RAND Safety and Justice Program

Autopsy as a Service – Distributed Forensic Compute That Combines Evidence Acquisition and Analysis

Presentation to OSDFCon 2016

Dan Gonzales, Zev Winkelman, John Hollywood, Dulani Woods, Ricardo Sanchez, Trung Tran

October 2016

This project was supported by Award No. 2014-IJ-CX-K102, awarded by the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice.

Objective and Background

- **RAND has been funded by the National Institute of Justice to accelerate the processing of digital forensics data**

- **Objective: Develop a Digital Forensics Compute Cluster (AutopsyCluster)**
  - – Based on open source, state of the art software
  - – Reduce processing time and storage costs

- **We have chosen Autopsy as a core component of AutopsyCluster**
  - – “Autopsy as a Service”

Vision

- Provide law enforcement with a cost effective and efficient digital forensics analysis capability

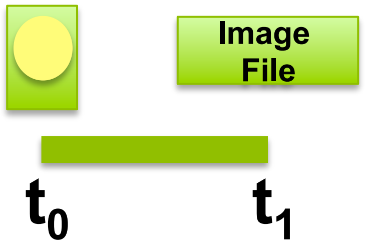
- Combine data ingest and analysis steps to speed up the digital evidence analysis process using
  - Distributed computing tools
  - Cloud computing services

- Approach designed to
  - Reduce infrastructure cost
  - Stand up infrastructure only when needed
  - Access infrastructure to perform multiple analyses in parallel

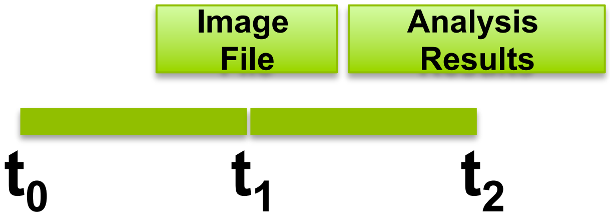
To implement the Vision We Stream Data into the Cloud

Old Way

- Step 1: make copy

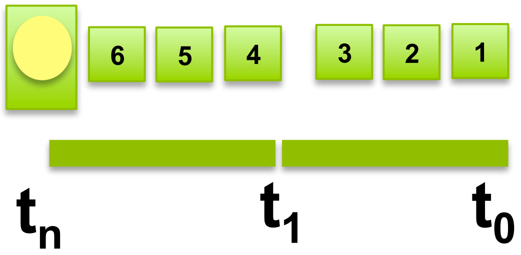


- Step 2: analyze image on standalone workstation

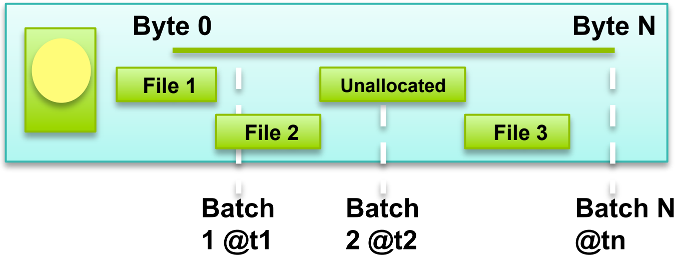


New Way

- Step 1: start stream



- Step 2: process stream on the fly in micro batches



If we can keep up with the data coming off the disk, we are processing as fast as is physically possible

Outline

- Objectives and vision
- Architecture
- Initial results
- Lessons Learned
- How to use AutopsyCluster
- Beta testing

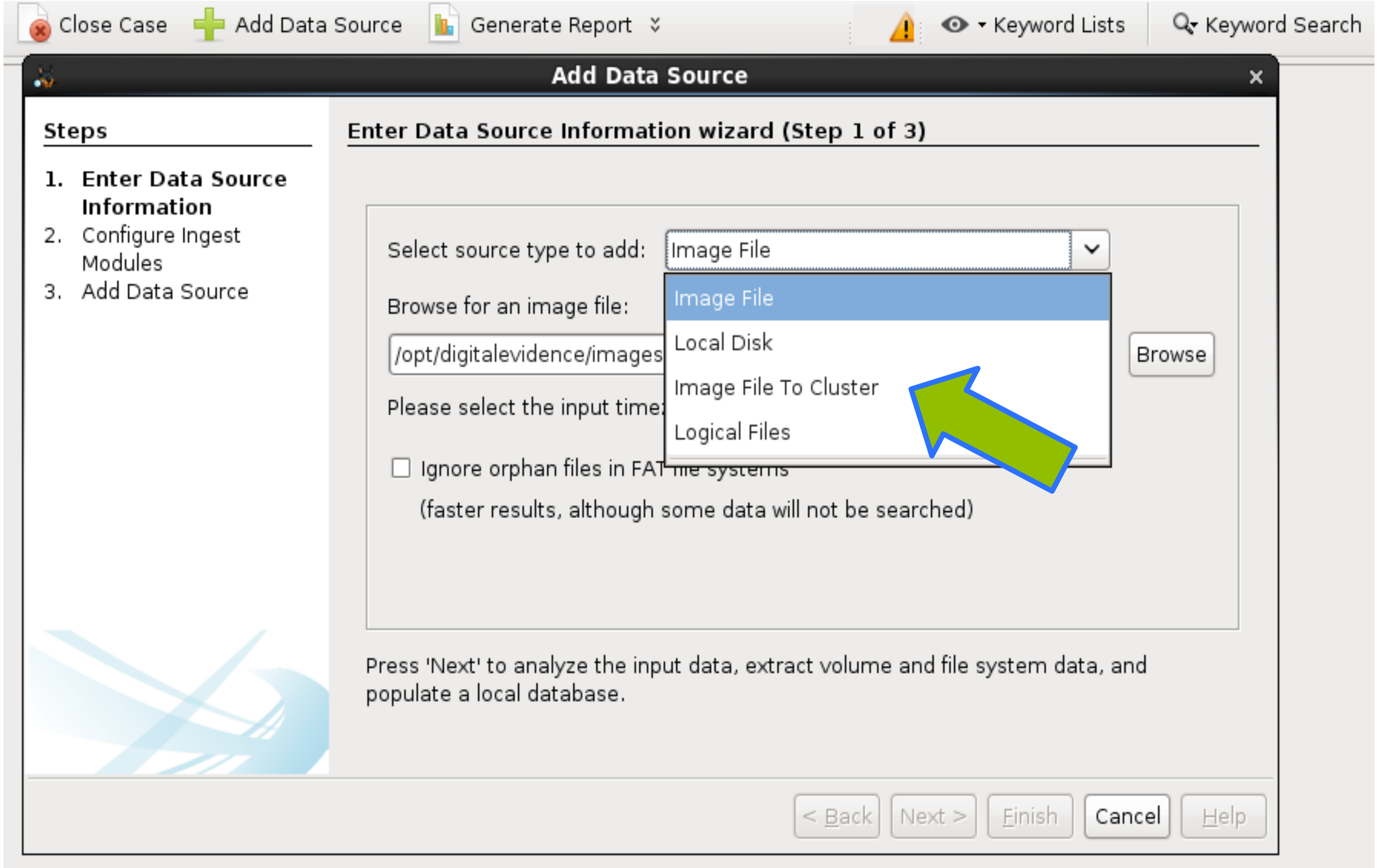
The Forensics Analysis Functions of AutopsyCluster are Based on Autopsy[a]

- Basis Technology has developed a version of Autopsy for collaborative forensics analysis over a network[b]
  - – We chose this version because it is designed to work over a network with supporting servers
- AutopsyCluster designed to run forensics processing tasks in parallel at near "streaming speed"
  - – Speed at which disk blocks are read from evidence disk
  - – With dc3dd with USB 3.0 this is about 15 MBps
- We modified the *Autopsy* so it is a streaming application
  - – Integrated with Apache Spark[c] (cluster computing framework) and Apache Kafka[d] (messaging)
- Autopsy analysis modules read from the stream

Autopsy Sleuth Kit

[a] http://www.sleuthkit.org/autopsy/
[b] https://github.com/sleuthkit/autopsy
[c] http://www.sleuthkit.org/autopsy/
[d] http://www.postgresql.org/

User Interface for Autopsy Streaming Branch



Add Data Source

Steps

1. **Enter Data Source Information**
2. Configure Ingest Modules
3. Add Data Source

Enter Data Source Information wizard (Step 1 of 3)

Select source type to add: Image File

Browse for an image file:

/opt/digitalevidence/images

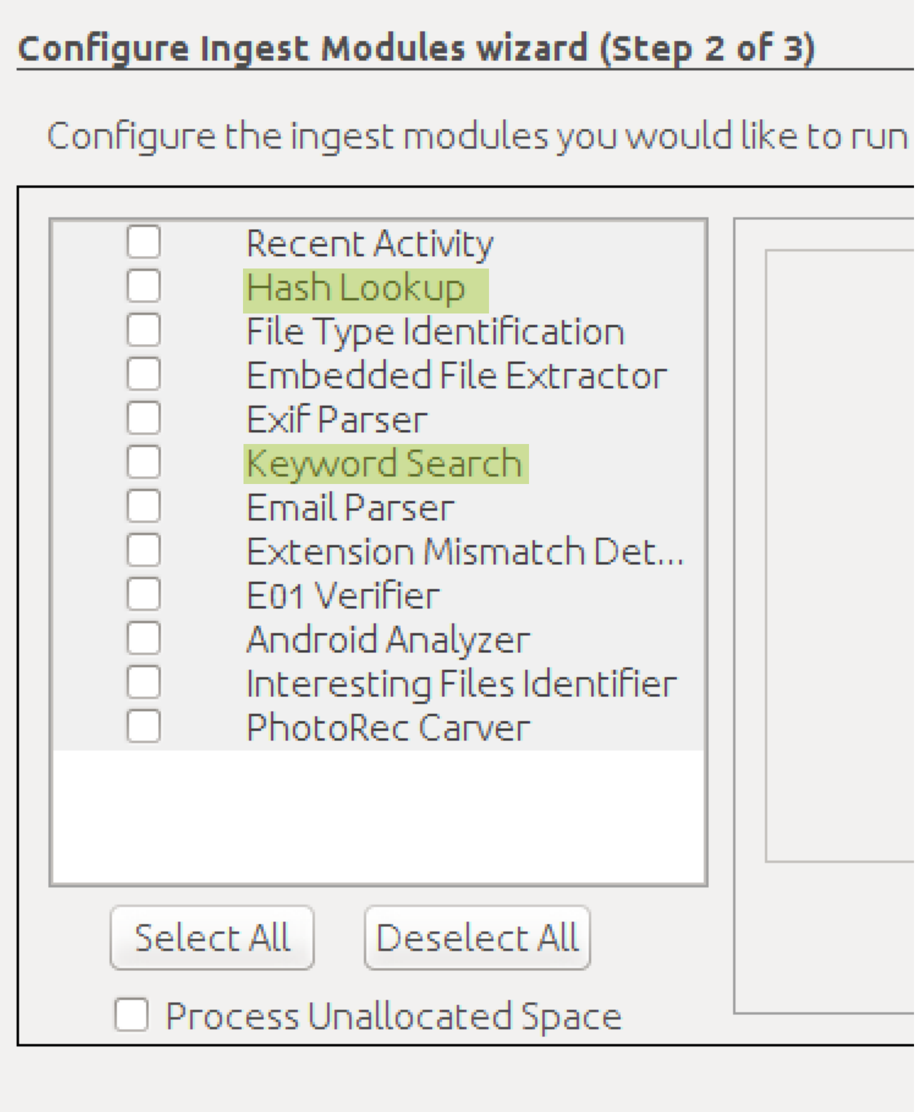
Please select the input time

☐ Ignore orphan files in FAT file systems

(faster results, although some data will not be searched)

Press 'Next' to analyze the input data, extract volume and file system data, and populate a local database.

Autopsy Modules For Autopsy Streaming Branch



Configure Ingest Modules wizard (Step 2 of 3)

Configure the ingest modules you would like to run

- □ Recent Activity
- □ Hash Lookup
- □ File Type Identification
- □ Embedded File Extractor
- □ Exif Parser
- □ Keyword Search
- □ Email Parser
- □ Extension Mismatch Det...
- □ E01 Verifier
- □ Android Analyzer
- □ Interesting Files Identifier
- □ PhotoRec Carver

□ Process Unallocated Space

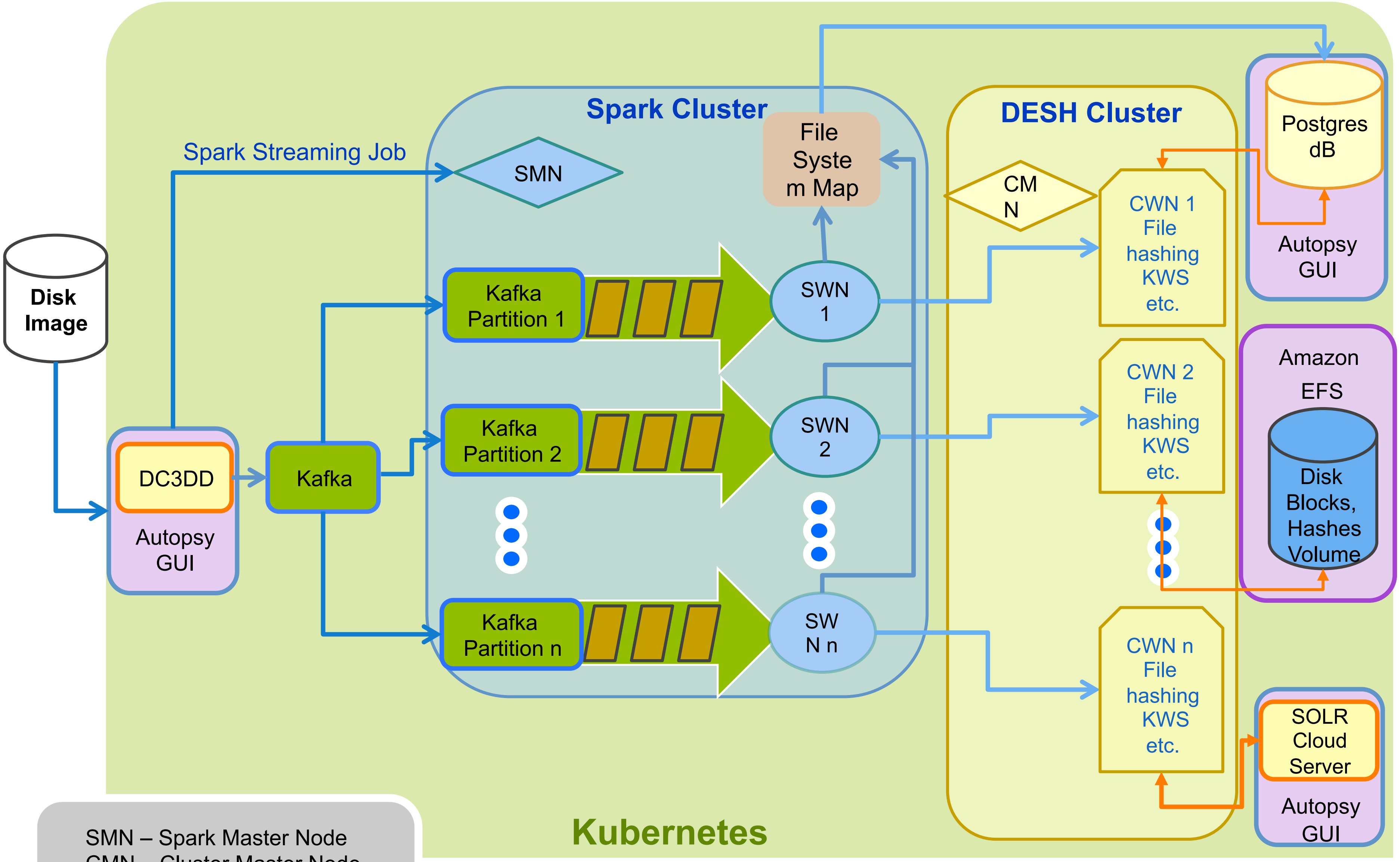
Currently Working in Spark:

- “Hash Lookup”
- “Keyword Search”
- Hardcoded configurations

Next Steps:

- Remaining modules starting with “Interesting Files Identifier”
- Implement configuration of modules with Autopsy UI

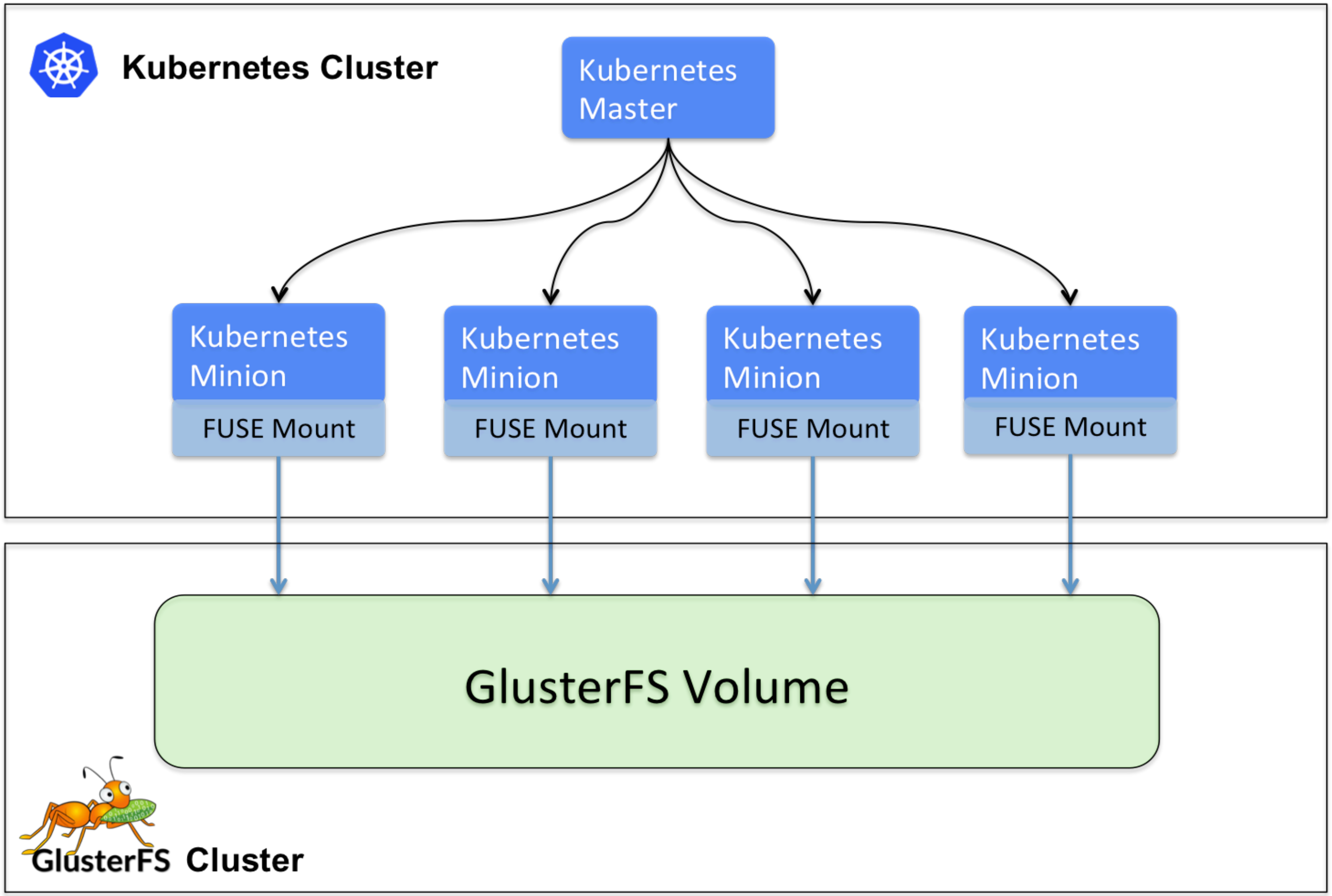
AutopsyCluster Architecture



SMN – Spark Master Node
CMN – Cluster Master Node
SWN – Spark Worker Node
CWN – Cluster Worker Node
KWS - Key Word Search

Gonzales and Winkelman-9 October 2016

Kubernetes + File Volumes



Kubernetes Cluster

Kubernetes Master

Kubernetes Minion

FUSE Mount

Kubernetes Minion

FUSE Mount

Kubernetes Minion

FUSE Mount

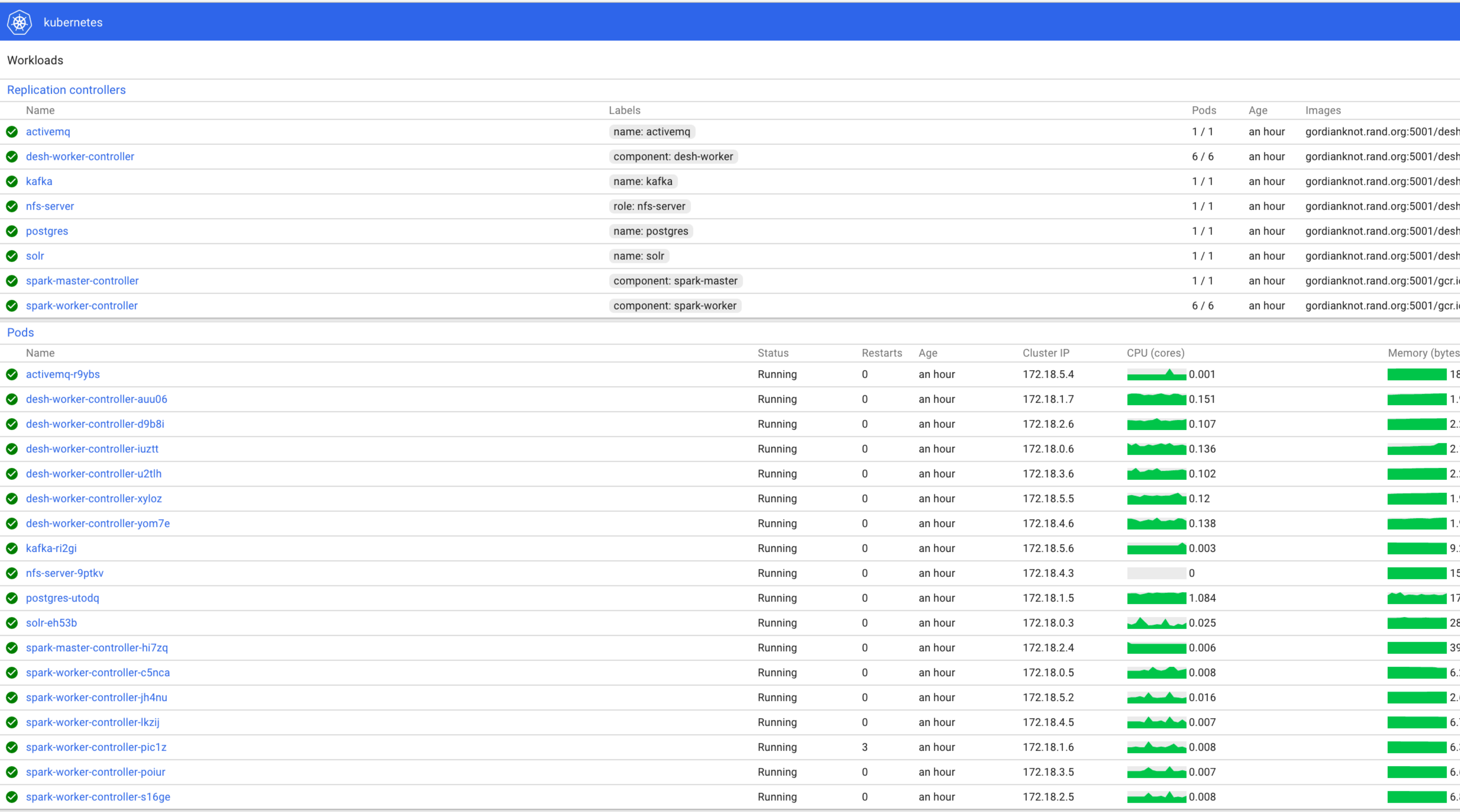
Kubernetes Minion

FUSE Mount

GlusterFS Volume

GlusterFS Cluster

AutopsyCluster Kubernetes Dashboard



kubernetes

Workloads

Replication controllers

	Name	Labels	Pods	Age	Images
✓	activemq	name: activemq	1 / 1	an hour	gordianknot.rand.org:5001/desh
✓	desh-worker-controller	component: desh-worker	6 / 6	an hour	gordianknot.rand.org:5001/desh
✓	kafka	name: kafka	1 / 1	an hour	gordianknot.rand.org:5001/desh
✓	nfs-server	role: nfs-server	1 / 1	an hour	gordianknot.rand.org:5001/desh
✓	postgres	name: postgres	1 / 1	an hour	gordianknot.rand.org:5001/desh
✓	solr	name: solr	1 / 1	an hour	gordianknot.rand.org:5001/desh
✓	spark-master-controller	component: spark-master	1 / 1	an hour	gordianknot.rand.org:5001/gcr.i
✓	spark-worker-controller	component: spark-worker	6 / 6	an hour	gordianknot.rand.org:5001/gcr.i

Pods

	Name	Status	Restarts	Age	Cluster IP	CPU (cores)	Memory (bytes
✓	activemq-r9ybs	Running	0	an hour	172.18.5.4	0.001	18
✓	desh-worker-controller-auu06	Running	0	an hour	172.18.1.7	0.151	1.
✓	desh-worker-controller-d9b8i	Running	0	an hour	172.18.2.6	0.107	2.
✓	desh-worker-controller-iuztt	Running	0	an hour	172.18.0.6	0.136	2.
✓	desh-worker-controller-u2tlh	Running	0	an hour	172.18.3.6	0.102	2.
✓	desh-worker-controller-xyloz	Running	0	an hour	172.18.5.5	0.12	1.
✓	desh-worker-controller-yom7e	Running	0	an hour	172.18.4.6	0.138	1.
✓	kafka-ri2gi	Running	0	an hour	172.18.5.6	0.003	9.
✓	nfs-server-9ptkv	Running	0	an hour	172.18.4.3	0	15
✓	postgres-utodq	Running	0	an hour	172.18.1.5	1.084	17
✓	solr-eh53b	Running	0	an hour	172.18.0.3	0.025	28
✓	spark-master-controller-hi7zq	Running	0	an hour	172.18.2.4	0.006	39
✓	spark-worker-controller-c5nca	Running	0	an hour	172.18.0.5	0.008	6.
✓	spark-worker-controller-jh4nu	Running	0	an hour	172.18.5.2	0.016	2.
✓	spark-worker-controller-lkzij	Running	0	an hour	172.18.4.5	0.007	6.
✓	spark-worker-controller-pic1z	Running	3	an hour	172.18.1.6	0.008	6.
✓	spark-worker-controller-poiur	Running	0	an hour	172.18.3.5	0.007	6.
✓	spark-worker-controller-s16ge	Running	0	an hour	172.18.2.5	0.008	6.

Outline

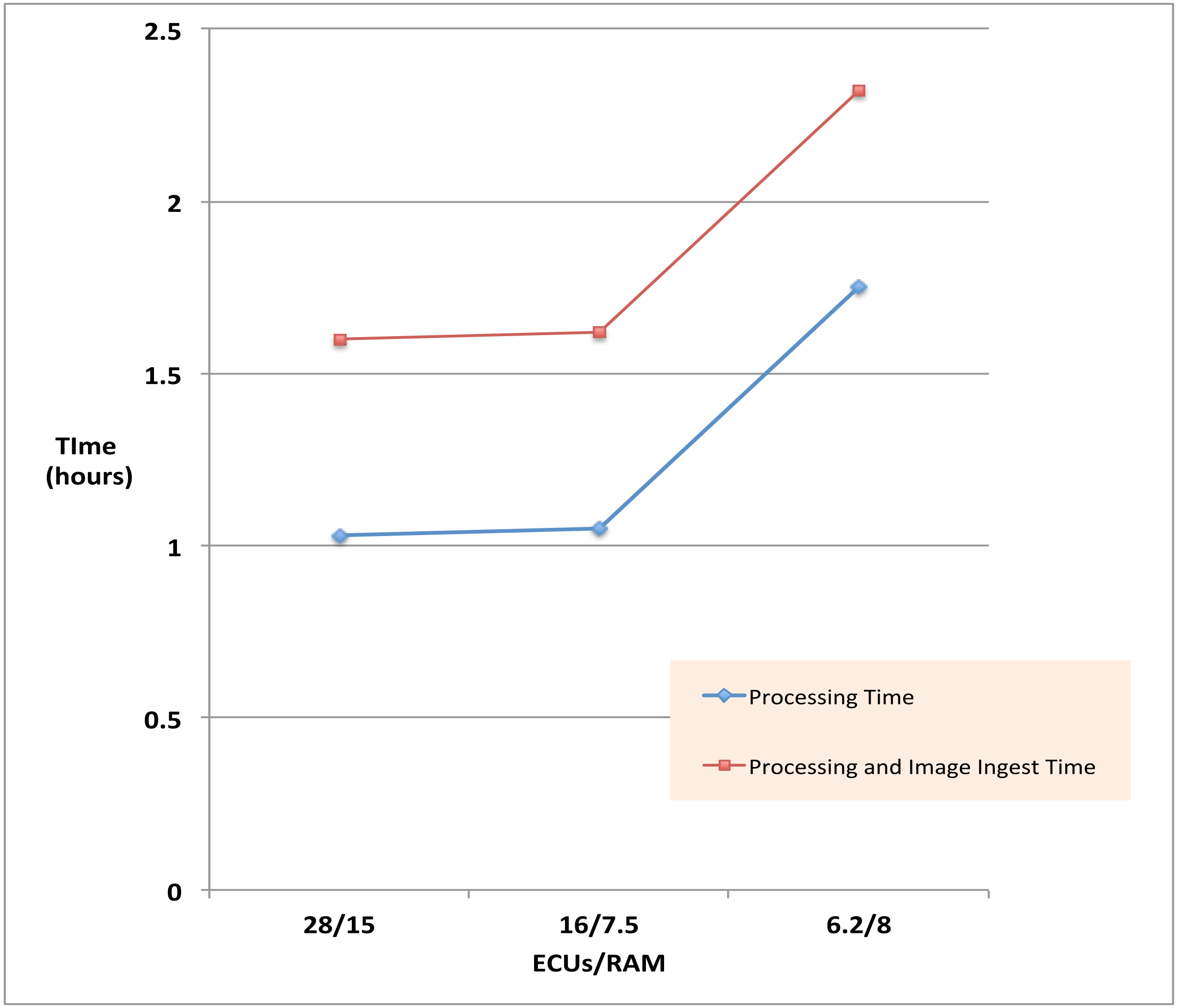
- Objectives and vision
- Architecture
- ➡ Initial results
- Lessons Learned
- How to use AutopsyCluster
- Beta testing

Forensic Images We are Using In Performance Testing

Image	**Size**	**Source**
Rhino Hunt	250 MB	NIST (CFReDS)
Data Leakage	20 GB	NIST (CFReDS)
NPS DOMEX Users, 2009	40 GB	Digital Corpora
NPS 1weapondeletion, 2011	75 GB	Digital Corpora
NPS 2weapons, 2011	253 GB	Digital Corpora
NPS 2 TB, 2011	2 TB	Digital Corpora

- **Initial tests conducted on**
  - **Stand alone machines**
  - **A typical RAND server (Digital Evidence)**
  - **Amazon Web Services (AWS)**

Stand Alone Autopsy Results on AWS Windows Virtual Machines (VMs)



40 GB Hard Disk Image

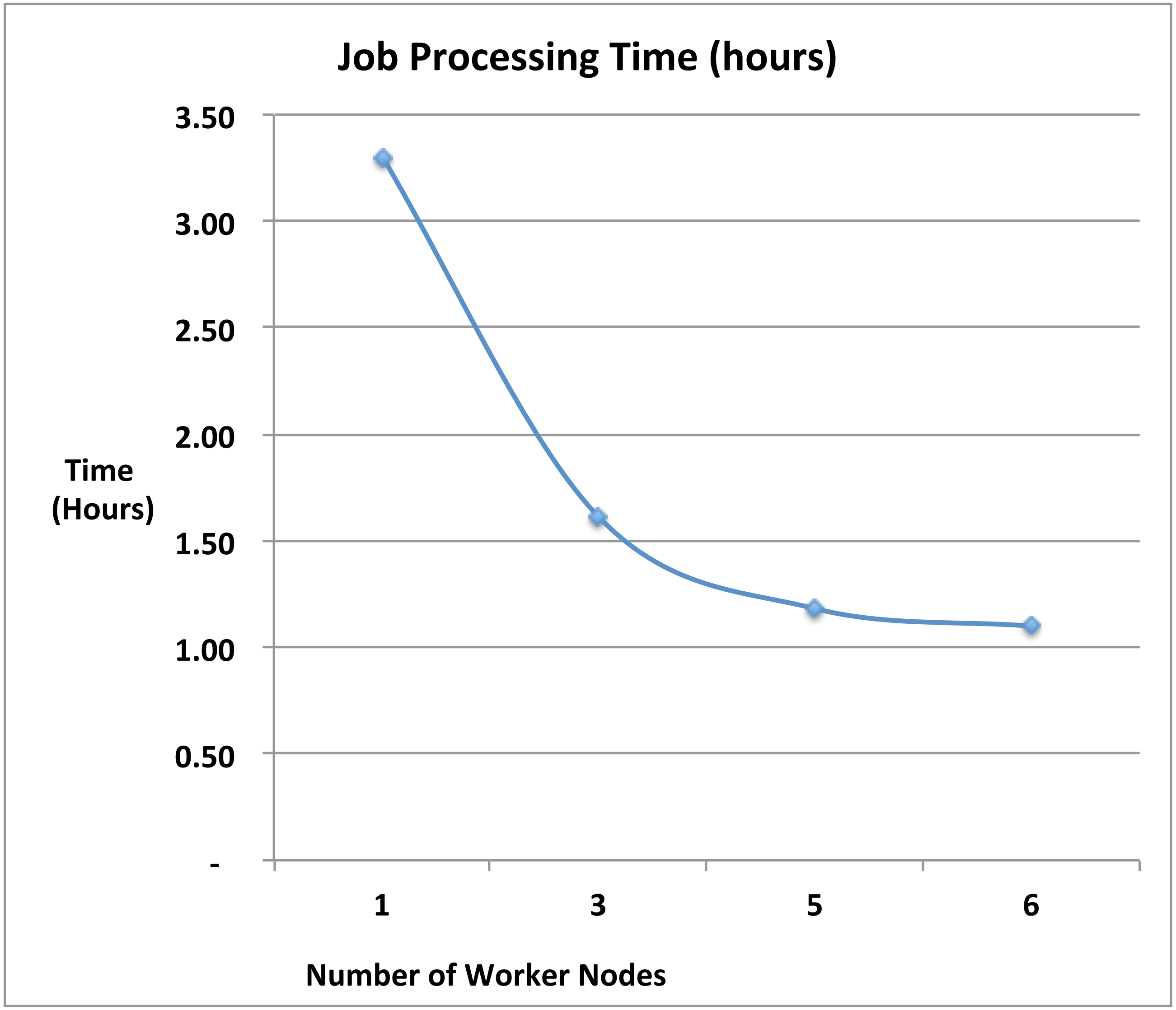
Ingestion, hashing, Key Word Search

ECU = Elastic Compute Unit = 2007, 1 GHz CPU

- **Autopsy performances varies based on machine capabilities**
- **All results are for raw HD images already ingested in cloud**

RAND

AutopsyCluster Results on a Single Server for a 40 GB Hard Disk Image



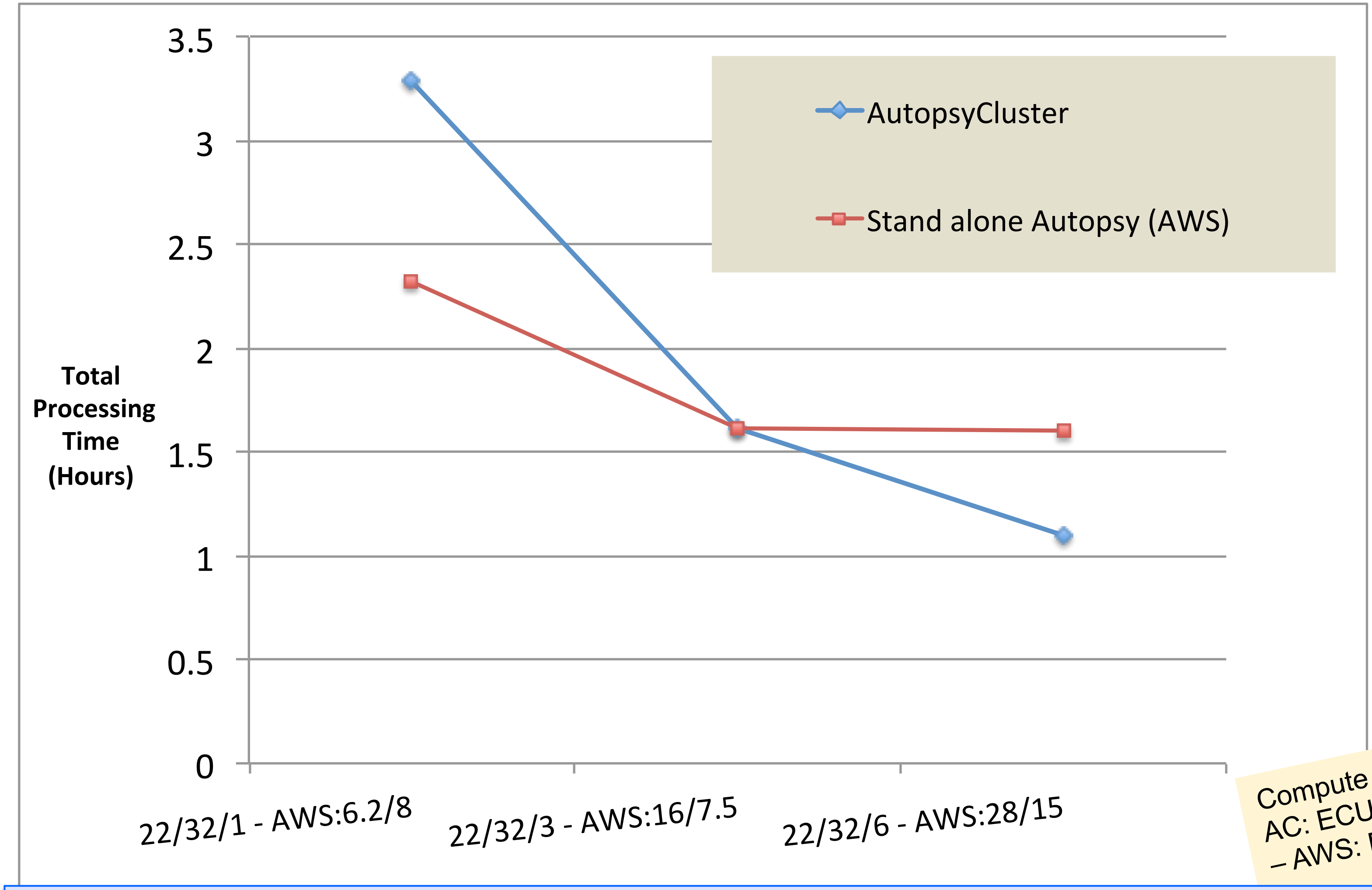
Local server equivalent
To 22 ECUs with
32 GB RAM (22/32)

Ingestion, hashing,
Key Word Search

Performance roughly
Comparable with
stand alone Autopsy
With 5 or more
worker nodes

Number of worker
nodes constrained by
memory limitations
on specific server
used

Stand Alone Autopsy (SAA), AutopsyCluster (AC) Performance Comparison for a 40 GB Drive

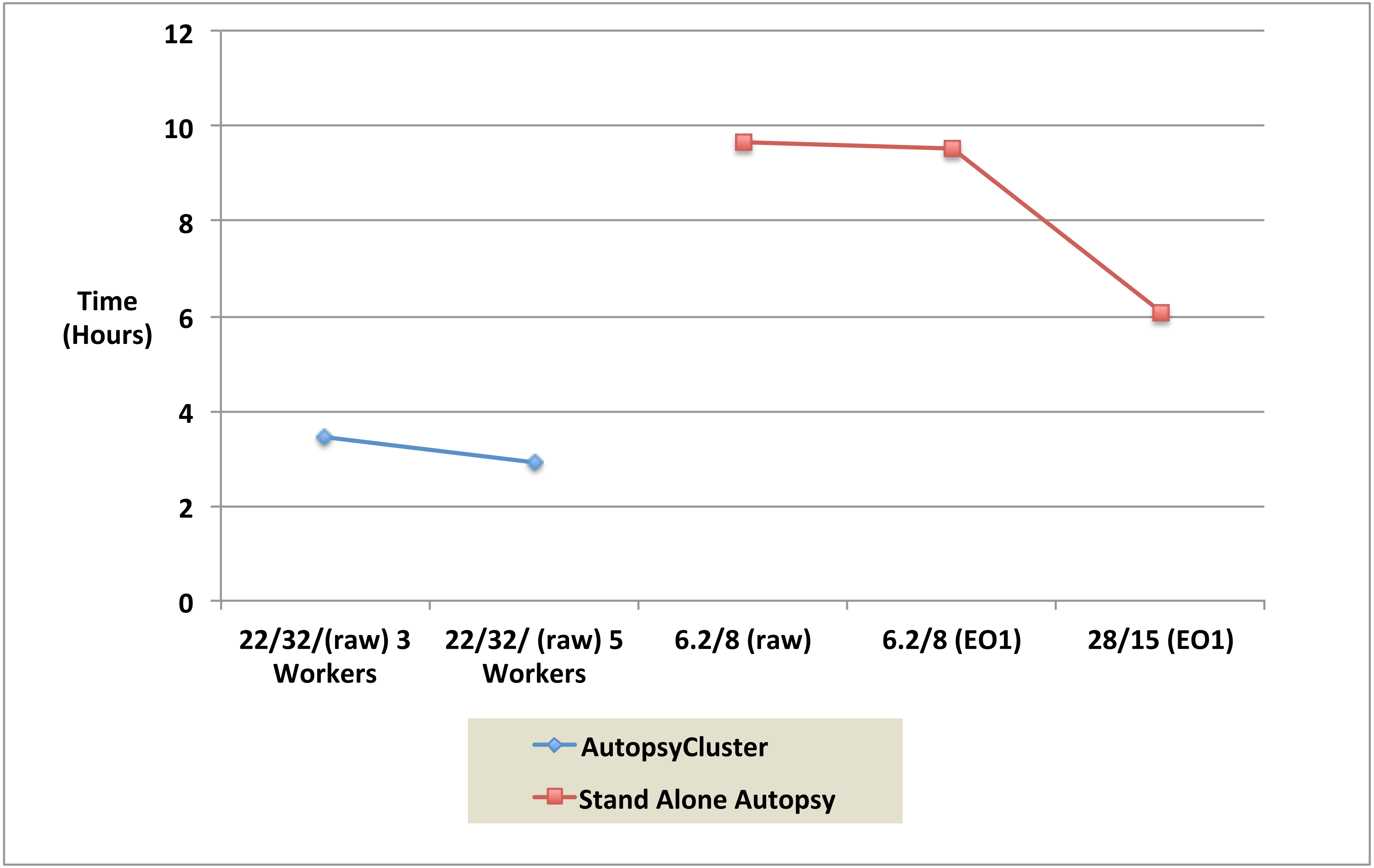


Compute Resources:
AC: ECU/RAM/#workers
– AWS: ECU/RAM

- As Worker nodes are added to the Server AutopsyCluster Performance improves; With 6 worker nodes AutopsyCluster is faster than Autopsy

RAND

Stand Alone Autopsy and AutopsyCluster Results on AWS for 75 GB Disk Images



Outline

- Objectives and vision
- Architecture
- Preliminary test results
- Lessons learned
- How to use AutopsyCluster
- Beta testing

Moving to the Cloud Can Present a Number of Challenges

- Good communications links to the cloud are essential for good performance

- Testing at RAND showed that communications links to AWS were frequently congested, adding time delays

- It is possible to purchase a direct link to AWS for many ISP links, which may improve performance significantly

Outline

- Objectives and vision
- Architecture
- Preliminary test results
- Lessons Learned
- How to use AutopsyCluster
- Beta testing

Four Ways to Use Fully Operational AutopsyCluster

- **Acquire and ingest locally on a single machine**
  - **Advantage is acquisition and analysis at the same time**
- **Acquire locally and ingest on local private distributed computing (e.g., on premises datacenter)**
- **Acquire locally, ingest remotely (e.g., cloud) and transmit via streaming**
- **Ship drive(s) to cloud service provider for remote acquisition, and multiple side-by-side ingest "jobs"**
  - **We plan to investigate feasibility with AWS**

AutopsyCluster Provides Scalable Options for Data Acquisition and Ingest

Option	Streaming	Distributed	Cloud
Autopsy Standalone	No	No	No
AutopsyCluster on premise single machine	Yes	No	No
AutopsyCluster on premise data center	Yes	Yes	No
Autopsy on premise – remote data center	Yes	Yes	Yes
Ship drives for AutopsyCluster processing in Cloud	No	Yes	Yes

How Much Would Acquisition and Ingest of a 1TB Drive Cost on AWS?

- **Example for a 1 TB drive:**
  - **Total hourly rate for 6 nodes (2 CPUs ea, 15GB RAM ea): $1**
  - **Total hourly rate for 6 Linux SSD "disks" (32 GB ea): $0.03**
  - **Total hourly rate for 2 TB of "elastic" storage (need 2x): $0.83**
  - **Run time to extract and stream 1TB at 15MB/s: ~19 hours (includes time for "setup" and "teardown" of the cluster)**
- **Total "cloud" cost to acquire and ingest:**

  (1 + 0.03 + 0.83)/hour * 19 hours = ~$35
- **Immediate access storage for uncompressed acquired image and case file data (1.2 TB): $36/month**
- **Delayed access archive storage (1.2 TB): $8/month**

Where Can You Get AutopsyCluster?

- **We still have to clean up the code and document it for broader use**

- **It will be posted at**
  - **https://github.com/orgs/RANDCorporation/AutopsyCluster**

Outline

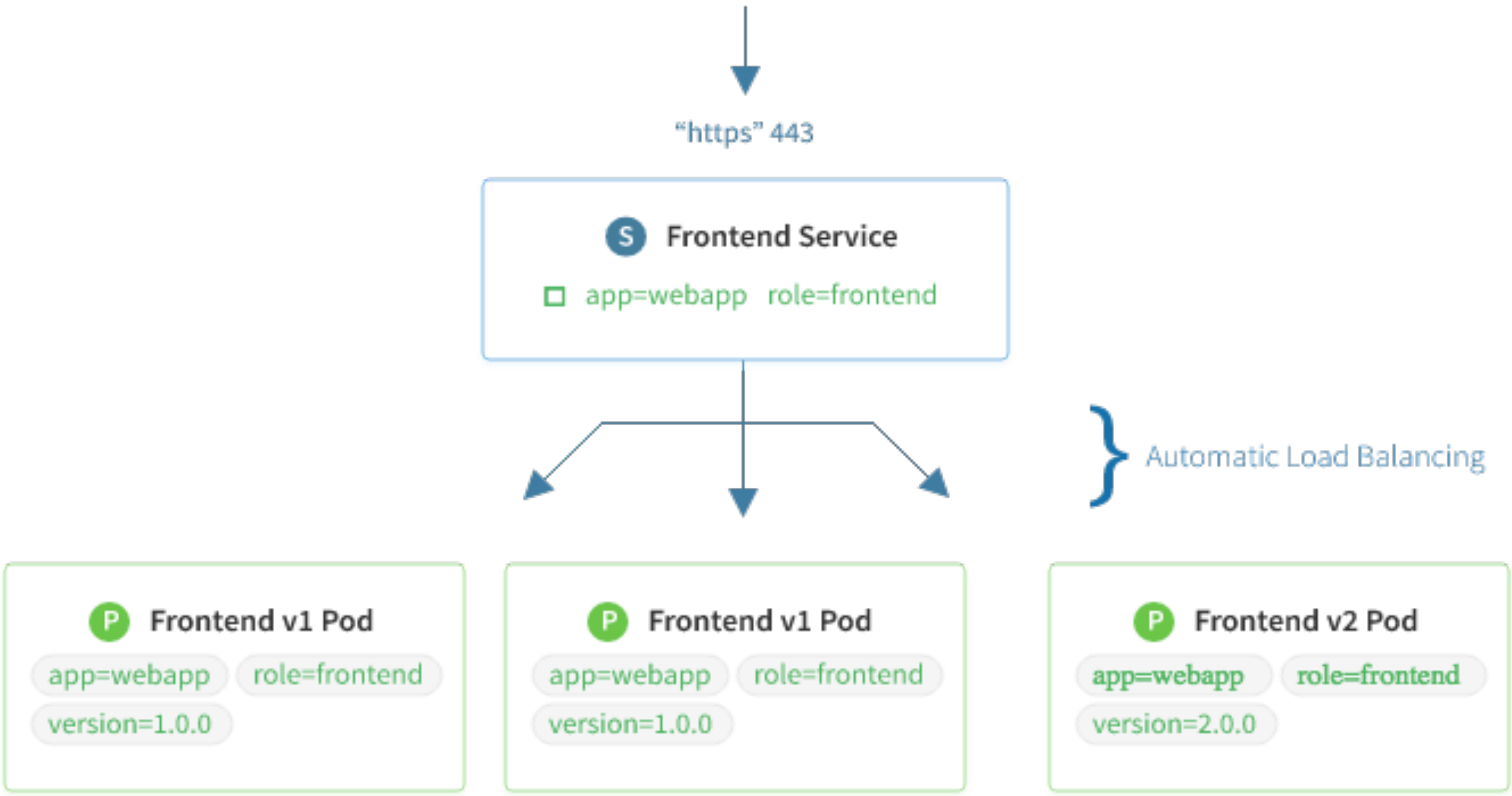
- Objectives and vision
- Architecture
- Preliminary test results
- Lessons Learned
- How to use DIGIFORC2
- ➡ Beta testing

We are Looking for Law Enforcement (LE) Partners as Beta Testers

- **RAND will conduct testing, training, and evaluation with local LE**
- **Objectives of beta testing are to:**
  - **Identify performance bottlenecks found during evaluation**
  - **Provide feedback on the user interface**
  - **Simplify system configuration in response to LE feedback**
- **We plan to use AWS for testing, but are open to other cloud candidates preferred by LE organizations**

Back Ups

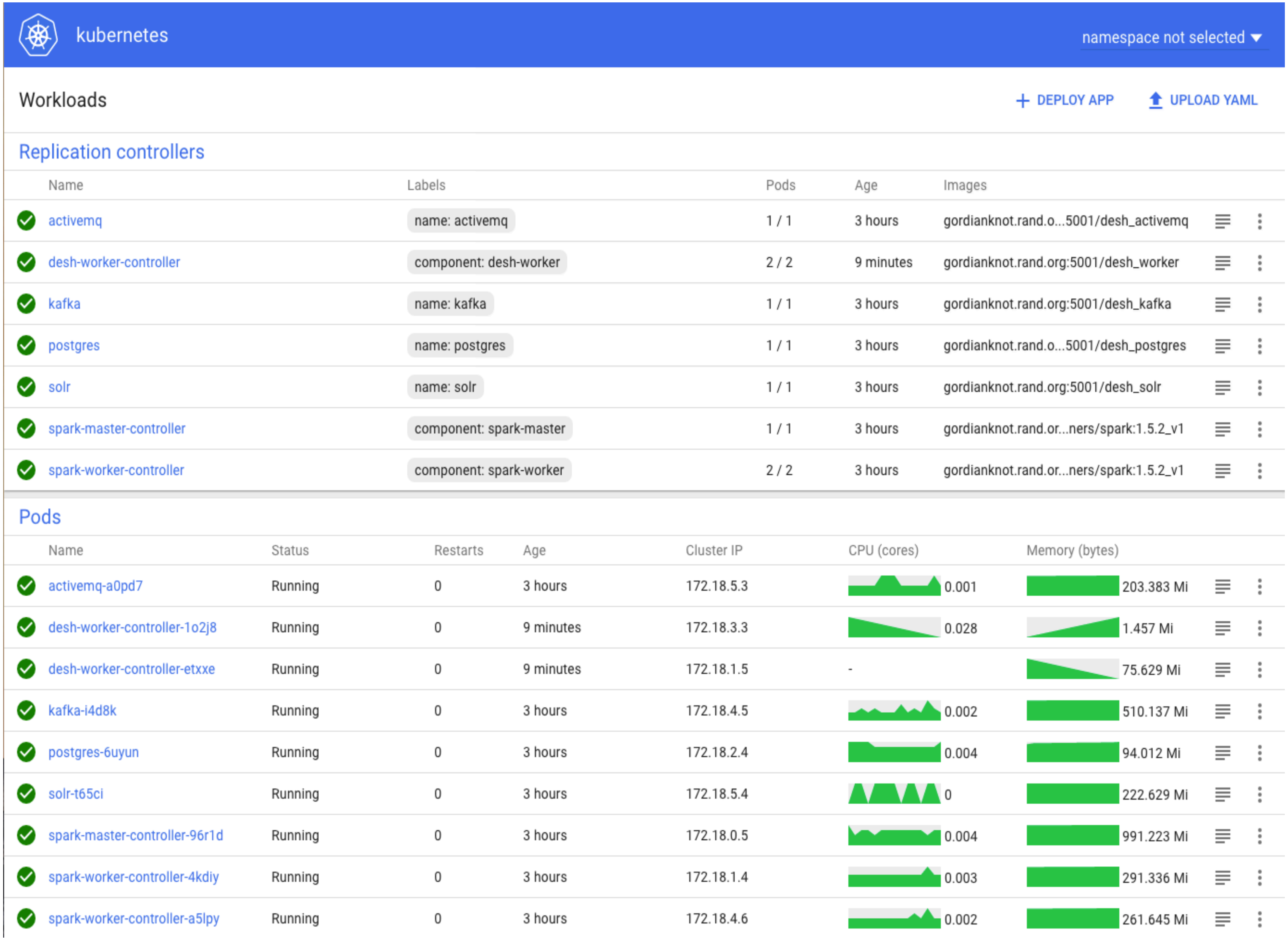
Kubernetes Can Provide Load Balancing



Overview of Project Tasks

1. Develop an appropriate cluster processing architecture
2. Integrate Autopsy with the cluster processor
3. Chain of custody analysis
4. Beta testing with law enforcement partners
5. Post DIGIFORC2 (Autopsy streaming branch) on Github

Kubernetes DIGIFORC2 Dashboard



Workloads

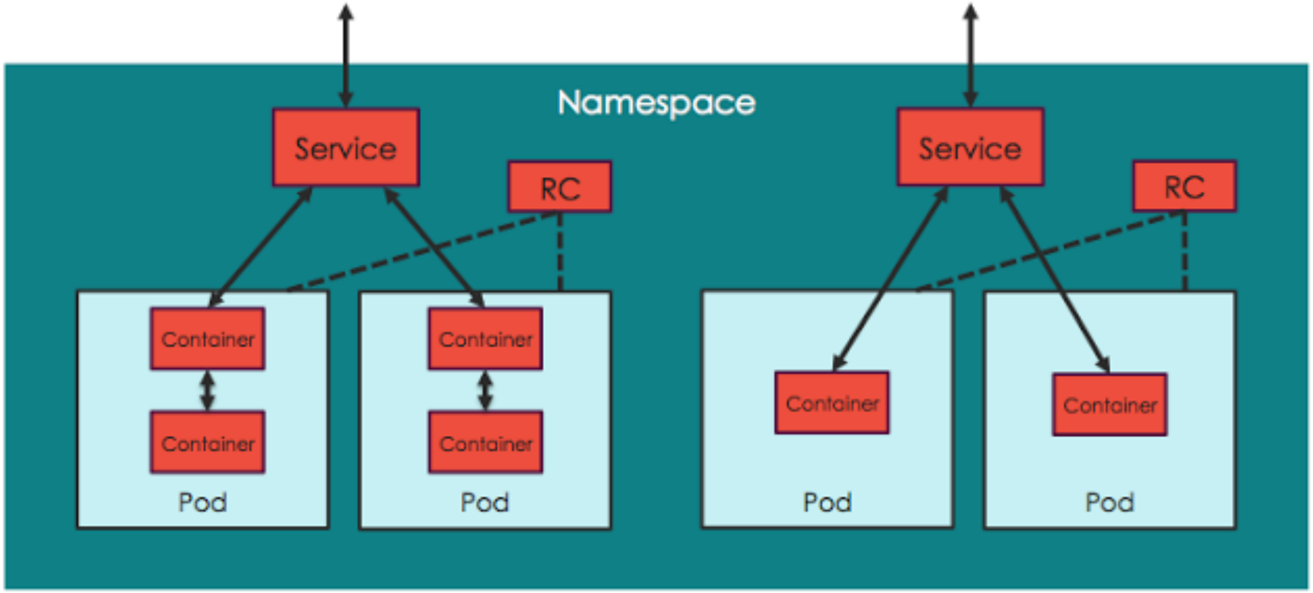
Replication controllers

	Name	Labels	Pods	Age	Images
✓	activemq	name: activemq	1 / 1	3 hours	gordianknot.rand.o...5001/desh_activemq
✓	desh-worker-controller	component: desh-worker	2 / 2	9 minutes	gordianknot.rand.org:5001/desh_worker
✓	kafka	name: kafka	1 / 1	3 hours	gordianknot.rand.org:5001/desh_kafka
✓	postgres	name: postgres	1 / 1	3 hours	gordianknot.rand.o...5001/desh_postgres
✓	solr	name: solr	1 / 1	3 hours	gordianknot.rand.org:5001/desh_solr
✓	spark-master-controller	component: spark-master	1 / 1	3 hours	gordianknot.rand.or...ners/spark:1.5.2_v1
✓	spark-worker-controller	component: spark-worker	2 / 2	3 hours	gordianknot.rand.or...ners/spark:1.5.2_v1

Pods

	Name	Status	Restarts	Age	Cluster IP	CPU (cores)	Memory (bytes)
✓	activemq-a0pd7	Running	0	3 hours	172.18.5.3	0.001	203.383 Mi
✓	desh-worker-controller-1o2j8	Running	0	9 minutes	172.18.3.3	0.028	1.457 Mi
✓	desh-worker-controller-etxxe	Running	0	9 minutes	172.18.1.5	-	75.629 Mi
✓	kafka-i4d8k	Running	0	3 hours	172.18.4.5	0.002	510.137 Mi
✓	postgres-6uyun	Running	0	3 hours	172.18.2.4	0.004	94.012 Mi
✓	solr-t65ci	Running	0	3 hours	172.18.5.4	0	222.629 Mi
✓	spark-master-controller-96r1d	Running	0	3 hours	172.18.0.5	0.004	991.223 Mi
✓	spark-worker-controller-4kdiy	Running	0	3 hours	172.18.1.4	0.003	291.336 Mi
✓	spark-worker-controller-a5lpy	Running	0	3 hours	172.18.4.6	0.002	261.645 Mi

Kubernetes



- Kubernetes is a open source platform for automating scaling and operations of containerized applications on clusters

- It enables applications to be scaled “on the fly”

RAND