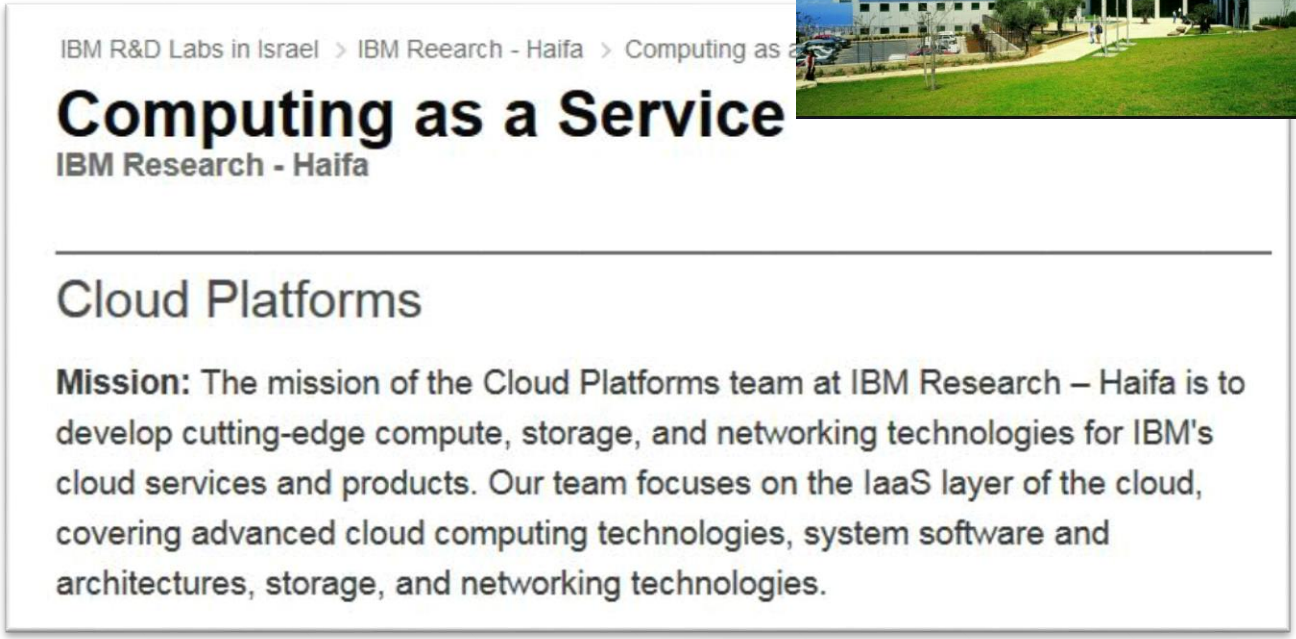
Auditing, Security and Data Analytics for Cloud Object Stores

Shelly Garion and Yaron Weinsberg
IBM Research Haifa

© 2015 IBM Corporation

Who are we?

- IBM Research Haifa
- Computing as a Service
- Cloud Platforms Department
- Cloud Security and Analytics Group



IBM R&D Labs in Israel > IBM Reearch - Haifa > Computing as a

Computing as a Service

IBM Research - Haifa

Cloud Platforms

Mission: The mission of the Cloud Platforms team at IBM Research – Haifa is to develop cutting-edge compute, storage, and networking technologies for IBM's cloud services and products. Our team focuses on the IaaS layer of the cloud, covering advanced cloud computing technologies, system software and architectures, storage, and networking technologies.

SOFTLAYER an IBM Company

IBM Cloud

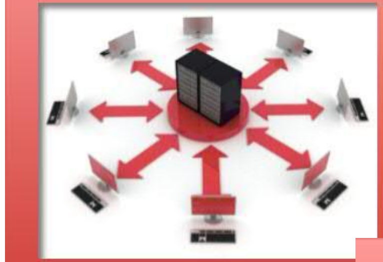
Come and join us!

https://www.research.ibm.com/haifa/dept/stt/ssp.html

© 2015 IBM Corporation

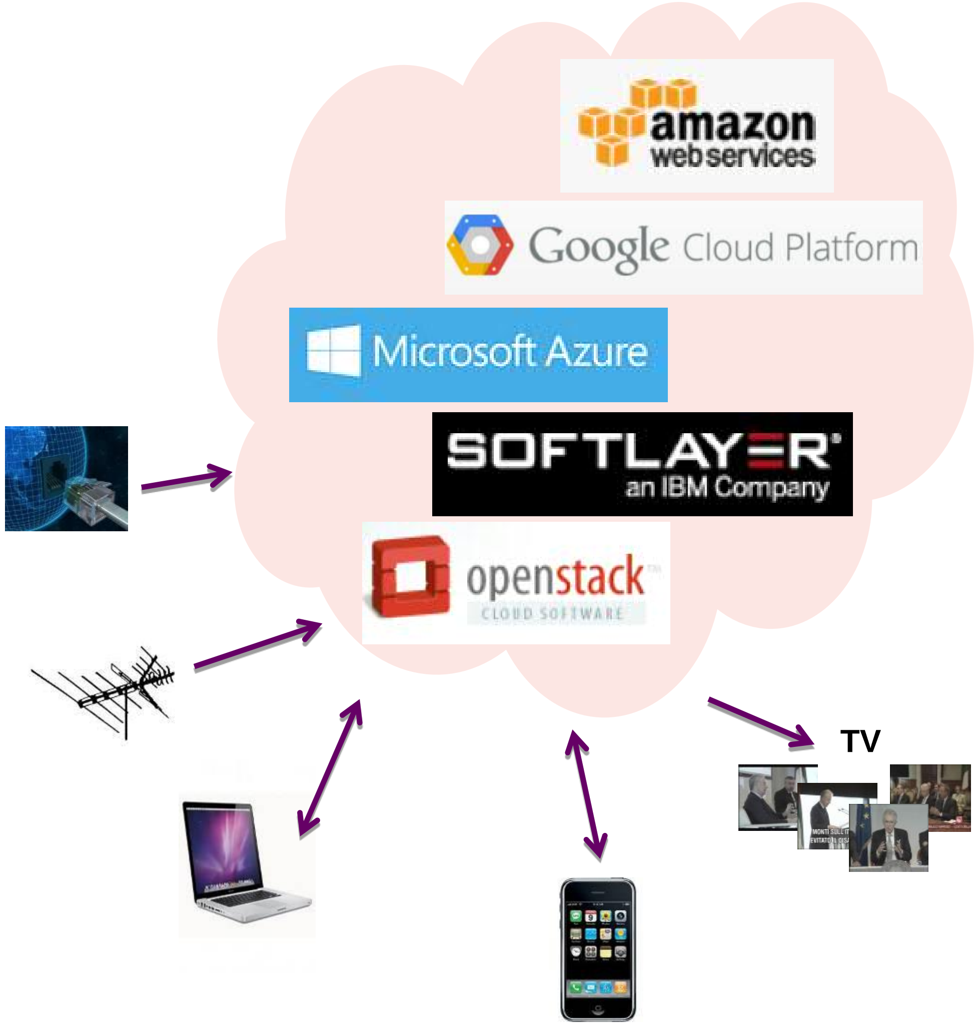
Cloud Storage

70's – 80's
Mainframe Era



90's-2000's
Client Server Era

2010-beyond
Cloud Era



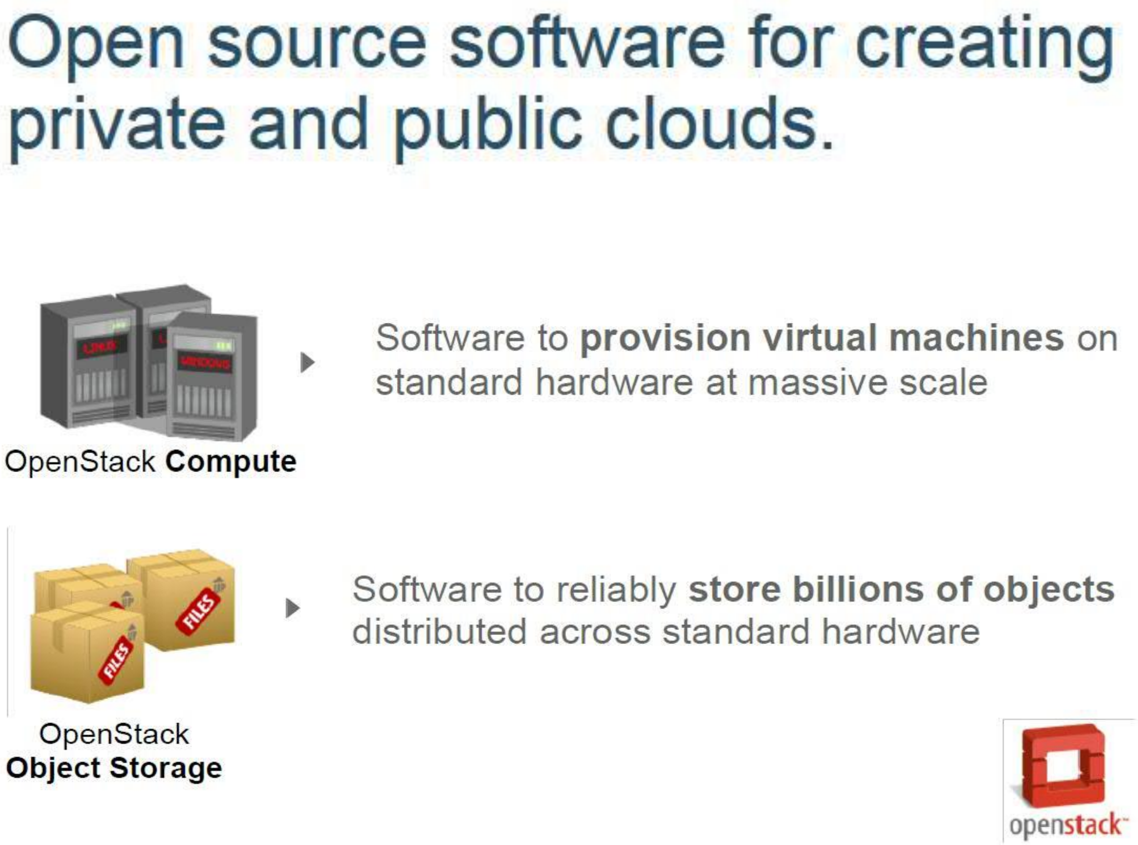
© 2015 IBM Corporation

Cloud Object Store

- Naturally suited for cloud
  - Un-structured and semi-structured data
  - Scalability
  - Accessed from everywhere anytime
  - Media, telco, healthcare, financial, government, backup,…
- Storage architecture that manages data as objects
  (as opposed to other architectures as file systems)
- An object encapsulates data and metadata
- Object data written once and not modified
  - Pictures, movies, tweets, blog-posts, etc.
- Accessed through RESTful HTTP
  - PUT, GET, DELETE…
- Runs on clusters of storage rich servers
- Provides high capacity at reduced costs

© 2015 IBM Corporation

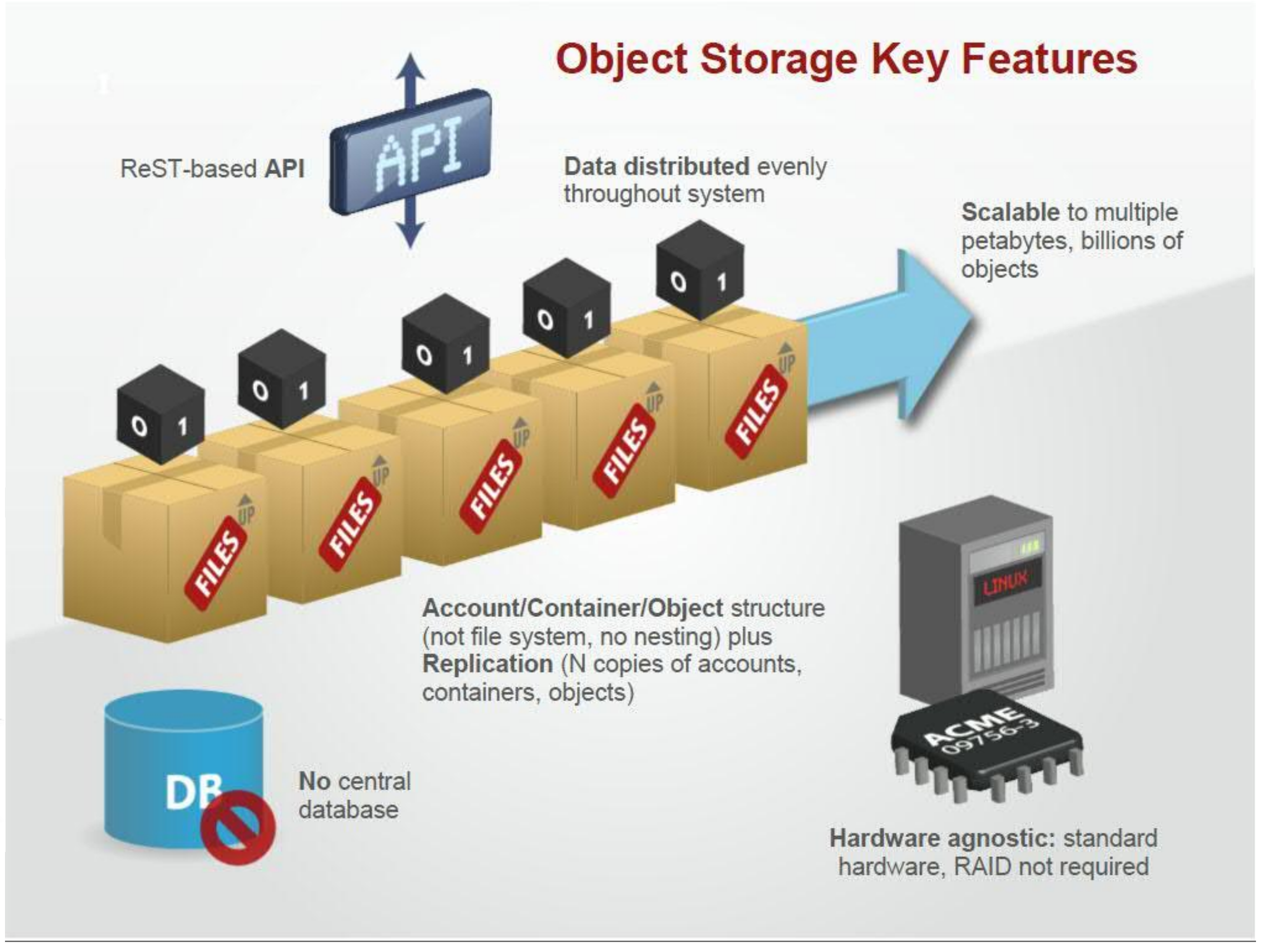
OpenStack Swift



Open source software for creating private and public clouds.

OpenStack **Compute** – Software to **provision virtual machines** on standard hardware at massive scale

OpenStack **Object Storage** – Software to reliably **store billions of objects** distributed across standard hardware



Object Storage Key Features

ReST-based **API**

Data distributed evenly throughout system

Scalable to multiple petabytes, billions of objects

Account/Container/Object structure (not file system, no nesting) plus **Replication** (N copies of accounts, containers, objects)

No central database

Hardware agnostic: standard hardware, RAID not required

Source: “OpenStack Tutorial”, *IEEE CloudCom 2010* Brett Piatt

http://salsahpc.indiana.edu/CloudCom2010/slides/PDF/tutorials/OpenStackTutorialIEEECloudCom.pdf

© 2015 IBM Corporation

Cloud Object Store – Security and Privacy Challenges

- Need to trust the cloud provider
- Multi-tenancy
- BigData
  - Scalability
  - Un-structured and semi-structured data
- Private data may be stored un-intentional
- User data privacy is required by regulations
  - HIPPA – medical data
  - PCI – financial data

10 Worst Cloud Security Threats Of 2015

Dan Kobialka | *Talkin Cloud*

1. Data Breaches
2. Data Loss
3. DDoS attacks
4. Account Hijacking
5. Insider Attacks
6. Malware
7. Viruses
8. Phishing attacks
9. Bring-Your-Own-Device
10. Insufficient Due Diligence

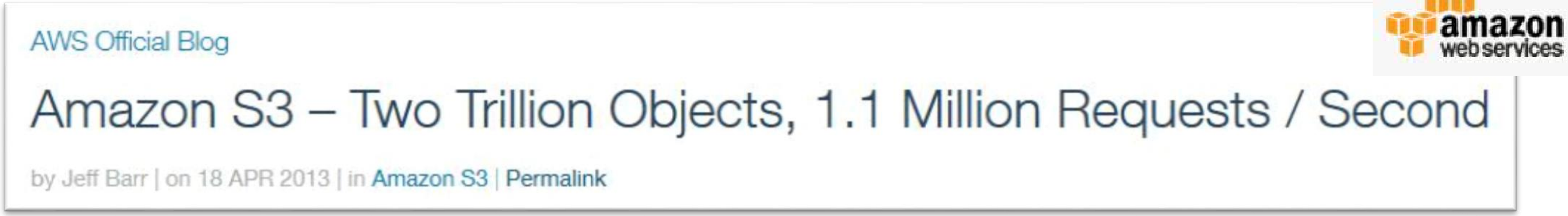
InformationWeek CONNECTING THE BUSINESS TECHNOLOGY COMMUNITY

9 Worst Cloud Security Threats

Leading cloud security group lists the "Notorious Nine" top threats to cloud computing in 2013; most are already known but defy 100% solution.

© 2015 IBM Corporation

Audit Trail of Cloud Object Store – BigData Challenge



AWS Official Blog

Amazon S3 – Two Trillion Objects, 1.1 Million Requests / Second

by Jeff Barr | on 18 APR 2013 | in Amazon S3 | Permalink

- Amazon's audit trail for 2013 (estimation):
  - **$35 \cdot 10^{12}$** log lines
  - **10 PB**

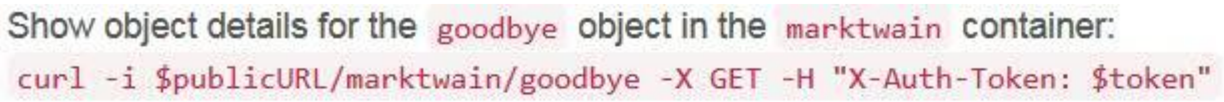
- Audit trail records all accesses:
  - **Who?** (IP address)
  - **What?** (PUT, GET, DELETE, …)
  - **Which?** (Account/Container/Object)
  - **When?** (Time)
  - **More** (failed attempts, recourse usage, latency…)

- Semi-structured data (noise, errors, missing fields, broken lines…)

© 2015 IBM Corporation

Audit Trail and Swift Logs

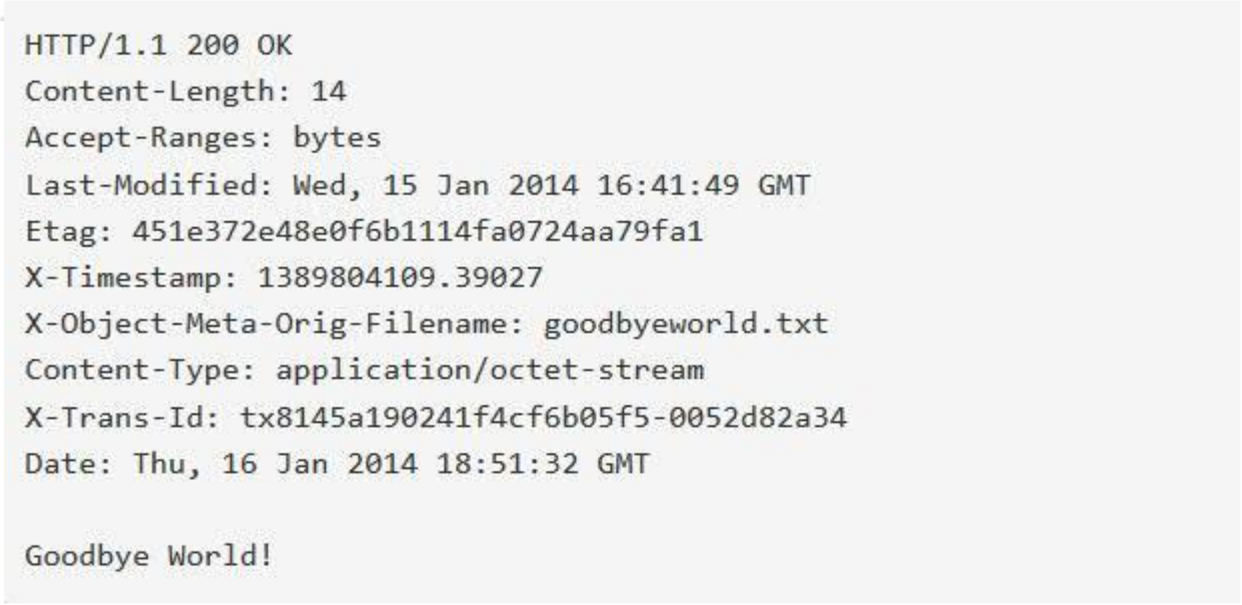
- Request:



Show object details for the `goodbye` object in the `marktwain` container:

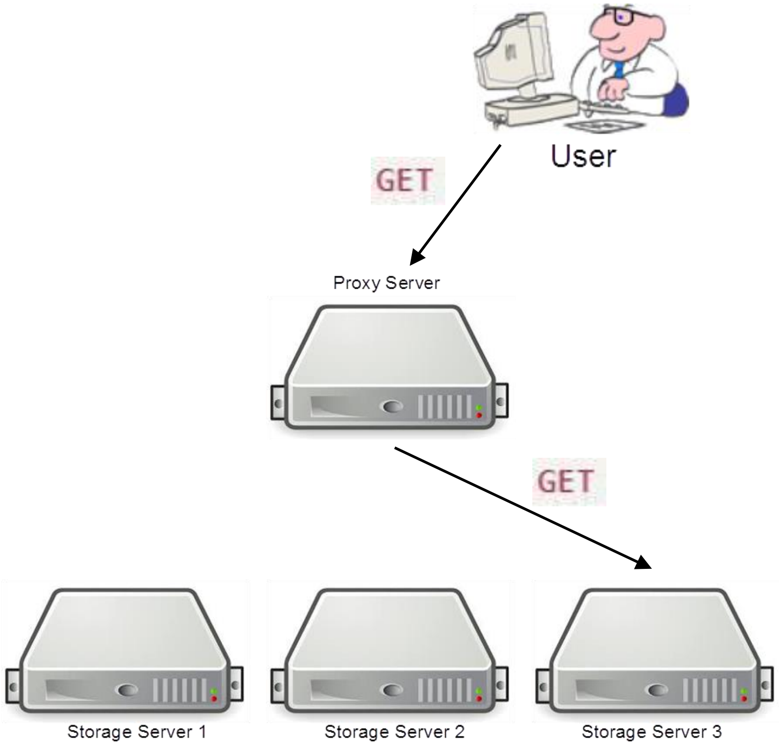
```
curl -i $publicURL/marktwain/goodbye -X GET -H "X-Auth-Token: $token"
```

- Response:



```
HTTP/1.1 200 OK
Content-Length: 14
Accept-Ranges: bytes
Last-Modified: Wed, 15 Jan 2014 16:41:49 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
X-Timestamp: 1389804109.39027
X-Object-Meta-Orig-Filename: goodbyeworld.txt
Content-Type: application/octet-stream
X-Trans-Id: tx8145a190241f4cf6b05f5-0052d82a34
Date: Thu, 16 Jan 2014 18:51:32 GMT

Goodbye World!
```



- Swift proxy log line:

```
Jan 16 18:51:32 copper proxy-server 208.80.152.165 127.0.0.1 6/Jan/2014/18/51/32 GET
/v1/my_account/marktwain/goodbye HTTP/1.0 200 - - - - 14 - tx8145a190241f4cf6b05f5 - 0.0020
```

http://developer.openstack.org/api-ref-objectstorage-v1.html
http://docs.openstack.org/developer/swift/logs.html

© 2015 IBM Corporation

Analysis of Swift Logs – Motivation and Goals

- Swift proxy log line – Example:

```
Jan 16 18:51:32 copper proxy-server 208.80.152.165 127.0.0.1 6/Jan/2014/18/51/32 GET
/v1/my_account/marktwain/goodbye HTTP/1.0 200 - - - - 14 - tx8145a190241f4cf6b05f5 - 0.0014
```

- Information in the logs:
  - Date and time
  - Client IP address
  - Request method (GET/PUT/DELETE…)
  - Request path: account/container/object
  - HTTP status code
  - Bytes received / Bytes sent
  - Request time (latency)

- Log analysis is useful for:

➢ Capacity planning

➢ Performance insight

➢ Predictive failure analysis

➢ System design

➢ Periodic & unusual behavior

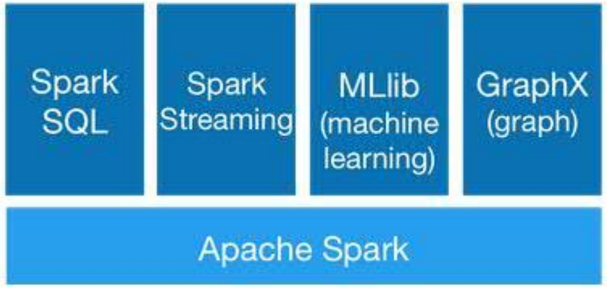
➢ Security & anomaly detection

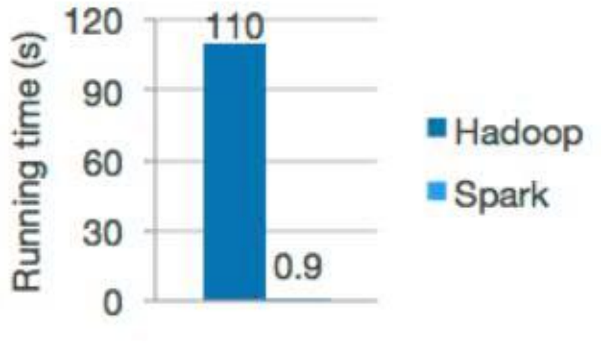
➢ ...

- Semi-structured data ⇒ cleaning and parsing

Apache Spark

- **Apache Spark™** is a fast and general open-source engine for large-scale data processing
- Spark is capable to run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk
- Includes the following libraries:
  - SPARK SQL
  - SPARK Streaming
  - MLlib (Machine Learning)
  - GraphX (graph processing)





Logistic regression in Hadoop and Spark

- Spark can run on Apache Mesos, Hadoop 2's YARN cluster manager, standalone or in the cloud, and can read any existing Hadoop data, and data from HDFS or swift
- Written in **Scala** language (a ‘Java’ like, executed in Java VM)
- Apache Spark is built by a wide set of developers from over 50 companies
  Since the project started in 2009, more than 400 developers have contributed to Spark

https://spark.apache.org/

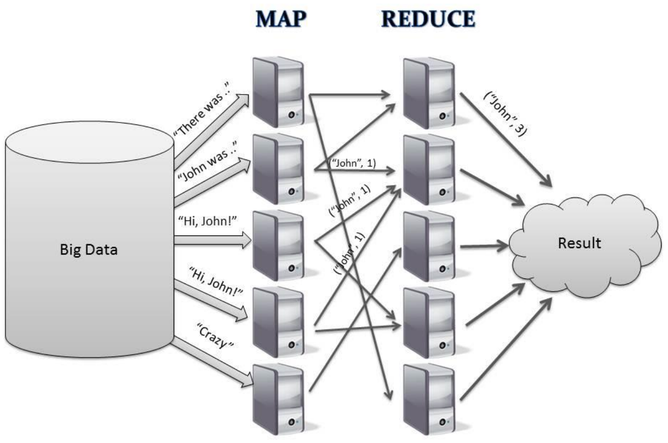
© 2015 IBM Corporation

Analysis of Swift Proxy Logs using Spark Map/Reduce
Example: Archiving

- **Goal:** How many objects can be archived? What should be the archive size?

- **Q:** Percentage of objects that have not been used in the past three months?

- **Challenge:** large number of objects (billions, trillions,…)
  - Parsing and cleaning
  - Map/Reduce
  - Sampling

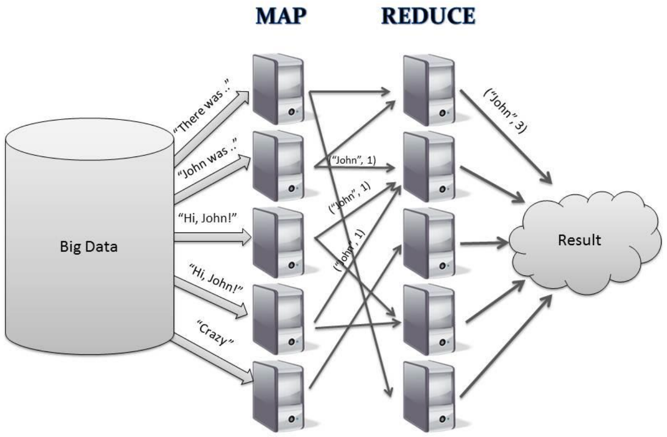


- **Algorithm:**
  - Map: `LogLine => (ObjectName, DayOfYear, ObjectSize)`
  - Filter: `=> Hashed ObjectName starting with "00"`
  - Reduce: `=> (Object, (NumberOfLogLines, MinDayOfYear, MaxDayOfYear, ObjectSize))`

© 2015 IBM Corporation

Analysis of Swift Proxy Logs using Spark Map/Reduce
Example: Archiving

- **Goal:** How many objects can be archived? What should be the archive size?
- **Q:** Percentage of objects that have not been used in the past three months?
- Spark Map/Reduce:



- Map:

```
def TakeAllObjects (line: Array[String]) = {
            val Month = line(0)
            val Day = line(1)
            val Object = line(9)
            val splitURI = line(9).split("/")
            val PUTsize = StrtoInt(line(15))
            val Objectchar = if (splitURI.length > 4) HashFunc(splitURI(4)).substring(0,2) else "*"
            val DayofYear = 31*(Month.toInt-1)+(Day.toInt-1)
            ( (Objectchar, Object), (1L, DayofYear, DayofYear, PUTsize) )   }
```

- Reduce:

```
def ReduceAllObjects ( a: (Long,Int,Int,Long), b: (Long,Int,Int,Long) ) = {
            val numobj = ( a._1+b._1 )
            val maxobj = ( if (a._2 > b._2) a._2 else b._2 )
            val minobj = ( if (a._3 < b._3) a._3 else b._3 )
            val sizeobj = ( if (a._4 > b._4) a._4 else b._4 )
            (numobj, maxobj, minobj, sizeobj) }
```

- Process:

```
val textFile = sc.textFile("hdfs:///projects/Data/2014*.gz")
val NewFile = textFile.map(_.split(" ")).filter(_.length > 19)
val AllObjects = NewFile.map(TakeAllObjects).filter(line => (line._1)._1 == “00").map(line => ( (line._1)._2, line._2) )
val DistinctObjects = AllObjects.reduceByKey(ReduceAllObjects)
val ArchivedObjects = DistinctObjects.filter(line => (line._2)._2<365-90)
```

Analysis of Swift Proxy Logs using Spark MLLib

Example: Machine learning clustering algorithm

- **Goal:** Cluster analysis of client access to an account by time of day
- **Challenges:**
  - Large number of clients
  - Integration of iterative Map/Reduce and Machine Learning clustering
- **Algorithm:**

```
– Map:          LogLine  => (Account, ClientIP, HourOfDay)
– Filter:                => Account is MyAccount
– ReduceByKey:           => (ClientIP, (HourOfDay, #Accesses))
– GroupByKey:            => (ClientIP, [Distribution of #Accesses By HourOfDay])
– Clustering:            => K-Means Clustering
```

© 2015 IBM Corporation

Analysis of Swift Proxy Logs using Spark MLLib
Example: Machine learning clustering algorithm

- **Goal:** Cluster analysis of client access to an account by time of day

- Spark iterative Map/Reduce and Machine Learning:
- Map:

```
def TakeClients (line: Array[String]) = {
          val Time = line(2).split(“:")
          val Hour = Time(0)
          val ClientIP = line(5)
          val URI = line(9)
          val Account = URI.split("/")(2)
          ( (Account, ClientIP), Hour)   }
```

- Process & Reduce:

```
val textFile = sc.textFile("hdfs:///projects/Data/2014*.gz")
val NewFile = textFile.map(_.split(" ")).filter(_.length > 19)
val AllClients = NewFile.map(TakeClients).filter(line => (line._1)._1 == “MyAccount”). map(line => (((line._1)._2,line._2),1L))
val ReduceByTime = AllClients.reduceByKey((a,b) => a+b).map(line => ((line._1)._1, ((line._1)._2,line._2))).groupByKey()
val ClientTimeVector = ReduceByTime.map(LinetoVect)
```

- K-means clustering:

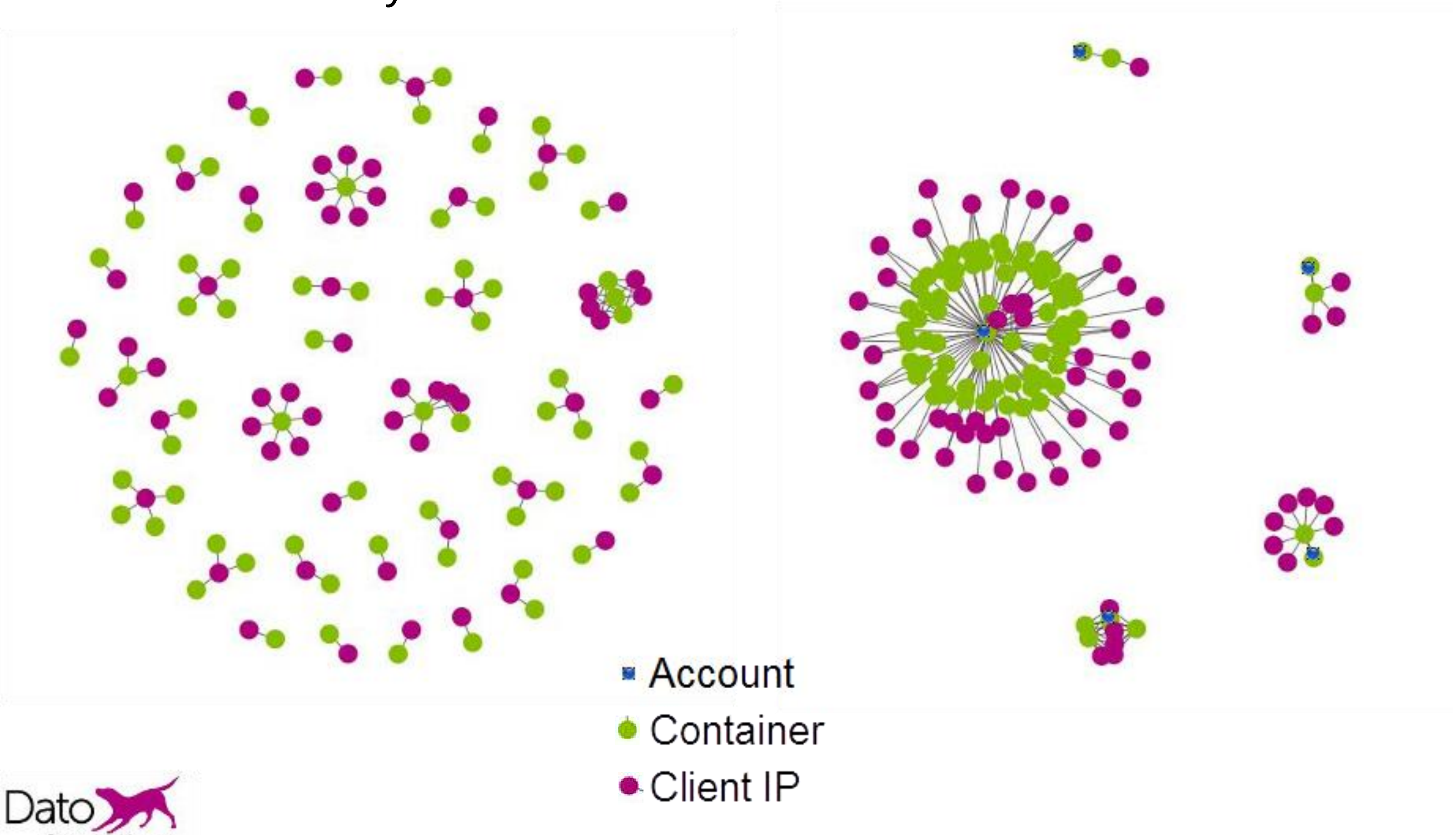
```
val iterationCount = 100
val clusterCount = 5
val model = KMeans.train(parsedData, clusterCount, iterationCount)
val clusterCenters = model.clusterCenters map(_.toArray)
val cost = model.computeCost(parsedData)
val clientsByGoup = ClientTimeVector.groupBy{rdd => model.predict(Vectors.dense(rdd))}.collect()
val clustersize = ClientsByGroup.map(item => (item._1, item._2.toSet.size) )
```

Analysis of Swift Proxy Logs using Spark & Dato

Example: Communities of Clients-Containers

Dato
Create Intelligence™

Goal: Narrow the data for better analytics



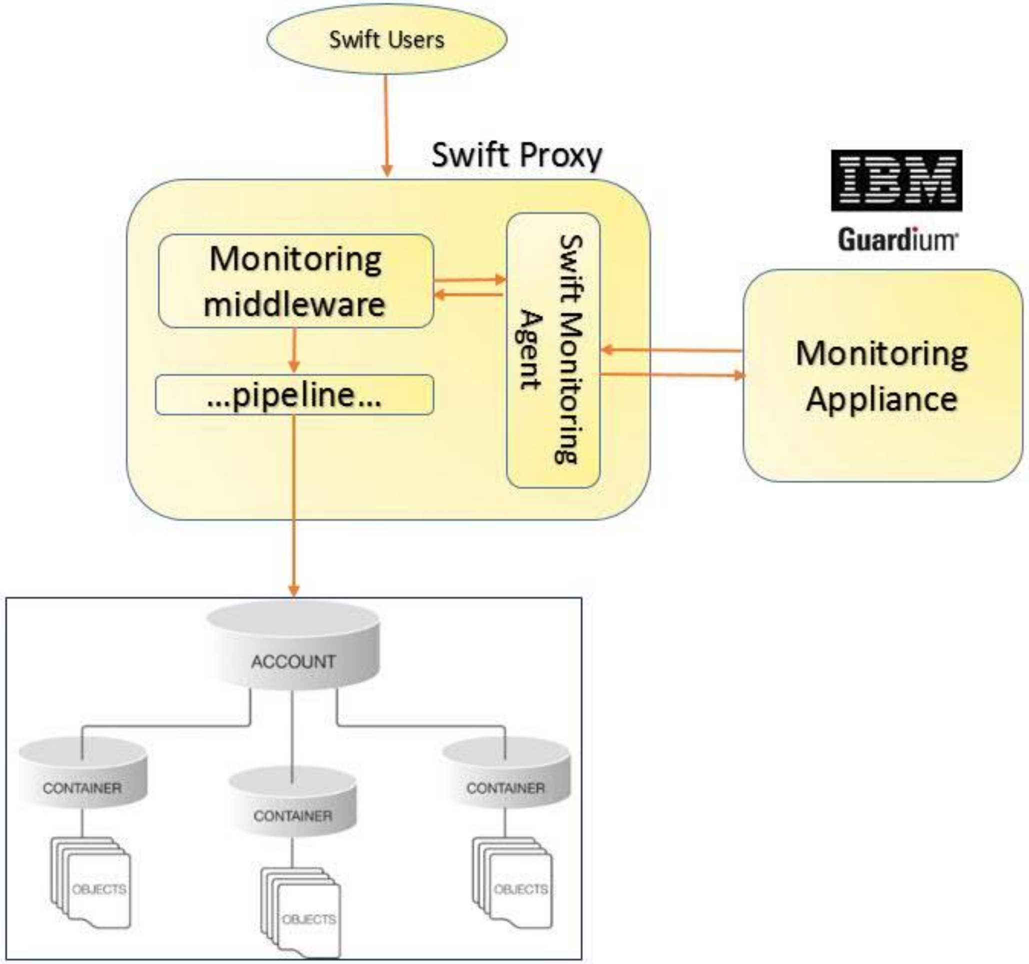
https://dato.com/ Dato Create Intelligence™

Activity Monitoring for OpenStack Swift

- **Goals**
  - Complete audit-trail of data access to Swift
  - Activity Monitoring
  - Compliance reports
  - Define policies and enforce them
  - Control data access

- Solutions for real-time database activity monitoring and protection (e.g. IBM InfoSphere Guardium)
  - Audit-trail for database access
  - Monitors database transactions and responds in real-time access policy violations
- Extension to OpenStack Swift

  Jointly with Guardium we have developed a POC for Guardium and Swift integration

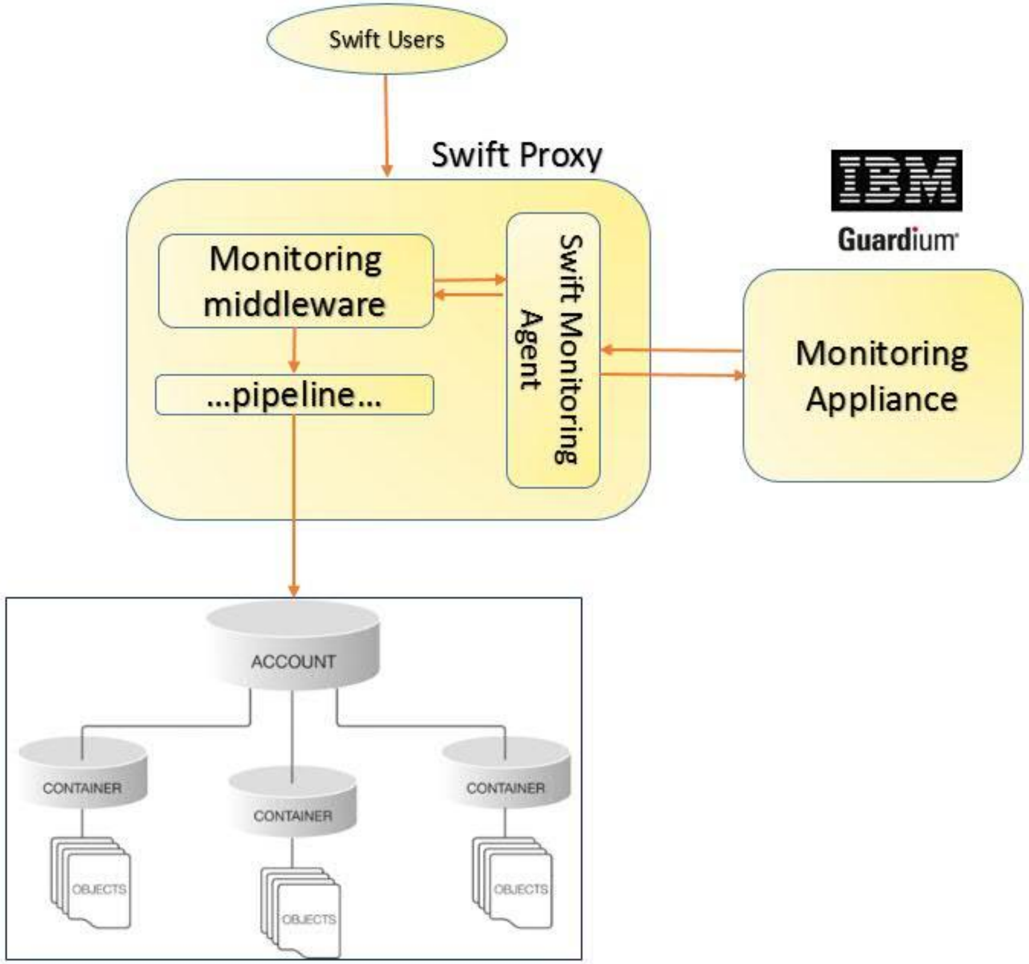


© 2015 IBM Corporation

Activity Monitoring for OpenStack Swift

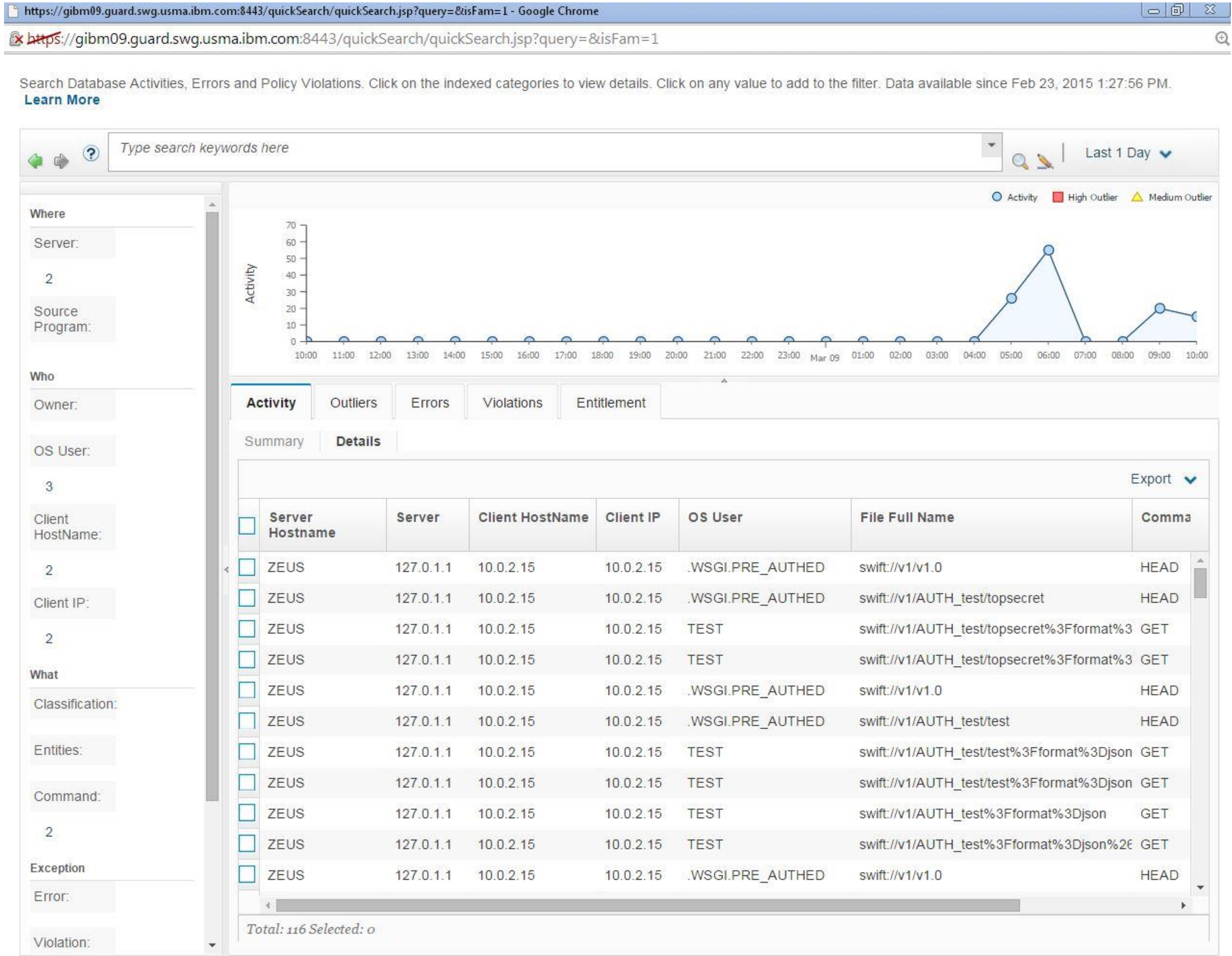
- **Challenge – BigData**
  - Scalability
  - Real-time
  - Which data is sensitive?

- **Solution – Selective monitoring**
  Use BigData analytics tools (such as Spark) to…
  - Identify sensitive data
  - Aggregation
  - Sampling
  - Machine Learning: clustering, communities
  - …



© 2015 IBM Corporation

Swift Report in Guardium (POC)



https://gibm09.guard.swg.usma.ibm.com:8443/quickSearch/quickSearch.jsp?query=&isFam=1 - Google Chrome

~~https~~://gibm09.guard.swg.usma.ibm.com:8443/quickSearch/quickSearch.jsp?query=&isFam=1

Search Database Activities, Errors and Policy Violations. Click on the indexed categories to view details. Click on any value to add to the filter. Data available since Feb 23, 2015 1:27:56 PM.
Learn More

Type search keywords here

Last 1 Day

Where

Server:
2

Source Program:

Who

Owner:

OS User:
3

Client HostName:
2

Client IP:
2

What

Classification:

Entities:

Command:
2

Exception

Error:

Violation:

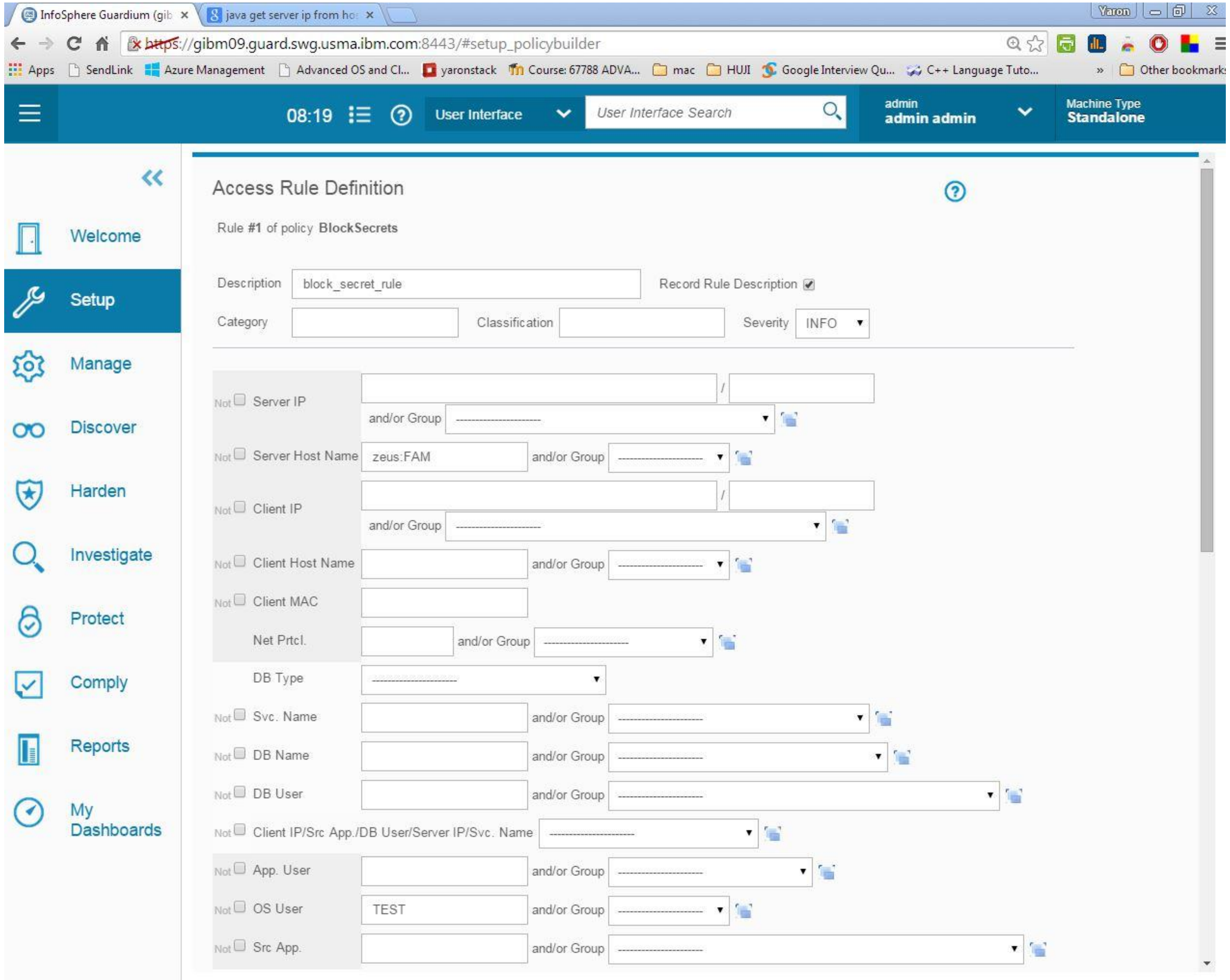
□	Server Hostname	Server	Client HostName	Client IP	OS User	File Full Name	Comma
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	.WSGI.PRE_AUTHED	swift://v1/v1.0	HEAD
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	.WSGI.PRE_AUTHED	swift://v1/AUTH_test/topsecret	HEAD
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	TEST	swift://v1/AUTH_test/topsecret%3Fformat%3	GET
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	TEST	swift://v1/AUTH_test/topsecret%3Fformat%3	GET
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	.WSGI.PRE_AUTHED	swift://v1/v1.0	HEAD
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	.WSGI.PRE_AUTHED	swift://v1/AUTH_test/test	HEAD
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	TEST	swift://v1/AUTH_test/test%3Fformat%3Djson	GET
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	TEST	swift://v1/AUTH_test/test%3Fformat%3Djson	GET
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	TEST	swift://v1/AUTH_test%3Fformat%3Djson	GET
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	TEST	swift://v1/AUTH_test%3Fformat%3Djson%2(	GET
□	ZEUS	127.0.1.1	10.0.2.15	10.0.2.15	.WSGI.PRE_AUTHED	swift://v1/v1.0	HEAD

Total: 116 Selected: 0

Swift access is logged (Guardium can control logging granularity)

© 2015 IBM Corporation

Swift Policy Definition (POC)



Access Rule Definition

Rule #1 of policy **BlockSecrets**

Description: block_secret_rule Record Rule Description ☑

Category: Classification: Severity: INFO

Not □ Server IP / and/or Group ------------------

Not □ Server Host Name zeus:FAM and/or Group ------------------

Not □ Client IP / and/or Group ------------------

Not □ Client Host Name and/or Group ------------------

Not □ Client MAC

Net Prtcl. and/or Group ------------------

DB Type ------------------

Not □ Svc. Name and/or Group ------------------

Not □ DB Name and/or Group ------------------

Not □ DB User and/or Group ------------------

Not □ Client IP/Src App./DB User/Server IP/Svc. Name ------------------

Not □ App. User and/or Group ------------------

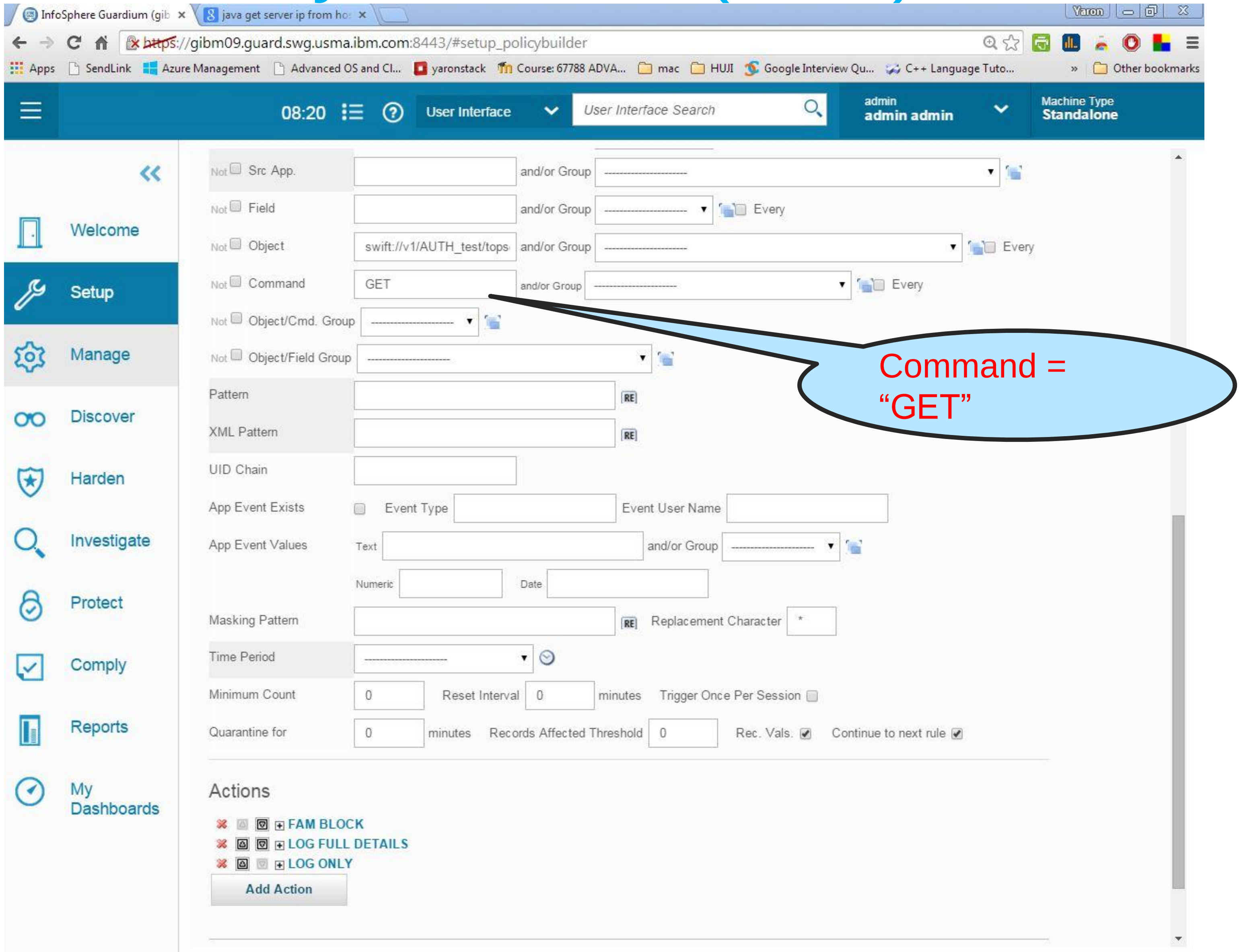
Not □ OS User TEST and/or Group ------------------

Not □ Src App. and/or Group ------------------

Policy is defined via standard Guardium UI

© 2015 IBM Corporation

Swift Policy Definition (POC)



https://gibm09.guard.swg.usma.ibm.com:8443/#setup_policybuilder

Not □ Src App.		and/or Group	------------------	
Not □ Field		and/or Group	------------------	□ Every
Not □ Object	swift://v1/AUTH_test/tops	and/or Group	------------------	□ Every
Not □ Command	GET	and/or Group	------------------	□ Every
Not □ Object/Cmd. Group	------------------			
Not □ Object/Field Group	------------------			
Pattern				
XML Pattern				
UID Chain				
App Event Exists	□	Event Type		Event User Name
App Event Values	Text	and/or Group	------------------	
	Numeric	Date		
Masking Pattern		Replacement Character	*	
Time Period	------------------			
Minimum Count	0	Reset Interval 0 minutes	Trigger Once Per Session □	
Quarantine for	0 minutes	Records Affected Threshold 0	Rec. Vals. ☑	Continue to next rule ☑

Actions

- FAM BLOCK
- LOG FULL DETAILS
- LOG ONLY

Add Action

Command =
“GET”

© 2015 IBM Corporation