

Security/Governance/GDPR Demo on CDP PvC Base 7.x

Summary

How to quickly setup Cloudera Security/Governance/GDPR (Worldwide Bank) demo using Cloudera Data Platform Data Center (CDP PvC Base). It can be deployed either on AWS using AMI or on your own setup via provided script

Recording

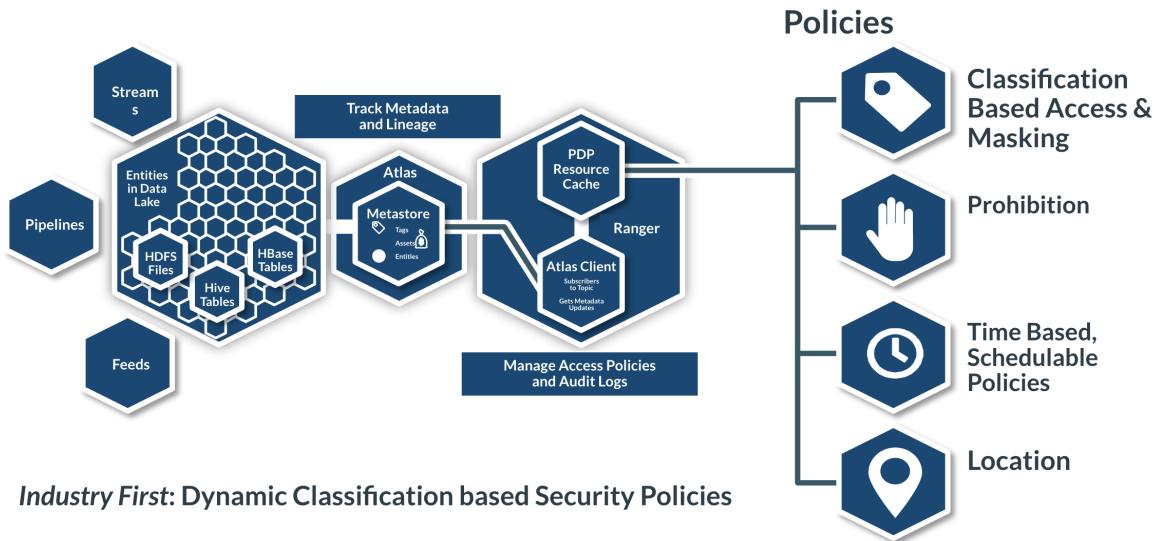
[Cloudera Connect Showcase - CDP Demo on Security and Governance](#)

What's included

- Single node CDP 7.1.4 including:
 - Cloudera Manager (60 day trial license included) - for managing the services
 - Kerberos - for authentication (via local MIT KDC)
 - Ranger - for authorization (via both resource/tag based policies for access and masking)
 - Atlas - for governance (classification/lineage/search)
 - Zeppelin - for running/visualizing Hive queries
 - Impala/Hive 3 - for Sql access and ACID capabilities
 - Spark/HiveWarehouseConnector - for running secure SparkSQL queries
- Worldwide Bank artifacts
 - Demo hive tables
 - Demo tags/attributes and lineage in Atlas
 - Demo Zeppelin notebooks to walk through demo scenario
 - Ranger policies across HDFS, Hive/Impala, Hbase, Kafka, SparkSQL to showcase:
 - Tag based policies across HDP components
 - Row level filtering in Hive columns
 - Dynamic tag based masking in Hive/Impala columns
 - Hive UDF execution authorization
 - Atlas capabilities like
 - Classifications (tags) and attributes
 - Tag propagation
 - Data lineage

- Business glossary: categories and terms
 - GDPR Scenarios around consent and data erasure via Hive ACID
- Hive ACID / MERGE labs

CDP – SECURITY & GOVERNANCE



Option 1: Steps to deploy on your own setup

- Launch a vanilla Centos 7 VM and set up a single node CDP cluster using this [Github](#) but instead of "base" CM template choose the "wwbank_krb.json" template:

```

yum install -y git
#setup KDC
curl -sSL https://gist.github.com/abajwa-hw/bca3d23fe146c3ebd59a9b5fd19480a3/raw | sudo -E sh

git clone https://github.com/fabiog1901/SingleNodeCDPCluster.git
cd SingleNodeCDPCluster
./setup_krb.sh gcp templates/wwbank_krb.json

#Setup worldwide bank demo using script
curl -sSL https://raw.githubusercontent.com/abajwa-hw/masterclass/master/ranger-atlas/setup-dc-703.sh | sudo -E
bash

```

Once the script completes, you will need to restart Zeppelin once (via CM) for it to pick up the demo notebooks

Option 2: Steps to launch prebuilt AMI on AWS

- 1. Login into the AWS EC2 console using your credentials
- 2. Select the AMI from 'N. California' region by clicking one the links below:
 - CDP 7.1.4 [here](#) (public)

Now choose instance type: select 'm4.4xlarge' and click Next

Note: if you choose a smaller instance type from the above recommendation, not all services may come up

The screenshot shows the 'Step 2: Choose an Instance Type' page. It lists several instance types under 'General purpose': m4.large, m4.xlarge, m4.2xlarge, m4.4xlarge (selected), m4.10xlarge, and m4.16xlarge. Each row includes columns for CPU, Memory, EBS support, and network performance.

Instance Type	CPU	Memory	EBS	Network
m4.large	2	8	EBS only	Yes Moderate
m4.xlarge	4	16	EBS only	Yes High
m4.2xlarge	8	32	EBS only	Yes High
m4.4xlarge	16	64	EBS only	Yes High
m4.10xlarge	40	160	EBS only	Yes 10 Gigabit
m4.16xlarge	64	256	EBS only	Yes 25 Gigabit

- 3.. Configure Instance Details: make sure "Auto-assign Public IP" is enabled and click 'Next'

The screenshot shows the 'Step 3: Configure Instance Details' page. It allows setting the number of instances (1), selecting a purchasing option (Request Spot instances), choosing a network (vpc-6fce50d | launchpad-test), a subnet (subnet-b6f41dd3 | jcustenborder-launchpad | us-west-2), and enabling auto-assign public IP.

Number of instances: 1

Purchasing option: Request Spot instances

Network: vpc-6fce50d | launchpad-test

Subnet: subnet-b6f41dd3 | jcustenborder-launchpad | us-west-2

Auto-assign Public IP: Enable

- 4. Add storage: use at least 100 GB and click 'Next'

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0d70977c47ec4cbd4	300	General Purpose SSD (gp2)	900 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

- 5. Add tags needed to prevent instances from being terminated. Then click 'Next'

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value	Instances	Volumes
enddate	01192020	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
owner	abajwa	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
project	RangerAtlas demo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Name	abajwa-cdp-worldwidebank	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

- 6. Configure security group: create a new security group and select 'All traffic' and open all ports to **only your IP**. Then click 'Review and Launch'

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security group name: launch-wizard-79
Description: launch-wizard-79 created 2020-01-12T22:51:05.535-08:00

Type	Protocol	Port Range	Source	Description
All traffic	All	0 - 65535	Custom 73.71.28.30/32	e.g. SSH for Admin Desktop

[Add Rule](#)

[Cancel](#) [Previous](#) [Review and Launch](#)

[Feedback](#) [English \(US\)](#)

- 7. Review your settings and click Launch

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, launch-wizard-1, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

⚠ Your instance configuration is not eligible for the free usage tier

To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about [free usage tier](#) eligibility and usage restrictions.

[Don't show me this again](#)

AMI Details [Edit AMI](#)

HDP 2.5 Demo kit cluster - 12/13/2016 - ami-ec65338c

Hortonworks HDP 2.5 single node running NiFi/Sentiment demo, IoT trucking demo and Zeppelin demos like AON earthquake. The admin users password for Ambari and Zeppelin is your AWS account number.Built 12/13/2016 using HDP-2.5.3.0-37. Doc at bit.ly/2fUMWMZ

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
m4.2xlarge	26	8	32	EBS only	Yes	High

Security Groups [Edit security groups](#)

[Cancel](#) [Previous](#) [Launch](#)

[Feedback](#) [English](#)

- 8. Create and download a new key pair (or choose an existing one). Then click 'Launch instances'

Select an existing key pair or create a new key pair

X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name
demokitkey

[Download Key Pair](#)

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location**. You will not be able to download the file again after it's created.

Cancel [Launch Instances](#)

- 9. Click the shown link under 'Your instances are now launching'

Services ▾ Resource Groups ▾ Ali Bajwa ▾ N. California ▾ Support ▾

Launch Status

Your instances are now launching

The following instance launches have been initiated: I-0916f66788a77b682 [View launch log](#)

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups

[View Instances](#)

- 10. This opens the EC2 dashboard that shows the details of your launched instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Target Groups, and Auto Scaling. The main content area shows a table of instances. One row is selected for the instance with the ID i-149733ab. The instance details are displayed in a modal window. The modal has tabs for Description, Status Checks, Monitoring, and Tags. The Description tab shows the following details:

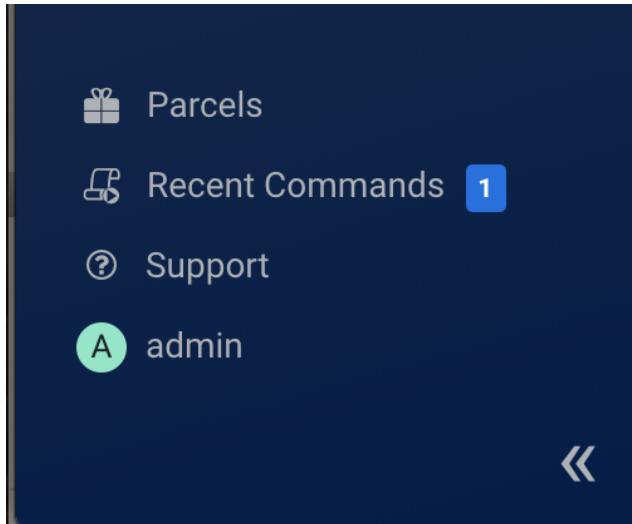
Instance ID	i-149733ab	Public DNS	ec2-54-193-109-204.us-west-1.compute.amazonaws.com
Instance state	running	Public IP	54.193.109.204
Instance type	m4.2xlarge	Elastic IPs	us-west-1b
Private DNS	ip-172-31-22-107.us-west-1.compute.internal	Security groups	all open . view rules
Private IPs	172.31.22.107	Scheduled events	No scheduled events
Secondary private IPs		AMI ID	HDP 2.5 Demo cluster - 11/08/2016 (ami-cde1abab)
VPC ID	vpc-e737dc82	Key pair	[REDACTED]
Subnet ID	subnet-5901fb3c	Owner	[REDACTED]
Network interfaces	eth0	Launch time	November 8, 2016 at 7:28:42 AM UTC-8 (2 hours)
Source/dest. check	True	Termination protection	False
EBS-optimized	True	Lifecycle	normal
Root device type	ebs	Monitoring	basic
Root device	/dev/sda1		
Block devices	/dev/sda1		

A tooltip for the 'Owner' field says: "The AWS account number of the AMI owner, without dashes."

- 11. Make note of your instance's 'Public IP' (which will be used to access your cluster). If the 'Public IP' is blank, wait 1-2 minutes for this to be populated
- 12. After 5-10 minutes, open the below URL in your browser to access Cloudera Manager (CM) console: <http://<PUBLIC IP>:7180>. Login as admin/admin

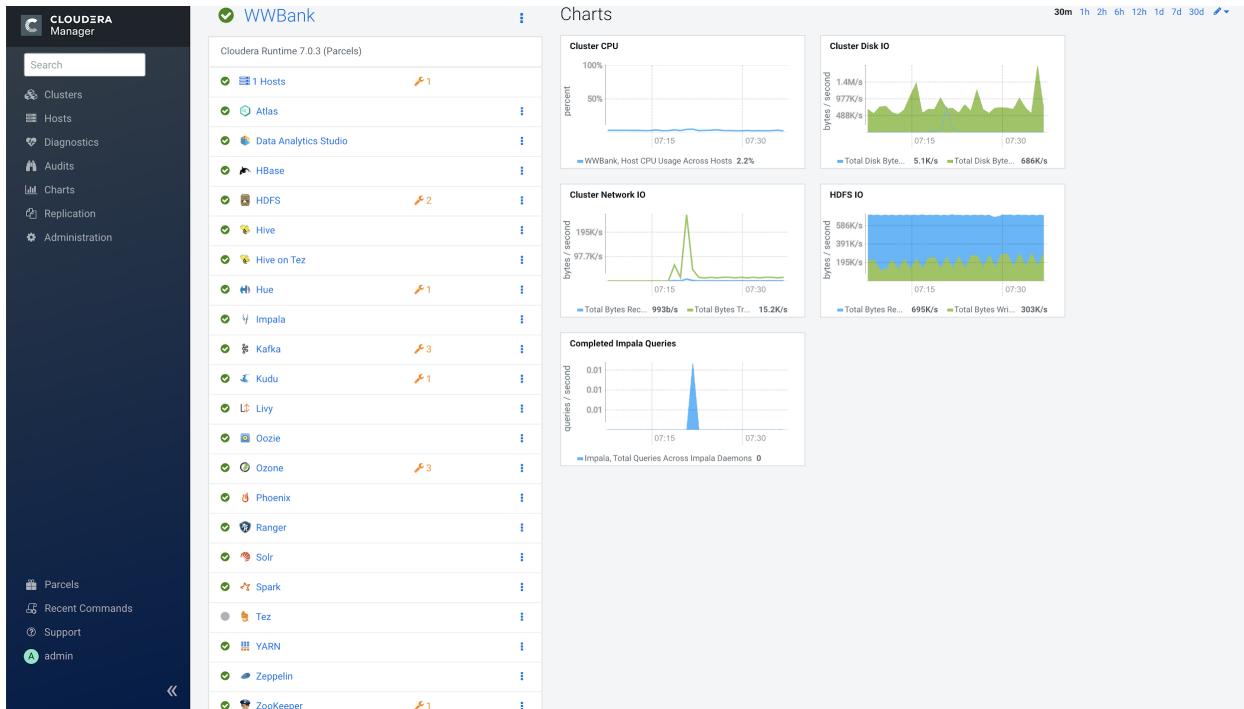
The screenshot shows the Cloudera Manager login interface. It features a dark header bar with the Cloudera Manager logo and a light-colored footer bar with links for Support Portal and Help. The main area contains a login form with fields for 'admin' in the username field and '*****' in the password field. There is also a 'Remember me' checkbox and a 'Sign in' button.

- 13. At this point, CM may still be in the process of starting all the services. You can tell by the presence of the blue operation notification near the bottom left of the page. If so, just wait until it is done.



(Optional) You can also monitor the startup using the log as below:

- Open SSH session into the VM using your key and the public IP e.g. from OSX:
`ssh -i ~/.ssh/mykey.pem centos@<publicIP>`
- Tail the startup log:
`tail -f /var/log/cdp_startup.log`
- Once you see "cluster is ready!" you can proceed
- 14. Once the blue operation notification disappears and all the services show a green check mark, the cluster is fully up.



If any services fail to start, use the hamburger icon next to SingleNodeCluster > Start button to start

Accessing cluster resources

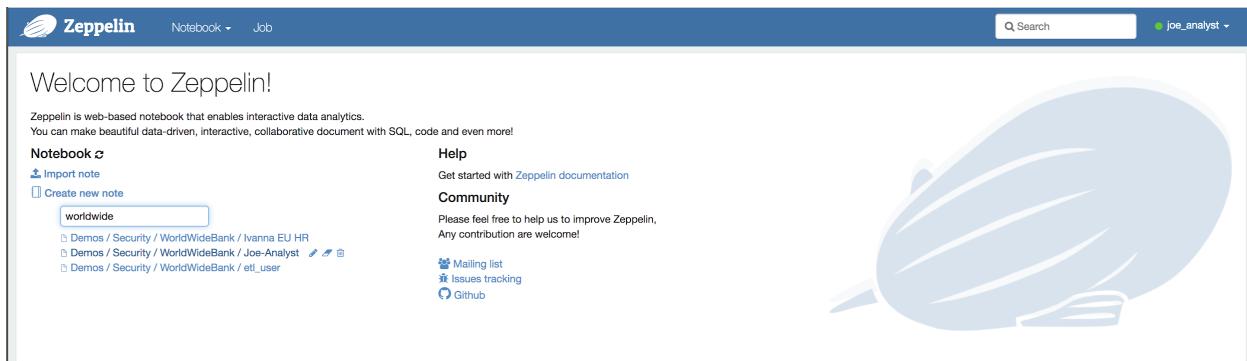
CDP urls

- Access CM at :7180 as admin/admin
- Access Ranger at :6080. Ranger login is admin/BadPass#1
- Access Atlas at :31000. Atlas login is admin/BadPass#1
- Access Zeppelin at :8885. Zeppelin users logins are:
 - joe_analyst = BadPass#1
 - ivanna_eu_hr = BadPass#1
 - etl_user = BadPass#1
- Access Hue at :8889. Hue users logins are:
 - joe_analyst = BadPass#1
 - ivanna_eu_hr = BadPass#1
 - etl_user = BadPass#1

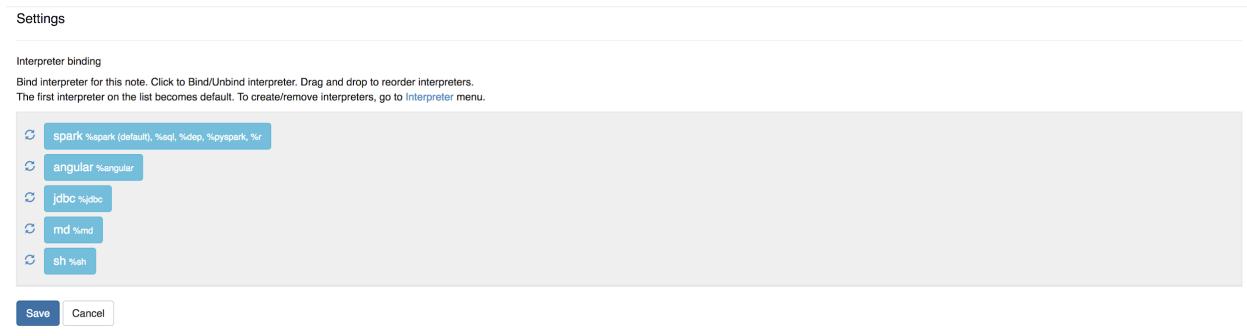
Demo walkthrough

Run queries as joe_analyst

1. Open Zeppelin and login as joe_analyst. Find his notebook by searching for "worldwide" using the text field under 'Notebook' section. Select the notebook called: "Worldwide Bank - Joe Analyst"



2. On the first launch of the notebook, you will be prompted to choose interpreters. You can keep the defaults but make sure you click Save button:



3. Run through the notebook. This notebook shows

- a) MRN/password masked via tag policy. Here is the Ranger policy that enables this:

tb) dynamic column level masking

Address, nationalID, credit card number are masked via Hive column policies specified in Ranger. Notice that Birthday and age columns are masked using custom mask

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
69	mask : nationalid show last 4	--	Enabled	Enabled	--	analyst	--	
70	mask: ccn show first 4	--	Enabled	Enabled	--	analyst	--	
71	mask: hash password	--	Disabled	Enabled	--	analyst	--	
72	mask: redact street address	--	Enabled	Enabled	--	analyst	--	
73	custom mask: randomize age	--	Enabled	Enabled	--	analyst	--	
74	custom mask: retain birth year	--	Enabled	Enabled	--	analyst	--	

b) It also shows prohibition policy where zipcode, insuranceid and bloodtype can not be combined in a query

Policy Details :

- Policy Type: Access
- Policy ID: 66
- Policy Name: prohibit zipcode, insuranceid, bl
- Policy Label: Policy Label
- database: worldwidebank (Include)
- table: ww_customers (Include)
- column: zipcode (Include)
- Description:
- Audit Logging: YES

Allow Conditions :

Deny All Other Accesses: False

Deny Conditions :

Select Role	Select Group	Select User
Select Roles	analyst	Select Users

Policy Conditions

No Conditions

add/edit conditions

Resources Accessed Together?:

- x worldwidebank.ww_customers.insuranceid
- x worldwidebank.ww_customers.bloodtype

Resources Not Accessed Together?:

Conditions Permissions Delegate Admin

resources-accessed-together : worldwidebank.ww_customers.insuranceid, worldwidebank.ww_customers.bloodtype

c) also shows tag based policies:

Attempts to access any object tagged with EXPIRES_ON accessed after expiry date, will be denied. As we will show later, the fed_tax column of tax_2015 table is tagged in Atlas as EXPIRED_ON with expiry date of 2016 so it should not be allowed to be queried

Policy Details :

- Policy Type: Access
- Policy ID: 94
- Policy Name: access: EXPIRES_ON
- Policy Label: Policy Label
- TAG: EXPIRES_ON
- Description: Policy for data with EXPIRES_ON tag
- Audit Logging: YES

Allow Conditions :

Deny All Other Accesses: False

Deny Conditions :

Select Role	Select Group	Select User
Select Roles	public	Select Users

Policy Conditions

No Conditions

add/edit conditions

Accessed after expiry_date (yes/no)?: yes

Enter boolean expression:

Please enter condition..

Syntax check

Component Permissions

accessed-after-expiry : yes

Also attempts to access object tagged with PII will be denied as per policy, only HR allowed. As we will show later, the ssn column of tax_2015 table is tagged as PII in Atlas.

Policy Details :

- Policy Type: Access
- Policy ID: 96
- Policy Name *: access: PII
- Policy Label: Policy Label
- TAG *: x_PII
- Description: Restrict access to Personally Identifiable information
- Audit Logging: YES

Allow Conditions :

Select Role	Select Group	Select User	Policy Conditions	Component Permissions
Select Roles	x_hr x_ed x_dpo	Select Users	Add Conditions + expression : JavaScript Condition	HDFS HBASE HIVE
Select Roles	x_csr	Select Users		HDFS HBASE HIVE

Deny All Other Accesses : False

Deny Conditions :

Select Role	Select Group	Select User	Policy Conditions	Component Permissions
Select Roles	x_contractor	Select Users	expression : JavaScript Condition	HDFS HBASE HIVE
Select Roles	x_public	Select Users	Add Conditions +	HDFS HBASE HIVE

Attempts to access `cost_savings.claim_savings` table as analyst will fail because there is a policy that minimum of 60% data quality score is required for analysts. As we will see, this table is tagged in Atlas as having score of 51%

Policy Details :

- Policy Type: Access
- Policy ID: 95
- Policy Name *: access: DATA_QUALITY
- Policy Label: Policy Label
- TAG *: x_DATA_QUALITY
- Description: Prevent analyst from accessing data with low data-quality score
- Audit Logging: YES

Allow Conditions :

Deny All Other Accesses : False

Deny Conditions :

Select Role	Select Group	Select User	Policy Conditions	Component Permissions
Select Roles	x_analyst	Select Users	expression : JavaScript Condition	HIVE

add/edit conditions

Accessed after expiry_date (yes/no)?:

Enter boolean expression :

```
tagAttr.score < 0.6
```

Syntax check

The same queries can also be run via SparkSQL via spark-shell (as described above).
Sample queries for joe_analyst:

```
hive.execute("SELECT surname, streetaddress, country, age, password, nationalid, ccnumber, mrn, birthday FROM worldwidebank.us_customers").show(10)

hive.execute("select zipcode, insuranceid, bloodtype from worldwidebank.ww_customers").show(10)

hive.execute("select * from cost_savings.claim_savings").show(10)
```

```
Welcome to
      _/\_ 
     / \ \_ 
    /   \_ 
   /     \_ 
  /       \_ 
 /         \_ 
/           \_ 
version 2.4.0.7.0.0.0-462

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_222)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import com.hortonworks.hwc.HiveWarehouseSession
import com.hortonworks.hwc.HiveWarehouseSession

scala> import com.hortonworks.hwc.HiveWarehouseSession._
import com.hortonworks.hwc.HiveWarehouseSession._

scala> val hive = HiveWarehouseSession.session(spark).build()
hive: com.hortonworks.spark.sql.hive.llap.HiveWarehouseSessionImpl = com.hortonworks.spark.sql.hive.llap.HiveWarehouseSessionImpl@2ee4c028

scala>

scala> hive.execute("SELECT surname, streetaddress, country, age, password, nationalid, ccnumber, mrn, birthday FROM worldwidebank.us_customers").show(10)
+-----+-----+-----+-----+-----+-----+-----+
| surname| streetaddress|country|age|password|nationalid|ccnumber|mrn|birthday|
+-----+-----+-----+-----+-----+-----+-----+
| Powersl| nnnn XXXXXX XXXXX| US1 54176a3fe33eb676cb12...|lxxx-xx-301615.49xxxx-xx|null|01-01-1973|
| Whitmanl| nnnn XXXXXX XXXXX| US1 5916103bc00fc877e5c...|lxxx-xx-825514.55xxxx-xx|null|01-01-1977|
| Maronel| nnnn XXXXX XXXXX| US1 441d01407d5922624b1...|lxxx-xx-614214.53xxxx-xx|null|01-01-1977|
| Harpl| nnnn XXX XXXX XXXX| US1 30171051c13b6bdd179...|lxxx-xx-401815.20xxxx-xx|null|01-01-1997|
| Pereiral| nnnn XXXXXXXX XXXXXX| US1 93147d2be1c1c114669...|lxxx-xx-192715.38xxxx-xx|null|01-01-2038|
| Blackburnl| nnnn XXXXXX XXXXX| US1 591127473e0e3d200b19...|lxxx-xx-790315.53xxxx-xx|null|01-01-1965|
| Gonzalezl| nnnn XXXXX XXXXX| US1 2814082d6873762e0f3b...|lxxx-xx-258214.71xxxx-xx|null|01-01-1995|
| Heltonl| nnnn XXXXXX XXXXX| US1 5815d5e10b606eb7599b...|lxxx-xx-429115.25xxxx-xx|null|01-01-1975|
| Meil| nnnn XXXXX XXXXXX| US1 77179027eba8d2ae53c9...|lxxx-xx-836615.20xxxx-xx|null|01-01-1941|
| Goldbergl| nnnn XXXXX XXXXX| US1 811478f719c117d97df4...|lxxx-xx-493714.55xxxx-xx|null|01-01-1944|
+-----+-----+-----+-----+-----+-----+-----+
```

4. Confirm using Ranger audits that the queries ran as joe_analyst. Also notice that column names, masking types, IPs and policy IDs were captured. Also notice tags (e.g. DATA_QUALITY or PII) are captured along with their attributes. Also notice that these audits were captured for operations across Hive, Hbase, Kafka and HDFS

Ranger Access Manager Audit Security Zone Settings admin

User: joe_analyst

Exclude Service Users : [] Entries: 1 to 25 of 26 Last Updated Time: 01/08/2020 01:07:19 PM

Policy ID	Policy Version	Event Time *	Application	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Agent Host Name	Client IP	Cluster Name	Zone Name	Event Count	Tags
100	1	01/07/2020 07:15:21 PM	hdfs	joe_analyst	cm_hdfs hdfs	/sensitive/private.csv path	EXECUTE	Denied	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		1	SENSITIVE
100	1	01/07/2020 07:15:10 PM	kafka	joe_analyst	cm_kafka kafka	PRIVATE topic	describe	Denied	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		1	SENSITIVE
54	1	01/07/2020 07:15:08 PM	kafka	joe_analyst	cm_kafka kafka	FOREX topic	consume	Allowed	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		2	-
54	1	01/07/2020 07:15:08 PM	kafka	joe_analyst	cm_kafka kafka	FOREX topic	describe	Allowed	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		2	-
54	1	01/07/2020 07:15:05 PM	kafka	joe_analyst	cm_kafka kafka	FOREX topic	describe	Allowed	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		1	-
100	1	01/07/2020 07:14:25 PM	hbaseRegional	joe_analyst	cm_hbase hbase	t_private/cf2 column-family	scannerOpen	Denied	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		1	SENSITIVE
40	1	01/07/2020 07:14:15 PM	hbaseRegional	joe_analyst	cm_hbase hbase	t_forex/cf2 column-family	scannerOpen	Allowed	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		2	-
40	1	01/07/2020 07:14:15 PM	hbaseRegional	joe_analyst	cm_hbase hbase	t_forex/cf1 column-family	scannerOpen	Allowed	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		2	-
95	1	01/07/2020 07:12:27 PM	hiveServer2	joe_analyst	cm_hive hive	cost_savings/claim_saving... @column	SELECT	Denied	ranger-acl	cdp.cloudera.com	172.31.1.131	SingleNodeCluster		1	DATA_QUALITY

Run queries as ivanna_eu_hr

5. Once services are up, open Ranger UI and also login to Zeppelin as ivanna_eu_hr and find her notebook by searching for "hortonia" using the text field under the 'Notebook' section. Select the notebook called: "Worldwide Bank - Ivana EU HR"

Zeppelin Notebook Job Search ivanna_eu_hr

Welcome to Zeppelin!

Zeppelin is web-based notebook that enables interactive data analytics. You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!

Notebook [Import note](#) [Create new note](#)

worldwide

- Demos / Security / WorldwideBank / Ivanna EU HR
- Demos / Security / WorldwideBank / Joe-Analyst
- Demos / Security / WorldwideBank / etl_user

Help Get started with [Zeppelin documentation](#)

Community Please feel free to help us to improve Zeppelin. Any contribution are welcome!

Mailing list Issues tracking Github



6. On first launch of the notebook, you may be prompted to choose interpreters. You can keep the defaults but make sure you click Save button:

Settings

Interpreter binding

Bind interpreter for this note. Click to Bind/Unbind interpreter. Drag and drop to reorder interpreters. The first interpreter on the list becomes default. To create/remove interpreters, go to Interpreter menu.

- spark %spark (default), %sql, %dep, %pyspark, %
- angular %angular
- jdbc %jdbc
- md %md
- sh %sh

Save **Cancel**

7. Run through the notebook cells using Play button in top right of each cell (or Shift-Enter)

Zeppelin Notebook - Job

Search your Notes ivanna_eu_hr

Demos / Security / HortoniaBank / HortoniaBank...

Accessing US Customers fails with an Access Control violation

```
%jdbc(hive)
select * from hortonibank.us_customers limit 10
```

ERROR

org.apache.hive.service.cli.HiveSQLException: Error while compiling statement: FAILED: HiveAccessControlException Permission denied: user [ivanna_eu_hr] does not have [SELECT] privilege on [hortonibank/us_customers/*]

at org.apache.hive.service.cli.HiveSQLException.(HiveSQLException.java:277)
at org.apache.hive.jdbc.Utils.verifySuccess(Utils.java:263)
at org.apache.hive.jdbc.HiveStatement.runAsyncOnServer(HiveStatement.java:303)
at org.apache.hive.jdbc.HiveStatement.executeQuery(HiveStatement.java:296)
at org.apache.commons.dbcp2.DelegatingStatement.executeQuery(DelegatingStatement.java:291)
at org.apache.commons.dbcp2.DelegatingStatement.executeDelegatingStatement(DelegatingStatement.java:201)
at org.apache.zepplin.jdbc.JDBCInterpreter.executeSql(JDBCInterpreter.java:682)
at org.apache.zepplin.interpreter.interpret JDBCInterpreter.java:763
at org.apache.zepplin.interpreter.interpret LazyOpenInterpreter.interpret(LazyOpenInterpreter.java:101)
at org.apache.zepplin.interpreter.remote.RemoteInterpreterServer\$InterpretJob.jobRun(RemoteInterpreterServer.java:502)
at org.apache.zepplin.scheduler.Job.run(Job.java:175)
at org.apache.zepplin.scheduler.ParallelScheduler\$JobRunner.run(ParallelScheduler.java:162)
at java.util.concurrent.Executors\$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTask.access\$201(ScheduledThreadPoolExecutor.java:180)
at java.util.concurrent.ScheduledThreadPoolExecutor\$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:293)

Took 0 sec. Last updated by ivanna_eu_hr at November 29 2017, 7:35:52 PM.

Customers by country - can only see EU customer data

```
%jdbc(hive)
select * from
(select countryfull, count(*) num_customers from hortonibank.ww_customers
group by countryfull) s
order by num_customers desc
limit 5
```

FINISHED

Italy Spain Belgium Denmark Cyprus (Anglicized)

Took 7 sec. Last updated by ivanna_eu_hr at November 29 2017, 7:35:51 PM.

Row Level Security - Customer data filtered to EU persons only based on location

```
%jdbc(hive)
select distinct(country) from hortonibank.ww_customers
```

country

FR
HU
IT
NL
PL

FINISHED

This notebook highlights:

a) Row level filtering: as Ivana can only see data for European customers who have given consent (even though she is querying `ww_customers` table which contains both US and EU customers). Below is the Ranger hive policy that enables this feature:

The screenshot shows the Ranger Access Manager interface under the 'Edit Policy' tab for a 'Row Level Filter' policy. The policy details are as follows:

- Policy Type:** Row Level Filter
- Policy ID:** 75
- Policy Name:** filter: ww_customers for consent
- Policy Label:** Policy Label
- Hive Database:** worldwidebank
- Hive Table:** ww_customers
- Description:** (empty)
- Audit Logging:** YES

On the right, there is a 'Policy Conditions' section with a 'No Conditions' message and a 'Add Validity Period' button.

Below the main form is a 'Row Filter Conditions' table:

Select Role	Select Group	Select User	Policy Conditions	Access Types	Row Level Filter
Select Roles	x us_employee	Select Users	Add Conditions +	select	country in (US)
Select Roles	x eu_employee	x admin	Add Conditions +	select	country in (select countrycode from worldwidebank.eu_countries) and insuranceid in (select insuranceid from consent_master.consent_data)
Select Roles	x etl x dpo	Select Users	Add Conditions +	select	country in (select countrycode from worldwidebank.eu_countries)

b) it also shows that since Ivana is part of HR group, there are no policies that limit her access: so she can see raw passwords, nationalIDs, credit card numbers, MRN #, birthdays etc

c) The last cells show that tag based policies d

3. Once you successfully run the notebook, you can open the Ranger Audits to show the policies and that the queries ran as her and that row filtering occurred (notice ROW_FILTER access type):

Ranger Access Manager Audit Security Zone Settings admin

User: ivanna_eu_hr

Exclude Service Users: []

Entries: 1 to 25 of 28 | Last Updated Time: 01/12/2020 11:16:27 PM

Policy ID	Policy Version	Event Time *	Application	User	Service	Resource	Name / Type	Access Type	Result	Access Enforcer	Agent Host Name	Client IP	Cluster Name	Zone Name	Event Count	Tags
100	1	01/12/2020 11:16:24 PM	hdfs	ivanna_eu_hr	cm_hdfs	/sensitive/private.csv	path	READ	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	SENSITIVE
100	1	01/12/2020 11:16:01 PM	kafka	ivanna_eu_hr	cm_kafka	PRIVATE topic		consume	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		2	SENSITIVE
100	1	01/12/2020 11:16:00 PM	kafka	ivanna_eu_hr	cm_kafka	PRIVATE topic		describe	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		2	SENSITIVE
40	1	01/12/2020 11:15:58 PM	hbaseRegional	ivanna_eu_hr	cm_hbase	L_forex/cf1 column-family	hbase	scannerOpen	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		2	--
40	1	01/12/2020 11:15:58 PM	hbaseRegional	ivanna_eu_hr	cm_hbase	T_forex/cf2 column-family	hbase	scannerOpen	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		2	--
100	1	01/12/2020 11:15:57 PM	kafka	ivanna_eu_hr	cm_kafka	PRIVATE topic		describe	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	SENSITIVE
54	1	01/12/2020 11:15:56 PM	kafka	ivanna_eu_hr	cm_kafka	FOREX topic		consume	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		2	--
54	1	01/12/2020 11:15:56 PM	kafka	ivanna_eu_hr	cm_kafka	FOREX topic		describe	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		2	--
54	1	01/12/2020 11:15:53 PM	kafka	ivanna_eu_hr	cm_kafka	FOREX topic		describe	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	--
81	1	01/12/2020 11:15:14 PM	hiveServer2	ivanna_eu_hr	cm_hive	hr/employees_encrypted/_@column	hive	SELECT	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	ENCRYPTED, ENCRYPTED
95	1	01/12/2020 11:15:03 PM	hiveServer2	ivanna_eu_hr	cm_hive	cost_savings/claim_savin... @column	hive	SELECT	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	DATA_QUALITY
99	1	01/12/2020 11:14:57 PM	hiveServer2	ivanna_eu_hr	cm_hive	consent_master/consent_@column	hive	UPDATE	Denied	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	ATTRIBUTE_DETAILS
64	1	01/12/2020 11:14:51 PM	hiveServer2	ivanna_eu_hr	cm_hive	worldwidebank/ww_custo... @column	hive	SELECT	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	REFERENCE_DATA
82	1	01/12/2020 11:14:51 PM	hiveServer2	ivanna_eu_hr	cm_hive	consent_master/consent_@column	hive	SELECT	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	REFERENCE_DATA
javascript:void(0)		01/12/2020 11:14:51 PM	hiveServer2	ivanna_eu_hr	cm_hive	worldwidebank/ew_countri... @column	hive	SELECT	Allowed	ranger-acl	cdp.cloudera.com	172.31.7.61	SingleNodeCluster		1	REFERENCE_DATA

Run queries as etl_user

8. Similarly, you can login to Zeppelin as etl_user and run his notebook as well

Zeppelin Notebook Job Search etl_user

Welcome to Zeppelin!

Zeppelin is web-based notebook that enables interactive data analytics. You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!

Notebook worldwide

Help Get started with [Zeppelin documentation](#)

Community Please feel free to help us to improve Zeppelin, Any contribution are welcome!

Mailing list [Issues tracking](#) Github



This notebook shows how an admin would handle GDPR scenarios like below using Hive ACID capabilities

- when a customer withdraws consent (so they no longer appear in searches)
- when a customer requests their data to be erased

Run Hive/Impala queries from Hue

Alternatively you can login to Hue as joe_analyst and select Query > Editor > Hive and click "Saved queries" to run Joe's sample queries via Hive:

The screenshot shows the Hue interface with the 'Query' menu open, highlighting the 'Hive' option. The main pane displays a table of saved queries, with one entry named 'Joe analyst queries' selected. The right sidebar shows a table schema for 'worldwidebank.us_customers'.

Name	Description	Owner	Last Modified
Joe analyst queries	Worldwide bank demo	joe_analyst	01/22/2020 11:31 AM -08:00

The screenshot shows the Hue interface with the 'Query' menu open, highlighting the 'Editor' option. The main pane displays the query code for 'Joe analyst queries'. The right sidebar shows a detailed table schema for 'worldwidebank.us_customers'.

```
1 -- Joe analyst queries
2
3 -- Dynamic Column Masking: MRN/password cols masked via classification policy. Others masked via Hive col policy. Custom masks
4 SELECT sumname, streetaddress, country, age, nationalid, cnumber, mmn, birthday FROM worldwidebank.us_customers LIMIT 10
5
6 -- Prohibition policy: Prevent toxic joins (prevent join of Zipcode, InsuranceId, Blood group)
7 select zipcode, insuranceid, bloodtype from worldwidebank.wv_customers limit 10
8
9 -- Prohibition policy - dropping insuranceid allows query to run
10 select zipcode, bloodtype from worldwidebank.wv_customers limit 10
11
12 -- Leased Data Asset: Lifecycle controlled by Classification based policy (fed_tax is tagged with EXPIRED_ON which is restrictive)
13 select fed_tax from finance.tax_2015
14
15
16 -- Analyst prohibited from accessing personal data through Data Classification based policy (SSN column is tagged with PII which is restrictive)
17 select ssn from finance.tax_2015
18
19
20 -- Querying for columns other than fed_tax/ssn works
21 select state_tax from finance.tax_2015
22
23
24 -- Data Quality annotation based policy: Don't waste time on poor quality datasets! (Analysts should not access table tagged with cost_savings)
25 select * from cost_savings.claim_savings limit 5
26
27
28 -- Decrypt UDF: US employee sees decrypted versions of email and phone number
29 select * from hr.employees_encrypted
30
31
```

Name	Description	Owner	Last Modified
Joe analyst queries	Worldwide bank demo	joe_analyst	01/22/2020 11:31 AM -08:00

You can also switch the editor to Impala to run Joe's sample queries via Impala to show tag based access policy working for Impala:

In 7.1.x, Impala also supports column based masking

surname	streetaddress	country	age	nationalid	ccnnumber	mrn	birthday
1 Powers	nnnn XXXXXXXX XXXXX	US	52	xxx-xx-3016	5.49xxxx+xx	NULL	NULL
2 Whitman	nnnn XXXXXXXX XXXXX	US	55	xxx-xx-8255	4.55xxxx+xx	NULL	NULL
3 Marone	nnn XXXXX XXXXX	US	47	xxx-xx-6142	4.53xxxx+xx	NULL	NULL
4 Harp	nnnn Xxx XXXX XXXXX	US	26	xxx-xx-4018	5.20xxxx+xx	NULL	NULL
5 Pereira	nnnn XXXXXXXX XXXXX	US	80	xxx-xx-1927	5.38xxxx+xx	NULL	NULL

Alternatively you can login to Hue as ivanna_eu_hr and click "Saved queries" to run Ivanna's sample queries via Hive:

```

1 -- Ivanna EU HR queries
2 -- EU employee can not access us_customers table
3
4 select * from worldwidebank.us_customers limit 10
5
6 -- Row Level Security - Customer data filtered to EU persons only based on location
7 select distinct(country) from worldwidebank.wm_customers
8
9
10
11 -- HR analyst can see unmasked records - but only for EU customers who have given consent
12 SELECT surname, streetaddress, country, age, nationalid, cccnumber, mrn, birthday FROM worldwidebank.wm_customers LIMIT 10
13
14
15 -- Analysts only see portion of customers - only those who have given consent. (Table actually has ~29k rows)
16 SELECT count(*) FROM worldwidebank.wm_customers
17
18 -- Analyst CAN'T see a customer who has not given consent (Row filtering)
19 SELECT insuranceid, surname, streetaddress, country, age FROM worldwidebank.wm_customers where insuranceid='23182722'
20
21 -- Analyst CAN see a customer who has given consent (Row Filtering)
22 SELECT insuranceid, surname, streetaddress, country, age FROM worldwidebank.wm_customers where insuranceid='62517316'
23
24 -- HR Analyst not be able to update consent master data tagged as REFERENCE_DATA (Tag based policy)
25 update consent_master.consent_data_trans set loyaltyconsent='NO' where insuranceid='57155949'
26
27 -- HR analyst can access table tagged with DATA_QUALITY even though it's score < 60% (joe_analyst can not access)
28 select * from cost_savings.claim_savings limit 5
29
30 -- Decrypt UUID: EU employee only see AES-256 encrypted versions of email and phone number
31 select * from hr.employees_encrypted
32

```

Query History Saved Queries

Name	Description	Owner	Last Modified
Ivanna EU HR queries	Worldwide bank demo	ivanna_eu_hr	01/22/2020 11:32 AM -08:00

Run SparkSQL queries via HWC

To run secure SparkSQL queries (using Hive Warehouse Connector)

- connect to instance via SSH using your keypair
- authenticate as the user you want to run queries as via keytabs
 - kinit -kt /etc/security/keytabs/joe_analyst.keytab joe_analyst/\$(hostname -f)@CLOUDERA.COM
- start SparkSql using HiveWarehouseConnector

```

spark-shell --jars
  /opt/cloudera/parcels/CDH/jars/hive-warehouse-connector-assembly*.jar
  --conf spark.sql.hive.hiveserver2.jdbc.url="jdbc:hive2://$(hostname
  -f):10000/default;"      --conf
  "spark.sql.hive.hiveserver2.jdbc.url.principal=hive/$(hostname
  -f)@CLOUDERA.COM"        --conf
  spark.security.credentials.hiveserver2.enabled=false

```

• import HWC classes and start session

- import com.hortonworks.hwc.HiveWarehouseSession
- import com.hortonworks.hwc.HiveWarehouseSession._
- val hive = HiveWarehouseSession.session(spark).build()

• run queries using hive.execute() e.g.

- hive.execute("select * from cost_savings.claim_savings").show(10)

- Sample script to automate above for joe_analyst here:
 - `/tmp/masterclass/ranger-atlas/HortonianMunichSetup/run_spark_sql.sh`

Troubleshooting Zeppelin

In case you encounter Thrift Exception below, it's likely the session was expired:

Prohibition policy - dropping insuranceid allows query to run

```
%jdbc(hive)
select zipcode, bloodtype from worldwidebank.ww_customers
limit 10

java.sql.SQLException: org.apache.hive.org.apache.thrift.TException: Error in calling method ExecuteStatement
    at org.apache.hive.jdbc.HiveStatement.runAsyncOnServer(HiveStatement.java:334)
    at org.apache.hive.jdbc.HiveStatement.execute(HiveStatement.java:265)
    at org.apache.commons.dbcp2.DelegatingStatement.execute(DelegatingStatement.java:291)
    at org.apache.commons.dbcp2.DelegatingStatement.execute(DelegatingStatement.java:291)
    at org.apache.zeppelin.jdbc.JDBCInterpreter.executeSql(JDBCInterpreter.java:718)
    at org.apache.zeppelin.jdbc.JDBCInterpreter.interpret(JDBCInterpreter.java:801)
    at org.apache.zeppelin.interpreter.LazyOpenInterpreter.interpret(LazyOpenInterpreter.java:103)
    at org.apache.zeppelin.interpreter.remote.RemoteInterpreterServer$InterpretJob.jobRun(RemoteInterpreterServer.java:633)
    at org.apache.zeppelin.scheduler.Job.run(Job.java:188)
    at org.apache.zeppelin.scheduler.ParallelScheduler$JobRunner.run(ParallelScheduler.java:162)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$201(ScheduledThreadPoolExecutor.java:180)
```

Just scroll to the top and click the gears icon (near top right) to display the interpreters and restart the jdbc one

The screenshot shows the Zeppelin settings interface. At the top, there are tabs for 'Notebook' and 'Job'. On the right, there is a search bar and a dropdown for 'joe_analyst'. Below the tabs, the URL is shown as "...os / Security / WorldWideBank / Joe-Analyst". The main area is titled 'Settings' and contains a section for 'Interpreter binding'. It says: 'Bind interpreter for this note. Click to Bind/Unbind interpreter. Drag and drop to reorder interpreters. The first interpreter on the list becomes default. To create/remove interpreters, go to Interpreter menu.' A list of interpreters is displayed with checkboxes:

- livy %livy (default), %sql, %pyspark, %spark, %shared (checkbox checked)
- md %md
- angular %angular
- jdbc %jdbc (checkbox checked)

 There is also a 'Restart' button next to the jdbc interpreter. At the bottom, there are 'Save' and 'Cancel' buttons.

Atlas walk through

9. Login to Atlas and show the Hive columns tagged as EXPIRES_ON

The screenshot shows the Apache Atlas search interface. On the left, there's a sidebar with search filters: 'Search By Type' (selected 'hive_column (293)'), 'Search By Classification' (selected 'EXPIRES_ON (1)'), and 'Search By Term' and 'Search By Text'. Below these are buttons for 'Clear' and 'Search'. A section for 'Favorite Searches' is shown with a note: 'You don't have any favorite search.' On the right, the main search results area has a header 'Results for: (Type: hive_column) AND (Classification: EXPIRES_ON)'. It displays one record: 'fed_tax' by 'etl_user' with type 'hive_column' and classification 'EXPIRES_ON'. There are checkboxes for 'Exclude sub-types', 'Exclude sub-classifications', and 'Show historical entities', along with a 'Columns' dropdown menu. A 'Page Limit' dropdown is set to 25.

To see the table name, you can select Table in the Column dropdown

This screenshot is identical to the first one, but the 'Columns' dropdown menu is open on the right side of the search results table. The menu lists various entity properties with checkboxes: 'Select', 'Name', 'Owner', 'Description', 'Type', 'Classifications', 'Term', 'Table', 'Comment', 'InputToProcesses', 'Meanings', 'OutputFromProcesses', 'Position', 'QualifiedName', 'ReplicatedFrom', and 'ReplicatedTo'. Most items have checkboxes checked.

Now notice the table name is also displayed

The screenshot shows the Apache Atlas search interface. On the left, there's a sidebar with search filters: 'Basic' selected, 'hive_column (293)', 'EXPIRES_ON (1)', and a search bar. The main area displays a table of results for 'hive_column' with 'EXPIRES_ON' classification. The table has columns: Name, Owner, Description, Type, Classifications, Term, and Table. One row is shown: 'fed_tax' (Owner: etl_user, Type: hive_column, Classification: EXPIRES_ON, Term: tax_2015). There are buttons for 'Exclude sub-types', 'Exclude sub-classifications', 'Show historical entities', and a 'Columns' dropdown. A page limit of 25 is set.

Selecting the fed_tax column and opening the 'Classifications' tab shows the attributes of tag (expiry_date) and value

This screenshot shows the 'Classifications' tab for the 'fed_tax' entity. It lists the classification 'EXPIRES_ON'. Under 'Attributes', it shows 'expiry_date' with a value of '2016-12-31T00:00:00.000Z'. There are edit and delete icons for this entry. The 'Properties', 'Lineage', and 'Relationships' tabs are also visible at the top.

To save this search, you can click the “Save As” button near bottom left:

The screenshot shows the Apache Atlas interface. A modal window titled "Create your favorite search" is open, containing a text input field with the value "hive cols tagged with EXPIRES_ON". Below the input field are two buttons: "Cancel" and "Create". The background shows the search results for "EXPIRES_ON" classification, listing one item: "EXPIRES_ON". The table has columns for Classification, Attributes, and Action. The "Attributes" row shows a single entry with Name: "expiry_date" and Value: "2016-12-31T00:00:00.000Z".

Similarly you can query for hive tables tagged with DATA_QUALITY...

The screenshot shows the Apache Atlas interface. A modal window titled "Search entities" is open, containing a search bar with the query "(Type: hive_table) AND (Classification: DATA_QUALITY)". Below the search bar, it says "Showing 2 records From 1 - 25". The main area displays a table with two rows. The columns are Name, Owner, Description, Type, Classifications, and Term. The first row contains "claim_savings" and "etl_user", with "hive_table" under Type and "DATA_Q..." under Classifications. The second row contains "claims_view" and "etl_user", with "hive_table" under Type and "DATA_QUALITY" under Classifications.

...and click on claim_savings to see that the quality score associated with this table is less than 60%

The screenshot shows the Apache Atlas interface for the 'claim_savings' table. The 'Classification' tab is active, displaying the following details:

- Classifications:** DATA_QUALITY
- Term:** DATA_QUALITY (1)
- Attributes:** Name: score, Value: 0.51

Click back, and select claims_view table instead and click the lineage tab. This shows that this table was derived from the claims_saving table

The screenshot shows the Apache Atlas interface for the 'claims_view' table. The 'Lineage' tab is active, displaying the following lineage steps:

```

graph LR
    A[/hive_data/cost_s.../] --> B[create external t...]
    B --> C[claim_savings]
    C --> D[create view if no...]
    D --> E[claims_view]
  
```

The 'claims_view' node is highlighted with a red circle.

Click on the Classifications tab and notice that because the table claims_view table was derived from (claims_savings) had a DATA_QUALITY tag, the tag was automatically propagated to claims_view table itself (i.e. no one had to manually tag it)

The screenshot shows the Apache Atlas interface. On the left, there's a sidebar with search filters for 'Basic' or 'Advanced' search, and sections for 'Search By Type' (selected 'hive_table (23)'), 'Search By Classification' (selected 'DATA_QUALITY (1)'), 'Search By Term', and 'Search By Text'. Below these are buttons for 'Clear' and 'Search'. A 'Favorite Searches' section indicates no favorites yet. The main content area is titled 'claims_view (hive_table)'. It shows 'Classifications' (DATA_QUALITY), 'Term' (DATA_QUALITY), and 'Propagated Classifications' (DATA_QUALITY). Below this, tabs for 'Properties', 'Lineage', 'Relationships', 'Classifications' (which is selected), 'Audits', and 'Schema' are visible. A table titled 'Showing 1 - 1' displays one row under 'Classification' (DATA_QUALITY [Propagated From]). The table has columns for 'Name' (score) and 'Value' (0.51). At the bottom right, there's a 'Page Limit' dropdown set to 25, and navigation icons for < 1 >. The top right corner shows the user 'admin'.

10. Use Atlas to query for hive_tables and pick provider_summary to show lineage and impact

Apache Atlas

SEARCH CLASSIFICATION GLOSSARY

Basic Advanced

Search By Type: hive_table (23)

Search By Classification: Select Classification

Search By Term: Search Term

Search By Text: prov*

Clear Search

Favorite Searches Save Save As

You don't have any favorite search.

Results for: (Type: hive_table) AND (Query: prov*)

If you do not find the entity in search result below then you can create new entity

Showing 3 records From 1 - 25

<input type="checkbox"/>	Name	Owner	Description	Type	Classifications	Term
<input type="checkbox"/>	provider_summary	etl_user		hive_table	+	+
<input type="checkbox"/>	prov_view2	etl_user		hive_table	+	+
<input type="checkbox"/>	prov_view	etl_user		hive_table	+	+

Exclude sub-types Exclude sub-classifications Show historical entities Columns

Page Limit: 25

Apache Atlas

SEARCH CLASSIFICATION GLOSSARY

Basic Advanced

Search By Type: hive_table (23)

Search By Classification: Select Classification

Search By Term: Search Term

Search By Text: prov*

Clear Search

Favorite Searches Save Save As

You don't have any favorite search.

Back To Results provider_summary (hive_table)

Classification: [+](#)

Term: [+](#)

Properties Lineage Relationships Classifications Audits Schema

Current Entity In Progress Lineage Impact

```

graph LR
    A[/hive_data/claim/] -- "create external t..." --> B(provider_summary)
    B -- "create view if no..." --> C(prov_view2)
    B -- "create view if no..." --> D(prov_view)
  
```

Switch to Beta UI

You can use the Audits tab to see audits on this table

The screenshot shows the Apache Atlas interface for the entity `provider_summary (hive_table)`. The 'Audits' tab is active, showing the following audit log entries:

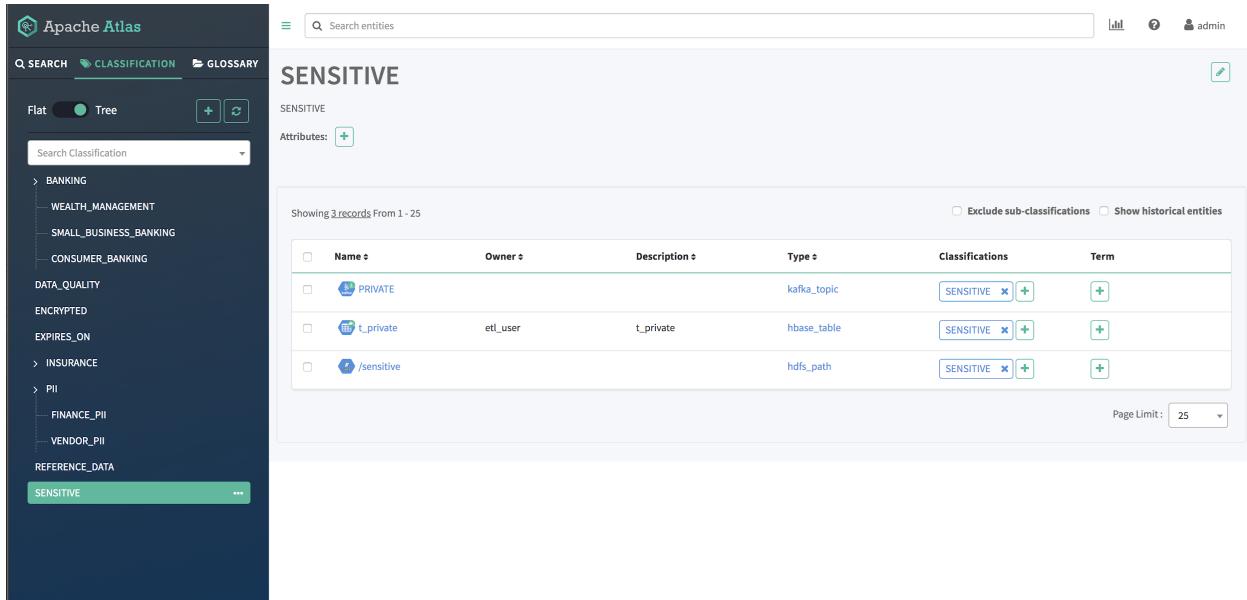
User	Timestamp	Action	Tools
etl_user	Thu Jan 09 2020 19:43:30 GMT-0800 (Pacific Standard Time)	Entity Updated	Detail
etl_user	Thu Jan 09 2020 19:43:29 GMT-0800 (Pacific Standard Time)	Entity Updated	Detail
hive	Thu Jan 09 2020 19:43:29 GMT-0800 (Pacific Standard Time)	Entity Created	Detail

You can use Schema tab to inspect table schema

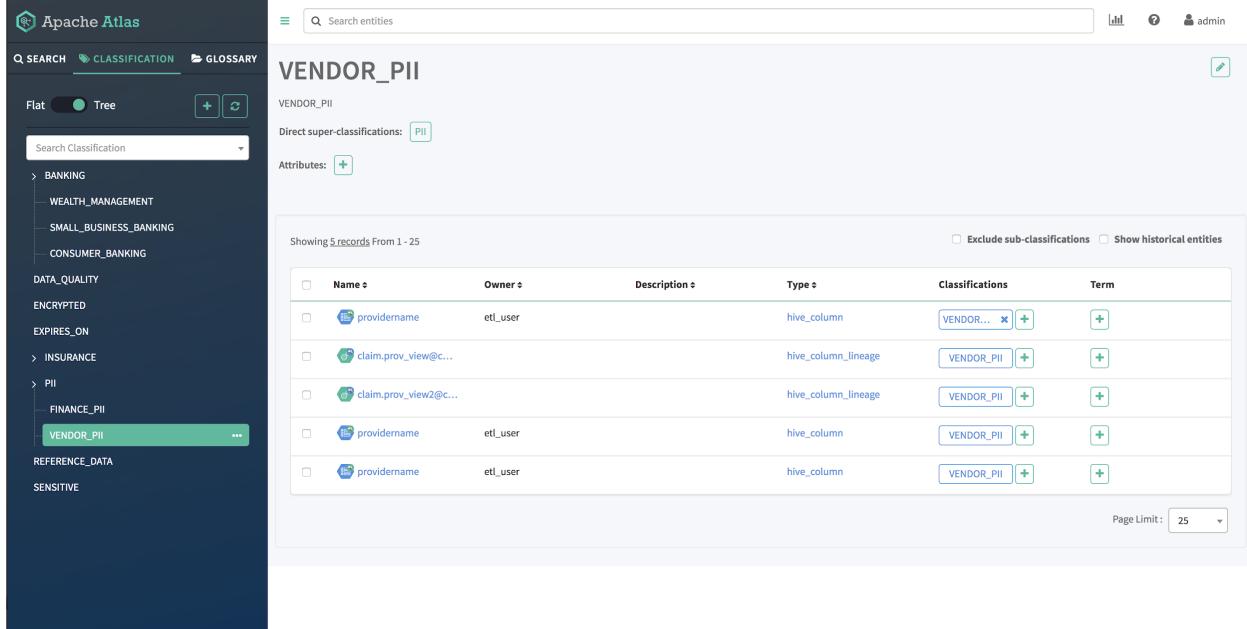
The screenshot shows the Apache Atlas interface for the entity `provider_summary (hive_table)`. The 'Schema' tab is active, showing the following column definitions:

Name	Owner	Type	Classifications
providerid	etl_user	string	+
totaldischarges	etl_user	int	+
providerstate	etl_user	string	+
averagemedicarepayments	etl_user	decimal(10,2)	+
providerreferralregion	etl_user	string	+
providername	etl_user	string	VENDOR... +
averagecoveredcharges	etl_user	decimal(10,2)	+
providerzip	etl_user	string	+
providerid	etl_user	string	+
providerstreetaddress	etl_user	string	+
averagetrainingamounts	etl_user	decimal(10,2)	+

11. Navigate to the Classification tab to see how you can easily see all entities tagged with a certain classification (across Hive, Hbase, Kafka, HDFS etc)



The screenshot shows the Apache Atlas interface with the 'CLASSIFICATION' tab selected. On the left, a navigation tree includes categories like BANKING, WEALTH MANAGEMENT, SMALL_BUSINESS_BANKING, CONSUMER_BANKING, DATA_QUALITY, ENCRYPTED, EXPIRES_ON, INSURANCE, PII, FINANCE_PII, VENDOR_PII, and REFERENCE_DATA. The 'SENSITIVE' category is highlighted. The main panel displays the 'SENSITIVE' classification details, showing 3 records from 1-25. The table columns are Name, Owner, Description, Type, Classifications, and Term. The first record is 'PRIVATE' (Owner: etl_user, Description: kafka_topic, Type: kafka_topic, Classifications: SENSITIVE). The second record is 't_private' (Owner: etl_user, Description: t_private, Type: hbase_table, Classifications: SENSITIVE). The third record is '/sensitive' (Owner: etl_user, Description: hdfs_path, Type: hdfs_path, Classifications: SENSITIVE). A page limit of 25 is set.



The screenshot shows the Apache Atlas interface with the 'CLASSIFICATION' tab selected. The left navigation tree is identical to the previous screenshot. The 'VENDOR_PII' category is highlighted. The main panel displays the 'VENDOR_PII' classification details, showing 5 records from 1-25. The table columns are Name, Owner, Description, Type, Classifications, and Term. The records include various provider names and their corresponding hive_column and lineage types, each associated with the VENDOR_PII classification. A page limit of 25 is set.

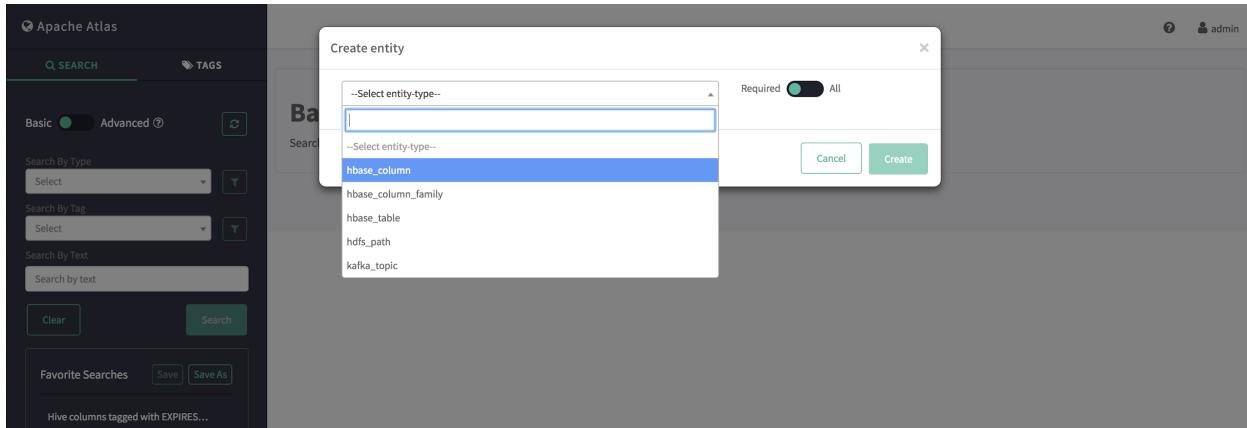
Navigate to the Glossary tab to see how you can define Glossary categories and terms, as well as search for any entities associated with those terms:

The screenshot shows the Apache Atlas interface. On the left, there's a sidebar with a tree view of terms under 'Automotive' and 'Finance'. Under 'Finance', 'Annuity' is selected and highlighted in green. The main content area is titled 'Annuity' and contains two sections: 'Short Description' and 'Long Description', both of which state: 'An annuity is a retirement vehicle sold by insurance companies that provides the benefits of tax deferral and protects your principal. It sounds really great, and in some respects it is.' Below these descriptions are 'Classifications' and 'Categories' sections, each with a '+' button. At the bottom of the main content area are three tabs: 'Entities', 'Classifications', and 'Related Terms'. To the right of these tabs are three checkboxes: 'Exclude sub-types', 'Exclude sub-classifications', and 'Show historical entities'. A message 'No Records found!' is displayed below the tabs. The bottom right corner of the main content area has a link 'Switch to Beta UI'.

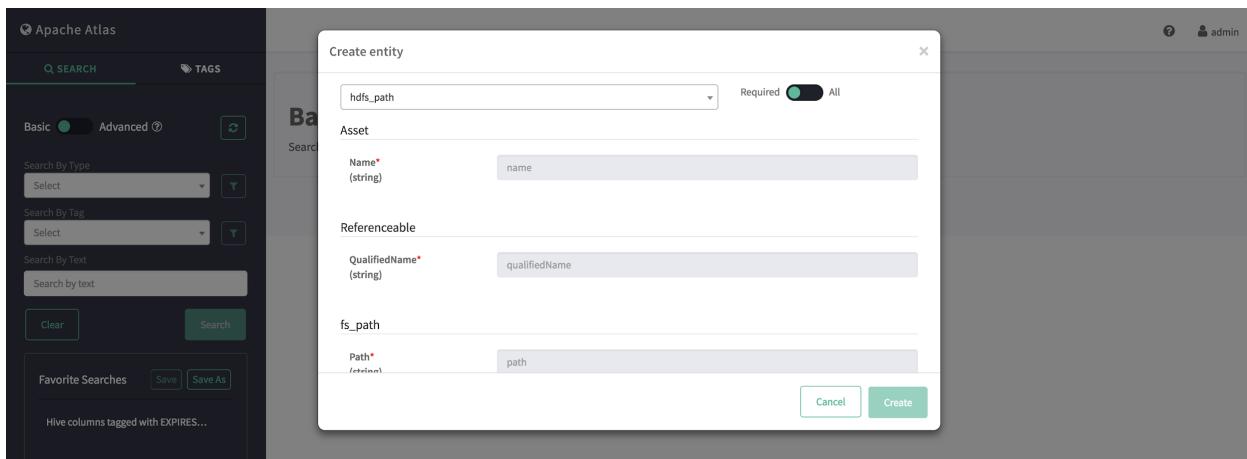
12. Navigate to Atlas home page and notice the option to create a new entity

The screenshot shows the Apache Atlas search interface. On the left, there's a sidebar with search filters: 'Search By Type' (dropdown menu), 'Search By Classification' (dropdown menu), 'Search By Term' (dropdown menu), and 'Search By Text' (text input field). Below these are buttons for 'Clear' and 'Search'. At the bottom of the sidebar are buttons for 'Favorite Searches', 'Save', and 'Save As', with a note 'You don't have any favorite search.' To the right of the sidebar is a main search area titled 'Basic Search' with the sub-instruction 'Search Atlas for existing entities or [create new entity](#)'. The top right corner of the search area has a link 'Switch to Beta UI'.

Sample out of the box entity types that you can create shown below:



Selecting an entity type (e.g. hdfs_path), shows what required and optional fields you would need to provide to manually create the new entity



Hive ACID/Merge walk through

13. In Zeppelin there are two Hive related notebooks provided to demonstrate Hive ACID and MERGE capabilities. Login to Zeppelin as etl_user to be able to run these:

The notebooks contain tutorials that walk through some of the theory and concepts before going through some basic examples:

Demos / Hive ACID

Introduction

Hadoop is gradually playing a larger role as a system of record for many workloads. Systems of record need robust and varied options for data updates that may range from single records to complex multi-step transactions.

Some reasons to perform updates may include:

- Data restatements from upstream data providers.
- Data pipeline reprocessing.
- Slowly-changing dimensions (e.g. SCD Type 1)
- Dimension history / evolution (e.g. SCD Type 2)

Standard SQL provides ACID operations through INSERT, UPDATE, DELETE, transactions, and the more recent MERGE operations. These have proven to be robust and flexible enough for most workloads.

Took 0 sec. Last updated by etl_user at January 08 2020, 6:22:09 PM. (outdated)

Concepts

Transactional Tables: Hive supports single-table transactions. Tables must be marked as transactional in order to support UPDATE and DELETE operations.

Partitioned Tables: Hive supports table partitioning as a means of separating data for faster writes and queries. Partitions are independent of ACID. Large tables in Hive are almost always partitioned. Large ACID tables should be partitioned for optimal performance.

ACID Operations (INSERT / UPDATE / DELETE): Standard SQL commands that allow data inserts, updates and deletes.

Primary Key: Databases use primary keys to make records easy to locate, which facilitates updates or deletes. Hive does not enforce the notion of primary keys, but if you plan to do large-scale updates and deletes you should establish a primary key convention within your application.

Streaming Ingest: Data can be streamed into transactional Hive tables in real-time using NiFi, Flume or a lower-level direct API.

Optimistic Concurrency: ACID updates and deletes to Hive tables are resolved by letting the first committer win. This happens at the partition level, or at the table level for unpartitioned tables.

Compactions: Data must be periodically compacted to save space and optimize data access. It is best to let the system handle these automatically, but these can also be scheduled in an external scheduler.

Took 0 sec. Last updated by etl_user at January 08 2020, 6:23:27 PM.

Hello ACID: Create a Partitioned ACID Table and Insert some Data

Let's start by creating a transactional table. Only transactional tables can support updates and deletes

Took 0 sec. Last updated by etl_user at January 08 2020, 5:42:36 PM. (outdated)

```
%jdbc(hive)
(CREATE TABLE IF NOT EXISTS hello_ocid (key int, value int)
PARTITIONED BY (load_date date)
CLUSTERED BY(key) INTO 3 BUCKETS)
```

FINISHED

Checking the table that was created using `describe formatted`, notice that with Hive 3 default created as managed with ACID and ORC enabled

Took 0 sec. Last updated by etl_user at January 08 2020, 6:15:03 PM.

Hive Merge demo

Hive Merge allows actions to be performed on a target table based on the results of a join with a source table by combining the UPDATE and INSERT statements into a single statement.

To understand it better, let's go through a basic tutorial (original here: <https://community.cloudera.com/t5/Community-Articles/Hive-ACID-Merge-by-Example/ta-p/245402>)

Let's create 2 tables, one as the target of merge and one as the source of merge. Please note that the target table must be bucketed, set as transaction enabled and stored in orc format

Query executed successfully. Affected rows : 0
Took 0 sec. Last updated by etl_user at January 08 2020, 5:23:05 PM.

Create database

```
%jdbc(hive)
CREATE DATABASE merge_data
```

Query executed successfully. Affected rows : 0
Took 0 sec. Last updated by etl_user at January 08 2020, 4:42:28 PM. (outdated)

Create target table

```
%jdbc(hive)
CREATE TABLE merge_data.transactions(
  ID int,
  TranValue string,
  last_update_user string)
PARTITIONED BY (tran_date string)
CLUSTERED BY (ID) into 5 buckets
```

Query executed successfully. Affected rows : 0
Took 0 sec. Last updated by etlUser at January 08 2020, 4:42:48 PM. (outdated)

Create source table

```
%jdbc(hive)
CREATE TABLE merge_data.merge_source(
  ID int,
  TranValue string,
  tran_date string)
```

Query executed successfully. Affected rows : 0
Took 0 sec. Last updated by etlUser at January 08 2020, 4:43:05 PM. (outdated)

Checking the tables that were created, notice that with Hive 3 tables are by default created as managed with ACID and ORC enabled

Query executed successfully. Affected rows : 0
Took 0 sec. Last updated by etlUser at January 08 2020, 6:13:19 PM.

Table View

col_name	data_type	comment
id	int	
tranvalue	string	
last_update_user	string	
	null	null
# Partition Information	null	null
# col_name	data_type	comment

Appendix: Older AMIs

Older AMI links (for HDP releases) can be found below:

- For HDP 3.1.4: click [here](#)
- For HDP 3.1 with Knox SSO: click [here](#)
- For HDP 2.6.5 with Knox SSO: click [here](#)
- For HDP 2.6.5: click [here](#)