleHTTPServer {{thermos.ports[http]}}'
)
task = SequentialTask(
  processes = [server],
  resources = Resources(cpu=2, ram=2*GB, disk=4*GB)
)
jobs = [
  Service(
    task=task,
    constraints = {'host': 'limit:1'},
    instances=4,
    cluster='rz1', role='www', environment='test', name='webserver'
  ),
]
```

Commandline Usage:

```
$ aurora update start rz1/www/test/webserver webserver.aurora
```

Aurora Job UI

Active tasks (4) | Completed tasks (27) | All tasks

Configuration Overview

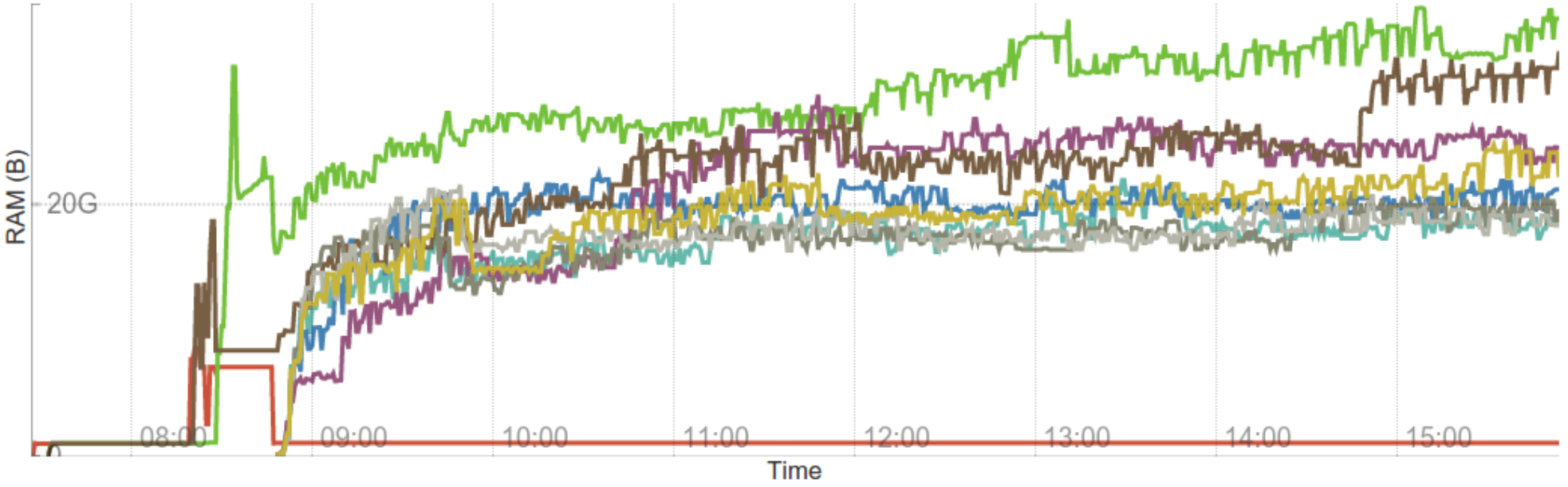
0-3

show config

Instance	Status	Host
0	− 17 days ago - RUNNING • 10/23 14:02:39 LOCAL • PENDING • 10/23 14:02:39 LOCAL • ASSIGNED • 10/23 14:02:40 LOCAL • STARTING • Initializing sandbox. • 10/23 14:02:41 LOCAL • RUNNING	[redacted]
1	+ 11 minutes ago - RUNNING	[redacted]
2	+ 11 minutes ago - RUNNING	[redacted]
3	+ 11 minutes ago - RUNNING	[redacted]

Previous 1 Next

Instance Monitoring
(e.g. via Prometheus)



User-centric Features

- long-running services
- cron jobs
- adhoc jobs
- rolling job updates, with automatic rollback
- service announcement in ZooKeeper
- scheduling constraints
- Python-based configuration language
- Docker support (optional)

Operator perspective
(below the fold)

Maintenance API

```
$ aurora_admin host_deactivate --filename=hosts.txt rz1

$ aurora_admin host_drain --filename=hosts.txt rz1

$ aurora_admin host_activate  --filename=hosts.txt rz1
```

Cluster-wide Monitoring
(e.g. via Prometheus)

Operator-centric Features

- high-availability
- multi-user support
- maintenance primitives (SLA-based)
- resource quotas and preemption
- instrumented for monitoring and debugging

Implications
(good and bad)

Opinionated Design

- Services are stateless (as they should be...)
- No snowflake machines (unless...)

Enforces best practices but does not fit every usecase

Gutefrage.net

- 6 nodes

Twitter

- 10.000s nodes per cluster
- 40% of their data center

Highly fault-tolerant but not foolproof

Stephan Erb @ErbStephan · 21. Okt.

Uuups. @ApacheMesos deploy vs me 1:0. "Slave asked to shut down by master because 'Slave is not authenticated'"

2 1

;-)

Questions
Demo
Pizza


If you think of
• literature when you hear **Kafka**
• mythology when you hear **Cassandra**
• animals when you hear **Zookeeper**
… then have a nice day.
If you think of distributed systems, then join us!
www.blue-yonder.com
blue yonder