Technical Report

Secure Hadoop using Apache Ranger with NetApp In-Place Analytics Module

Deployment Guide

Karthikeyan Nagalingam, NetApp
February 2019 | TR-4744

Abstract

This document introduces the NetApp® In-Place Analytics Module for Apache Hadoop and Spark with Ranger. The topics covered in this report include the Ranger configuration, underlying architecture, integration with Hadoop, and benefits of Ranger with NetApp In-Place Analytics Module using Hadoop with NetApp ONTAP® data management software.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

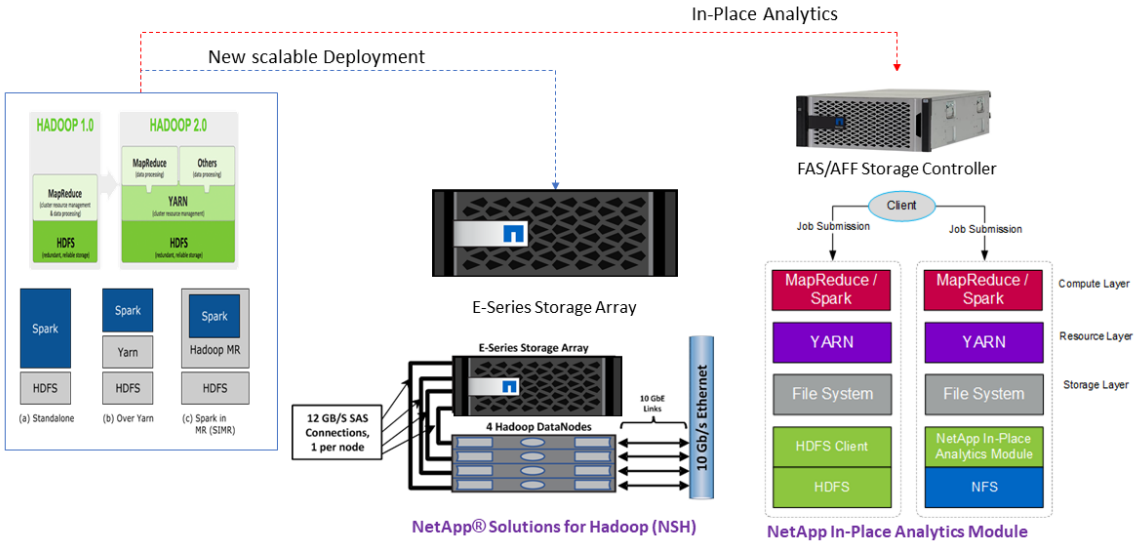
This document describes NetApp In-Place Analytics Module (NIPAM) configurations, with Ranger support, for ecosystem components such as Hadoop Distributed File System (HDFS), YARN, HBase, Hive, and Spark.

1.1 Overview

Scalable Hadoop deployments can be built on NetApp ONTAP with AFF or FAS storage arrays or on NetApp E-Series systems. These solutions are complementary and targeted at different use cases, as highlighted in Figure 1. This document focuses on the use of Ranger with NIPAM connected to NetApp ONTAP on AFF or FAS.

Note: For more information about the NIPAM solution for NetApp ONTAP, see TR-4382: NetApp In-Place Analytics Module. For more information about the E-Series solution, see TR-3969: NetApp Open Solution for Hadoop Solutions.

Figure 1) NetApp In-Place Analytics Module.



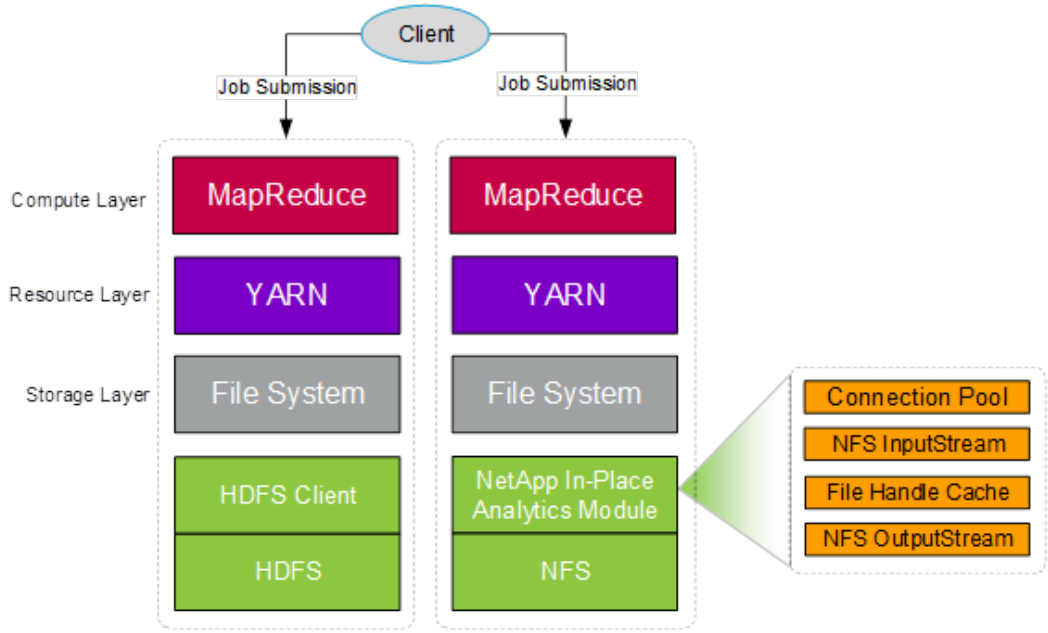
NIPAM enables analytics software to run on the data stored in ONTAP®. It works with Apache Hadoop and Apache Spark by using a simple configuration file change that enables data on NFSv3 storage to be analyzed. By using ONTAP, NIPAM decouples analytics from storage, leveraging the benefits of NAS. ONTAP data management software enables high availability and storage efficiency.

NIPAM is an implementation of the file system interface defined by Apache Hadoop and is based on NFSv3. Currently it supports `AUTH_NONE` and `AUTH_SYS` security policies, which are UNIX authentication mechanisms.

NIPAM consists of a Ranger client communicating with a Ranger admin server to download configured policies. A Ranger policy enforcement engine is built in to enforce policies for the file system API calls or Hadoop file system commands when the `nfs://<server_ip>:2049/` scheme is used with Hadoop and Spark ecosystem components.

Figure 2 shows how NIPAM plugs into Apache Hadoop. This design has two deployment options, which are covered in Deployment Options.

Figure 2) NetApp In-Place Analytics Module plugs into Apache Hadoop.



1.2 Deployment Options

There are two different deployment options for NIPAM:

1. Run the HDFS as the primary file system and use NIPAM to analyze data on the NFS storage systems.
2. Deploy NFS as the primary storage or default file system.

   Note: NetApp does not currently support this option in the NetApp In-Place Analytics Module 3.x release. However, NetApp is actively working with Hadoop distributors to support this option in future versions.

The appropriate deployment option is based on Apache Hadoop and Apache Spark use cases and applications.

1.3 NetApp In-Place Analytics Module 3.0.1 Features

NIPAM version 3.0.1 has the following features:

- **Performance enhancements:** Network load distribution across multiple network interfaces.
- **Hortonworks certification:** Hortonworks certification for the Hortonworks Data Platform (HDP).
- **Apache Ambari module for management:** Manage installation and configuration from the Ambari user interface.
- **Simplicity enhancements:**
  - User and access management
  - Configuration management
- **Azure integration with HDInsight application:** Run Azure HDInsight cluster on NFS data.
- **Apache Ranger support**: Provides secure access to NFS and Hadoop ecosystem components such as HBase, Hive, and YARN.

2 Ranger

Apache Ranger delivers a comprehensive approach to security for a Hadoop cluster. It provides a centralized system to define, administer, and manage security policies consistently across Hadoop components.

Using the Apache Ranger console in Hortonworks, security administrators can easily manage policies for access to files, folders, databases, tables, or columns. These policies can be set for individual users or groups and then enforced consistently across HDP stack.

Ranger also provides security administrators with deep visibility into their Hadoop environment through a centralized audit location that tracks all the access requests in real time and supports multiple destination sources including HDFS and Solr.

2.1 Components Validated with Ranger

Apache Ranger offers a centralized security framework to manage fine-grained access for Apache HDFS, Apache Hive, Apache HBase, Apache Storm, Apache Knox, Apache Kafka, Apache NiFi, and YARN. In this report, we validated NFS such as HDFS, Apache Hive, Apache HBase, and YARN.

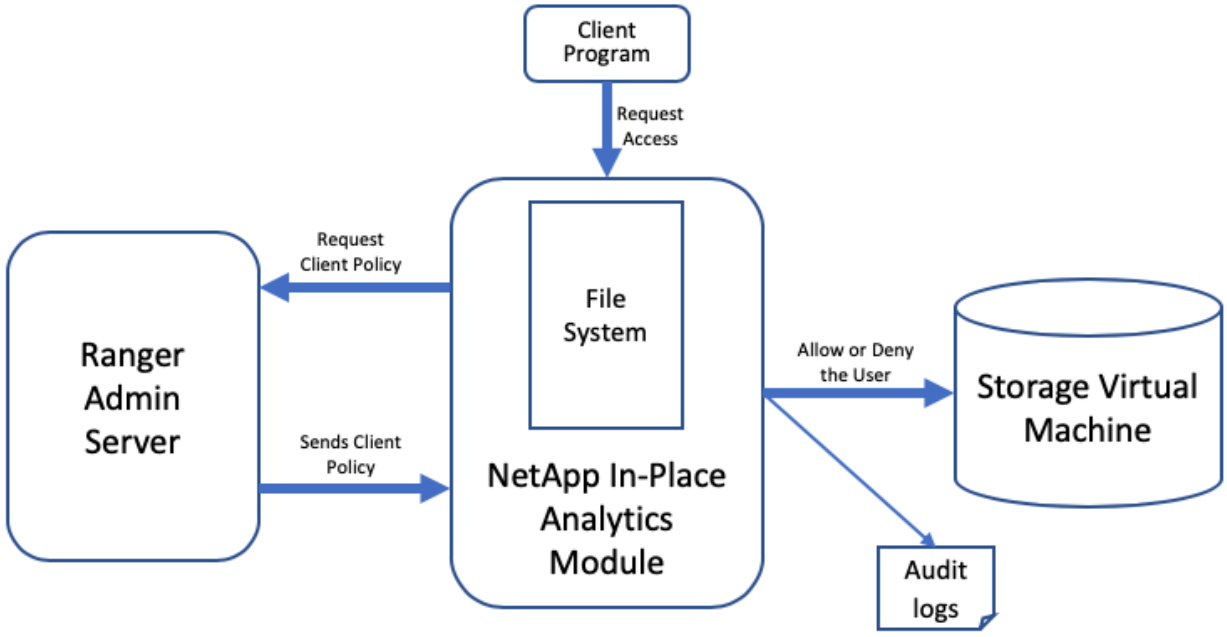
NIPAM uses an HDFS repository to create policies for the NFS folder; only one repository is supported. You can create multiple policies under one HDFS repository because NIPAM uses the HDFS repository name to connect to Ranger server to get the policies.

Note: When you create or modify an HDFS, HBase, YARN, or Hive Ranger policy, you must restart the required services and wait until the specific service type time for the Active, Download, and Last Update columns are the same. It might take five minutes to update some components. From the Ranger UI portal, select Audit > Plugin Status > `<hdp cluster name>_<component_name>`.

3 NetApp In-Place Analytics Module Design with Ranger

In this section, we provide the details about NetApp In-Place Analytics design with Ranger based on a client application request.

Figure 3) Design of NetApp In-Place Analytics Module with Ranger.



The design of NIPAM with Ranger (client application accessing NFS storage):

- The client program submits an application to read and write the data to and from NFS.
- The client request goes to the file system API in NIPAM.
- NIPAM checks that the parameter `IsNfsRangerEnabled` is set to true in its JSON configuration file.
- If the response is yes, the request passes to the Ranger Admin server to collect the Ranger policy for the HDFS repository user who initiated the client program.
- Based on the permissions defined in the Ranger policy for the ONTAP NFS folder, NIPAM either allows or denies the client request for access to NFS export in SVM.
- NIPAM also maintains the audit logs for data access to the `nfsRangerAuditLogPath` location mentioned in the JSON configuration file.

3.1 Technical Advantages

NIPAM with ranger has the following technical advantages:

- It works with NFS in the same way HDFS works. Using Ambari framework, you can manage NFS from the HDFS plug-in framework.
- Hive, HBase, and YARN components are configured identically to HDFS.

For existing deployments, only the configuration file (JSON file used for NIPAM) is modified, everything else is the same.

4 Solution Architecture

4.1 Key Components of Validated Configuration

Table 1 lists the products used for validating NIPAM with Ranger. You can configure the storage based on the NetApp sizer recommendations for your bandwidth (in Mbps), IOPS, and capacity requirements.

Table 1) Components used for the NetApp In-Place Analytics Module tests.

Component	Product or Solution	Details
Storage	NetApp AFF A300 storage array with ONTAP 9.3	• 2 controllers (HA pair) with 24 x 900GB, solid-state drives (SSDs) • 1 hot spare per disk shelf • 1 data aggregate (23 drives, shared to both controllers) per controller • NetApp ONTAP FlexGroup and some test cases with 2 aggregates and 2 volumes • 2 x 4 x 10GbE • RAID DP®
Servers	SuperMicro X11DPi-N(T) Intel Xeon Platinum 8160 CPU @ 2.10GHz	• 8 servers, and each has two 2.10GHz (24-core) • 16 DIMMs 16GB, up to 2400MHz (256GB/node) • 1 x 1Gbps Ethernet port, 1x 10GbE network port
Networking	• 10GbE nonblocking network switch • 1GbE network switch	A Cisco Nexus 5000 switch was used for testing. Any compatible 10GbE network switch can also be used.

Component	Product or Solution	Details
Server operating system	Red Hat Enterprise Linux Server 7.5 (x86_64) or later	Hadoop typically requires a Linux distribution.
Hadoop distribution used in the testing	Cloudera Distribution for Hadoop	Cloudera Manager 5.14
	Hortonworks Data Platform 3.0.1 (tested)	Apache Ambari 2.7.1

5 Installation and Configuration

Installing NIPAM with Ranger is simple and consists of three parts:

3. Configure the FAS or AFF storage controller.
4. Configure the Hadoop cluster.
5. Create the JSON configuration file.

Note: Before you begin, download NIPAM software and installation guide: https://mysupport.netapp.com.

5.1 Configure FAS or AFF Storage Controller

To configure the FAS or AFF storage controller, see TR-4382.

5.2 Configure the Hadoop Cluster

Complete the following steps to configure the Hadoop cluster:

1. Install Hadoop cluster.
2. Add Ranger service in Hortonworks or add Apache Ranger in other Hadoop distributions.
   a. In Hortonworks Hadoop cluster, go to the custom `ranger-admin-site.xml` and update the `ranger.servicedef.enableDenyAndExceptionsInPolicies` parameter to `true` using Ambari framework.

   b. For Apache Hadoop and Spark-based distributions, such as Cloudera and MapR, copy the `hadoop-nfs-connector-3.x.x.jar and ranger-plugins-audit-1.0.0.jar` files to `hadoop classpath` and replace the `hadoop-nfs-<version>.jar` file with the `hadoop-nfs-2.7.1.jar` file provided by NetApp.
3. For Hortonworks, use the Ambari UI plug-in for In-Place Analytics Module for Hadoop to install and configure NIPAM with or without Ranger. Download the Ambari UI plug-in for In-Place Analytics Module for Hadoop software and the Installation and Administration guide from https://mysupport.netapp.com.

5.3 Create JSON Configuration File

You must create a configuration file for NIPAM which is then distributed to all the nodes on your Hadoop cluster. The location of the file can be anywhere on your Hadoop node as long as it is accessible to all the

Hadoop users in the system. See the Installation Guide for a detailed explanation of the JSON file and its parameters at https://mysupport.netapp.com.

For Ranger configuration, you must add the parameters in Table 2 to in NIPAM JSON file along with existing JSON file in the options{} parameter.

Table 2) Ranger configuration (NFS configuration options).

Parameter	Default Value	Example Values	Description
nfsIsRangerEnabled	false	false	Enables Ranger authorization support for NFS.
nfsRangerServiceName	N/A	“hdpmaster_nfsconnector_hadoop”	The HDFS service name used for each NFS namespace configured in Ranger admin server under service manager for resource-based policies for HDFS. **Note:** In this version, we support one service name for HDFS and create multiple policies under a service name.
nfsRangerAdminUrl	N/A	“http://10.141.46.9:6080/"	Ranger admin URL used to import the policies to NFS client instance.
nfsRangerAuditLogPath	N/A	“/usr/nfs/ranger/logs”	Path where the audit logs for this module are stored. **Note:** Create a folder with all the Hadoop members in the cluster.
nfsRangerAuditEnabled	false	false	Enable audit logging for file system API operation calls.

Best Practices
If you add Ranger configuration parameters outside the options{} parameter, the Ranger configuration will not work.

Example Configuration File NetApp In-Place Analytics Module with Ranger

```
[root@stl-kserver-08 ~]# cat /etc/NetAppNFSConnector/conf/nfs-mapping.json
{
    "spaces": [
        {
            "endpoints": [
                {
                    "exportPath": "/nc_ranger1",
                    "hosts": [
                        "nfs://10.63.150.213:2049/",
                        "nfs://10.63.150.123:2049/",
```

```
                        "nfs://10.63.150.124:2049/",
                        "nfs://10.63.150.125:2049/",
                        "nfs://10.63.150.127:2049/"
                    ],
                    "path": "/nc_ranger1"
                }
            ],
            "name": "test1",
            "options": {
                "nfsAuthScheme": "AUTH_SYS",
                "nfsExportPath": "/nc_ranger1",
                "nfsGid": 0,
                "nfsGroupname": "root",
                "nfsIsRangerEnabled": true,
                "nfsMountPort": -1,
                "nfsPort": 2049,
                "nfsRangerConfig": {
                    "nfsRangerAdminUrl": "http://10.63.150.58:6080/",
                    "nfsRangerAuditEnabled": true,
                    "nfsRangerAuditLogPath": "/usr/nipam/ranger/logs",
                    "nfsRangerServiceName": "ntapnipam_hadoop"
                },
                "nfsReadMaxPoolThreads": 128,
                "nfsReadMinPoolThreads": 32,
                "nfsReadSizeBits": 20,
                "nfsRpcbindPort": 111,
                "nfsSplitSizeBits": 30,
                "nfsUid": 0,
                "nfsUserCacheTimeout": 30,
                "nfsUsername": "root",
                "nfsWriteMaxPoolThreads": 128,
                "nfsWriteMinPoolThreads": 32,
                "nfsWriteSizeBits": 20
            },
            "uri": "nfs://10.63.150.213:2049/"
        }
    ]
}
```

5.4 Verification

To verify the installation of NIPAM with Ranger:

```
[root@stl-kserver-08 ~]# hadoop fs -ls nfs://10.63.150.213:2049/
18/12/19 15:53:30 ERROR nfs.NFSv3FileSystem: NfsConnectorV3.0.1 Ranger ACL : execute permission
for listStatus() operation denied for path nfs://10.63.150.213:2049/ for user : root, primary
group : root
18/12/19 15:53:30 INFO NfsRangerAudit: {"repoType":1,"repo":"ntapnipam_hadoop","evtTime":"2018-
12-19
15:53:30.432","access":"EXECUTE","resource":"nfs://10.63.150.213:2049/","resType":"directory","re
sult":0,"policy":0,"reason":"Ranger ACL : execute permission for listStatus() operation denied
for path nfs://10.63.150.213:2049/","enforcer":"ranger-acl","cliIP":"stl-kserver-
08","seq_num":0,"event_count":1,"event_dur_ms":0,"tags":[]}
ls: Ranger ACL : execute permission for listStatus() operation denied for path
nfs://10.63.150.213:2049/ for user : root, primary group : root
[root@stl-kserver-08 ~]#
```

Note: In the output, Ranger does not have policy for the user “root” or a resource path `nfs://`.

6 NFS Validation through HDFS Module

When configuring HDFS with Ranger, NetApp recommends restricting all user, group, and other permissions (read, write, and execute) in the HDFS (native) file system. Ranger then overrides the permissions based on those defined in the Ranger policy for the user or group through Ranger enforcer. Otherwise the request fails back to native file system permissions.

Ranger with NFS is bit different from HDFS because the NFS should have full permissions (read, write, and execute) for user, group, and others. However, you must have a Ranger policy to access the NFS. If you (or any user including root user) do not have permissions in Ranger policy or have no Ranger policy, then NIPAM denies access to the NFS. The following examples show the test results using various permissions.

1. The root user either does not have a Ranger policy or has no permissions in the Ranger policy and tries to do a read operation by listing the NFS:

```
[root@stl-kserver-08 ~]# hadoop fs -ls nfs://10.63.150.213:2049/
18/12/19 15:53:30 ERROR nfs.NFSv3FileSystem: NfsConnectorV3.0.1 Ranger ACL : execute permission
for listStatus() operation denied for path nfs://10.63.150.213:2049/ for user : root, primary
group : root
18/12/19 15:53:30 INFO NfsRangerAudit: {"repoType":1,"repo":"ntapnipam_hadoop","evtTime":"2018-
12-19
15:53:30.432","access":"EXECUTE","resource":"nfs://10.63.150.213:2049/","resType":"directory","re
sult":0,"policy":0,"reason":"Ranger ACL : execute permission for listStatus() operation denied
for path nfs://10.63.150.213:2049/","enforcer":"ranger-acl","cliIP":"stl-kserver-
08","seq_num":0,"event_count":1,"event_dur_ms":0,"tags":[]}
ls: Ranger ACL : execute permission for listStatus() operation denied for path
nfs://10.63.150.213:2049/ for user : root, primary group : root
[root@stl-kserver-08 ~]# id
uid=0(root) gid=0(root) groups=0(root)
```

2. The tester does not have read permissions or a Ranger policy and is trying to list directories in the NFS:

```
[root@stl-kserver-08 ~]# su - tester
Last login: Wed Dec 19 15:49:41 EST 2018 on pts/0
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/
18/12/19 15:53:54 ERROR nfs.NFSv3FileSystem: NfsConnectorV3.0.1 Ranger ACL : execute permission
for listStatus() operation denied for path nfs://10.63.150.213:2049/ for user : tester, primary
group : tester
18/12/19 15:53:54 INFO NfsRangerAudit: {"repoType":1,"repo":"ntapnipam_hadoop","evtTime":"2018-
12-19
15:53:54.188","access":"EXECUTE","resource":"nfs://10.63.150.213:2049/","resType":"directory","re
sult":0,"policy":0,"reason":"Ranger ACL : execute permission for listStatus() operation denied
for path nfs://10.63.150.213:2049/","enforcer":"ranger-acl","cliIP":"stl-kserver-
08","seq_num":0,"event_count":1,"event_dur_ms":0,"tags":[]}
ls: Ranger ACL : execute permission for listStatus() operation denied for path
nfs://10.63.150.213:2049/ for user : tester, primary group : tester
```

3. The tester does not have write permissions and is trying to create a directory in the NFS.

```
[tester@stl-kserver-08 ~]$ hadoop fs -mkdir
nfs://10.63.150.213:2049/nc_ranger1/tester_ranger_nopolicy_foruser_write_test
18/12/19 15:54:05 ERROR nfs.NFSv3FileSystem: NfsConnectorV3.0.1 Ranger ACL : write permission for
mkdirs() operation denied for path nfs://10.63.150.213:2049/nc_ranger1 for user : tester, primary
group : tester
18/12/19 15:54:05 INFO NfsRangerAudit: {"repoType":1,"repo":"ntapnipam_hadoop","evtTime":"2018-
12-19
15:54:05.797","access":"WRITE","resource":"nfs://10.63.150.213:2049/nc_ranger1","resType":"direct
ory","result":0,"policy":0,"reason":"Ranger ACL : write permission for mkdirs() operation denied
for path nfs://10.63.150.213:2049/nc_ranger1","enforcer":"ranger-acl","cliIP":"stl-kserver-
08","seq_num":0,"event_count":1,"event_dur_ms":0,"tags":[]}
mkdir: Ranger ACL : write permission for mkdirs() operation denied for path
nfs://10.63.150.213:2049/nc_ranger1 for user : tester, primary group : tester
[tester@stl-kserver-08 ~]$
```

Best Practices

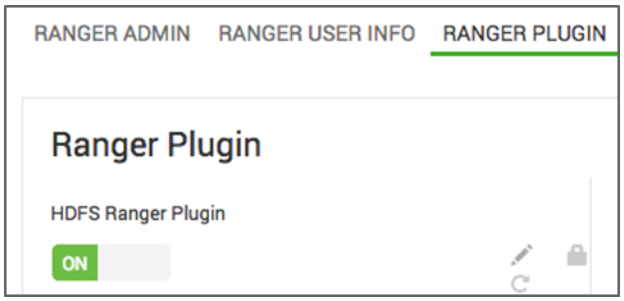
NetApp recommends using Hortonworks Ambari framework to configure Ranger. All the excerpts in this guide are from the Ambari framework and Ranger panel configured in Ambari framework for Hortonworks cluster.

6.1 Configure Ranger for NFS Access

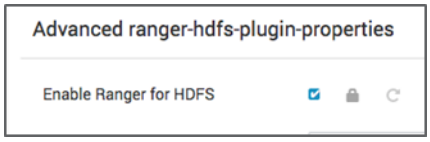
Use the following steps to configure Ranger for NFS:

Note: NIPAM leverages HDFS Ranger plug-in to configure NFSNFS access.

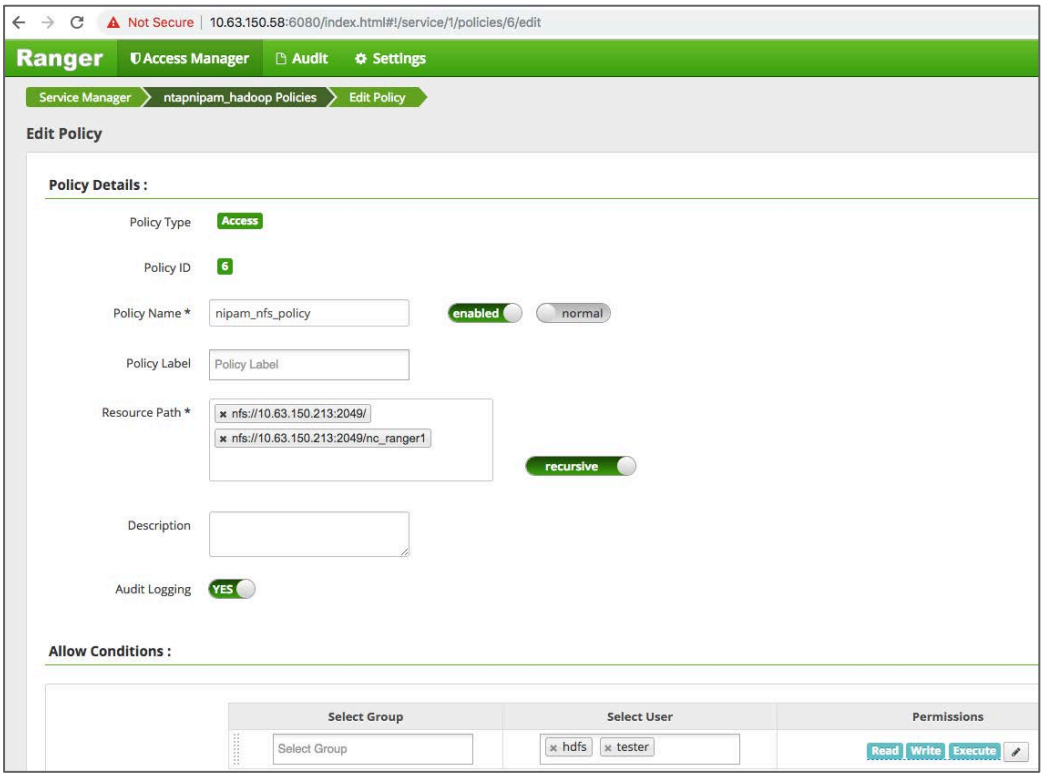
1. In Ranger configuration, enable HDFS plug-in: Ranger > Ranger Plugin > HDFS Ranger Plugin > On > Save.



2. Check the HDFS configuration: HDFS > Configs > Advanced ranger-hdfs-plugin-properties > Enable Ranger for HDFS.

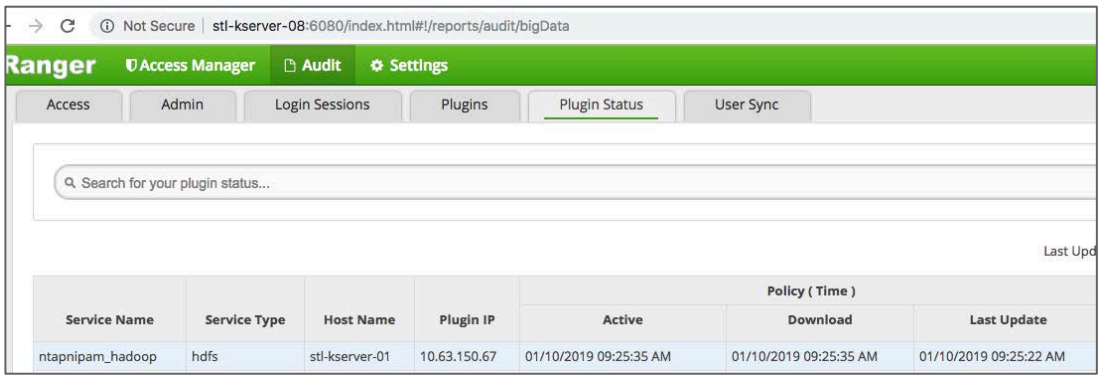


3. Log in to the Ranger portal and add new Ranger policy for NFS: Ranger > Ranger Admin UI > username/password > select HDFS service name link <Hortonworks clustername>_hadoop.
4. Create a Ranger policy, for example `nipam_nfs_policy` with read, write, and execute permissions for the tester.



Note: HDP 2.6.3 allows you to add a Ranger policy for “root” user and group, but this option is not available in HDP 3.0.1.

5. Check that the time updates and check the Active and Download columns to verify that the new or updated policy is activated: Audit > Plugin Status > <clustername>_hadoop.



6.2 Ranger with NFS Validation

1. Check that the tester and the UID is the same across all servers in Hadoop cluster; the same UID is a requirement for NFS.

   Note: You can also use LDAP configuration.

```
[root@stl-kserver-08 ~]# pssh -h /root/hosts -P "id tester"
10.63.150.58: uid=7010(tester) gid=7003(tester) groups=7003(tester)
[1] 09:35:27 [SUCCESS] 10.63.150.58
10.63.150.59: uid=7010(tester) gid=7003(tester) groups=7003(tester)
[2] 09:35:27 [SUCCESS] 10.63.150.59
10.63.150.60: uid=7010(tester) gid=7003(tester) groups=7003(tester)
[3] 09:35:27 [SUCCESS] 10.63.150.60
10.63.150.62: uid=7010(tester) gid=7003(tester) groups=7003(tester)
[4] 09:35:27 [SUCCESS] 10.63.150.62
10.63.150.66: uid=7010(tester) gid=7003(tester) groups=7003(tester)
[5] 09:35:27 [SUCCESS] 10.63.150.66
10.63.150.67: uid=7010(tester) gid=7003(tester) groups=7003(tester)
[6] 09:35:27 [SUCCESS] 10.63.150.67
10.63.150.65: uid=7010(tester) gid=7003(tester) groups=7003(tester)
[7] 09:35:27 [SUCCESS] 10.63.150.65
[root@stl-kserver-08 ~]#
```

2. In Ranger policy, provide the tester with write permissions only and attempt to do both write and read operations such as creating and listing folders.

```
[root@stl-kserver-08 ~]# su - tester
Last login: Wed Dec 19 15:53:50 EST 2018 on pts/1
[tester@stl-kserver-08 ~]$
[tester@stl-kserver-08 ~]$ hadoop fs -mkdir
nfs://10.63.150.213:2049/nc_ranger1/tester_ranger_policy_foruser_write_test
[tester@stl-kserver-08 ~]$
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/
18/12/19 15:58:53 ERROR ranger.NfsRangerAuthorization: NfsConnectorV3.0.1 Access denied : default
policy condition is restrictive in NFS ranger configuration.
18/12/19 15:58:53 ERROR nfs.NFSv3FileSystem: NfsConnectorV3.0.1 Ranger ACL : read permission for
listStatus() operation denied for path nfs://10.63.150.213:2049/ for user : tester, primary group
: tester
18/12/19 15:58:53 INFO NfsRangerAudit: {"repoType":1,"repo":"ntapnipam_hadoop","evtTime":"2018-
12-19
15:58:53.379","access":"READ","resource":"nfs://10.63.150.213:2049/","resType":"directory","resul
t":0,"policy":0,"reason":"Ranger ACL : read permission for listStatus() operation denied for path
nfs://10.63.150.213:2049/","enforcer":"ranger-acl","cliIP":"stl-kserver-
08","seq_num":0,"event_count":1,"event_dur_ms":0,"tags":[]}
ls: Ranger ACL : read permission for listStatus() operation denied for path
nfs://10.63.150.213:2049/ for user : tester, primary group : tester
```

```
[tester@stl-kserver-08 ~]$
[tester@stl-kserver-08 ~]$ hadoop fs -cat
nfs://10.63.150.213:2049/nc_ranger1/filecopytest_woranger_README
18/12/19 15:59:04 ERROR ranger.NfsRangerAuthorization: NfsConnectorV3.0.1 Access denied : default
policy condition is restrictive in NFS ranger configuration.
18/12/19 15:59:04 ERROR nfs.NFSv3FileSystem: NfsConnectorV3.0.1 Ranger ACL : read permission for
open() operation denied for path nfs://10.63.150.213:2049/nc_ranger1/filecopytest_woranger_README
for user : tester, primary group : tester
18/12/19 15:59:04 INFO NfsRangerAudit: {"repoType":1,"repo":"ntapnipam_hadoop","evtTime":"2018-
12-19
15:59:04.273","access":"READ","resource":"nfs://10.63.150.213:2049/nc_ranger1/filecopytest_worang
er_README","resType":"file","result":0,"policy":0,"reason":"Ranger ACL : read permission for
open() operation denied for path
nfs://10.63.150.213:2049/nc_ranger1/filecopytest_woranger_README","enforcer":"ranger-
acl","cliIP":"stl-kserver-08","seq_num":0,"event_count":1,"event_dur_ms":0,"tags":[]}
cat: Ranger ACL : read permission for open() operation denied for path
nfs://10.63.150.213:2049/nc_ranger1/filecopytest_woranger_README for user : tester, primary group
: tester
```

3. In Ranger policy, provide read permission for the tester and attempt to do read operations (ls and cat).

```
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/
Found 6 items
-rw-r--r--   1 tester tester       1145 2018-12-18 07:11
nfs://10.63.150.213:2049/filecopytest_woranger_README
drwxrwxrwx   - tester tester       4096 2018-12-18 07:07
nfs://10.63.150.213:2049/tester_noranger_write_test
drwxrwxrwx   - tester tester       4096 2018-12-19 10:02
nfs://10.63.150.213:2049/tester_ranger_policy_foruser_write_test
drwxrwxrwx   - root   root         4096 2018-12-18 07:05 nfs://10.63.150.213:2049/tg1
drwxrwxrwx   - root   root         4096 2018-12-18 07:06 nfs://10.63.150.213:2049/ts1
drwxrwxrwx   - root   root         4096 2018-12-18 07:06 nfs://10.63.150.213:2049/tv1
[tester@stl-kserver-08 ~]$
[tester@stl-kserver-08 ~]$
[tester@stl-kserver-08 ~]$
[tester@stl-kserver-08 ~]$ hadoop fs -cat
nfs://10.63.150.213:2049/nc_ranger1/filecopytest_woranger_README

                                util-linux

             util-linux is a random collection of Linux utilities

     Note that in years 2006-2010 this project used the name "util-linux-ng".

MAILING LIST:

      E-MAIL: util-linux@vger.kernel.org
      URL:    http://vger.kernel.org/vger-lists.html#util-linux


[tester@stl-kserver-08 ~]$
[tester@stl-kserver-08 ~]$
```

7 YARN Validation

YARN is the architectural center of Hadoop that allows multiple data processing engines such as interactive SQL, real-time streaming, data science, and batch processing to handle data stored in a single platform, unlocking an entirely new approach to analytics. YARN provides resource management and a center platform to deliver consistent operations, security, and data governance tools across Hadoop clusters.

This sections describes how to test user permissions for submitting jobs to YARN queues using Ranger policy.

Before using Ranger with YARN, you must consider the following:

- By default, YARN allows you to submit jobs to the YARN queue when Ranger is enabled for YARN. You must explicitly deny the user permission to submit the job.
- If you allow the users in Ranger plug-in for YARN to submit a read/write job to the NFS, they must also have read/write permission for the NFS used by YARN jobs, otherwise the YARN job fails with the error `read/write denied error for nfs://<folder> for user<> primary group:<>`
- You must provide recursive permission for the NFS, because YARN jobs create subfolders like .temp in the NFS. Without the correct permission, the job fails.

7.1 Sample with Error Configurations in YARN with NFS

1. The job fails for the tester who does not have submit job permission.

```
[tester@stl-kserver-08 ~]$ hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-
examples.jar wordcount nfs://10.63.150.213:2049/filecopytest_woranger_README
nfs://10.63.150.213:2049/wordcount/wc_result_1
18/12/20 19:07:10 INFO client.RMProxy: Connecting to ResourceManager at stl-kserver-
01/10.63.150.67:8050
18/12/20 19:07:10 INFO client.AHSProxy: Connecting to Application History server at stl-kserver-
02/10.63.150.66:10200
18/12/20 19:07:10 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/user/tester/.staging/job_1545348528811_0008
18/12/20 19:07:11 INFO input.FileInputFormat: Total input files to process : 1
18/12/20 19:07:11 INFO mapreduce.JobSubmitter: number of splits:1
18/12/20 19:07:11 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1545348528811_0008
18/12/20 19:07:11 INFO mapreduce.JobSubmitter: Executing with tokens: []
18/12/20 19:07:11 INFO conf.Configuration: found resource resource-types.xml at
file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
18/12/20 19:07:11 INFO mapreduce.JobSubmitter: Cleaning up the staging area
/user/tester/.staging/job_1545348528811_0008
java.io.IOException: org.apache.hadoop.yarn.exceptions.YarnException:
org.apache.hadoop.security.AccessControlException: User tester does not have permission to submit
application_1545348528811_0008 to queue default
at org.apache.hadoop.yarn.ipc.RPCUtil.getRemoteException(RPCUtil.java:38)
at
org.apache.hadoop.yarn.server.resourcemanager.RMAppManager.createAndPopulateNewRMApp(RMAppManager
.java:427)
at
org.apache.hadoop.yarn.server.resourcemanager.RMAppManager.submitApplication(RMAppManager.java:32
0)
at
org.apache.hadoop.yarn.server.resourcemanager.ClientRMService.submitApplication(ClientRMService.j
ava:645)
at
org.apache.hadoop.yarn.api.impl.pb.service.ApplicationClientProtocolPBServiceImpl.submitApplicati
on(ApplicationClientProtocolPBServiceImpl.java:277)
at
org.apache.hadoop.yarn.proto.ApplicationClientProtocol$ApplicationClientProtocolService$2.callBlo
ckingMethod(ApplicationClientProtocol.java:563)
at
org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:524
)
at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:1025)
at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:876)
at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:822)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1730)
at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2682)
Caused by: org.apache.hadoop.security.AccessControlException: User tester does not have
permission to submit application_1545348528811_0008 to queue default
```

2. The example shows the tester with submit job permission in Ranger plugin of YARN , but no read permission on NFS.

```
[tester@stl-kserver-08 ~]$ hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-
examples.jar wordcount nfs://10.63.150.213:2049/filecopytest_woranger_README
nfs://10.63.150.213:2049/wordcount/wc_result_1
```

```
18/12/20 19:08:28 INFO client.RMProxy: Connecting to ResourceManager at stl-kserver-
01/10.63.150.67:8050
18/12/20 19:08:28 INFO client.AHSProxy: Connecting to Application History server at stl-kserver-
02/10.63.150.66:10200
18/12/20 19:08:28 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/user/tester/.staging/job_1545348528811_0009
18/12/20 19:08:29 INFO input.FileInputFormat: Total input files to process : 1
18/12/20 19:08:29 INFO mapreduce.JobSubmitter: number of splits:1
18/12/20 19:08:29 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1545348528811_0009
18/12/20 19:08:29 INFO mapreduce.JobSubmitter: Executing with tokens: []
18/12/20 19:08:29 INFO conf.Configuration: found resource resource-types.xml at
file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
18/12/20 19:08:29 INFO impl.YarnClientImpl: Submitted application application_1545348528811_0009
18/12/20 19:08:29 INFO mapreduce.Job: The url to track the job: http://stl-kserver-
01:8088/proxy/application_1545348528811_0009/
18/12/20 19:08:29 INFO mapreduce.Job: Running job: job_1545348528811_0009
18/12/20 19:08:34 INFO mapreduce.Job: Job job_1545348528811_0009 running in uber mode : false
18/12/20 19:08:34 INFO mapreduce.Job:  map 0% reduce 0%
18/12/20 19:08:38 INFO mapreduce.Job: Task Id : attempt_1545348528811_0009_m_000000_0, Status :
FAILED
Error: org.apache.hadoop.security.AccessControlException: Ranger ACL : read permission for open()
operation denied for path nfs://10.63.150.213:2049/filecopytest_woranger_README for user :
tester, primary group : tester
at org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem.enforceRangerPolicy(NFSv3FileSystem.java:331)
at org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem.open(NFSv3FileSystem.java:432)
at org.apache.hadoop.fs.FileSystem.open(FileSystem.java:950)
at org.apache.hadoop.mapreduce.lib.input.LineRecordReader.initialize(LineRecordReader.java:86)
at org.apache.hadoop.mapred.MapTask$NewTrackingRecordReader.initialize(MapTask.java:560)
at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:798)
at org.apache.hadoop.mapred.MapTask.run(MapTask.java:347)
at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:174)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1730)
at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:168)
```

3. You can check NIPAM logs from the folder which you identified in `nfsRangerAuditLogPath` parameter in `nfs-mapping.json`. In this example `nfsRangerAuditLogPath` value is `/usr/nipam/ranger/logs`.

```
[root@stl-kserver-08 NetApp Ambari Plugin 1.0.1]# catr
/usr/nipam/ranger/logs/ranger_nfs_audit_stl-kserver-08_tester_9610_1545350929553.log
bash: catr: command not found...
Similar command is: 'cat'
[root@stl-kserver-08 NetApp Ambari Plugin 1.0.1]# cat
/usr/nipam/ranger/logs/ranger_nfs_audit_stl-kserver-08_tester_9610_1545350929553.log
{"repoType":1,"repo":"ntapnipam_hadoop","evtTime":"2018-12-20
19:08:50.760","access":"READ","resource":"nfs://10.63.150.213:2049/filecopytest_woranger_README",
"resType":"file","result":0,"policy":0,"reason":"Ranger ACL : read permission for open()
operation denied for path
nfs://10.63.150.213:2049/filecopytest_woranger_README","enforcer":"ranger-acl","cliIP":"stl-
kserver-08","seq_num":0,"event_count":1,"event_dur_ms":0,"tags":[]}
[root@stl-kserver-08 NetApp Ambari Plugin 1.0.1]#
```

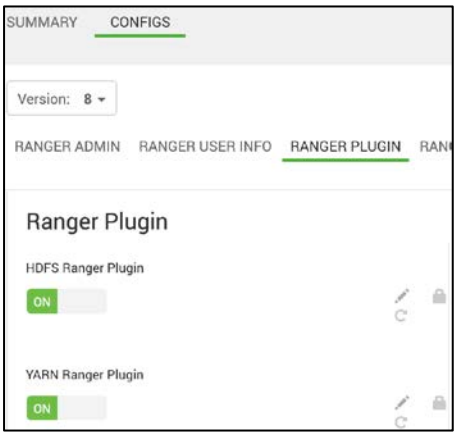
4. The tester has “submit job” permission in the Ranger plug-in of YARN. The tester also has read permission but no write permission to NFS.

```
 [tester@stl-kserver-08 ~]$ hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-
examples.jar wordcount nfs://10.63.150.213:2049/wordcount/README
nfs://10.63.150.213:2049/wordcount/wc_result_3
18/12/20 19:10:51 INFO client.RMProxy: Connecting to ResourceManager at stl-kserver-
01/10.63.150.67:8050
18/12/20 19:10:51 INFO client.AHSProxy: Connecting to Application History server at stl-kserver-
02/10.63.150.66:10200
18/12/20 19:10:52 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/user/tester/.staging/job_1545348528811_0011
18/12/20 19:10:52 INFO input.FileInputFormat: Total input files to process : 1
18/12/20 19:10:52 INFO mapreduce.JobSubmitter: number of splits:1
18/12/20 19:10:52 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1545348528811_0011
18/12/20 19:10:52 INFO mapreduce.JobSubmitter: Executing with tokens: []
```

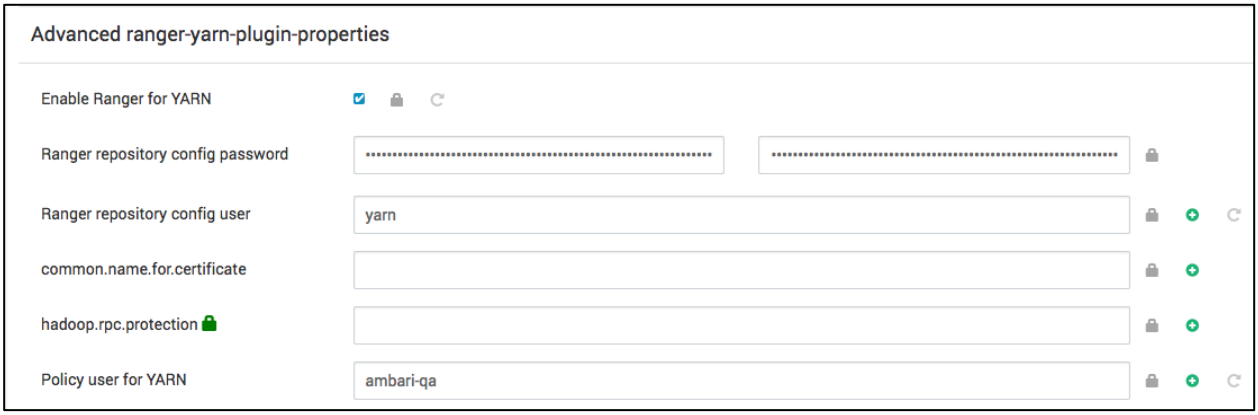
```
18/12/20 19:10:52 INFO conf.Configuration: found resource resource-types.xml at
file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
18/12/20 19:10:52 INFO impl.YarnClientImpl: Submitted application application_1545348528811_0011
18/12/20 19:10:52 INFO mapreduce.Job: The url to track the job: http://stl-kserver-
01:8088/proxy/application_1545348528811_0011/
18/12/20 19:10:52 INFO mapreduce.Job: Running job: job_1545348528811_0011
18/12/20 19:10:57 INFO mapreduce.Job: Job job_1545348528811_0011 running in uber mode : false
18/12/20 19:10:57 INFO mapreduce.Job:  map 0% reduce 0%
18/12/20 19:11:00 INFO mapreduce.Job: Job job_1545348528811_0011 failed with state FAILED due to:
Job setup failed : org.apache.hadoop.security.AccessControlException: Ranger ACL : write
permission for mkdirs() operation denied for path
nfs://10.63.150.213:2049/wordcount/wc_result_3/_temporary for user : tester, primary group :
tester
at org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem.enforceRangerPolicy(NFSv3FileSystem.java:331)
at org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem.mkdirs(NFSv3FileSystem.java:776)
at org.apache.hadoop.fs.FileSystem.mkdirs(FileSystem.java:2326)
at
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter.setupJob(FileOutputCommitter.java:354)
at
org.apache.hadoop.mapreduce.v2.app.commit.CommitterEventHandler$EventProcessor.handleJobSetup(Com
mitterEventHandler.java:255)
at
org.apache.hadoop.mapreduce.v2.app.commit.CommitterEventHandler$EventProcessor.run(CommitterEvent
Handler.java:235)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
```

7.2 Ranger with YARN and NFS Configuration

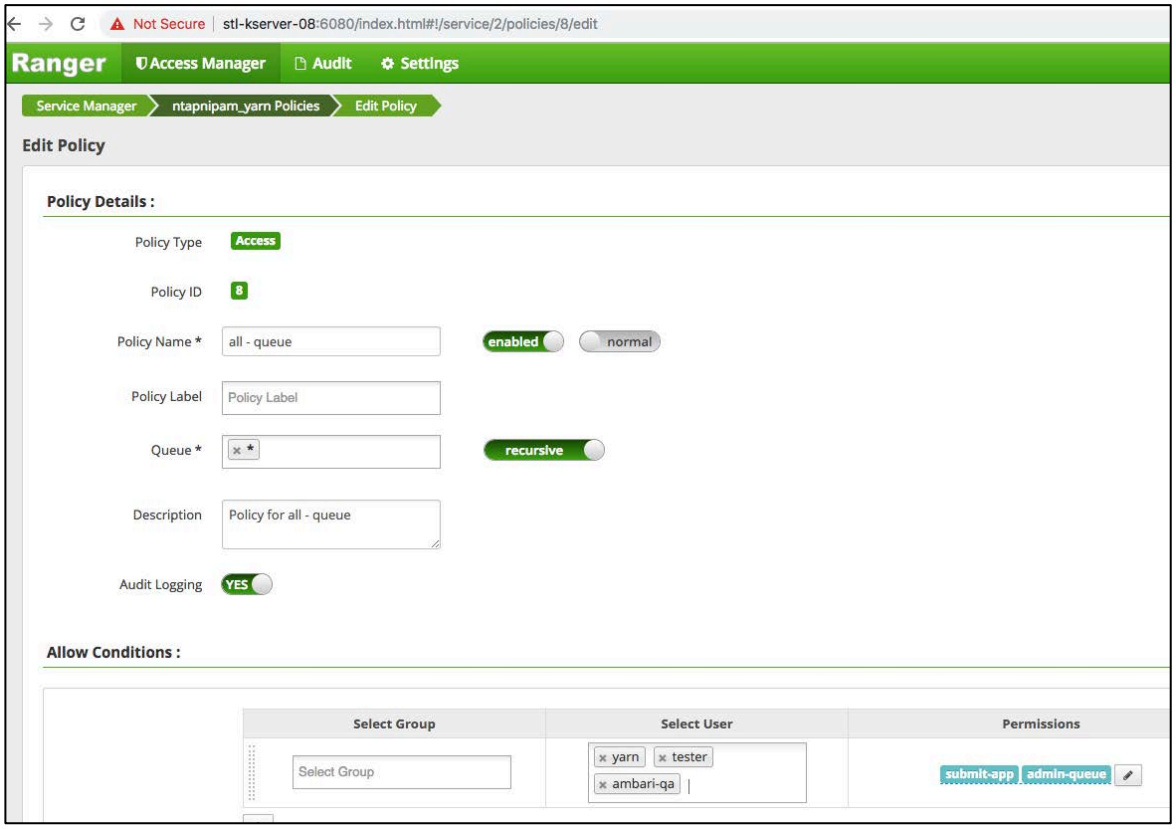
1. Enable Ranger for Yarn: Ranger > Configs > Ranger Plugin > enable Yarn Ranger Plugin.



2. Check that YARN is enabled with Ranger: Yarn > Config > Advanced ranger-yarn-plugin-properties > Enable Ranger for Yarn.



3. Log in to the Ranger portal and add a new Ranger policy for YARN: Ranger > Ranger Admin UI > username/password > select YARN service name link <hortonworks clustername>_yarn > Add new policy.
4. Enter Queue, either * or the name of the YARN queue.



5. Check that the time is the same and updated in the Active and Download columns in Ranger Audit: Ranger > Audit > plugin status > check <clustername>_yarn.

Service Name	Service Type	Host Name	Plugin IP	Policy (Time)		
				Active	Download	Last Update
ntapnipam_yarn	yarn	stl-kserver-01	10.63.150.67	01/11/2019 10:29:49 AM	01/11/2019 10:29:49 AM	01/11/2019 10:29:25 AM

6. Provide NFS access (read and write) in Ranger to run the YARN job. For example, the username is tester and the NFS is nfs://10.63.150.213:2049/wordcount.

```
 [tester@stl-kserver-08 ~]$ hadoop fs -copyFromLocal /usr/share/doc/util-linux-2.23.2/README
nfs://10.63.150.213:2049/wordcount
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/wordcount
Found 3 items
-rw-r--r--   1 tester tester       1145 2018-12-20 13:13
nfs://10.63.150.213:2049/wordcount/README
drwxrwxrwx   - tester tester       4096 2018-12-20 13:08
nfs://10.63.150.213:2049/wordcount/wc_result
drwxrwxrwx   - tester tester       4096 2018-12-20 13:12
nfs://10.63.150.213:2049/wordcount/wc_result_1
[tester@stl-kserver-08 ~]$
```

Best Practices

There is no NFS service name for NFS; we leverage HDFS service in Ranger Portal to configure NFS.

7.3 YARN with NFS Validation

1. The tester has submit permission in Ranger configuration for YARN and also read, write, and execute permission for NFS.

```
[tester@stl-kserver-08 ~]$ hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-
examples.jar wordcount nfs://10.63.150.213:2049/wordcount/README
nfs://10.63.150.213:2049/wordcount/wc_result_3
18/12/20 19:11:26 INFO client.RMProxy: Connecting to ResourceManager at stl-kserver-
01/10.63.150.67:8050
18/12/20 19:11:26 INFO client.AHSProxy: Connecting to Application History server at stl-kserver-
02/10.63.150.66:10200
18/12/20 19:11:26 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/user/tester/.staging/job_1545348528811_0012
18/12/20 19:11:27 INFO input.FileInputFormat: Total input files to process : 1
18/12/20 19:11:27 INFO mapreduce.JobSubmitter: number of splits:1
18/12/20 19:11:27 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1545348528811_0012
18/12/20 19:11:27 INFO mapreduce.JobSubmitter: Executing with tokens: []
18/12/20 19:11:27 INFO conf.Configuration: found resource resource-types.xml at
file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
18/12/20 19:11:27 INFO impl.YarnClientImpl: Submitted application application_1545348528811_0012
18/12/20 19:11:27 INFO mapreduce.Job: The url to track the job: http://stl-kserver-
01:8088/proxy/application_1545348528811_0012/
18/12/20 19:11:27 INFO mapreduce.Job: Running job: job_1545348528811_0012
18/12/20 19:11:32 INFO mapreduce.Job: Job job_1545348528811_0012 running in uber mode : false
18/12/20 19:11:32 INFO mapreduce.Job:  map 0% reduce 0%
18/12/20 19:11:37 INFO mapreduce.Job:  map 100% reduce 0%
18/12/20 19:11:42 INFO mapreduce.Job:  map 100% reduce 100%
18/12/20 19:11:42 INFO mapreduce.Job: Job job_1545348528811_0012 completed successfully
18/12/20 19:11:42 INFO mapreduce.Job: Counters: 58
File System Counters
FILE: Number of bytes read=1339
FILE: Number of bytes written=466651
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=106
HDFS: Number of bytes written=0
HDFS: Number of read operations=1
HDFS: Number of large read operations=0
HDFS: Number of write operations=0
NFS: Number of bytes read=0
NFS: Number of bytes written=1025
```

```
NFS: Number of read operations=0
NFS: Number of large read operations=0
NFS: Number of write operations=0
Job Counters
Launched map tasks=1
Launched reduce tasks=1
Rack-local map tasks=1
Total time spent by all maps in occupied slots (ms)=224329
Total time spent by all reduces in occupied slots (ms)=289810
Total time spent by all map tasks (ms)=3073
Total time spent by all reduce tasks (ms)=1985
Total vcore-milliseconds taken by all map tasks=3073
Total vcore-milliseconds taken by all reduce tasks=1985
Total megabyte-milliseconds taken by all map tasks=229712896
Total megabyte-milliseconds taken by all reduce tasks=296765440
Map-Reduce Framework
Map input records=47
Map output records=83
Map output bytes=1241
Map output materialized bytes=1339
Input split bytes=106
Combine input records=83
Combine output records=77
Reduce input groups=77
Reduce shuffle bytes=1339
Reduce input records=77
Reduce output records=77
Spilled Records=154
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=106
CPU time spent (ms)=2830
Physical memory (bytes) snapshot=2966777856
Virtual memory (bytes) snapshot=197357330432
Total committed heap usage (bytes)=3364880384
Peak Map Physical memory (bytes)=2559164416
Peak Map Virtual memory (bytes)=66161278976
Peak Reduce Physical memory (bytes)=407613440
Peak Reduce Virtual memory (bytes)=131196051456
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=0
File Output Format Counters
Bytes Written=1025
```

7. In Deny Conditions, deny the tester permission to submit job and admin queue and wait until the time is updated in the Active and Download column in Ranger audit > plugin status.

8. Validate the deny condition.

```
[tester@stl-kserver-08 ~]$ hadoop jar /usr/hdp/3.0.1.0-187/hadoop-mapreduce/hadoop-mapreduce-
examples.jar wordcount nfs://10.63.150.213:2049/filecopytest_woranger_README
nfs://10.63.150.213:2049/yarntest_nfs/filecopytest_woranger_README_out2
19/01/11 10:45:42 INFO client.RMProxy: Connecting to ResourceManager at stl-kserver-
01/10.63.150.67:8050
19/01/11 10:45:42 INFO client.AHSProxy: Connecting to Application History server at stl-kserver-
02/10.63.150.66:10200
19/01/11 10:45:42 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/user/tester/.staging/job_1546623102606_0028
19/01/11 10:45:43 INFO input.FileInputFormat: Total input files to process : 1
19/01/11 10:45:43 INFO mapreduce.JobSubmitter: number of splits:1
19/01/11 10:45:43 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1546623102606_0028
19/01/11 10:45:43 INFO mapreduce.JobSubmitter: Executing with tokens: []
19/01/11 10:45:43 INFO conf.Configuration: found resource resource-types.xml at
file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
19/01/11 10:45:43 INFO mapreduce.JobSubmitter: Cleaning up the staging area
/user/tester/.staging/job_1546623102606_0028
java.io.IOException: org.apache.hadoop.yarn.exceptions.YarnException:
org.apache.hadoop.security.AccessControlException: User tester does not have permission to submit
application_1546623102606_0028 to queue default
        at org.apache.hadoop.yarn.ipc.RPCUtil.getRemoteException(RPCUtil.java:38)
        at
org.apache.hadoop.yarn.server.resourcemanager.RMAppManager.createAndPopulateNewRMApp(RMAppManager
.java:427)
        at
org.apache.hadoop.yarn.server.resourcemanager.RMAppManager.submitApplication(RMAppManager.java:32
0)
        at
org.apache.hadoop.yarn.server.resourcemanager.ClientRMService.submitApplication(ClientRMService.j
ava:645)
        at
org.apache.hadoop.yarn.api.impl.pb.service.ApplicationClientProtocolPBServiceImpl.submitApplicati
on(ApplicationClientProtocolPBServiceImpl.java:277)
        at
org.apache.hadoop.yarn.proto.ApplicationClientProtocol$ApplicationClientProtocolService$2.callBlo
ckingMethod(ApplicationClientProtocol.java:563)
        at
org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:524
)
        at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:1025)
        at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:876)
        at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:822)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1730)
        at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2682)
Caused by: org.apache.hadoop.security.AccessControlException: User tester does not have
permission to submit application_1546623102606_0028 to queue default
```

8 HBase Validation

HBase is an open-source, nonrelational, distributed database modeled after Google's Bigtable and is written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS and non-HDFS such as NFS and S3 storage, providing Bigtable-like capabilities for

Hadoop. This means it provides a fault-tolerant way of storing large quantities of sparse data (small amounts of information caught in a large collection of empty or unimportant data), such as finding the 50 largest items in a group of 2 billion records, or finding the nonzero items representing less than 0.1% of a huge collection.

In this section, we focus on HBase validation with Ranger which keeps the HBase data in NFS.

Note: HBase supports only one file system for HBase database and tables. That means you can use either HDFS or NFS or S3 storage which can be configured using the `hbase.rootdir` parameter.

8.1 HBase Prerequisite

1. In this section, we use the NFS and change the `hbase.rootdir` to NFS:

```
<property>
<name>hbase.rootdir</name>
        <value>nfs://10.63.150.213:2049/apps/hbase/data</value>
</property>
```

2. Add the required folder in NFS and provide `hbase` ownership.

```
hadoop fs -mkdir -p nfs://10.63.150.213:2049/apps/hbase/data
hadoop fs -chown hbase:hadoop nfs://10.63.150.213:2049/apps/hbase/data
```

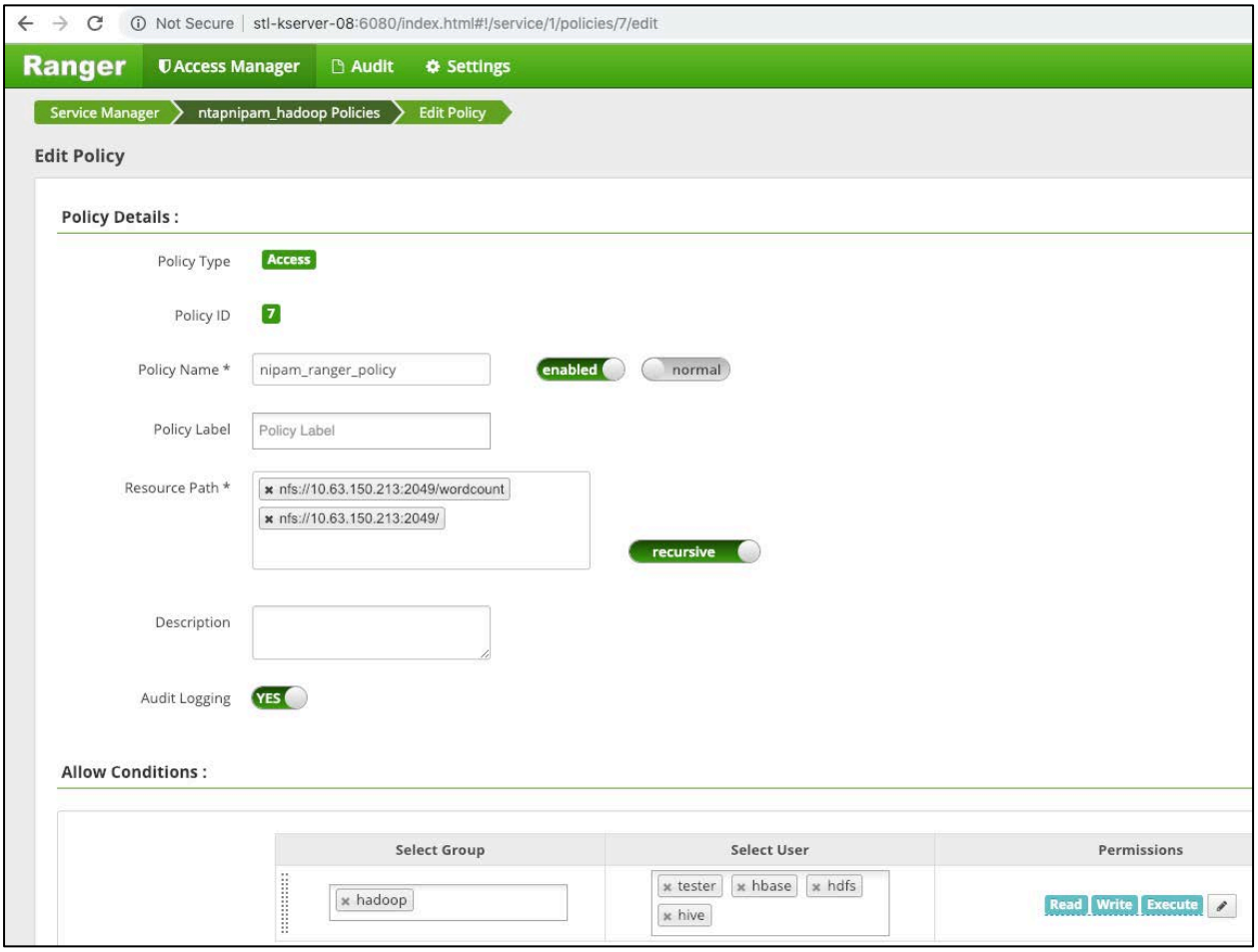
3. After you change the `hbase.rootdir` to `nfs://` and remove the `/hbase-unsecure` folder from zookeeper then restart Zookeeper and HBase in order.

```
[root@stl-kserver-01 ~]# cd /usr/hdp/3.0.1.0-187/zookeeper/bin/
[root@stl-kserver-01 bin]# ./zkCli.sh
Connecting to localhost:2181
…
<output shortened to reduce the size of the page>
…
Welcome to ZooKeeper!
2019-01-08 18:27:10,275 - INFO  [main-SendThread(localhost:2181):ClientCnxn$SendThread@1019] -
Opening socket connection to server localhost/0:0:0:0:0:0:0:1:2181. Will not attempt to
authenticate using SASL (unknown error)
JLine support is enabled
2019-01-08 18:27:10,316 - INFO  [main-SendThread(localhost:2181):ClientCnxn$SendThread@864] -
Socket connection established, initiating session, client: /0:0:0:0:0:0:0:1:58388, server:
localhost/0:0:0:0:0:0:0:1:2181
2019-01-08 18:27:10,321 - INFO  [main-SendThread(localhost:2181):ClientCnxn$SendThread@1279] -
Session establishment complete on server localhost/0:0:0:0:0:0:0:1:2181, sessionid =
0x16810ba89de03b2, negotiated timeout = 30000

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: localhost:2181(CONNECTED) 0] ls /
[hiveserver2-interactive, registry, hiveserver2, hiveserver2-leader, tez-am-unsecure, zookeeper,
infra-solr, hbase-unsecure, llap-unsecure, rmstore, atsv2-hbase-unsecure, ambari-metrics-cluster]
[zk: localhost:2181(CONNECTED) 1] rmr /hbase-unsecure
```

4. Check the HBase master status becomes Active master and is running.

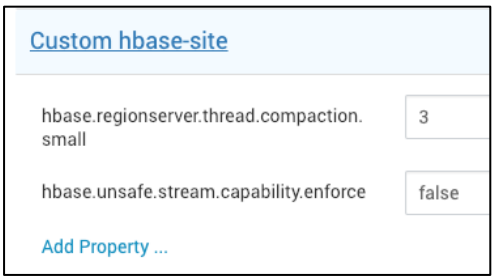
Note: Sometimes it shows standby master. You can fix this after adding the Ranger policy for NFS access.

5. Create or modify the HDFS policy for the HBase user with read, write, and execute permission for `hbase.rootdir` value. In this example `nfs://10.63.150.213:2049/apps/hbase/data` or `nfs://10.63.150.213:2049/` with recursive is enabled because in HBase the files and folders are created by the HBase user even though the HBase operations are done by the end user.



6. Check the Ranger policy update time is the same in the Active and Download columns in Ranger Audit: Ranger portal > Audit > Plugin Status > <hdp cluster name>_hbase and Audit > Plugin Status > <hdp cluster name>_hadoop.
7. If the HBase master status is standby master, add the `hbase.unsafe.stream.capability.enforce=false` parameter to the custom hbase-site.xml. When you restart the HBase service, the HBase master becomes active master.

```
<property>
  <name>hbase.unsafe.stream.capability.enforce</name>
  <value>false</value>
</property>
```



8. Check the zookeeper `/hbase-unsecure` folder has master status and is running.

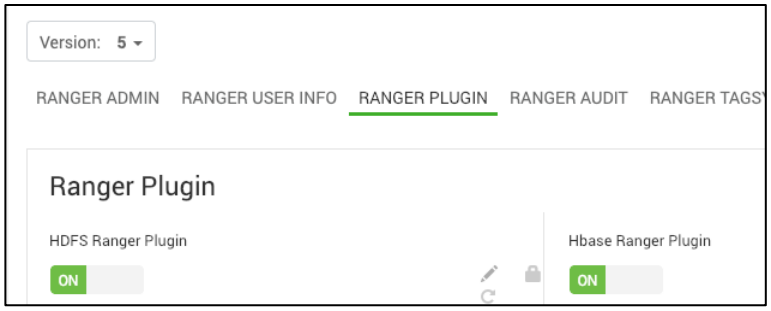
```
[zk: localhost:2181(CONNECTED) 7] ls /hbase-unsecure
[replication, meta-region-server, rs, splitWAL, backup-masters, table-lock, flush-table-proc,
master-maintenance, online-snapshot, switch, master, running, draining, namespace, hbaseid,
table]
[zk: localhost:2181(CONNECTED) 8]
```

9. If the HBase master is still showing as standby master, check the HBase master log file to find the right cause for the issue.
10. Check all folders and files created in the NFS and their permissions.

```
[root@stl-kserver-08 ~]# su - tester
Last login: Mon Jan 14 17:22:45 EST 2019 on pts/0
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/apps/hbase
Found 1 items
drwxrwxrwx   - hbase hdfs       4096 2019-01-11 07:24 nfs://10.63.150.213:2049/apps/hbase/data
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/apps/hbase/data
Found 12 items
drwxrwxrwx   - hbase hadoop       4096 2019-01-02 11:40
nfs://10.63.150.213:2049/apps/hbase/data/.hbck
drwxrwxrwx   - hbase hadoop       4096 2019-01-11 07:24
nfs://10.63.150.213:2049/apps/hbase/data/.tmp
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 04:44
nfs://10.63.150.213:2049/apps/hbase/data/MasterProcWALs
drwxrwxrwx   - hbase hadoop       4096 2019-01-11 07:24
nfs://10.63.150.213:2049/apps/hbase/data/WALs
drwxrwxrwx   - hbase hadoop       4096 2019-01-14 16:03
nfs://10.63.150.213:2049/apps/hbase/data/archive
drwxrwxrwx   - hbase hadoop       4096 2019-01-02 11:40
nfs://10.63.150.213:2049/apps/hbase/data/corrupt
drwxrwxrwx   - hbase hadoop       4096 2019-01-02 12:06
nfs://10.63.150.213:2049/apps/hbase/data/data
-rw-r--r--   1 hbase hadoop         42 2019-01-02 11:40
nfs://10.63.150.213:2049/apps/hbase/data/hbase.id
-rw-r--r--   1 hbase hadoop          7 2019-01-02 11:40
nfs://10.63.150.213:2049/apps/hbase/data/hbase.version
drwxrwxrwx   - hbase hadoop       4096 2019-01-02 11:40
nfs://10.63.150.213:2049/apps/hbase/data/mobdir
drwxrwxrwx   - hbase hadoop      16384 2019-01-15 05:33
nfs://10.63.150.213:2049/apps/hbase/data/oldWALs
drwx--x--x   - hbase hadoop       4096 2019-01-02 11:40
nfs://10.63.150.213:2049/apps/hbase/data/staging
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/apps/hbase/data/data
Found 2 items
drwxrwxrwx   - hbase hadoop       4096 2019-01-02 12:33
nfs://10.63.150.213:2049/apps/hbase/data/data/default
drwxrwxrwx   - hbase hadoop       4096 2019-01-02 12:06
nfs://10.63.150.213:2049/apps/hbase/data/data/hbase
[tester@stl-kserver-08 ~]$
```

8.2 HBase with NFS Validation

1. Enable Ranger for HBase: Ranger > Configs > Ranger Plugin > enable HBase Ranger plugin.

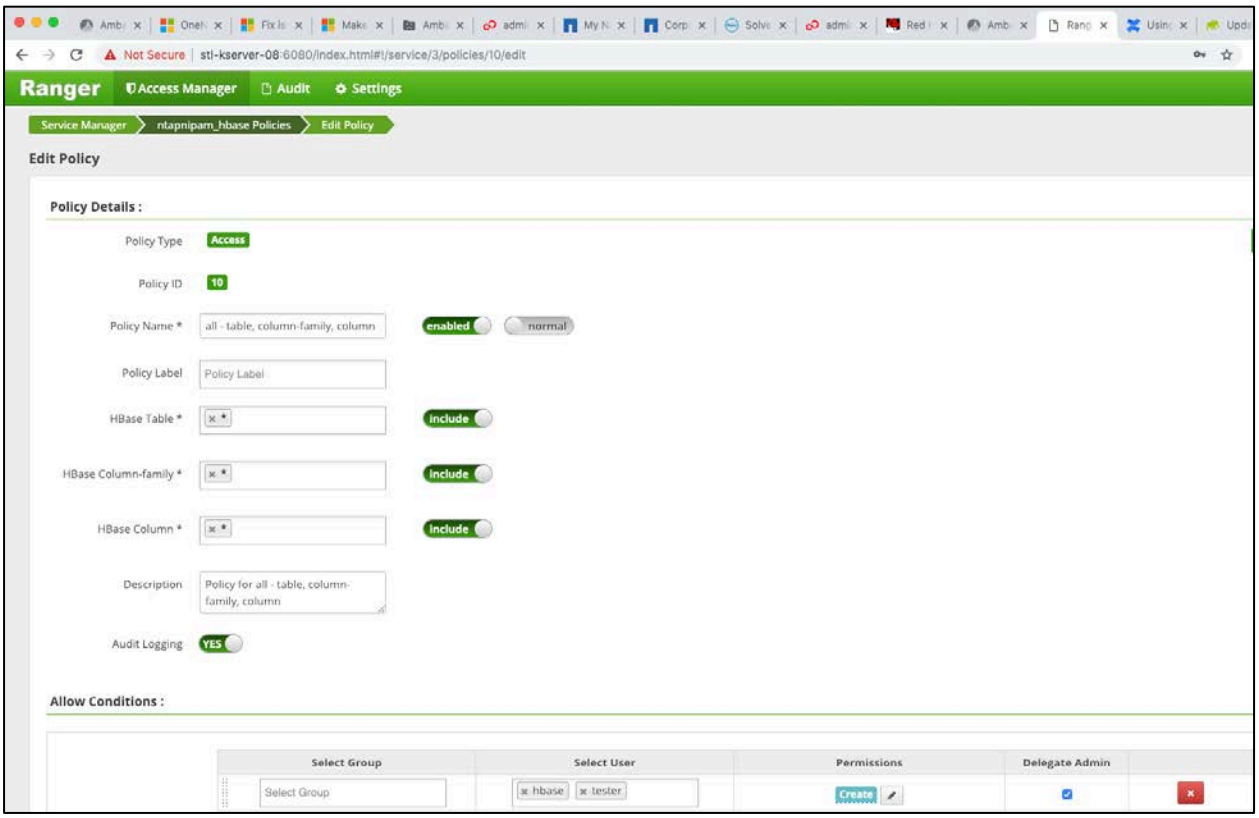


2. Check the Ranger checkbox enabled in HBase configuration: HBase > configs > Advanced ranger-hbase-plugin-properties.

3. Check the tester in HBase and create a table where the user does not have a policy or permissions in Ranger for HBase.

```
[tester@stl-kserver-08 ~]$ hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 2.0.0.3.0.1.0-187, re9fcf450949102de5069b257a6dee469b8f5aab3, Wed Sep 19 10:16:35 UTC
2018
Took 0.0011 seconds
hbase(main):001:0> create 'salesnfsnopolicynew','cf'

ERROR: org.apache.hadoop.hbase.security.AccessDeniedException: Insufficient permissions for user
'tester' (action=create)
…
…
```

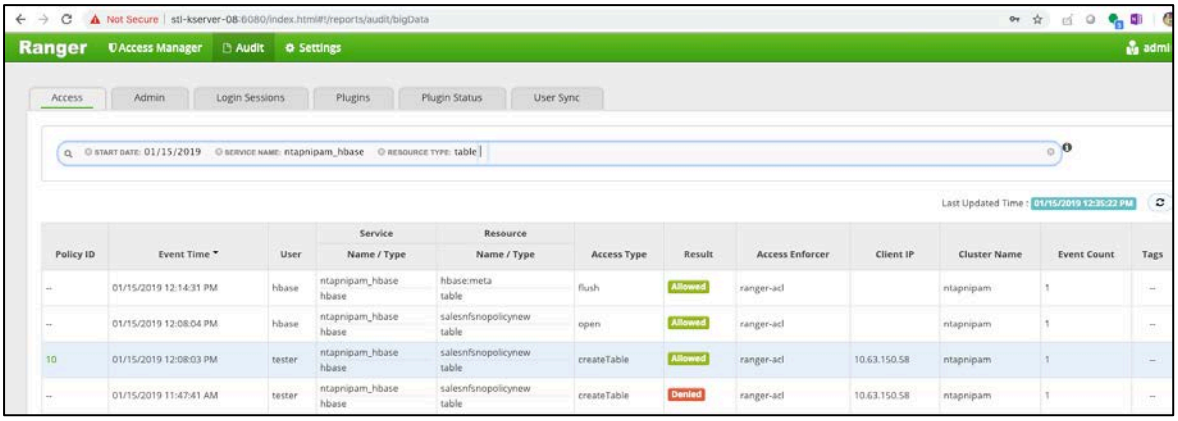
4. Create a Ranger policy for HBase with create permission for the tester.



5. Wait until the time is updated in the Active and Download columns in Ranger audit and then create the HBase table.

```
[tester@stl-kserver-08 ~]$ hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 2.0.0.3.0.1.0-187, re9fcf450949102de5069b257a6dee469b8f5aab3, Wed Sep 19 10:16:35 UTC
2018
Took 0.0012 seconds
hbase(main):001:0> create 'salesnfsnopolicynew','cf'
Created table salesnfsnopolicynew
Took 1.5496 seconds
=> Hbase::Table - salesnfsnopolicynew
hbase(main):002:0>
```

6. Check the access details in Ranger Audit.



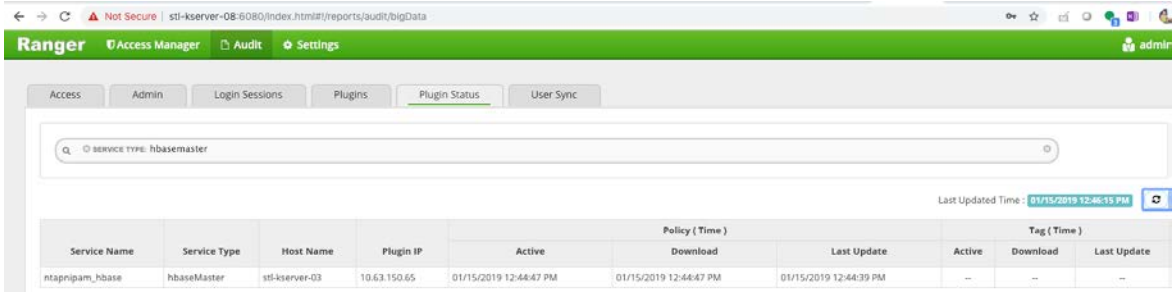
7. Check the folder and files in NFS for the newly created table.

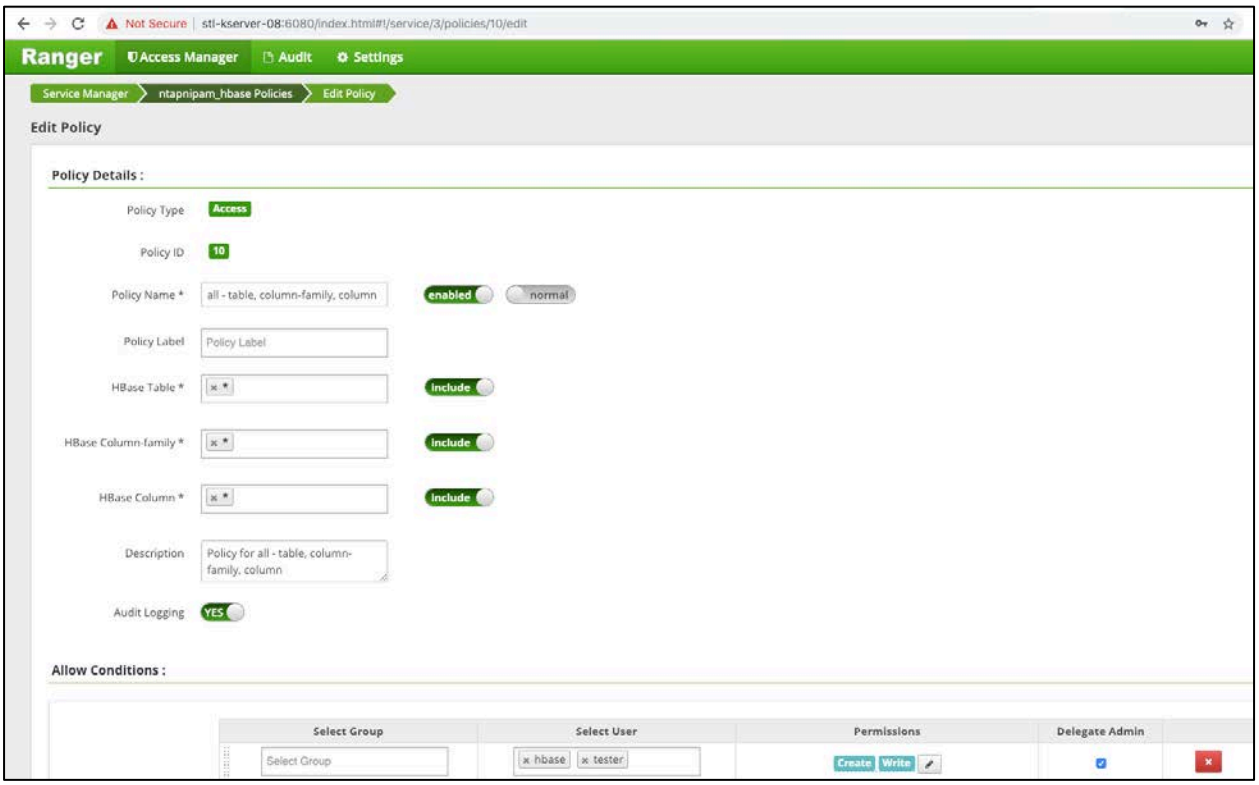
```
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/apps/hbase/data/data
Found 2 items
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default
drwxrwxrwx   - hbase hadoop       4096 2019-01-02 12:06
nfs://10.63.150.213:2049/apps/hbase/data/data/hbase
[tester@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/apps/hbase/data/data/default
Found 2 items
drwxrwxrwx   - hbase hadoop       4096 2019-01-02 12:33
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicy
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew
[tester@stl-kserver-08 ~]$ hadoop fs -ls
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew
Found 3 items
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/.tabledesc
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/.tmp
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/f2dd72bd681fd480dedce36
fc1c208d6
[tester@stl-kserver-08 ~]$
```

8. The tester does not have write permission and is trying to do a write operation in the new table.

```
[tester@stl-kserver-08 ~]$ hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 2.0.0.3.0.1.0-187, re9fcf450949102de5069b257a6dee469b8f5aab3, Wed Sep 19 10:16:35 UTC
2018
Took 0.0011 seconds
hbase(main):001:0> list
TABLE
salesnfsnopolicy
salesnfsnopolicynew
2 row(s)
Took 0.3095 seconds
=> ["salesnfsnopolicy", "salesnfsnopolicynew"]
hbase(main):002:0> put 'salesnfsnopolicynew','row1','cf:col1','value1'

ERROR: org.apache.hadoop.hbase.security.AccessDeniedException: Insufficient permissions for user
'tester',action: put, tableName:salesnfsnopolicynew, family:cf, column: col1
…
…
```

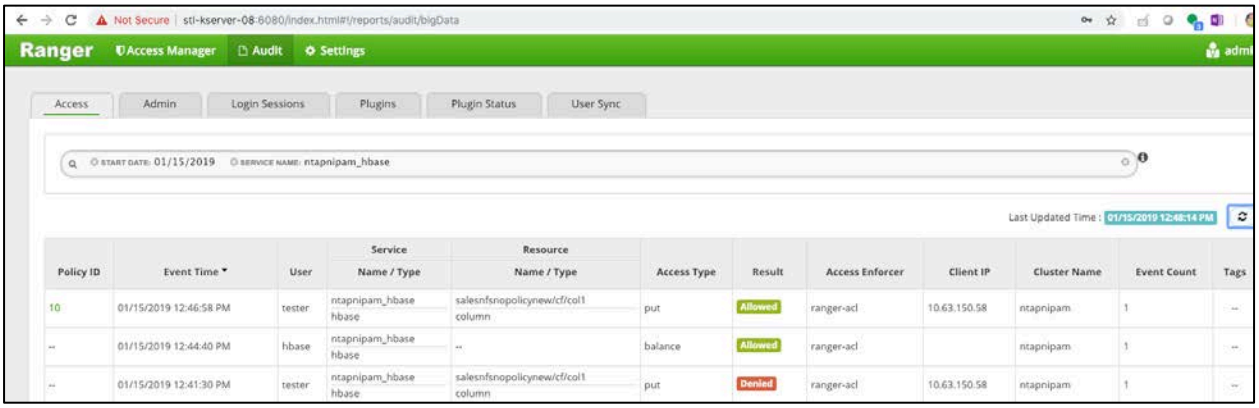
9. Provide write permission to the tester and check the audit log for the updated time in the Active and Download column then try to write into the table again.






```
hbase(main):003:0> put 'salesnfsnopolicynew','row1','cf:col1','value1'
Took 0.0723 seconds
hbase(main):004:0>
```

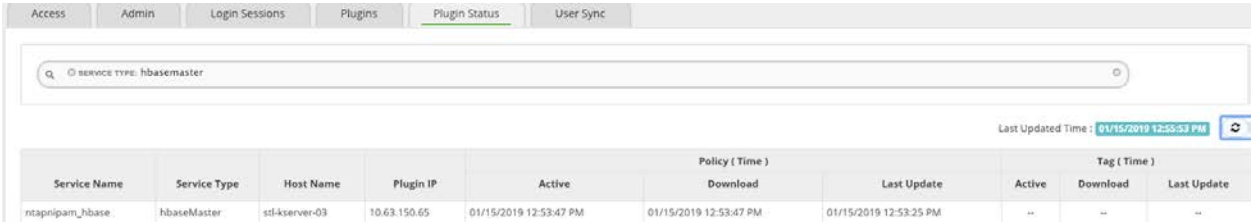
10. Check the access details in the audit log.



11. So far, the tester has create and write permissions and tries to read the content from table.

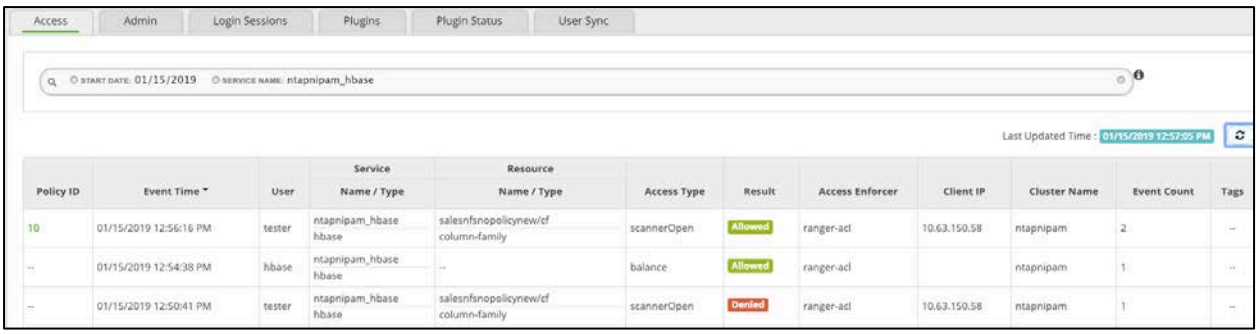
```
hbase(main):004:0> scan 'salesnfsnopolicynew'
ROW                                                    COLUMN+CELL

ERROR: org.apache.hadoop.hbase.security.AccessDeniedException: Insufficient permissions for user
‘tester',action: scannerOpen, tableName:salesnfsnopolicynew, family:cf.
```

12. Provide read permission to the tester, check the audit log for the updated time in Active and Download column and read from the table again.




```
hbase(main):005:0> scan 'salesnfsnopolicynew'
ROW                                                   COLUMN+CELL
 row1                                                 column=cf:col1,
timestamp=1547574418110, value=value1
1 row(s)
Took 0.0520 seconds
hbase(main):006:0>
```

13. Check the access detail in Ranger.



14. Check the NFS permissions and new files created. You must provide the NFS permission for the HBase user because all files and folders are created by the HBase user.

```
[tester@stl-kserver-08 ~]$ hadoop fs -ls
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew
Found 3 items
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/.tabledesc
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/.tmp
```

```
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/f2dd72bd681fd480dedce36
fc1c208d6
[tester@stl-kserver-08 ~]$ hadoop fs -ls
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/f2dd72bd681fd480dedce36
fc1c208d6
Found 3 items
-rw-rw-rw-   1 hbase hadoop         54 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/f2dd72bd681fd480dedce36
fc1c208d6/.regioninfo
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/f2dd72bd681fd480dedce36
fc1c208d6/cf
drwxrwxrwx   - hbase hadoop       4096 2019-01-15 06:07
nfs://10.63.150.213:2049/apps/hbase/data/data/default/salesnfsnopolicynew/f2dd72bd681fd480dedce36
fc1c208d6/recovered.edits
[tester@stl-kserver-08 ~]$
```

9 Hive Validation

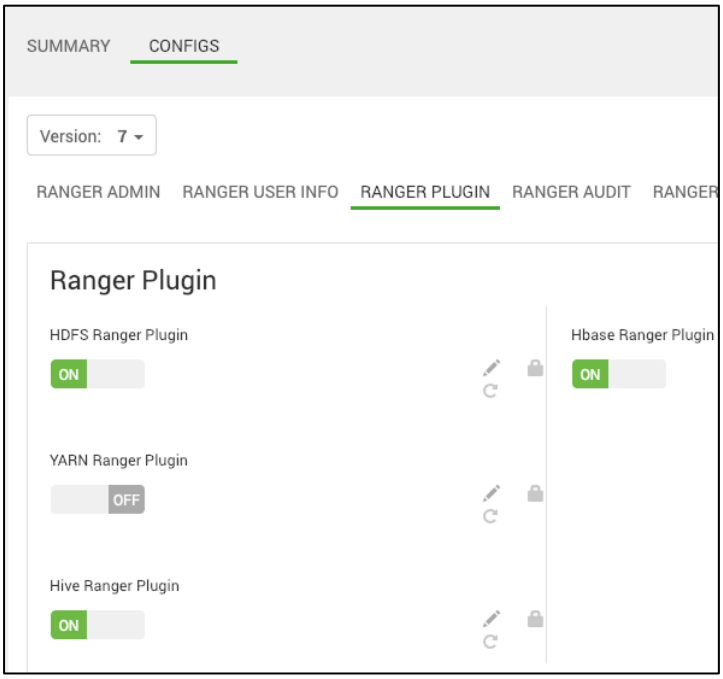
Hive is similar to an SQL query, and is often used for Apache Hadoop data warehouses. This section provides details about NIPAM with Hive using MapReduce and the Tez execution engine. The MapReduce engine is not supported from HDP 3.0.1. This section describes Ranger configuration for Hive with NFS storage using NIPAM.

In this section, we focus on Hive with Ranger configuration to control access to Hive.

Note: You can can use Hive with NFS for external tables; the eligible operations are use, create, select, and drop commands.

9.1 Ranger Configuration for Hive

1. Enable Hive ranger plugin: Ranger > Ranger plugin > enable Hive Ranger Plugin

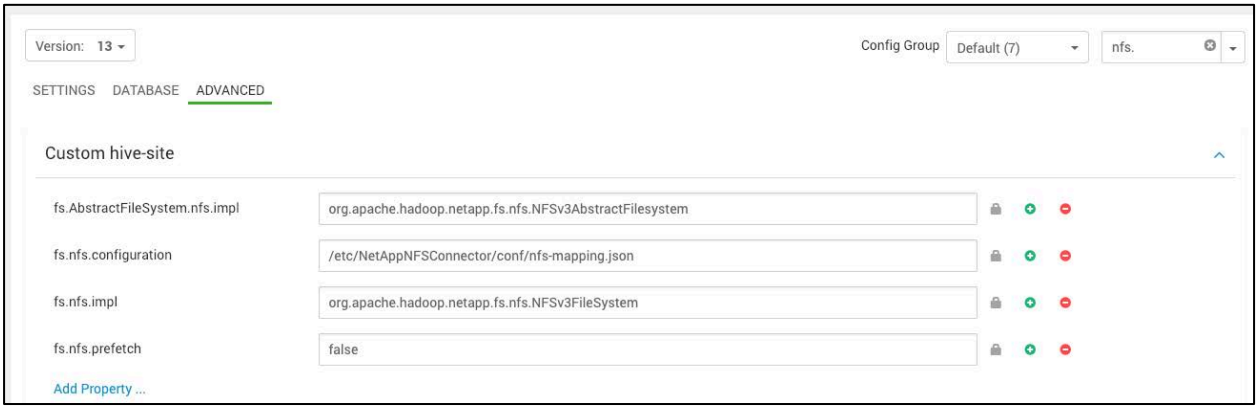


2. Add NIPAM configuration parameters in `hive-site.xml`.

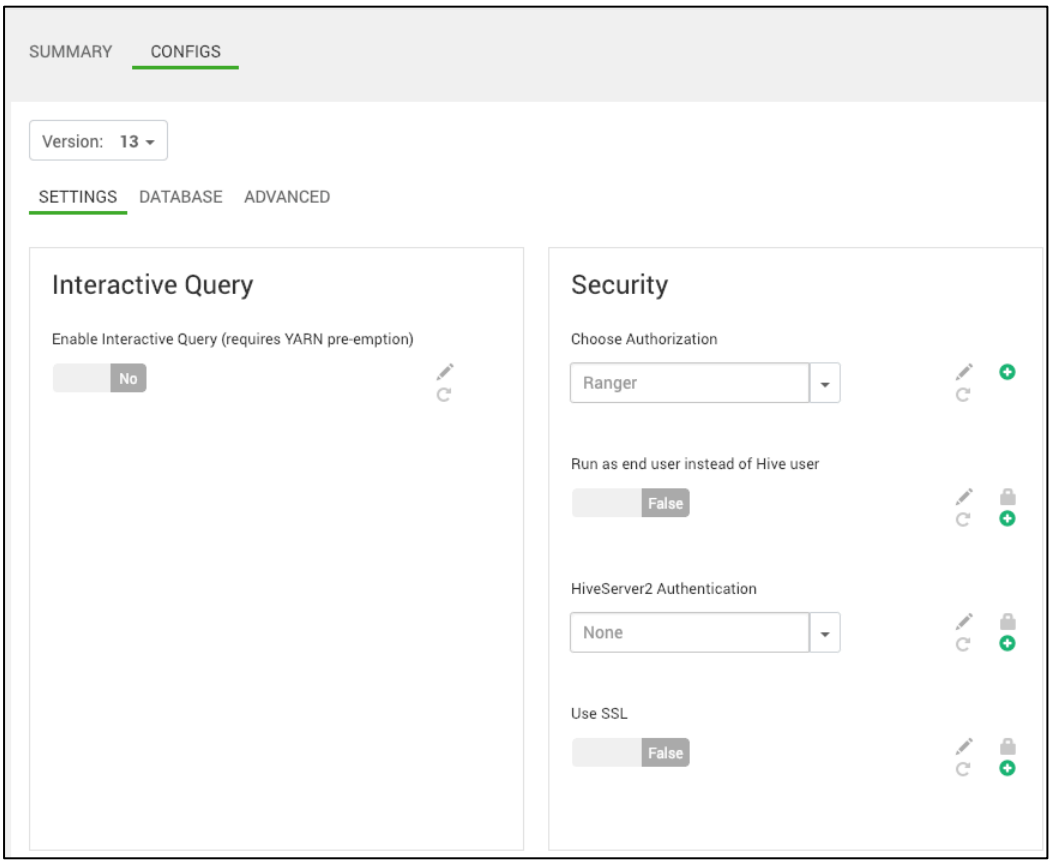
```
  <property>
     <name>fs.AbstractFileSystem.nfs.impl</name>
```

```
  <value>org.apache.hadoop.netapp.fs.nfs.NFSv3AbstractFilesystem</value>
</property>

<property>
  <name>fs.nfs.configuration</name>
  <value>/etc/NetAppNFSConnector/conf/nfs-mapping.json</value>
</property>

<property>
  <name>fs.nfs.impl</name>
  <value>org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem</value>
</property>

<property>
  <name>fs.nfs.prefetch</name>
  <value>false</value>
</property>
```

3. Check NIPAM configuration details from Ambari framework.

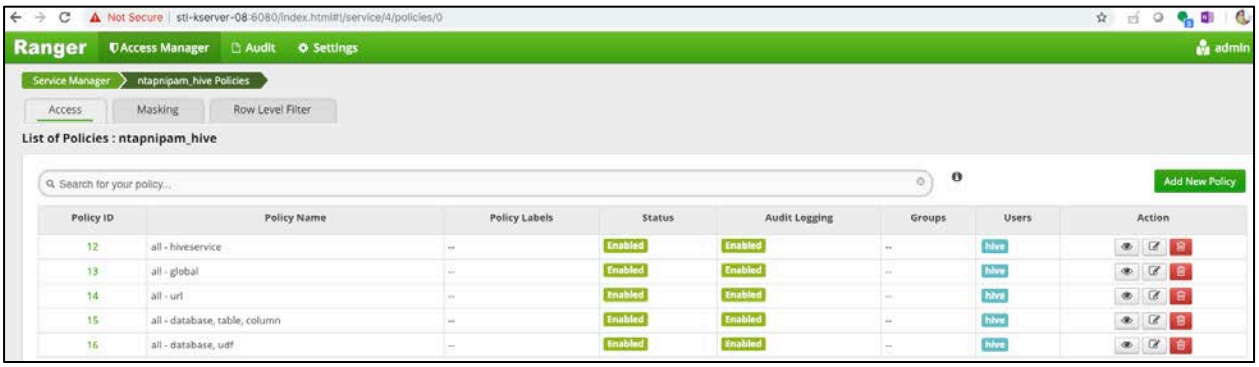


4. The Hive security settings automatically change in Ambari framework.

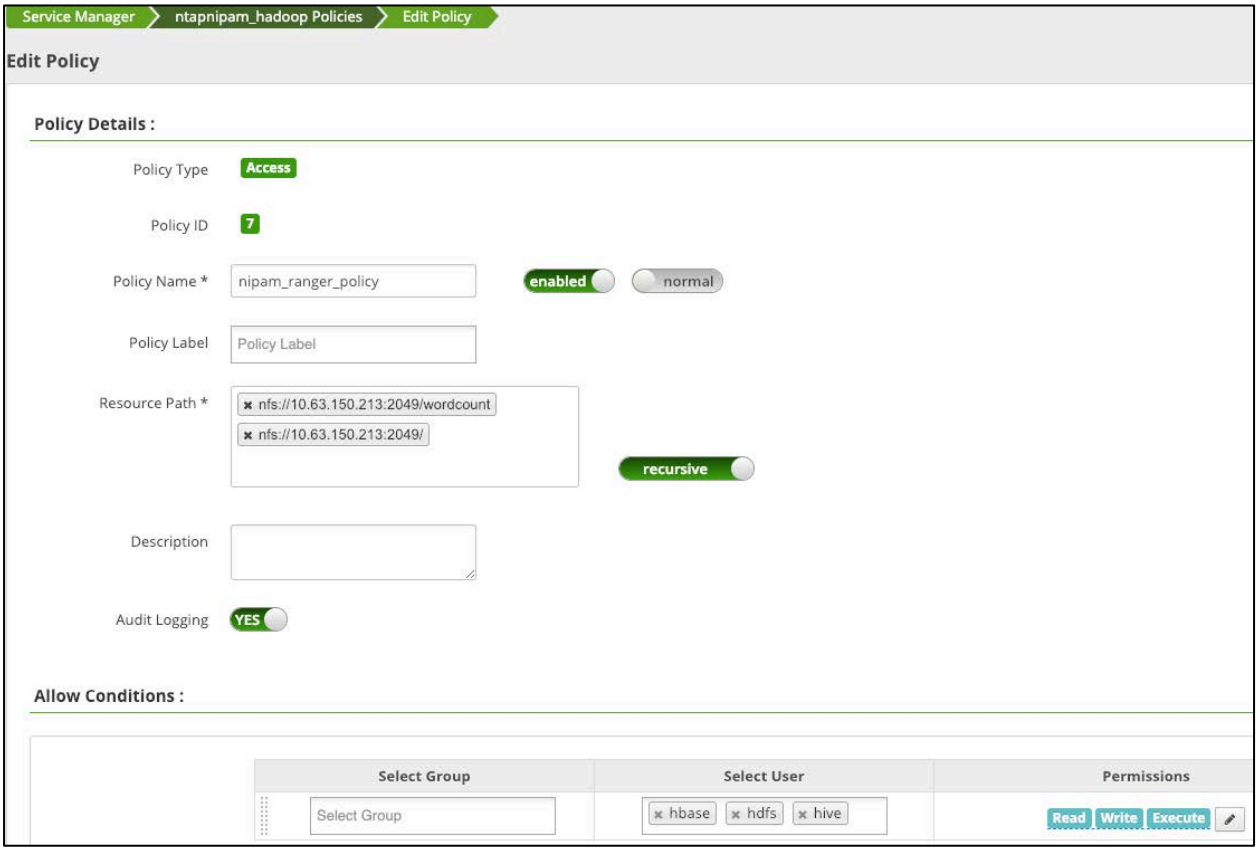


Note: "Run as end user instead of Hive user" automatically changes to false which means that Hive operations are run as Hive user not as an end user who runs the Hive commands. The authentication changed from none to Ranger.

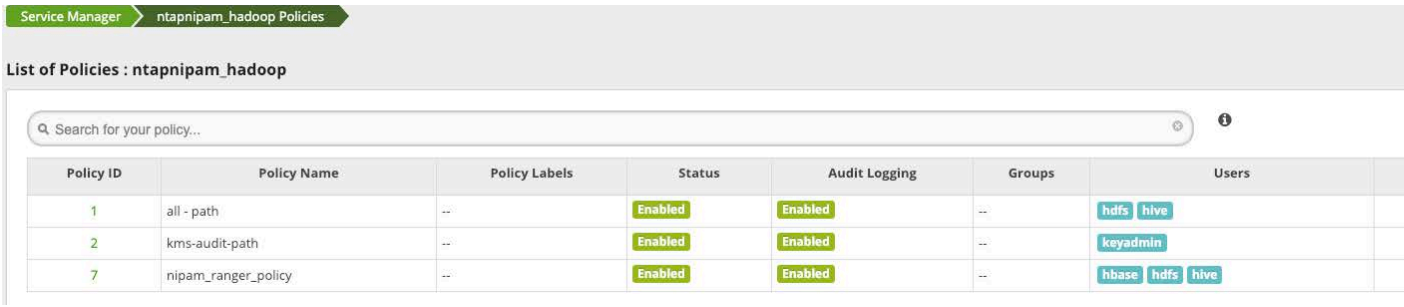
5. The new Hive repository <hdpclustername_hive> for Hives is created automatically in Hive service at the Ranger portal. Full permissions are applied to the Hive user for Hive service, global, URL, database/table/column, and database with UDF. Each of them has a separate policy.



6. Check the Hive user has read, write, and execute permissions for the NFS in Ranger HDFS policy where Hive ecosystem creates the database and table files and folders.



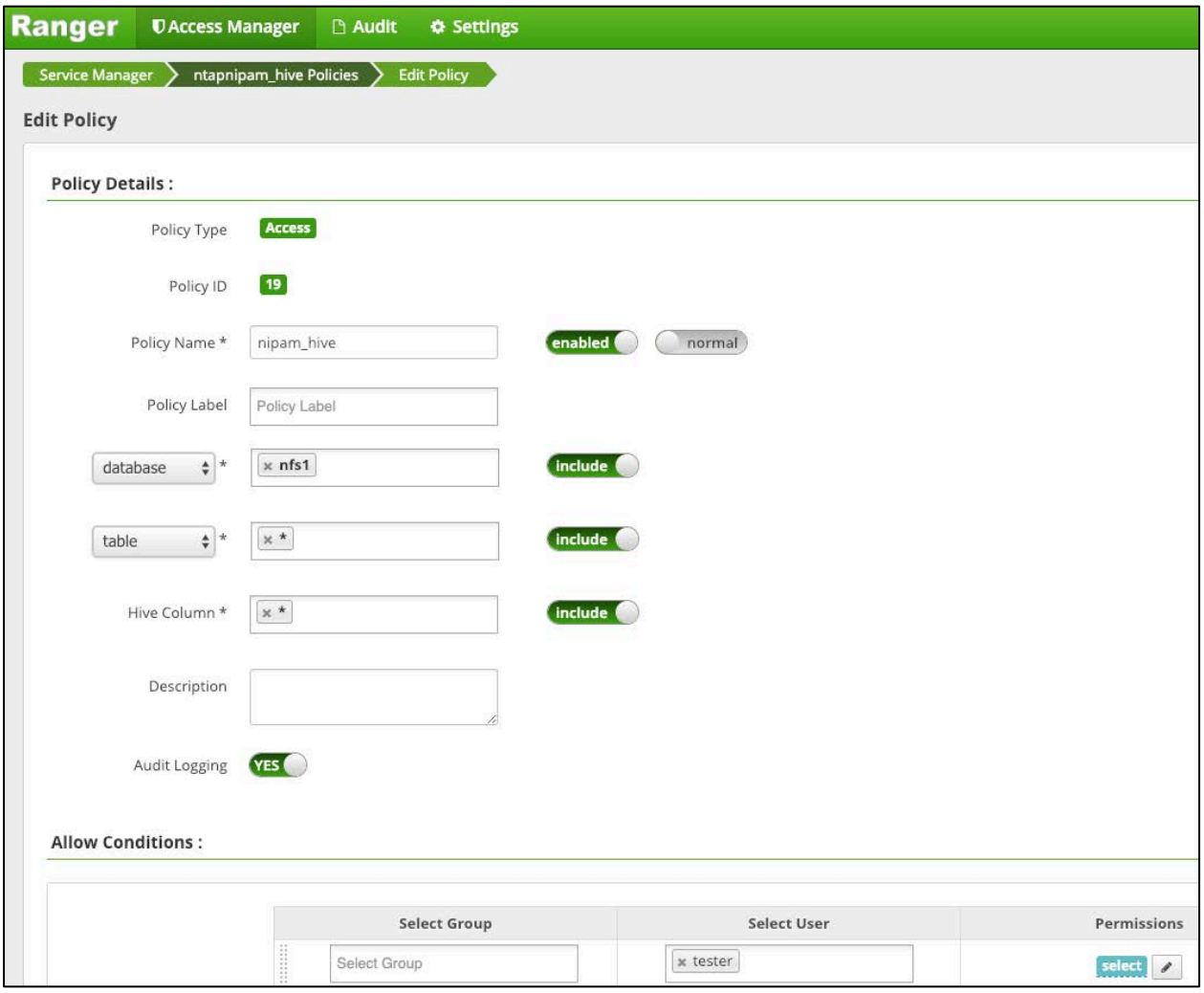
7. Until now, the tester does not have Hive permissions in Ranger Hive policy or NFS permissions in Ranger HDFS policy.

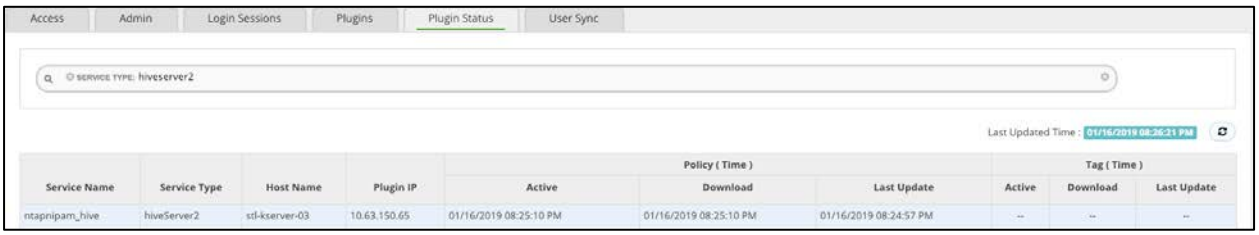


8. By default, Hive ecosystem denies the access to end user (the tester in this example) for Hive operations when Ranger is enabled for Hive. The tester does not have any Hive policy in Hive service to perform Hive operations such as show tables, show databases, use <db>, and so on.

```
[root@stl-kserver-08 tez]# su - tester
Last login: Wed Jan 16 17:52:31 EST 2019 on pts/0
[tester@stl-kserver-08 ~]$ beeline
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hive/lib/log4j-slf4j-impl-
2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hadoop/lib/slf4j-log4j12-
1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://stl-kserver-01:2181,stl-kserver-02:2181,stl-kserver-
03:2181/default;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2
Enter username for jdbc:hive2://stl-kserver-01:2181,stl-kserver-02:2181,stl-kserver-
03:2181/default: tester
Enter password for jdbc:hive2://stl-kserver-01:2181,stl-kserver-02:2181,stl-kserver-
03:2181/default: ******
19/01/16 20:16:19 [main]: INFO jdbc.HiveConnection: Connected to stl-kserver-03:10000
Connected to: Apache Hive (version 3.1.0.3.0.1.0-187)
Driver: Hive JDBC (version 3.1.0.3.0.1.0-187)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.0.3.0.1.0-187 by Apache Hive
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> show tables;
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied:
user [tester] does not have [USE] privilege on [default] (state=42000,code=40000)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> show databases;
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied:
user [tester] does not have [USE] privilege on [Unknown resource!!] (state=42000,code=40000)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> use nfs1;
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied:
user [tester] does not have [USE] privilege on [nfs1] (state=42000,code=40000)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

9. Ranger admin creates a Hive Ranger policy for the tester and provides select privilege for database nfs1. You can see when the policy is activated by checking the update time is same in the Active and Download column in Ranger policy audit.





10. You can execute the show table/database and use database command in Hive shell:

    Note: You can only view the nfs1 database and its tables.

```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> show tables;
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied:
user [tester] does not have [USE] privilege on [default] (state=42000,code=40000)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> show databases;
INFO  : Compiling command(queryId=hive_20190116203101_b0fb126b-019d-40d6-889d-e1fa9c368044): show
databases
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:database_name, type:string,
comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20190116203101_b0fb126b-019d-40d6-889d-
e1fa9c368044); Time taken: 0.014 seconds
```

```
INFO  : Executing command(queryId=hive_20190116203101_b0fb126b-019d-40d6-889d-e1fa9c368044): show
databases
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20190116203101_b0fb126b-019d-40d6-889d-
e1fa9c368044); Time taken: 0.004 seconds
INFO  : OK
+----------------+
| database_name  |
+----------------+
| nfs1           |
+----------------+
1 row selected (0.114 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> use nfs1;
INFO  : Compiling command(queryId=hive_20190116203108_fbb66c91-cab6-4fc8-aa06-b3dc5d32c5b3): use
nfs1
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20190116203108_fbb66c91-cab6-4fc8-aa06-
b3dc5d32c5b3); Time taken: 0.017 seconds
INFO  : Executing command(queryId=hive_20190116203108_fbb66c91-cab6-4fc8-aa06-b3dc5d32c5b3): use
nfs1
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20190116203108_fbb66c91-cab6-4fc8-aa06-
b3dc5d32c5b3); Time taken: 0.006 seconds
INFO  : OK
No rows affected (0.031 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> show tables;
INFO  : Compiling command(queryId=hive_20190116203113_5e5fce9e-9b8e-47df-b3f6-b5c16b040a82): show
tables
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string,
comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20190116203113_5e5fce9e-9b8e-47df-b3f6-
b5c16b040a82); Time taken: 0.018 seconds
INFO  : Executing command(queryId=hive_20190116203113_5e5fce9e-9b8e-47df-b3f6-b5c16b040a82): show
tables
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20190116203113_5e5fce9e-9b8e-47df-b3f6-
b5c16b040a82); Time taken: 0.009 seconds
INFO  : OK
+--------------------------------+
|            tab_name            |
+--------------------------------+
| netapphive_external_nfs_table2 |
| netapphive_external_nfs_table3 |
| netapphive_external_nfs_table4 |
| netapphive_external_nfs_table5 |
| netapphive_internal_nfs_table1 |
+--------------------------------+
5 rows selected (0.038 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

11. Show the description of the external table (look at location value) and show the text file used for the external table:

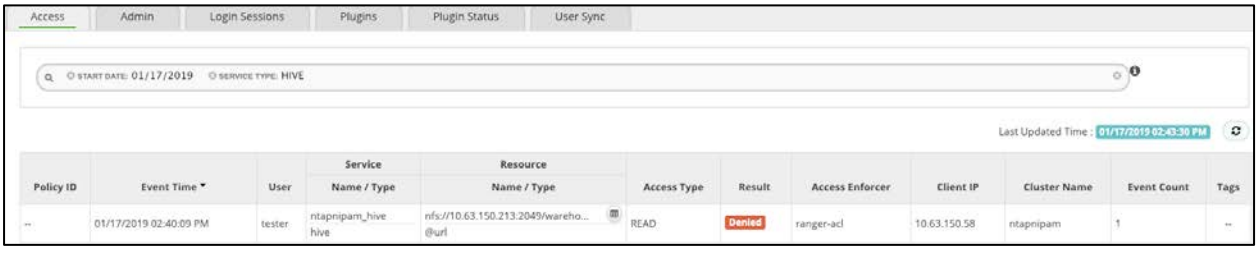
```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> desc formatted netapphive_external_nfs_table2;
INFO  : Compiling command(queryId=hive_20190116204022_017e6515-b4eb-4f47-ad83-ba96e1abf9b6): desc
formatted netapphive_external_nfs_table2
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:col_name, type:string,
comment:from deserializer), FieldSchema(name:data_type, type:string, comment:from deserializer),
FieldSchema(name:comment, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20190116204022_017e6515-b4eb-4f47-ad83-
ba96e1abf9b6); Time taken: 0.074 seconds
INFO  : Executing command(queryId=hive_20190116204022_017e6515-b4eb-4f47-ad83-ba96e1abf9b6): desc
formatted netapphive_external_nfs_table2
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20190116204022_017e6515-b4eb-4f47-ad83-
ba96e1abf9b6); Time taken: 0.022 seconds
INFO  : OK
```

```
+-------------------------------+----------------------------------------------------+-----------
------------+
|           col_name            |                     data_type                      |
comment        |
+-------------------------------+----------------------------------------------------+-----------
------------+
| # col_name                    | data_type                                          | comment
|
| id                            | int                                                |
|
| name                          | string                                             |
|
|                               | NULL                                               | NULL
|
| # Detailed Table Information  | NULL                                               | NULL
|
| Database:                     | nfs1                                               | NULL
|
| OwnerType:                    | USER                                               | NULL
|
| Owner:                        | hive                                               | NULL
|
| CreateTime:                   | Thu Jan 10 17:50:14 EST 2019                       | NULL
|
| LastAccessTime:               | UNKNOWN                                            | NULL
|
| Retention:                    | 0                                                  | NULL
|
| Location:                     | nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive |
NULL                  |
| Table Type:                   | EXTERNAL_TABLE                                     | NULL
|
| Table Parameters:             | NULL                                               | NULL
|
|                               | EXTERNAL                                           | TRUE
|
|                               | bucketing_version                                  | 2
|
|                               | numFiles                                           | 1
|
|                               | totalSize                                          | 2029
|
|                               | transient_lastDdlTime                              | 1547160614
|
|                               | NULL                                               | NULL
|
| # Storage Information         | NULL                                               | NULL
|
| SerDe Library:                | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe | NULL
|
| InputFormat:                  | org.apache.hadoop.mapred.TextInputFormat           | NULL
|
| OutputFormat:                 | org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat |
NULL                  |
| Compressed:                   | No                                                 | NULL
|
| Num Buckets:                  | -1                                                 | NULL
|
| Bucket Columns:               | []                                                 | NULL
|
| Sort Columns:                 | []                                                 | NULL
|
| Storage Desc Params:          | NULL                                               | NULL
|
|                               | field.delim                                        | \t
|
|                               | serialization.format                               | \t
|
+-------------------------------+----------------------------------------------------+-----------
------------+
31 rows selected (0.123 seconds)
```

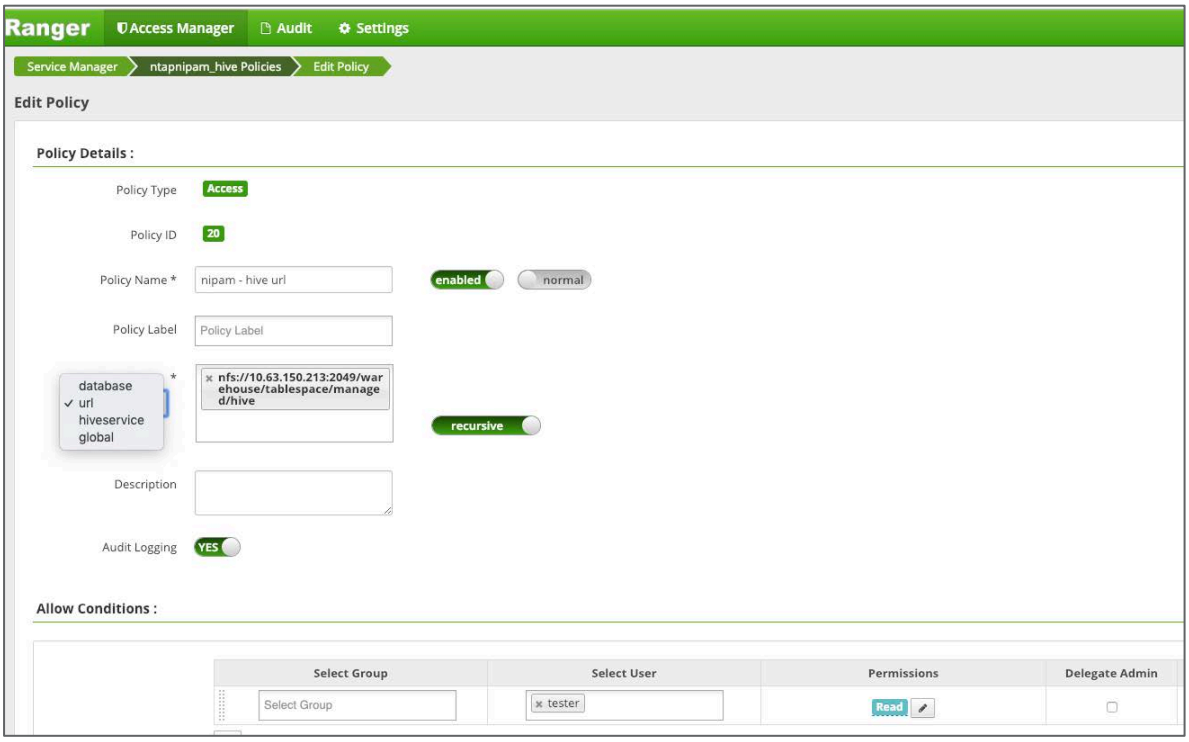
```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> exit
. . . . . . . . . . . . . . . . . . . . . . . .> ;
Error: Error while compiling statement: FAILED: ParseException line 1:0 cannot recognize input
near 'exit' '<EOF>' '<EOF>' (state=42000,code=40000)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> Closing: 0: jdbc:hive2://stl-kserver-01:2181,stl-
kserver-02:2181,stl-kserver-
03:2181/default;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2

[root@stl-kserver-08 tez]# su - hive
Last login: Thu Jan 10 16:45:43 EST 2019 on pts/2
[hive@stl-kserver-08 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive
Found 3 items
-rw-r--r--   1 hive hadoop       2029 2019-01-10 10:45
nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/idnames2_copy_1.txt
drwxrwxrwx   - hive hadoop       4096 2019-01-10 10:39
nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/netapphive_internal_nfs_table1
drwxrwxrwx   - hive hadoop       4096 2019-01-10 11:19
nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/nfs1
[hive@stl-kserver-08 ~]$ hadoop fs -cat
nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/idnames2_copy_1.txt
1       Olivia
2       Cora
3       Amelia
4       Charlotte
5       Isla
6       Isabella
7       Maia
8       Aurora
9       Amara
10      Ava
…

11      Rumi
199     Edward
200     Knox
201
[hive@stl-kserver-08 ~]$
[hive@stl-kserver-08 ~]$
```

12. If you do not have create permission in Hive and try to create an external table using the `create external table` command, the following error displays: `user [tester] does not have [READ] privilege on`.

```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> CREATE external TABLE
netapphive_external_nfs_table6 (id INT,Name STRING)  Row format delimited  Fields terminated by
'\t' stored as textfile location "nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/";
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied:
user [tester] does not have [READ] privilege on
[nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive] (state=42000,code=40000)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> Closing: 0: jdbc:hive2://stl-kserver-01:2181,stl-
kserver-02:2181,stl-kserver-
03:2181/default;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2
[tester@stl-kserver-08 ~]$
```



13. Create a Hive URL policy in Ranger Hive service and select ‘url’ in Hive Policy. Enter the ‘url’ value to `nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/` and provide read access for URL in Hive URL policy.



Service Name	Service Type	Host Name	Plugin IP	Policy (Time)			Tag (Time)		
				Active	Download	Last Update	Active	Download	Last Update
ntapnipam_hive	hiveServer2	stl-kserver-03	10.63.150.65	01/17/2019 02:47:48 PM	01/17/2019 02:47:48 PM	01/17/2019 02:47:30 PM	--	--	--

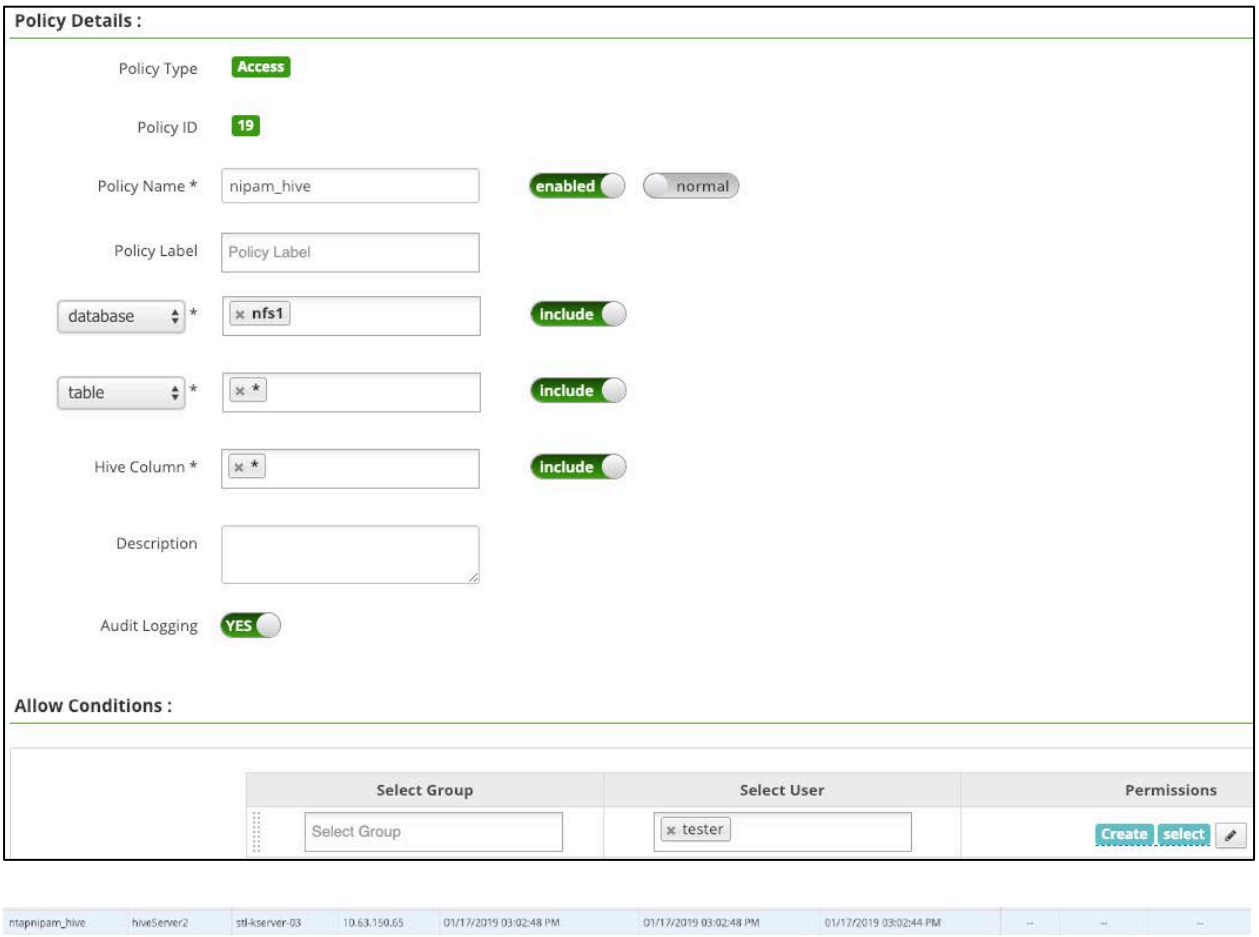
14. Create the external table without create permission in Hive.

```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> CREATE external TABLE
netapphive_external_nfs_table6 (id INT,Name STRING)  Row format delimited  Fields terminated by
'\t' stored as textfile location "nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/";
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied:
user [tester] does not have [CREATE] privilege on [nfs1/netapphive_external_nfs_table6]
(state=42000,code=40000)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

15. Check the audit log.

Policy ID	Event Time	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Client IP	Cluster Name	Event Count	Tags
--	01/17/2019 02:57:05 PM	tester	ntapnipam_hive hive	nfs1/netapphive_external_nfs_ta... @table	CREATE	Denied	ranger-acl	10.63.150.58	ntapnipam	1	--
--	01/17/2019 02:50:23 PM	tester	ntapnipam_hive hive	nfs://10.63.150.213:2049/wareho... @url	READ	Denied	ranger-acl	10.63.150.58	ntapnipam	1	--
19	01/17/2019 02:50:03 PM	tester	ntapnipam_hive hive	nfs1 @database	USE	Allowed	ranger-acl	10.63.150.58	ntapnipam	1	--
19	01/17/2019 02:50:00 PM	tester	ntapnipam_hive hive	nfs1 @database	USE	Allowed	ranger-acl	10.63.150.58	ntapnipam	1	--
19	01/17/2019 02:49:57 PM	tester	ntapnipam_hive hive	--	USE	Allowed	ranger-acl	10.63.150.58	ntapnipam	1	--

16. Provide create permission and try again.



17. Create an external table in the NFS.

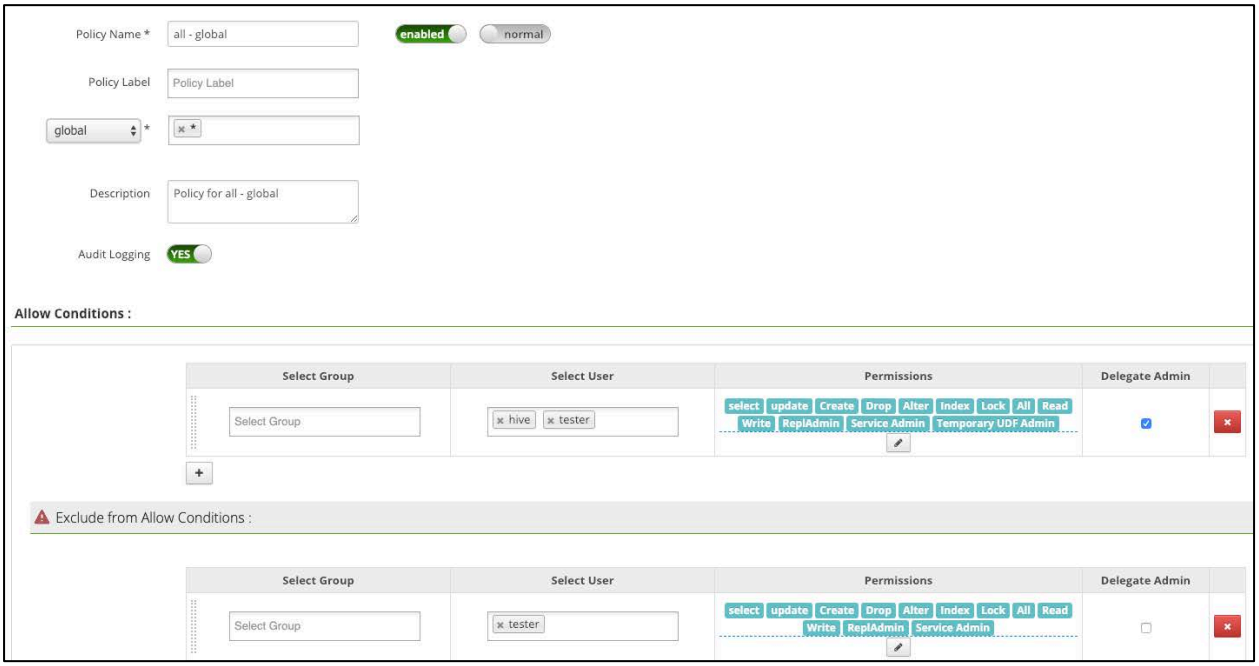
```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> CREATE external TABLE
netapphive_external_nfs_table6 (id INT,Name STRING)  Row format delimited  Fields terminated by
'\t' stored as textfile location "nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/";
INFO  : Compiling command(queryId=hive_20190117150413_34c16fe0-cb7a-4d00-9c1d-2b2a086bd114):
CREATE external TABLE netapphive_external_nfs_table6 (id INT,Name STRING)  Row format delimited
Fields terminated by '\t' stored as textfile location
"nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/"
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20190117150413_34c16fe0-cb7a-4d00-9c1d-
2b2a086bd114); Time taken: 0.034 seconds
INFO  : Executing command(queryId=hive_20190117150413_34c16fe0-cb7a-4d00-9c1d-2b2a086bd114):
CREATE external TABLE netapphive_external_nfs_table6 (id INT,Name STRING)  Row format delimited
Fields terminated by '\t' stored as textfile location
"nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive/"
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20190117150413_34c16fe0-cb7a-4d00-9c1d-
2b2a086bd114); Time taken: 0.094 seconds
INFO  : OK
No rows affected (0.15 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

18. Check the number of counts in the external table using “select count(*) from tablename”. It fails with `Class org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem not found.` Try to add the NetApp In-Place Analytics Module jar files in `beeline` or `hbase shell`.

```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> select count(*) from
netapphive_external_nfs_table6;
```

```
INFO  : Compiling command(queryId=hive_20190117150602_c5d52540-514c-41e7-8bda-50fca3f1199e):
select count(*) from netapphive_external_nfs_table6
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:_c0, type:bigint,
comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20190117150602_c5d52540-514c-41e7-8bda-
50fca3f1199e); Time taken: 0.135 seconds
INFO  : Executing command(queryId=hive_20190117150602_c5d52540-514c-41e7-8bda-50fca3f1199e):
select count(*) from netapphive_external_nfs_table6
INFO  : Query ID = hive_20190117150602_c5d52540-514c-41e7-8bda-50fca3f1199e
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Subscribed to counters: [] for queryId: hive_20190117150602_c5d52540-514c-41e7-8bda-
50fca3f1199e
INFO  : Tez session hasn't been created yet. Opening session
INFO  : Dag name: select count(*) from ..._external_nfs_table6 (Stage-1)
ERROR : Status: Failed
ERROR : Vertex failed, vertexName=Map 1, vertexId=vertex_1547242149891_0052_1_00,
diagnostics=[Vertex vertex_1547242149891_0052_1_00 [Map 1] killed/failed due
to:ROOT_INPUT_INIT_FAILURE, Vertex Input: netapphive_external_nfs_table6 initializer failed,
vertex=vertex_1547242149891_0052_1_00 [Map 1], java.lang.RuntimeException:
java.lang.ClassNotFoundException: Class org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem not found
…
<output shortened to reduce the size of the page>
…
Vertex killed, vertexName=Reducer 2, vertexId=vertex_1547242149891_0052_1_01, diagnostics=[Vertex
received Kill in INITED state., Vertex vertex_1547242149891_0052_1_01 [Reducer 2] killed/failed
due to:OTHER_VERTEX_FAILURE]DAG did not succeed due to VERTEX_FAILURE. failedVertices:1
killedVertices:1 (state=08S01,code=2)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> add jar /usr/hdp/3.0.1.0-187/tez/hadoop-nfs-
2.7.1.jar;
Error: Error while processing statement: Permission denied: user [tester] does not have
[TEMPUDFADMIN] privilege on [global] (state=,code=1)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

19. You need a temporary UDF admin access on Hive global policy for the tester. First add all global permissions to the tester and then exclude all permissions except “Temporary UDF Admin’ from the “Exclude from allow conditions” tab.



20. Exit from beeline and log in again to set the memory parameters and add jar files for NIPAM.

```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> set hive.tez.container.size=2048;
No rows affected (0.005 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> set hive.tez.java.opts=-Xmx1640m;
No rows affected (0.003 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> set tez.runtime.io.sort.mb=820;
No rows affected (0.003 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>  set tez.runtime.unordered.output.buffer.size-
mb=205;
No rows affected (0.003 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> add jar /usr/hdp/3.0.1.0-187/tez/hadoop-nfs-
2.7.1.jar;
INFO  : Added [/usr/hdp/3.0.1.0-187/tez/hadoop-nfs-2.7.1.jar] to class path
INFO  : Added resources: [/usr/hdp/3.0.1.0-187/tez/hadoop-nfs-2.7.1.jar]
No rows affected (0.006 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> add jar /usr/hdp/3.0.1.0-187/tez/hadoop-nfs-
connector-3.0.1.jar;
INFO  : Added [/usr/hdp/3.0.1.0-187/tez/hadoop-nfs-connector-3.0.1.jar] to class path
INFO  : Added resources: [/usr/hdp/3.0.1.0-187/tez/hadoop-nfs-connector-3.0.1.jar]
No rows affected (0.004 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>  add jar /usr/hdp/3.0.1.0-187/tez/ranger-plugins-
audit-1.0.0.jar;
INFO  : Added [/usr/hdp/3.0.1.0-187/tez/ranger-plugins-audit-1.0.0.jar] to class path
INFO  : Added resources: [/usr/hdp/3.0.1.0-187/tez/ranger-plugins-audit-1.0.0.jar]
No rows affected (0.004 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> select count(*) from
netapphive_external_nfs_table6;
…
<output shortened to reduce the size of the page>
…
INFO  : Status: DAG finished successfully in 2.97 seconds
INFO  :
INFO  : Query Execution Summary
INFO  : ----------------------------------------------------------------------------------------------
-----
INFO  : OPERATION                            DURATION
INFO  : ----------------------------------------------------------------------------------------------
-----
INFO  : Compile Query                          0.20s
INFO  : Prepare Plan                           3.96s
INFO  : Get Query Coordinator (AM)             0.00s
INFO  : Submit Plan                            0.23s
INFO  : Start DAG                              0.80s
INFO  : Run DAG                                2.97s
INFO  : ----------------------------------------------------------------------------------------------
-----
INFO  :
INFO  : Task Execution Summary
INFO  : ----------------------------------------------------------------------------------------------
-----
INFO  :   VERTICES      DURATION(ms)   CPU_TIME(ms)    GC_TIME(ms)   INPUT_RECORDS
OUTPUT_RECORDS
INFO  : ----------------------------------------------------------------------------------------------
-----
INFO  :      Map 1           1008.00          3,670             60             201
1
INFO  :  Reducer 2            442.00          3,070             53               1
0
INFO  : ----------------------------------------------------------------------------------------------
-----
…
<output shortened to reduce the size of the page>
…
INFO  : Completed executing command(queryId=hive_20190117151810_36b11f89-65f0-4e16-b5bf-
012be85ff889); Time taken: 7.967 seconds
INFO  : OK
------------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
------------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
------------------------------------------------------------------------------------------------
```

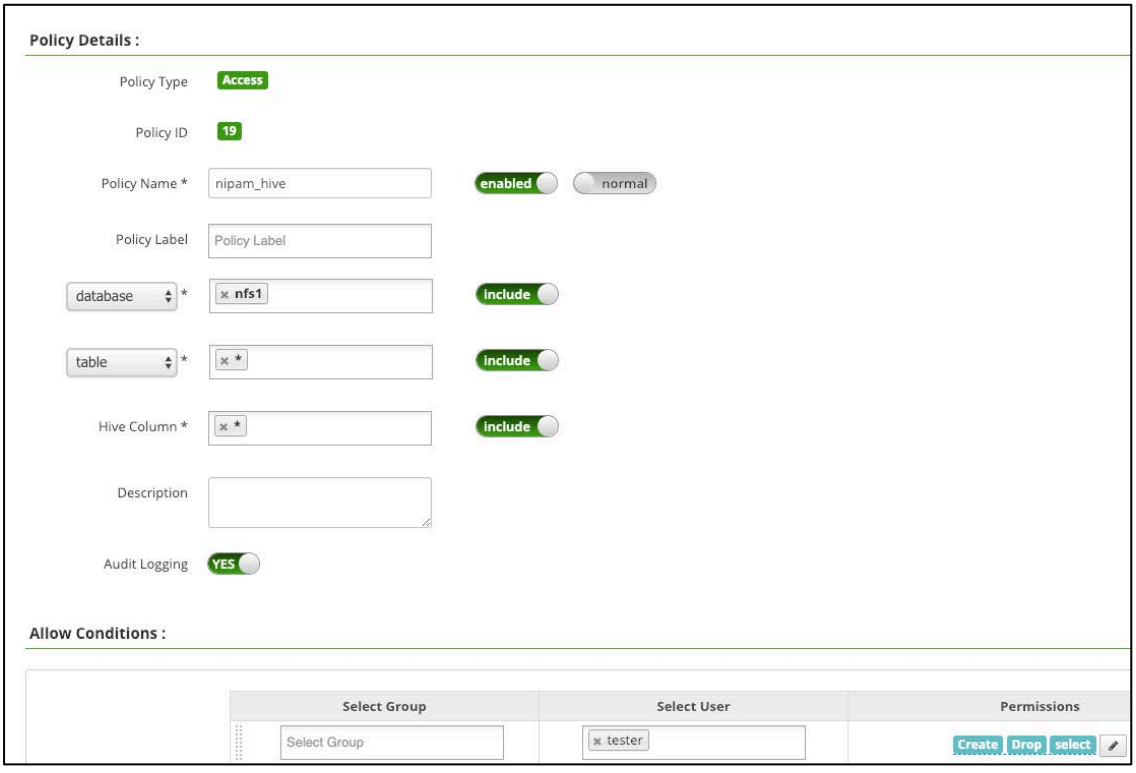
```
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 3.70 s
--------------------------------------------------------------------------------------------------------
+------+
| _c0  |
+------+
| 201  |
+------+
1 row selected (8.912 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

Policy ID	Event Time	User	Service Name / Type	Resource Name / Type	Access Type	Result	Access Enforcer	Client IP	Cluster Name	Event Count	Tags
19	01/17/2019 03:18:10 PM	tester	ntapnipam_hive hive	nfs1/netapphive_external_nfs_ta... @table	SELECT	Allowed	ranger-acl	10.63.150.58	ntapnipam	1	--
13	01/17/2019 03:18:01 PM	tester	ntapnipam_hive hive	global @global	TEMPUDFADMIN	Allowed	ranger-acl	10.63.150.58	ntapnipam	1	--
13	01/17/2019 03:17:56 PM	tester	ntapnipam_hive hive	global @global	TEMPUDFADMIN	Allowed	ranger-acl	10.63.150.58	ntapnipam	1	--
13	01/17/2019 03:17:52 PM	tester	ntapnipam_hive hive	global @global	TEMPUDFADMIN	Allowed	ranger-acl	10.63.150.58	ntapnipam	1	--

21. Drop the external table without drop permission in Ranger.

```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> drop table netapphive_external_nfs_table6;
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied:
user [tester] does not have [DROP] privilege on [nfs1/netapphive_external_nfs_table6]
(state=42000,code=40000)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

22. Provide drop permission in Ranger and wait until the active and download time are same in Ranger audit.



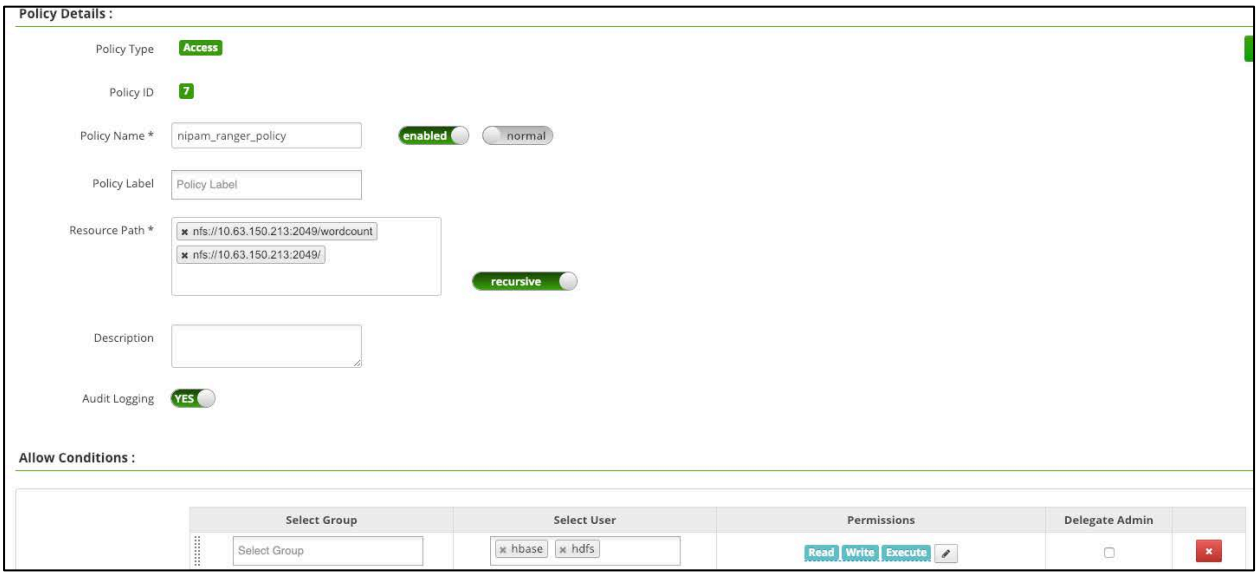
Service Name	Service Type	Host Name	Plugin IP	Policy (Time) Active	Policy (Time) Download	Policy (Time) Last Update	Tag (Time) Active	Tag (Time) Download	Tag (Time) Last Update
ntapnipam_hive	hiveServer2	stl-kserver-03	10.63.150.65	01/17/2019 03:27:48 PM	01/17/2019 03:27:48 PM	01/17/2019 03:27:29 PM	--	--	--

23. Check the drop operation in Hive.

```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> drop table netapphive_external_nfs_table6;
INFO  : Compiling command(queryId=hive_20190117152909_0b66f9b8-2b54-492a-a5b9-e6dbc2422c01): drop
table netapphive_external_nfs_table6
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20190117152909_0b66f9b8-2b54-492a-a5b9-
e6dbc2422c01); Time taken: 0.029 seconds
INFO  : Executing command(queryId=hive_20190117152909_0b66f9b8-2b54-492a-a5b9-e6dbc2422c01): drop
table netapphive_external_nfs_table6
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20190117152909_0b66f9b8-2b54-492a-a5b9-
e6dbc2422c01); Time taken: 0.067 seconds
INFO  : OK
No rows affected (0.111 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

24. Operations are performed by the tester but they are executed by the Hive user because when you enable Ranger the “doAs” parameter is false. This means all Hive operations are done by Hive user. For NFS, you must provide read, write, and execute permission for the Hive user and not for the user “tester”. The user column does not have a user “tester”.

25. Remove the Hive from Ranger NFS folder from HDFS policy which was created for NFS folder and check select count(*) from the <external table> command.



26. When you change the NFS policy in HDFS service for Hive, you must restart the Hive service. Remove the user Hive from the NFS policy and restart Hive service from Ambari framework.

```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> set hive.tez.container.size=2048;
No rows affected (0.009 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> set hive.tez.java.opts=-Xmx1640m;
No rows affected (0.005 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> set tez.runtime.io.sort.mb=820;
No rows affected (0.005 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>  set tez.runtime.unordered.output.buffer.size-
mb=205;
No rows affected (0.005 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> add jar /usr/hdp/3.0.1.0-187/tez/hadoop-nfs-
2.7.1.jar;
INFO  : Added [/usr/hdp/3.0.1.0-187/tez/hadoop-nfs-2.7.1.jar] to class path
INFO  : Added resources: [/usr/hdp/3.0.1.0-187/tez/hadoop-nfs-2.7.1.jar]
No rows affected (0.013 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> add jar /usr/hdp/3.0.1.0-187/tez/hadoop-nfs-
connector-3.0.1.jar;
INFO  : Added [/usr/hdp/3.0.1.0-187/tez/hadoop-nfs-connector-3.0.1.jar] to class path
INFO  : Added resources: [/usr/hdp/3.0.1.0-187/tez/hadoop-nfs-connector-3.0.1.jar]
No rows affected (0.006 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>  add jar /usr/hdp/3.0.1.0-187/tez/ranger-plugins-
audit-1.0.0.jar;
INFO  : Added [/usr/hdp/3.0.1.0-187/tez/ranger-plugins-audit-1.0.0.jar] to class path
INFO  : Added resources: [/usr/hdp/3.0.1.0-187/tez/ranger-plugins-audit-1.0.0.jar]
No rows affected (0.006 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> select count(*) from
netapphive_external_nfs_table2;
INFO  : Compiling command(queryId=hive_20190117154643_5da34f31-deb0-4bbb-883b-a3c5ad49e377):
select count(*) from netapphive_external_nfs_table2
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:_c0, type:bigint,
comment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20190117154643_5da34f31-deb0-4bbb-883b-
a3c5ad49e377); Time taken: 2.463 seconds
INFO  : Executing command(queryId=hive_20190117154643_5da34f31-deb0-4bbb-883b-a3c5ad49e377):
select count(*) from netapphive_external_nfs_table2
INFO  : Query ID = hive_20190117154643_5da34f31-deb0-4bbb-883b-a3c5ad49e377
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Subscribed to counters: [] for queryId: hive_20190117154643_5da34f31-deb0-4bbb-883b-
a3c5ad49e377
INFO  : Tez session hasn't been created yet. Opening session
```

```
INFO  : Dag name: select count(*) from ..._external_nfs_table2 (Stage-1)
ERROR : Status: Failed
ERROR : Vertex failed, vertexName=Map 1, vertexId=vertex_1547242149891_0055_1_00,
diagnostics=[Vertex vertex_1547242149891_0055_1_00 [Map 1] killed/failed due
to:ROOT_INPUT_INIT_FAILURE, Vertex Input: netapphive_external_nfs_table2 initializer failed,
vertex=vertex_1547242149891_0055_1_00 [Map 1], org.apache.hadoop.security.AccessControlException:
Ranger ACL : execute permission for listStatus() operation denied for path
nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive for user : hive, primary group :
Hadoop
…
<output shortened to reduce the size of the page>
…
INFO  : Completed executing command(queryId=hive_20190117154643_5da34f31-deb0-4bbb-883b-
a3c5ad49e377); Time taken: 5.075 seconds
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1            container  INITIALIZING     -1          0        0       -1       0       0
Reducer 2        container        INITED      1          0        0        1       0       0
----------------------------------------------------------------------------------------------
VERTICES: 00/02  [>>-------------------------] 0%    ELAPSED TIME: 4.20 s
----------------------------------------------------------------------------------------------
Error: Error while processing statement: FAILED: Execution Error, return code 2 from
org.apache.hadoop.hive.ql.exec.tez.TezTask. Vertex failed, vertexName=Map 1,
vertexId=vertex_1547242149891_0055_1_00, diagnostics=[Vertex vertex_1547242149891_0055_1_00 [Map
1] killed/failed due to:ROOT_INPUT_INIT_FAILURE, Vertex Input: netapphive_external_nfs_table2
initializer failed, vertex=vertex_1547242149891_0055_1_00 [Map 1],
org.apache.hadoop.security.AccessControlException: Ranger ACL : execute permission for
listStatus() operation denied for path nfs://10.63.150.213:2049/warehouse/tablespace/managed/hive
for user : hive, primary group : Hadoop
…
<output shortened to reduce the size of the page>
…
```

27. Provide read, write, and execute access for the Hive user in the NFS in HDFS policy which was created for NFS folder. Restart the Hive service from Ambari and try the select count(*) from external table command from beeline again.


```
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv> select count(*) from
netapphive_external_nfs_table2;
…
<output shortened to reduce the size of the page>
…
INFO  : Completed executing command(queryId=hive_20190117155609_052c9917-3c54-43bf-ace3-
33ba703d1482); Time taken: 7.861 seconds
INFO  : OK
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 3.80 s
----------------------------------------------------------------------------------------------
+------+
| _c0  |
+------+
| 201  |
+------+
1 row selected (10.513 seconds)
0: jdbc:hive2://stl-kserver-01:2181,stl-kserv>
```

10 Solutions for Error Messages

This section provides solutions for error messages that NetApp encountered during the validation process.

10.1 Failed to Export Ranger Policies from Admin Server

```
[tester@hdp01 ~]$ hadoop fs -ls nfs://10.67.12.26:2049/datalake
19/01/10 14:47:44 WARN ranger.NfsRangerAuthorization: NfsConnectorV3.0.1 Failed to export ranger
policies from admin server.
19/01/10 14:47:44 WARN ranger.NfsRangerAuthorization: NfsConnectorV3.0.1 Failed to export ranger
policies from admin server.
ls: Failed to export ranger policies from admin server.
[tester@hdp01 ~]$
```

Solution:

1. Check the `nfsRangerServiceName` parameter to HDFS repository name in your Ranger cluster. To do this, navigate to Ranger portal > Access Manager > HDFS > <clustername>_hadoop.
2. Update the `nfsRangerServiceName` parameter in nfs-mapping.json in NIPAM configuration.
3. Click Save > Restart Required > Restart.

10.2 NfsRangerAudit() Failed to Initialize Logger

```
[tester@hdp01 ~]$ hadoop fs -ls nfs://10.67.12.26:2049/datalake
19/01/10 14:55:15 ERROR netapp.NfsRangerAudit: NfsConnectorV3.0.1 NfsRangerAudit() failed to
initialiaze logger.
 Exception: java.io.FileNotFoundException:
/home/ranger/RANGER_AUDIT_NFS/ranger_nfs_audit_hdp01.stratus9.co.uk_tester_7019_1547132115340.log
(Permission denied)
ls: NfsRangerAudit() failed to initialiaze logger.
[tester@hdp01 ~]$
```

Solution:

1. Check that the `nfsRangerAuditLogPath` parameter folder exists in all members of Hadoop cluster.

   `pssh -h /etc/hosts “mkdir -p <folder_full_path>”`

   `pssh -h /etc/hosts “chmod -R 777 <folder_full_path>”`
2. Check the permission for `<folder_full_path>`.

11 Conclusion

NIPAM for Apache Hadoop and Spark with Ranger covered in this report includes the Ranger configuration, underlying architecture, integration with Hadoop, and benefits of Ranger with NIPAM using Hadoop with NetApp ONTAP data management software.

Where to Find Additional Information

To learn more about the information described in this document, refer to the following documents and/or websites:

- Hortonworks and Ranger
  https://hortonworks.com/apache/ranger/
- NetApp In-Place Analytics Module Best Practices
  https://www.netapp.com/us/media/tr-4382.pdf
- NetApp In-Place Analytics Module Joint Reference Architecture
  https://www.netapp.com/us/media/tr-4715.pdf

- Apache Hadoop YARN—Concepts and Applications
  http://hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/
- Apache HBase
  https://en.wikipedia.org/wiki/Apache_HBase
- HBase Performance Evaluation
  https://www.cloudera.com/documentation/enterprise/5-9-x/topics/cdh_ig_hbase_tools.html
- Hive and Ranger Configuration in Hortonworks
  https://hortonworks.com/blog/best-practices-for-hive-authorization-using-apache-ranger-in-hdp-2-2/
- NetApp Documentation Center
  https://docs.netapp.com

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Trademark Information