

# Apache Ranger



## Objectives

- Describe Apache Ranger
- Understand the need for Apache Ranger
- Discuss Apache Ranger Architecture
- Configure Apache Solr
- Install and Configure Apache Ranger
- Optional Apache Ranger Configurations

276 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Objectives



- What is Apache Ranger



## What is Apache Ranger

- Framework to Enable, Monitor, & Manage Comprehensive Data Security Across the Apache Hadoop Platform
- Centralized Security Administration to Manage All Security Related Tasks in a Central UI or using REST APIs.
- Fine Grained Authorization for a Specific Action and/or Operation with Hadoop Component/Tool
- Enhanced Support for Different Authorization Methods
  - Role Based Access Control,
  - Attribute Based Access Control
- Centralize Auditing of User Access and Administrative Actions Within All the Hadoop Components

278 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



<http://ranger.apache.org>

Ranger is a framework to enable, monitor and manage comprehensive data security across the Hadoop platform.

The vision with Ranger is to provide comprehensive security across the Apache Hadoop ecosystem. With the advent of Apache YARN, the Hadoop platform can now support a true data lake architecture. Enterprises can potentially run multiple workloads, in a multi tenant environment. Data security within Hadoop needs to evolve to support multiple use cases for data access, while also providing a framework for central administration of security policies and monitoring of user access.

Please read the FAQs if you need to understand how it works over Apache Hadoop components.

## What is Apache Ranger

Centralized Security Administration has Four Aspects

- Authentication – Previously Discussed
  - Effect by Kerberos in native Apache Hadoop
  - Secured by the Apache Knox Gateway Via the HTTP/REST API
- Authorization
  - Fine-Grained Access Control Provides Flexibility in Defining Policies on
    - The Directory and File Level, Via HDFS
    - The Database, Table and Column Level, Via Hive
    - The Table, Column Family and Column Level, Via Hbase

279 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Centralized security administration in a Hadoop environment has four aspects:

Authentication

Effect by Kerberos in native Apache Hadoop, and secured by the Apache Knox Gateway via the HTTP/REST API. (For further information, see the Apache Knox Gateway Manager Guide.)

Authorization

Fine-grained access control provides flexibility in defining policies...

on the folder and file level, via HDFS

on the database, table and column level, via Hive

## What is Apache Ranger

Centralized Security Administration has Four Aspects

- Audit

- Controls Access into the System Via Extensive User Access Auditing in HDFS, Hive, HBase at...
  - IP Address
  - Resource/Resource Type
  - Timestamp
  - Access Granted or Denied

- Data Protection Provided By

- Wire Encryption
- File/Directory Encryption Via HDFS Encryption
- File/Table Encryption Via HDFS Encryption and Hortonworks Partners
- File/Column Encryption Via Hortonworks partners

280 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Centralized security administration in a Hadoop environment has four aspects:

Authentication

Effectuated by Kerberos in native Apache Hadoop, and secured by the Apache Knox Gateway via the HTTP/REST API. (For further information, see the Apache Knox Gateway Manager Guide.)

Authorization

Fine-grained access control provides flexibility in defining policies...

on the folder and file level, via HDFS

on the database, table and column level, via Hive

## What is Apache Ranger

HDP 2.5

### Centralized Security Administration w/ Ranger

#### Authentication

Who am I/prove it?

- *Kerberos*
- API security with *Apache Knox*

#### Authorization

What can I do?

- Fine grain access control with *Apache Ranger*

#### Audit

What did I do?

- Centralized audit reporting w/ *Apache Ranger*

#### Data Protection

Can data be encrypted at rest and over the wire?

- Wire encryption in Hadoop
- *Native HDFS* encryption & Partners



## Objectives



- What is Apache Ranger?
- Why is Apache Ranger Needed?



## Why is Apache Ranger Needed

- Offers Centralized Security Framework to Manage Fine Grained Access Control Over Hadoop and Ecosystem Components
- Standardize Authorization Method Across All Hadoop Components.
- Using Administration Console, users can easily manage policies around accessing a resource (file, folder, database, table, column etc) for a particular set of users and/or groups, and enforce the policies within Hadoop.
- Can Enable Audit Tracking and Policy Analytics for Deeper Control of the Hadoop environment
- Provides Ability to Delegate Administration of Certain Data to Other Group Owners, with the Aim of Decentralizing Data Ownership

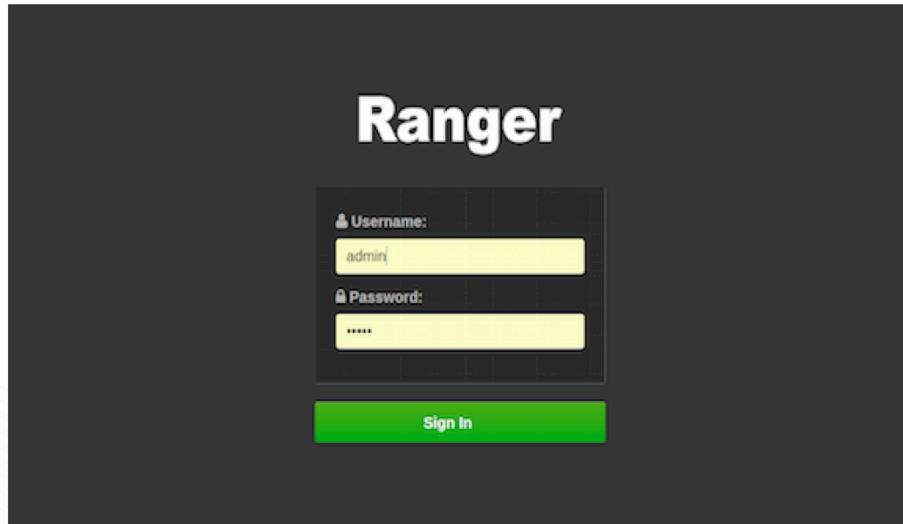


<http://ranger.apache.org>

What does Apache Ranger offer for Apache Hadoop and related components?

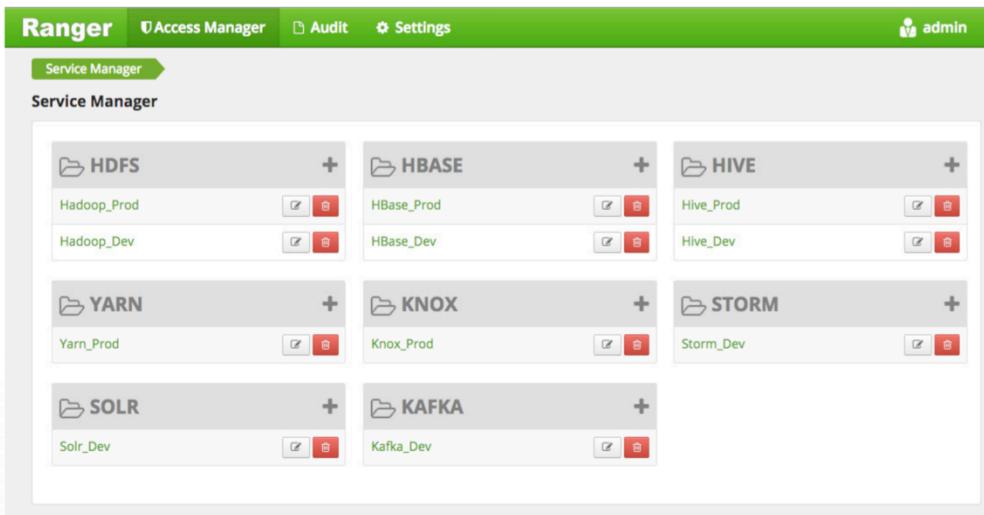
Apache Ranger offers a centralized security framework to manage fine grained access control over Hadoop and related components (Apache Hive, HBase etc.). Using Ranger administration console, users can easily manage policies around accessing a resource (file, folder, database, table, column etc) for a particular set of users and/or groups, and enforce the policies within Hadoop. They also can enable audit tracking and policy analytics for deeper control of the environment. Ranger' solution also provides ability to delegate administration of certain data to other group owners, with an aim of decentralizing data ownership

## Why is Apache Ranger Needed



284 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Why is Apache Ranger Needed



The screenshot shows the Apache Ranger Service Manager interface. At the top, there is a green header bar with the Ranger logo, 'Access Manager', 'Audit', 'Settings', and a user icon labeled 'admin'. Below the header is a breadcrumb navigation bar with 'Service Manager' highlighted. The main area is titled 'Service Manager' and contains a grid of service configurations. Each service has a folder icon and a '+' sign. Inside each folder are two entries: 'Hadoop\_Prod' and 'Hadoop\_Dev' for HDFS; 'HBase\_Prod' and 'HBase\_Dev' for HBASE; 'Hive\_Prod' and 'Hive\_Dev' for HIVE; 'Yarn\_Prod' for YARN; 'Knox\_Prod' for KNOX; 'Storm\_Dev' for STORM; 'Solr\_Dev' for SOLR; and 'Kafka\_Dev' for KAFKA. Each entry has a small edit icon and a delete icon. In the bottom right corner of the interface, there is a green hexagonal badge with the text 'Hortonworks UNIVERSITY'.

285 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Why is Apache Ranger Needed

Ranger Policy Manager

Manage Repository > sandbox\_hdfs Policies > Edit Policy

Edit Policy

Policy Details :

Policy Name : Marketing Policy      enabled

Resource Path \* : /demo/data/Customer\*

Recursive : NO

Audit Logging : ON

User and Group Permissions :

Group Permissions	Select Group	Read	Write	Execute	Admin
	Marketing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	+ [Add]				

file level  
access  
control,  
flexible  
definition

Control  
permissions



286 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Setup Authorization Policies for HDFS

## Why is Apache Ranger Needed

The screenshot shows the 'Create Policy' page in the Hortonworks Policy Manager. At the top, there are tabs for 'Policy Manager', 'Users/Groups', 'Analytics', and 'Audit'. Below the tabs, the breadcrumb navigation shows 'Manage Repository > hivedev Policies > Edit Policy'. The main section is titled 'Create Policy' and contains 'Policy Details' and 'User and Group Permissions' sections.

**Policy Details:**

- Select Database Name:
- Select Table:   **Include**
- Select Column Name:
  - phone\_number
  - plan
  - date
  - status
  - balance
  - region
- Audit Logging: **ON**

**User and Group Permissions:**

- Group Permissions:** Select Group:   Select  Update  Create  Drop  Alter  Index  Lock  All  Admin
- User Permissions:** Select User:   Select  Update  Create  Drop  Alter  Index  Lock  All  Admin

287 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Setup Authorization Policies for Hive

## Why is Apache Ranger Needed

The screenshot shows the Apache Ranger web interface with a green header bar containing links for Policy Manager, Users/Groups, Analytics, and Audit. The user 'admin' is logged in. Below the header is a navigation bar with tabs for Access, Admin, Login Sessions, and Agents. A search bar is present with the text 'REPOSITORY TYPE: Hive'. A table displays audit logs with the following columns: Event Time, User, Repository Name / Type, Resource Name, Access Type, Result, Access Enforcer, and Client IP. The data in the table is as follows:

Event Time	User	Repository Name / Type	Resource Name	Access Type	Result	Access Enforcer	Client IP
02/04/2015 03:02:04 PM	mktg1	sandbox_hive Hive	xademo/customer_details/phone_num...	SELECT	Allowed	xasecure-acl	127.0.0.1
02/04/2015 03:02:03 PM	mktg1	sandbox_hive Hive	xademo	USE	Allowed	xasecure-acl	127.0.0.1
02/04/2015 03:01:32 PM	mktg1	sandbox_hive Hive	xademo/customer_details/balance	SELECT	Denied	xasecure-acl	127.0.0.1
02/04/2015 03:01:22 PM	mktg1	sandbox_hive Hive	xademo	USE	Allowed	xasecure-acl	127.0.0.1
01/21/2015 11:22:33 AM	mktg1	sandbox_hive Hive	xademo/customer_details/phone_num...	SELECT	Allowed	xasecure-acl	127.0.0.1

288 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Monitor through Auditing

## Objectives



- What is Apache Ranger?
- Why is Apache Ranger Needed?
- Apache Ranger Architecture

289 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Apache Ranger Architecture

- At the Core Has Centralized Web Application
- Modules Consisting of
  - Policy Administration
  - Audit and Reporting
- Authorized User Are Able to Manage Security Policies using Web Tools or REST API's
- Security Policies Are Enforced within Hadoop using Lightweight Plugins
- Plugins Run as Part of the Service – No Additional OS Level Process to Manage
  - Namenode – HDFS
  - HiveServer2 – Hive
  - Hbase Server – Hbase
  - Nimbus Server – Storm
  - Knox Server - Knox



290 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

How does it work over Hadoop and related components

Apache Ranger' solution at the core has a centralized web application, which consists of the policy administration, audit and reporting modules. Authorized users will be able to manage their security policies using the web tool or using REST APIs. These security policies are enforced within Hadoop ecosystem using lightweight Ranger Java plugins, which run as part of the same process as the namenode (HDFS), Hive2Server(Hive), HBase server (Hbase), Nimbus server (Storm) and Knox server (Knox) respectively. Thus there is no additional OS level process to manage.

Is there a single point of failure?

No, Apache Ranger is not a Single Point of Failure. Ranger' plugins run within the same process as the component, e.g. NameNode for HDFS. These agents pull the policy-changes using REST API at a configured regular interval (e.g.: 30 second). The plugin is able to function even if the policy server is temporarily down and will provide the authorization enforcement. Also, the policy manager web application can be hosted on a HA infrastructure. (with multiple apache server, multiple tomcat servers and a standby database server w/o replication setup).

## Apache Ranger Architecture

- No Single Point of Failure
- Plugins Run Within the Same Process as the Component Service
- Agent Pull Policy-Changes using REST API at Configured Regular Interval
- Plugin Able to Function
  - When Policy Server is Temporarily Down
  - Will Still Provide Authorization Enforcement
- Policy Manager Web Application can be Host on High Available Infrastructure Utilizing
  - Multiple Apache Servers
  - Multiple Tomcat Servers
  - Standby Database Server Without Replication Setup

291 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



<http://ranger.apache.org>

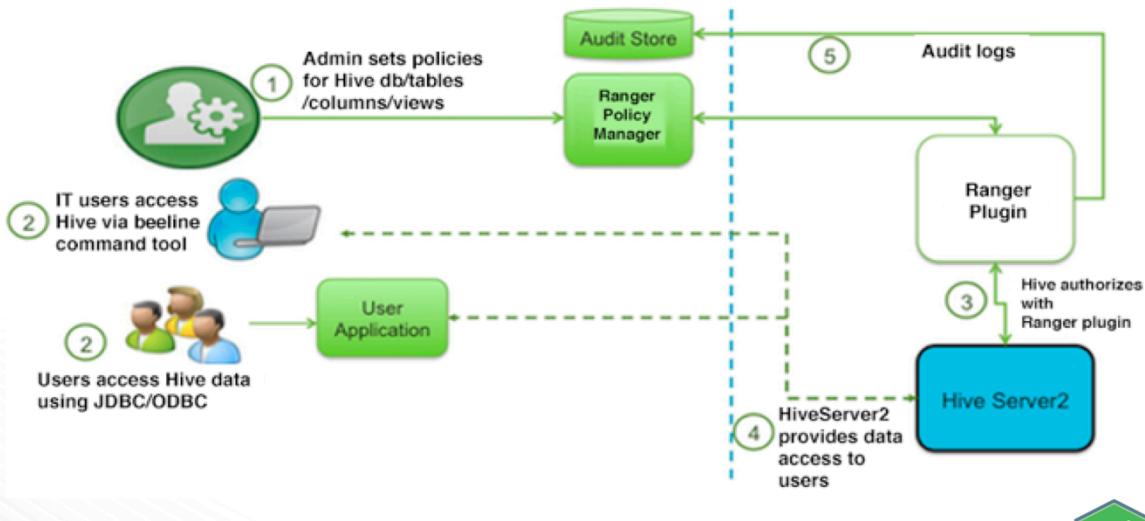
How does it work over Hadoop and related components

Apache Ranger's solution at the core has a centralized web application, which consists of the policy administration, audit and reporting modules. Authorized users will be able to manage their security policies using the web tool or using REST APIs. These security policies are enforced within Hadoop ecosystem using lightweight Ranger Java plugins, which run as part of the same process as the namenode (HDFS), Hive2Server(Hive), HBase server (Hbase), Nimbus server (Storm) and Knox server (Knox) respectively. Thus there is no additional OS level process to manage.

Is there a single point of failure?

No, Apache Ranger is not a Single Point of Failure. Ranger's plugins run within the same process as the component, e.g. NameNode for HDFS. These agents pull the policy-changes using REST API at a configured regular interval (e.g.: 30 second). The plugin is able to function even if the policy server is temporarily down and will provide the authorization enforcement. Also, the policy manager web application can

## Apache Ranger Architecture



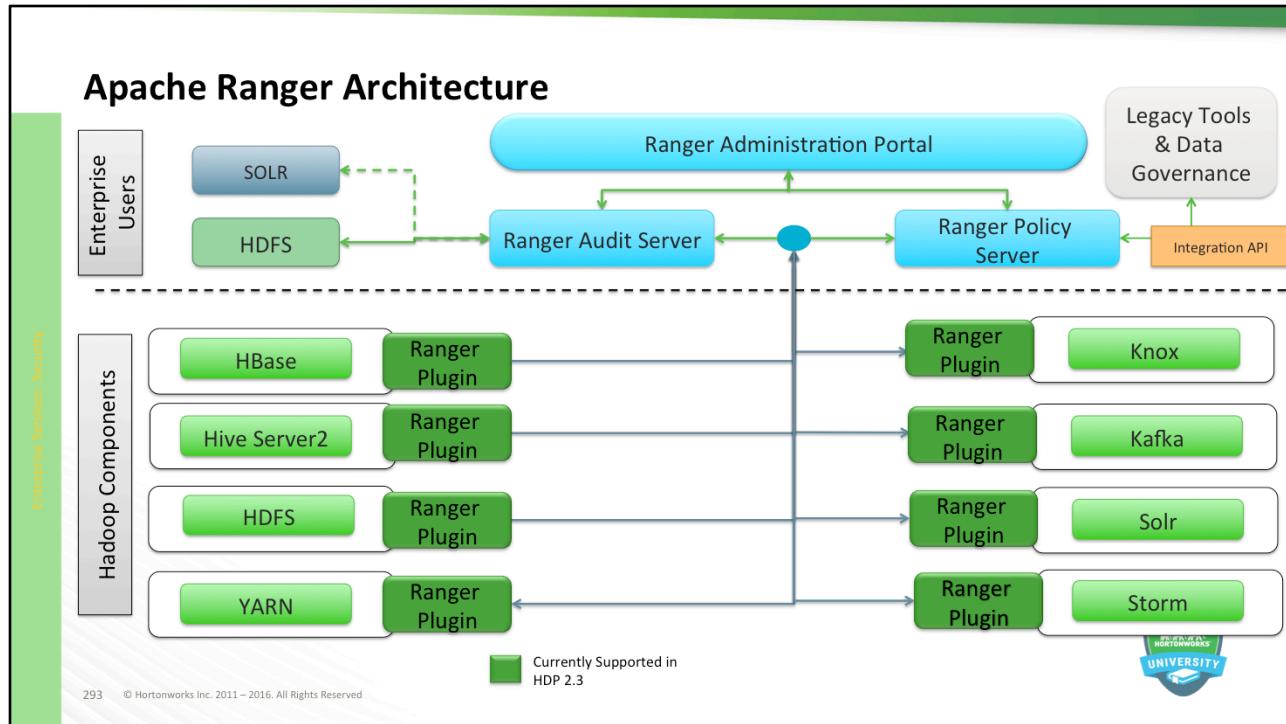
292 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



<http://ranger.apache.org>

Lets take a min to step through how Hive plugin would work from end user perspective

( then talk through each step)



## Architectuall overview of how it works

Similar to Hive plugin in earlier slide, plugins for other Hadoop components would work the same way

Plugins are embedded with the component so if Admin goes down, policies are still in effect (may not have latest policies downloaded that's all)

No OS administrative overhead (i.e. no new daemons spawned) for each plugin

## Objectives



- What is Apache Ranger?
- Why is Apache Ranger Needed?
- Apache Ranger Architecture
- Prerequisites for Apache Ranger



## Prerequisites for Apache Ranger

- Recommended that Audits Stored in Both HDFS and Solr
- Install Apache Solr
- Setup either OS/AD sync (e.g. SSSD) or Hadoop Group Mapping for AD
  - To Ensure LDAP/AD Group Level Authorization is Enforced in Hadoop
- Ranger Installation Will Create Two New Users
  - rangeradmin
  - rangerlogger
- Ranger Installation Will Create Two New Databases
  - ranger
  - ranger\_audit
- Select and Configure Database Instance – MySQL, PostgreSQL, Oracle

295 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Before you install Ranger, make sure your cluster meets the following requirements:

It is recommended that you store audits in both HDFS and Solr, so you should install Apache Solr.

To ensure that LDAP/AD group level authorization is enforced in Hadoop, you should set up Hadoop group mapping for LDAP.

A MySQL, Oracle, PostgreSQL, MS SQL, or SQL Anywhere database instance must be running and available to be used by Ranger.

The Ranger installation will create two new users (default names: rangeradmin and rangerlogger) and two new databases (default names: ranger and ranger\_audit).

Configuration of the database instance for Ranger is described in the following sections for some of the databases supported by Ranger.

## Objectives



- What is Apache Ranger?
- Why is Apache Ranger Needed?
- Apache Ranger Architecture
- Prerequisites for Apache Ranger
- Install and Configure MySQL Database



## Prerequisites for Apache Ranger

- Install MySQL Database Instance

```
# yum install mysql-server  
# chkconfig mysqld on  
# service mysqld start  
# mysql-secure-installation
```

- Add Grant to Root User or Database Administrator Account

```
CREATE USER 'root'@'%';  
GRANT ALL PRIVILEGES ON *.* to 'root'@'%' WITH GRANT OPTION;  
SET PASSWORD FOR 'root'@'%' = PASSWORD('BadPass#1');  
SET PASSWORD = PASSWORD('BadPass#1');  
FLUSH PRIVILEGES;  
exit;
```

297 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



The MySQL database administrator should be used to create the Ranger databases.

The following series of commands could be used to create the rangerdba user with password rangerdba.

Log in as the root user, then use the following commands to create the rangerdba user and grant it adequate privileges.

Use the exit command to exit MySQL.

You should now be able to reconnect to the database as rangerdba using the following command:

```
mysql -u root -p <password>
```

## Prerequisites for Apache Ranger

- Install MySQL Database Instance - Continued
- Test Connection - Prompts for Password  
    # mysql -u root -p
- Install mysql-connector-java if needed  
    yum install mysql-connector-java
- Setup Ambari for MySQL  
    # ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar
- Add Service – Ranger Install will Create the Databases “ranger” & “ranger\_audit”

298 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



The MySQL database administrator should be used to create the Ranger databases.

The following series of commands could be used to create the rangerdba user with password rangerdba.

Log in as the root user, then use the following commands to create the rangerdba user and grant it adequate privileges.

Use the exit command to exit MySQL.

You should now be able to reconnect to the database as rangerdba using the following command:

```
mysql -u root -p <password>
```

## Objectives



- What is Apache Ranger?
- Why is Apache Ranger Needed?
- Apache Ranger Architecture
- Prerequisites for Apache Ranger
- Install and Configure MySQL Database
- Install and Configure Apache Solr

299 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Solr with Ambari

- Apache Solr a Open-Source Enterprise Search Platform Utilized by Ranger
- Stores Audit Logs
- Provides Search Capability Through Ranger Admin UI
- Solr Must be Installed/Configured Prior to Installing Ranger
- Recommended Ranger Audits Written Both Solr and HDFS
- HDFS is a long-term destination for Audit Logs
- SolrCloud is Recommended Configuration
- Scalable Architecture – Can Run as Single Node or Multi-Node Cluster
- Includes Features Such as Replication and Sharding - Useful for High Availability (HA) and Scalability

300 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Apache Solr is an open-source enterprise search platform. Apache Ranger can use Apache Solr to store audit logs, and Solr can also provide a search capability of the audit logs through the Ranger Admin UI.

### Important

Solr must be installed and configured before installing RangerAdmin or any of the Ranger component plugins.

It is recommended that Ranger audits be written to both Solr and HDFS. Audits to Solr are primarily used to enable search queries from the Ranger Admin UI. HDFS is a long-term destination for audits -- audits stored in HDFS can be exported to any SIEM system, or to another audit store.

### Configuration Options

Solr Standalone -- Solr Standalone is only recommended for testing and evaluation.

## Install Apache Solr with Ambari

- Ambari service for Apache Solr Currently Not Included in Ambari 2.2.x
- Demo Service Available via GitHub At <https://github.com/abajwa-hw/solr-stack>:  
– git clone https://github.com/abajwa-hw/solr-stack.git /var/lib/ambari-server/resources/stacks/HDP/2.4/services/SOLR
- Restart Ambari Server



To deploy the Solr stack, run below

```
VERSION=`hdp-select status hadoop-client | sed 's/hadoop-client - \([0-9]\.[0-9]\)\.*/\n\1/'\n\nsudo git clone https://github.com/abajwa-hw/solr-stack.git /var/lib/ambari-server/\nresources/stacks/HDP/$VERSION/services/SOLR
```

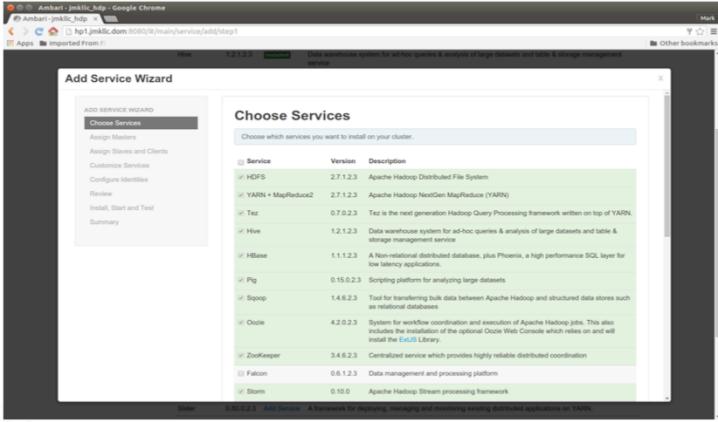
## Install Apache Solr with Ambari

The screenshot shows the Ambari interface for managing a Hadoop cluster. The top navigation bar includes links for Dashboard, Services, Hosts, Alerts, Admin, and a user icon. The main content area is titled "Stack and Services" and displays a table of installed services. The columns are Service, Version, Status, and Description. The services listed are:

Service	Version	Status	Description
HDFS	2.7.1.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Apache Hadoop Distributed File System
MapReduce2	2.7.1.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Apache Hadoop NextGen MapReduce (YARN)
YARN	2.7.1.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Apache Hadoop NextGen MapReduce (YARN)
Tez	0.7.0.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
Hive	1.2.1.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
HBase	1.1.1.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications.
Pig	0.15.0.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Scripting platform for analyzing large datasets
Sqoop	1.4.6.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
Oozie	4.2.0.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the <a href="#">ExtJS Library</a> .
ZooKeeper	3.4.6.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Centralized service which provides highly reliable distributed coordination
Falcon	0.6.1.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Data management and processing platform
Storm	0.10.0	<span style="background-color: green; color: white;">HEALTHY</span>	Apache Hadoop Stream processing framework
Flume	1.5.2.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS
Accumulo	1.7.0.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Robust, scalable, high performance distributed key/value store.
Ambari Metrics	0.1.0	<span style="background-color: green; color: white;">HEALTHY</span>	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
Atlas	0.5.0.2.3	<span style="background-color: green; color: white;">HEALTHY</span>	Atlas Metadata and Governance platform
Kafka	0.9.0.2.5	<span style="background-color: green; color: white;">HEALTHY</span>	A high-throughput distributed messaging system

At the bottom left, it says "302 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". On the right side, there is a blue hexagonal badge with the text "Hortonworks UNIVERSITY" and a graduation cap icon.

## Install Apache Solr with Ambari



The screenshot shows the Ambari Add Service Wizard interface. The title bar says "Install Apache Solr with Ambari". The main window is titled "Add Service Wizard" and "Choose Services". It displays a list of services to choose from, each with a checkbox, version number, and description. The services listed are:

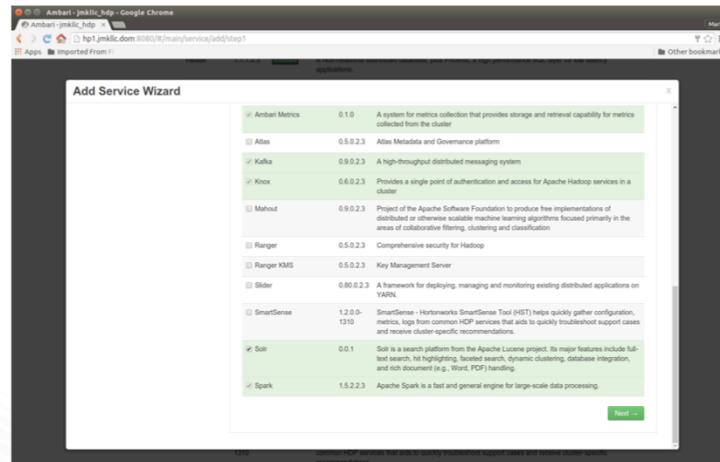
Service	Version	Description
HDFS	2.7.1.2.3	Apache Hadoop Distributed File System
YARN + MapReduce2	2.7.1.2.3	Apache Hadoop NextGen MapReduce (YARN)
Tez	0.7.0.2.3	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
Hive	1.2.1.2.3	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
HBase	1.1.1.2.3	A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications
Pig	0.15.0.2.3	Scripting platform for analyzing large datasets
Sqoop	1.4.8.2.3	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
Oozie	4.2.0.2.3	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ELide Library.
ZooKeeper	3.4.6.2.3	Centralized service which provides highly reliable distributed coordination
Falcon	0.6.1.2.3	Data management and processing platform
Storm	0.10.0	Apache Hadoop Stream processing framework

At the bottom of the wizard, there are buttons for "Next Step" and "Cancel".

303 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Solr with Ambari



304 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Solr with Ambari

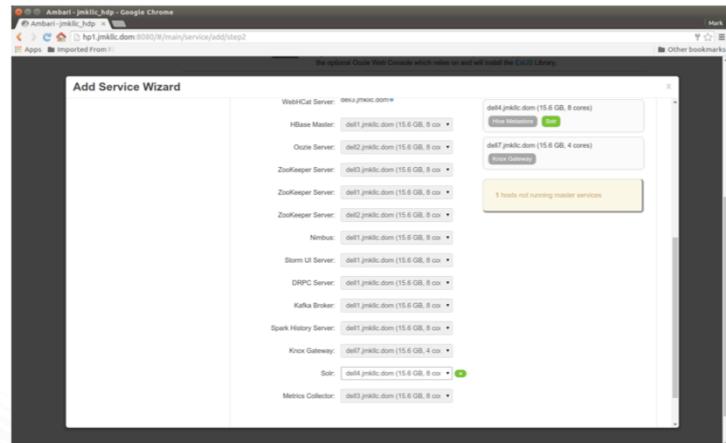
The screenshot shows the Ambari Add Service Wizard interface. The title bar says "Install Apache Solr with Ambari". The main window is titled "Add Service Wizard" and "Assign Masters". On the left, there's a sidebar with tabs: "Add Service Wizard", "Choose Services" (which is selected), "Assign Masters", "Assign Slaves and Clients", "Customize Services", "Configure Identities", "Review", "Install, Start and Test", and "Summary". The main area is titled "Assign Masters" and contains a sub-section "Assign master components to hosts you want to run them on". It lists several master components and their assigned hosts:

- NameNode: dell2.predictive.com (15.6 GB, 8 cores) assigned to dell1.predictive.com (15.6 GB, 8 cores)
- History Server: dell2.predictive.com (15.6 GB, 8 cores) assigned to dell2.predictive.com (15.6 GB, 8 cores)
- ResourceManager: dell1.predictive.com (15.6 GB, 8 cores) assigned to dell2.predictive.com (15.6 GB, 8 cores)
- App Timeline Server: dell2.predictive.com (15.6 GB, 8 cores) assigned to dell2.predictive.com (15.6 GB, 8 cores)
- ResourceManager: dell2.predictive.com (15.6 GB, 8 cores) assigned to dell3.predictive.com (15.6 GB, 8 cores)
- Hive Metastore: dell2.predictive.com (15.6 GB, 8 cores) assigned to dell3.predictive.com (15.6 GB, 8 cores)
- Hive Metastore: dell3.predictive.com (15.6 GB, 8 cores) assigned to dell4.predictive.com (15.6 GB, 8 cores)
- HiveServer2: dell3.predictive.com (15.6 GB, 8 cores) assigned to dell4.predictive.com (15.6 GB, 8 cores)
- WebHCat Server: dell3.predictive.com assigned to dell4.predictive.com (15.6 GB, 8 cores)
- HBase Master: dell1.predictive.com (15.6 GB, 8 cores) assigned to dell4.predictive.com (15.6 GB, 8 cores)
- Ozone Server: dell1.predictive.com (15.6 GB, 8 cores) assigned to dell7.predictive.com (15.6 GB, 4 cores)

Below the list, there's a note: "Note: If you have multiple hosts assigned to a service, it will be replicated across all hosts." At the bottom right of the main window, there's a "Next Step" button.

At the bottom of the page, there's a footer: "305 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". On the right side, there's a green hexagonal badge with the text "Hortonworks UNIVERSITY" and a graduation cap icon.

## Install Apache Solr with Ambari



306 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Solr with Ambari

The screenshot shows the Ambari Add Service Wizard interface. The title bar says "Install Apache Solr with Ambari". The main window is titled "Customize Services" and displays a message: "We have come up with recommended configurations for the services you selected. Customize them as you see fit." Below this, a list of services is shown: HDFS, MapReduce2, YARN, Tez, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Storm, Ambari Metrics, Kafka, Knox, Solr, Spark, Mac. A note indicates "There are 2 configuration changes in 1 service Show Details". A list of configuration groups is provided: Advanced solr-config, Advanced solr-env, Advanced solr-log4j-env, Advanced solr-env-env, Advanced solr-zoo-env, Custom solr-config. A green banner at the bottom states "All configurations have been addressed.".

307 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Solr with Ambari

Advanced solr-config

solr.maxmem	512m
solr.cloudmode	true
solr.conf	
solr.datadir	/opt/ranger_audit_server
solr.dir	/opt/solr
solr.download.location	HDPSEARCH
solr.minmem	512m
solr.mode	/ranger_audits

Advanced solr-env

solr.user	solr
solr.group	solr
solr.log.dir	/var/log/solr
solr.port	6083
solr.pid.dir	/var/run/solr

308 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Solr with Ambari

The screenshot shows the Ambari Add Service Wizard interface for installing Apache Solr. The current step is 'Step 5: Apache Solr - Preparing for configuration'. The main content area is titled 'Add Service Wizard' and contains a table titled 'Ambari Principals'. The table lists several keytab entries:

Principal	Keytab
smokeuser keytab	\$keytab_dir/smokeuser.headless.keytab
smokeuser principal name	\$cluster-env/smokeuser\$cluster_name\$realm
spark history kerberos principal	\$spark-env/spark_user\$cluster_name\$realm
spark history kerberos keytab	\$keytab_dir/spark.headless.keytab
storm principal name	\$system-env:storm_user\$cluster_name\$realm
Path to storm keytab file	\$keytab_dir/storm.headless.keytab
hbase user principal	\$hbase-env:hbase_user\$cluster_name\$realm
Path to hbase user keytab file	\$keytab_dir/hbase.headless.keytab
hdfs user principal	\$hadoop-env:hdfs_user\$cluster_name\$realm
Path to hdfs user keytab file	\$keytab_dir/hdfs.headless.keytab

A green progress bar at the bottom states 'All configurations have been addressed.' Below the table are 'Back' and 'Next >' buttons. The Ambari logo is visible in the top left corner, and the Hortonworks University logo is in the bottom right corner.

309 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Install Apache Solr with Ambari

The screenshot shows the Ambari Add Service Wizard in progress, specifically the 'Review' step. The title bar says 'Install Apache Solr with Ambari'. The left sidebar lists steps: 'ADD SERVICE WIZARD', 'Choose Services', 'Assign Masters', 'Assign Slaves and Clients', 'Customize Services', 'Configure Identities', and 'Review'. The 'Review' step is selected. The main panel displays configuration details:

- Admin Name:** admin
- Cluster Name:** jnklc\_hdp
- Total Hosts:** 6 (0 new)
- Repositories:**
  - metainf (HDP-2.3): http://hp1.jnklc.dom/hdp/HDP/metainf/2.x/repos/2.3.4.0/
  - centos (HDP-UTILS-1.1.0.20): http://hp1.jnklc.dom/hdp/HDP-UTILS-1.1.0.20/repos/centos/
- Services:**
  - Solr
  - Solr - de04.jnklc.dom

At the bottom, there are buttons for 'Back', 'Download CSV', 'Post', and 'Deploy ...'. Below the main panel, there are two service status cards:

- Solr**: Version 0.9.1, Add Service. Description: Solr is a search platform from the Apache Lucene project. Its major features include full-text search, highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling.
- Spark**: Version 1.5.2.2.3, Add Service. Description: Apache Spark is a fast and general engine for large-scale data processing.

310 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Solr with Ambari

The screenshot shows the Ambari Add Service Wizard in progress, specifically the 'Review' step. The title bar says 'Add Service Wizard' and 'Review'. The left sidebar lists steps: 'ADD SERVICE WIZARD', 'Choose Services', 'Assign Masters', 'Assign Slaves and Clients', 'Customize Services', 'Configure Identities', and 'Review'. The 'Review' step is selected. The main content area displays configuration details:

- Admin Name:** admin
- Cluster Name:** jmk1\_hdp
- Total Hosts:** 6 (0 new)
- Repositories:**
  - resolv (HDP-2.3): http://ip-1.jmk1.dom/hdp/HDPcentos6/2.x/repos/2.3.4/
  - resolv (HDP-UTILS-1.1.0.20): http://ip-1.jmk1.dom/hdp/HDP-UTILS-1.1.0.20/repos/centos/
- Services:**
  - Solr**: 0.9.1      **Add Service**: Solr is a search platform from the Apache Lucene project. Its major features include full-text search, highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling.
  - Spark**: 1.5.2.2.3      **Add Service**: Apache Spark is a fast and general engine for large-scale data processing.

At the bottom right are buttons: 'Download CSV', 'Print', and 'Deploy ...'.

311 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



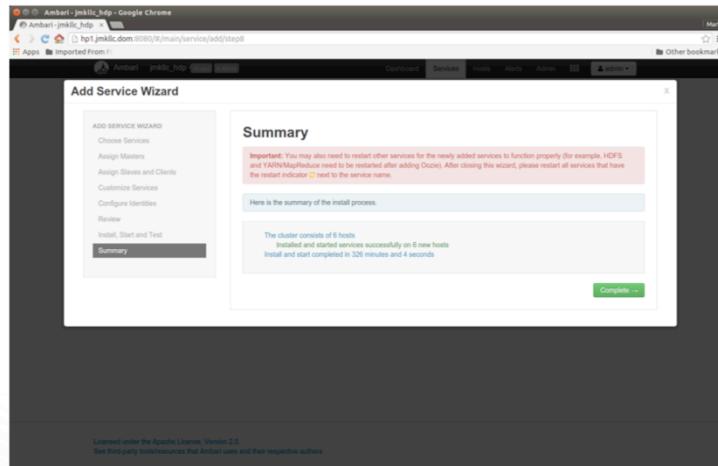
## Install Apache Solr with Ambari

The screenshot shows a web browser window for the Ambari interface. The title bar reads "Ambari - Jenkins - Google Chrome". The address bar shows "http://192.168.1.100:8080/0/main/service/add?step=7". The main content area is titled "Add Service Wizard" and "Install, Start and Test". It displays a progress bar at 100% overall. Below it is a table with columns "Host", "Status", and "Message". The table shows six hosts, each with a green bar indicating 100% success. The "Message" column for all hosts shows "Success". At the bottom of the table, there is a message: "Successfully installed and started the services." A "Next ..." button is visible at the bottom right of the table area.

312 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Solr with Ambari



313 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Setup Apache Solr for Apache Ranger Audits

- Download the solr\_for\_audit\_setup\_v3 file to the /usr/local/ directory:  

```
wget https://issues.apache.org/jira/secure/attachment/12761323/solr_for_audit_setup_v3.tgz -O /usr/local/solr_for_audit_setup_v3.tgz
```
- Switch to the /usr/local/ directory and extract the solr\_for\_audit\_setup\_v3 file  

```
cd /usr/local  
tar xvf solr_for_audit_setup_v3.tgz
```
- Extract into a /usr/local/solr\_for\_audit\_setup\_v3 directory
- Switch to the /usr/local/solr\_for\_audit\_setup\_v3 directory  

```
cd /usr/local/solr_for_audit_setup
```
- Open the install.properties file in the vi text editor.  

```
vi install.properties
```
- Set property values, save the changes to the install.properties file.



314 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

Use the following procedure to configure SolrCloud.

Download the solr\_for\_audit\_setup\_v3 file to the /usr/local/ directory:

```
wget https://issues.apache.org/jira/secure/attachment/12761323/  
solr_for_audit_setup_v3.tgz -O /usr/local/solr_for_audit_setup_v3.tgz
```

Use the following commands to switch to the /usr/local/ directory and extract the solr\_for\_audit\_setup\_v3 file.

```
cd /usr/local
```

```
tar xvf solr_for_audit_setup_v3.tgz
```

The contents of the .tgz file will be extracted into a /usr/local/solr\_for\_audit\_setup\_v3 directory.

Use the following command to switch to the /usr/local/solr\_for\_audit\_setup\_v3

## Setup Apache Solr for Apache Ranger Audits

```
● JAVA_HOME=$JAVA_HOME
● SOLR_USER=solr
● SOLR_INSTALL=false
● SOLR_INSTALL_FOLDER=/opt/lucidworks-hdpsearch/solr
● SOLR_RANGER_HOME=/opt/ranger_audit_server
● SOLR_RANGER_PORT=6083
● SOLR_DEPLOYMENT=solrcloud
● SOLR_ZK=localhost:2181/ranger_audits
● SOLR_HOST_URL=http://$host:${SOLR_RANGER_PORT}
● SOLR_SHARDS=1
● SOLR_REPLICATION=2
● SOLR_LOG_FOLDER=/var/log/solr/ranger_audits
● SOLR_MAX_MEM=1g
```

315 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Use the following procedure to configure SolrCloud.

Download the `solr_for_audit_setup_v3` file to the `/usr/local/` directory:

```
wget https://issues.apache.org/jira/secure/attachment/12761323/
solr_for_audit_setup_v3.tgz -O /usr/local/solr_for_audit_setup_v3.tgz
```

Use the following commands to switch to the `/usr/local/` directory and extract the `solr_for_audit_setup_v3` file.

```
cd /usr/local
```

```
tar xvf solr_for_audit_setup_v3.tgz
```

The contents of the `.tgz` file will be extracted into a `/usr/local/solr_for_audit_setup_v3` directory.

Use the following command to switch to the `/usr/local/solr_for_audit_setup_v3`

## Setup Apache Solr for Apache Ranger Audits

- Use the following command to run the set up script.  
./setup.sh
- Run the following command only once from any node to add the Ranger Audit configuration (including schema.xml) to ZooKeeper:  
/opt/lucidworks-hdpsearch/solr/ranger\_audit\_server/scripts/add\_ranger\_audits\_conf\_to\_zk.sh
- Run the following command only once from any node to create the Ranger Audit collection:  
/opt/lucidworks-hdpsearch/solr/ranger\_audit\_server/scripts/create\_ranger\_audits\_collection.sh
- You can use a web browser to open the Solr Admin Console at the following address:  
[http:<solr\\_host>:6083/solr](http://<solr_host>:6083/solr)

316 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Use the following procedure to configure SolrCloud.

Download the solr\_for\_audit\_setup\_v3 file to the /usr/local/ directory:

```
wget https://issues.apache.org/jira/secure/attachment/12761323/
solr_for_audit_setup_v3.tgz -O /usr/local/solr_for_audit_setup_v3.tgz
```

Use the following commands to switch to the /usr/local/ directory and extract the solr\_for\_audit\_setup\_v3 file.

```
cd /usr/local
```

```
tar xvf solr_for_audit_setup_v3.tgz
```

The contents of the .tgz file will be extracted into a /usr/local/solr\_for\_audit\_setup\_v3 directory.

Use the following command to switch to the /usr/local/solr\_for\_audit\_setup\_v3

## Apache SolrCloud

- SolrCloud provides support for
  - Distributing Index Process and Queries Automatically
  - ZooKeeper provides failover and load balancing
  - Every shard can have multiple replicas for robustness
- In Cloud Mode, there are Leaders and Replicas
  - Leaders are automatically elected
  - Initially on a first-come-first-served basis
- Leader goes down,
  - one of its replicas is automatically elected as the new leader.
- As each node is started, it's assigned to the shard with the fewest replicas

317 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



SolrCloud provides support for distributing both the index process and the queries automatically, and ZooKeeper provides failover and load balancing. Additionally, every shard can also have multiple replicas for additional robustness.

In Solr Cloud Mode, there are leaders and replicas. Leaders are automatically elected, initially on a first-come-first-served basis, and then based on the Zookeeper process described at [http://zookeeper.apache.org/doc/trunk/recipes.html#sc\\_leaderElection..](http://zookeeper.apache.org/doc/trunk/recipes.html#sc_leaderElection..)

If a leader goes down, one of its replicas is automatically elected as the new leader. As each node is started, it's assigned to the shard with the fewest replicas. When there's a tie, it's assigned to the shard with the lowest shard ID.

When a document is sent to a machine for indexing, the system first determines if the machine is a replica or a leader.

If the machine is a replica, the document is forwarded to the leader for processing.

If the machine is a leader, SolrCloud determines which shard the document should go to, forwards the document to the leader for that shard, indexes the document for this

## Apache SolrCloud

- When document is sent to node for indexing...
  - System determines if its a Replica or a Leader
- If Replica:
  - Document forwarded to Leader for processing
- If Leader:
  - Determines which shard to forward document to
  - Forwards document to Leader for that shard
  - Indexes document for this shard
  - Forwards index notation to itself and any Replicas

318 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



SolrCloud provides support for distributing both the index process and the queries automatically, and ZooKeeper provides failover and load balancing. Additionally, every shard can also have multiple replicas for additional robustness.

In Solr Cloud Mode, there are leaders and replicas. Leaders are automatically elected, initially on a first-come-first-served basis, and then based on the Zookeeper process described at [http://zookeeper.apache.org/doc/trunk/recipes.html#sc\\_leaderElection..](http://zookeeper.apache.org/doc/trunk/recipes.html#sc_leaderElection..)

If a leader goes down, one of its replicas is automatically elected as the new leader. As each node is started, it's assigned to the shard with the fewest replicas. When there's a tie, it's assigned to the shard with the lowest shard ID.

When a document is sent to a machine for indexing, the system first determines if the machine is a replica or a leader.

If the machine is a replica, the document is forwarded to the leader for processing.

If the machine is a leader, SolrCloud determines which shard the document should go to, forwards the document to the leader for that shard, indexes the document for this

# Apache Solr Cloud Leader

The screenshot shows the Apache Solr Cloud Leader interface. At the top, there's a navigation bar with icons for back, forward, search, and other browser functions. The URL is 52.37.1.122:6083/solr/#/~cloud. Below the header is a sidebar with the Solr logo and a list of tabs: Dashboard, Logging, Cloud (selected), Tree, Graph (selected), Graph (Radial), Dump, Core Admin, Java Properties, and Thread Dump. A "Core Selector" dropdown is also present. The main area displays a network graph with nodes labeled "ranger\_audits", "shard1", and two IP addresses: "172.30.0.47" and "172.30.0.133". A legend on the right side defines node states: ● Leader (black dot), ○ Active (green outline), ○ Recovering (yellow outline), ○ Down (orange outline), ○ Recovery Failed (red outline), and ○ Gone (grey outline). At the bottom of the page, there are links for Documentation, Issue Tracker, IRC Channel, Community forum, and Solr Query Syntax. A small "UNIVERSITY" logo is in the bottom right corner. The footer contains the text "319 © Hortonworks Inc. 2011–2016. All Rights Reserved".

Click the Cloud > Graph tab to find the leader host (172.30.0.242 in below example)

## Objectives



- What is Apache Ranger?
- Why is Apache Ranger Needed?
- Apache Ranger Architecture
- Prerequisites for Apache Ranger
- Install and Configure MySQL Database
- Install and Configure Apache Solr
- Install and Configure Apache Ranger

320 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Ranger with Ambari

- Install Ranger Via Add Service Wizard Under “Admin > Stack and Versions”
- Add Service Wizard – Select Ranger
- Choose Services – Confirm Ranger is Selected
- Assign Master – Choose Hosts to Run Ranger
- Customize Services – See Following Slides
- Review Configuration – Deploy
- Monitor Deployment – Install, Start and Test
- Review Summary
- Installation Complete

321 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



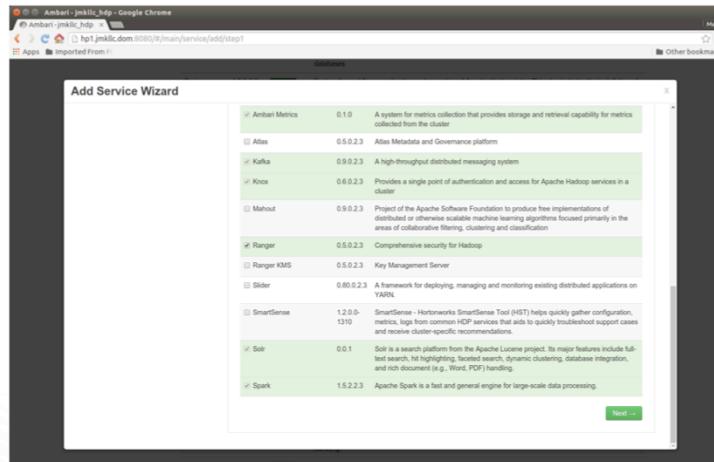
## Install Apache Ranger with Ambari

Service	Version	Description
HDFS	2.7.1.2.3	Apache Hadoop Distributed File System
YARN + MapReduce2	2.7.1.2.3	Apache Hadoop NextGen MapReduce (YARN)
Tez	0.7.0.2.3	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
Hive	1.2.1.2.3	Apache Hive provides support for ad-hoc queries & analysis of large datasets and table & storage management services.
HBase	1.1.1.2.3	A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications.
Pig	0.15.0.2.3	Scripting platform for analyzing large datasets
Sqoop	1.4.8.2.3	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
Oozie	4.2.0.2.3	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the Eclipse Library.
ZooKeeper	3.4.6.2.3	Centralized service which provides highly reliable distributed coordination
Falcon	0.6.1.2.3	Data management and processing platform
Storm	0.10.0	Apache Hadoop Stream processing framework

322 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



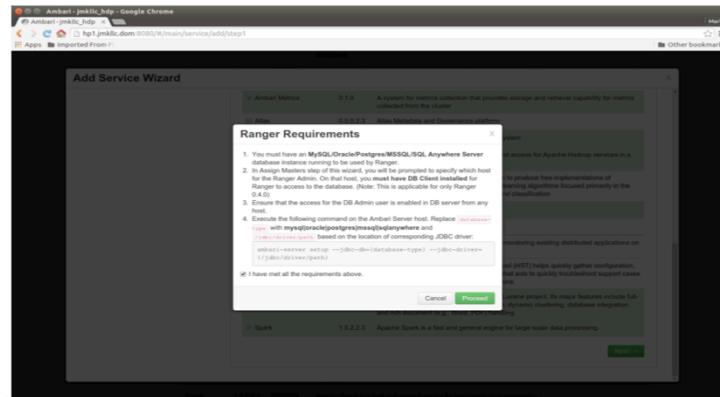
## Install Apache Ranger with Ambari



323 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Ranger with Ambari



324 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Ranger with Ambari

325 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

**Ambari - jmklic\_hdp - Google Chrome**  
http://192.168.1.137:8080/main/service/add/step2  
Imported From: /

Add Service Wizard

Choose Services

Assign Masters

Assign Slaves and Clients

Customize Services

Configure Identities

Review

Install, Start and Test

Summary

Assign Masters

NameNode: del2.prdlc.dom (15.6 GB, 8 cores)

NameNode: del1.prdlc.dom (15.6 GB, 8 cores)

History Server: del2.prdlc.dom (15.6 GB, 8 cores)

ResourceManager: del1.prdlc.dom (15.6 GB, 8 cores)

ResourceManager: del2.prdlc.dom (15.6 GB, 8 cores)

App Timeline Server: del2.prdlc.dom (15.6 GB, 8 cores)

HiveServer2: del1.prdlc.dom (15.6 GB, 8 cores)

WebHCat Server: del2.prdlc.dom

Hive Metastore: del4.prdlc.dom (15.6 GB, 8 cores)

Hive Metastore: del3.prdlc.dom (15.6 GB, 8 cores)

Hbase Master: del1.prdlc.dom (15.6 GB, 8 cores)

Oozie Server: del2.prdlc.dom (15.6 GB, 8 cores)

Assign master components to hosts you want to run them on.

del1.prdlc.dom (15.6 GB, 8 cores)  
NameNode, History Server, ResourceManager, App Timeline Server, Oozie Server, WebHCat Server

del2.prdlc.dom (15.6 GB, 8 cores)  
NameNode, History Server, ResourceManager, App Timeline Server, Oozie Server, WebHCat Server

del3.prdlc.dom (15.6 GB, 8 cores)  
Hive Metastore, Zookeeper Server, Metrics Collector

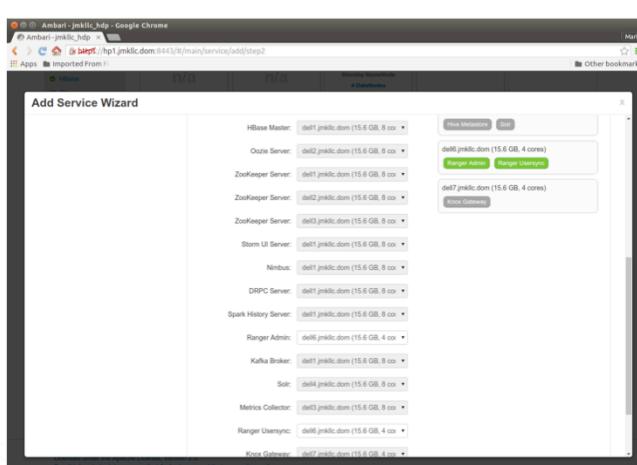
del4.prdlc.dom (15.6 GB, 8 cores)  
Hive Metastore, Oozie

Ranger Admin

Change Username

UNIVERSITY

## Install Apache Ranger with Ambari

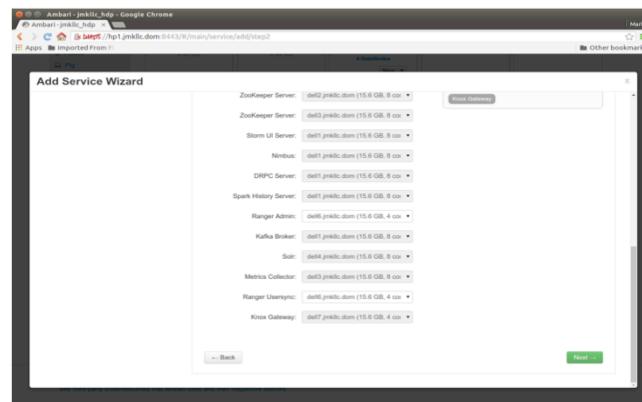


The screenshot shows the Ambari 'Add Service Wizard' interface. The main window displays a list of services with their assigned hosts. The 'Ranger Admin' service is highlighted with a green border, indicating it is currently selected or being configured. Other services listed include HBase Master, Oozie Server, ZooKeeper Server, ZooKeeper Server, Storm UI Server, Nimbus, DRPC Server, Spark History Server, Ranger Admin, Kafka Broker, Solr, Metrics Collector, Ranger UserSync, and Knox Gateway. Each service has a dropdown menu next to its name, likely for selecting host and configuration options.

326 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Ranger with Ambari



327 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Ranger with Ambari

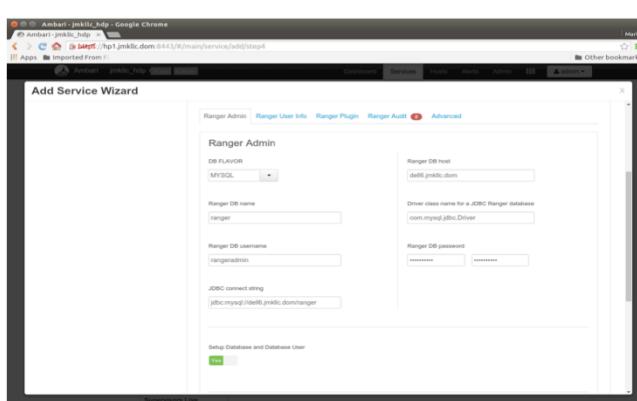
The screenshot shows the Ambari Add Service Wizard at Step 4, titled "Customize Services". The left sidebar lists steps: Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services (selected), Configure Identities, Review, Install, Start and Test, and Summary. The main panel shows a list of services: HDFS, MapReduce2, YARN, Tez, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Storm, Ambari Metrics, Kafka, Knox, Ranger (selected), Solr, Spark, and Mac. A message indicates 38 configuration changes in 7 services. The Ranger Admin configuration section shows "DB FLAVOR" set to "MySQL", "Ranger DB host" input field, and "Driver class name for a JDBC Ranger database" set to "com.mysql.jdbc.Driver". Navigation tabs include Ranger Admin, Ranger User Info, Ranger Plugin, Ranger Audit, and Advanced.

Ranger DB Host = FQDN of host where Mysql is running (e.g. ip-172-30-0-242.us-west-2.compute.internal)

Enter passwords



## Install Apache Ranger with Ambari



The screenshot shows the 'Add Service Wizard' interface for installing Apache Ranger. The current step is 'Ranger Admin'. The configuration fields are as follows:

- Ranger Admin**
  - DB FLAVOR: MYSQL
  - Ranger DB name: ranger
  - Ranger DB username: rangeradmin
  - JDBC connect string: jdbc:mysql://localhost:3306/ranger
- Ranger DB host**: localhost
- Driver class name for a JDBC Ranger database**: com.mysql.jdbc.Driver
- Ranger DB password**: (two masked input fields)

At the bottom, there is a 'Setup Database and Database User' button.

At the bottom left: 329 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

At the bottom right: Hadoop University logo

Ranger DB Host = FQDN of host where Mysql is running (e.g. ip-172-30-0-242.us-west-2.compute.internal)

Enter passwords

## Install Apache Ranger with Ambari

Ambari - jmklic\_hdp - Google Chrome  
http://hp1.jmklic.com:8443/#/main/service/add/step4  
Add Service Wizard  
Setup Database and Database User  
Database Administrator (DBA) username: root  
Database Administrator (DBA) password: password  
JDBC connect string for root user: jdbc:mysql://dell1.jmklic.com  
Test Connection Connection OK  
Attention: Some configurations need your attention before you can proceed.  
Show me properties with issues  
Back Next  
4/4

330 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Enter Passwords - BadPass#1

Test Connection

## Install Apache Ranger with Ambari

The screenshot shows the Ambari web interface with a green header bar. The main content area has a title 'Install Apache Ranger with Ambari'. Below the title is a sub-header 'Add Service Wizard' with a sidebar on the left containing steps: 'Choose Services', 'Assign Masters', 'Select Slaves and Clients', 'Customize Services' (which is selected), 'Configure Identites', 'Review', 'Install, Start and Test', and 'Summary'. The main panel is titled 'Customize Services' and displays a message: 'We have come up with recommended configurations for the services you selected. Customize them as you see fit.' It lists several services: HDFS, MapReduce2, YARN, Tez, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Storm, and Ambari Metrics. Below this is a note: 'There are 41 configuration changes in 8 services Show Details'. A dropdown menu shows 'Group: Default (6)'. At the bottom, there are tabs: Ranger Admin, Ranger User Info (with a red exclamation mark), Ranger Plugin, Ranger Audit (with a red exclamation mark), and Advanced.

'Sync Source' = AD/LDAP  
Common configs subtab  
Enter password



## Install Apache Ranger with Ambari

Sync Source  
LDAP/AD

Common Configs User Config Group Configs

LDAP URL  
ldap://ad-server.lab.hortonworks.com:389

Bind Anonymous

Bind User  
cnldap-reader,ou=ServiceUsers,dc=lab,dc=hortonworks,dc=com

Bind User Password

Attention: Some configurations need your attention before you can proceed.  
Show me properties with issues

'Sync Source' = AD/LDAP  
Common configs subtab  
Enter password

## Install Apache Ranger with Ambari

The screenshot shows the Ambari interface with the title "Install Apache Ranger with Ambari". A subwindow titled "Add Service Wizard" is open, specifically the "User Configs" tab. The configuration fields shown are:

- User Account Attribute: sAMAccountName
- User Object Class: user
- User Search Base: ou=CorpUsers,dc=lab,dc=hortonworks,dc=net
- User Search Filter: (objectcategory=person)
- User Search Scope: sub
- User Group Name Attribute: memberof, ismemberof

At the bottom of the subwindow, there is a link "Supervisors Live". In the bottom right corner of the main Ambari window, there is a green hexagonal badge with the text "Hortonworks UNIVERSITY".

User configs subtab

User Search Base = ou=CorpUsers,dc=lab,dc=hortonworks,dc=net

User Search Filter = (objectcategory=person)

## Install Apache Ranger with Ambari

The screenshot shows the Ambari web interface with the title "Install Apache Ranger with Ambari". The main content area displays the "Add Service Wizard" for the "Ranger" service. The "Ranger" service is highlighted in red. Below the service list, a message states "There are 41 configuration changes in 8 services Show Details". The "Manage Config Groups" tab is selected. In the "Ranger Plugin" section, there are six listed plugins, each with a small icon and a status indicator: HDFS Ranger Plugin (disabled), Hive Ranger Plugin (disabled), Knox Ranger Plugin (disabled), YARN Ranger Plugin (disabled), Storm Ranger Plugin (disabled), and Kafka Ranger Plugin (disabled).

Plugins disabled

## Install Apache Ranger with Ambari

The screenshot shows the Ambari web interface with the title "Install Apache Ranger with Ambari". The main content area displays the "Add Service Wizard" for Ranger. In the "Ranger Plugin" section, there are five categories of Ranger Plugins: HDFS Ranger Plugin, YARN Ranger Plugin, Hive Ranger Plugin, Storm Ranger Plugin, and Knox Ranger Plugin. Each category has a green "Enabled" button next to its name. At the top of the page, a message indicates "There are 62 configuration changes in 8 services Show Details". Below the message, there is a "Manage Config Groups" dropdown and a "Filter..." input field. The footer of the page includes the text "335 © Hortonworks Inc. 2011 – 2016. All Rights Reserved" and a small Hortonworks logo.

Enable all plugins

## Install Apache Ranger with Ambari

The screenshot shows the Ambari interface during the 'Add Service Wizard'. The 'Ranger' service is selected for configuration. The 'Audit to Solr' section has 'Audit to Solr' turned on and 'SolrCloud' set to 'ON'. The 'Audit to HDFS' section has 'Audit to HDFS' turned on and 'Destination HDFS Directory' set to 'hdfs://jmklc1/rangeraudit'. A message at the top indicates 62 configuration changes across 8 services. The bottom left shows the copyright notice '© Hortonworks Inc. 2011 – 2016. All Rights Reserved'.

SolrCloud = ON

enter password

## Install Apache Ranger with Ambari

337 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

SolrCloud = ON

enter password

## Install Apache Ranger with Ambari

Add Service Wizard

Ranger Admin   Ranger User Info   Ranger Plugin   Ranger Audit   Advanced

**Audit to Solr**

Audit to Solr  
 ON

SolrCloud  
 ON

ranger\_audit.solr.zookeepers  
ip-172-30-0-105.us-west-2.compute.int

ranger\_audit.solr.username  
ranger\_solr

ranger\_audit.solr.password  
\*\*\*\*

**Audit to HDFS**

Audit to HDFS  
 ON

Destination HDFS Directory  
hdfs://ip-172-30-0-104.us-west-2.comp

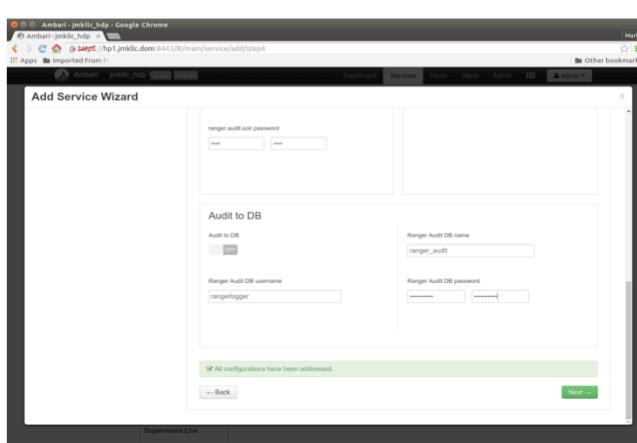
338 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



SolrCloud = ON

enter password

## Install Apache Ranger with Ambari



The screenshot shows the Ambari Add Service Wizard step 4, titled "Audit to DB". It displays fields for "ranger\_audit\_password" (with values "audit" and "audit"), "Audit to DB" (selected), "Ranger Audit DB name" (set to "ranger\_audit"), "Ranger Audit DB username" (set to "rangerlogger"), and "Ranger Audit DB password" (with values "password" and "password"). A green status bar at the bottom indicates "All configurations have been addressed." Navigation buttons "Back" and "Next >" are visible.

339 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



The screenshot shows the Ambari 'Add Service Wizard' interface. The title bar says 'Install Apache Ranger with Ambari'. The left sidebar lists steps: 'ADD SERVICE WIZARD', 'Choose Services', 'Assign Masters', 'Assign Slaves and Clients', 'Customize Services' (which is selected and highlighted in dark grey), 'Configure Identities', 'Review', 'Install, Start and Test', and 'Summary'. The main panel is titled 'Customize Services' and contains the following content:

- A message: 'We have come up with recommended configurations for the services you selected. Customize them as you see fit.'
- A navigation bar with tabs: HDFS, MapReduce2, YARN, Tez, Hive, HBase, Pig, ZooKeeper, Ambari Metrics, Knox, **Ranger**, Solr, Misc. The 'Ranger' tab is selected.
- A message: 'There are 23 configuration changes in 5 services' with a 'Show Details' link.
- A search/filter bar with 'Group' dropdown set to 'Default (4)', 'Manage Config Groups' button, and 'Filter...' dropdown.
- Two tabs at the bottom: 'Settings' (selected) and 'Advanced'.
- A collapsed section header: '▼ NameNode'.

The footer of the wizard window includes the text '340 © Hortonworks Inc. 2011 – 2016. All Rights Reserved' and the Hortonworks University logo.

Now configure Hadoop components so Ranger can use rangeradmin@LAB.HORTONWORKS.NET principal to query HDFS, YARN, Hive, Hbase, Knox.

We will do this by clicking the tabs for each of these services and modifying Ranger specific properties.

## Install Apache Ranger with Ambari

Advanced ranger-hdfs-plugin-properties

Enable Ranger for HDFS

Ranger repository config password

Ranger repository config user

common.name.for. certificate

hadoop.rpc.protection

Policy user for HDFS

341 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Now configure HDFS so Ranger can use rangeradmin@LAB.HORTONWORKS.NET principal to query HDFS

## Install Apache Ranger with Ambari

▼ Advanced ranger-hive-plugin-properties

Ranger repository config password	.....	.....	lock	refresh	
Ranger repository config user	rangeradmin@LAB.HORTONWORKS.NET		lock	refresh	cancel
common.name.for. certificate		lock	refresh		
jdbc.driverClassName	org.apache.hive.jdbc.HiveDriver	lock	refresh	cancel	
Policy user for HIVE	rangeradmin	lock	refresh	refresh	cancel

342 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Now configure Hive so Ranger can use rangeradmin@LAB.HORTONWORKS.NET principal to query Hive

## Install Apache Ranger with Ambari

▼ Advanced ranger-hbase-plugin-properties

Enable Ranger for HBASE

Ranger repository config password

Ranger repository config user

common.name.for. certificate

Policy user for HBASE

343 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Now configure Hbase so Ranger can use rangeradmin@LAB.HORTONWORKS.NET principal to query Hbase

## Install Apache Ranger with Ambari

Advanced ranger-yarn-plugin-properties

Enable Ranger for YARN

Ranger repository config password

Ranger repository config user

common.name.for.certificate

hadoop.rpc.protection

Policy user for YARN

344 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

Now configure YARN so Ranger can use rangeradmin@LAB.HORTONWORKS.NET principal to query YARN

## Install Apache Ranger with Ambari

Advanced ranger-knox-plugin-properties

Enable Ranger for KNOX

Knox Home

Ranger repository config password

Ranger repository config user

common.name.for. certificate

Policy user for KNOX

345 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

Now configure KNOX so Ranger can use rangeradmin@LAB.HORTONWORKS.NET principal to query KNOX

## Install Apache Ranger with Ambari

346 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

After clicking Next, Ambari will prompt for:

Admin principal: hadoopadmin@LAB.HORTONWORKS.NET

Admin password: BadPass#1

## Install Apache Ranger with Ambari

The screenshot shows a web browser window titled "Ambari - jmklic\_hdp - Google Chrome". The URL is "http://192.168.1.10:8080/main/service/add/step/1". The main content is the "Add Service Wizard" with the "Review" step selected. The review screen displays configuration details:

- Admin Name:** admin
- Cluster Name:** jmklic\_hdp
- Total Hosts:** 6 (0 new)
- Repositories:**
  - hortonworks/HDP-2.3:  
http://192.168.1.10:20070/repos/hortonworks/HDP-2.3/updates/2.3.4.0/
  - hortonworks/HDP-UTILS-1.1.0.20:  
http://192.168.1.10:20070/repos/centos/
- Services:**
  - Ranger**
    - Admin: dell1.jmklic.dom
    - User: dell1.jmklic.dom

At the bottom of the review screen, there are three buttons: "Back", "Download CSV", "Find", and "Deploy". Below the review screen, a note states: "Licensed under the Apache License, Version 2.0. See third party license resources that Ambari uses and their respective authors".

In the bottom right corner of the Ambari interface, there is a small green hexagonal icon with the text "Hortonworks UNIVERSITY" and a graduation cap graphic.

347 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Install Apache Ranger with Ambari

The screenshot shows the Ambari web interface with the title "Install Apache Ranger with Ambari". The main window is titled "Add Service Wizard" and is currently on the "Install, Start and Test" step. The interface displays a table of host statuses:

Host	Status	Message
del1.jnklic.dom	100%	Success
del2.jnklic.dom	100%	Success
del3.jnklic.dom	100%	Success
del4.jnklic.dom	100%	Success
del5.jnklic.dom	100%	Success
del6.jnklic.dom	100%	Success

A green banner at the bottom of the table area says "Successfully installed and started the services." At the bottom left of the main window, it says "Downloaded under the Apache License, Version 2.0. See [http://www.apache.org/licenses/LICENSE-2.0.html](#) for more information." On the right side of the interface, there is a small "Hortonworks UNIVERSITY" logo.

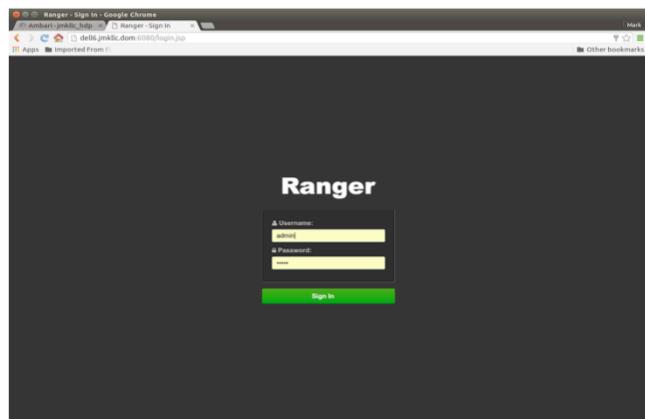
348 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Install Apache Ranger with Ambari

The screenshot shows a browser window titled "Ambari - jmklic.hdp - Google Chrome". The URL is "http://jmklc.hdp:8443/navegator/main/service/add/step8". The main content is the "Add Service Wizard" with the "Summary" step selected. On the left, there's a sidebar with steps: ADD SERVICE WIZARD, Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Configure Identities, Review, Install, Start and Test, and finally Summary. The Summary section contains a message: "Important: You may also need to restart other services for the newly added services to function properly (for example, HDFS and YARN/MapReduce need to be restarted after Oozie). After closing this wizard, please restart all services that have the restart indicator (●) next to the service name." Below this, it says "Here is the summary of the install process." and "The cluster consists of 6 hosts: Installed and started services successfully on 6 new hosts. Install and start completed in 2 minutes and 40 seconds." At the bottom right of the summary box is a "Complete" button. In the bottom left corner of the main area, it says "Supervisors Live" and "4/4". At the very bottom of the page, it says "Licensed under the Apache License, Version 2.0. See third-party license resources that Ambari uses and their respective authors." To the right of the main content area, there is a small "Hortonworks UNIVERSITY" logo.

Point out that Services will need to be restarted for the Ranger Plugins show up in the Ranger UI

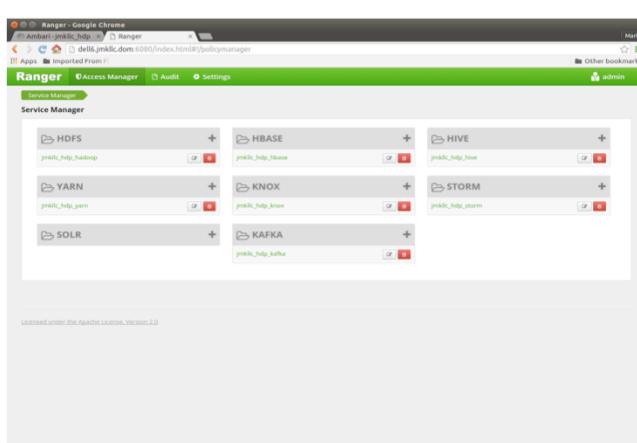
## Install Apache Ranger with Ambari



350 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Ranger with Ambari



The screenshot shows the Apache Ranger Service Manager interface. At the top, there are tabs for Service Manager, Access Manager, Audit, and Settings. The Service Manager tab is selected. Below the tabs, there are six service groups: HDFS, YARN, SOLR, HBASE, KNOX, and KAFKA. Each group has a plus sign (+) icon to its right, indicating it can be expanded. Under each group, there are several sub-components listed, each with a red minus sign (-) icon to its right, suggesting they can be removed or collapsed. The background of the interface is light gray, and the overall design is clean and modern.

351 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Objectives



- What is Apache Ranger?
- Why is Apache Ranger Needed?
- Apache Ranger Architecture
- Prerequisites for Apache Ranger
- Install and Configure MySQL Database
- Install and Configure Apache Solr
- Install and Configure Apache Ranger
- Apache Ranger REST API



## Apache Ranger REST API

- Sample REST API calls - more details available on Apache Ranger wiki

- GET request to *fetch details for policy* with ID = 2:

```
curl -iv -u <user>:<password> -H "Content-type:application/json" -X GET http://<RANGER_HOST>:6080/service/public/api/policy/2
```

```
{  
  "id":2, "createDate":"2015-11-21T07:03:21Z", "updateDate":"2015-12-08T05:54:24Z",  
  "owner":"Admin", "updatedBy":"Admin", "policyName":"Ranger_audits", "description": "",  
  "repositoryName":"bigdata_hadoop", "repositoryType":"hdfs", "resourceName":"/apps/solr/ranger_audits",  
  "permMapList": [  
    {  
      "userList":["solr"],  
      "groupList":[ ],  
      "permList":["Read","Write","Execute"]  
    }  
  ],  
  "isEnabled":true, "isRecursive":true, "isAuditEnabled":false, "version":"5",  
  "replacePerm":false  
}
```

353 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



<http://ranger.apache.org>

How does it work over Hadoop and related components

Apache Ranger's solution at the core has a centralized web application, which consists of the policy administration, audit and reporting modules. Authorized users will be able to manage their security policies using the web tool or using REST APIs. These security policies are enforced within Hadoop ecosystem using lightweight Ranger Java plugins, which run as part of the same process as the namenode (HDFS), Hive2Server(Hive), HBase server (Hbase), Nimbus server (Storm) and Knox server (Knox) respectively. Thus there is no additional OS level process to manage.

Is there a single point of failure?

No, Apache Ranger is not a Single Point of Failure. Ranger's plugins run within the same process as the component, e.g. NameNode for HDFS. These agents pull the policy-changes using REST API at a configured regular interval (e.g.: 30 second). The plugin is able to function even if the policy server is temporarily down and will provide the authorization enforcement. Also, the policy manager web application can

## Apache Ranger REST API

- POST request to create HDFS policy:

```
curl -iv -u <user>:<password> -d @<policy payload> -H "Content-Type: application/json" -X POST http://<RANGER-Host>:6080/service/public/api/policy
```

- Payload to be passed in:

```
{  
    "policyName": "name_of_policy",  
    "resourceName": "/path1,/path2/blub",  
    "description": "", "repositoryName": "",  
    "repositoryType": "hdfs",  
    "isEnabled": "true", "isRecursive": "true", "isAuditEnabled": "true",  
    "permMapList": [{  
        "groupList": ["somegroup"],  
        "permList": ["Read", "Execute", "Write", "Admin"]  
    }]  
}
```

354 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



<http://ranger.apache.org>

How does it work over Hadoop and related components

Apache Ranger's solution at the core has a centralized web application, which consists of the policy administration, audit and reporting modules. Authorized users will be able to manage their security policies using the web tool or using REST APIs. These security policies are enforced within Hadoop ecosystem using lightweight Ranger Java plugins, which run as part of the same process as the namenode (HDFS), Hive2Server(Hive), HBase server (Hbase), Nimbus server (Storm) and Knox server (Knox) respectively. Thus there is no additional OS level process to manage.

Is there a single point of failure?

No, Apache Ranger is not a Single Point of Failure. Ranger's plugins run within the same process as the component, e.g. NameNode for HDFS. These agents pull the policy-changes using REST API at a configured regular interval (e.g.: 30 second). The plugin is able to function even if the policy server is temporarily down and will provide the authorization enforcement. Also, the policy manager web application can

## Apache Ranger REST API

- POST request to create *Hive* policy:

```
curl -iv -u <user>:<password> -d @<policy payload> -H "Content-Type: application/json" -X POST http://<RANGER-Host>:6080/service/public/api/policy
```

- Payload to be passed in:

```
{  
    "policyName": "name_of_policy",  
    "databases": "db1,db2",  
    "tables": "mytable,yourtable", "columns": "",  
    "udfs": "", "description": "", "repositoryName": "",  
    "repositoryType": "hive", "tableType": "Inclusion", "columnType": "Inclusion",  
    "isEnabled": "true", "isAuditEnabled": "true",  
    "permMapList": [{  
        "groupList": ["somegroup"],  
        "permList": ["Select"]  
    }]  
}
```

355 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



<http://ranger.apache.org>

How does it work over Hadoop and related components

Apache Ranger's solution at the core has a centralized web application, which consists of the policy administration, audit and reporting modules. Authorized users will be able to manage their security policies using the web tool or using REST APIs. These security policies are enforced within Hadoop ecosystem using lightweight Ranger Java plugins, which run as part of the same process as the namenode (HDFS), Hive2Server(Hive), HBase server (Hbase), Nimbus server (Storm) and Knox server (Knox) respectively. Thus there is no additional OS level process to manage.

Is there a single point of failure?

No, Apache Ranger is not a Single Point of Failure. Ranger's plugins run within the same process as the component, e.g. NameNode for HDFS. These agents pull the policy-changes using REST API at a configured regular interval (e.g.: 30 second). The plugin is able to function even if the policy server is temporarily down and will provide the authorization enforcement. Also, the policy manager web application can

## Objectives



- What is Apache Ranger?
- Why is Apache Ranger Needed?
- Apache Ranger Architecture
- Prerequisites for Apache Ranger
- Install and Configure MySQL Database
- Install and Configure Apache Solr
- Install and Configure Apache Ranger
- Apache Ranger REST API
- Optional Apache Ranger Configurations



## Optional Apache Ranger Configurations

- Supported Databases

- MySQL
- PostgreSQL
- Oracle
- MS SQL Server

- Ranger Authentication - Access to Console

- UNIX
- Active Directory
- LDAP/LDAPS

- User Synchronization

- UNIX Usersync
- LDAP Usersync
- AD Usersync

357 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Knowledge Check



## Knowledge Check

1. Which of the 5 Security Pillars does Apache Ranger address?
2. Apache Ranger provides a framework to do what?
3. Apache Ranger provide enhanced support for Authorization Methods, name the two methods?
4. True/False - Ranger provides centralized auditing of all user access and administrative action with in all the Hadoop components?



1. Administration, Authorization, Audit and Data Protection (Ranger KMS)
2. Enable, Monitor and Manage comprehensive data security across the Hadoop platform
3. Role Based Access Control and Attribute Bases Access Control
4. TRUE

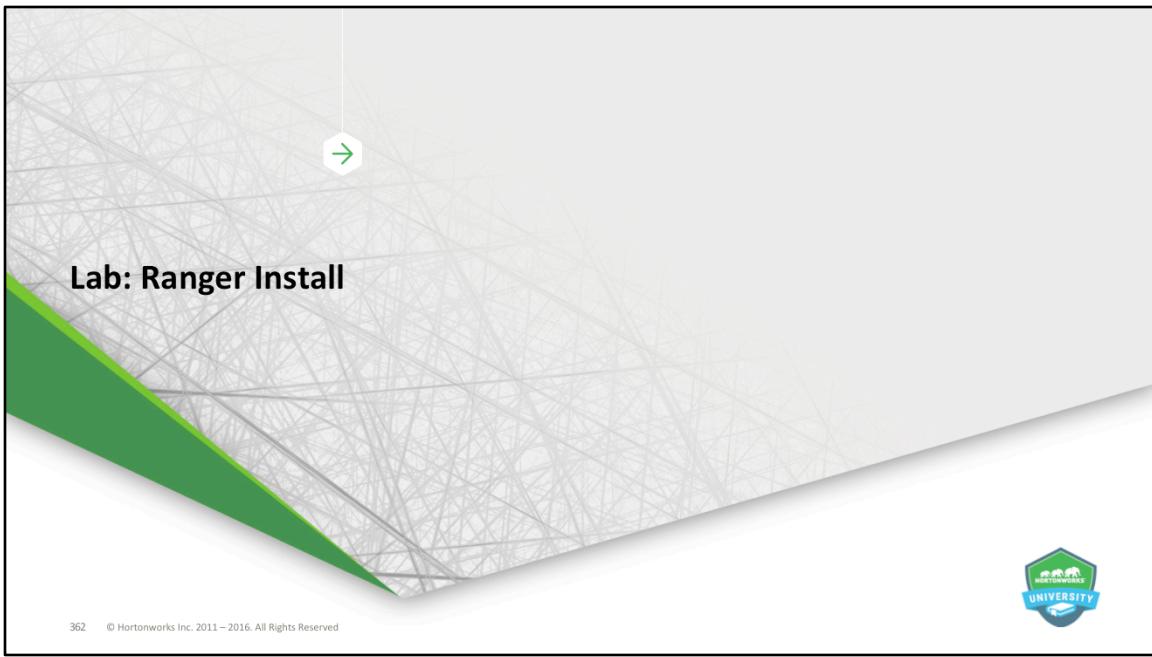
# Summary



## Summary

- Apache Ranger Provides a Framework to Enable, Monitor, & Manage Comprehensive Data Security Across the Apache Hadoop Platform
- Apache Ranger has Centralized Security Administration to Manage All Security Related Tasks in a Central UI or using REST APIs.
- Centralized Security Administration has Four Aspects: Authentication, Authorization, Audit and Data Protection
- Can Enable Audit Tracking and Policy Analytics for Deeper Control of the Hadoop environment
- Provides Ability to Delegate Administration of Certain Data to Other Group Owners, with the Aim of Decentralizing Data Ownership







## Apache Ranger KMS and HDFS Encryption



## Objectives

- Describe Apache Ranger KMS
- Describe HDFS Encryption
- Install and Configure Apache Ranger KMS
- Configure HDFS to Use Encryption
- Describe HDFS Encryption Zones



## Objectives



- What is Apache Ranger KMS

365 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## What is Apache Ranger KMS

- Open Source, Scalable Cryptographic Key Management Service
- Supports HDFS "Data at Rest" Encryption
- Ranger KMS Based on Hadoop KMS Originally Developed by the Apache Community
- Hadoop KMS Stores Keys by Default in a File-Based Java Keystore
- Ranger Extends Native Hadoop KMS Functionality by Storing Keys in a Secure Database

366 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Ranger Key Management Service (Ranger KMS) is a open source, scalable cryptographic key management service supporting HDFS "data at rest" encryption\*.

Ranger KMS is based on the Hadoop KMS originally developed by the Apache community. The Hadoop KMS stores keys in a file-based Java keystore by default. Ranger extends the native Hadoop KMS functionality by allowing you to store keys in a secure database.

Ranger provides centralized administration of the key management server through the Ranger admin portal.

There are three main functions within the Ranger KMS:

Key management. Ranger admin provides the ability to create, update or delete keys using the Web UI or REST APIs.

## What is Apache Ranger KMS

- Ranger Provides Centralized Administration of Key Management
- Three Main Functions Within Ranger KMS:
  - Key Management
    - Ranger Admin Provides the Ability to Create, Update or Delete Keys using Web UI or REST APIs
  - Access Control Policies
    - Ranger Admin Provides the Ability to Manage Access Control Policies Within Ranger KMS.
    - Access Policies Control Permissions to Generate or Manage Keys
    - Adds Another Layer of Security for Data Encrypted in Hadoop
  - Audit
    - Ranger Provides Full Audit Trace of All Actions Performed by Ranger KMS

367 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Ranger Key Management Service (Ranger KMS) is a open source, scalable cryptographic key management service supporting HDFS "data at rest" encryption\*.

Ranger KMS is based on the Hadoop KMS originally developed by the Apache community. The Hadoop KMS stores keys in a file-based Java keystore by default. Ranger extends the native Hadoop KMS functionality by allowing you to store keys in a secure database.

Ranger provides centralized administration of the key management server through the Ranger admin portal.

There are three main functions within the Ranger KMS:

Key management. Ranger admin provides the ability to create, update or delete keys using the Web UI or REST APIs.

## What is Apache Ranger KMS

- Ranger KMS with HDFS Encryption are Recommended For Use in All Clusters
- Ranger KMS Service is Scalable
- Multiple Instances of Can Be Configured Behind a Load Balancer.

368 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Ranger Key Management Service (Ranger KMS) is a open source, scalable cryptographic key management service supporting HDFS "data at rest" encryption\*.

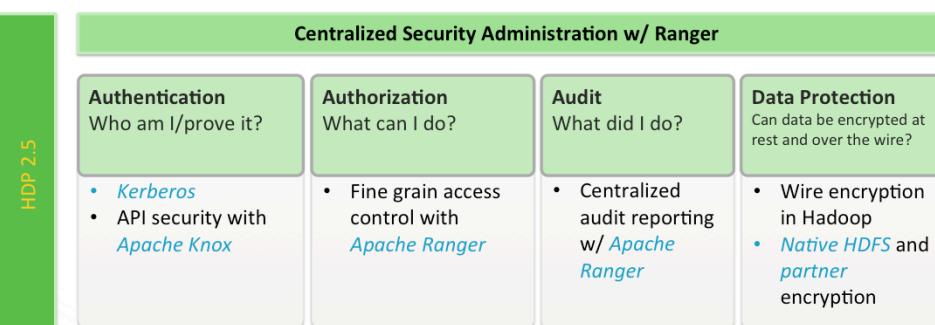
Ranger KMS is based on the Hadoop KMS originally developed by the Apache community. The Hadoop KMS stores keys in a file-based Java keystore by default. Ranger extends the native Hadoop KMS functionality by allowing you to store keys in a secure database.

Ranger provides centralized administration of the key management server through the Ranger admin portal.

There are three main functions within the Ranger KMS:

Key management. Ranger admin provides the ability to create, update or delete keys using the Web UI or REST APIs.

## What is Apache Ranger KMS



369 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Objectives



- What is Apache Ranger KMS
- HDFS Encryption



## HDFS Encryption

- Encryption is a Form of Data Security
- Hadoop Provides Several Ways to Encrypt Stored Data
- Lowest Level is Volume Encryption – Entire Disk Drive
  - Protects Data After Physical Theft
  - Accidental Loss of Disk Volume
  - Does Not Support Finer-Grained Encryption of Specific Files/Directories
  - Will Not Protect Against Viruses or Other Attacks Occurring While System is Running
- Application Level Encryption
  - Encryption Within Application Running on Hadoop
  - Supports Higher Level of Granularity
  - Prevents "Rogue Admin" Access
  - But Adds Layer of Complexity to Application Architecture



371 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

Encryption is a form of data security that is required in industries such as healthcare and the payment card industry. Hadoop provides several ways to encrypt stored data.

The lowest level of encryption is volume encryption, which protects data after physical theft or accidental loss of a disk volume. The entire volume is encrypted; this approach does not support finer-grained encryption of specific files or directories. In addition, volume encryption does not protect against viruses or other attacks that occur while a system is running.

Application level encryption (encryption within an application running on top of Hadoop) supports a higher level of granularity and prevents "rogue admin" access, but adds a layer of complexity to the application architecture.

A third approach, HDFS data at rest encryption, encrypts selected files and directories stored ("at rest") in HDFS. This approach uses specially designated HDFS directories known as "encryption zones."

## HDFS Encryption

- HDFS Data at Rest Encryption
  - Encrypts Selected/Specific Files and Directories
  - Uses Specially Designated HDFS Directories known as "Encryption Zones"

372 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Encryption is a form of data security that is required in industries such as healthcare and the payment card industry. Hadoop provides several ways to encrypt stored data.

The lowest level of encryption is volume encryption, which protects data after physical theft or accidental loss of a disk volume. The entire volume is encrypted; this approach does not support finer-grained encryption of specific files or directories. In addition, volume encryption does not protect against viruses or other attacks that occur while a system is running.

Application level encryption (encryption within an application running on top of Hadoop) supports a higher level of granularity and prevents "rogue admin" access, but adds a layer of complexity to the application architecture.

A third approach, HDFS data at rest encryption, encrypts selected files and directories stored ("at rest") in HDFS. This approach uses specially designated HDFS directories known as "encryption zones."

## HDFS Encryption

- Data At Rest Encryption Implements End-to-End Encryption of Data Read From/Written To HDFS
- End-to-End Encryption – Data is Encrypted/Decrypted by Client
- HDFS Does Not Have Access to Unencrypted Data/Keys
- Encryption Involves Several Elements
  - HDFS Encryption Zone – Special HDFS Directory where Data
    - Encrypted on Write
    - Decrypted on Read
  - EZ Key – Master Encryption Key associated with all files in an EZ
  - Each Zone Associated with Encryption Key When Zone is Created
  - Each File within Zone has Unique Encryption Key – “Data Encryption Key” (DEK)

373 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



HDFS data at rest encryption implements end-to-end encryption of data read from and written to HDFS. End-to-end encryption means that data is encrypted and decrypted only by the client. HDFS does not have access to unencrypted data or keys.

HDFS encryption involves several elements:

Encryption key: A new level of permission-based access protection, in addition to standard HDFS permissions.

HDFS encryption zone: A special HDFS directory within which all data is encrypted upon write, and decrypted upon read.

Each encryption zone is associated with an encryption key that is specified when the zone is created.

Each file within an encryption zone has a unique encryption key, called the "data

## HDFS Encryption

- HDFS Does Not Have Access to DEKs (only to EDEKs)
- DataNodes See Stream of Encrypted Bytes.
- HDFS Stores "Encrypted Data Encryption Keys" (EDEKs) as Part of the File's Metadata on the NameNode.
- Clients Decrypt an EDEK and Use the Associated DEK to Encrypt/Decrypt Data During Write/Read Operations
- Ranger KMS has these Basic Responsibilities
  - Provide Access to Stored Encryption Zone Keys
  - Generate and Manage Encryption Zone Keys and Create Encrypted Data Keys to be Stored in Hadoop
  - Decrypt EDEKs to DEKs for HDFS clients
  - Audit All Access Events in Ranger KMS



374 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

HDFS data at rest encryption implements end-to-end encryption of data read from and written to HDFS. End-to-end encryption means that data is encrypted and decrypted only by the client. HDFS does not have access to unencrypted data or keys.

HDFS encryption involves several elements:

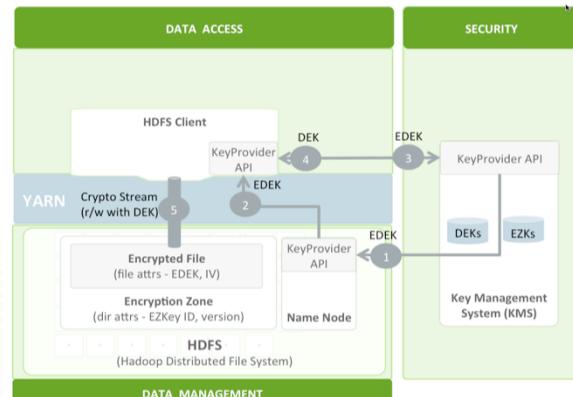
**Encryption key:** A new level of permission-based access protection, in addition to standard HDFS permissions.

**HDFS encryption zone:** A special HDFS directory within which all data is encrypted upon write, and decrypted upon read.

Each encryption zone is associated with an encryption key that is specified when the zone is created.

Each file within an encryption zone has a unique encryption key, called the "data

## Apache Ranger KMS



Acronym	Description
EZ	Encryption Zone (an HDFS directory)
EZK	Encryption Zone Key; master key associated with all files in an EZ
DEK	Data Encryption Key, unique key associated with each file. EZ Key used to generate DEK.
EDEK	Encrypted DEK, Name Node only has access to encrypted DEK.
IV	Initialization Vector



375 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

Process for writing a file when HDFS encryption enabled is shown here

0. Pre-req: EZ must be created

- KMS admin creates a Encryption Zone (EZ) Key (hadoop key create mykey)
- HDFS admin creates an Encryption Zone using the EZKey (hdfs crypto -createZone -keyName myKey -path /home/me/zone1). Assuming that /home/me/zone1 already exists and empty
- EZ created with EZK id/version stored as dir attributes (as shown in EZ box here)

When user creates a file in the EZ with appropriate access permissions,

1. Name Node uses EZ Key ID and version to get an EDEK (encrypted data encryption key) from KMS
2. Name Node returns EDEK to HDFS Client
3. HDFS client requests KMS to decrypt EDEK
4. KMS checks that the client has permission to access the encryption zone key version. If successful, passes client DEK
5. HDFS client uses DEK and Hadoop Cryptographic File System (CryptoOutputStream) to write an encrypted file in the HDFS EZ. The EDEK is then stored persistently as part of the file xattr (as shown in Encrypted File box above)

When user reads a file in the EZ, check for appropriate permissions and KMS authorization

- Name Node passes EDEK to KMS
- KMS provides HDFS Client with DEK
- HDFS client uses DEK and HDFS Cryptographic File System (CryptoInputStream) to read the un-encrypted contents of the file

## HDFS Encryption

### ● Role Separation

- Access to the Key Encryption/Decryption Process Typically Restricted to End Users
  - Encrypted Keys Safely Stored and Handled by HDFS
  - HDFS Admin User does not have access to the Encrypted Keys
  - So even if HDFS compromised, malicious user only gains access to ciphertext and encrypted keys
- Recommended: create a separate HDFS admin user for Key Management.
- Results in Two Types of HDFS Administrator Accounts:
  - HDFS service user: the system-level account associated with HDFS (hdfs by default)
  - HDFS admin user: an account in hdfs supergroup, used by HDFS administrators to configure/manage HDFS
- For example, to create HDFS admin user called : **encrypter**
  - In Advanced hdfs-site: dfs.cluster.administrators = hdfs, **encrypter**
  - In Advanced dbks-site: hadoop.kms.blacklist.DECRYPT\_EEK = hdfs, **encrypter**



376 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

Ranger Key Management Service (Ranger KMS): An open source key management service based on Hadoop's KeyProvider API.

For HDFS encryption, the Ranger KMS has three basic responsibilities:

Provide access to stored encryption zone keys.

Generate and manage encryption zone keys, and create encrypted data keys to be stored in Hadoop.

Audit all access events in Ranger KMS.

Role Separation

Access to the key encryption/decryption process is typically restricted to end users.

## Objectives



- What is Apache Ranger KMS
- HDFS Encryption
- HDFS Encryption and Ranger KMS

377 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Overview HDFS Encryption and Ranger KMS

- Once Ranger KMS is Set Up – NameNode and HDFS Client Configured
- HDFS Administrator can Use Command Line Tools
  - Create Encryption Keys – “hadoop key”
  - Set Up New Encryption Zones – “hdfs crypto”
- Overall WorkFlow
  - Create HDFS encryption zone key that will be
    - used to encrypt the file-level data in the encryption zone
    - key is stored and managed by Ranger KMS
  - Create a new HDFS folder
    - Specify required permissions, owner, and group for the folder
    - Using the new encryption zone key, designate the folder as an encryption zone.

378 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



After the Ranger KMS has been set up and the NameNode and HDFS clients have been configured, an HDFS administrator can use the hadoop key and hdfs crypto command-line tools to create encryption keys and set up new encryption zones.

The overall workflow is as follows:

Create an HDFS encryption zone key that will be used to encrypt the file-level data encryption key for every file in the encryption zone. This key is stored and managed by Ranger KMS.

Create a new HDFS folder. Specify required permissions, owner, and group for the folder.

Using the new encryption zone key, designate the folder as an encryption zone.

Configure client access. The user associated with the client application needs

## Overview HDFS Encryption and Ranger KMS

- Overall WorkFlow - Continued

- Configure client access

- User Associated with Client Application Needs Sufficient Permission to Access Encrypted Data
    - Within Encryption Zone User Needs File/Directory Access and Access for Certain Key Operations through Posix Permissions or Ranger Access Control

- After Permission are Set – HDFS Applications/Java API Clients

- With Sufficient HDFS/Ranger KMS Access
  - Can Read/Write To/From Encryption Zone



379 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

After the Ranger KMS has been set up and the NameNode and HDFS clients have been configured, an HDFS administrator can use the hadoop key and hdfs crypto command-line tools to create encryption keys and set up new encryption zones.

The overall workflow is as follows:

Create an HDFS encryption zone key that will be used to encrypt the file-level data encryption key for every file in the encryption zone. This key is stored and managed by Ranger KMS.

Create a new HDFS folder. Specify required permissions, owner, and group for the folder.

Using the new encryption zone key, designate the folder as an encryption zone.

Configure client access. The user associated with the client application needs

## Objectives



- What is Apache Ranger KMS
- HDFS Encryption
- HDFS Encryption and Ranger KMS
- Install / Configure Apache Ranger KMS



## Install Apache Ranger KMS with Ambari

- Install Ranger KMS Via Add Service Wizard Under “Admin > Stack and Versions”
- Add Service Wizard – Select Ranger KMS
- Choose Services – Confirm Ranger KMS is Selected
- Assign Master – Choose Hosts to Run Ranger KMS
- Customize Services – Details in next slides
- Review Configuration – Deploy
- Monitor Deployment – Install, Start and Test
- Review Summary
- Installation Complete

381 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



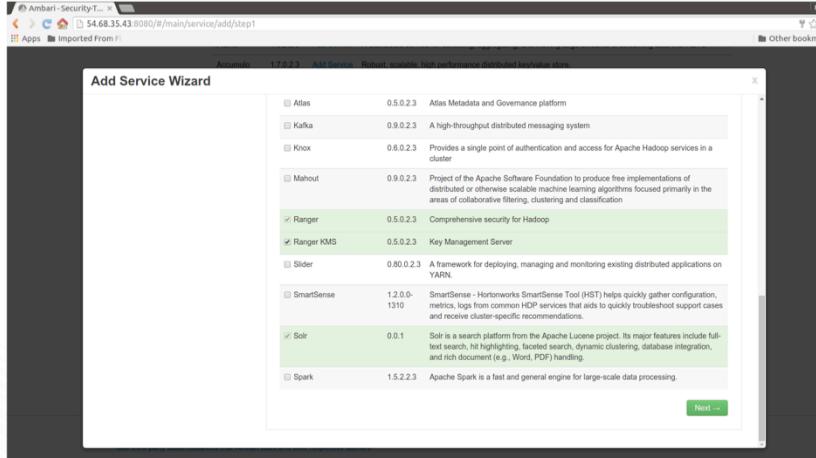
## Install Apache Ranger KMS with Ambari

The screenshot shows the 'Add Service Wizard' interface. The title bar says 'Install Apache Ranger KMS with Ambari'. The main window is titled 'Choose Services' with the sub-instruction 'Choose which services you want to install on your cluster.' A sidebar on the left lists steps: 'ADD SERVICE WIZARD', 'Choose Services' (which is highlighted in dark grey), 'Assign Masters', 'Assign Slaves and Clients', 'Customize Services', 'Configure Identities', 'Review', 'Install, Start and Test', and 'Summary'. The central area displays a table of available services:

Service	Version	Description
HDFS	2.7.1.2.3	Apache Hadoop Distributed File System
YARN + MapReduce2	2.7.1.2.3	Apache Hadoop NextGen MapReduce (YARN)
Tez	0.7.0.2.3	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
Hive	1.2.1.2.3	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
HBase	1.1.1.2.3	A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications.
Pig	0.15.0.2.3	Scripting platform for analyzing large datasets
Sqoop	1.4.6.2.3	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
Oozie	4.2.0.2.3	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ExtJS Library.

At the bottom left of the wizard window, it says '382 © Hortonworks Inc. 2011 – 2016. All Rights Reserved.' In the bottom right corner of the entire screenshot, there is a small blue and green logo for 'Hortonworks UNIVERSITY'.

## Install Apache Ranger KMS with Ambari



The screenshot shows the Ambari Add Service Wizard interface. The title bar reads "Add Service Wizard". Below it, a sub-header says "Accumulo 1.7.0.2.3 Add Service Retired, available, high performance distributed key/value store". A list of services is displayed in a table:

Service	Version	Description
Atlas	0.5.0.2.3	Atlas Metadata and Governance platform
Kafka	0.9.0.2.3	A high-throughput distributed messaging system
Knox	0.8.0.2.3	Provides a single point of authentication and access for Apache Hadoop services in a cluster
Mahout	0.8.0.2.3	Project of the Apache Software Foundation to produce free implementations of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification
<input checked="" type="checkbox"/> Ranger	0.5.0.2.3	Comprehensive security for Hadoop
<input checked="" type="checkbox"/> Ranger KMS	0.5.0.2.3	Key Management Server
Slider	0.80.0.2.3	A framework for deploying, managing and monitoring existing distributed applications on YARN
SmartSense	1.2.0.0-1310	SmartSense - Hortonworks SmartSense Tool (NST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations
<input checked="" type="checkbox"/> Solr	0.8.1	Solr is a search platform from the Apache Lucene project. Its major features include full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling.
Spark	1.5.2.2.3	Apache Spark is a fast and general engine for large-scale data processing.

A green "Next >" button is at the bottom right of the wizard window. The Ambari logo is in the top right corner of the browser window.

383 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari Add Service Wizard interface. The title bar reads "Accumulo 1.7.0.2.3 Add Service Robust, scalable, high performance distributed key/value store". The main window is titled "Add Service Wizard" and "Choose Services". The current step is "Assign Masters". On the left, there is a sidebar with navigation links: ADD SERVICE WIZARD, Choose Services, Assign Masters (which is highlighted in blue), Assign Slaves and Clients, Customize Services, Configure Identities, Review, Install, Start and Test, and Summary.

The main content area is titled "Assign Masters" and contains the sub-instruction "Assign master components to hosts you want to run them on." It lists several master components and their assigned hosts:

- NameNode: ip-172-30-0-104.us-west-2.com
- SNameNode: ip-172-30-0-105.us-west-2.com
- History Server: ip-172-30-0-105.us-west-2.com
- App Timeline Server: ip-172-30-0-105.us-west-2.com
- ResourceManager: ip-172-30-0-105.us-west-2.com
- HiveServer2: ip-172-30-0-105.us-west-2.com
- Hive Metastore: ip-172-30-0-105.us-west-2.com
- WebHCat Server: ip-172-30-0-105.us-west-2.compute.internal
- ZooKeeper Server: ip-172-30-0-105.us-west-2.com

Each host entry includes a dropdown menu showing other available hosts and a "Select" button. To the right of the host list, there are three boxes representing different hosts, each with a list of services assigned to it:

- ip-172-30-0-104.us-west-2.compute.internal (15.3 GB, 4 cores): NameNode, ZooKeeper Server, Ranger KMS Server, Solr
- ip-172-30-0-105.us-west-2.compute.internal (15.3 GB, 4 cores): SNameNode, History Server, App Timeline Server, ResourceManager, HiveServer2, Hive Metastore, WebHCat Server, ZooKeeper Server, Ranger Admin, Solr, Ranger UserSync
- ip-172-30-0-106.us-west-2.compute.internal (15.3 GB, 4 cores): ZooKeeper Server, Solr

At the bottom left of the wizard window, it says "384 © Hortonworks Inc. 2011 – 2016. All Rights Reserved." At the bottom right, there is a green "Hortonworks UNIVERSITY" logo.

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari interface with the title "Install Apache Ranger KMS with Ambari". The main window is titled "Add Service Wizard" and displays configuration options for the Ranger KMS service. The configuration fields include:

- WebHCat Server: ip-172-30-0-105.us-west-2.compute.internal
- ZooKeeper Server: ip-172-30-0-105.us-west-2.com
- ZooKeeper Server: ip-172-30-0-104.us-west-2.com
- ZooKeeper Server: ip-172-30-0-106.us-west-2.com
- Solr: ip-172-30-0-106.us-west-2.com
- Ranger Admin: ip-172-30-0-105.us-west-2.com
- Ranger KMS Server: ip-172-30-0-34.us-west-2.com
- Solr: ip-172-30-0-104.us-west-2.com
- Solr: ip-172-30-0-105.us-west-2.com
- Ranger Usersync: ip-172-30-0-105.us-west-2.com

At the bottom of the wizard, there are "Back" and "Next >" buttons. A small Hortonworks logo is visible in the bottom right corner.

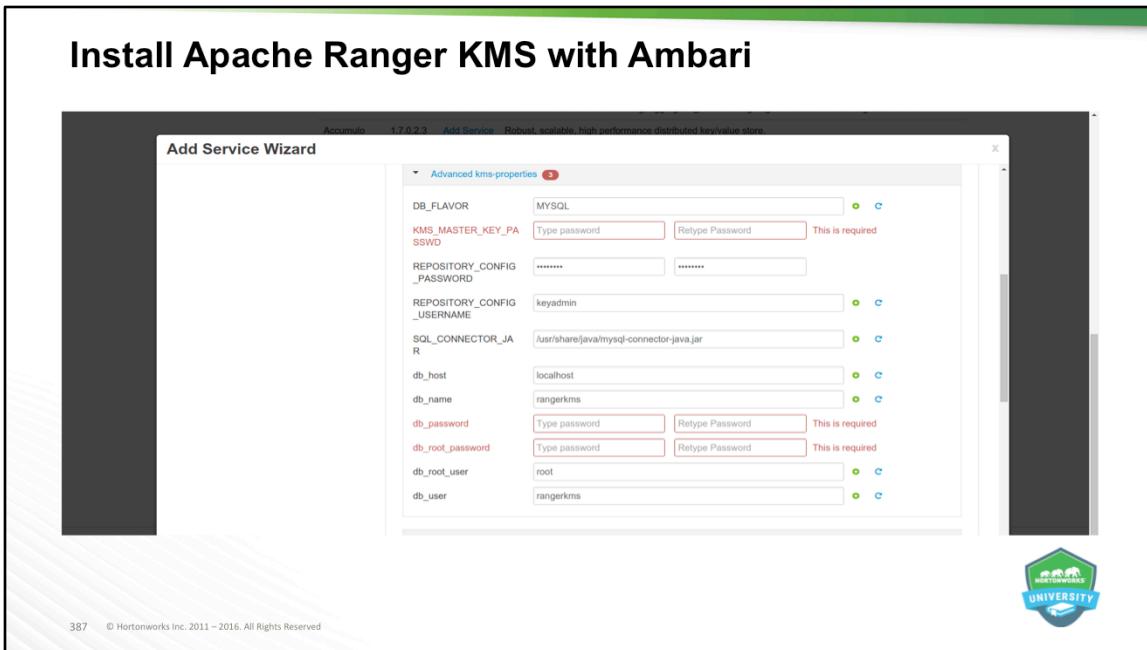
385 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari web interface with the title "Install Apache Ranger KMS with Ambari". The main window is titled "Add Service Wizard" and is currently on the "Customize Services" step. On the left, there's a sidebar with options like "ADD SERVICE WIZARD", "Choose Services", "Assign Masters", "Customize Services" (which is selected), "Configure Identities", "Review", "Install, Start and Test", and "Summary". The main content area is titled "Customize Services" and contains a message: "We have come up with recommended configurations for the services you selected. Customize them as you see fit." Below this, it says "There are 2 configuration changes in 1 service Show Details". A table lists the configuration changes:

Group	Default (4)	Manage Config Groups
Advanced dbks-site		
Advanced kms-env		
Advanced kms-log4j		
Advanced kms-properties		1
Advanced kms-site		

At the bottom of the window, there are "Next Step" and "Cancel" buttons. The footer of the page includes the text "386 © Hortonworks Inc. 2011 – 2016. All Rights Reserved" and a small "Hortonworks UNIVERSITY" logo.



Advanced kms-properties:

**KMS\_MASTER\_KEY\_PASSWORD** = BadPass#1

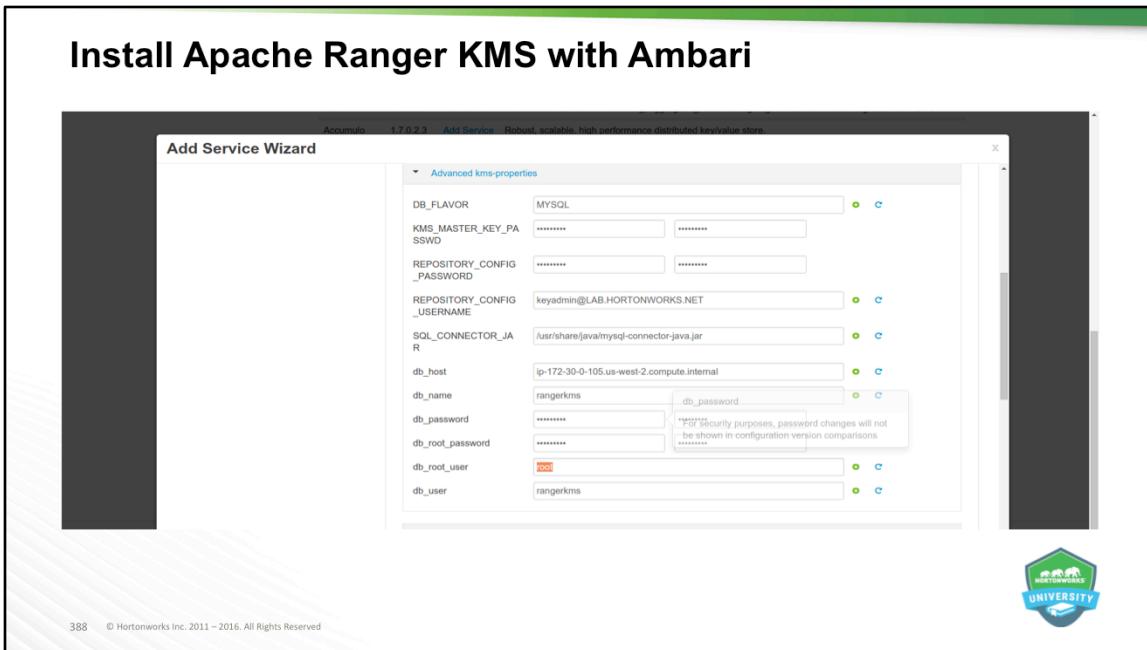
**REPOSITORY\_CONFIG\_USERNAME** = keyadmin@LAB.HORTONWORKS.NET

**REPOSITORY\_CONFIG\_PASSWORD** = BadPass#1

**db\_host** = Internal FQDN of MySQL node e.g. ip-172-30-0-181.us-west-2.compute.internal

**db\_password** = BadPass#1

**db\_root\_password** = BadPass#1



Advanced kms-properties:

**KMS\_MASTER\_KEY\_PASSWORD** = BadPass#1

**REPOSITORY\_CONFIG\_USERNAME** = keyadmin@LAB.HORTONWORKS.NET

**REPOSITORY\_CONFIG\_PASSWORD** = BadPass#1

**db\_host** = Internal FQDN of MySQL node e.g. ip-172-30-0-181.us-west-2.compute.internal

**db\_password** = BadPass#1

**db\_root\_password** = BadPass#1

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari interface with a modal dialog titled 'Add Property'. The 'Type' dropdown is set to 'kms-site.xml'. The 'Properties key-value (one per line)' input field contains the following three lines:

```
hadoop.kms.proxyuser.keyadmin.groups=*
hadoop.kms.proxyuser.keyadmin.hosts=*
hadoop.kms.proxyuser.keyadmin.users=*
```

Below the dialog, the Ambari navigation menu is visible, showing sections like 'Custom kms-properties', 'Custom kms-site', 'Custom ranger-kms-audit', 'Custom ranger-kms-policymgr-ssl', and 'Custom ranger-kms-security'. At the bottom left, it says '389 © Hortonworks Inc. 2011 – 2016. All Rights Reserved.' and at the bottom right is a 'Hortonworks UNIVERSITY' logo.

Kms-site:

```
hadoop.kms.proxyuser.keyadmin.groups=*
hadoop.kms.proxyuser.keyadmin.hosts=*
hadoop.kms.proxyuser.keyadmin.users=*
```

## Install Apache Ranger KMS with Ambari

Ambari - SecurityT... X

54.68.35.43:8080/#/main/services/RANGER\_KMS/configs

Imported From F...

Mark

Discard Save

Advanced ranger-kms-audit

Advanced ranger-kms-policymgr-ssl

Advanced ranger-kms-security

Advanced ranger-kms-site

Custom dbks-site

Custom kms-properties

Custom kms-site

hadoop.kms.proxyuser.keyadmin.groups: \* [keyadmin groups] [keyadmin host] [keyadmin users] Add Property ...

hadoop.kms.proxyuser.keyadmin.hosts: \* [keyadmin groups] [keyadmin host] [keyadmin users]

hadoop.kms.proxyuser.keyadmin.users: \* [keyadmin groups] [keyadmin host] [keyadmin users]

Custom ranger-kms-audit

Custom ranger-kms-policymgr-ssl

390 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

UNIVERSITY

Kms-site:

```
hadoop.kms.proxyuser.keyadmin.groups=*
hadoop.kms.proxyuser.keyadmin.hosts=*
hadoop.kms.proxyuser.keyadmin.users=*
```

## Install Apache Ranger KMS with Ambari

Add Property

Type: kms-site.xml

Properties  
key=value (one per line)

```
hadoop.kms.proxyuser.hive.users=*
hadoop.kms.proxyuser.oozie.users=*
hadoop.kms.proxyuser.HTTP.users=*
hadoop.kms.proxyuser.ambari.users=*
```

Cancel Add

391 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



### Kms-site

```
hadoop.kms.authentication.type=kerberos
hadoop.kms.authentication.kerberos.keytab=/etc/security/keytabs/
spnego.service.keytab
hadoop.kms.authentication.kerberos.principal=*

hadoop.kms.proxyuser.hive.users=*
hadoop.kms.proxyuser.oozie.users=*
hadoop.kms.proxyuser.HTTP.users=*
hadoop.kms.proxyuser.ambari.users=*
hadoop.kms.proxyuser.yarn.users=*
hadoop.kms.proxyuser.hive.hosts=*
hadoop.kms.proxyuser.oozie.hosts=*
hadoop.kms.proxyuser.HTTP.hosts=*
hadoop.kms.proxyuser.ambari.hosts=*
```

## Install Apache Ranger KMS with Ambari

**Advanced ranger-kms-audit**

xasecure.audit.credential.provider.file	[file://file[[credential_file]]]	<input type="radio"/>	<input checked="" type="radio"/>
Audit to DB	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
xasecure.audit.destination.db.batch.filespool.dir	[var/log/rangerkms/audit/db/spool]	<input type="radio"/>	<input checked="" type="radio"/>
xasecure.audit.destination.db.jdbc.driver	[@dbc_driver]	<input type="radio"/>	<input checked="" type="radio"/>
xasecure.audit.destination.db.jdbc.url	[{{audit_jdbc_url}}]	<input type="radio"/>	<input checked="" type="radio"/>
xasecure.audit.destination.db.password	cryptd	<input type="radio"/>	<input checked="" type="radio"/>
xasecure.audit.destination.db.user	[xa_audit_db_user]	<input type="radio"/>	<input checked="" type="radio"/>
Audit to HDFS	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
xasecure.audit.destination.hdfs.batch.filespool.dir	[var/log/ranger/kms/audit/hdfs/spool]	<input type="radio"/>	<input checked="" type="radio"/>
xasecure.audit.destination.hdfs.dir	[hdfs://ip-172-30-0-180.us-west-2.compute.internal:8020/ranger/audit]	<input type="radio"/>	<input checked="" type="radio"/>
Audit to SOLR	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
xasecure.audit.destination.solr.batch.filespool.dir	[var/log/ranger/kms/audit/solr/spool]	<input type="radio"/>	<input checked="" type="radio"/>
xasecure.audit.destination.solr.urls	[{ranger_audit_solr_urls}]	<input type="radio"/>	<input checked="" type="radio"/>
xasecure.audit.destination.zookeeper.zookeepers	[ip-172-30-0-180.us-west-2.compute.internal:2181,ip-172-30-0-182.us-west-2.compute.	<input type="radio"/>	<input checked="" type="radio"/>
xasecure.audit.is.enabled	true	<input type="radio"/>	<input checked="" type="radio"/>
Audit provider summary enabled	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

392 © Hortonworks Inc.



Advanced ranger-kms-audit:

xasecure.audit.destination.hdfs.dir = hdfs://YOUR\_NN\_INTERNAL\_HOSTNAME:8020/ranger/audit

xasecure.audit.destination.solr.zookeepers=YOUR\_ZK\_QUORUM/ranger\_audits

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari interface for adding a new service. The title bar reads "Accumulo 1.7.0.2.3 Add Service Robust, scalable, high performance distributed key/value store". The main window is titled "Add Service Wizard" and is currently on the "Configure Identities" step. The left sidebar lists steps: "ADD SERVICE WIZARD", "Choose Services", "Assign Masters", "Assign Slaves and Clients", "Customize Services", "Configure Identities" (which is selected), "Review", "Install, Start and Test", and "Summary". The right panel is titled "Configure Identities" and contains fields for "Admin principal", "Admin password", "Keytab Dir", "Realm", "Additional Realms", "Spnego Principal", and "Spnego Keytab". A "General" tab is selected, and a "Global" dropdown is open. A "Save Admin Credentials" checkbox is present. The "Keytab Dir" field contains "/etc/security/keytabs". The "Realm" field contains "LAB.HORTONWORKS.NET". The "Spnego Principal" field contains "HTTP/\_HOST@\${realm}" and the "Spnego Keytab" field contains "\${keytab\_dir}/spnego.service.keytab". A "Configure Identities" logo is visible in the bottom right corner.

393 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari interface for adding a new service. The title bar reads "Accumulo 1.7.0.2.3 - Add Service Robust, scalable, high performance distributed key/value store". The main window is titled "Add Service Wizard" and is currently on the "Configure Identities" step. The left sidebar lists steps: "ADD SERVICE WIZARD", "Choose Services", "Assign Masters", "Assign Slaves and Clients", "Customize Services", "Configure Identities" (which is selected and highlighted in dark grey), "Review", "Install, Start and Test", and "Summary". The right panel is titled "Configure Identities" and contains the following fields:

Configure identities	
Configure principal name and keytab location for service users and hadoop service components.	
General	Advanced
Global	
Admin principal	hadoopadmin@LAB.HORTONWORKS.NET
Admin password	*****
<input type="checkbox"/> Save Admin Credentials	
Keytab Dir	/etc/security/keytabs
Realm	LAB.HORTONWORKS.NET
Additional Realms	(Optional)
Spnego Principal	HTTP/_HOST@\${realm}
Spnego Keytab	\$[keytab_dir]/spnego.service.keytab

At the bottom left of the wizard window, it says "394 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". At the bottom right, there is a small logo for "Hortonworks UNIVERSITY".

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari interface with the title "Install Apache Ranger KMS with Ambari". The main window is titled "Add Service Wizard" and is on the "Review" step. The left sidebar lists steps: Choose Services, Assign Masters, Customize Services, Configure Identities, Review (selected), Install, Start and Test, and Summary. The review panel displays configuration details:

- Admin Name : admin
- Cluster Name : Security-TTT-William-SantaClara-112
- Total Hosts : 4 (0 new)
- Repositories:
  - debian7 (HDP-2.3): http://public-repo-1.hortonworks.com/HDP/debian7/2/x/updates/2.3.4.0
  - debian7 (HDP-UTILS-1.1.0.20): http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/debian6
  - redhat6 (HDP-2.3): http://public-repo-1.hortonworks.com/HDP/centos6/2/x/updates/2.3.4.0
  - redhat6 (HDP-UTILS-1.1.0.20): http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/centos6
  - redhat7 (HDP-2.3): http://public-repo-1.hortonworks.com/HDP/centos7/2/x/updates/2.3.4.0
  - redhat7 (HDP-UTILS-1.1.0.20): http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.20/repos/centos7
  - suse11 (HDP-2.3): http://public-repo-1.hortonworks.com/HDP/suse11sp3/2/x/updates/2.3.4.0

At the bottom right of the Ambari interface is a small "Hortonworks UNIVERSITY" logo.

395 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari Add Service Wizard interface. The main title is "Install Apache Ranger KMS with Ambari". On the left, a sidebar lists steps: ADD SERVICE WIZARD, Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Configure Identities, Review, and two buttons: Install, Start and Test (which is highlighted in dark grey), and Summary. The main content area is titled "Install, Start and Test" with the sub-instruction "Please wait while the selected services are installed and started." Below this is a progress bar at 100% overall. A table follows, showing the status of four hosts:

Host	Status	Message
ip-172-30-0-104.us-west-2.compute.internal	100%	Success
ip-172-30-0-105.us-west-2.compute.internal	100%	Success
ip-172-30-0-106.us-west-2.compute.internal	100%	Success
ip-172-30-0-34.us-west-2.compute.internal	100%	Success

At the bottom of the table, it says "4 of 4 hosts showing - Show All". To the right of the table are buttons for "Show" (dropdown), "In Progress (0)", "Warning (0)", "Success (4)", and "Fail (0)". Below the table is a green banner with the text "Successfully installed and started the services." and a "Next >" button. At the very bottom of the page, there is a small footer: "396 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". On the right side of the Ambari interface, there is a blue "Hortonworks UNIVERSITY" logo.

**Install Apache Ranger KMS with Ambari**

The screenshot shows the Ambari interface with the title "Install Apache Ranger KMS with Ambari". The main window is titled "Add Service Wizard" and is on the "Summary" step. A message at the top right says: "Important: You may also need to restart other services for the newly added services to function properly (for example, HDFS and YARN/MapReduce need to be restarted after adding Oozie). After closing this wizard, please restart all services that have the restart indicator next to the service name." Below this, a summary box states: "The cluster consists of 4 hosts. Installed and started services successfully on 4 new hosts. Install and start completed in 1 minutes and 5 seconds." At the bottom right of the summary box is a green "Complete →" button. Below the summary box, there is a "RECOMMENDED SERVICES" section with two entries:

- Solr: Version 0.0.1, status **Installed**. Description: Solr is a search platform from the Apache Lucene project. Its major features include full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling.
- Spark: Version 1.5.2.2.3, status **Add Service**. Description: Apache Spark is a fast and general engine for large-scale data processing.

At the bottom left of the Ambari interface, it says "397 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". On the right side, there is a blue "Hortonworks UNIVERSITY" logo.

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari Metrics dashboard with the following data:

Metric	Value
HDFS Disk Usage	8%
DataNodes Live	4/4
HDFS Links	NameNode Secondary NameNode 4 DataNodes
Memory Usage	No Data Available
Network Usage	No Data Available
CPU Usage	No Data Available
Cluster Load	No Data Available
NameNode Heap	7%
NameNode RPC	0.25 ms
NameNode CPU WIO	n/a
NameNode Uptime	18.0 hr
ResourceManager Heap	19%
ResourceManager Uptime	17.9 hr
NodeManagers Live	4/4
YARN Memory	0%
YARN Links	(link icon)

Other visible elements include a sidebar with service icons (HDFS, MapReduce2, YARN, Tez, Hive, Pig, ZooKeeper, Kerberos, Ranger, Ranger KMS, Solr) and an "Actions" button. The footer includes the text "398 © Hortonworks Inc. 2011 – 2016. All Rights Reserved" and a "UNIVERSITY" logo.

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari UI interface for managing configurations. The top navigation bar includes links for Ambari, Security-T., Dashboard, Services, Hosts, Alerts, Admin, and a user dropdown. A red alert icon is visible in the top right corner.

The left sidebar lists various services: HDFS, MapReduce2, YARN, Tez, Hive, Pig, ZooKeeper, Kerberos, Ranger, and Ranger KMS. The Ranger KMS service is currently selected, indicated by a dark grey background.

The main content area is titled "Configs" and shows a "Default (4)" group. It displays two configuration versions: V2 and V1. Version V2 was authored by admin on Wednesday, January 27, 2016, at 17:40. The configuration details for V2 are listed below:

- Advanced dbks-site
- Advanced kms-env
- Advanced kms-log4j
- Advanced kms-properties
- Advanced kms-site
- Advanced ranger-kms-audit
- Advanced ranger-kms-policymgr-ssl

At the bottom of the configuration list, there are "Discard" and "Save" buttons. A small note indicates that changes will take effect after a cluster restart.

In the bottom right corner of the Ambari interface, there is a green hexagonal badge with the text "Hortonworks UNIVERSITY".

At the bottom left of the screenshot, there is a copyright notice: "399 © Hortonworks Inc. 2011 – 2016. All Rights Reserved".

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari web interface with the title "Install Apache Ranger KMS with Ambari". The left sidebar lists services: HDFS, MapReduce2, YARN, Tez, Hive, Pig, ZooKeeper, Kerberos, Ranger, and Ranger KMS. The Ranger KMS service is selected. The main content area shows a "Configs" tab with a "Summary" sub-tab. A yellow banner at the top states "Restart Required: 1 Component on 1 Host" with a "Restart" button. Below this is a "Manage Config Groups" section with a "Default (4)" group selected. It displays three configurations: V3 (a moment ago, HDP-2.3), V2 (13 minutes ago, HDP-2.3), and V1 (13 minutes ago, HDP-2.3). A "Save" button is visible. On the right, there's a "Service Actions" dropdown and a "Hosts" link. At the bottom left is a copyright notice: "400 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". At the bottom right is a "Hortonworks UNIVERSITY" logo.

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari interface for managing Hadoop services. The top navigation bar includes links for Dashboard, Services, Hosts, Alerts, Admin, and a user dropdown. The left sidebar lists services: HDFS, MapReduce2, YARN, Tez, Hive, Pig, ZooKeeper, Kerberos, Ranger, Ranger KMS, and Solr. The main content area is titled "Configs" and shows a list of configurations for the "Default (4)" group. A message at the top states "admin authored on Wed, Jan 27, 2016 17:40". Below this, there are sections for "NameNode", "Secondary NameNode", "DataNode", "General", "NFS Gateway", and "Advanced core-site". A "Save" button is visible at the bottom right of the configuration list. At the bottom left, a footer note reads "401 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". On the right side, there is a small "Hortonworks UNIVERSITY" logo.

## Install Apache Ranger KMS with Ambari

402 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



After KMS install, add proxyuser for KMS

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari interface with a modal dialog titled "Add Property". The "Type" dropdown is set to "core-site.xml". In the "Properties" section, the key "hadoop.proxyuser.kms.groups" is set to the value "\*". Below the dialog, the main Ambari configuration interface shows the "hosts" section with several properties listed, including "hadoop.proxyuser.hdfs.groups", "hadoop.proxyuser.hdfs.hosts", "hadoop.proxyuser.hive.groups", "hadoop.proxyuser.hive.hosts", and "hadoop.proxyuser.yarn.groups". The "hadoop.proxyuser.hdfs.hosts" property is currently set to "ec2-52-27-218-75.us-west-2.compute.amazonaws.com". At the bottom left of the interface, there is a copyright notice: "403 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". At the bottom right, there is a "Hortonworks UNIVERSITY" logo.

HDFS > Configs > Custom core-site:

`hadoop.proxyuser.kms.groups = *`

## Install Apache Ranger KMS with Ambari

V8 ✓ admin authored on Wed, Jan 27, 2016 17:40

Discard Save

hadoop.proxyuser.HTTP. groups	ip-172-30-0-105.us-west-2.compute.internal	green	red
hadoop.proxyuser.HDFS. hosts	*	green	red
hadoop.proxyuser.hcat. groups	*	green	red
hadoop.proxyuser.hcat. hosts	ec2-52-27-218-75.us-west-2.compute.amazonaws.com	green	red
hadoop.proxyuser.hdfs. groups	*	green	red
hadoop.proxyuser.hdfs. hosts	*	green	red
hadoop.proxyuser.hive. groups	*	green	red
hadoop.proxyuser.hive. hosts	ec2-52-27-218-75.us-west-2.compute.amazonaws.com	green	red
hadoop.proxyuser.yarn. groups	*	green	red
hadoop.proxyuser.yarn. hosts	ip-172-30-0-105.us-west-2.compute.internal	green	red
hadoop.proxyuser.kms. groups	*	green	red

404 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari web interface with the title "Install Apache Ranger KMS with Ambari". The URL is 54.68.35.43:8080/#/main/services/HDFS/configs. The "Configs" tab is selected. At the top, there is a message: "Restart Required: 14 Components on 4 Hosts". Below this, a "Manage Config Groups" section shows a group named "Default (4)". A log entry from "admin" is displayed: "admin authored on Wed, Jan 27, 2016 18:06". The main area is divided into two sections: "NameNode" and "DataNode". Each section contains configuration parameters with sliders and current values. The "NameNode" section includes "NameNode directories" (set to "/hadoop/hdfs/namenode"), "NameNode Java heap size" (set to 775 MB), and "NameNode Server threads" (set to 1). The "DataNode" section includes "DataNode directories" (set to "/hadoop/hdfs/data"), "DataNode failed disk tolerance" (set to 1), and "DataNode maximum Java heap size" (set to 1 GB). A "Save" button is located at the bottom right of the configuration section. The footer of the page includes the Hortonworks logo and the text "405 © Hortonworks Inc. 2011 – 2016. All Rights Reserved".

Restart all affected services

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari UI configuration page for the 'Advanced ranger-kms-audit' service. The URL is [http://54.68.35.43:8080/#/main/services/RANGER\\_KMS/configs](http://54.68.35.43:8080/#/main/services/RANGER_KMS/configs). The configuration includes:

- xasecure.audit.credential.provider.file: jceks://file{{credential\_file}}
- Audit to DB:
  - xasecure.audit.destination.db.batch.size: 1000
  - xasecure.audit.destination.db.jdbc.driver: {{jdbc\_driver}}
  - xasecure.audit.destination.db.jdbc.url: {{audit\_jdbc\_url}}
  - xasecure.audit.destination.db.password: crypted
  - xasecure.audit.destination.db.user: {{xa\_audit\_db\_user}}
- Audit to HDFS:
  - xasecure.audit.destination.hdfs.batch.size: 1000
  - xasecure.audit.destination.hdfs.dir: hdfs://NAMENODE\_HOSTNAME:8020/ranger/audit
- Audit to SOLR:
  - xasecure.audit.destination.solr.dir: /var/log/ranger/kms/audit/solr/spool

At the bottom left, it says "406 © Hortonworks Inc. 2011 – 2016. All Rights Reserved". At the bottom right, there is a "Hortonworks UNIVERSITY" logo.

## Install Apache Ranger KMS with Ambari

The screenshot shows the Ambari configuration interface for the 'Advanced ranger-kms-audit' service. The page title is 'Install Apache Ranger KMS with Ambari'. At the top right, there are 'Discard' and 'Save' buttons. The configuration section contains several parameters:

- xasecure.audit.credential.provider.file: jceks://file/{{credential\_file}}
- Audit to DB:
  - xasecure.audit.destination.db.batch.filespool.dir: /var/log/ranger/kms/audit/db/spool
  - xasecure.audit.destination.db.jdbc.driver: {{jdbc\_driver}}
  - xasecure.audit.destination.db.jdbc.url: {{audit\_dbc\_url}}
  - xasecure.audit.destination.db.password: encrypted
  - xasecure.audit.destination.db.user: {{xa\_audit\_db\_user}}
- Audit to HDFS:
  - xasecure.audit.destination.hdfs.batch.filespool.dir: /var/log/ranger/kms/audit/hdfs/spool
  - xasecure.audit.destination.hdfs.dir: hdfs://ip-172-30-0-172.us-west-2.compute.internal:5020/ranger/audit

At the bottom left, it says '407 © Hortonworks Inc. 2011 – 2016. All Rights Reserved.' On the right side, there is a small 'Hortonworks UNIVERSITY' logo.

## Objectives



- What is Apache Ranger KMS
- HDFS Encryption
- HDFS Encryption and Ranger KMS
- Install / Configure Apache Ranger KMS
- Configure HDFS to Use Encryption



## Configure HDFS to Use Encryption

- HDFS administrator can use the hadoop key and hdfs crypto tools to create encryption keys and set up new encryption zones
- Overall workflow follows:
  - Create an HDFS encryption zone key that will be used to encrypt the file-level data encryption key for every file in the encryption zone.
  - This key is stored and managed by Ranger KMS.
  - Create a new HDFS folder.
  - Specify required permissions, owner, and group for the folder.
  - Using the new encryption zone key, designate the folder as an encryption zone.
  - Configure client access. The user associated with the client application needs sufficient permission to access encrypted data.
  - In an encryption zone, the user needs file/directory access (through Posix permissions or Ranger access control), as well as access for certain key operations.



409 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

After the Ranger KMS has been set up and the NameNode and HDFS clients have been configured, an HDFS administrator can use the hadoop key and hdfs crypto command-line tools to create encryption keys and set up new encryption zones.

The overall workflow is as follows:

Create an HDFS encryption zone key that will be used to encrypt the file-level data encryption key for every file in the encryption zone. This key is stored and managed by Ranger KMS.

Create a new HDFS folder. Specify required permissions, owner, and group for the folder.

Using the new encryption zone key, designate the folder as an encryption zone.

Configure client access. The user associated with the client application needs

## Configure HDFS to Use Encryption

- Overall workflow follows: - Continued
- Set permissions on which users/groups have access to key via Ranger policy
- After permissions are set, Java API clients and HDFS applications with sufficient HDFS and Ranger KMS access privileges can write and read to/from files in the encryption zone.

410 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



After the Ranger KMS has been set up and the NameNode and HDFS clients have been configured, an HDFS administrator can use the hadoop key and hdfs crypto command-line tools to create encryption keys and set up new encryption zones.

The overall workflow is as follows:

Create an HDFS encryption zone key that will be used to encrypt the file-level data encryption key for every file in the encryption zone. This key is stored and managed by Ranger KMS.

Create a new HDFS folder. Specify required permissions, owner, and group for the folder.

Using the new encryption zone key, designate the folder as an encryption zone.

Configure client access. The user associated with the client application needs

## Configure HDFS to Use Encryption

- HDP supports hardware acceleration with Advanced Encryption Standard New Instructions (AES-NI).
- Compared with the software implementation of AES, hardware acceleration offers an order of magnitude faster encryption/decryption
- To use AES-NI optimization you need CPU and library support
- AES-NI optimization requires an extended CPU instruction set for AES hardware acceleration.
- To check for this
  - # cat /proc/cpuinfo | grep aes
- Look for output with flags and 'aes'

411 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



HDP supports hardware acceleration with Advanced Encryption Standard New Instructions (AES-NI). Compared with the software implementation of AES, hardware acceleration offers an order of magnitude faster encryption/decryption.

To use AES-NI optimization you need CPU and library support,

CPU Support for AES-NI optimization

AES-NI optimization requires an extended CPU instruction set for AES hardware acceleration.

There are several ways to check for this; for example:

```
$ cat /proc/cpuinfo | grep aes
```

## Configure HDFS to Use Encryption

- OpenSSL 1.0.1e version of the libcrypto.so library supports hardware acceleration
- A version of the libcrypto.so library with AES-NI support must be installed on HDFS cluster nodes and MapReduce client hosts
- The following instructions describe how to install and configure the libcrypto.so library.
- RHEL/CentOS 6.5 or later
- On HDP cluster nodes, the installed version of libcrypto.so supports AES-NI, but you will need to make sure that the symbolic link exists:  
\$ sudo ln -s /usr/lib64/libcrypto.so.1.0.1e /usr/lib64/libcrypto.so
- On MapReduce client hosts, install the openssl-devel package:  
\$ sudo yum install openssl-devel

412 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



HDP supports hardware acceleration with Advanced Encryption Standard New Instructions (AES-NI). Compared with the software implementation of AES, hardware acceleration offers an order of magnitude faster encryption/decryption.

To use AES-NI optimization you need CPU and library support,

CPU Support for AES-NI optimization

AES-NI optimization requires an extended CPU instruction set for AES hardware acceleration.

There are several ways to check for this; for example:

```
$ cat /proc/cpuinfo | grep aes
```

## Configure HDFS to Use Encryption

- Verify client host is ready to use the AES-NI instruction set optimization for HDFS encryption

```
# hadoop checknative
```

- Output similar to the following:

```
15/08/12 13:48:39 INFO bzip2.Bzip2Factory: Successfully loaded & initialized native-bzip2 library system-native
```

```
14/12/12 13:48:39 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
```

```
Native library checking:
```

```
hadoop: true /usr/lib/hadoop/lib/native/libhadoop.so.1.0.0
```

```
zlib: true /lib64/libz.so.1
```

```
snappy: true /usr/lib64/libsnappy.so.1
```

```
lz4: true revision:99
```

```
bzip2: true /lib64/libbz2.so.1
```

```
openssl: true /usr/lib64/libcrypto.so
```

413 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



HDP supports hardware acceleration with Advanced Encryption Standard New Instructions (AES-NI). Compared with the software implementation of AES, hardware acceleration offers an order of magnitude faster encryption/decryption.

To use AES-NI optimization you need CPU and library support,

CPU Support for AES-NI optimization

AES-NI optimization requires an extended CPU instruction set for AES hardware acceleration.

There are several ways to check for this; for example:

```
$ cat /proc/cpuinfo | grep aes
```

## Configure HDFS to Use Encryption

- Verify client host is ready to use the AES-NI instruction set optimization for HDFS encryption
- True in the openssl row - Hadoop has detected the right version of libcrypto.so and optimization will work.
- False in this row - do not have the correct version.

414 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



HDP supports hardware acceleration with Advanced Encryption Standard New Instructions (AES-NI). Compared with the software implementation of AES, hardware acceleration offers an order of magnitude faster encryption/decryption.

To use AES-NI optimization you need CPU and library support,

CPU Support for AES-NI optimization

AES-NI optimization requires an extended CPU instruction set for AES hardware acceleration.

There are several ways to check for this; for example:

```
$ cat /proc/cpuinfo | grep aes
```

## Objectives



- What is Apache Ranger KMS
- HDFS Encryption
- HDFS Encryption and Ranger KMS
- Install / Configure Apache Ranger KMS
- Configure HDFS to Use Encryption
- HDFS Encryption Zone



## HDFS Encryption Zone – Create Encryption Key

- Create a "master" encryption key for the new encryption zone.
- Each key will be specific to an encryption zone.
- Ranger supports AES/CTR/NoPadding as the cipher suite.
- Key size can be 128 or 256 bits.
- Recommendation: create a new superuser for key management.
- In the following examples, **superuser encrypter** creates the key.
- This separates the data access role from the encryption role, strengthening security.
- Create an Encryption Key using Ranger KMS (Recommended)
- In the Ranger Web UI screen:
  - Choose the Encryption tab at the top of the screen.
  - Select the KMS service from the drop-down list.

416 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



### Create an Encryption Key

Create a "master" encryption key for the new encryption zone. Each key will be specific to an encryption zone.

Ranger supports AES/CTR/NoPadding as the cipher suite. (The associated property is listed under HDFS -> Configs in the Advanced hdfs-site list.)

Key size can be 128 or 256 bits.

Recommendation: create a new superuser for key management. In the following examples, superuser encr creates the key. This separates the data access role from the encryption role, strengthening security.

### Create an Encryption Key using Ranger KMS (Recommended)

**HDFS Encryption Zone – Create Encryption Key**

**Key Management**

Select Service : cl1\_kms

Add New Key

Key Name	Cipher	Version	Attributes	Length	Created Date	Action
sensitivefolder	AES/CTR/NoPadding	1	key.acl.name => SensitiveFolder	128	08/06/2015 01:30:44 PM	
test	AES/CTR/NoPadding	1	key.acl.name => test	128	08/13/2015 01:49:35 PM	
testkeyfromcli	AES/CTR/NoPadding	1	key.acl.name => testkeyfromcli	128	07/24/2015 06:04:36 PM	
testkeyfromui	AES/CTR/NoPadding	1	key.acl.name => testkeyfromui	128	07/24/2015 06:04:16 PM	
testkeygml	AES/CTR/NoPadding	1	key.acl.name => testkeyGML	128	08/06/2015 02:02:40 PM	
tk1	AES/CTR/NoPadding	1	key.acl.name => tk1	128	08/25/2015 12:22:23 PM	

417 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Create an Encryption Key

Create a "master" encryption key for the new encryption zone. Each key will be specific to an encryption zone.

Ranger supports AES/CTR/NoPadding as the cipher suite. (The associated property is listed under HDFS -> Configs in the Advanced hdfs-site list.)

Key size can be 128 or 256 bits.

Recommendation: create a new superuser for key management. In the following examples, superuser encr creates the key. This separates the data access role from the encryption role, strengthening security.

### Create an Encryption Key using Ranger KMS (Recommended)

## HDFS Encryption Zone – Create Encryption Key

- To create a new key:

- Click on "Add New Key":
- Add a valid key name.
- Select the cipher name.
  - Ranger supports AES/CTR/NoPadding as the cipher suite.
- Specify the key length, 128 or 256 bits.
- Add other attributes as needed, and then save the key.



EZ Key

418 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



To create a new key:

Click on "Add New Key":

Add a valid key name.

Select the cipher name. Ranger supports AES/CTR/NoPadding as the cipher suite.

Specify the key length, 128 or 256 bits.

Add other attributes as needed, and then save the key.

## HDFS Encryption Zone – Create Encryption Key

Ranger Access Manager Encryption keyadmin

KMS > d1\_kms > Key Create

**Key Detail**

Key Name *	<input type="text"/>				
Cipher	AES/CTR/NoPadding				
Length	128				
Description	<input type="text"/>				
Attributes	<table border="1"><thead><tr><th>Name</th><th>Value</th></tr></thead><tbody><tr><td><input type="text"/></td><td><input type="text"/></td></tr></tbody></table> <input type="button"/> <input type="button"/>	Name	Value	<input type="text"/>	<input type="text"/>
Name	Value				
<input type="text"/>	<input type="text"/>				

419 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



To create a new key:

Click on "Add New Key":

Add a valid key name.

Select the cipher name. Ranger supports AES/CTR/NoPadding as the cipher suite.

Specify the key length, 128 or 256 bits.

Add other attributes as needed, and then save the key.

## HDFS Encryption Zone – Create Encryption Zone

- Each encryption zone must be defined using an empty directory and an existing encryption key.
- An encryption zone cannot be created on top of a directory that already contains data.
- Recommendation: use one unique key for each encryption zone
- Use the crypto “createZone” command to create a new encryption zone:  
`# crypto -createZone -keyName <keyName> -path <path>`  
● where:
  - keyName: specifies the name of the key to use for the encryption zone
  - path: specifies the path of the encryption zone to be created, must be empty directory

420 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Each encryption zone must be defined using an empty directory and an existing encryption key. An encryption zone cannot be created on top of a directory that already contains data.

Recommendation: use one unique key for each encryption zone.

Use the crypto createZone command to create a new encryption zone. The syntax is:

`-createZone -keyName <keyName> -path <path>`

where:

-keyName: specifies the name of the key to use for the encryption zone.

-path specifies the path of the encryption zone to be created. It must be an empty

## HDFS Encryption Zone – Create Encryption Zone

- Service account “hdfs” can create zones, but cannot write data unless the account has sufficient permission
- Recommendation:
  - Define a separate user account for the HDFS administrator
  - Do not provide access to keys for this user in Ranger KMS
  - For example, to create HDFS admin user called **encryptper**:
    - In Advanced hdfs-site: dfs.cluster.administrators = hdfs,encrypter
    - In Advanced dbks-site: hadoop.kms.blacklist.DECRYPT\_EEK = hdfs,encrypter

421 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Each encryption zone must be defined using an empty directory and an existing encryption key. An encryption zone cannot be created on top of a directory that already contains data.

Recommendation: use one unique key for each encryption zone.

Use the crypto createZone command to create a new encryption zone. The syntax is:

`-createZone -keyName <keyName> -path <path>`

where:

`-keyName`: specifies the name of the key to use for the encryption zone.

`-path` specifies the path of the encryption zone to be created. It must be an empty

## HDFS Encryption Zone – Create Encryption Zone

- As HDFS administrator, create a new empty directory. For example:

```
# hdfs dfs -mkdir /zone_encr
```

- Using the encryption key, make the directory an encryption zone. For example:

```
# hdfs crypto -createZone -keyName key1 -path /zone_encr
```

- When finished, the NameNode will recognize the folder as an HDFS encryption zone.

- To verify creation of the new encryption zone as HDFS administrator run:

```
$ hdfs crypto -listZones
```

- To List the encryption zone and its key

```
$ hdfs crypto -listZones
```

```
/zone-encr key1
```



422 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

As HDFS administrator, create a new empty directory. For example:

```
# hdfs dfs -mkdir /zone_encr
```

Using the encryption key, make the directory an encryption zone. For example:

```
# hdfs crypto -createZone -keyName key1 -path /zone_encr
```

When finished, the NameNode will recognize the folder as an HDFS encryption zone.

To verify creation of the new encryption zone, run the crypto -listZones command as an HDFS administrator:

```
-listZones
```

## HDFS Encryption Zone – Create Encryption Zone

- Property in hdfs-default.xml file causes listZone requests to be batched:
    - dfs.namenode.list.encryption.zones.num.responses
  - Improves NameNode performance
  - Specifies the maximum number of zones that will be returned in a batch
  - Default is 100
- 
- To remove an encryption zone, delete the root directory of the zone.  
\$hdfs dfs -rm -R /zone\_encr

423 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



As HDFS administrator, create a new empty directory. For example:

```
# hdfs dfs -mkdir /zone_encr
```

Using the encryption key, make the directory an encryption zone. For example:

```
# hdfs crypto -createZone -keyName key1 -path /zone_encr
```

When finished, the NameNode will recognize the folder as an HDFS encryption zone.

To verify creation of the new encryption zone, run the crypto -listZones command as an HDFS administrator:

```
-listZones
```

## HDFS Encryption Zone – Read/Write Encryption Zone

- HDFS Clients/Application with Sufficient HDFS/Ranger KMS Permission can:
  - Read Files From EZ
  - Write File To EZ
- Client Write Process:
  - Client Writes to Encryption Zone
  - NameNode Checks Clients Write Access Permissions
  - Asks Ranger KMS to Create File-Level Key, Encrypted with Encryption Zone Master Key
  - NameNode Store File-Level Encrypted Data Encryption Key (EDEK) as File's Metadata
  - Returns EDEK to Client
  - Client Asks Ranger KMS to Decode EDEK to DEK
  - Client Uses DEK to Write Encrypted Data
  - Ranger Check Permission for User Before Decrypting EDEK and Producing DEK for Client

424 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Clients and HDFS applications with sufficient HDFS and Ranger KMS permissions can read and write files from/to an encryption zone.

Overview of the client write process:

The client writes to the encryption zone.

The NameNode checks to make sure that the client has sufficient write access permissions. If so, the NameNode asks Ranger KMS to create a file-level key, encrypted with the encryption zone master key.

The Namenode stores the file-level encrypted data encryption key (EDEK) generated by Ranger KMS as part of the file's metadata, and returns the EDEK to the client.

The client asks Ranger KMS to decode the EDEK (to DEK), and uses the DEK to write encrypted data. Ranger KMS checks for permissions for the user before decrypting

## HDFS Encryption Zone – Read/Write Encryption Zone



425 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Clients and HDFS applications with sufficient HDFS and Ranger KMS permissions can read and write files from/to an encryption zone.

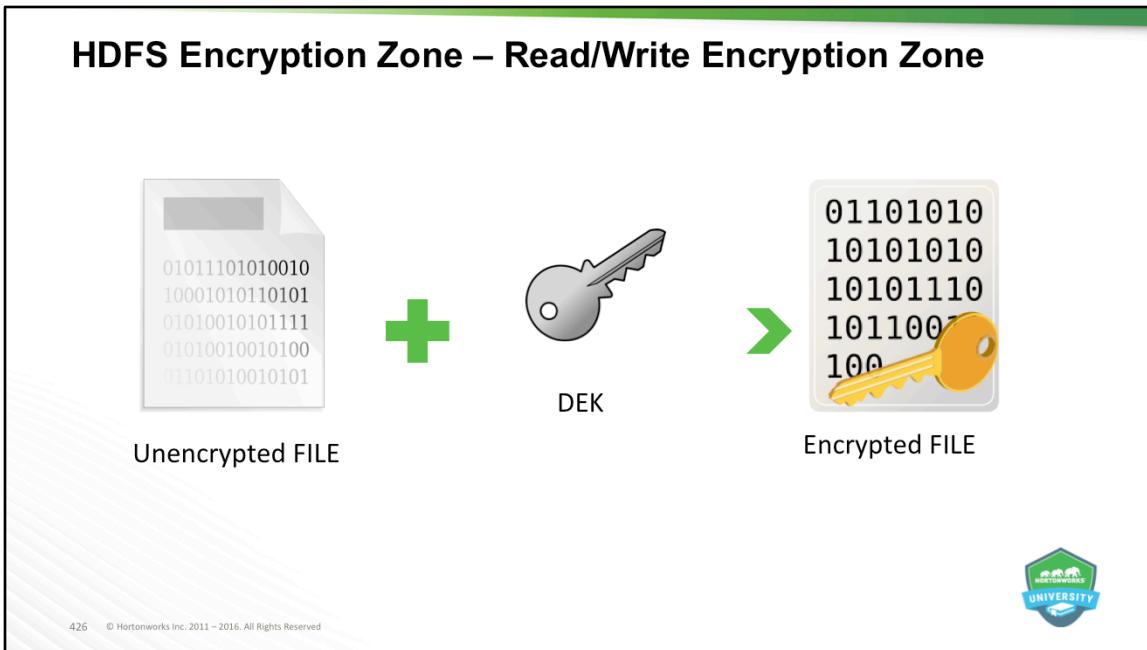
Overview of the client write process:

The client writes to the encryption zone.

The NameNode checks to make sure that the client has sufficient write access permissions. If so, the NameNode asks Ranger KMS to create a file-level key, encrypted with the encryption zone master key.

The Namenode stores the file-level encrypted data encryption key (EDEK) generated by Ranger KMS as part of the file's metadata, and returns the EDEK to the client.

The client asks Ranger KMS to decode the EDEK (to DEK), and uses the DEK to write encrypted data. Ranger KMS checks for permissions for the user before decrypting



Clients and HDFS applications with sufficient HDFS and Ranger KMS permissions can read and write files from/to an encryption zone.

Overview of the client write process:

The client writes to the encryption zone.

The NameNode checks to make sure that the client has sufficient write access permissions. If so, the NameNode asks Ranger KMS to create a file-level key, encrypted with the encryption zone master key.

The Namenode stores the file-level encrypted data encryption key (EDEK) generated by Ranger KMS as part of the file's metadata, and returns the EDEK to the client.

The client asks Ranger KMS to decode the EDEK (to DEK), and uses the DEK to write encrypted data. Ranger KMS checks for permissions for the user before decrypting

## HDFS Encryption Zone – Read/Write Encryption Zone

### Client Read Process:

- Client Issues Read Request for File in Encryption Zone
- NameNode Checks Clients Read Access Permissions
- Returns File's EDEK and Encryption Zone Key Version Used to Encrypt EDEK
- Client Asks Ranger KMS to Decrypt EDEK
- Ranger KMS Checks for Permission to Decrypt EDEK for End User
- Ranger KMS Decrypts and Returns Unencrypted Data Encryption Key (DEK)
- Client Uses DEK to Decrypt and Read the Files

427 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Clients and HDFS applications with sufficient HDFS and Ranger KMS permissions can read and write files from/to an encryption zone.

Overview of the client write process:

The client writes to the encryption zone.

The NameNode checks to make sure that the client has sufficient write access permissions. If so, the NameNode asks Ranger KMS to create a file-level key, encrypted with the encryption zone master key.

The Namenode stores the file-level encrypted data encryption key (EDEK) generated by Ranger KMS as part of the file's metadata, and returns the EDEK to the client.

The client asks Ranger KMS to decode the EDEK (to DEK), and uses the DEK to write encrypted data. Ranger KMS checks for permissions for the user before decrypting

## HDFS Encryption Zone – Read/Write Encryption Zone



428 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Clients and HDFS applications with sufficient HDFS and Ranger KMS permissions can read and write files from/to an encryption zone.

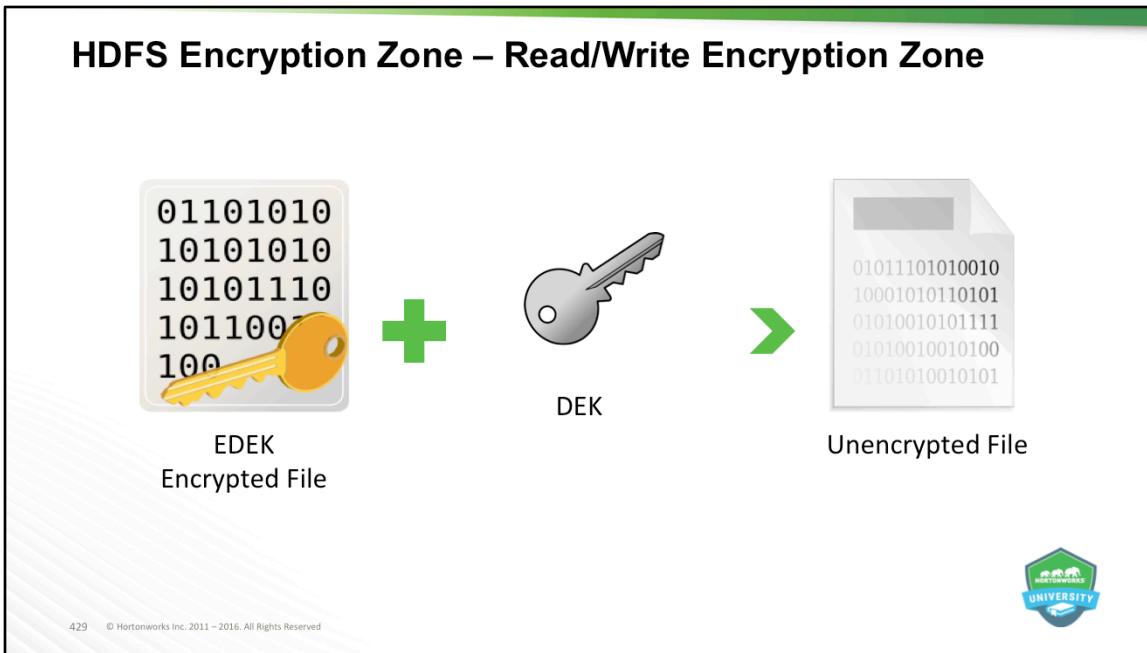
Overview of the client write process:

The client writes to the encryption zone.

The NameNode checks to make sure that the client has sufficient write access permissions. If so, the NameNode asks Ranger KMS to create a file-level key, encrypted with the encryption zone master key.

The Namenode stores the file-level encrypted data encryption key (EDEK) generated by Ranger KMS as part of the file's metadata, and returns the EDEK to the client.

The client asks Ranger KMS to decode the EDEK (to DEK), and uses the DEK to write encrypted data. Ranger KMS checks for permissions for the user before decrypting



Clients and HDFS applications with sufficient HDFS and Ranger KMS permissions can read and write files from/to an encryption zone.

Overview of the client write process:

The client writes to the encryption zone.

The NameNode checks to make sure that the client has sufficient write access permissions. If so, the NameNode asks Ranger KMS to create a file-level key, encrypted with the encryption zone master key.

The Namenode stores the file-level encrypted data encryption key (EDEK) generated by Ranger KMS as part of the file's metadata, and returns the EDEK to the client.

The client asks Ranger KMS to decode the EDEK (to DEK), and uses the DEK to write encrypted data. Ranger KMS checks for permissions for the user before decrypting

## Data Protection – HDFS Encryption

In summary:

- To convert DEK to EDEK (or vice versa): Need EZ Key
- To encrypt or decrypt a file: Need DEK
- EZ key and DEK stored in KMS
- EDEK stored in NameNode
- Notice that with HDFS encryption enabled:
  - Client needs to interact with *both* HDFS (for data/EDEK) as well as KMS (to decrypt EDEK) to get access to the DEK (required to read/write files)
  - There are two levels of checks - whether the client has permission to access:
    - the HDFS dir/file – check performed by NameNode
    - the encryption zone key version – check performed by KMS
- Recommended: create a separate HDFS admin user for Key Management.



## Knowledge Check



## Knowledge Check

1. True/False - Apache Ranger KMS is not based on Hadoop KMS?
2. True/False - Apache Ranger KMS supports encryption with “Data in Motion”?
3. Data at Rest implements \_\_\_\_\_ Encryption?
4. True/False - HDFS has direct access to unencrypted data and keys?
5. What two items must exists to create an encryption zone?
6. True/False - It is recommended that one unique key be used for each encryption zone?

432 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



1. False
2. False
3. End-to-End
4. False
5. Empty Directory and Encryption Key
6. True

# Summary



## Summary

- Apache Range KMS is a Open Source, Scalable Cryptographic Key Management Services
- Based in Hadoop KMS Originally Developed by the Apache Community
- Supports HDFS “Data at Rest” Encryption
- HDFS Encryption is a Form of Data Security
- Data At Rest Encryption Implements End-to-End Encryption of Data Read From/Written To HDFS
- HDFS Does Not Have Access to Unencrypted Data/Keys
- Each encryption zone must be defined using an empty directory and an existing encryption key.
- An encryption zone cannot be created on top of a directory that already contains data.
- Use one unique key for each encryption zone



## Lab: Ranger KMS/Data Encryption Setup





## Secure Access With Ranger



## Lesson Objectives

- Describe Apache Ranger Plugin Integration with
  - HDFS
  - Hive
  - HBase
  - Knox
  - Storm



The slide features a light gray background with a faint, large network graph pattern. In the upper right quadrant, there is a white rounded rectangle containing a green circular icon with a white arrow pointing right, followed by a bulleted list: "Ranger Integration with Hadoop" and "- HDFS". To the left of this area, the word "Objectives" is written in bold black text. In the bottom right corner, there is a small blue hexagonal logo with the text "Hortonworks UNIVERSITY" and a graduation cap icon.

Objectives

- Ranger Integration with Hadoop
  - HDFS

438 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Ranger Plugin - HDFS Integration

- HDFS Permissions
- POSIX like permission model (owner/group for files and folders)
- ACL's for fine-grained permissions (for specific set of users/groups)
  - hdfs dfs -getfacl [-R] <path>
  - hdfs dfs -setfacl [-R] [options] <path>
- dfs.permissions.enabled must be set to true

439 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Ranger Plugin - HDFS Integration

- Ranger Plugin acts as an authorizer within Namenode.
- Need to install on all namenodes (in HA environment)
- User can define policies on files and folders
  - Use of wildcard to define policies (/finance/audit\_\*)
  - Read, Write, Execute permissions are allowed
- Plugin evaluates HDFS requests and provide access
- No specific ranger policy exists - HDFS ACLs are used as fallback
- Kerborized HDFS Access how-to:
  - Just kinit and access as usual (assuming Ranger policy/ACL already in place)
  - \$ kinit someuser
  - \$ hdfs fs –ls /tmp

440 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Objectives



- Ranger Integration with Hadoop
  - HDFS
  - Hive



## Ranger Plugin - Hive Integration

- Hive facilitates querying and managing large datasets in distributed storage
- HiveServer2 (HS2) is a server interface to Hive
- HiveServer2 supports Access Control similar to relational database model
  - SELECT/UPDATE/DELETE permissions on tables/columns
  - Permission defined for USERS/ROLES
  - Also provides a pluggable authorizer model



## Ranger Plugin - Hive Integration

- Ranger Plugin acts as authorization provider for HiveServer2
- User can define policies on databases, tables/view, columns and UDFs
- Select, Update, Create, Drop, Alter, Index, Lock Permissions allowed
- Plugin evaluates Hive requests and grants/denies access based on the policies
- Creates necessary audit logs based on audit
- Specific Ranger policy must exist for gaining access



## Ranger Plugin - Hive Integration

- When the user executes GRANT/REVOKE statements
  - Hive plugin will creates/deletes necessary Ranger Policies
  - Provided the user has permission to create/delete policies
- GRANT SELECT, UPDATE, ALTER ON TABLE test\_data TO USER guest;
  - will create a new Ranger policy
- REVOKE UPDATE, ALTER ON TABLE test\_data FROM USER guest;
  - will update/delete existing Ranger policy
- ROLE specified in the GRANT/REVOKE statements will be mapped to corresponding GROUPS from your corporate directories
- Optionally, can disable GRANT/REVOKE commands
  - Force authorization policies management solely via Ranger Policy Admin



## Ranger Plugin - Hive Integration

- Hive Access How-to (using Beeline or another Hive JDBC client)
- Assuming Ranger policy/ACL already in place:
  - Access unsecure Hive

```
beeline -u "jdbc:hive2://localhost:10000/default"
```
  - Access secure Hive - Binary transport mode (the default)
    - Note the update to include hive principal

```
beeline -u "jdbc:hive2://localhost:10000/default;principal=hive/$(hostname -f)@HORTONWORKS.COM"
```
  - Access secure Hive – HTTP transport mode
    - Note the update to use HTTP and the need to provide the kerberos principal.

```
beeline -u "jdbc:hive2://localhost:10001/default;transportMode=http;httpPath=cliservice;principal=HTTP/$(hostname -f)@HORTONWORKS.COM"
```



## Objectives



- Ranger Integration with Hadoop
  - HDFS
  - Hive
  - HBase



## Ranger Plugin - HBase Integration

- Hbase is a non-relational DB on top of Hadoop/HDFS
- HBase Provides Role Based Access Control/ACLs
- ACLs are implemented as a coprocessor called AccessController
- Ranger implements a similar coprocessor for enforcing access control based on Ranger Policies



## Ranger Plugin - HBase Integration

- Ranger Plugin is implemented as a coprocessor of HBase Master/Region Servers to enforce Ranger Policies
- User can define policies on tables, column families and qualifiers
- Supports wildcard in defining policies (table = fin\_\*, col\_fam = audit\*)
- Read, Write, Create, Admin permissions allowed - RWCA
- Plugin evaluates Hbase requests grants/denies access based on policies
- Creates necessary audit logs based on audit
- Specific ranger policy must exist for gaining access
- Kerborized Hbase Access how-to (assuming Ranger policy/ACL in place):
  - Just kinit and access as usual
- \$ kinit someuser
- \$ hbase shell

448 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Ranger Plugin - HBase Integration

- Similar to Hive, Hbase also supports Grant/Revoke commands to manage access control from within Hbase shell.
- Ranger HBase plug-in creates/updates Ranger policies to reflect permissions set via Grant/Revoke (from within Hbase)
  - grant 'bob', 'RWCA', 'test\_data'
    - will create a new Ranger policy
  - grant 'bob', 'R', 'test\_data'
    - will update the Ranger policy created by earlier grant;
- Option to disable Grant/revoke commands
  - Force authorization policies management solely via Ranger Policy Admin tool



The slide features a background graphic of a network of grey lines on a light grey gradient. A green diagonal band with a yellow border runs from the bottom-left towards the top-right. In the top-left corner, the word "Objectives" is written in bold black font. A small white arrow with a green outline points from the text towards the network. On the right side, there is a bulleted list of objectives:

- Ranger Integration with Hadoop
  - HDFS
  - Hive
  - Hbase
  - Knox

At the bottom left, the number "450" and the copyright notice "© Hortonworks Inc. 2011 – 2016. All Rights Reserved" are visible. In the bottom right corner, there is a logo for "Hortonworks UNIVERSITY" featuring a graduation cap and a shield.

## Ranger Plugin - Knox Integration

- Knox provides perimeter security for Hadoop REST API
- Authentication and token verification at the perimeter
- Authentication integration with enterprise and cloud identity management systems
- Service level authorization at the perimeter
- Single URL hierarchy that aggregates REST APIs of a Hadoop cluster
- Hadoop services with built-in support
  - WebHDFS
  - Hive
  - WebHCat
  - Oozie
  - Hbase (Stargate)
  - Yarn Resource Manager
  - Storm



451 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Ranger Plugin - Knox Integration

- Knox provides service level authorization based on xml configuration
- Ranger Plugin will allow service level authorization enforcement via Ranger Policies by acting as Authorization Provider within Knox Gateway
- User can define policies on topologies and services
  - Provide access to services based on user/group/ip-address
  - E.g. Finance group will have access to WebHDFS from 10.1.1.\*
- Plugin evaluates Knox requests and grants/denies access based on the policies
- Creates necessary audit logs based on audit policies
- WebHDFS access over Knox how-to (assuming Ranger policy/ACL in place):  

```
$ curl -ik -u sales1:BadPass#1 https://$(hostname -f):8443/gateway/default/webhdfs/v1/?op=LISTSTATUS  
– Unsecure WebHDFS access sample:  
$ curl -sk -L http://$(hostname -f):50070/webhdfs/v1/?op=LISTSTATUS
```



452 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Objectives



- Ranger Integration with Hadoop
  - HDFS
  - Hive
  - Hbase
  - Knox
  - Storm

453 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Ranger Plugin - Storm Integration

- Storm is a distributed real-time computation system
- Storm provides general primitives for real-time computation similar to how Hadoop provides general primitives for batch processing
- Storm Topologies :: Hadoop MR Jobs
- Nimbus server (runs on master node)
- Supervisor (runs on each worker node)
- Communication via ZooKeeper
- Storm security is implemented based on Kerberos authentication



## Ranger Plugin - Storm Integration

- Ranger Plugin acts as an authorizer within Nimbus server
- Ranger Plugin can authorize all incoming requests based on the Ranger Policies
- User can define policies on topologies
- Permissions can be set
  - getClusterInfo, Submit/Get/Kill/Activate/Deactivate Topology
- Plugin evaluates Storm requests and grants/denies access based on the policies
- Creates necessary audit logs based on audit policies
- Kerborized Storm Access (assuming Ranger policy/ACL in place):
  - Just kinit and access as usual
  - \$ kinit someuser
  - \$ storm jar /usr/hdp/current/storm-client/contrib/storm-starter/storm-starter-topologies-\*.jar  
storm.starter.WordCountTopology WordCountTopology -c localhost

455 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Knowledge Check



## Knowledge Check

1. Ranger Plugin's act as a authorizer for which services?
2. If a Ranger policy does not exists for HDFS what is used as a fallback?
3. True/False - Ranger policy is not needed to gain access to Hive?
4. What type of policies can be defined for Hive?
5. True/False – Ranger Plugin is implemented as a coprocessor for Hbase?
6. What type of policies can be defined for Hbase?

457 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



1. HDFS Namenode, HiveServer2, Knox and Storm
2. HDFS ACL's
3. FALSE – Policy must exist
4. Databases, tables/views columns and UDF's
5. TRUE
6. Tables, Column Families and Qualifiers

# Summary



## Summary

- Ranger Plugin acts as an authorizer within Namenode
  - Plugin evaluates HDFS requests and provide access
  - No specific ranger policy exists, HDFS ACLs are used as fallback
- Ranger Plugin acts as authorization provider for HiveServer2
  - User can define policies on databases, tables/view, columns and UDFs
  - Plugin evaluates Hive requests and grants/denies access based on the policies
  - Specific Ranger policy must exist for gaining access
- Ranger Plugin is implemented as a coprocessor of HBase Master/Region Servers to enforce Ranger Policies
  - User can define policies on tables, column families and qualifiers
  - Plugin evaluates Hbase requests grants/denies access based on policies
  - Specific ranger policy must exist for gaining access

459 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



SSO = Single Sign On

SAML = Security Assertion Markup Language - XML Standard that allow secure web domains to exchange user authentication and authorization data

## Summary

- Ranger Plugin will allow service level authorization enforcement via Ranger Policies by acting as Authorization Provider within Knox Gateway
  - User can define policies on topologies and services
  - Plugin evaluates Knox requests and grants/denies access based on the policies
- Ranger Plugin acts as an authorizer within Storm - Nimbus server
  - Ranger Plugin can authorize all incoming requests based on the Ranger Policies
  - User can define policies on topologies
  - Plugin evaluates Storm requests and grants/denies access based on the policies

460 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



SSO = Single Sign On

SAML = Security Assertion Markup Language - XML Standard that allow secure web domains to exchange user authentication and authorization data

## Lab: Secured Hadoop Exercises



# Knox Overview



## Lesson Objectives

- Describe Apache Knox
- Understand the need for Apache Knox

463 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



**Objectives**

→ What is Apache Knox?

464 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## What is Apache Knox

- System to Extend Reach of Apache Hadoop Services to Users Outside of the Cluster
- Provides Single Point of Authentication and Access
- Simplifies Hadoop Security for
  - Users Who Access the Cluster Data/Execute Job
  - Operators Who Control Access/Manage the Cluster
- Provide perimeter security for Hadoop REST API'
  - Provide authentication and token verification at the perimeter
  - Enable authentication integration with enterprise and cloud identity management systems
  - Provide service level authorization at the perimeter
- Expose a single URL hierarchy that aggregates REST APIs of the Cluster
  - Limit the network endpoints/firewall holes required to access the Cluster
  - Hide the internal Hadoop cluster topology from potential attackers

465 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



The Apache Knox Gateway is a system that provides a single point of authentication and access for Apache Hadoop services in a cluster. The goal is to simplify Hadoop security for both users (i.e. who access the cluster data and execute jobs) and operators (i.e. who control access and manage the cluster). The gateway runs as a server (or cluster of servers) that provide centralized access to one or more Hadoop clusters. In general the goals of the gateway are as follows:

Provide perimeter security for Hadoop REST APIs to make Hadoop security easier to setup and use

Provide authentication and token verification at the perimeter

Enable authentication integration with enterprise and cloud identity management systems

Provide service level authorization at the perimeter

Expose a single URL hierarchy that aggregates REST APIs of a Hadoop cluster

Limit the network endpoints (and therefore firewall holes) required to access a Hadoop cluster

## Objectives



- What is Apache Knox?
- Why is Apache Knox Needed?



## Why is Apache Knox Needed

- Integrates with Identity Management and Single Sign-On (SSO) Systems
- Allows Identity from these Systems to be Used for Access to the Cluster
- Provides Security to Multiple Hadoop Clusters with the Following Advantages:
  - Simplifies access: Extends Hadoop's REST/HTTP services by encapsulating Kerberos to within the Cluster
  - Enhances security: Exposes Hadoop's REST/HTTP services without revealing network details, providing SSL out of the box.
  - Centralized control: Enforces REST API security centrally, routing requests to multiple Hadoop clusters.
  - Enterprise integration: Supports LDAP, Active Directory, SSO, SAML and other authentication systems.

467 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



The Apache Knox Gateway (“Knox”) is a system to extend the reach of Apache™Hadoop® services to users outside of a Hadoop cluster without reducing Hadoop Security. Knox also simplifies Hadoop security for users who access the cluster data and execute jobs.

Knox integrates with Identity Management and SSO systems used in enterprises and allows identity from these systems be used for access to Hadoop clusters.

Knox Gateways provides security for multiple Hadoop clusters, with these advantages:

Simplifies access: Extends Hadoop's REST/HTTP services by encapsulating Kerberos to within the Cluster.

Enhances security: Exposes Hadoop's REST/HTTP services without revealing network details, providing SSL out of the box.

## Why Apache Knox Needed

### Enhanced Security

- Protect network details
- SSL for non-SSL services
- WebApp vulnerability filter

### Centralized Control

- Central REST API auditing
- Service-level authorization
- Alternative to SSH “edge node”

### Simplified Access

- Kerberos encapsulation
- Extends API reach
- Single access point
- Multi-cluster support
- Single SSL certificate

### Enterprise Integration

- LDAP integration
- Active Directory integration
- SSO integration
- Apache Shiro extensibility
- Custom extensibility



## Hadoop REST API with Knox

Service	Direct URL	Knox URL
WebHDFS	<a href="http://namenode-host:50070/webhdfs">http://namenode-host:50070/webhdfs</a>	<a href="https://knox-host:8443/webhdfs">https://knox-host:8443/webhdfs</a>
WebHCat	<a href="http://webhcatt-host:50111/templeton">http://webhcatt-host:50111/templeton</a>	<a href="https://knox-host:8443/templeton">https://knox-host:8443/templeton</a>
Oozie	<a href="http://oozie-host:11000/oozie">http://oozie-host:11000/oozie</a>	<a href="https://knox-host:8443/oozie">https://knox-host:8443/oozie</a>
HBase	<a href="http://hbase-host:60080">http://hbase-host:60080</a>	<a href="https://knox-host:8443/hbase">https://knox-host:8443/hbase</a>
Hive	<a href="http://hive-host:10001/cliservice">http://hive-host:10001/cliservice</a>	<a href="https://knox-host:8443/hive">https://knox-host:8443/hive</a>
YARN	<a href="http://yarn-host:8088/ws">http://yarn-host:8088/ws</a>	<a href="https://knox-host:8443/resourcemanager">https://knox-host:8443/resourcemanager</a>

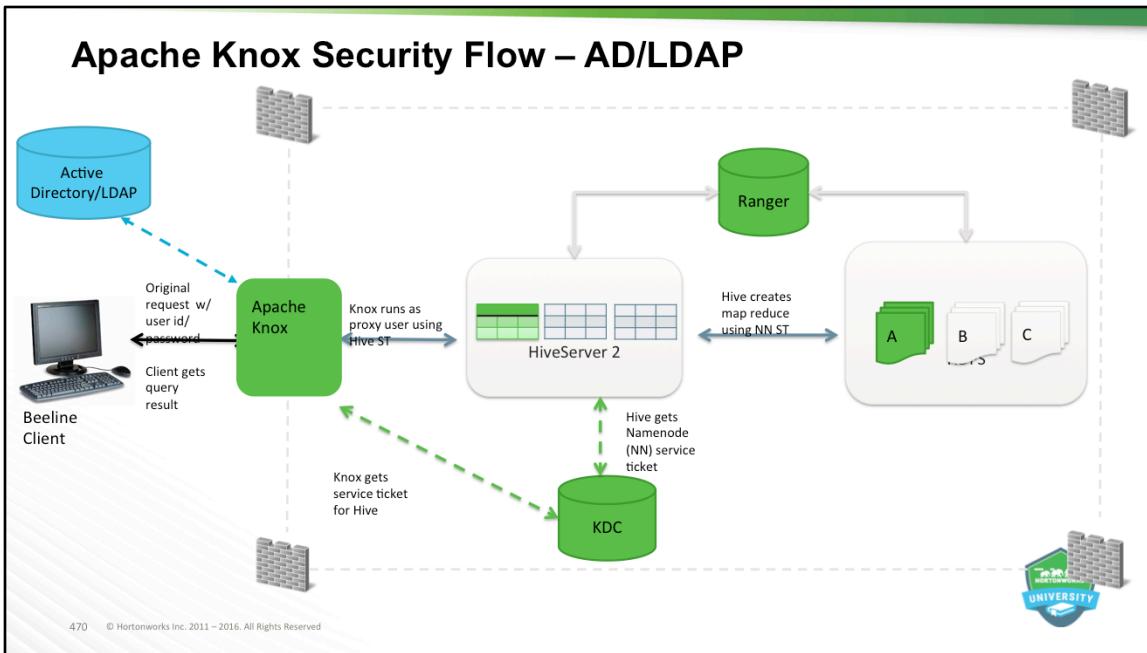
Masters could be  
on many different  
hosts

SSL config on one  
host

One host, one port

Consistent paths





## Objectives



- What is Apache Knox?
- Why is Apache Knox Needed?
- Apache Knox Architecture

471 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Apache Knox Architecture

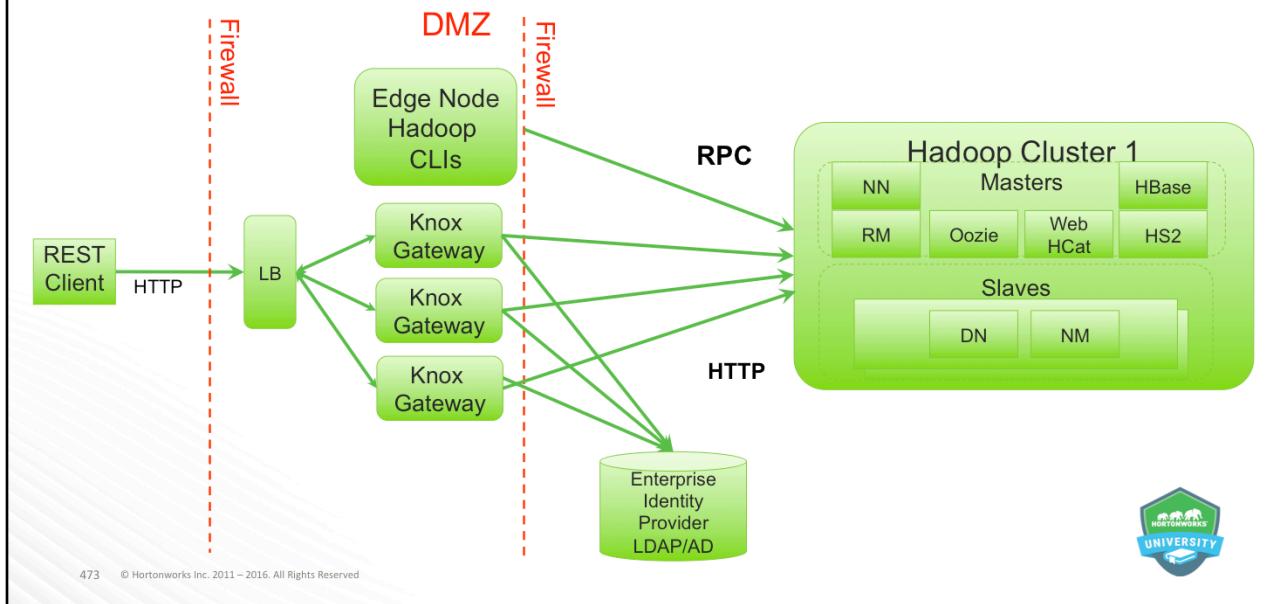
- Users Access Hadoop Externally Do So Through Knox
  - Via Apache REST API
  - Through Hadoop CLI tools
- Supported Hadoop Services Both Unsecure and Kerberized

Service	Version
YARN	2.6.0
WebHDFS	2.6.0
WebHCat/Templeton	0.13.0
Oozie	4.1.0
Hbase/Stargate	0.98.4
Hive Via WebHCat	0.14.0
Hive Via JDBC	0.14.0

472 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Hadoop REST API Security



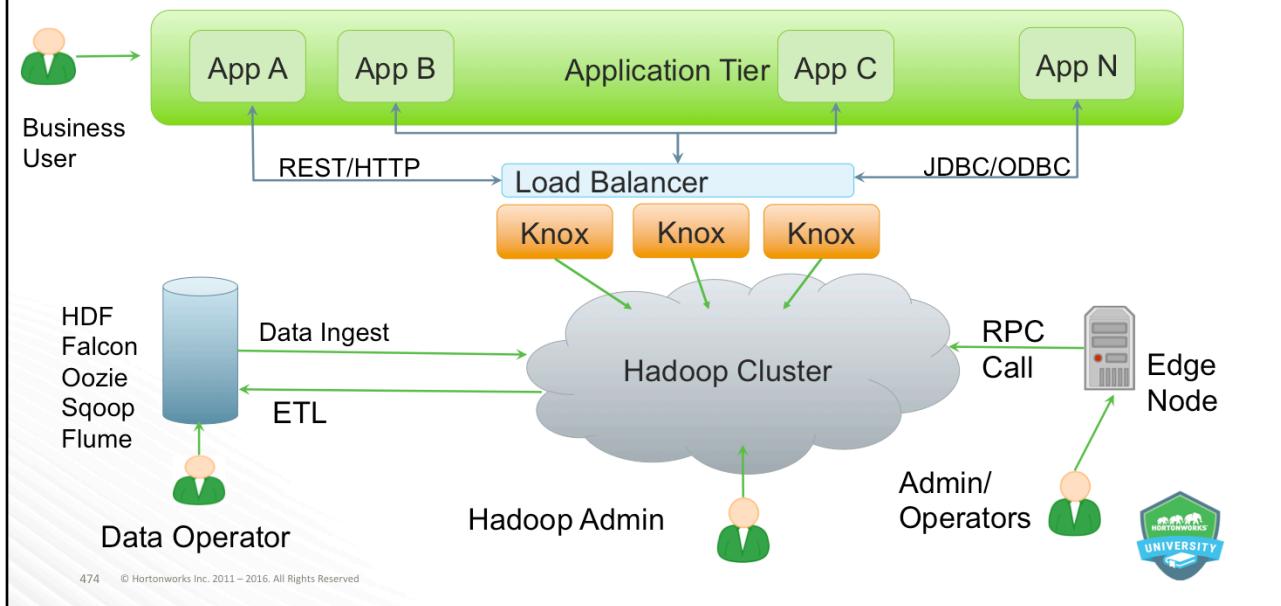
Typical topology would look like this where you have 3 layers:

1. Presentation
2. Application
3. Data

Knox can run in DMZ in between firewalls so your cluster and AD remains hidden from outside world

With Knox you can optionally expose data from multiple clusters (without end users knowing where data came from)

## BI Tools/Application Utilizing ODBC/JDBC



Applications can interact with Knox via REST/HTTP or JDBC/ODBC

Can run multiple Knox gateways via LoadBalancers

Knox is not meant to bulk import/export of data (would be like drinking water from pool using a straw)

Bulk operations should happen within cluster via HDFS, Sqoop etc

## Knowledge Check



## Knowledge Check

1. What does Apache Knox provide for Hadoop REST API's?
2. What are the Four Advantages of Apache Knox Security?
3. How does Apache Knox Simplifies Access and Enhance Security?
4. True/False - Apache Knox Gateway provides SSL out of the box?
5. What Enterprise Directory Services does Knox support?

476 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



1. Perimeter Security
2. Simplifies Access; Enhances Security; Centralized Control; Enterprise Integration
3. Extends Hadoop's REST/HTTP Services by Encapsulating Kerberos to within the Hadoop cluster and Exposes Single URL Hierarchy without revealing Hadoop Cluster details,
4. TRUE
5. LDAP, Active Directory, Single Sign-On and SAML (Security Assertion Markup Language)

# Summary



## Summary

- Apache Knox Gateway is a System to Extend the Reach of Apache Hadoop Services to Users Outside of the Cluster
- Knox Gateway Provides a Single Point of Authentication and Access
- Provide Perimeter Security for Hadoop REST API's
- Simplifies Hadoop Security by Encapsulating Kerberos to within the Hadoop Cluster
- Knox Exposes a Single URL Hierarchy that Aggregates REST API's of the Hadoop Cluster
- Integrates with Enterprise Authentication: LDAP, Active Directory, SSO and SAML

478 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



SSO = Single Sign On

SAML = Security Assertion Markup Language - XML Standard that allow secure web domains to exchange user authentication and authorization data

## Knox Installation



## Objectives

- Install and Configure Apache Knox
- Configure Apache Knox Authentication - LDAP/AD

480 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



**Objectives**

- Install and Configure Apache Knox

481 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Knox with Ambari

- Install Knox Via Add Service Wizard Under “Admin > Stack and Versions”
- Add Service Wizard – Select Knox
- Choose Services – Confirm Knox is Selected
- Assign Master – Choose Hosts to Run Knox
- Customize Services – Create Knox Master Secret Password
- Review Configuration – Deploy
- Monitor Deployment – Install, Start and Test
- Review Summary
- Installation Complete



## Install Apache Knox with Ambari



483 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



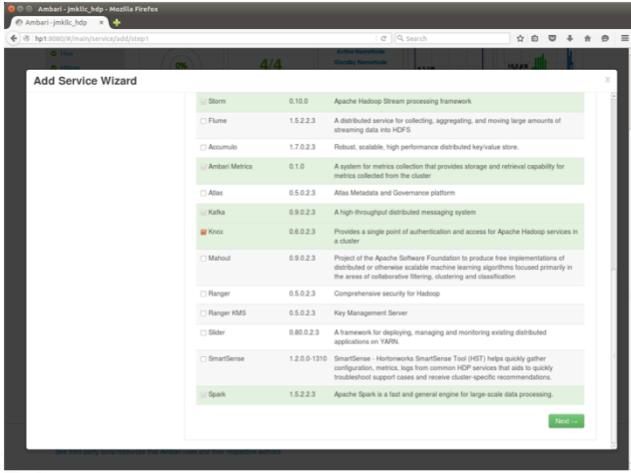
## Install Apache Knox with Ambari

The screenshot shows the Ambari interface with the title "Install Apache Knox with Ambari". A modal window titled "Add Service Wizard" is open, specifically the "Choose Services" step. The modal has a sidebar on the left with steps: "ADD SERVICE WIZARD", "Choose Services", "Assign Masters", "Assign Slaves and Clients", "Customize Services", "Configure Identities", "Review", "Install, Start and Test", and "Summary". The main area of the modal is titled "Choose Services" and contains a table with the following data:

Service	Version	Description
HDFS	2.7.1.2.3	Apache Hadoop Distributed File System
YARN + MapReduce2	2.7.1.2.3	Apache Hadoop NextGen MapReduce (YARN)
Tez	0.7.0.2.3	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
Hive	1.2.1.2.3	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
HBase	1.1.1.2.3	A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications.
Pig	0.15.0.2.3	Scripting platform for analyzing large datasets
Sqoop	1.4.6.2.3	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
Oozie	4.2.0.2.3	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the <a href="#">Elasticsearch</a> Library.
ZooKeeper	3.4.6.2.3	Centralized service which provides highly reliable distributed coordination
Falcon	0.6.1.2.3	Data management and processing platform
Storm	0.10.0	Apache Hadoop Stream processing framework
Flume	1.5.2.2.3	A distributed service for collecting, aggregating, and moving large amounts of

At the bottom right of the modal, there is a blue "Next Step" button. The Ambari interface background shows a green bar at the top with the title and some navigation icons. The footer of the page contains the text "484 © Hortonworks Inc. 2011 – 2016. All Rights Reserved" and a small "UNIVERSITY" logo.

## Install Apache Knox with Ambari



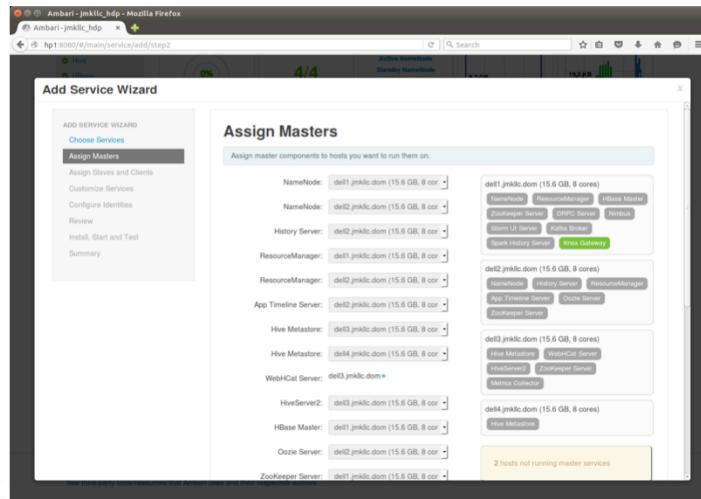
The screenshot shows the Ambari Add Service Wizard interface. A list of services is displayed, with 'Knox' selected. The 'Next >' button is visible at the bottom right of the dialog.

Service	Version	Description
Storm	0.10.0	Apache Hadoop Stream processing framework
Flume	1.5.2.2.3	A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS
Accumulo	1.7.0.2.3	Robust, scalable, high-performance distributed key-value store
Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
Atlas	0.5.0.2.3	Atlas Metadata and Governance platform
Kafka	0.8.0.2.3	A high-throughput distributed messaging system
<b>Knox</b>	0.6.0.2.3	Provides a single point of authentication and access for Apache Hadoop services in a cluster
Mahout	0.8.0.2.3	Project of the Apache Software Foundation to produce free implementations of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification
Ranger	0.5.0.2.3	Comprehensive security for Hadoop
Ranger KMS	0.5.0.2.3	Key Management Server
Slider	0.80.0.2.3	A framework for deploying, managing and monitoring existing distributed applications on YARN
SmartSense	1.2.0.0-1210	SmartSense - Hortonworks SmartSense Tool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations
Spark	1.5.2.2.3	Apache Spark is a fast and general engine for large-scale data processing

485 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



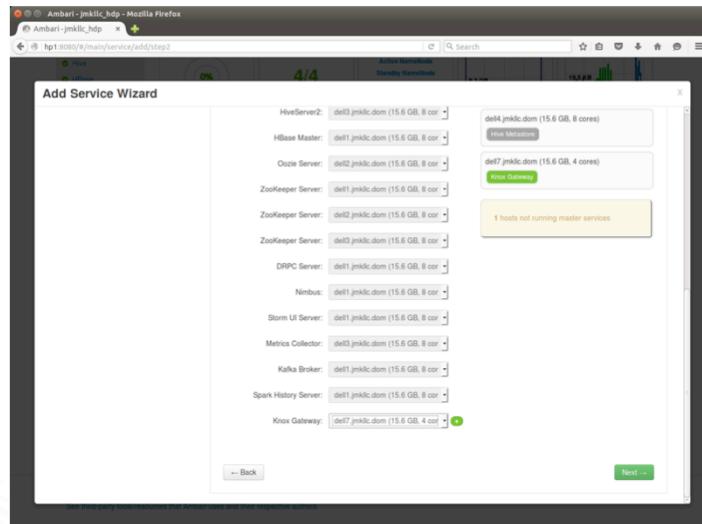
# Install Apache Knox with Ambari



486 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



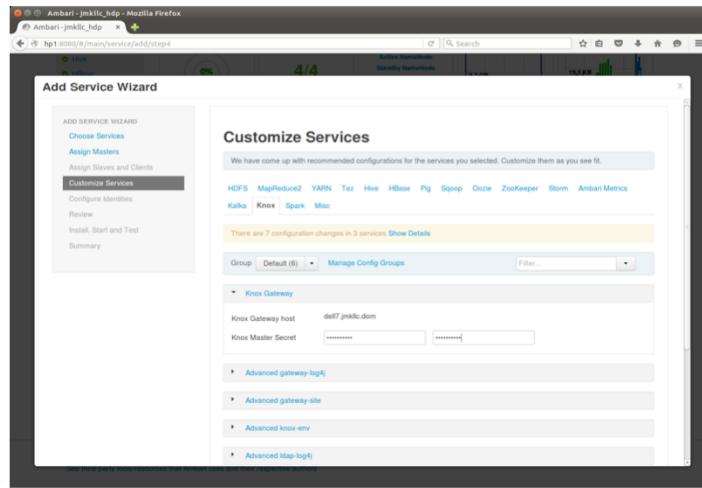
## Install Apache Knox with Ambari



487 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

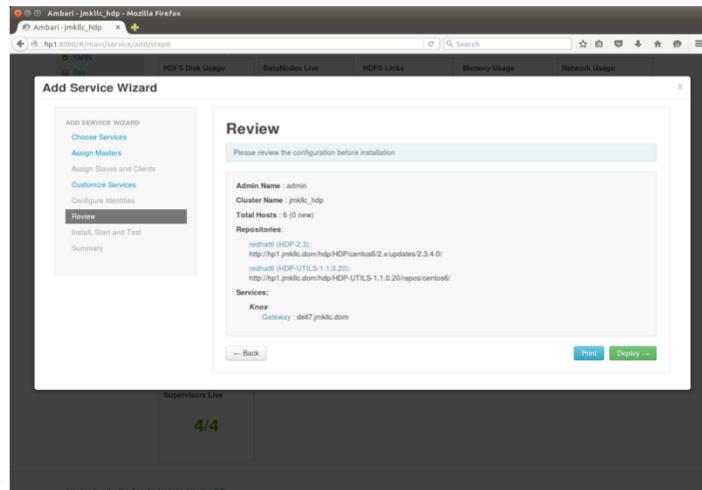


## Install Apache Knox with Ambari



488 © Hortonworks Inc. 2011–2016. All Rights Reserved

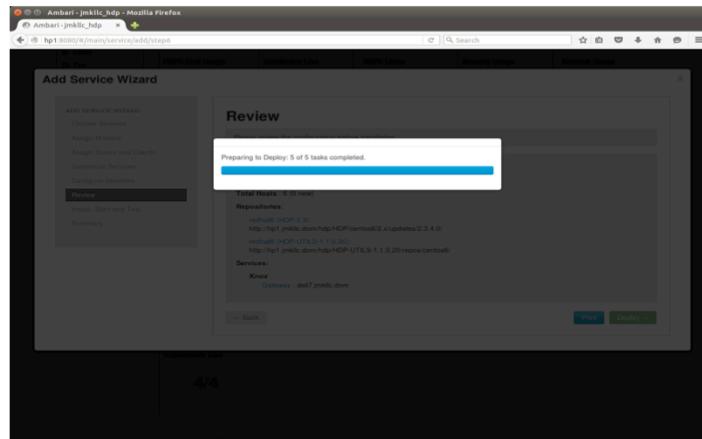
## Install Apache Knox with Ambari



489 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



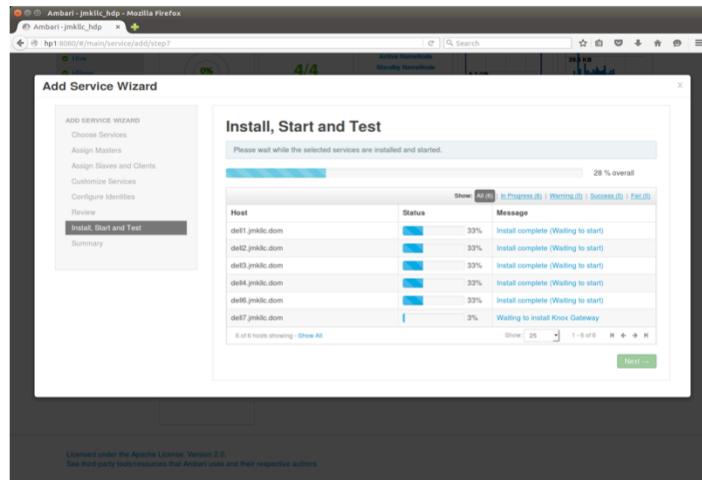
## Install Apache Knox with Ambari



490 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Knox with Ambari



491 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



# Install Apache Knox with Ambari

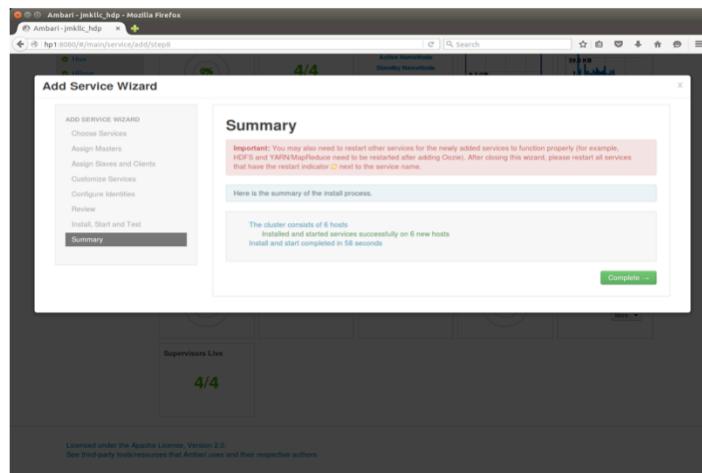
The screenshot shows the Ambari interface for installing Apache Knox. The left sidebar lists steps: Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Configure Identities, Review, Install, Start and Test (which is selected), and Summary. The main panel is titled 'Install, Start and Test' and contains a message: 'Please wait while the selected services are installed and started.' Below this is a progress bar at 100% overall. A table lists hosts and their status: def1.jnklic.dom, def2.jnklic.dom, def3.jnklic.dom, def4.jnklic.dom, def6.jnklic.dom, and def7.jnklic.dom, all marked as 'Success' at 100%. At the bottom, it says 'Successfully installed and started the services.' and has a 'Next >' button.

Host	Status	Message
def1.jnklic.dom	100%	Success
def2.jnklic.dom	100%	Success
def3.jnklic.dom	100%	Success
def4.jnklic.dom	100%	Success
def6.jnklic.dom	100%	Success
def7.jnklic.dom	100%	Success

492 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



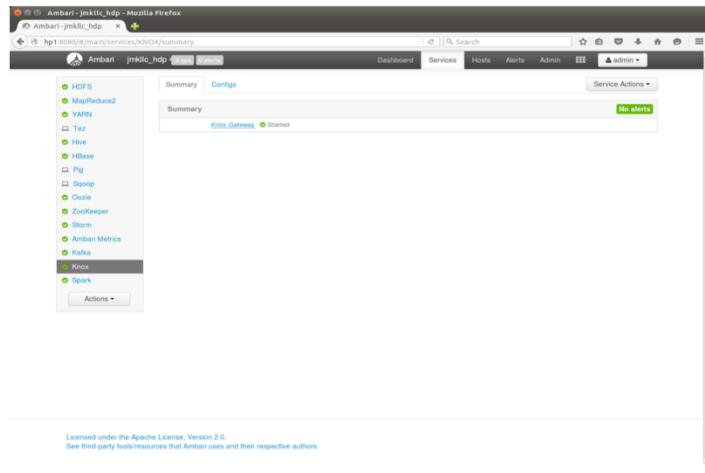
# Install Apache Knox with Ambari



493 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Install Apache Knox with Ambari



494 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Objectives



- Install and Configure Apache Knox
- Configure Apache Knox



## Configure Apache Knox

- Customize Gateway Port and Path
- Properties Effect the URL Used by External Clients
- Default Port is Set to 8443
- Default Context Path is “gateway”  
`https://knox.hortonworks.com:8443/gateway/default/...`
- To Change Port or Context Path – Edit gateway-site.xml File

Property	Value
gateway.port	8443
gateway.path	gateway

- Restart Service



496 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

## Configure Apache Knox

The screenshot shows the Ambari web interface for configuring the Apache Knox service. The URL is `node1:30800/n/main/services/KNOX/configs`. The configuration page is titled "Knox Gateway". Key configuration parameters visible include:

- Knox Master Secret: Two input fields for "base" and "salt".
- Advanced gateway-site:
  - gateway.gateway.conf.dir: deployments
  - gateway.hadoop.kerberos.secured: true
  - gateway.path: gateway
  - gateway.port: 8443
  - java.security.auth.login.config: /etc/knox/conf/kerb5JAASLogin.conf
  - java.security.krb5.conf: /etc/knox/conf/krb5.conf
  - sun.security.krb5.debug: true

497 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Configure Apache Knox

- Master Secret is Required to Start Gateway
- Protects Artifacts Used by Gateway Instance
  - Keystore
  - Trust Stores
  - Credential Stores
- Ambari Persist the Master Secret in “master” File  
/usr/hdp/<HDP VERSION>/knox/data/security/master
- Secret File Encrypted with AES 128 Bit Encryptions
- File is Owned by and Readable/Writable by Knox User

498 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Configure Apache Knox

- Defining Cluster Topologies – Ambari Install Provides Default Topology
- Gateway Supports One or More Hadoop Clusters
- Each Cluster Configuration is Defined in a Topology Deployment Descriptor File
  - /usr/hdp/<HDP VERSION>/knox/conf/topologies
- Deployed to Corresponding WAR File
  - /usr/hdp/<HDP VERSION>/knox/data/deployments
- These Files Define How Gateway Communicate with Cluster
- Descriptor File is XML File with Following Sections
  - gateway/provider – Configuration Settings Enforced by Gateway While Providing Access
  - service – Defines Hadoop Service URL's Used to Proxy Communication From External Clients
- Automatically Redeploys when Detection of New/Change Topology Descriptor File

499 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



# Configure Apache Knox

## Cluster Topology Provider/Service Roles

Type	Role	Description
gateway/provider	hostmap	Maps external to internal node hostnames, replacing the internal hostname with the mapped external name when the hostname is embedded in a response from the cluster.
	authentication	Integrates an LDAP store to authenticate external requests accessing the cluster via the Knox Gateway. Refer to Set Up LDAP Authentication for more information.
	federation	Defines HTTP header authentication fields for an SSO or federation solution provider. Refer to Set up HTTP Header Authentication for Federation/SSO.
	identity-assertion	Responsible for the way that the authenticated user's identity is asserted to the service that the request is intended for. Also maps external authenticated users to an internal cluster that the gateway asserts as the current session user or group. Refer to Configure Identity Assertion for more information.
	authorization	Service level authorization that restricts cluster access to specified users, groups, and/or IP addresses. Refer to Configure Service Level Authorization for more information.
	webappspec	Configures a web application security plugin that provides protection filtering against Cross Site Request Forgery Attacks. Refer to Configure Web Application Security for more information.
	HA provider	Syncs all Knox instances to use the same topologies credentials keystores.
service	\$service_name	Binds a Hadoop service with an internal URL that the gateway uses to proxy requests from external clients to the internal cluster services. Refer to Configure Hadoop Service URLs for more information.

500 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



gateway/provider                  hostmap                  Maps external to internal  
node hostnames, replacing the internal hostname with the mapped external name  
when the hostname is embedded in a response from

the cluster.

authentication                  Integrates an LDAP store  
to authenticate external requests accessing the cluster via the Knox Gateway. Refer  
to Set Up LDAP Authentication for more  
information.

federation                  Defines  
HTTP header authentication fields for an SSO or federation solution provider. Refer to  
Set up HTTP Header Authentication for Federation/SSO

identity-assertion  
Responsible for the way that the authenticated user's identity is  
asserted to the service that the request is intended for. Also maps external  
authenticated users

## Configure Apache Knox

- Setting Up Hadoop Service URL's
- To Configure Access to Internal Hadoop Service – Edit  
/usr/hdp/<HDP VERSION>/knox/conf/topologies/<CLUSTER NAME>.xml
- Add/Modify Entry for each Hadoop Service

```
<topology>
  <gateway>
    ...
    </gateway>
    <service>
      <role>NAMENODE</role>
      <url>hdfs://node1.hortonworks.com:8020</url>
    </service>
  </topology>
```
- Gateway Creates New WAR File with Modified Timestamp in  
/usr/hdp/<HDP VERSION>/knox/data/deployments

501 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



To configure access to an internal Hadoop service through the Knox Gateway:

Edit \$gateway/conf/topologies\$cluster-name.xml to add an entry similar to the following, for each Hadoop service:

```
<topology>
  <gateway>
    ...
    </gateway>
    <service>
      <role> $service_name </role>
      <url> $schema://$hostname:$port</url>
    </service>
  </topology>
```

where:

## Objectives



- Install and Configure Apache Knox
- Configure Apache Knox
- Configure Apache Knox Authentication



## Configure Apache Knox Authentication

- Setting Up LDAP/AD Authentication
- Knox Gateway uses Apache Shiro to Authenticate Users Against LDAP Store
- Provides HTTP BASIC Authentication against LDAP User Directory
- Currently Support Only a Single Organizational Unit (OU)
- Does Not Support Nested OUs
- To Enable Add "ShiroProvider" authentication provider to Cluster's Topology File



LDAP authentication is configured by adding a "ShiroProvider" authentication provider to the cluster's topology file. When enabled, the Knox Gateway uses Apache Shiro (`org.apache.shiro.realm.ldap.JndiLdapRealm`) to authenticate users against the configured LDAP store.

Knox Gateway provides HTTP BASIC authentication against an LDAP user directory. It currently supports only a single Organizational Unit (OU) and does not support nested OUs.

# Configure Apache Knox Authentication

## Enable LDAP Authentication

- Create Keystore Alias for the LDAP Manager User
  - Read Password for Use In knoxcli.sh Script
  - Handy Way to Set Environment Variable without Storing in Shell Command History

```
$ read -s -p "Password: " knoxpass
```
  - Create Password Alias for Knox called knoxLdapSystemPassword

```
$ sudo sudo -u knox /usr/hdp/current/knox-server/bin/knoxcli.sh create-alias knoxLdapSystemPassword --cluster default --value ${knoxpass}
```
  - unset knoxpass
- Add Following LDAP Entries in Topology

504 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



LDAP authentication is configured by adding a "ShiroProvider" authentication provider to the cluster's topology file. When enabled, the Knox Gateway uses Apache Shiro (`org.apache.shiro.realm.ldap.JndiLdapRealm`) to authenticate users against the configured LDAP store.

Knox Gateway provides HTTP BASIC authentication against an LDAP user directory. It currently supports only a single Organizational Unit (OU) and does not support nested OUs.

To enable LDAP authentication:

Open the cluster topology descriptor file, `$cluster-name.xml`, in a text editor.

Add the ShiroProvider authentication provider to `/topology/gateway` as follows:

```
<provider>
```

# Configure Apache Knox Authentication

Locate the Following Section

```
<param>
  <name>main.ldapRealm</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm</value>
</param>

<!-- Add these changes for AD/user sync -->
<param>
  <name>main.ldapContextFactory</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapContextFactory</value>
</param>

<!-- main.ldapRealm.contextFactory needs to be placed before other main.ldapRealm.contextFactory* entries -->
<param>
  <name>main.ldapRealm.contextFactory</name>
  <value>$ldapContextFactory</value>
</param>
```



505 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

```
<param>
  <name>main.ldapRealm</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm</
value>
</param>

<!-- changes for AD/user sync -->

<param>
  <name>main.ldapContextFactory</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapContextFactory</value>
</param>

<!-- main.ldapRealm.contextFactory needs to be placed before other
main.ldapRealm.contextFactory* entries -->
```

## Configure Apache Knox Authentication

```
<!-- Add these change for AD url -->
<param>
  <name>main.ldapRealm.contextFactory.url</name>
  <value>ldap://ad01.lab.hortonworks.net:389</value>
</param>
<!-- system user -->
<param>
  <name>main.ldapRealm.contextFactory.systemUsername</name>
  <value>cn=ldap-reader,ou=ServiceUsers,dc=lab,dc=hortonworks,dc=net</value>
</param>
<!-- pass in the password using the alias created earlier -->
<param>
  <name>main.ldapRealm.contextFactory.systemPassword</name>
  <value>${ALIAS=knoxLdapSystemPassword}</value>
</param>
```

506 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



```
<param>
  <name>main.ldapRealm</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm</
value>
</param>

<!-- changes for AD/user sync -->

<param>
  <name>main.ldapContextFactory</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapContextFactory</value>
</param>

<!-- main.ldapRealm.contextFactory needs to be placed before other
main.ldapRealm.contextFactory* entries -->
```

# Configure Apache Knox Authentication

Locate the Following Section

```
<param>
  <name>main.ldapRealm.contextFactory.authenticationMechanism</name>
  <value>simple</value>
</param>
<param>
  <name>urls/**</name>
  <value>authcBasic</value>
</param>
<!-- Add these changes for AD groups of users to allow --&gt;
&lt;param&gt;
  &lt;name&gt;main.ldapRealm.searchBase&lt;/name&gt;
  &lt;value&gt;ou=CorpUsers,dc=lab,dc=hortonworks,dc=net&lt;/value&gt;
&lt;/param&gt;</pre>
```

507 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



```
<param>
  <name>main.ldapRealm</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm</
value>
</param>

<!-- changes for AD/user sync --&gt;

&lt;param&gt;
  &lt;name&gt;main.ldapContextFactory&lt;/name&gt;
  &lt;value&gt;org.apache.hadoop.gateway.shirorealm.KnoxLdapContextFactory&lt;/value&gt;
&lt;/param&gt;

<!-- main.ldapRealm.contextFactory needs to be placed before other
main.ldapRealm.contextFactory* entries --&gt;</pre>
```

## Configure Apache Knox Authentication

```
<param>
  <name>main.ldapRealm.userObjectClass</name>
  <value>person</value>
</param>
<param>
  <name>main.ldapRealm.userSearchAttributeName</name>
  <value>sAMAccountName</value>
</param>
<!-- changes needed for group sync-->
<param>
  <name>main.ldapRealm.authorizationEnabled</name>
  <value>true</value>
</param>
<param>
  <name>main.ldapRealm.groupSearchBase</name>
  <value>ou=CorpUsers,dc=lab,dc=hortonworks,dc=net</value>
</param>
```

508 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



```
<param>
  <name>main.ldapRealm</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm</
value>
</param>

<!-- changes for AD/user sync -->

<param>
  <name>main.ldapContextFactory</name>
  <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapContextFactory</value>
</param>

<!-- main.ldapRealm.contextFactory needs to be placed before other
main.ldapRealm.contextFactory* entries -->
```

## Configure Apache Knox Authentication

```
<param>
  <name>main.ldapRealm.groupObjectClass</name>
  <value>group</value>
</param>
<param>
  <name>main.ldapRealm.groupIdAttribute</name>
  <value>cn</value>
</param>


- Save the Advanced Topology
- Restart Knox Service

```



## Configure Apache Knox Authentication

- HDFS Configuration for Knox
- Allow Knox to Access Cluster on User Behalf
- Edit HDFS Custom Core-Site Add:  

```
hadoop.proxyuser.knox.groups=users,hadoop-admins,sales,hr,legal  
hadoop.proxyuser.knox.hosts=*
```
- Restart HDFS
  - The Impersonation is needed otherwise the Following Error is Displayed:  

```
org.apache.hadoop.security.authorize.AuthorizationException: User: knox is not allowed to impersonate sales1"
```

510 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Tell Hadoop to allow our users to access Knox from any node of the cluster. Make the below change in Ambari > HDFS > Config > Custom core-site

```
hadoop.proxyuser.knox.groups=users,hadoop-admins,sales,hr,legal
```

```
hadoop.proxyuser.knox.hosts=*
```

(better would be to put a comma separated list of the FQDNs of the hosts)

Now restart HDFS

Without this step you will see an error like below when you run the WebHDFS request later on:

```
org.apache.hadoop.security.authorize.AuthorizationException: User: knox is not allowed to impersonate sales1"
```

## Configure Apache Knox Authentication

- Ranger Configuration for Knox
- Setup Knox Policy for “sales” Group Access Via WEBHDFS
- Login to Ranger - Access Manager > Knox
- Click on Add New Policy
  - Policy name: webhdfs
  - Topology name: default
  - Service name: WEBHDFS
  - Group permissions: sales
  - Permission: check Allow
  - Add

511 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



### Ranger Configuration for Knox

Setup a Knox policy for sales group for WEBHDFS by:

Login to Ranger > Access Manager > KNOX > click the cluster name link > Add new policy

Policy name: webhdfs

Topology name: default

Service name: WEBHDFS

Group permissions: sales

Permission: check Allow

Add

## Configure Apache Knox Authentication

The screenshot shows the Ranger Access Manager interface with the following details:

- Policy Details:**
  - Policy Name: webhdfs
  - Knox Topology: default (Include)
  - Knox Service: WEBHDFS (Include)
  - Description: (empty)
  - Audit Logging: YES
- User and Group Permissions:**
  - Permissions: sales
  - Select Group: sales
  - Select User: (empty)
  - Policy Conditions: Add Conditions
  - Permissions: Allow (selected)
  - Delegate Admin: (checkbox)



512 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

### Ranger Configuration for Knox

Setup a Knox policy for sales group for WEBHDFS by:

Login to Ranger > Access Manager > KNOX > click the cluster name link > Add new policy

Policy name: webhdfs

Topology name: default

Service name: WEBHDFS

Group permissions: sales

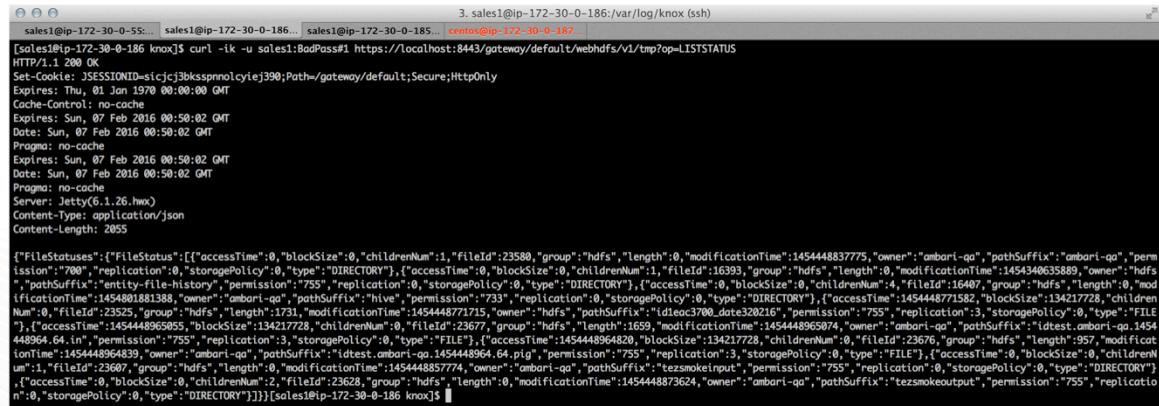
Permission: check Allow

Add

# Configure Apache Knox Authentication

## Testing Provider

```
$ curl -ik -u sales1:BadPass#1 https://localhost:8443/gateway/default/webhdfs/v1/tmp?op=LISTSTATUS
```



```
[sales1@ip-172-30-0-186 ~] curl -ik -u sales1:BadPass#1 https://localhost:8443/gateway/default/webhdfs/v1/tmp?op=LISTSTATUS
HTTP/1.1 200 OK
Set-Cookie: JSESSIONID=cicjcj3bksspmolcyje390;Path=/gateway/default;Secure;HttpOnly
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: no-cache
Expires: Sun, 07 Feb 2016 00:50:02 GMT
Date: Sun, 07 Feb 2016 00:50:02 GMT
Pragma: no-cache
Server: Jetty(6.1.26.wmx)
Content-Type: application/json
Content-Length: 2055

{"FileStatuses": [{"accessTime": 0, "blockSize": 0, "childrenNum": 1, "fileId": 23580, "group": "hdfs", "length": 0, "modificationTime": 145444883775, "owner": "ambari-qa", "pathSuffix": "ambari-qa", "permission": "700", "replication": 0, "storagePolicy": 0, "type": "DIRECTORY"}, {"accessTime": 0, "blockSize": 0, "childrenNum": 1, "fileId": 16393, "group": "hdfs", "length": 0, "modificationTime": 145434063589, "owner": "hdfs", "pathSuffix": "entity-file-history", "permission": "755", "replication": 0, "storagePolicy": 0, "type": "DIRECTORY"}, {"accessTime": 0, "blockSize": 0, "childrenNum": 4, "fileId": 16407, "group": "hdfs", "length": 0, "modificationTime": 14544801881388, "owner": "ambari-qa", "pathSuffix": "hive", "permission": "733", "replication": 0, "storagePolicy": 0, "type": "DIRECTORY"}, {"accessTime": 0, "blockSize": 0, "childrenNum": 1, "fileId": 23525, "group": "hdfs", "length": 1731, "modificationTime": 145444871715, "owner": "hdfs", "pathSuffix": "idlec3700_dote320216", "permission": "755", "replication": 3, "storagePolicy": 0, "type": "FILE"}, {"accessTime": 0, "blockSize": 0, "childrenNum": 0, "fileId": 23677, "group": "hdfs", "length": 1659, "modificationTime": 14544489659874, "owner": "ambari-qa", "pathSuffix": "idtest.ambari-qa.145444896464.in", "permission": "755", "replication": 3, "storagePolicy": 0, "type": "FILE"}, {"accessTime": 0, "blockSize": 0, "childrenNum": 0, "fileId": 23676, "group": "hdfs", "length": 957, "modificationTime": 145444896482, "blockSize": 134217228, "owner": "ambari-qa", "pathSuffix": "idtest.ambari-qa.1454448964839", "permission": "755", "replication": 3, "storagePolicy": 0, "type": "FILE"}, {"accessTime": 0, "blockSize": 0, "childrenNum": 0, "fileId": 23628, "group": "hdfs", "length": 0, "modificationTime": 145444885774, "owner": "ambari-qa", "pathSuffix": "tezsmokeinput", "permission": "755", "replication": 0, "storagePolicy": 0, "type": "DIRECTORY"}, {"accessTime": 0, "blockSize": 0, "childrenNum": 2, "fileId": 23628, "group": "hdfs", "length": 0, "modificationTime": 1454448873624, "owner": "ambari-qa", "pathSuffix": "tezsmokeoutput", "permission": "755", "replication": 0, "storagePolicy": 0, "type": "DIRECTORY"}]}[sales1@ip-172-30-0-186 knox]$
```

513 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Using cURL, you can test your LDAP configuration as follows:

Open the command line on an external client.

Enter the following command to list the contents of the directory tmp/test:

```
curl -i -k -u ldap_user : password -X GET / 'https:// gateway_host :8443/gateway_path / cluster_name /webhdfs/api/v1/tmp/test?op=LISTSTATUS'
```

If the directory exists, a content list displays; if the user cannot be authenticated, the request is rejected with an HTTP status of 401 unauthorized.

# Lab: Knox





## Ambari Views for Controlled Access

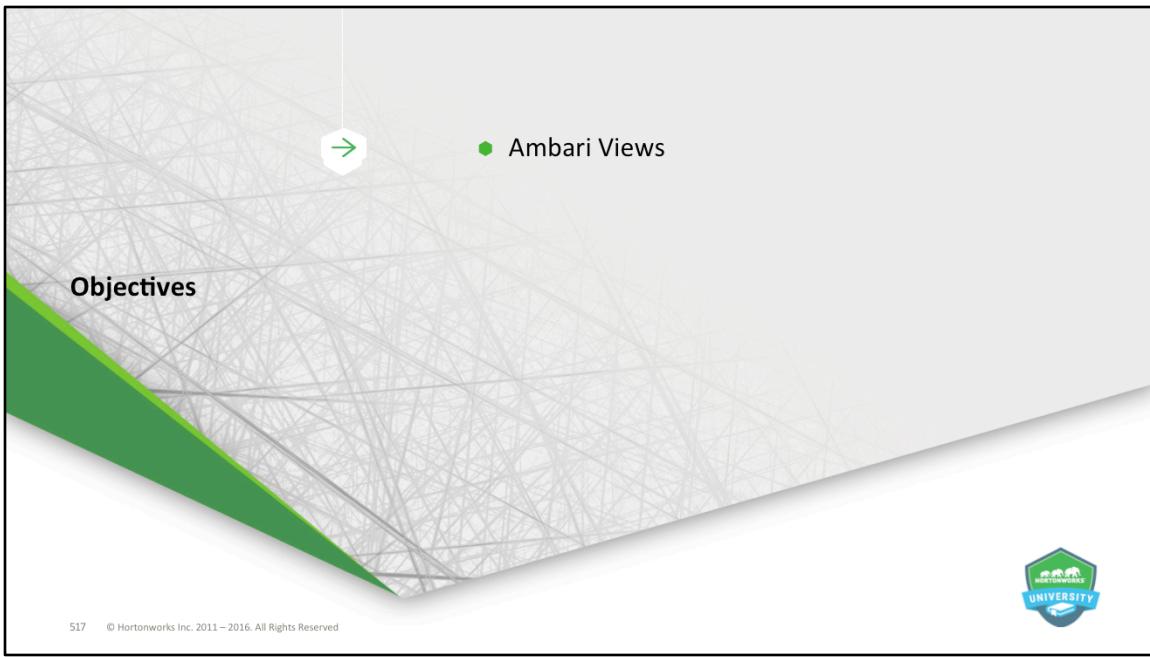


## Lesson Objectives

- Summarize the purpose of Ambari Views
- Configure a Standalone Ambari Views Server
- Configure Ambari Views Server for Kerberos
- Enable Kerberos for Views

516 © Hortonworks Inc. 2011 – 2016. All Rights Reserved





## Ambari Views

- Framework to Allow Developers to Create UI Components that “Plug-Into” Ambari Web UI
- Included are a Built-In Set of Views that are Pre-Deployed for Use
- Views Can Be Deployed/Managed in “Operational” Ambari Server
- View Can Also Be Deployed/Manager in One or More Separate “Standalone” Servers
- Standalone Instances
  - Useful When Users Will Access Cluster Via Views
  - Useful For User Who Should Not Have Access to “Operational” Ambari Server
  - Allow A Scale-Out Approach for Handling Large Number of Users

518 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Objectives



- Ambari Views
- Setup Standalone Ambari Views Server

519 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Setup Standalone Ambari View Server

### Prerequisites

- Instances Should Be Same Version
- Point to the Same Underlying Database - Not DB Being Used by Operational Server
- Database Should Be Scaled and Highly-Available
- Authentication LDAP/AD Should Be Configured the Same for All Instances
- Kerberos-Enabled Clusters Require Instances Configured for Kerberos
- Run Multiple Instances Behind Reverse Proxy



## Setup Standalone Ambari View Server

- Setup Similar to Operational Ambari Server
- Does Not Install/Manage a Cluster
- Does Not Deploy/Communicate with Ambari Agents
- Run as Web Server Instance Serving View for Users
- Must Be Configured Same as Operational Ambari Server
  - Configure Server for Non-Root
  - Encrypt Database and Passwords
  - Enable LDAP/AD Authentication
  - Enable HTTPS/SSL Server
  - Enable SPNEGO Authentication for Hadoop



## Setup Standalone Ambari View Server

- Installation Overview
- Install Ambari-Server Package
- Execute Ambari Setup
- Configure LDAP/AD Authentication
- Configure Server for Non-Root
- Encrypt Database/Passwords
- Configure HTTPS/SSL
- Configure SPNEGO Authentication for Hadoop
- Deploy Views
- Create and Configure View Instances
- Repeat Process for Additional View Instances

522 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Setup Standalone Ambari View Server

- Strongly Recommend to Increase Amount of Memory Available When Host Multiple Views
- Each View Requires It's Own Memory Footprint
- Increasing Maximum Allocable Memory Helps Support Multiple Views and Concurrent Use
- Recommended Values in /var/lib/ambari-server/ambari-env.sh  
AMBARI\_JVM\_ARGS=-Xmx4096m -XX:PermSize=128m -XX:MaxPermSize=128m
- Restart Ambari Server



## Objectives



- Ambari Views
- Setup Standalone Ambari Views Server
- Configure Ambari Views for Kerberos



## Configure Ambari Views for Kerberos

- All Ambari Server Instance Must Be Kerberos Enabled
- Refer to “Set Up Kerberos for Ambari” - Module 6
- Install Kerberos Client Utilities on All Ambari Server Instances
  - # yum install krb5-workstation
- Ambari Will Need to “kinit” to Kerberos KDC
- Follow Specific Instructions to Configure Each View for Kerberos
- Views that Require a Proxy User to Be Configured, Use Ambari’s Kerberos Principal  
ambari@LAB.HORTONWORKS.NET

525 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



If the cluster your views will communicate with is Kerberos-enabled, you need to configure the Ambari Server instance(s) for Kerberos and be sure to configure the views to work with Kerberos.

Refer to the Set Up Kerberos for Ambari for the instructions on how to configure Ambari Server for Kerberos. Be sure to configure all standalone Ambari Server instances for Kerberos.

### Important

Be sure to install the Kerberos client utilities on the Ambari Server so that Ambari can kinit.

RHEL/CentOS/Oracle Linux

```
yum install krb5-workstation
```

SLES

## Objectives



- Ambari Views
- Setup Standalone Ambari Views Server
- Configure Ambari Views for Kerberos
- Kerberos Setup for Files View



## Kerberos Setup for Files View

- Setup HDFS Proxy User For Ambari Server Daemon
- Browse to Services > HDFS > Configs
- Advanced Tab – Navigate to Custom Core-Site Section
- Add Property
  - hadoop.proxyuser.ambari-server.groups=\*
  - hadoop.proxyuser.ambari-server.hosts=\*
- Save Change and
- Restart Affected Components
- Create/Configure Files View Instance

527 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



To set up an HDFS proxy user for the Ambari Server daemon account, you need to configure the proxy user in the HDFS configuration. This configuration is determined by the account name the ambari-server daemon is running as. For example, if your ambari-server is running as root, you set up an HDFS proxy user for root with the following:

In Ambari Web, browse to Services > HDFS > Configs.

Under the Advanced tab, navigate to the Custom core-site section.

Click Add Property... to add the following custom properties:

if you have configured Ambari Server for Kerberos, be sure to modify this property name for the primary Kerberos principal user. For example, if ambari-server is setup for Kerberos using principal ambari-server@EXAMPLE.COM, you would use the following properties instead:

## Kerberos Setup for Files View

- Configure File View Instance
- Browse to Ambari Administration Interface
- Click on Views – Expand File Views
- Setting and Cluster Configuration Enter Following Properties

Property	Description	Value
WebHDFS Username	This is the username the view will access HDFS as. Leave this default value intact to represent the authenticated view user.	<code> \${username}</code>
WebHDFS Authorization	This is the semicolon-separated authentication configuration for WebHDFS access.	<code>auto=KERBEROS; proxyuser=ambari-server</code>

528 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



You must first set up Kerberos for Ambari by configuring the Ambari Server daemon with a Kerberos principal and keytab. Refer to Configuring Views for Kerberos for instructions. After you have set up Kerberos for Ambari, in the Settings section of the Files View, enter the following:

Property	Description	Example Value
WebHDFS Username	This is the username the view will access HDFS as. Leave this default value intact to represent the authenticated view user.	<code> \${username}</code>
WebHDFS Authorization	This is the semicolon-separated authentication configuration for WebHDFS access.	

## Objectives



- Ambari Views
- Setup Standalone Ambari Views Server
- Configure Ambari Views for Kerberos
- Kerberos Setup for Files View
- Kerberos Setup for Tez View



## Kerberos Setup for Tez View

- Browse to Services > YARN > Configs
- Advanced Tab – Navigate to Custom Yarn-Site Section
- Add Property

Property	Value
yarn.timeline-server.http-authentication.proxyuser. <b>ambari-server</b> .hosts	*
yarn.timeline-server.http-authentication.proxyuser. <b>ambari-server</b> .user	*
yarn.timeline-server.http-authentication.proxyuser. <b>ambari-server</b> .groups	*
yarn.resourcemanager.webapp.delegation-token-auth-filter.enabled	true
yarn.resourcemanager.proxyuser. <b>ambari-server</b> .hosts	*
yarn.resourcemanager.proxyuser. <b>ambari-server</b> .users	*
yarn.resourcemanager.proxyuser. <b>ambari-server</b> .groups	*

- Restart Affected Components



530 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

yarn.resourcemanager.webapp.delegation-token-auth-filter.enabled

yarn.timeline-service.http-authentication.proxyuser.\${ambari principal name}.hosts

yarn.timeline-service.http-authentication.proxyuser.\${ambari principal name}.groups

yarn.timeline-service.http-authentication.proxyuser.\${ambari principal name}.users

The following settings are optional, if proxyusers settings are added in core-site.xml:

yarn.resourcemanager.proxyuser.\${ambari principal name}.hosts

yarn.resourcemanager.proxyuser.\${ambari principal name}.users

yarn.resourcemanager.proxyuser.\${ambari principal name}.groups

# Kerberos Setup for Tez View

- Browse to Services > HDFS > Configs
  - Advanced Tab – Navigate to Core-Site Section
  - Verify Following Properties – Add As Necessary

Property	Value
hadoop.http.authentication.type	kerberos
hadoop.http.filter.initializers	org.apache.hadoop.security.AuthenticationFilterInitializer
hadoop.http.authentication.kerberos.keytab	/etc/security/keytabs/spnego.service.keytab
hadoop.http.authentication.kerberos.principal	HTTP/_HOST@HORTONWORKS.COM
hadoop.http.authentication.signature.secret.file	/etc/hadoop/conf/secret_http_file
hadoop.http.authentication.cookie.domain	hortonworks.com
hadoop.proxyuser.ambari-server.hosts	*
hadoop.proxyuser.ambari-server.groups	*
hadoop.proxyuser.ambari-server.users	*

531 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

Property	Value
hadoop.http.authentication.type	kerberos
hadoop.http.filter.initializers	org.apache.hadoop.security.AuthenticationFilterInitializer
hadoop.http.authentication.kerberos.keytab HTTP principal>	<Path to keytab container
spnego.service.keytab>	for example: <etc/security/keytabs/
hadoop.http.authentication.kerberos.principal	HTTP/_HOST@REALM
	for example: HTTP/_HOST@EXAMPLE.COM

## Kerberos Setup for Tez View

- Restart Affected Components
- Create/Configure Tez View Instance

532 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



Property	Value
hadoop.http.authentication.type	kerberos
hadoop.http.filter.initializers	org.apache.hadoop.security.AuthenticationFilterInitializer
hadoop.http.authentication.kerberos.keytab HTTP principal>	<Path to keytab container for example: <etc/security/keytabs/ spnego.service.keytab>
hadoop.http.authentication.kerberos.principal	HTTP/_HOST@REALM for example: HTTP/_HOST@EXAMPLE.COM

## Objectives



- Ambari Views
- Setup Standalone Ambari Views Server
- Configure Ambari Views for Kerberos
- Kerberos Setup for Files View
- Kerberos Setup for Tez View
- Kerberos Setup for Pig View



## Kerberos Setup for Pig View

- Setup HDFS Proxy User For Ambari Server Daemon – Already Done for Files View
- Browse to Services > HDFS > Configs
- Advanced Tab – Navigate to Custom Core-Site Section
- Add Property
  - hadoop.proxyuser.ambari-server.groups=\*
  - hadoop.proxyuser.ambari-server.hosts=\*
- Save Change and
- Restart Affected Components

534 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



To set up an HDFS proxy user for the Ambari Server daemon account, you need to configure the proxy user in the HDFS configuration. This configuration is determined by the account name the ambari-server daemon is running as. For example, if your ambari-server is running as root, you set up an HDFS proxy user for root with the following:

In Ambari Web, browse to Services > HDFS > Configs.

Under the Advanced tab, navigate to the Custom core-site section.

Click Add Property... to add the following custom properties:

if you have configured Ambari Server for Kerberos, be sure to modify this property name for the primary Kerberos principal user. For example, if ambari-server is setup for Kerberos using principal ambari-server@EXAMPLE.COM, you would use the following properties instead:

## Kerberos Setup for Pig View

- Setup HDFS Proxy User For WebHCat
- Setup WebHCat Proxy User for Ambari Server Daemon
- Browse to Services > HDFS > Configs
- Advanced Tab – Navigate to Custom Core-Site Section
- Add Property
  - hadoop.proxyuser.hcat.groups=\*
  - hadoop.proxyuser.hcat.hosts=\*
  - webhcatt.proxyuser.ambari-server.groups=\*
  - webhcatt.proxyuser.ambari-server.hosts=\*
- Save Change and
- Restart Affected Components
- Create/Configure Pig View Instance

535 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



You must set up an HDFS proxy user for WebHCat and a WebHCat proxy user for the Ambari Server daemon account.

To setup the HDFS proxy user for WebHCat :

In Ambari Web, browse to Services > HDFS > Configs.

Under the Advanced tab, navigate to the Custom core-site section.

Click Add Property... to add the following custom properties:

```
hadoop.proxyuser.hcat.groups=*
hadoop.proxyuser.hcat.hosts=*
```

Save the configuration change and restart the required components as indicated by Ambari.

## Kerberos Setup for Pig View

- Pig View Requires WebHDFS Authentication to be Set

Settings

WebHDFS Username	<code> \${username}</code>
WebHDFS Authentication	<code>auth=KERBEROS;proxyuser=ambari-server</code>
WebHCat Username	
Scripts HDFS Directory*	<code>/user/\${username}/pig/scripts</code>
Jobs HDFS Directory*	<code>/user/\${username}/pig/jobs</code>
Meta HDFS Directory	<code>/user/\${username}/pig/store</code>

536 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



After you have set up basic Kerberos for the Pig View, Pig requires that WebHDFS Authentication be set to `auth=KERBEROS;proxyuser=<ambari-user-principal>`

## Objectives



- Ambari Views
- Setup Standalone Ambari Views Server
- Configure Ambari Views for Kerberos
- Kerberos Setup for Files View
- Kerberos Setup for Tez View
- Kerberos Setup for Hive View



## Kerberos Setup for Hive View

- Setup HDFS Proxy User For Ambari Server Daemon – Already Done for Files View
- Browse to Services > HDFS > Configs
- Advanced Tab – Navigate to Custom Core-Site Section
- Add Property
  - hadoop.proxyuser.ambari-server.groups=\*
  - hadoop.proxyuser.ambari-server.hosts=\*
- Save Change and
- Restart Affected Components
- Create/Configure Hive View Instance

538 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



## Kerberos Setup for Hive Views

- Hive View Requires Hive and WebHDFS Authentication to be Set

Settings

Hive Authentication  
auth=KERBEROS;principal=hive/\_HOST@HORTONWORKS.COM;hive.server2.proxy.user=\${username}

WebHDFS Username  
\${username}

WebHDFS Authentication  
auth=KERBEROS;proxyuser=ambari-server

Instance name of Tez view

Scripts HDFS Directory\*  
/user/\${username}/hive/scripts

Jobs HDFS Directory\*  
/user/\${username}/hive/jobs

Default script settings file\*  
/user/\${username}/.\${instanceName}.defaultSettings

539 © Hortonworks Inc. 2011 – 2016. All Rights Reserved

After you have set up basic Kerberos for the Hive View, Hive requires the following additional settings:

Table 8.4. Kerberos Settings for Hive Views

Property	Value
WebHDFS Authentication	auth=KERBEROS;proxyuser=<ambari-principal>
Hive Authentication	KERBEROS

and the principal is set to the same principal that is specified in `hive-site.xml` for `hive.server2.authentication.kerberos.principal`.

For example:

## Knowledge Check



## Knowledge Check

1. True/False - Ambari Servers does not include a set of built-in views?
2. What is the purpose of a standalone Ambari Views server?
3. True/False - Ambari Views Server can not communicate with a Kerberos enabled Hadoop cluster?
4. True/False - Multiple Ambari Views Server can be setup behind a reverse proxy?
5. True/False - The Ambari Views Server are used to manage the Hadoop cluster?

541 © Hortonworks Inc. 2011 – 2016. All Rights Reserved



1. FALSE
2. Used to provide access to user who will access views but will not and should not have access to the operational Ambari Server.
3. FALSE - The Ambari Views Server requires Kerberos configuration as well, in order to communicate with Kerberized Hadoop Cluster.
4. TRUE
5. FALSE - Used to provide access to user.

# Summary



## Summary

- Ambari Views allow developers to create UI components that plug into the Ambari Web interface.
- Ambari Server includes a set of built-in views that are pre-deployed with your cluster.
- Views can be deployed and managed in the “operational” Ambari server that is managing your cluster however, a standalone Ambari Views server is useful when users who will access views will not and should not have access to the operational Ambari server.
- Ambari Views Server requires Kerberos configuration if communicating with a Kerberos-enabled cluster.



## Lab: Other Security Features for Ambari

