



BDM401

# Deep Dive: Amazon EMR

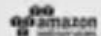
Best Practices & Design Patterns

Jonathan Fritz, Sr. Product Manager, Amazon EMR

Naveen Avalareddy, Sr. Principal Architect, Asurion

November 29, 2016

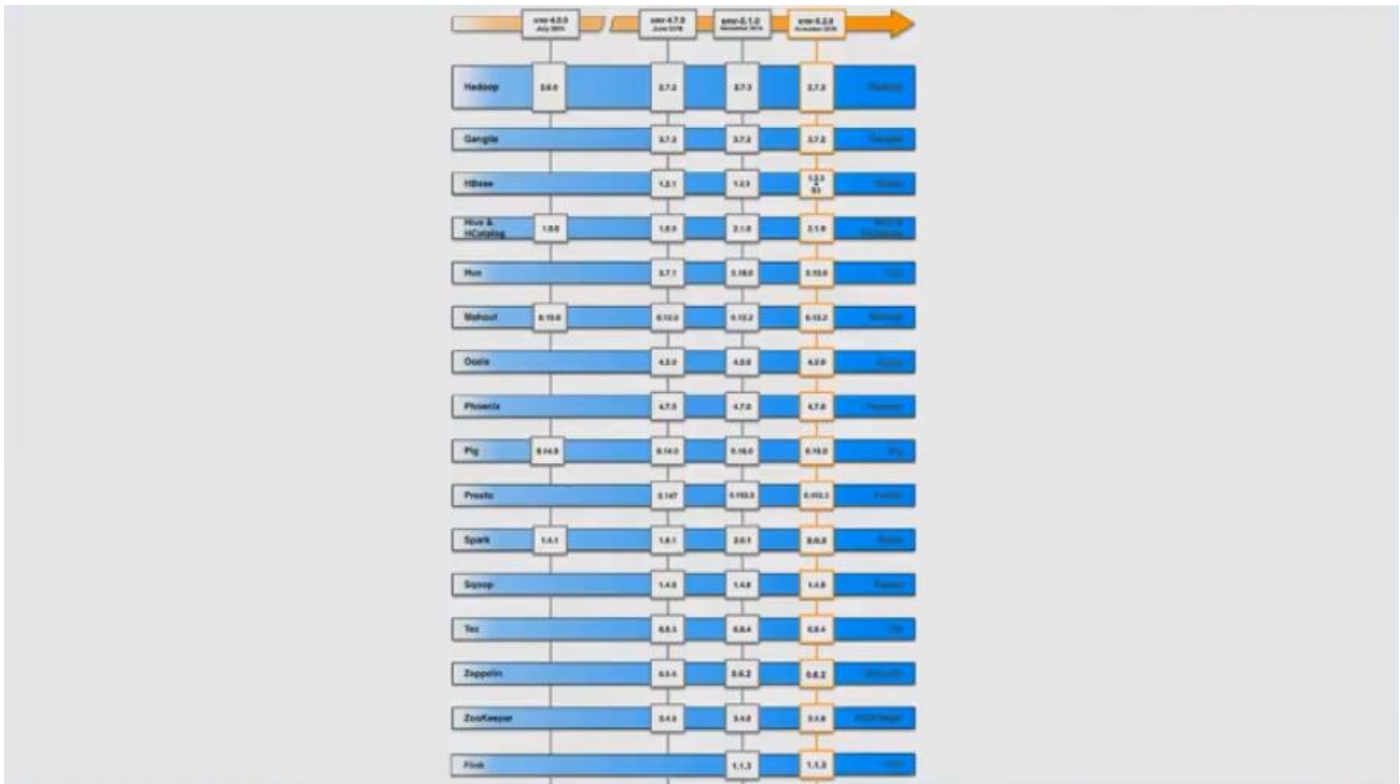
© 2016, Amazon Web Services, Inc., or its Affiliates. All rights reserved.



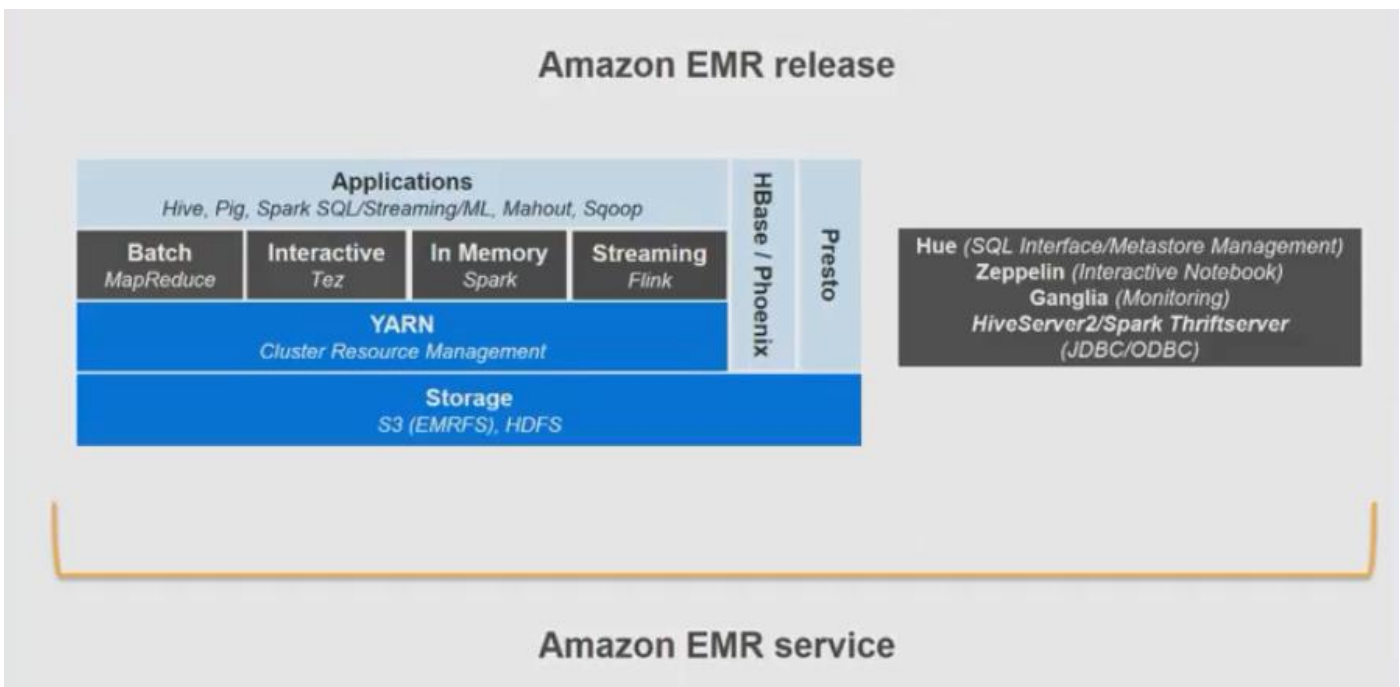
Amazon EMR is one of the largest Hadoop operators in the world. In this session, we introduce you to Amazon EMR design patterns such as using Amazon S3 instead of HDFS, taking advantage of both long and short-lived clusters, and other Amazon EMR architectural best practices. We talk about how to scale your cluster up or down dynamically and introduce you to ways you can fine-tune your cluster. We also share best practices to keep your Amazon EMR cluster cost-efficient. Finally, we dive into some of our recent launches to keep you current on our latest features. This session will feature Asurion, a provider of device protection and support services for over 280 million smartphones and other consumer electronics devices. Asurion will share how they architected their petabyte-scale data platform using Apache Hive, Apache Spark, and Presto on Amazon EMR.

## What to expect from the session

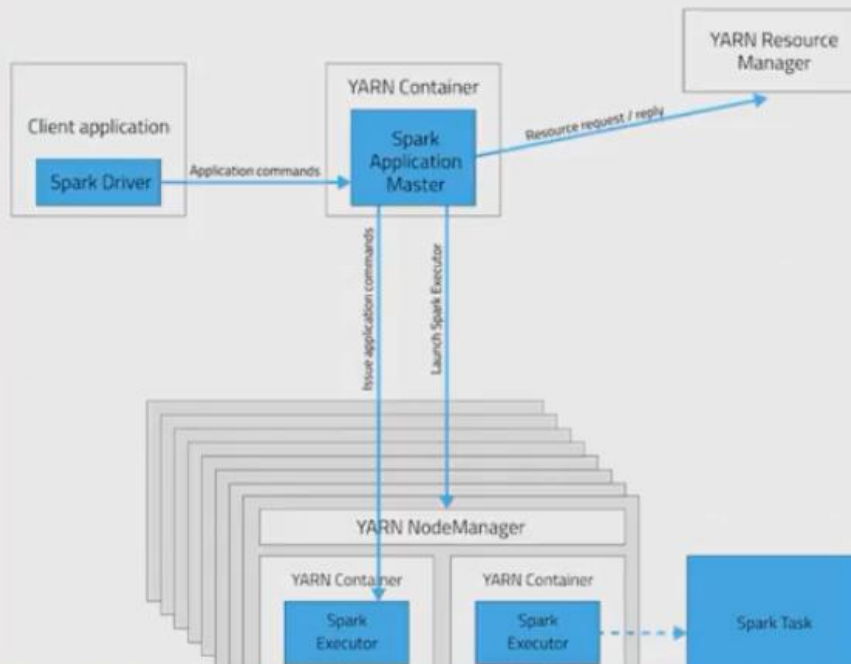
- Overview of Apache ecosystem on EMR
- Using EMR with S3 and other AWS services
- Securing your EMR application stack
- Lowering costs with Auto Scaling and Spot Instances
- Building a data lake with EMR at Asurion
- Q&A



This is a list of the EMR 4.0, EMR 5.2 applications and all the OSS projects that EMR now supports



# YARN overview

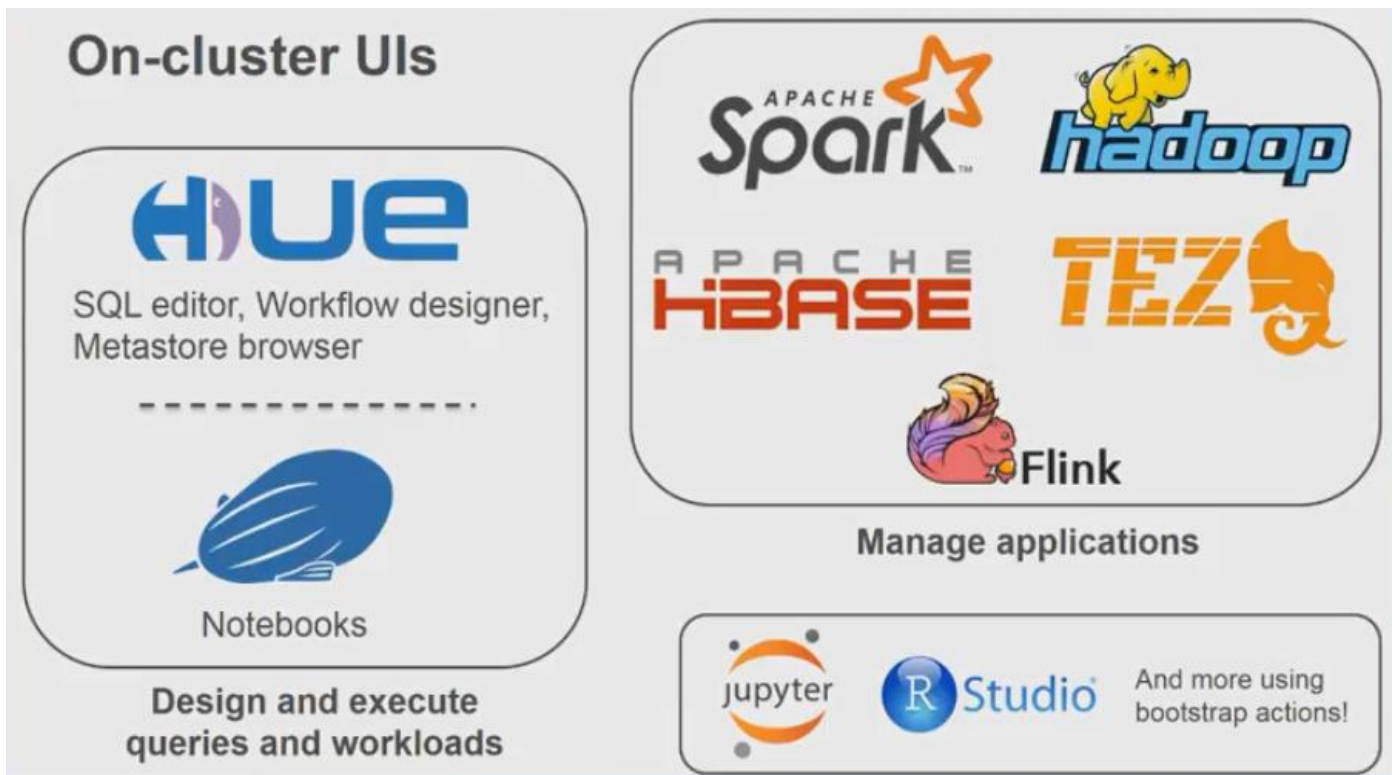


- Dynamically share and centrally configure the same pool of cluster resources across engines
- Schedulers for categorizing, isolating, and prioritizing workloads
- Spark dynamic allocation of executors
- Kerberos authentication

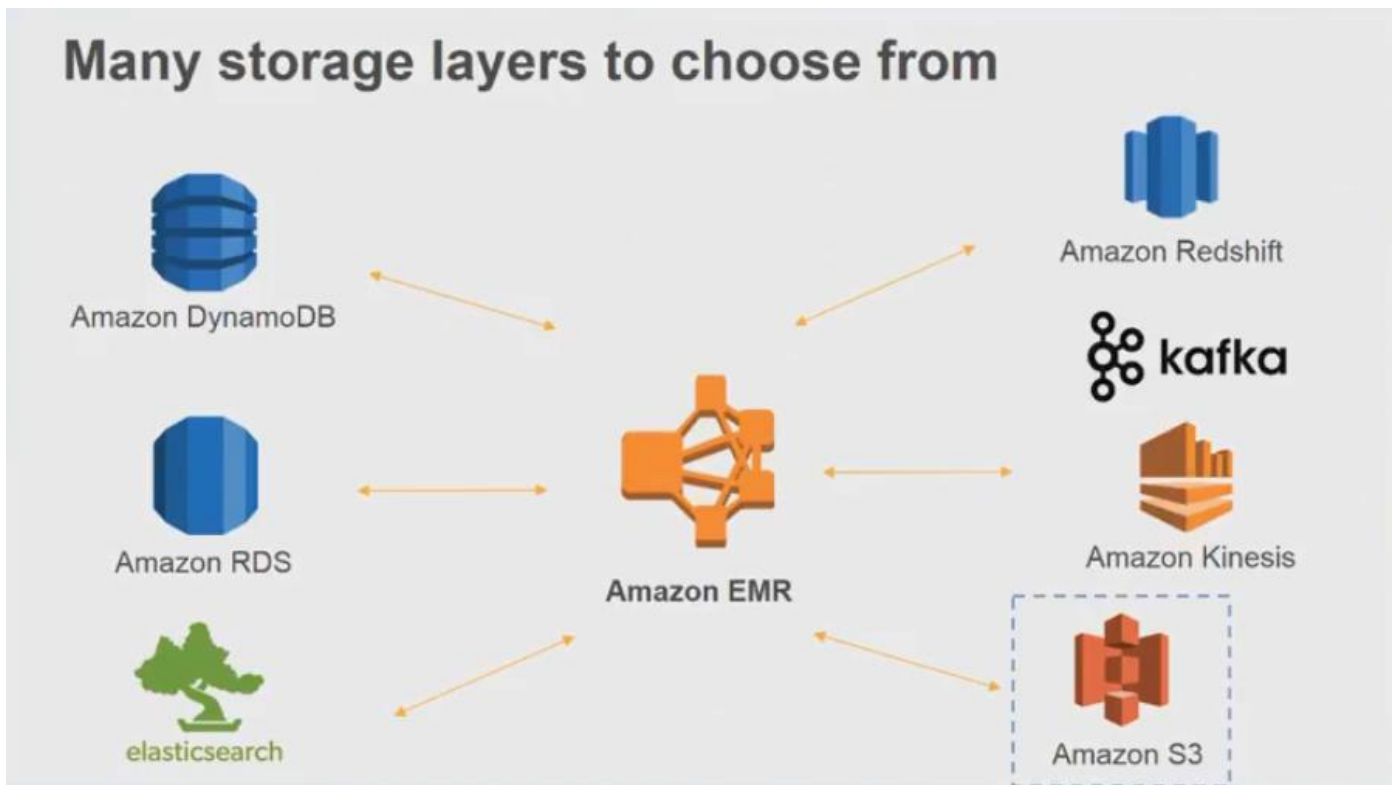
This is how YARN might run a Spark application

## YARN schedulers - CapacityScheduler

- Default scheduler specified in Amazon EMR
- Queues
  - Single queue is set by default
  - Can create additional queues for workloads based on multitenancy requirements
- Capacity guarantees
  - set minimal resources for each queue
  - Programmatically assign free resources to queues
- Adjust these settings using the `classification capacity-scheduler` in an EMR configuration object

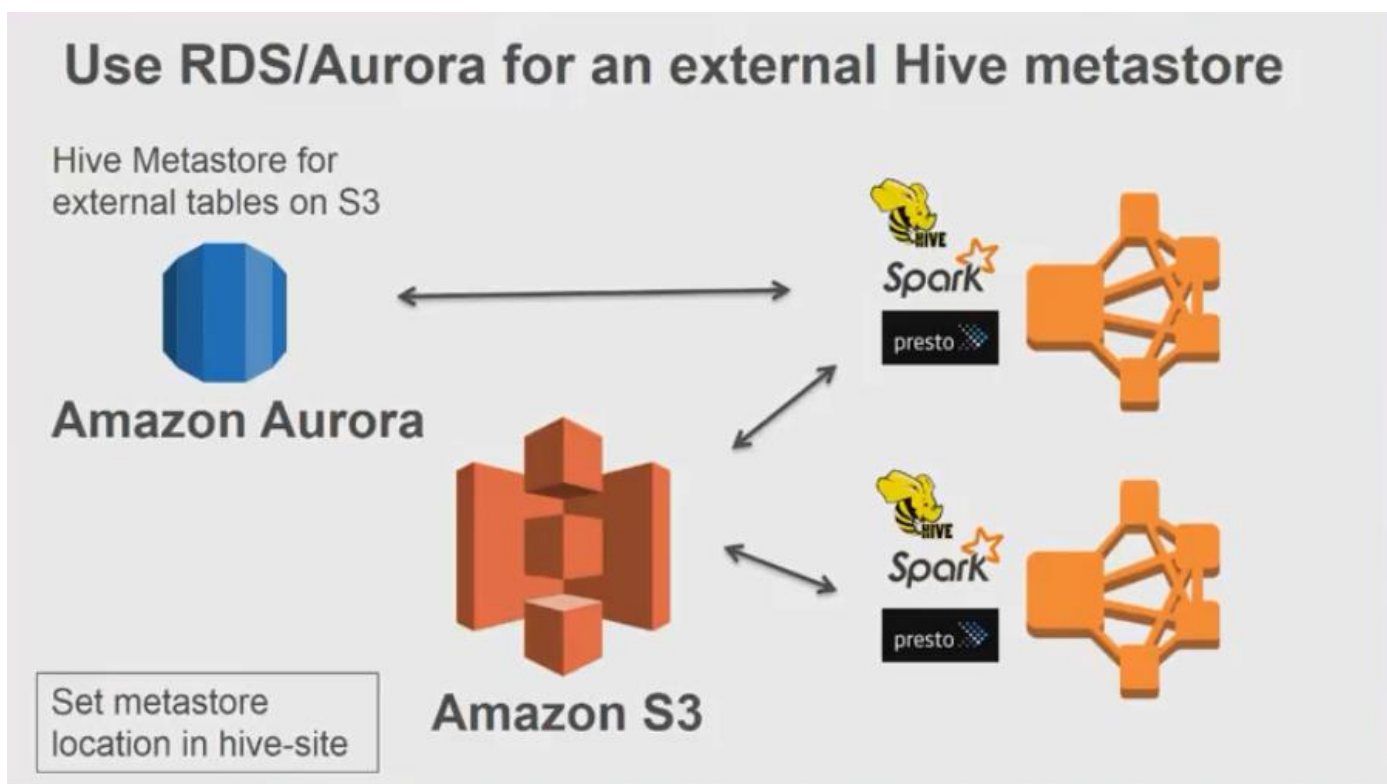
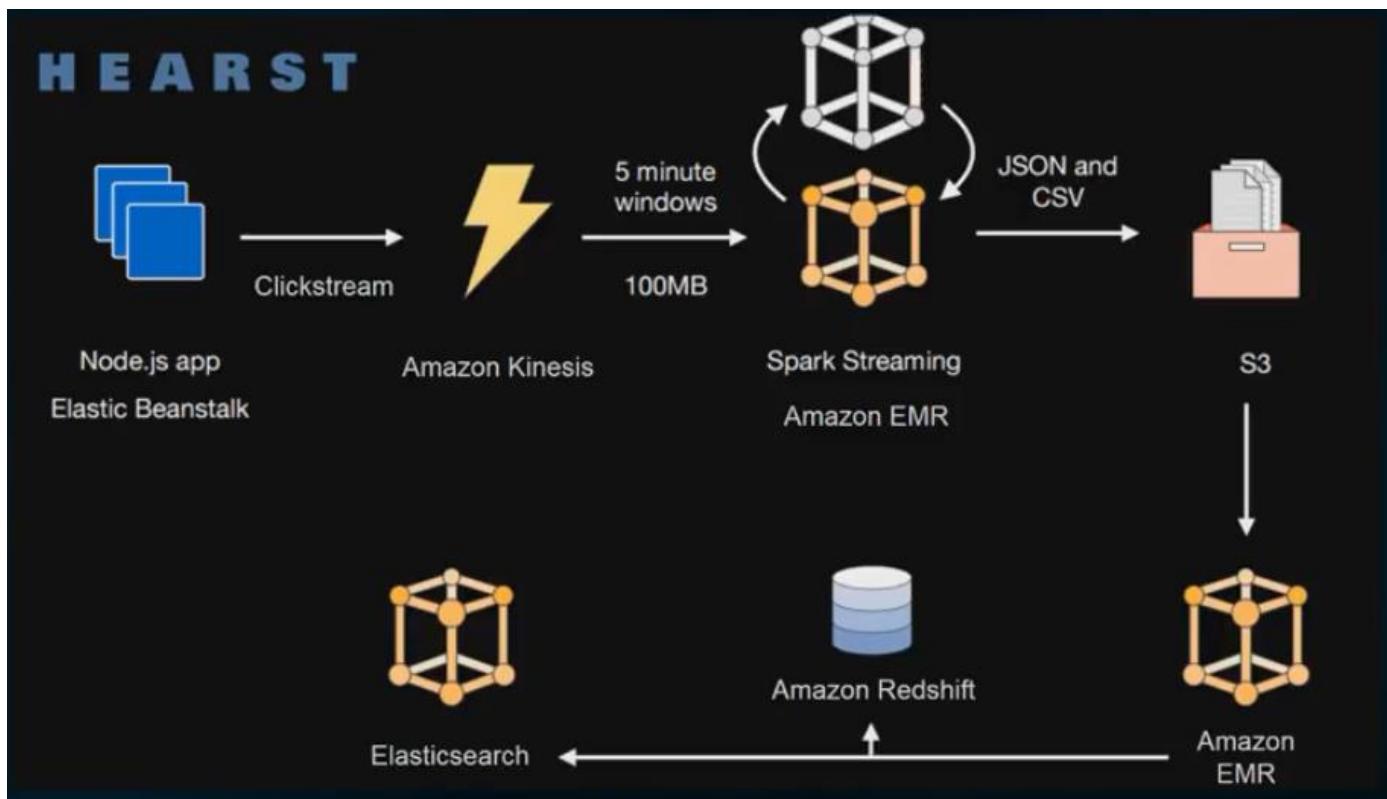


The on-cluster UIs are all available on the master node like Hue for browsing the Hive metastore, zeppelin for data notebooks, the Spark and Resoucre Manager UIs, Tez for DAG jobs, etc.



EMR can access data from a variety of sources and data formats for complex analytics pipelines.



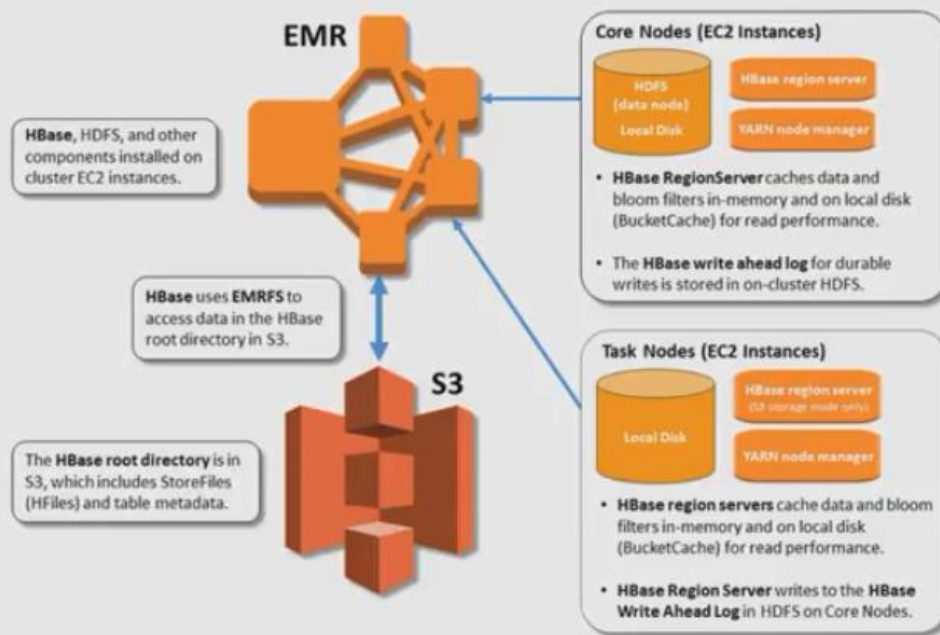


You can use Aurora or RDS to store your table information outside of any of your clusters, so that when your cluster comes up, you simply point it at the database and you get your tables back, you don't need to recover your data partitions every single time.

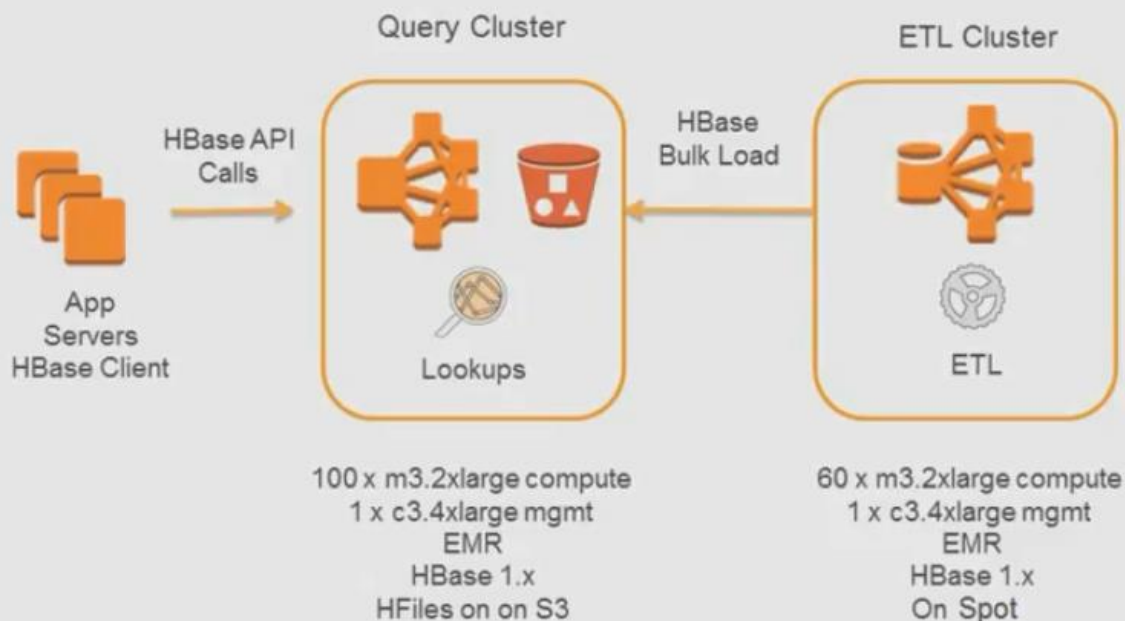
## S3 tips: Partitions, compression, and file formats

- Avoid key names in lexicographical order
- Improve throughput and S3 list performance
- Use hashing/random prefixes or reverse the date-time
- Compress data set to minimize bandwidth from S3 to EC2
  - Make sure you use splittable compression or have each file be the optimal size for parallelization on your cluster
- Columnar file formats like Parquet can give increased performance on reads

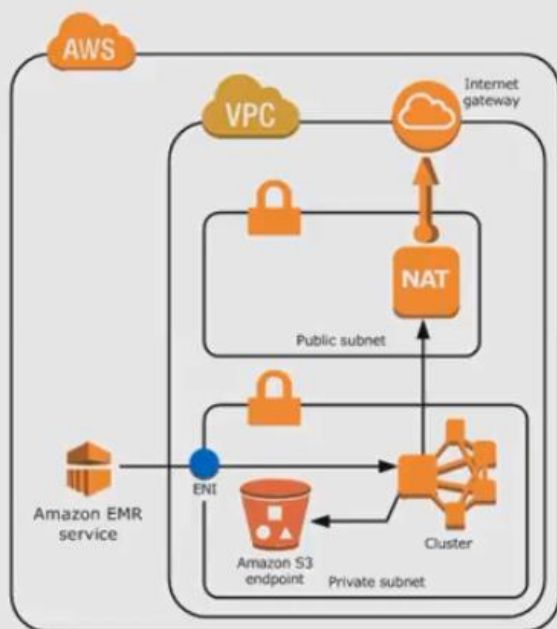
## New – Run HBase on S3



## FINRA saved 60% by moving to HBase on EMR



## Security - Configuring VPC subnets



- Use Amazon S3 endpoints in VPC for connectivity to S3
- Use managed NAT for connectivity to other services or the Internet
- Control the traffic using security groups
  - ElasticMapReduce-Master-Private
  - ElasticMapReduce-Slave-Private
  - ElasticMapReduce-ServiceAccess

You can run EMR in a private subnet and use a S3 private endpoint within the subnet

# Access control by cluster tag and IAM roles



IAM user: MyUser



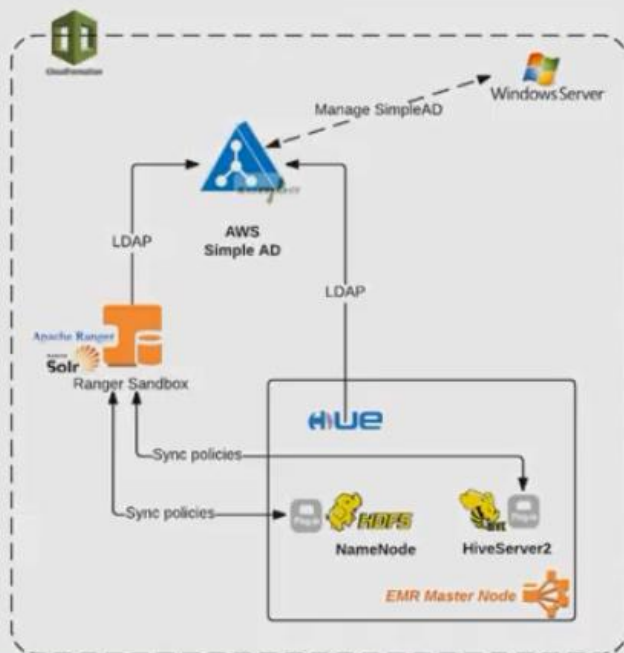
EMR role  
EC2 role  
SSH key

Tag: user = MyUser

```
5  "Sid": "Stmt1479329681000",
6  "Effect": "Allow",
7  "Action": [
8    "elasticmapreduce:AddTags",
9    "elasticmapreduce:RunJobFlow"
10 ],
11 "Condition": {
12   "StringEquals": {
13     "elasticmapreduce:RequestTag/user": "MyUser"
14   }
15 },
16 "Resource": [
17   "*"
18 ]
19 }
```

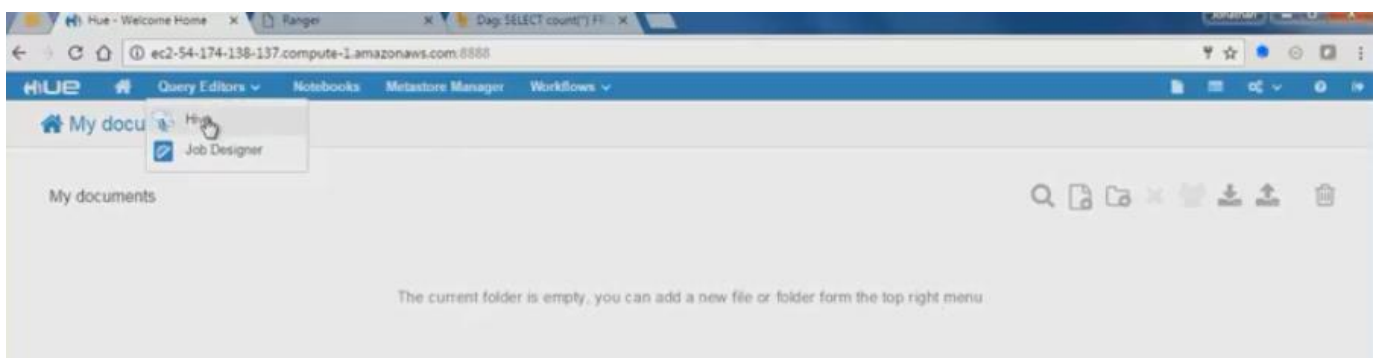
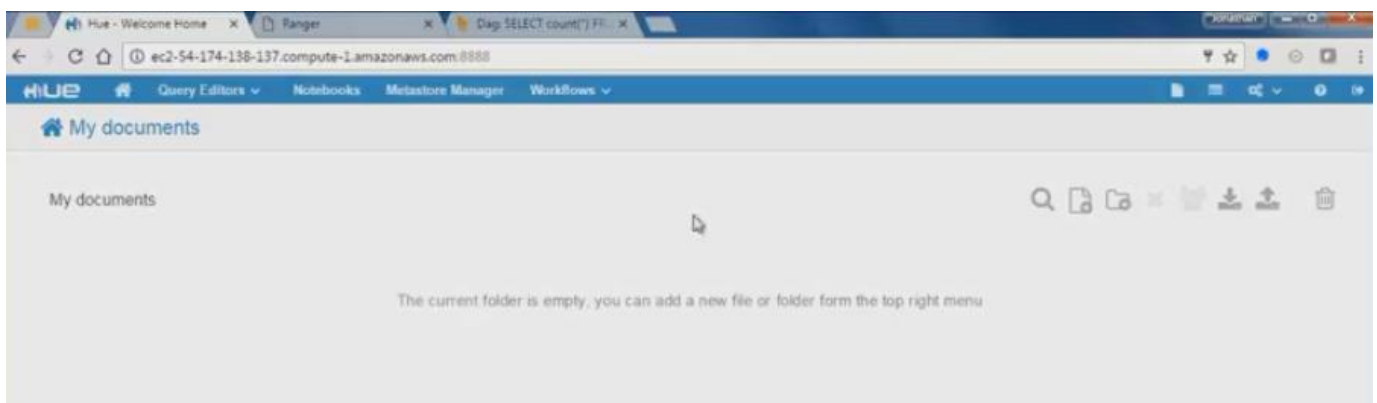
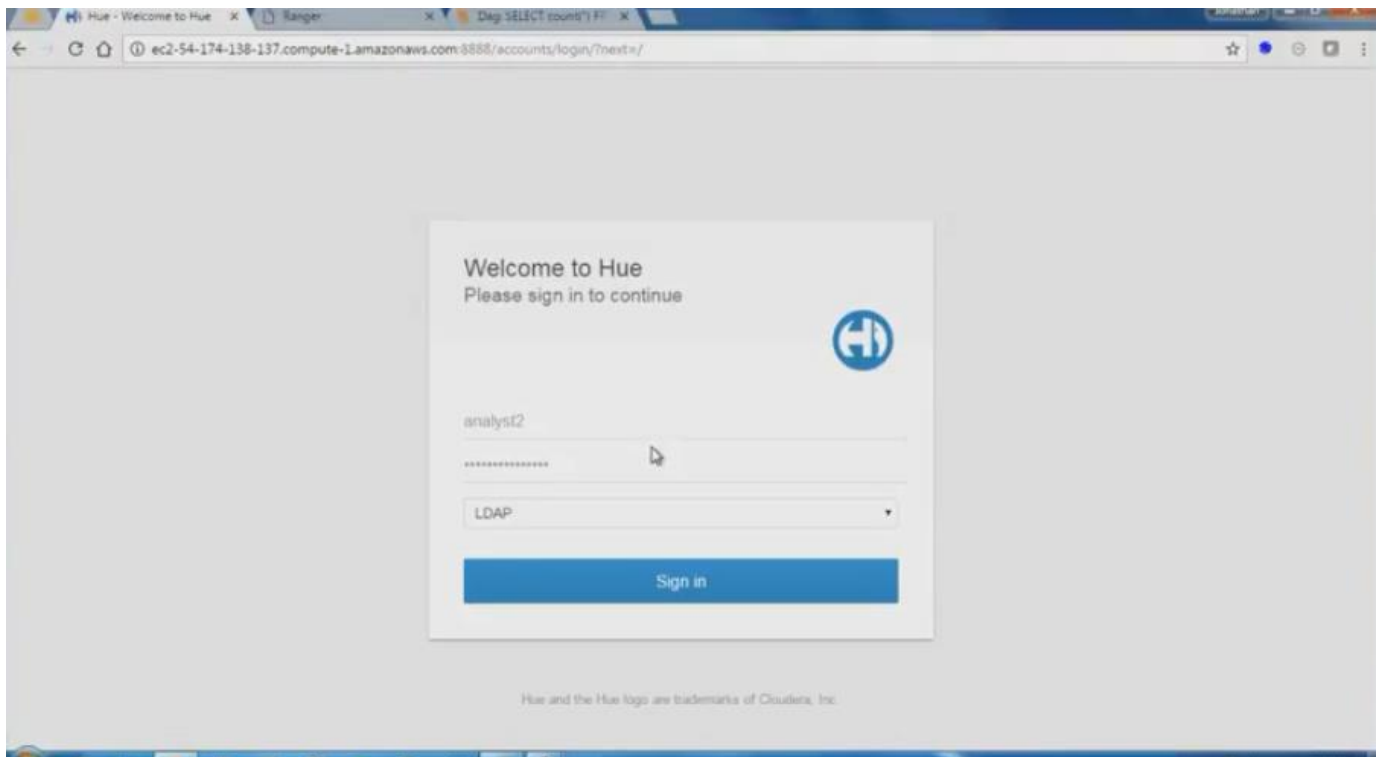
```
5  "Effect": "Allow",
6  "Action": [
7    "elasticmapreduce:AddJobFlowSteps",
8    "elasticmapreduce:DescribeCluster",
9    "elasticmapreduce:DescribeStep",
10   "elasticmapreduce:ListSteps",
11   "elasticmapreduce:TerminateJobFlows"
12 ],
13 "Condition": {
14   "StringEquals": {
15     "elasticmapreduce:ResourceTag/user": "MyUser"
16   }
17 },
18 "Resource": [
19   "*"
20 ]
21 }
```

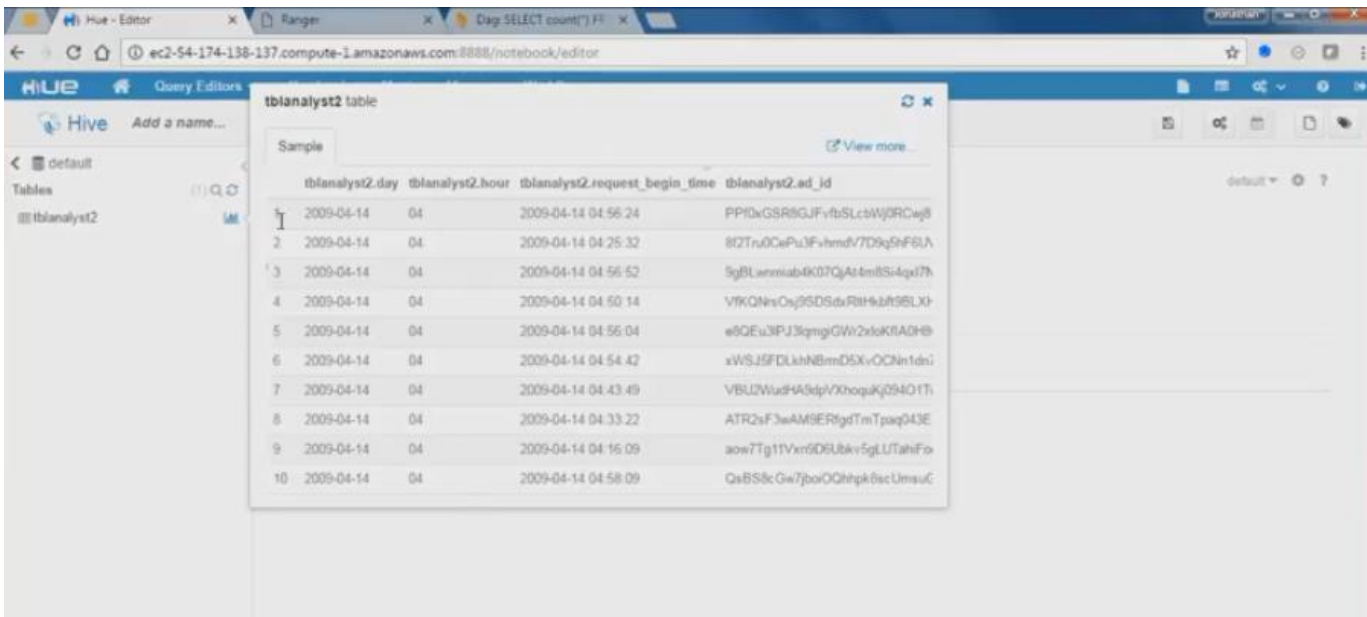
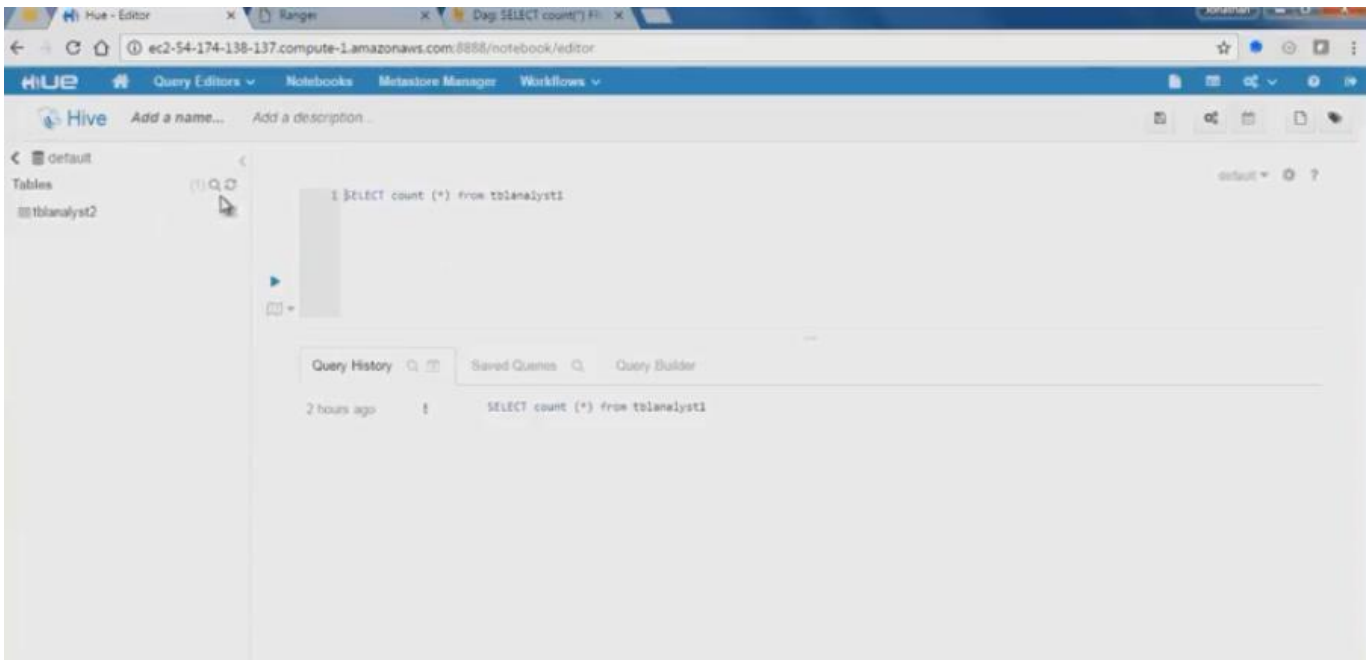
## Fine-grained access control with Apache Ranger



- Plug-ins for Hive, HBase, YARN, and HDFS
- Configure Hue and Zeppelin authentication with LDAP/AD
- Row-level authorization for Hive (with data-masking)
- Full auditing capabilities with embedded search
- Run Ranger on an edge node – visit the AWS Big Data Blog







Hue - Editor Ranger Dag: SELECT count(\*) FROM

ec2-54-174-138-137.compute-1.amazonaws.com:8888/notebook/editor

Hive Query Editors Notebooks Metastore Manager Workflows

Hive Add a name... Add a description...

default

Tables

tblanalyst2

```
1 SELECT count (*) from tblanalyst1
```

Error while compiling statement: FAILED: HiveAccessControlException Permission denied: user [analyst2] does not have [SELECT] privilege on [default:tblanalyst1]

Query History Saved Queries Query Builder

2 hours ago SELECT count (\*) from tblanalyst1

Hue - Editor Ranger Dag: SELECT count(\*) FROM

52.72.106.110:6080/index.html#/reports/audit/bigdata

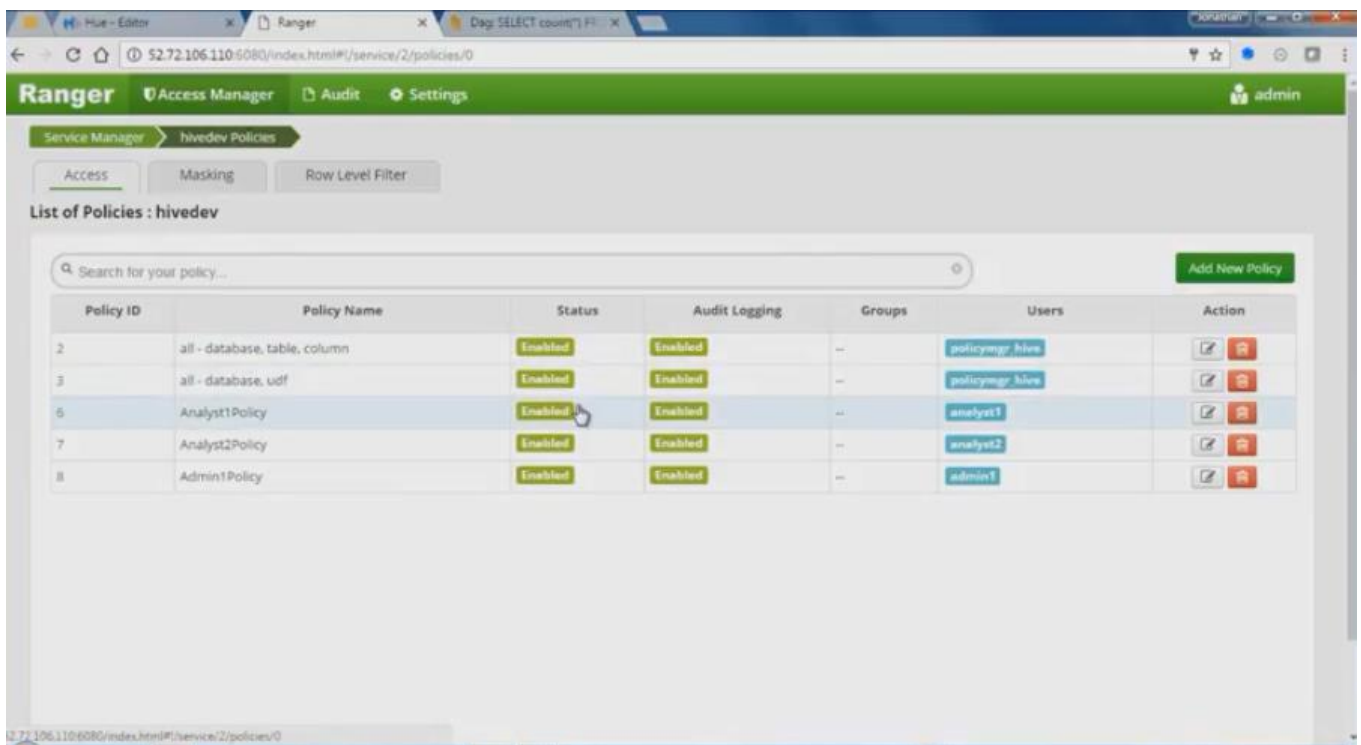
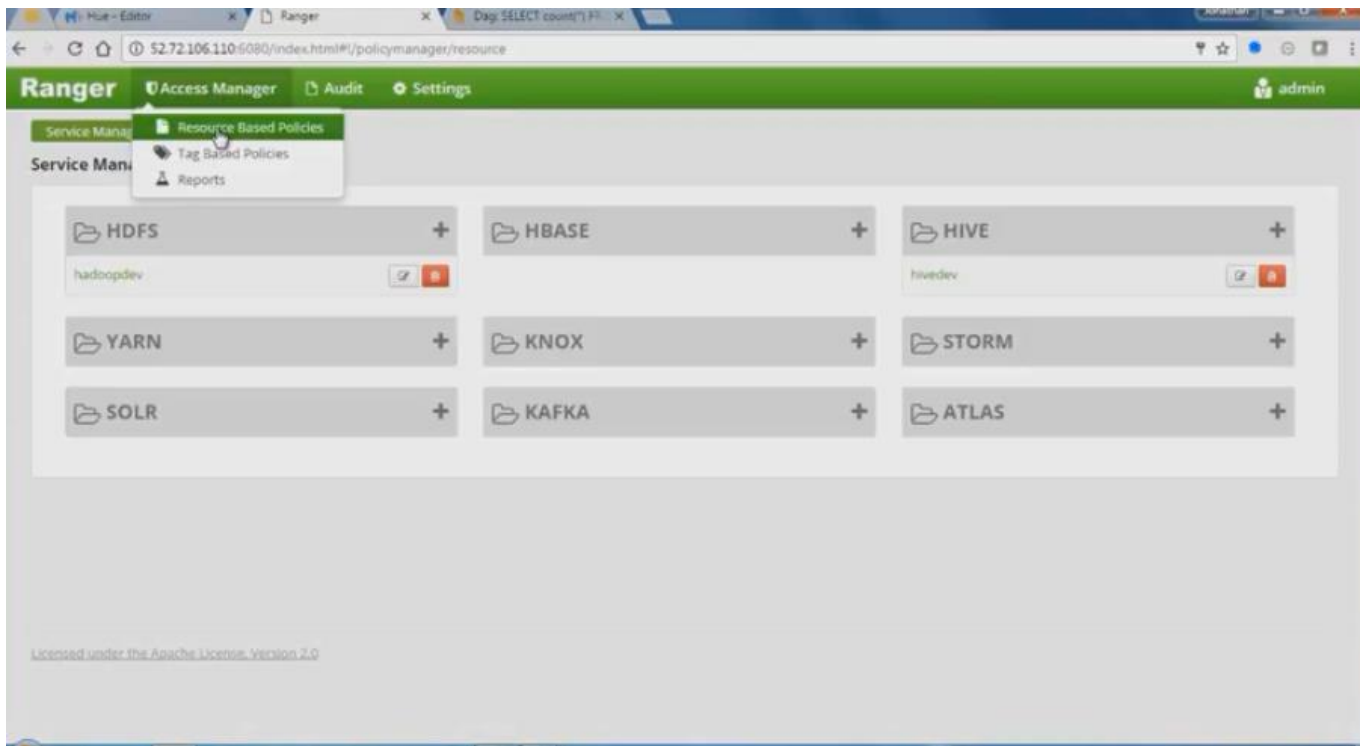
Ranger Access Manager Audit Settings admin

Access Admin Login Sessions Plugins

START DATE: 11/29/2016 USER: analyst2

Last Updated Time: 11/29/2016 12:50:33 PM

Policy ID	Event Time	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Client IP	Event Count	Tags
--	11/29/2016 10:57:55 AM	analyst2	hadoopdev hdfs	/tmp/hive/analyst2/7acdb5f0-126... path	WRITE	Allowed	hadoop-acl	10.0.0.150	1	
--	11/29/2016 10:57:55 AM	analyst2	hadoopdev hdfs	/tmp/hive/analyst2/7acdb5f0-126... path	WRITE	Allowed	hadoop-acl	10.0.0.150	1	
--	11/29/2016 10:57:55 AM	analyst2	hivedev hive	default/tblanalyst1 @table	SELECT	Denied	ranger-acl	10.0.0.150	1	
--	11/29/2016 10:57:55 AM	analyst2	hadoopdev hdfs	/tmp/hive/analyst2/7acdb5f0-126... path	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.150	1	
--	11/29/2016 10:57:55 AM	analyst2	hadoopdev hdfs	/tmp/hive/analyst2/7acdb5f0-126... path	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.150	1	
--	11/29/2016 10:57:55 AM	analyst2	hadoopdev hdfs	/tmp/hive/analyst2/7acdb5f0-126... path	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.150	1	
--	11/29/2016 10:57:55 AM	analyst2	hadoopdev hdfs	/tmp/hive/analyst2/7acdb5f0-126... path	READ_EXECUTE	Allowed	hadoop-acl	10.0.0.150	1	





Hue - Editor Ranger Day: SELECT count(\*) FROM ... jonathan

52.72.106.110:6080/index.html#/service/2/policies/7/edit

Ranger Access Manager Audit Settings admin

Service Manager hivedev Policies Edit Policy

### Edit Policy

**Policy Details :**

Policy Type **Access**

Policy ID **7**

Policy Name \* Analyst2Policy **enabled**

Hive Database \* default **include**

table \* tblanalyst2 **include**

Hive Column \* \* **include**

Audit Logging **YES**

Description Hive Policy

52.72.106.110:6080/index.html#/service/2/policies/7/edit

Hue - Editor Ranger Day: SELECT count(\*) FROM ... jonathan

52.72.106.110:6080/index.html#/service/2/policies/7/edit

Ranger Access Manager Audit Settings admin

Policy Name \* Analyst2Policy **enabled**

Hive Database \* default **include**




table \* tblanalyst2 **include**

Hive Column \* \* **include**

Audit Logging **YES**

Description Hive Policy

**Allow Conditions :**

Select Group	Select User	Permissions	Delegate Admin
Select Group	* analyst2	All	 
			

# Encryption – Use security configurations

Name:

☒ **At-rest encryption**  
Enable and choose options for at-rest data encryption features in Amazon EMR, including Amazon S3 with EMRFS, local volumes attached to cluster instances, and block-transfer encryption for HDFS. [Learn more](#)

**S3 encryption** ⓘ

Encryption mode:  ⓘ

AWS KMS Key:  ⓘ

**Local disk encryption** ⓘ

Key provider type:  ⓘ

AWS KMS Key:  ⓘ

☒ **In-transit encryption**  
Enable and choose options for open-source encryption features that apply to in-transit data for specific applications. Available encryption options may vary by Amazon EMR release. [Learn more](#)

**TLS certificate provider**

Certificate provider type:  ⓘ

S3 object:  ⓘ

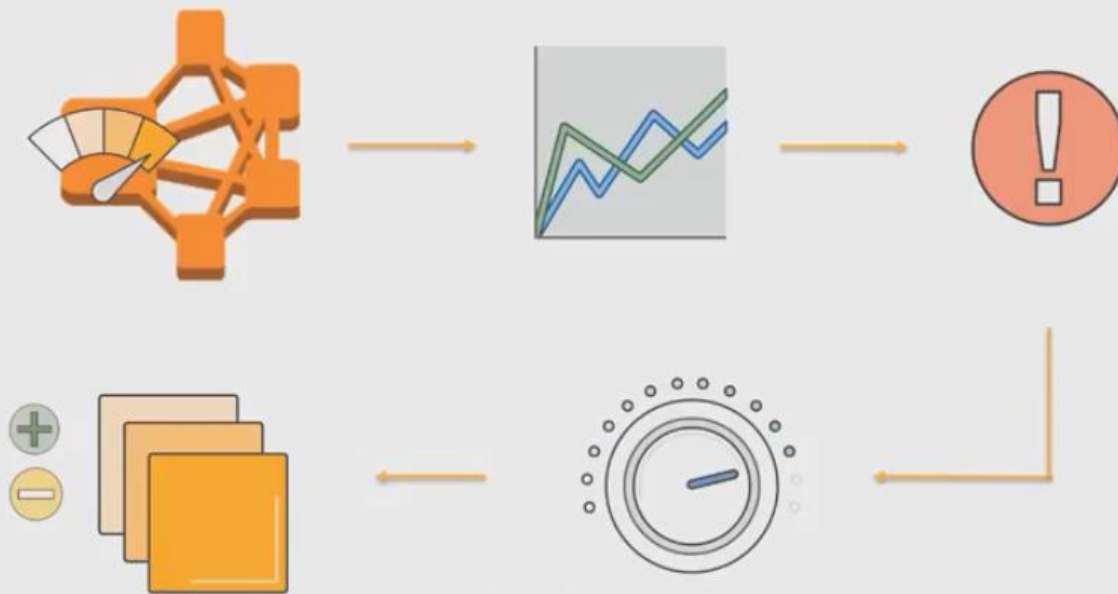
Supported encryption features vary by EMR release

## Nasdaq uses Presto on Amazon EMR and Amazon Redshift as a tiered data lake



Full Presentation: <https://www.youtube.com/watch?v=LuHxnOQarXU>

## New - Auto Scaling



## Coming Soon: Advanced Spot provisioning

Master Node



Core Instance Fleet



Task Instance Fleet



- Provision from a list of instance types with Spot and On-Demand
- Launch in the most optimal Availability Zone based on capacity/price
- Spot Block support

## Building a data lake at Asurion

- Introduction
- Logical Architecture
- S3\EMR - Processing
- Low Latency Query – Data Analysts
- Sandbox – Support Data Scientist
- Auto Scaling – Cost Efficiency
- Security
- Summary / take away points

## Asurion's continuous innovation is helping 290M customers globally stay connected while driving loyalty to our partners' brands

### Corporate Overview

- Founded in the mid 1990's, Asurion has been serving the communications and retail industries for over 20 years
- Based in Nashville, Tennessee, Asurion has over 17,000 associates worldwide
- Serving more than 290 million consumers globally through our operations in 18 countries:
  - Australia
  - Brazil
  - Canada
  - China/Hong-Kong
  - Colombia
  - England
  - France
  - Israel
  - Japan
  - Korea
  - Malaysia
  - Mexico
  - Philippines
  - Peru
  - Singapore
  - Taiwan
  - Thailand
  - United States
- Asurion is privately-held with annual revenues in excess of \$5.8 billion
- Our management team comes from best-in-class companies with experience across mobile, wireline telecom, logistics, insurance, service contracts, consulting, customer care, marketing, retail and more
- Asurion partners with the worlds leading mobile carriers, retailers cable satellite and cable providers.

### Expanding Global Presence

- North America**
- Global Headquarters
  - 15 Corporate Owned Call Centers
  - Logistics Center
- South America**
- 2 Corporate Offices



- Europe**
- 3 Corporate Offices
  - 1 Corporate Owned Call Center

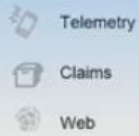
- Asia Pacific**
- 13 Corporate Offices
  - Logistics Center
  - 2 Corporate Owned Call Centers



## Growing Data From Multiple Sources...

A wide variety of data...

### Structured



### Unstructured



...at a large scale...

290 million customers worldwide  
10 Billion interactions annually  
52 million voice interactions annually  
24 million claims annually  
24 million online visitors annually

...that continues to accelerate

Petabytes (>1 million GB) by EOY



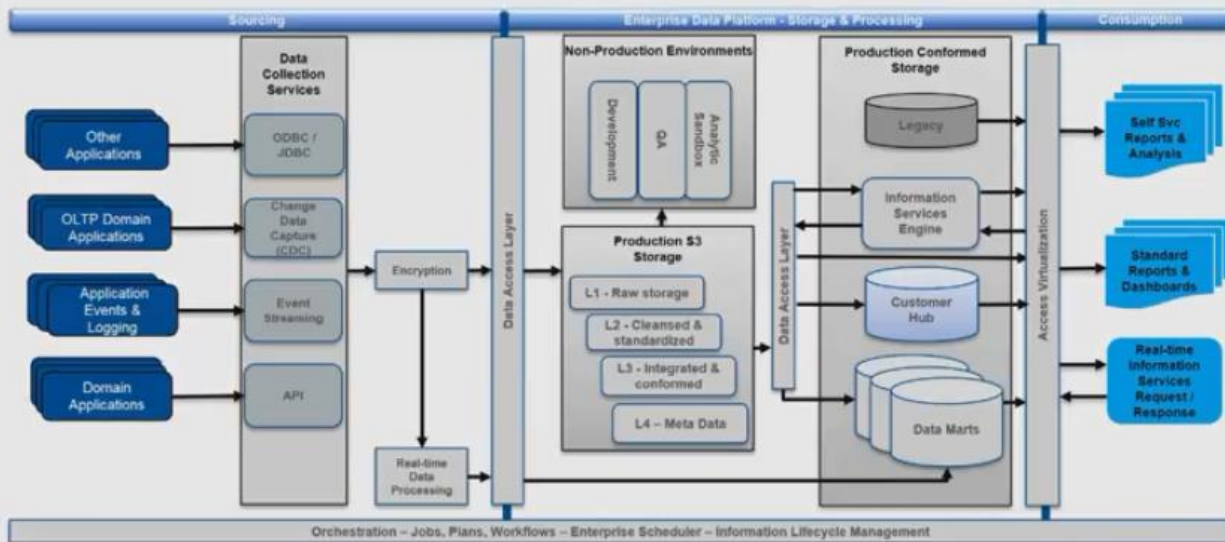
## Data Platform Solution Guiding Principles

- Store *all* enterprise data & enable user access
- ELT instead of ETL. Transform JIT
- Data quality, an absolute must
- Tackle data security at the foundation level
- Embrace variability, velocity, and volume
- Focus on value, agility, and flexible delivery
- Scale On Demand without manual intervention
- Pay as you go



Platform-as-a-Service based core architecture

# Data Platform Solution Logical Architecture

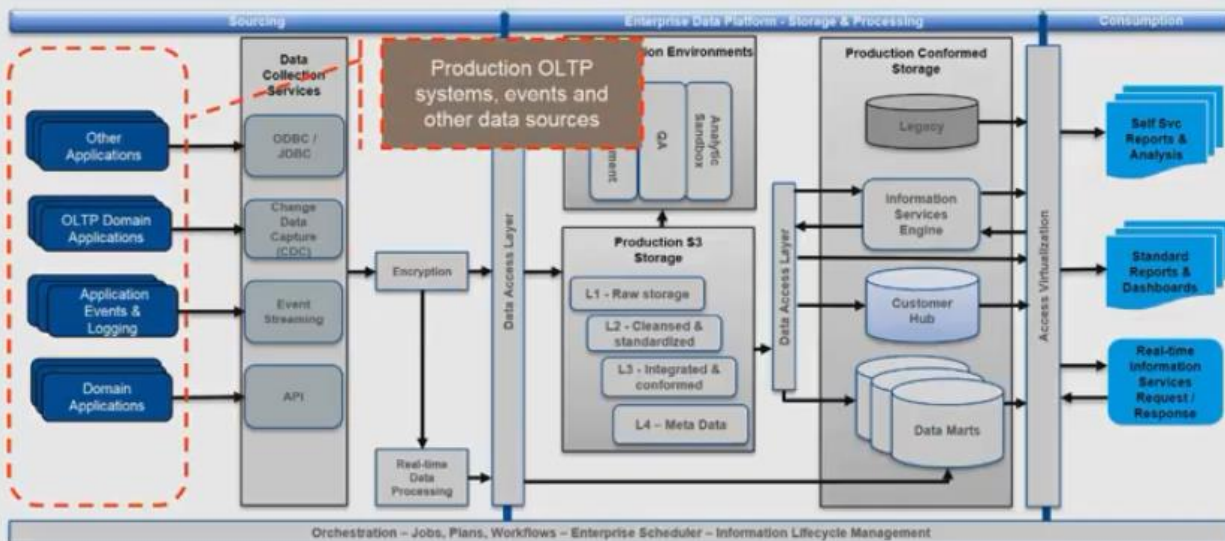


© Asurion. All rights reserved.

25

asurion

# Data Platform Solution Logical Architecture

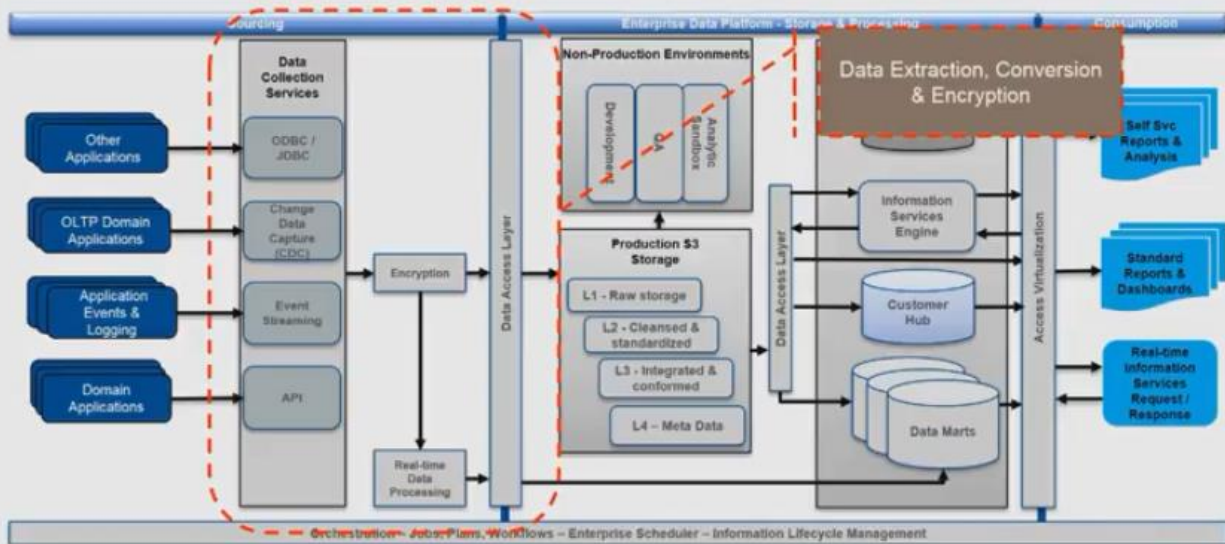


© Asurion. All rights reserved.

25

asurion

# Data Platform Solution Logical Architecture

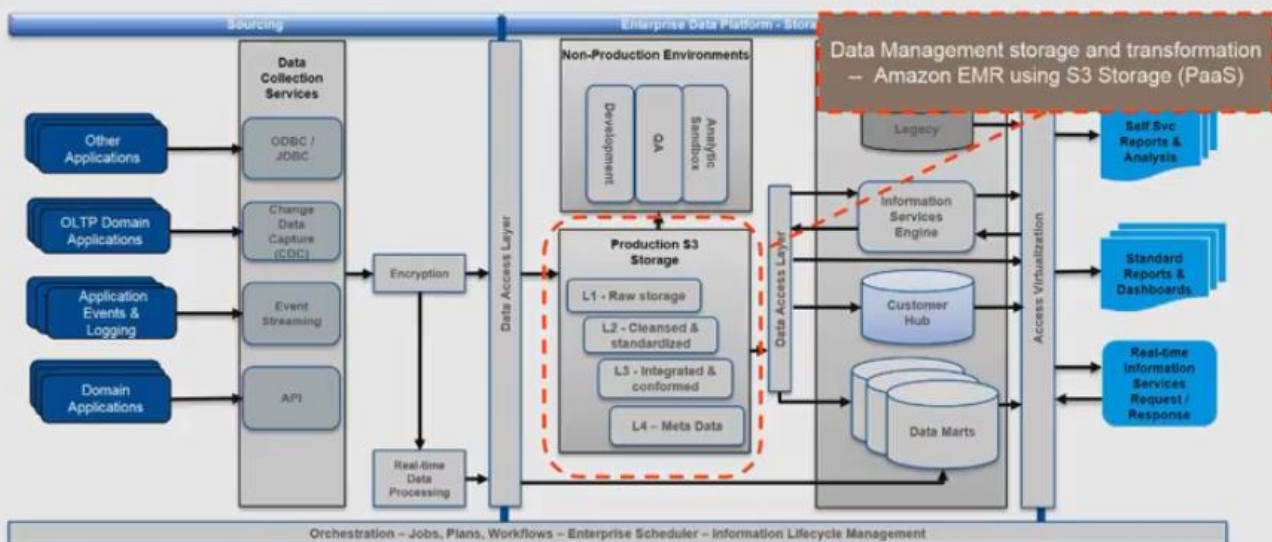


© Asurion. All rights reserved

25

asurion

# Data Platform Solution Logical Architecture



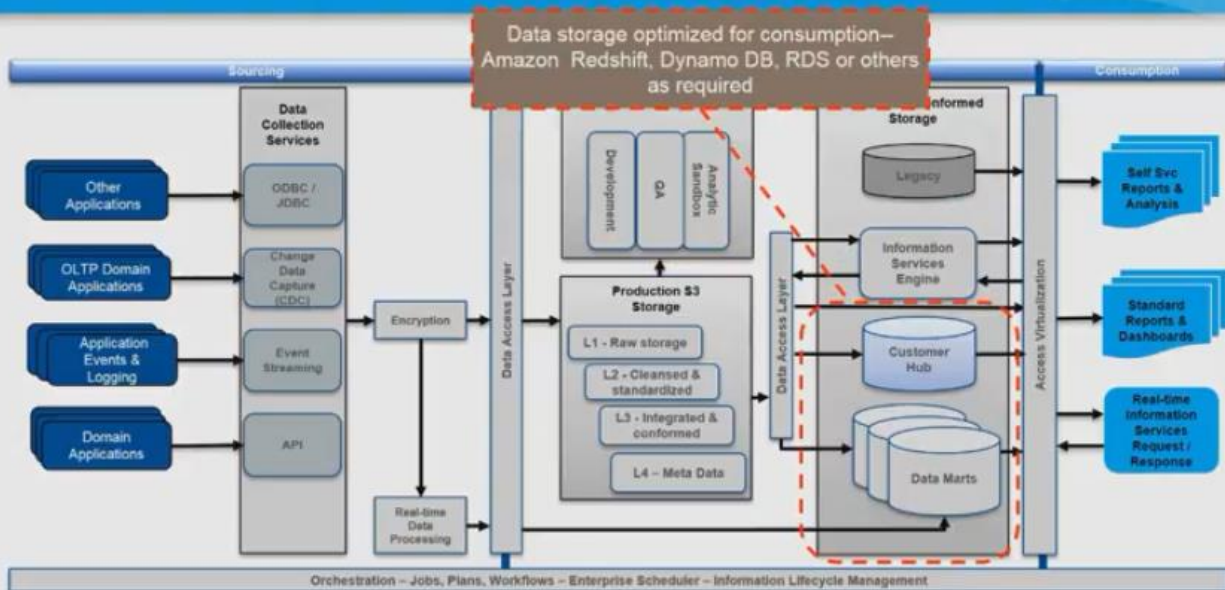
© Asurion. All rights reserved

25

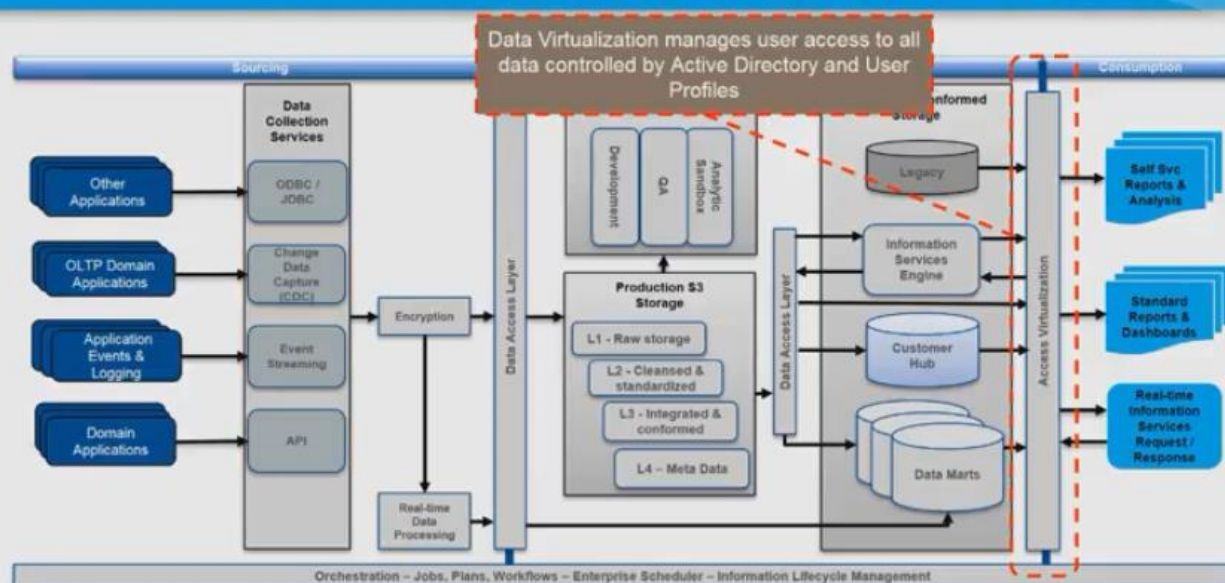
asurion



# Data Platform Solution Logical Architecture

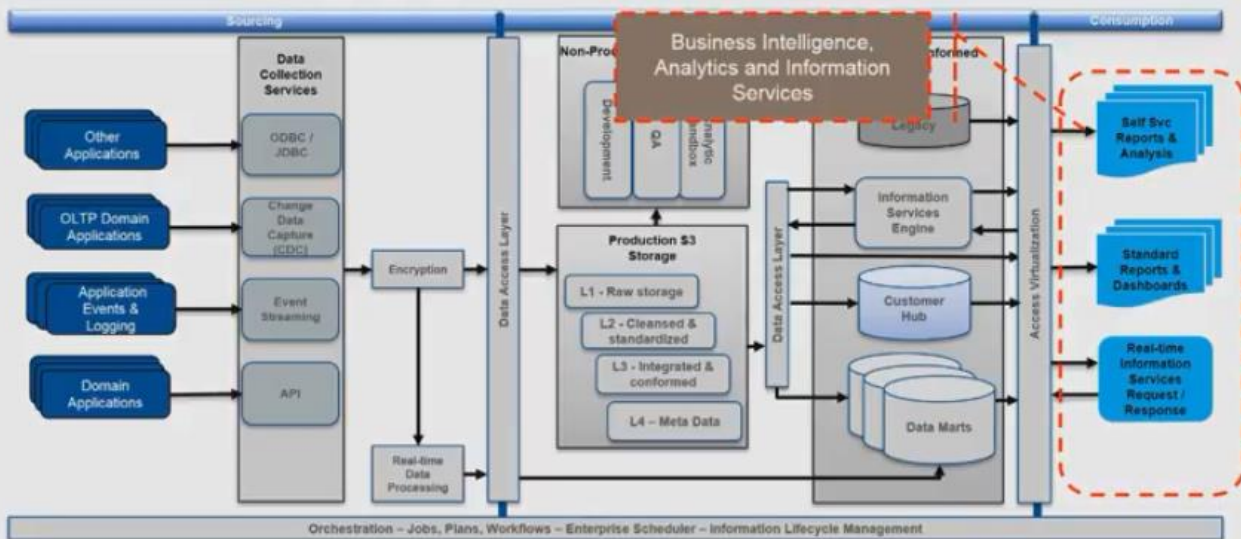


# Data Platform Solution Logical Architecture





# Data Platform Solution Logical Architecture

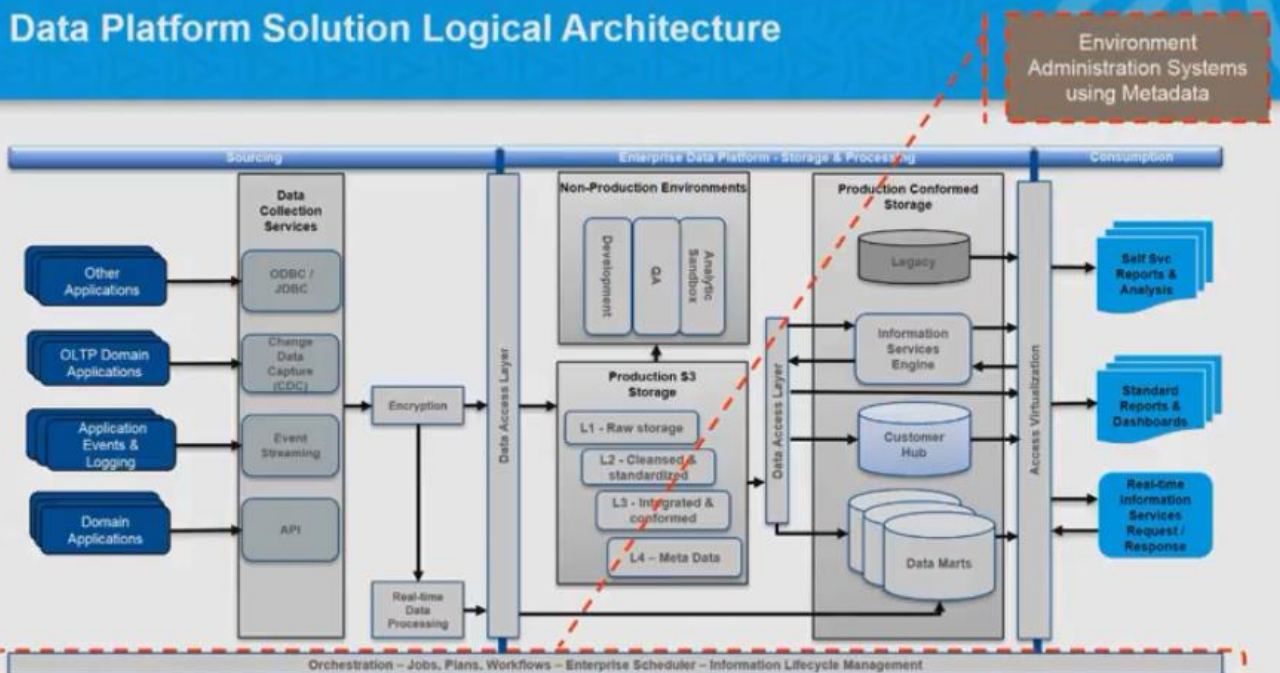


© Asurion. All rights reserved.

25

asurion

# Data Platform Solution Logical Architecture



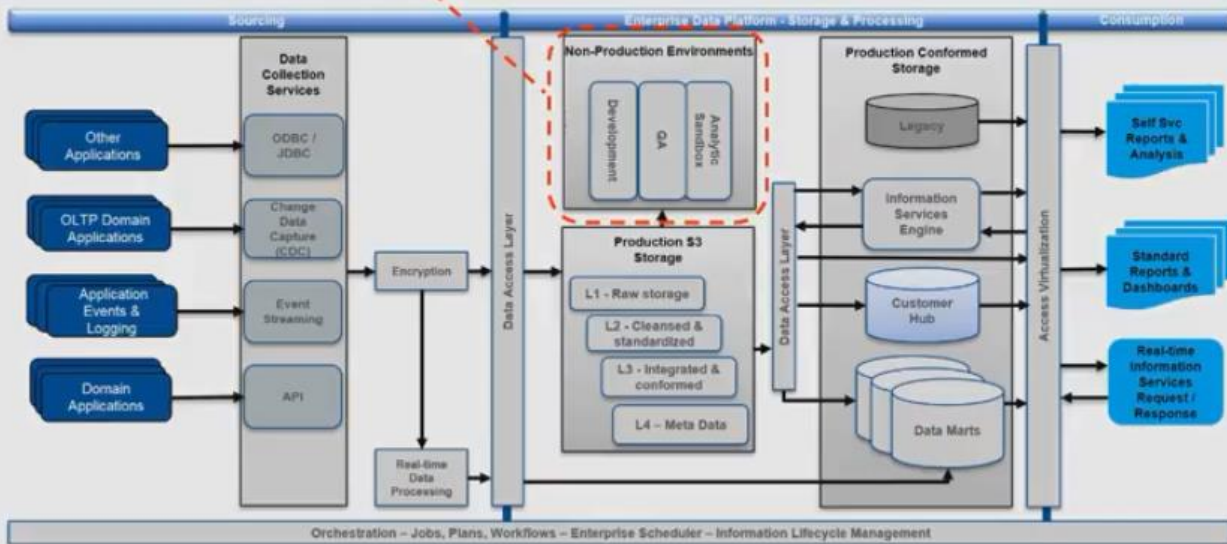
© Asurion. All rights reserved.

25

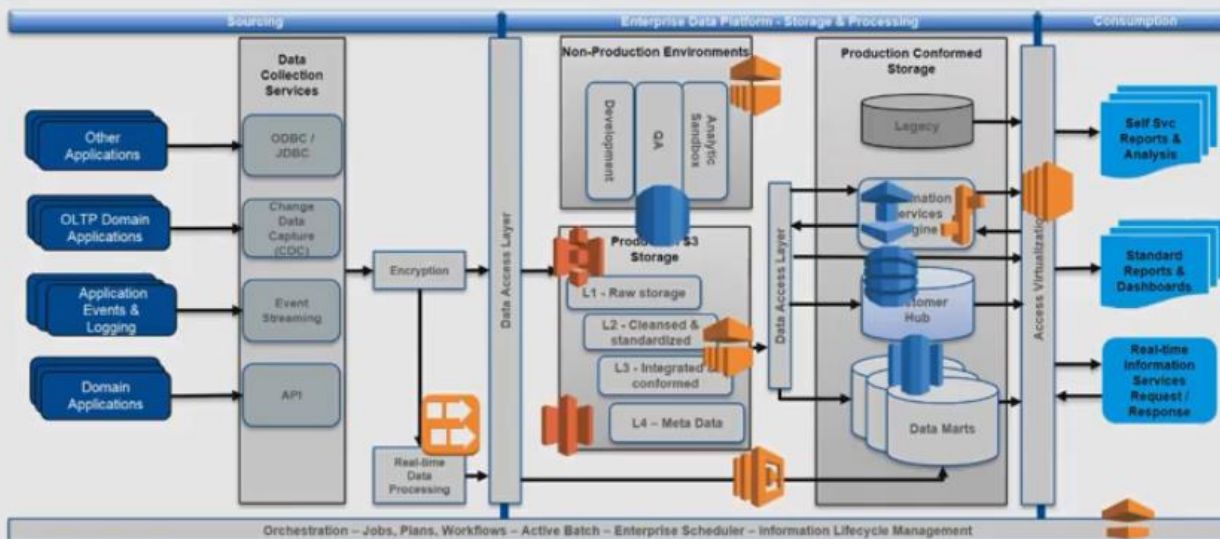
asurion

## Data Platform Solution Logical Architecture

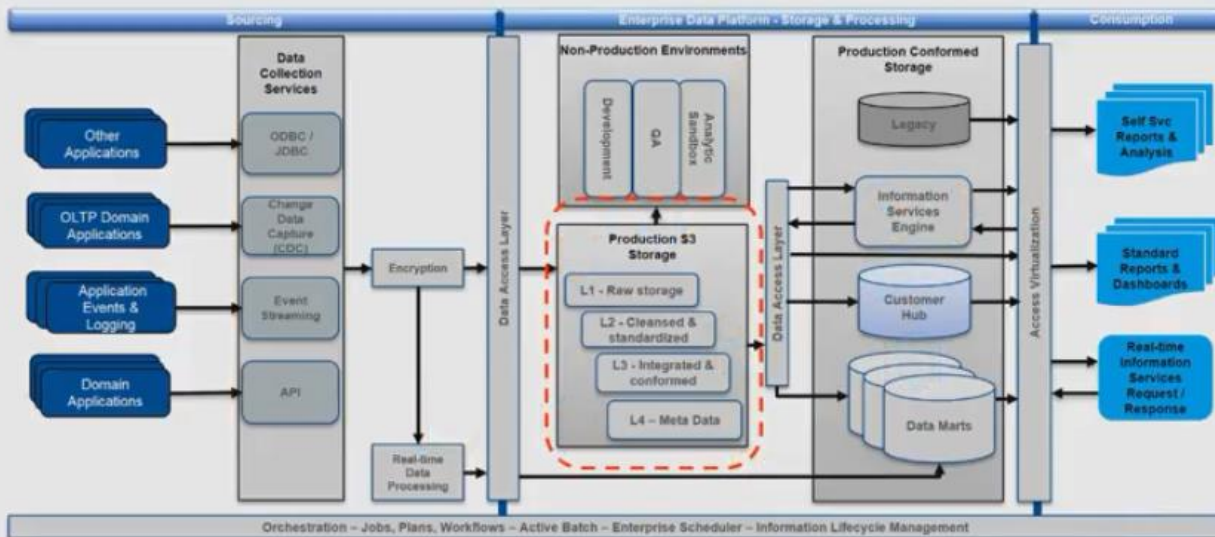
Development, Business Sandbox and QA environments for creation of new capabilities



## Logical Architecture – Amazon Web Services



# Using S3 as a Central Data Lake



© Asurion. All rights reserved.

27

asurion

## Storage & Ingestion – Key Takeaways



### Ingestion Transformation

- Handle CR, LF, and delimiter in the data
- Handle time to preserve the time zone
- Handle multi-byte characters – UTF8
- Split & Compress Files 128 MB are more efficient
- Partition data for performance
- S3 object path is case-sensitive – lowercase



### Data Transfer to S3

- Python Boto
- Multipart upload
- Storage class (Standard, RRS, IA)
- Lifecycle policies for cost savings



### S3

- Compute proximity to storage
- Handle request throttling – partition data in buckets
- Tagging for cost allocation

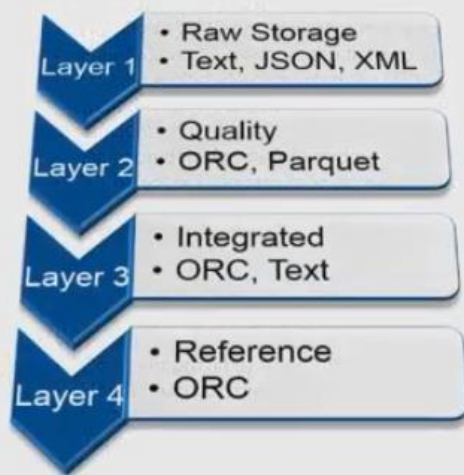


### S3 Security

- S3 bucket policies
- IAM, ACL



## Data Storage Layers



### Layer 1

- Source of truth
- Minimal transformation
- Field Level SPI/PII Encryption
- Select a delimiter
- Cleanse data before ingesting for CR, LF, delimiter
- Standardize all times to GMT

### Layer 2

- Data Cleansing, Profiling on EMR – Hive
- Partition the data based on query usage – ORC
- Handle deletes and updates – Merge pattern

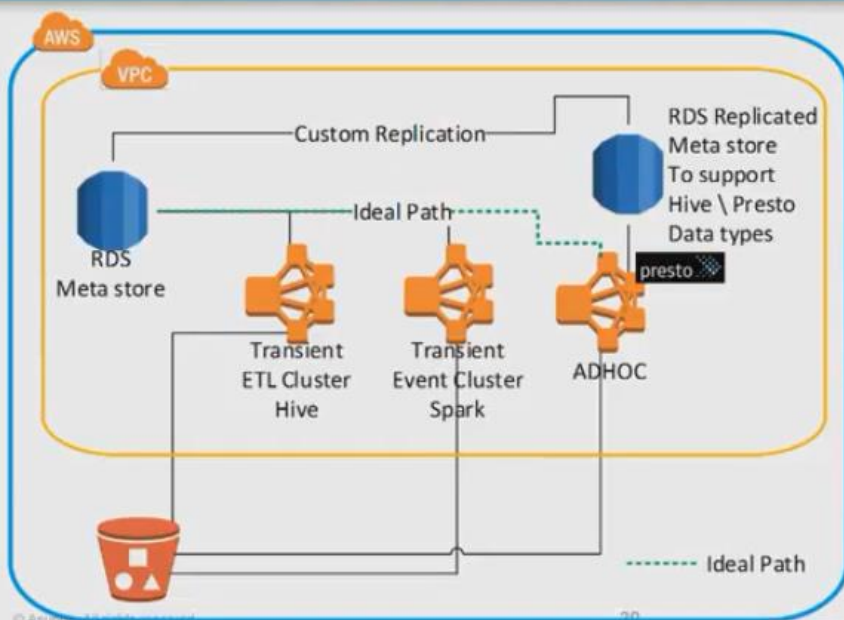
### Layer 3

- Integrate data from multiple systems
- Model based on Data Vault Pattern – EMR

### Layer 4

- Conformed, Master & Reference Data

## Using EMR for Data Processing



### Data Lake

- US
- EU
- JAPAN

### Data Lake - Stats

- 50+ Ad hoc Users
- 1000+ Ad hoc Queries \ Day
- 20+ Sources of Data
- 100+ ETL Hive Jobs
- 25+ Spark Jobs
- 2+ PB of Data



## EMR Key Takeaways

### EMR



- Use S3 path when creating Hive database
- Use external tables for the data in S3
- Use external metastore
- Recover partitions automatically with MSCK repair
- Alter table to add partitions
- EMRFS - choose between "s3" or "s3n" (not "s3a")
- Clusters are transient /long running
- Data Pipeline/Python to launch transient cluster

### Compression



- Gzip - high compression – not splittable
- Snappy – low compression – splittable

### Hive Takeaways



- Partitioning
- De-Normalizing
- Speculative Execution
- File Format – ORC
- `hive.vectorized.execution.enabled`
- `hive.exec.parallel`
- `hive.hadoop.supports.splittable.combineinputformat`
- `hive.exec.compress.intermediate`
- `hive.intermediate.compression.codec`
- `hive.auto.convert.join`

## EMR – Presto for Low-Latency BI



- ORC most interoperable
- Predicate push down
- Low-latency BI ad hoc queries
- R3.2xlarge has yielded better results

### Presto – Watch out

- Bucketing is a challenge
- Complex data structures
- Memory limit
- Data type float, char(), varchar()

### Presto Settings – Gave optimal results

- `query.max-memory`
- `query.client.timeout`
- `query.max-age`
- `query.max-memory-per-node` – max. memory on a node for a query. 42 %
- `node-scheduler.max-splits-per-node`
- `optimizer.columnar-processing`
- `hive.orc.bloom-filters.enabled`
- `hive.rcfile-optimized-reader.enabled`
- `hive.msck.path.validation`

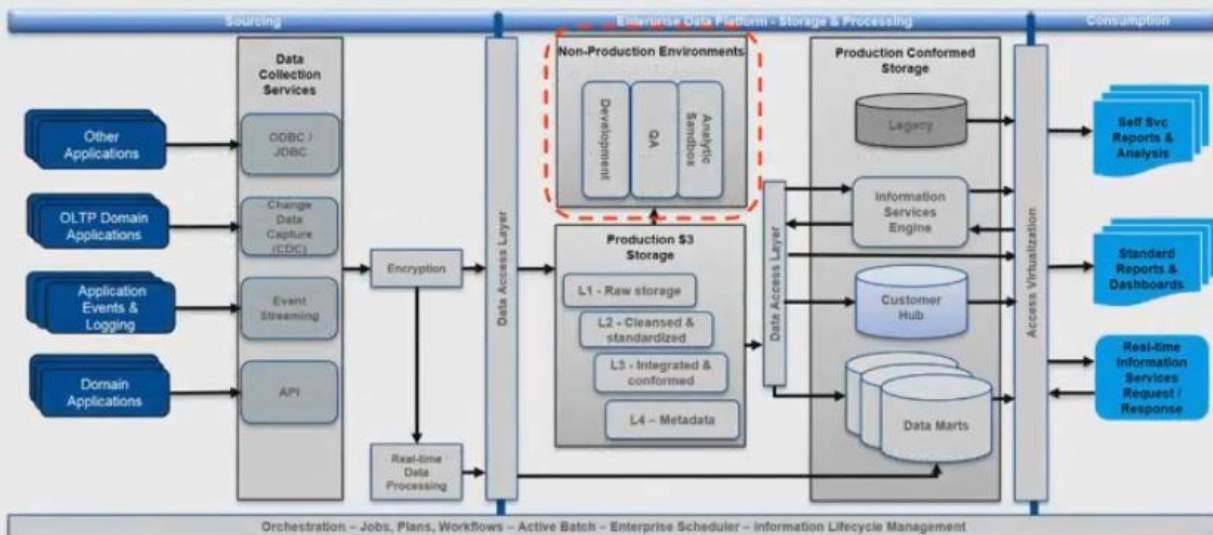
## EMR - External Hive Metastore with Presto



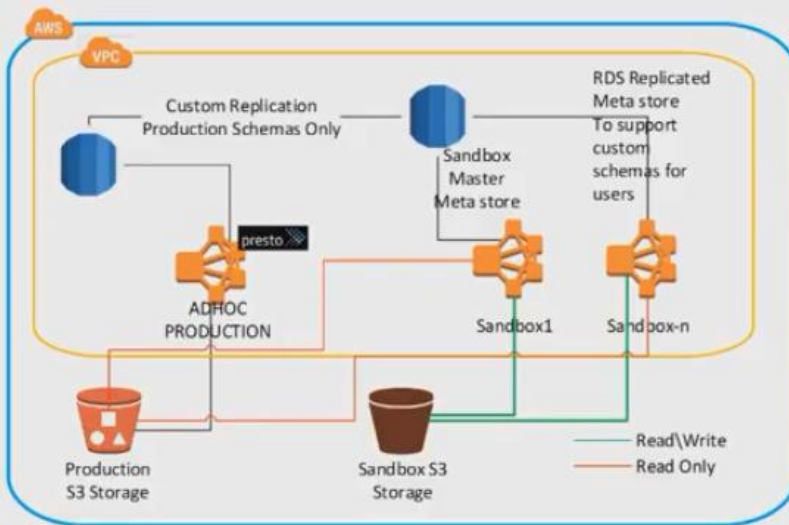
### Metastore

- *Best practice is to leverage a single metastore*
- Second metastore to handle data type challenges between Hive & Presto
- Modify the table schema to account for data type challenges during sync process
- It's temporary until Presto supports Hive data types.
- Ex.
  - Decimal
  - Varchar()
  - Char()
  - Timestamp – with milliseconds
- Most of the above challenges were resolved in Presto 0.152 (EMR 5.x supports it)

## Logical Architecture – Amazon Web Services



## EMR Sandbox



### Sandbox

- Enables data lake access to users
- Enables dedicated compute capacity – instead of working with YARN queues
- Enables cost savings with help of EMR Scaling & Spot

© Asurion. All rights reserved

35

asurion

## EMR Sandbox

### EMR Sandbox



- Separate metastore for production and sandbox
- Single metastore for multiple sandboxes
- Sync production schema DDL with sandbox
- Assign new schema for each sandbox

### EMR Sandbox Security



- Read-only access to Production S3
- Read/write access to Sandbox S3
- Specific sandbox storage with access control
- Controlled via IAM roles & policies

### EMR creation with IAM roles

```
aws emr create-cluster --applications Name=Pig Name=HIVE Name=SPARK Name=ZEPPELIN
Name=GANGLIA --tags 'PLATFORM=ANALYTICS' 'ENVIRONMENT=SB' 'Name=sandbox1' --ec2-attributes
'{"KeyName":"sandbox1","InstanceProfile":"EMR_EC2_Sandbox1Role"}
```

## EMR Sandbox IAM Policies

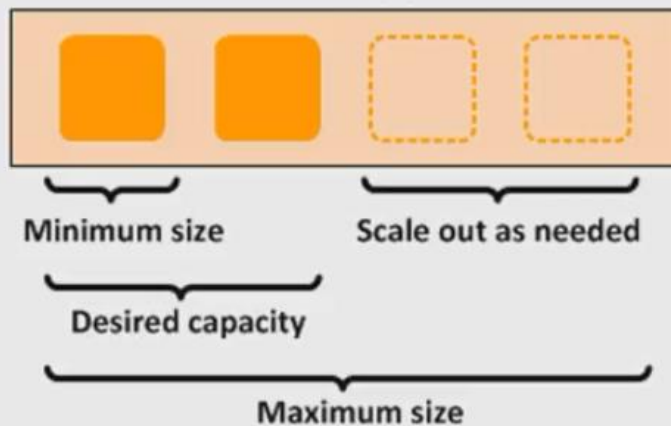
### Policy to protect production data

```
{
  "Effect": "Deny",
  "Action": [
    "s3:DeleteObject", "s3:Put*"
  ],
  "Resource": [
    "arn:aws:s3:::productions3001/*"
  ]
},
```

### Policy to allow sandbox read \ write

```
{
  "Effect": "Allow",
  "Action": [
    "s3:Put*", "s3:DeleteObject"
  ],
  "Resource": [
    "arn:aws:s3:::reinventsandbox1/sandbo
x1/*"
  ]
}
```

## EMR Auto Scaling



Automatic resizing based on CloudWatch metrics

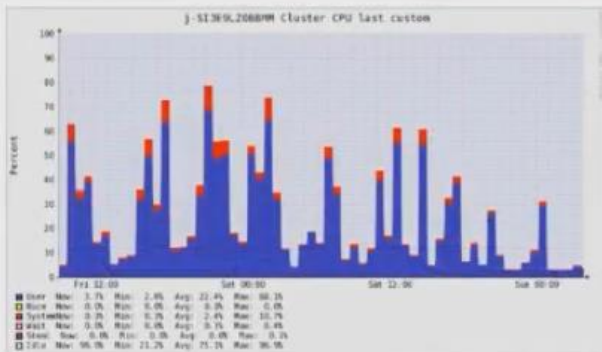
- Is Cluster Idle
- Containers pending
- Apps Pending
- AVG CPU Usage – Custom Metric

Define minimum \ maximum instance count

Define when scaling should occur



## EMR Auto Scaling – Ganglia



Based on the CPU usage of EMR cluster, scaling function adds and removes nodes accordingly

CPU metrics is captured from Ganglia using Lambda

## EMR Auto Scaling – Custom Metrics

### Sample Ganglia JSON

[http://clusterip/ganglia/graph.php?r=hour&c=clusterid&m=load\\_one&s=by+name&mc=2&g=cPU\\_report&json=1](http://clusterip/ganglia/graph.php?r=hour&c=clusterid&m=load_one&s=by+name&mc=2&g=cPU_report&json=1)

```
[{"ds_name": "ccpu_user", "cluster_name": "", "graph_type": "stack", "host_name": "", "metric_name": "User\\g", "color": "#3333bb", "datapoints": [[2.278333333, 1479435030], [2.021666667, 1479435045], [2.465, 1479435060]]}
```

### Calculate Last 1 Minute Average

```
datapointCount = Object.keys(ccpu_user_datapoints).length;  
datapointSlice = ccpu_user_datapoints.slice(Math.max(datapointCount - 6, 1))
```

### Calculate CPU Usage

```
Object.keys(datapointSlice).forEach(function(datapointSliceKey){  
  //console.log('datapointKeyValue', datapointSlice[datapointSliceKey][0]);  
  if(datapointSlice[datapointSliceKey][0] != 'NaN'){  
    TotalCPUUsage = TotalCPUUsage + (datapointSlice[datapointSliceKey][0] - 0);  
    loopCount = loopCount + 1;  
  }  
}); //End of for eachfor datapointSlice  
AvgCpuUsage = (TotalCPUUsage/loopCount);
```



## EMR Auto Scaling – Custom Metrics



### CloudWatch alarm

```
var CloudWatchMetricparams = {
  MetricData: [
    {
      MetricName: 'CPUAvgUsageLastOneMin',
      Timestamp: new Date().toISOString(),
      Unit: 'Percent',
      Value: AvgCpuUsage,
      Dimensions :[
        { Name : 'JobFlowId', Value : emrClusterID}
      ]
    }
  ],
  Namespace: 'EMR-CPUUsage'
}; //End of Paramet definition for Alarm

cloudwatch.putMetricData(CloudWatchMetricparams, function(err, data) {
  if (err) console.log(err, err.stack);
  else console.log(data);
});
```

## EMR Auto Scaling – Cost Savings (55% savings when compared with On Demand)

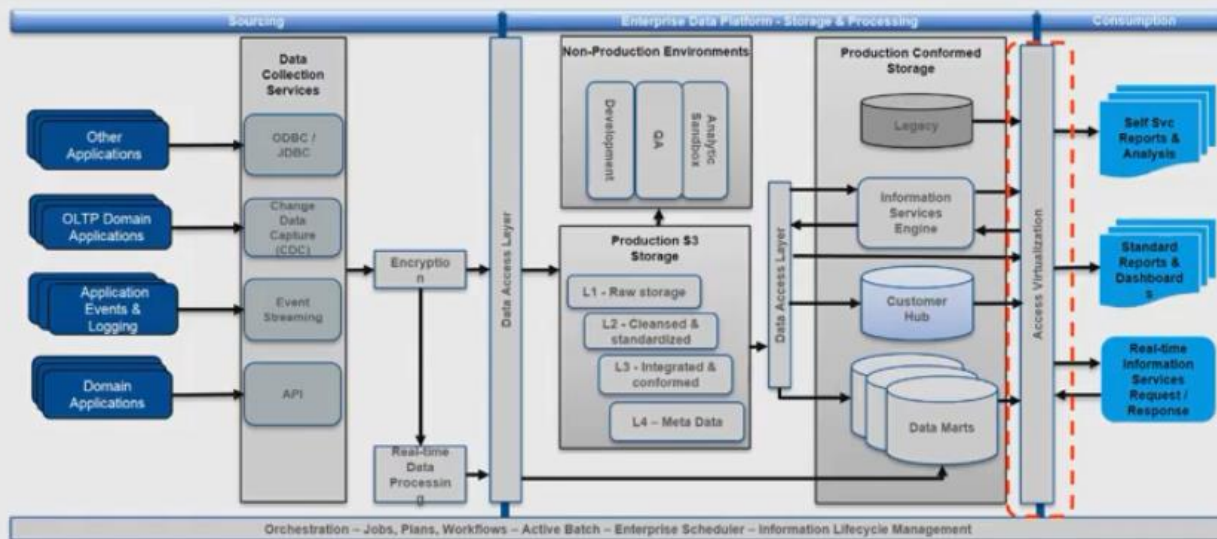
Usage type	Count	Blended cost	Usage quantity hrs.	Unit price	On Demand price
BoxUsage:d2.2xlarge	4	0.36	200	1.38	276
BoxUsage:r3.2xlarge	1	26.83	50	0.665	33.25
BoxUsage:r3.4xlarge	69	272.65	205	1.33	272.65
SpotUsage:m3.xlarge	30	13.44	300	0.266	79.8
SpotUsage:r3.4xlarge	203	277.67	809	1.33	1075.97
		<b>590.97</b>	<b>1564</b>		<b>1737.67</b>

55% cost savings when compared with On Demand

### Note:

- Only EC2 cost is depicted here – EMR cost is not added
- BoxUsage:d2.2xlarge – Reserved Instance
- Without Reserved Instance, approx. 40% cost savings

## Logical Architecture – Data Virtualization

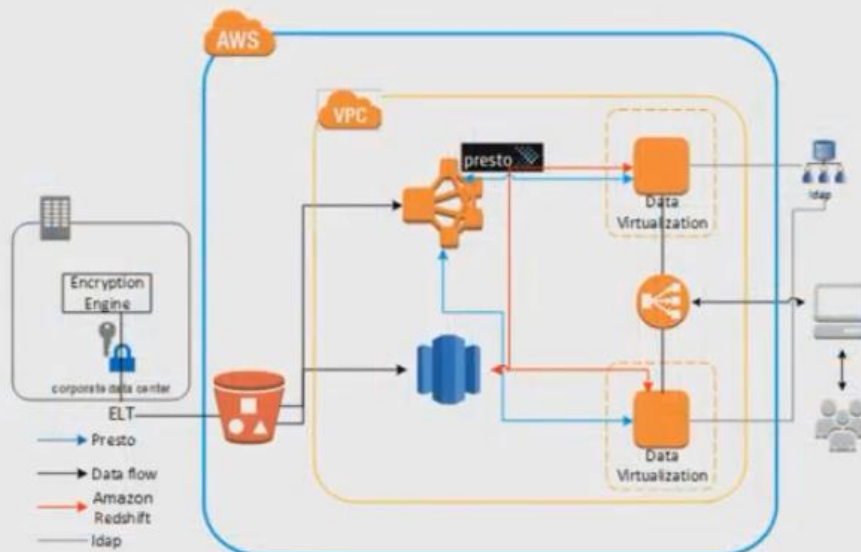


© Asurion. All rights reserved.

43

asurion

## Data Lake Security



© Asurion. All rights reserved.

44

asurion

## Data Lake Security

- Data virtualization to handle data lake security
- ANSI SQL compliant and native pushdown enabled
- Enabled column and row level security
- Users are authenticated by on-premises existing LDAP
- Users are authorized based on the roles defined in data virtualization
- Ad hoc queries & reports are through JDBC & ODBC against Amazon Redshift & Presto

## What have we learned?

- Manage cost – adjustments may be required with scale
- Real-time frictionless scaling – automate where possible
- Align to core design patterns
- Leverage readily available solutions
- Design for security and compliance at the start
- Fail forward and adjust as needed
- Harden solution for one market – Integrate regional variance and deploy globally



AWS  
re:Invent

**Thank you!**