cloudera®

Authorization in the Cloud: Enforcing Access Control Across Compute Engines

Hao Hao - hao.hao@cloudera.com

Li Li - lili@cloudera.com

About us

- Software Engineers @ Cloudera
- Working on Data Access Control projects
- Apache Sentry PMC and committer

cloudera

Presentation Agenda

- Challenges for Authorization in the cloud
- Solution: Apache Sentry + RecordService
- Use Case + Demo
- Project Status

Challenges in the cloud

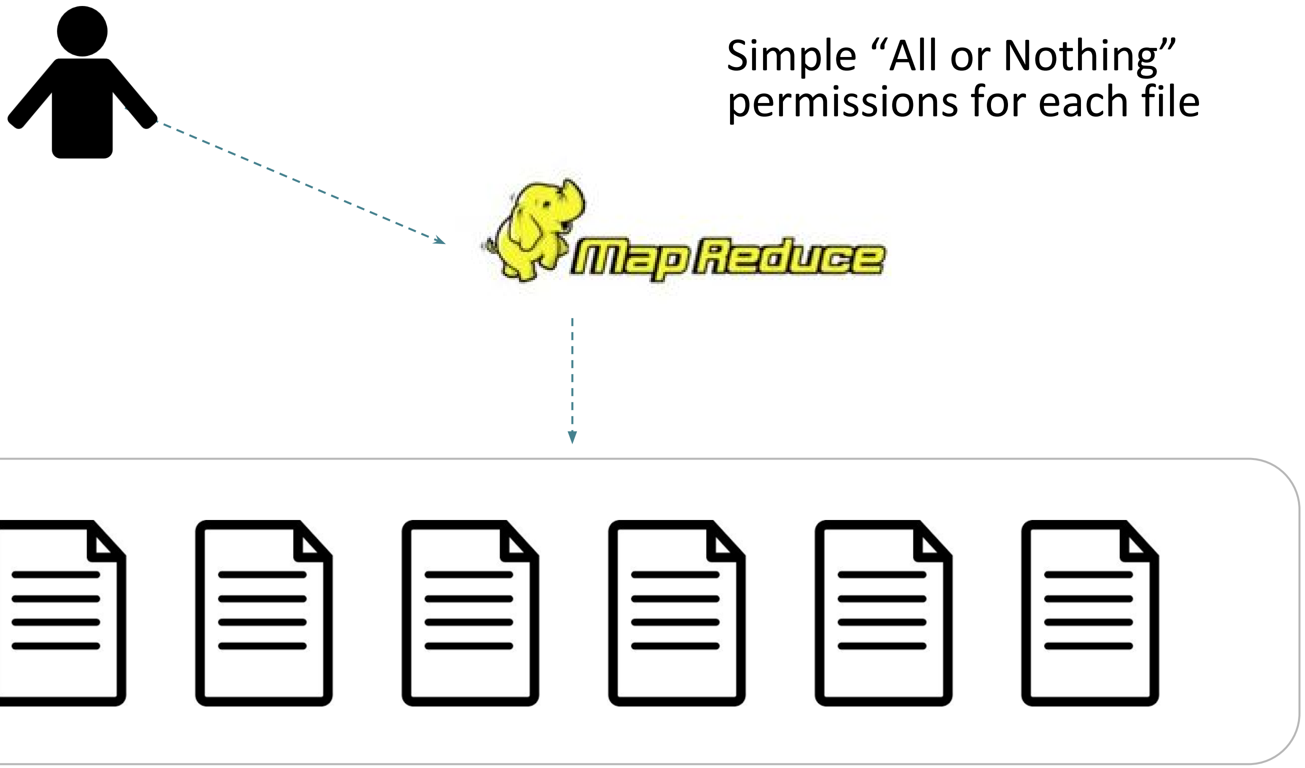
cloudera

Moving to Cloud

- As cloud provides rapid access to flexible and low expense IT resources. Hadoop in Cloud becomes an increasingly common use case.

- “I can’t approve to buy hardware to expand my existing hadoop cluster, because we’ve got a CIO mandate to move IT to the cloud.”

Existing Cloud Provider Security



Simple "All or Nothing" permissions for each file

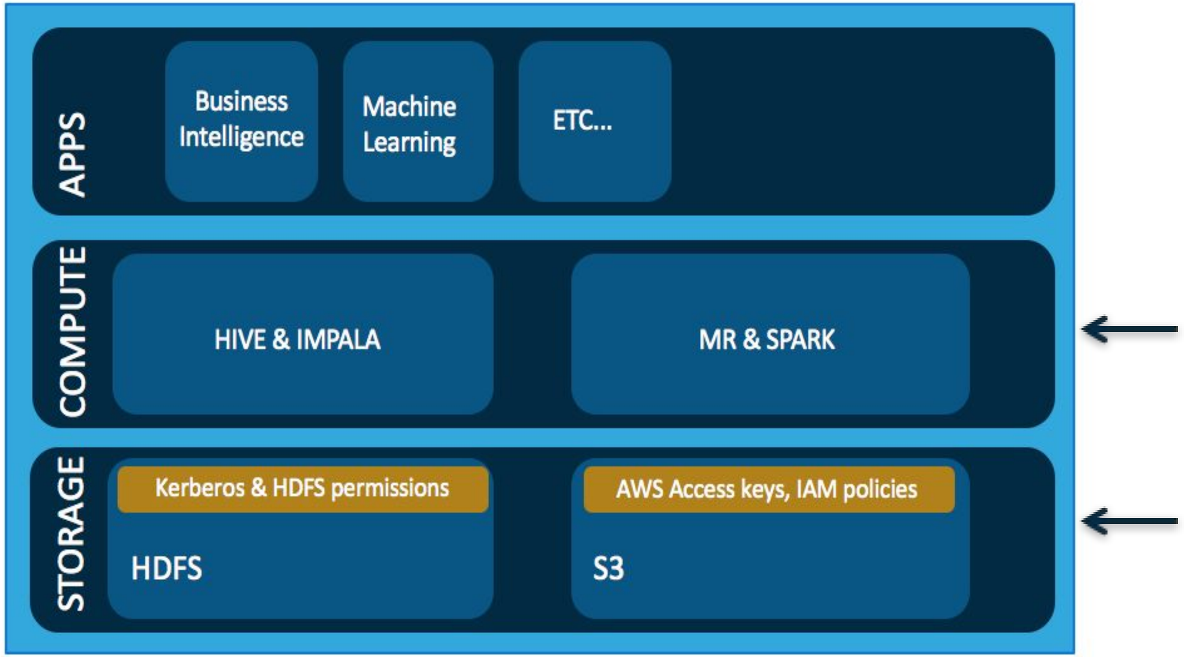
Challenges

"I have tables in S3 and user permissions assigned to files with IAM policy. However, the tables contain records from different, separately licensed sources. Only certain user groups are allowed to see certain records. "

How can I enforce this?

"In addition, we currently have multiple computing applications running on top of the same data, such as Spark, Hive, etc. " **How can we enforce the same access control policy?**

Challenges



How to enforce authorization for structured data?

Access control for unstructured data, plus allow compute services above to access all structured data.

cloudera

Challenges

“In our cloud scenarios, we have multiple storage engines involved - HDFS and S3. “

How can we be isolated from needing to know where our data is actually stored?

Demand for Fine-grained Authorization

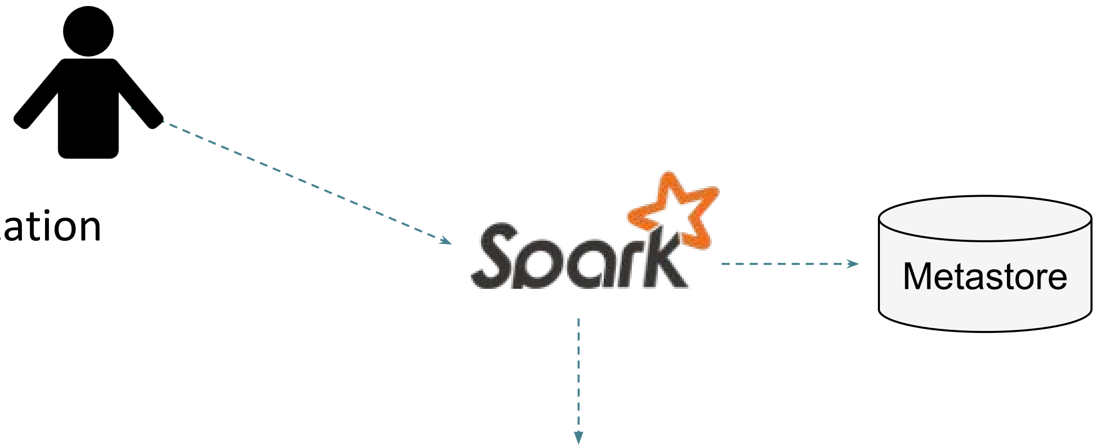
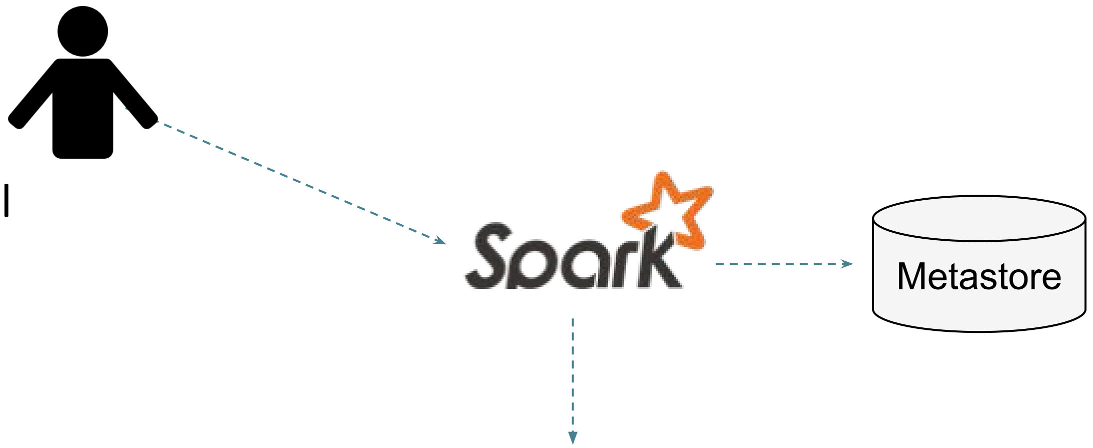


Table level authorization

Date/time	Account #	SSN	Asset	Trade	Country
11:33:01 16-Feb-2015	3947848494	329-44-9847	TBT	Buy	EU
09:33:11 16-Feb-2015	0234837823	238-23-9876	AZP	Sell	US
14:12:34 16-Feb-2015	4848367383	123-56-2345	IDI	Sell	EU

cloudera

Demand for Fine-grained Authorization

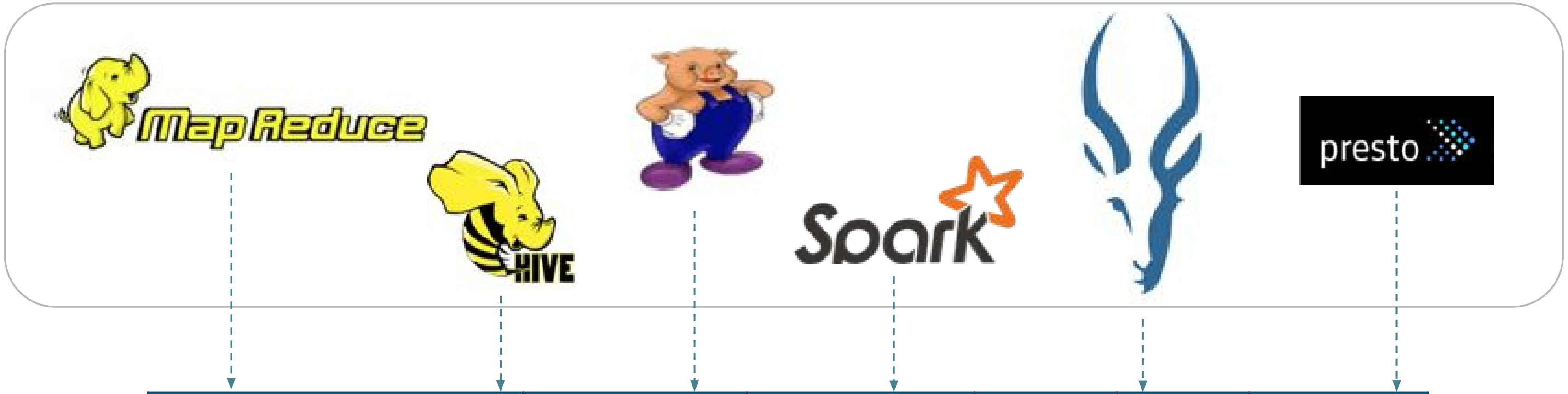


Column or row level authorization

Date/time	Account #	SSN	Asset	Trade	Country
11:33:01 16-Feb-2015	494	9847	TBT	Buy	EU
14:12:34 16-Feb-2015	383	2345	IDI	Sell	EU

cloudera

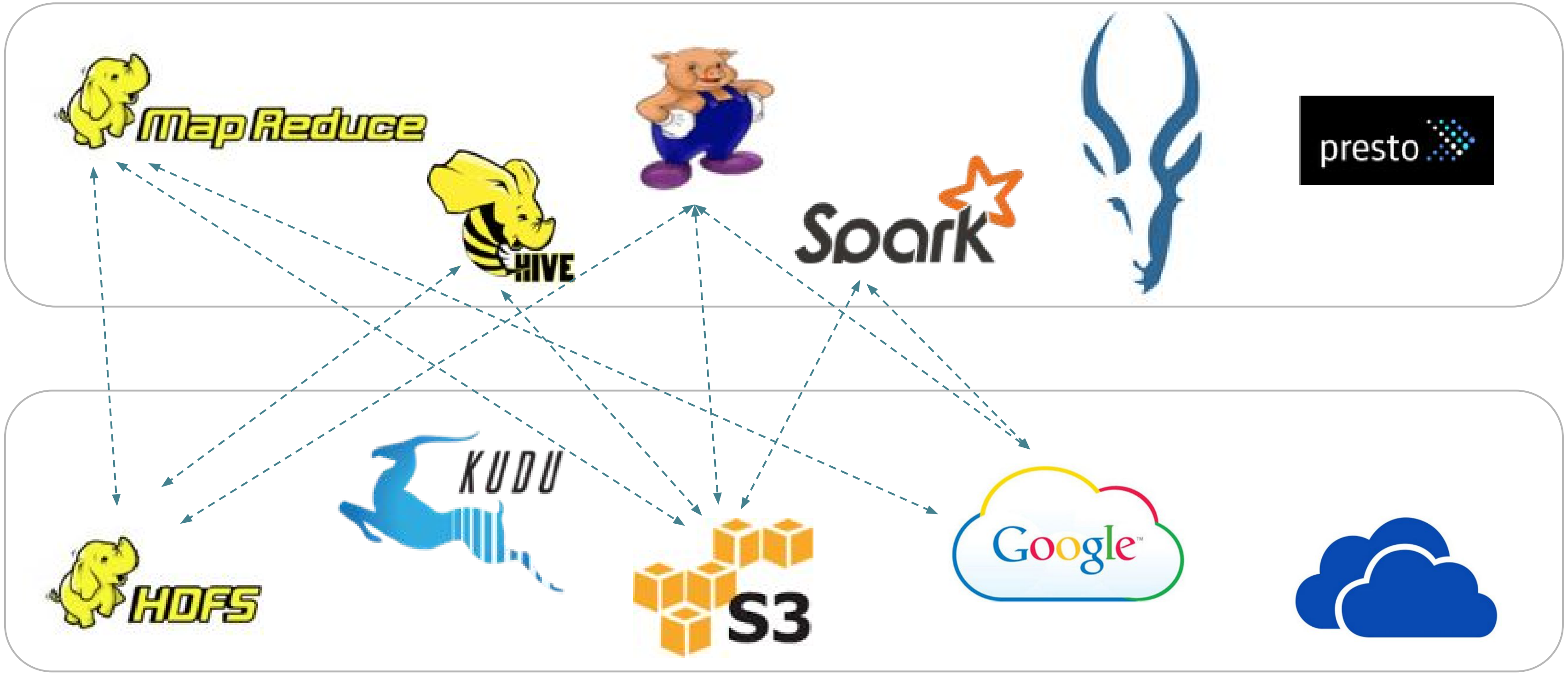
Demand for Unified Authorization Enforcement



Date/time	Account #	SSN	Asset	Trade	Country
11:33:01 16-Feb-2015	[redacted]494	[redacted]9847	TBT	Buy	EU
[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
14:12:34 16-Feb-2015	[redacted]383	[redacted]2345	IDI	Sell	EU

cloudera

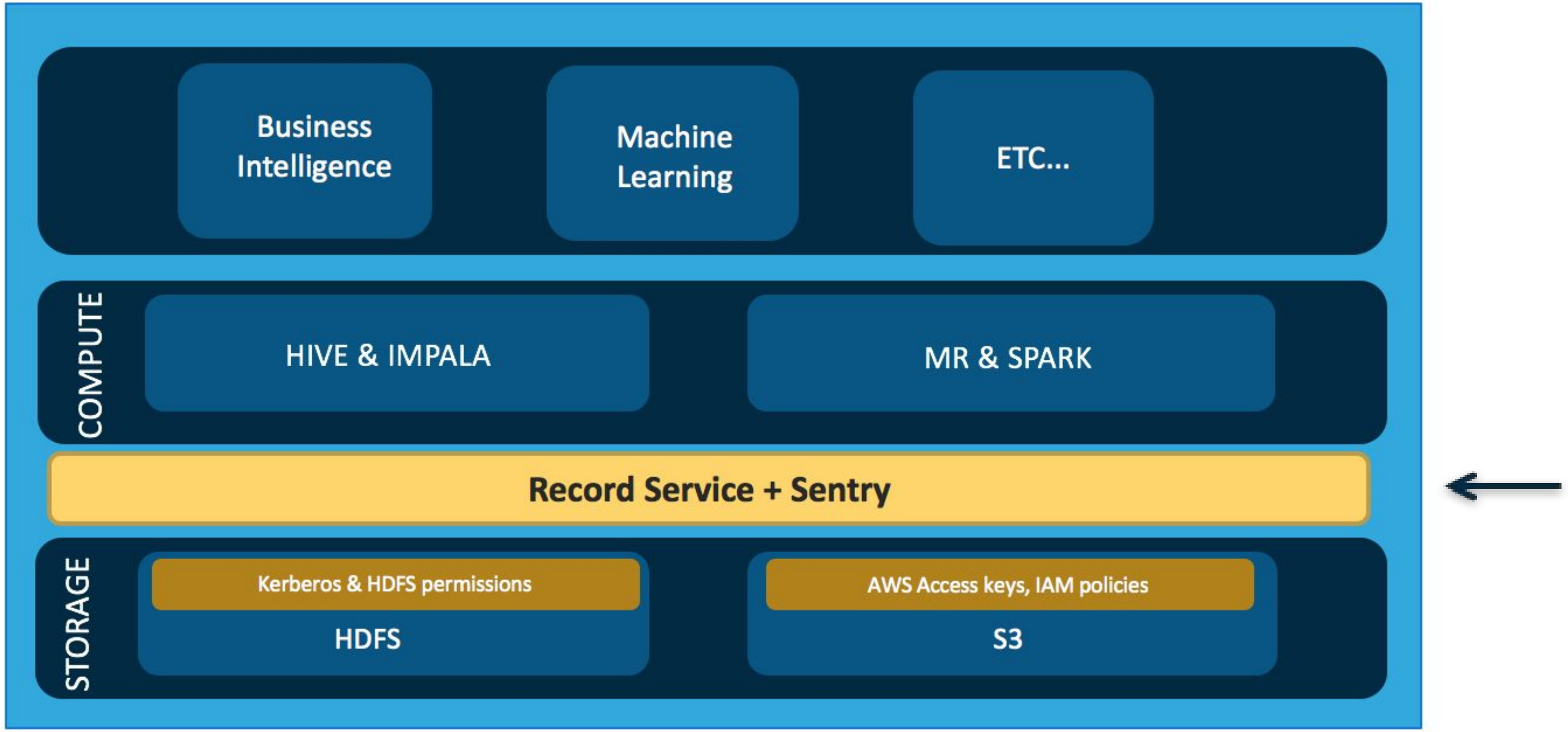
Demand for Storage and Compute layer Isolation



Solution: Apache Sentry + RecordService

cloudera

Solution


Business Intelligence
Machine Learning
ETC...
COMPUTE
HIVE & IMPALA
MR & SPARK
Record Service + Sentry
STORAGE
Kerberos & HDFS permissions
HDFS
AWS Access keys, IAM policies
S3

Apache Sentry

Authorization Service

- provides the ability to enforce role-based access control (RBAC) to data and/or metadata for authenticated users in a fine-grained manner.
- Enterprise grade big data security.
- Provides unified policy management.
- Pluggable and highly modular.

Apache Sentry

Work out of the box with Apache Hive, Hive metastore/HCataglog, Apache Solr, Apache Kafka, Apache Sqoop and Apache Impala

Apache Sentry

- Actors
  - User
  - User group membership
  - Role
  - Resources
  - Privilege

Apache Sentry

- User
  - Authenticated user
  - User identity obtained from session context

- User group membership
  - Defined outside of sentry policy
  - Obtained from user directory (LDAP, AD)

Apache Sentry

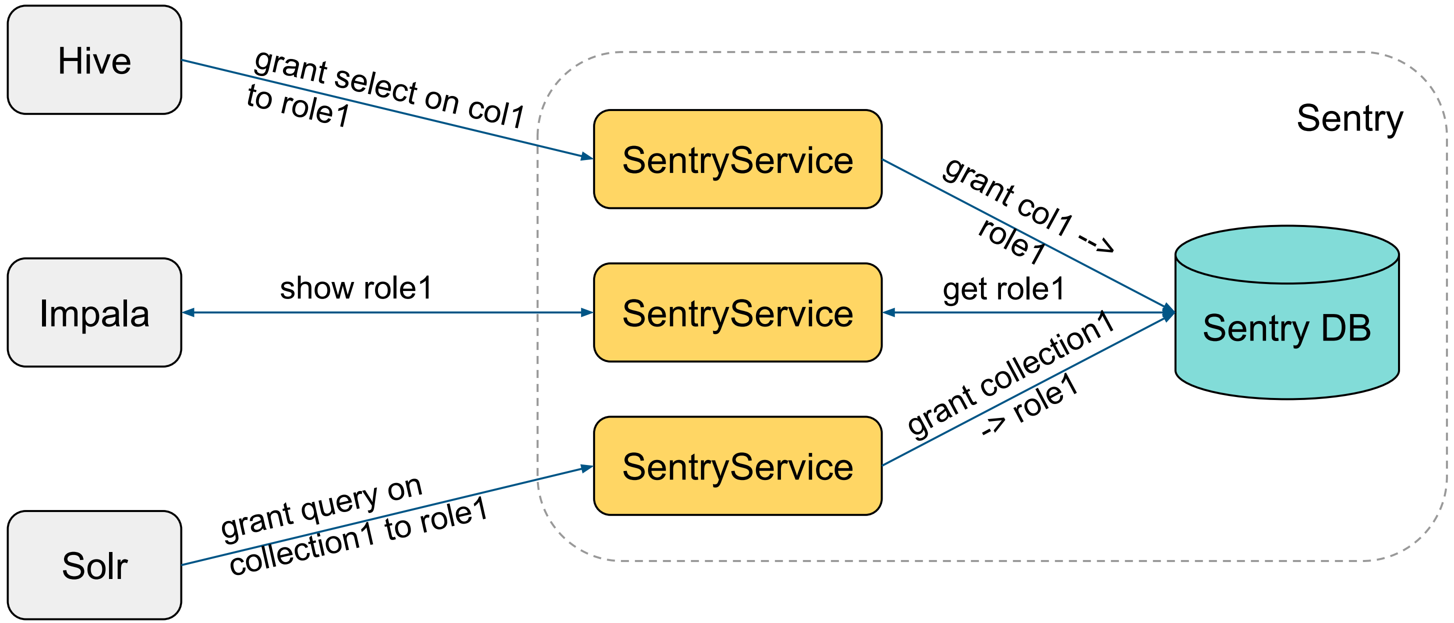
- Resources: is hierarchical
    - Data to be protected
    - Collection in Solr
    - Topic in Kafka
    - Columns in Hive
    - URI

Apache Sentry

- Privilege
  - Action or operation associated with a resource
    - SELECT on a given Column or Table
    - CREATE a TABLE or VIEW
    - QUERY on a SEARCH COLLECTION
    - Example: db=db1->table=t1->col=c1->action=SELECT
  - Assigned to a role

Apache Sentry

- Centralized Authorization Polices Store

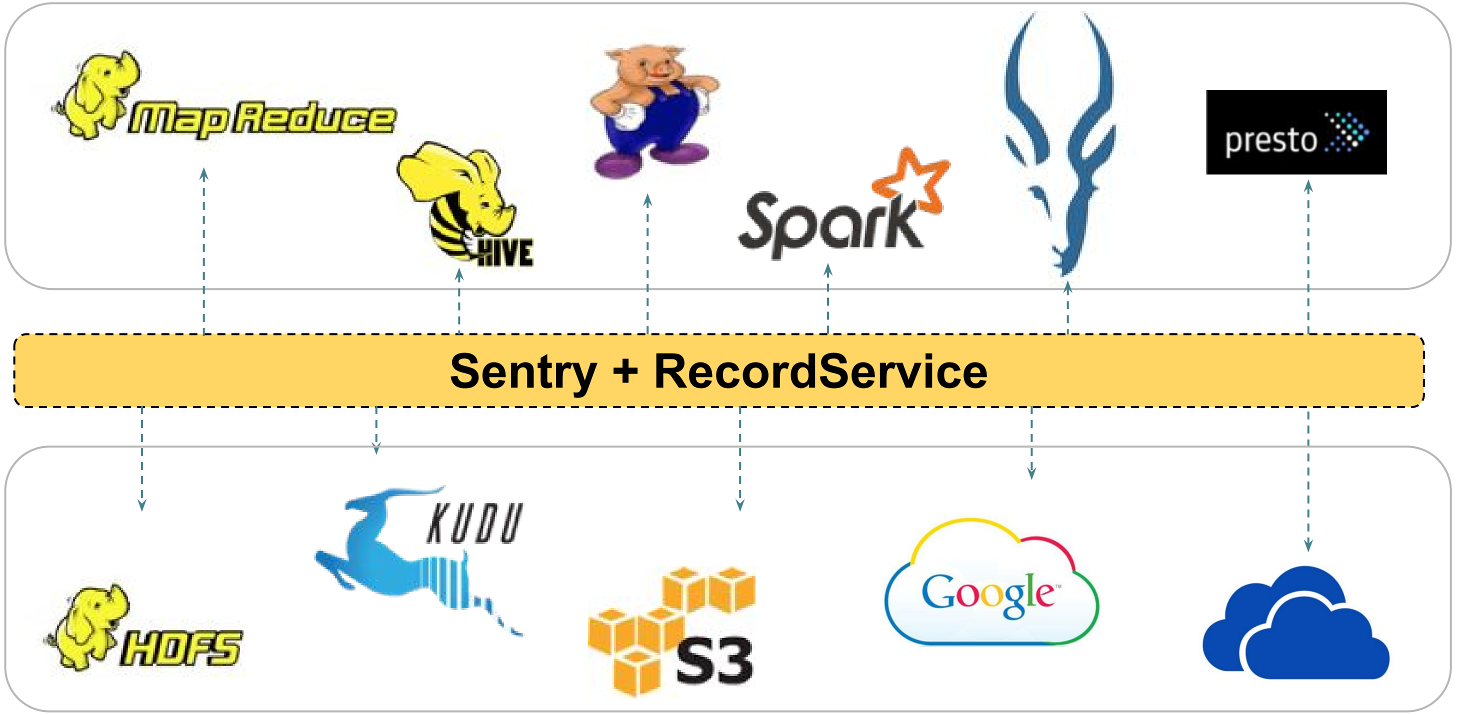


cloudera

Apache Sentry

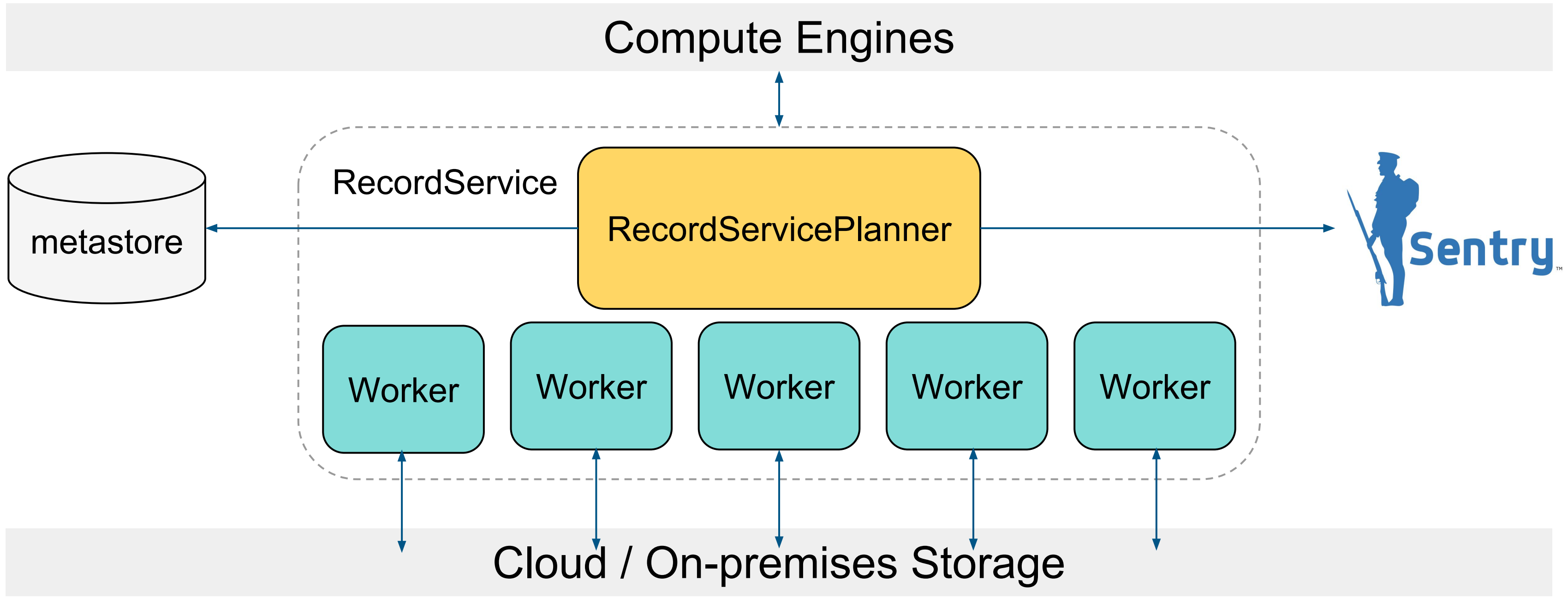
Sentry provides fine-grained authorization on **S3** as well as HDFS.

Apache Sentry + Record Service



cloudera

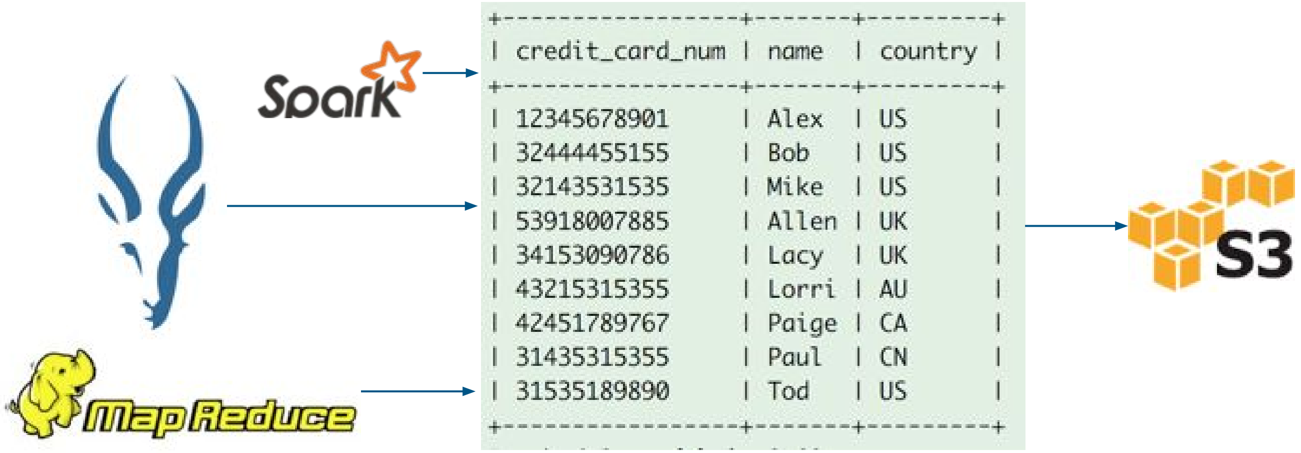
Apache Sentry + Record Service



Use Case

Fine-grained Authorization: Column + Row + Masking

- Given a table
  - CREATE TABLE s3_credit (credit_card_num STRING, name STRING, country STRING) STORED AS TEXTFILE LOCATION 's3a://recordservice-test-data/credit';
- Enforce same authorization policies on spark, impala and MR.

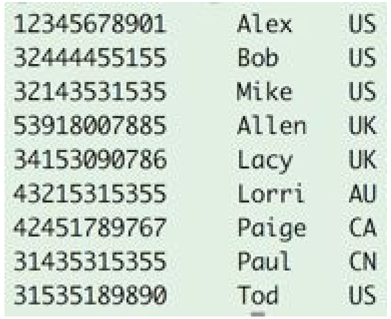


```
+-----------------+-------+---------+
| credit_card_num | name  | country |
+-----------------+-------+---------+
| 12345678901     | Alex  | US      |
| 32444455155     | Bob   | US      |
| 32143531535     | Mike  | US      |
| 53918007885     | Allen | UK      |
| 34153090786     | Lacy  | UK      |
| 43215315355     | Lorri | AU      |
| 42451789767     | Paige | CA      |
| 31435315355     | Paul  | CN      |
| 31535189890     | Tod   | US      |
+-----------------+-------+---------+
```

cloudera

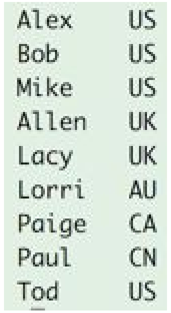
If Using Cloud Provider Security

Cloud providers only provide storage level permissions.



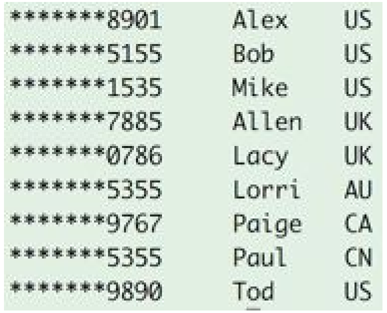
12345678901	Alex	US
32444455155	Bob	US
32143531535	Mike	US
53918007885	Allen	UK
34153090786	Lacy	UK
43215315355	Lorri	AU
42451789767	Paige	CA
31435315355	Paul	CN
31535189890	Tod	US

s3://user/**credit**



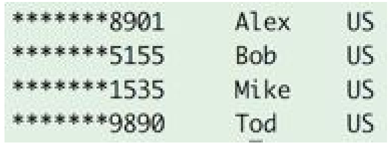
Alex	US
Bob	US
Mike	US
Allen	UK
Lacy	UK
Lorri	AU
Paige	CA
Paul	CN
Tod	US

s3://user/**credit_copy1**



*******8901	Alex	US
*******5155	Bob	US
*******1535	Mike	US
*******7885	Allen	UK
*******0786	Lacy	UK
*******5355	Lorri	AU
*******9767	Paige	CA
*******5355	Paul	CN
*******9890	Tod	US

s3://user/**credit_copy2**

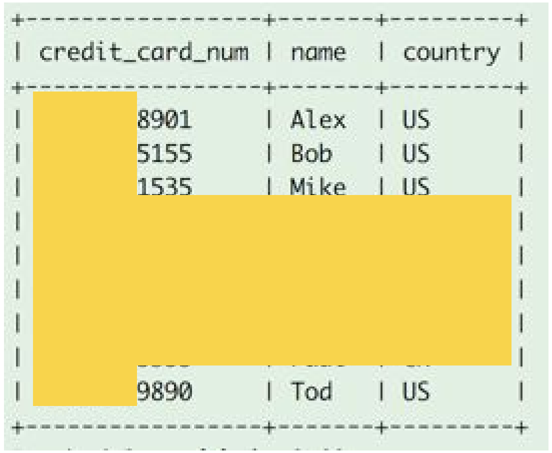


*******8901	Alex	US
*******5155	Bob	US
*******1535	Mike	US
*******9890	Tod	US

s3://user/**credit_copy3**

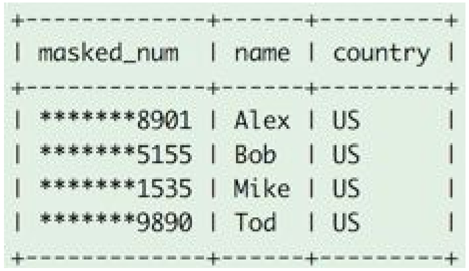
Using Sentry + RecordService

- Step 1:
  - Create Hive / Impala UDF: mask(String credit_card_num)
  - CREATE VIEW s3_credit_view AS SELECT mask(credit_card_num) masked_num, name name, country country FROM s3_credit where country='US';



```
+-----------------+-------+---------+
| credit_card_num | name  | country |
+-----------------+-------+---------+
| [redacted]8901            | Alex  | US      |
|     5155            | Bob   | US      |
|     1535            | Mike  | US      |
|
|
|
|
|
|     9890            | Tod   | US      |
+-----------------+-------+---------+
```

select * from s3_credit_view

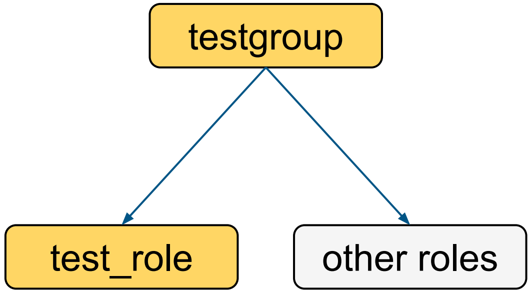


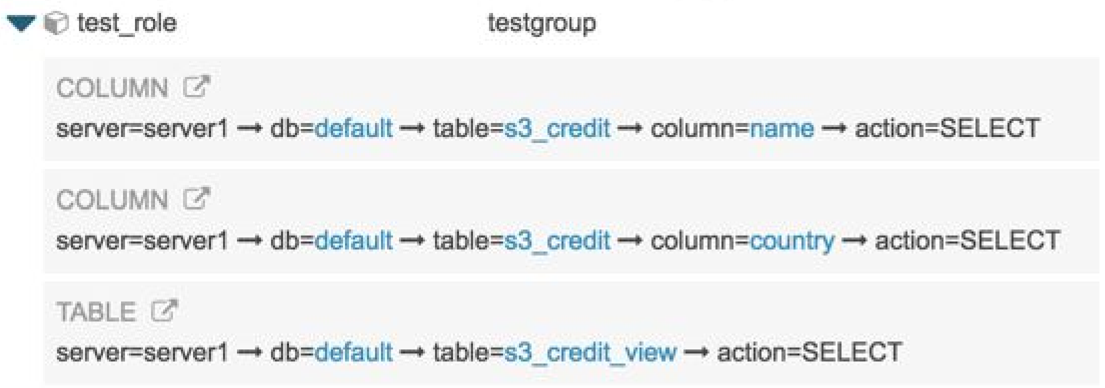
```
+-------------+------+---------+
| masked_num  | name | country |
+-------------+------+---------+
| *******8901 | Alex | US      |
| *******5155 | Bob  | US      |
| *******1535 | Mike | US      |
| *******9890 | Tod  | US      |
+-------------+------+---------+
```

cloudera

Using Sentry + RecordService

- Step 2: Grant the Sentry privileges
  - CREATE ROLE **test_role**;
  - GRANT SELECT (name, country) on TABLE s3_credit to **test_role**;
  - GRANT SELECT on TABLE s3_credit_view to **test_role**;
  - GRANT ROLE **test_role** to GROUP **testgroup**;




▼ test_role testgroup

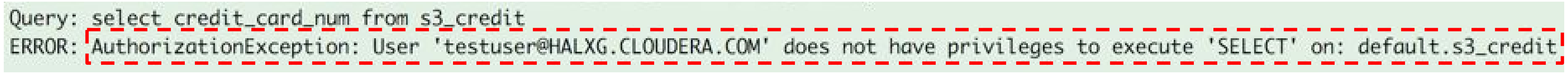
COLUMN
server=server1 → db=default → table=s3_credit → column=name → action=SELECT

COLUMN
server=server1 → db=default → table=s3_credit → column=country → action=SELECT

TABLE
server=server1 → db=default → table=s3_credit_view → action=SELECT

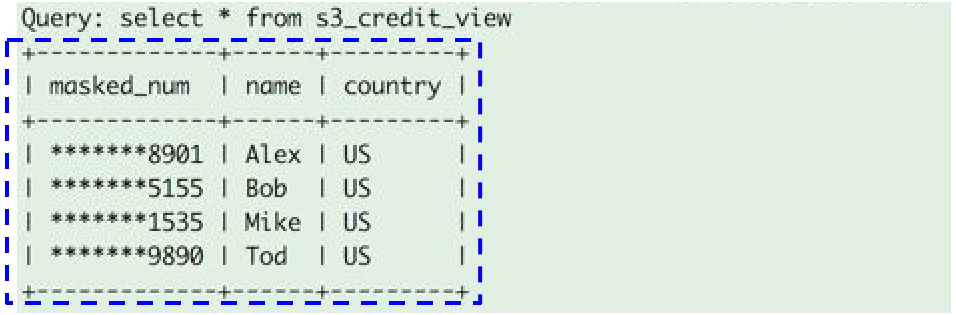
Impala

select **credit_card_num** from **s3_credit**;



```
Query: select credit_card_num from s3_credit
ERROR: AuthorizationException: User 'testuser@HALXG.CLOUDERA.COM' does not have privileges to execute 'SELECT' on: default.s3_credit
```

select * from **s3_credit_view**;

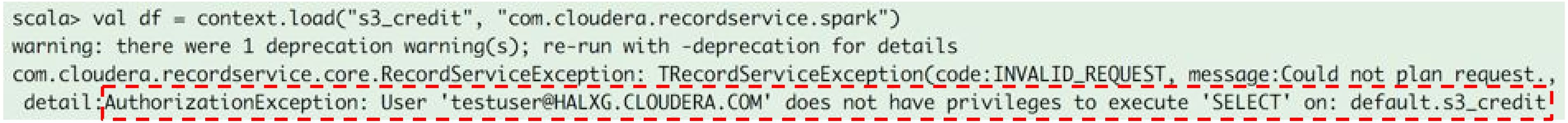


```
Query: select * from s3_credit_view
+-------------+------+---------+
| masked_num  | name | country |
+-------------+------+---------+
| *******8901 | Alex | US      |
| *******5155 | Bob  | US      |
| *******1535 | Mike | US      |
| *******9890 | Tod  | US      |
+-------------+------+---------+
```

Spark

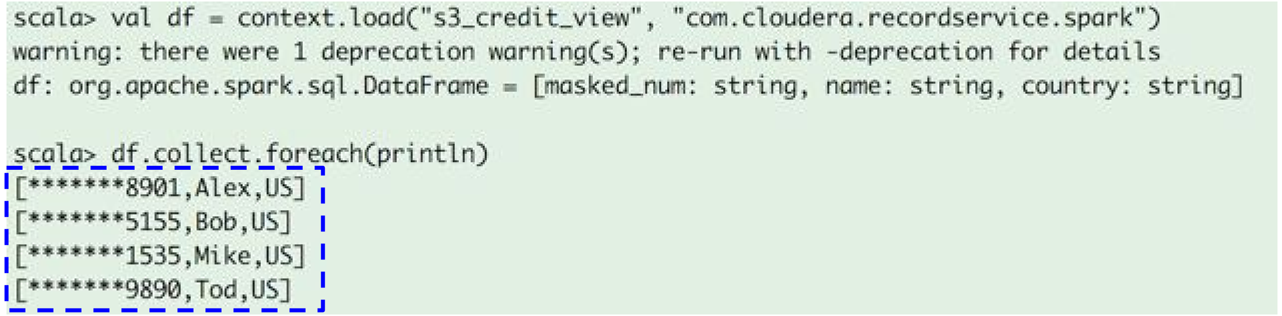
spark-shell --jars recordservice-spark-0.4.0-cdh5.8.x.jar

s3_credit



```
scala> val df = context.load("s3_credit", "com.cloudera.recordservice.spark")
warning: there were 1 deprecation warning(s); re-run with -deprecation for details
com.cloudera.recordservice.core.RecordServiceException: TRecordServiceException(code:INVALID_REQUEST, message:Could not plan request.,
 detail:AuthorizationException: User 'testuser@HALXG.CLOUDERA.COM' does not have privileges to execute 'SELECT' on: default.s3_credit
```

s3_credit_view

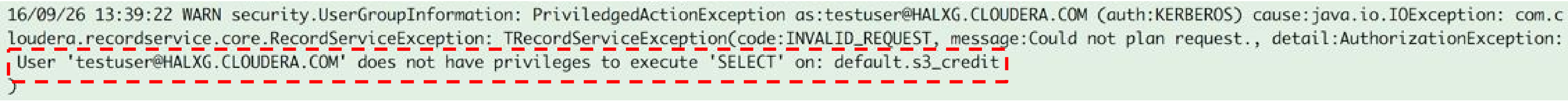


```
scala> val df = context.load("s3_credit_view", "com.cloudera.recordservice.spark")
warning: there were 1 deprecation warning(s); re-run with -deprecation for details
df: org.apache.spark.sql.DataFrame = [masked_num: string, name: string, country: string]

scala> df.collect.foreach(println)
[*******8901,Alex,US]
[*******5155,Bob,US]
[*******1535,Mike,US]
[*******9890,Tod,US]
```

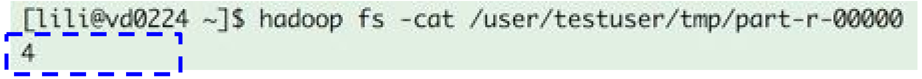
MapReduce

hadoop jar recordservice-examples-0.4.0-cdh5.8.x.jar \
com.cloudera.recordservice.examples.mapreduce.RecordCount \
"select credit_card_num from **s3_credit**" "/user/testuser/tmp"



```
16/09/26 13:39:22 WARN security.UserGroupInformation: PriviledgedActionException as:testuser@HALXG.CLOUDERA.COM (auth:KERBEROS) cause:java.io.IOException: com.c
loudera.recordservice.core.RecordServiceException: TRecordServiceException(code:INVALID_REQUEST, message:Could not plan request., detail:AuthorizationException:
 User 'testuser@HALXG.CLOUDERA.COM' does not have privileges to execute 'SELECT' on: default.s3_credit
)
```

hadoop jar recordservice-examples-0.4.0-cdh5.8.x.jar \
com.cloudera.recordservice.examples.mapreduce.RecordCount \
"select * from **s3_credit_view**" "/user/testuser/tmp"



```
[lili@vd0224 ~]$ hadoop fs -cat /user/testuser/tmp/part-r-00000
4
```

cloudera

Demo

Project Status

cloudera

Project Status

Apache Sentry

- Graduated from Incubation – a top-level Apache project
- Hundreds of Cloudera customers using it

RecordService

- Open source project, and released up to Beta 0.3.0.
- Apache 2.0 Licensed
- Intent to donate to Apache Software Foundation

How to contribute?

- Mailing list:
  - recordservice-user@googlegroups.com
  - dev-subscribe@sentry.apache.org
- Contributions:
  - http://github.com/cloudera/RecordServiceClient/
  - https://cwiki.apache.org/confluence/display/SENTRY/Home
- Documentation:
  - http://recordservice.io/
  - https://cwiki.apache.org/confluence/display/SENTRY/Documentation

cloudera

Q & A

Meet us @ Booth #721

cloudera