Technical Report

NetApp In-Place Analytics Module with Kerberos

Massachusetts Institute of Technology and Windows Active Directory

Karthikeyan Nagalingam and Justin Parisi, NetApp
September 2019 | TR-4797

Abstract

This document describes Kerberos with the NetApp® In-Place Analytics Module. We cover the following two Kerberos configurations:

- Massachusetts Institute of Technology Key Distribution Center (MIT KDC)
- Microsoft Active Directory and LDAP with the MIT KDC

NetApp®

TABLE OF CONTENTS

1 Introduction 3

1.1 Kerberos Terminology 3

2 NetApp In-Place Analytics Module Installation and Configuration 6

3 MIT Kerberos 6

3.1 MIT KDC Configuration 6

3.2 Hortonworks Kerberos Configuration 10

4 Kerberos with Active Directory and MIT 24

4.1 Client/Hadoop Cluster Configuration – Kerberos Client: Active Directory 24

4.2 Active Directory Configuration – LDAP 25

4.3 ONTAP Cluster Configuration – LDAP Client 26

4.4 Client/Hadoop Cluster Configuration – LDAP Client: SSSD 29

4.5 Active Directory Configuration – DNS 30

4.6 ONTAP Cluster Configuration – Kerberos 30

4.7 Windows AD KDC Configuration – Kerberos Machine Accounts 33

4.8 Generate Keytab Files for Passwordless Access to Hadoop NFS Connector 33

4.9 Test the NFSv3 Kerberos Mount 34

4.10 Hadoop NFS Connector Config File 35

4.11 Test the NetApp In-Place Analytics Module(NIPAM) 36

4.12 NFSv4 ACLs on NFSv3 Objects 37

Where to Find Additional Information 38

Version History 39

LIST OF TABLES

Table 1) MIT Kerberos cluster test bed details. 4

Table 2) Windows active directory and Hadoop cluster details. 5

LIST OF FIGURES

Figure 1) Kerberos realm 3

Figure 2) General environment 24

1 Introduction

This document describes using Kerberos with the NetApp In-Place Analytics Module. We cover the following two Kerberos configurations:

- Massachusetts Institute of Technology Key Distribution Center (MIT KDC)
- Microsoft Active Directory and LDAP with the MIT KDC

1.1 Kerberos Terminology

This section defines key terminology for Kerberos processes that might be unfamiliar to storage administrators.

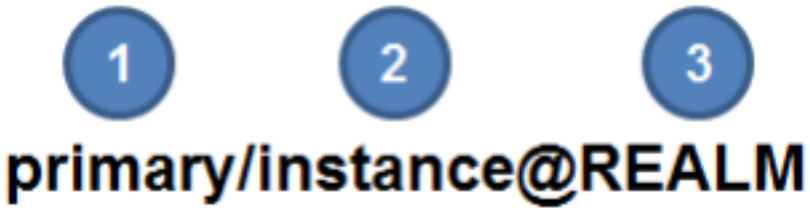
Key Distribution Center

A key distribution center (KDC) is an authentication server that includes a ticket-granting service (TGS) and authentication services (AS). In Microsoft environments, an Active Directory domain controller serves as the KDC, but customers also use the MIT KDC with Active directory for user authentication via LDAP.

Principal

The term principal refers to every entity within a Kerberos database. Users, computers, and services that run on a client are all principals. Every principal is unique within the Kerberos database and is defined by its name. A principal can be a user principal name (UPN) or a service principal name (SPN).

Figure 1) Kerberos realm.



Realm (or Kerberos Realm)

A realm (or Kerberos realm) can use any ASCII string. Typically, the domain name is formatted in uppercase. For example, domain.com becomes the realm DOMAIN.COM.

Administratively, each principal@REALM is unique. To avoid a single point of failure, each realm can have numerous KDCs that share the same database (principals and their passwords) and have the same KDC master keys. Microsoft Windows Active Directory does this natively by way of Active Directory replication, which takes place every 15 minutes by default.

The Primary Component

The primary component can be a user or a service such as the NFS service. It can also be the special service host that signifies that this service principal is set up to provide various network services such as FTP, RSH, NFS, and so on.

The Instance Component

The instance component is optional in the case of a user. A user can have more than one principal. For example, one user might have one principal that is for everyday use and another principal that allows privileged use, such as a sysadmin account. The instance is required for service principals and designates the fully qualified domain name (FQDN) of the host that provides the service.

The Realm Component

A Kerberos realm is the set of Kerberos principals that are registered within a Kerberos server. By convention, the realm name is usually the same as the DNS name, but it is converted to uppercase letters. Uppercase letters are not obligatory, but this convention allows easy distinction between the DNS name and the realm name.

Examples of principals:

- user@DOMAIN.COM
- user/admin@DOMAIN.COM
- host/host.domain.com@DOMAIN.COM
- root/host.domain.com@DOMAIN.COM
- nfs/host.domain.com@DOMAIN.COM

Tickets

A ticket is a temporary set of credentials that verifies the identity of a principal for a service and contains the session key. A ticket can be a service or an application ticket or a ticket-granting ticket (TGT).

Secret Keys

Kerberos uses a symmetrical key system in which the secret key is used for both encryption and decryption. The secret key is generated from the principal's Kerberos password with a one-way hash function. The KDC stores the password for each principal and can thus generate the principal's secret key. For users who request a Kerberos service, the secret key is typically derived from a password that is presented to the `kinit` program. Service and daemon principals typically don't use a password; instead, the result of the one-way hash function is stored in a keytab.

Keytab

A keytab contains a list of principals and their secret keys. The secret keys in a keytab are often created by using a random password and are used mostly for service or daemon principals.

Test Bed Details

In this validation, we used two test beds. One was for the MIT cluster Kerberos configuration, which was managed by a Hadoop cluster. The other used Windows Active Directory and a Hadoop cluster.

Table 1) MIT Kerberos cluster test bed details.

Component	Product or Solution	Details
Storage	NetApp A300 storage array with ONTAP 9.6	• 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 with 2 aggregates • 2 x 4 x 10GbE • NetApp RAID DP® technology

Component	Product or Solution	Details
Servers	PRIMERGY RX2540 M1 Intel Xeon CPU E5-2670 v3 at 2.30GHz	• 4 servers, each with two 2.4GHz (6-core) or 2.3GHz (8-core) processors (40 cores) • 12 DIMMs (up to 192GB), up to 1600MHz (512GB per node) • 1 x 1Gbps Ethernet port, 2 x 10GbE network ports
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.
Server operating system	Red Hat Enterprise Linux Server 7.6 (x86_64) or later	Hadoop typically requires a Linux distribution.
Hadoop distribution	Hortonworks Data Platform 3.1.0.0-78 (tested)	Apache Ambari 2.7.3

Table 2) Windows active directory and Hadoop cluster details.

Component	Product or Solution	Details
Storage	NetApp A300 storage array with ONTAP 9.6	• 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 with 2 aggregates • 2 x 4 x 10GbE • RAID DP
Servers	PRIMERGY RX2540 M1 Intel Xeon CPU E5-2670 v3 at 2.30GHz	• 10 servers, and each with two 2.4GHz (6-core) or 2.3GHz (8-core) processors (40 cores) • 12 DIMMs (up to 192GB), up to 1600MHz (512GB/node) • 1 x 1Gbps Ethernet port, 2 x 10GbE network ports
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.
Server operating system	Red Hat Enterprise Linux Server 7.6 (x86_64) or later	Hadoop typically requires a Linux distribution.
	Windows Operating system 2012	Active Directory configuration
Hadoop distribution	CDH-5.14.0-1.cdh5.14.0.p0.24 (tested)	Cloudera manager 5.14.1-1.cm5141.p0.1.el7.x86_64

2 NetApp In-Place Analytics Module Installation and Configuration

The NetApp In-Place Analytics Module installation consists of three parts:

- Configure the FAS/AFF storage controller.
- Configure the Hadoop cluster.
- Create the JSON configuration file.

These three sections are explained in details in TR-4382.

The first step is to download the NetApp In-Place Analytics Module software and installation guide from the NetApp website.

Note: For a sample JSON file with Kerberos, see the section "MIT Kerberos."

3 MIT Kerberos

3.1 MIT KDC Configuration

To install and configure MIT KDC, complete the following steps:

1. Install the following packages on the KDC server using YUM or RPM.

```
yum install -y krb5-server krb5-workstation krb5-libs pam_krb5
```

2. Update the Kerberos configuration file `/etc/krb5.conf` for the `default_realm`, `admin_server`, `kdc`, and `domain_realm` as follows:

```
[root@hdp2 ~]# cat /etc/krb5.conf

[libdefaults]
  renew_lifetime = 7d
  forwardable = true
  default_realm = NIPAMTEST.COM
  ticket_lifetime = 24h
  dns_lookup_realm = false
  dns_lookup_kdc = false
  default_ccache_name = /tmp/krb5cc_%{uid}
  #default_tgs_enctypes = aes des3-cbc-sha1 rc4 des-cbc-md5
  #default_tkt_enctypes = aes des3-cbc-sha1 rc4 des-cbc-md5

[domain_realm]
  NIPAMTEST.COM = NIPAMTEST.COM
  .nipamtest.com = NIPAMTEST.COM
  .rtpppe.netapp.com = NIPAMTEST.COM

[logging]
  default = FILE:/var/log/krb5kdc.log
  admin_server = FILE:/var/log/kadmind.log
  kdc = FILE:/var/log/krb5kdc.log

[realms]
  NIPAMTEST.COM = {
    admin_server = hdp2.rtpppe.netapp.com
    kdc = 10.63.150.176
  }
```

Note: We can use an FQDN for both the `admin_server` and KDC. In this example, we used the FQDN and the IP.

3. The Kerberos `kadmind` daemon uses an access control list (ACL) file to manage access rights to the Kerberos database. For operations that affect principals, the ACL file also controls which principals can operate on which other principals. The default location of the Kerberos ACL file is `/var/kerberos/krb5kdc/kadm5.acl`.

```
[root@hdp2 ~]# cat /var/kerberos/krb5kdc/kadm5.acl
*/admin@NIPAMTEST.COM  *
[root@hdp2 ~]#
```

4. Add host file entries for the Hadoop cluster nodes, KDCs, and NFS SVM data LIFS.

```
[testeruser1@hdp2 nc_volume2]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

10.63.150.161   hdp1.rtpppe.netapp.com hdp1
10.63.150.163   hdp1.rtpppe.netapp.com hdp1
10.63.150.165   hdp1.rtpppe.netapp.com hdp1
10.63.150.176   hdp2.rtpppe.netapp.com hdp2
10.63.150.177   hdp3.rtpppe.netapp.com hdp3
10.63.150.178   hdp4.rtpppe.netapp.com hdp4

10.193.92.159 hortonworks-g1.stl.netapp.com

10.63.150.176 hdp2.rtpppe.netapp.com hdp2 hdp2.rtpppe.netapp.com.
10.63.150.177 hdp3.rtpppe.netapp.com hdp3 hdp3.rtpppe.netapp.com.

10.63.150.213   hadoop_svm_nfs.nipamtest.com hadoop_svm_nfs
10.63.150.211   hadoop_svm_nfs.nipamtest.com hadoop_svm_nfs
10.63.150.123   hadoop_svm_nfs.nipamtest.com hadoop_svm_nfs
10.63.150.124   hadoop_svm_nfs.nipamtest.com hadoop_svm_nfs
10.63.150.125   hadoop_svm_nfs.nipamtest.com hadoop_svm_nfs
10.63.150.126   hadoop_svm_nfs.nipamtest.com hadoop_svm_nfs
10.63.150.127   hadoop_svm_nfs.nipamtest.com hadoop_svm_nfs
10.63.150.128   hadoop_svm_nfs.nipamtest.com hadoop_svm_nfs
```

5. The kdc.conf file supplements the krb5.conf file for programs that are typically only used on a KDC, such as the `krb5kdc` and `kadmind` daemons and the `kdb5_util` program.

```
 [root@hdp2 ~]# cat /var/kerberos/krb5kdc/kdc.conf
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
 NIPAMTEST.COM = {
  #master_key_type = aes256-cts
  acl_file = /var/kerberos/krb5kdc/kadm5.acl
  dict_file = /usr/share/dict/words
  admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
  supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal camellia256-
cts:normal camellia128-cts:normal des-hmac-sha1:normal des-cbc-md5:normal des-cbc-crc:normal
 }
[root@hdp2 ~]#
```

Note: ONTAP only supports des, 3des, Aes-128 and aes-256. ONTAP does not support arcfour. Check TR-4073 for more details about MIT configuruation.

6. `kdb5_util` allows an administrator to perform maintenance procedures on the KDC database. Databases can be created, destroyed, and dumped to or loaded from ASCII files. `kdb5_util` can create a Kerberos master key stash file or perform live rollover of the master key. When `kdb5_util` is run, it attempts to acquire the master key and open the database. However, execution continues regardless of whether `kdb5_util` successfully opens the database, because the database might not exist yet or the stash file may be corrupt.

```
[root@hdp2 ~]# sudo kdb5_util create -s NIPAMTEST.COM
Loading random data
Initializing database '/var/kerberos/krb5kdc/principal' for realm 'NIPAMTEST.COM',
master key name 'K/M@NIPAMTEST.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
```



```
[root@hdp2 ~]#
```

You might see the following error:

```
kdb5_util: Cannot open DB2 database '/var/kerberos/krb5kdc/principal': File exists while creating
database '/var/kerberos/krb5kdc/principal'
```

If so, the KDC database already exists. To recreate the database, you can remove the `/var/kerberos/krb5kdc/principal` folder and run `kdb5_util create` again.

7. krb5kdc is the Kerberos version 5 authentication service and key distribution center (AS/KDC).

   kadmin and kadmin.local are command-line interfaces to the Kerberos V5 administration system. They provide nearly identical functionalities; the difference is that kadmin.local directly accesses the KDC database, whereas kadmin performs operations using kadmind. Unless otherwise noted, this document uses kadmin to refer to both versions. kadmin provides for the maintenance of Kerberos principals, password policies, and service key tables (keytabs).

8. Start and enable KDC and administrative server services.

```
[root@hdp2 ~]# sudo systemctl restart krb5kdc
[root@hdp2 ~]# sudo systemctl restart kadmin
[root@hdp2 ~]# sudo systemctl enable krb5kdc
Created symlink from /etc/systemd/system/multi-user.target.wants/krb5kdc.service to
/usr/lib/systemd/system/krb5kdc.service.
[root@hdp2 ~]# sudo systemctl enable kadmin
Created symlink from /etc/systemd/system/multi-user.target.wants/kadmin.service to
/usr/lib/systemd/system/kadmin.service.
[root@hdp2 ~]#
[root@hdp2 ~]#
[root@hdp2 ~]#
[root@hdp2 ~]# sudo systemctl status krb5kdc
● krb5kdc.service - Kerberos 5 KDC
   Loaded: loaded (/usr/lib/systemd/system/krb5kdc.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2019-04-29 11:40:42 EDT; 36s ago
 Main PID: 28602 (krb5kdc)
   CGroup: /system.slice/krb5kdc.service
           └─28602 /usr/sbin/krb5kdc -P /var/run/krb5kdc.pid

Apr 29 11:40:42 hdp2.rtpppe.netapp.com systemd[1]: Starting Kerberos 5 KDC...
Apr 29 11:40:42 hdp2.rtpppe.netapp.com systemd[1]: PID file /var/run/krb5kdc.pid not readable
(yet?) after start.
Apr 29 11:40:42 hdp2.rtpppe.netapp.com systemd[1]: Started Kerberos 5 KDC.
[root@hdp2 ~]# sudo systemctl status kadmin
● kadmin.service - Kerberos 5 Password-changing and Administration
   Loaded: loaded (/usr/lib/systemd/system/kadmin.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2019-04-29 11:40:51 EDT; 46s ago
 Main PID: 28635 (kadmind)
   CGroup: /system.slice/kadmin.service
           └─28635 /usr/sbin/kadmind -P /var/run/kadmind.pid

Apr 29 11:40:51 hdp2.rtpppe.netapp.com systemd[1]: Starting Kerberos 5 Password-changing and
Administration...
Apr 29 11:40:51 hdp2.rtpppe.netapp.com systemd[1]: Started Kerberos 5 Password-changing and
Administration.
[root@hdp2 ~]#
```

9. Create an administrative principal and keytab for administrative users, which is used to authenticate with the KDC when configuring the SVM for Kerberos. In this example, we use the root user as the adminstrator.

```
[root@hdp2 ~]# sudo kadmin.local -q "addprinc root/admin@NIPAMTEST.COM"
Authenticating as principal root/admin@NIPAM.COM with password.
WARNING: no policy specified for root/admin@NIPAMTEST.COM; defaulting to no policy
Enter password for principal "root/admin@NIPAMTEST.COM":
Re-enter password for principal "root/admin@NIPAMTEST.COM":
Principal "root/admin@NIPAMTEST.COM" created.
[root@hdp2 ~]# sudo kadmin.local -q "addprinc admin/admin@NIPAMTEST.COM"
Authenticating as principal root/admin@NIPAM.COM with password.
WARNING: no policy specified for admin/admin@NIPAMTEST.COM; defaulting to no policy
```



```
Enter password for principal "admin/admin@NIPAMTEST.COM":
Re-enter password for principal "admin/admin@NIPAMTEST.COM":
Principal "admin/admin@NIPAMTEST.COM" created.
[root@hdp2 ~]#
```

10. Create the host principal and keytabs.

```
kadmin.local:  ank -randkey host/hdp2
WARNING: no policy specified for host/hdp2@NIPAMTEST.COM; defaulting to no policy
Principal "host/hdp2@NIPAMTEST.COM" created.
kadmin.local:  ank -randkey host/hdp2.rtpppe.netapp.com
WARNING: no policy specified for host/hdp2.rtpppe.netapp.com@NIPAMTEST.COM; defaulting to no
policy
Principal "host/hdp2.rtpppe.netapp.com@NIPAMTEST.COM" created.
kadmin.local:  ktadd -k /etc/krb5.keytab host/hdp2.rtpppe.netapp.com
Entry for principal host/hdp2.rtpppe.netapp.com with kvno 2, encryption type aes256-cts-hmac-
sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2.rtpppe.netapp.com with kvno 2, encryption type aes128-cts-hmac-
sha1-96 added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2.rtpppe.netapp.com with kvno 2, encryption type des3-cbc-sha1 added
to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2.rtpppe.netapp.com with kvno 2, encryption type arcfour-hmac added
to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2.rtpppe.netapp.com with kvno 2, encryption type camellia256-cts-cmac
added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2.rtpppe.netapp.com with kvno 2, encryption type camellia128-cts-cmac
added to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2.rtpppe.netapp.com with kvno 2, encryption type des-hmac-sha1 added
to keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2.rtpppe.netapp.com with kvno 2, encryption type des-cbc-md5 added to
keytab WRFILE:/etc/krb5.keytab.
kadmin.local:  ktadd -k /etc/krb5.keytab host/hdp2
Entry for principal host/hdp2 with kvno 2, encryption type aes256-cts-hmac-sha1-96 added to
keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2 with kvno 2, encryption type aes128-cts-hmac-sha1-96 added to
keytab WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2 with kvno 2, encryption type des3-cbc-sha1 added to keytab
WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2 with kvno 2, encryption type arcfour-hmac added to keytab
WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2 with kvno 2, encryption type camellia256-cts-cmac added to keytab
WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2 with kvno 2, encryption type camellia128-cts-cmac added to keytab
WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2 with kvno 2, encryption type des-hmac-sha1 added to keytab
WRFILE:/etc/krb5.keytab.
Entry for principal host/hdp2 with kvno 2, encryption type des-cbc-md5 added to keytab
WRFILE:/etc/krb5.keytab.
```

11. Test and validate the KDC installation and configuration.

```
[[root@hdp2 ~]# kinit root/admin@NIPAMTEST.COM
Password for root/admin@NIPAMTEST.COM:
[root@hdp2 ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: root/admin@NIPAMTEST.COM

Valid starting       Expires              Service principal
04/29/2019 11:44:21  04/30/2019 11:44:21  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
[root@hdp2 ~]# kinit admin/admin@NIPAMTEST.COM
Password for admin/admin@NIPAMTEST.COM:
[root@hdp2 ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin/admin@NIPAMTEST.COM

Valid starting       Expires              Service principal
04/29/2019 11:44:52  04/30/2019 11:44:52  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
[root@hdp2 ~]#
```

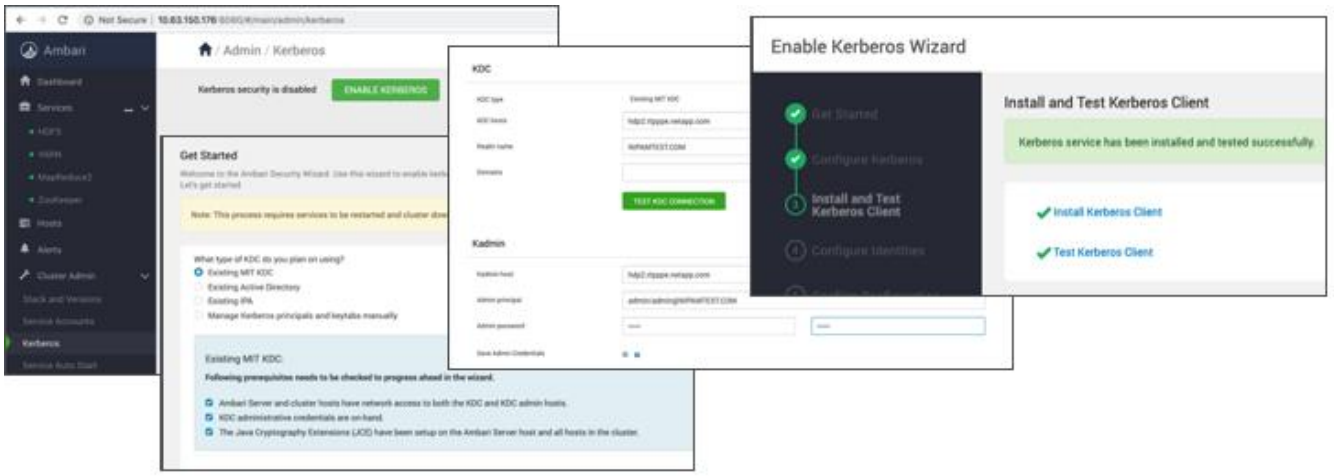

3.2 Hortonworks Kerberos Configuration

In this configuration, we are using KDC on the same Hadoop server. However, you can use external KDCs outside the Hadoop server. If you are using Hortonworks, complete the following steps to configure Kerberos in Hortonworks:

1. Install the Java Cryptography Extension (JCE).

```
root@hdp2 ~]# cd /usr/src/
[root@hdp2 src]#
[root@hdp2 src]#
[root@hdp2 src]# wget --no-check-certificate --no-cookies --header "Cookie: oraclelicense=accept-
securebackup-cookie" http://download.oracle.com/otn-pub/java/jce/8/jce_policy-8.zip
[root@hdp2 src]# ls -ltrh jce*
-rw-r--r-- 1 root root 8.3K Mar 18  2014 jce_policy-8.zip
[root@hdp2 src]#
[root@hdp2 src]# unzip -o -j -q jce_policy-8.zip -d /usr/jdk64/jdk1.8.0_112/jre/lib/security/
[root@hdp2 src]# cd /usr/jdk64/jdk1.8.0_112/jre/lib/security/
[root@hdp2 security]# ls -ltrh
total 172K
-rw-rw-r-- 1 root root 3.0K Dec 20  2013 US_export_policy.jar
-rw-r--r-- 1 root root 7.2K Dec 20  2013 README.txt
-rw-rw-r-- 1 root root 3.0K Dec 20  2013 local_policy.jar
-rw-r--r-- 1 root root    0 Sep 23  2016 trusted.libraries
-rw-r--r-- 1 root root   98 Sep 23  2016 javaws.policy
-rw-r--r-- 1 root root  27K Sep 23  2016 java.security
-rw-r--r-- 1 root root 2.5K Sep 23  2016 java.policy
-rw-r--r-- 1 root root 111K Sep 23  2016 cacerts
-rw-r--r-- 1 root root 1.3K Sep 23  2016 blacklisted.certs
-rw-r--r-- 1 root root 4.0K Sep 23  2016 blacklist
[root@hdp2 security]#
```

2. Restart the Ambari service.
3. Enable Kerberos in Hortonworks using the Ambari framework.



4. Enable the Kerberos config, verify that the prerequisites are met, and enter the admin principal username and password.

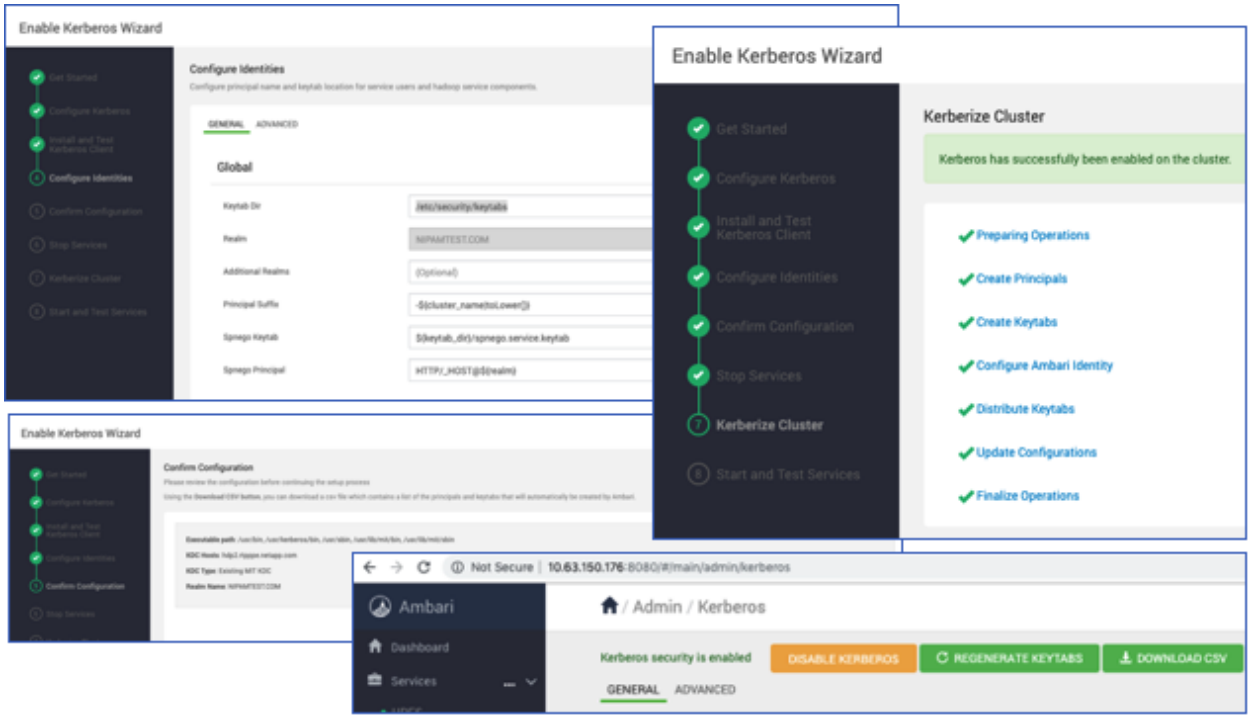
   Note: The admin principal parameter should be in the admin/admin@NIPAMTEST.COM format, or the Test Kerberos Client process fails.

5. Check the Kerberos log to see if the Kerberos login failed.

```
May 07 18:03:32 hdp2.rtpppe.netapp.com krb5kdc[31459](info): AS_REQ (8 etypes {18 17 20 19 16 23
25 26}) 10.63.150.176: ISSUE: authtime 1557266612, etypes {rep=18 tkt=18 ses=18},
admin/admin@NIPAMTEST.COM for kadmin/hdp2.rtpppe.netapp.com@NIPAMTEST.COM
```

```
May 07 18:03:32 hdp2.rtpppe.netapp.com krb5kdc[31459](info): AS_REQ (8 etypes {18 17 20 19 16 23
25 26}) 10.63.150.176: ISSUE: authtime 1557266612, etypes {rep=18 tkt=18 ses=18},
admin/admin@NIPAMTEST.COM for kadmin/hdp2.rtpppe.netapp.com@NIPAMTEST.COM
May 07 18:03:34 hdp2.rtpppe.netapp.com krb5kdc[31459](info): AS_REQ (8 etypes {18 17 20 19 16 23
25 26}) 10.63.150.176: ISSUE: authtime 1557266614, etypes {rep=18 tkt=18 ses=18},
admin/admin@NIPAMTEST.COM for kadmin/hdp2.rtpppe.netapp.com@NIPAMTEST.COM
May 07 18:03:35 hdp2.rtpppe.netapp.com krb5kdc[31459](info): AS_REQ (8 etypes {18 17 20 19 16 23
25 26}) 10.63.150.176: ISSUE: authtime 1557266615, etypes {rep=18 tkt=18 ses=18},
admin/admin@NIPAMTEST.COM for kadmin/hdp2.rtpppe.netapp.com@NIPAMTEST.COM
May 07 18:03:37 hdp2.rtpppe.netapp.com krb5kdc[31459](info): AS_REQ (8 etypes {18 17 20 19 16 23
25 26}) 10.63.150.176: ISSUE: authtime 1557266617, etypes {rep=18 tkt=18 ses=18},
admin/admin@NIPAMTEST.COM for kadmin/hdp2.rtpppe.netapp.com@NIPAMTEST.COM
May 07 18:03:39 hdp2.rtpppe.netapp.com krb5kdc[31459](info): AS_REQ (8 etypes {18 17 16 23 20 19
25 26}) 10.63.150.177: ISSUE: authtime 1557266619, etypes {rep=18 tkt=18 ses=18}, nipamtesthdp-
050719@NIPAMTEST.COM for krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
May 07 18:03:40 hdp2.rtpppe.netapp.com krb5kdc[31459](info): AS_REQ (8 etypes {18 17 20 19 16 23
25 26}) 10.63.150.176: ISSUE: authtime 1557266620, etypes {rep=18 tkt=18 ses=18},
admin/admin@NIPAMTEST.COM for kadmin/hdp2.rtpppe.netapp.com@NIPAMTEST.COM
May 07 18:03:42 hdp2.rtpppe.netapp.com krb5kdc[31459](info): AS_REQ (8 etypes {18 17 20 19 16 23
25 26}) 10.63.150.176: ISSUE: authtime 1557266622, etypes {rep=18 tkt=18 ses=18},
admin/admin@NIPAMTEST.COM for kadmin/hdp2.rtpppe.netapp.com@NIPAMTEST.COM
```

6. If the login succeeds, then the Kerberos configuration continues.



7. Create the keytab file for the user.

```
[root@hdp2 ~]# su - tester
Last login: Thu Oct 18 16:41:53 EDT 2018 on pts/0
[tester@hdp2 ~]$ hadoop fs -ls /
19/04/26 14:42:17 WARN ipc.Client: Exception encountered while connecting to the server :
org.apache.hadoop.security.AccessControlException: Client cannot authenticate via:[TOKEN,
KERBEROS]
ls: DestHost:destPort hdp2.rtpppe.netapp.com:8020 , LocalHost:localPort
hdp2.rtpppe.netapp.com/10.63.150.176:0. Failed on local exception: java.io.IOException:
org.apache.hadoop.security.AccessControlException: Client cannot authenticate via:[TOKEN,
KERBEROS]
[root@hdp2 ~]# kinit root/admin@NIPAMTEST.COM
Password for root/admin@NIPAMTEST.COM:
kinit: Password incorrect while getting initial credentials
[root@hdp2 ~]# kinit root/admin@NIPAMTEST.COM
```

```
Password for root/admin@NIPAMTEST.COM:
[root@hdp2 ~]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: root/admin@NIPAMTEST.COM

Valid starting       Expires              Service principal
05/10/2019 12:23:03  05/11/2019 12:23:03  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
[root@hdp2 ~]# kadmin.local
Authenticating as principal root/admin@NIPAMTEST.COM with password.
kadmin.local:  add_principal tester@NIPAMTEST.COM
WARNING: no policy specified for tester@NIPAMTEST.COM; defaulting to no policy
Enter password for principal "tester@NIPAMTEST.COM":
Re-enter password for principal "tester@NIPAMTEST.COM":
Principal "tester@NIPAMTEST.COM" created.
kadmin.local:  xst -k /etc/security/keytabs/tester.keytab tester@NIPAMTEST.COM
Entry for principal tester@NIPAMTEST.COM with kvno 2, encryption type aes256-cts-hmac-sha1-96
added to keytab WRFILE:/etc/security/keytabs/tester.keytab.
Entry for principal tester@NIPAMTEST.COM with kvno 2, encryption type aes128-cts-hmac-sha1-96
added to keytab WRFILE:/etc/security/keytabs/tester.keytab.
Entry for principal tester@NIPAMTEST.COM with kvno 2, encryption type des3-cbc-sha1 added to
keytab WRFILE:/etc/security/keytabs/tester.keytab.
Entry for principal tester@NIPAMTEST.COM with kvno 2, encryption type arcfour-hmac added to
keytab WRFILE:/etc/security/keytabs/tester.keytab.
Entry for principal tester@NIPAMTEST.COM with kvno 2, encryption type camellia256-cts-cmac added
to keytab WRFILE:/etc/security/keytabs/tester.keytab.
Entry for principal tester@NIPAMTEST.COM with kvno 2, encryption type camellia128-cts-cmac added
to keytab WRFILE:/etc/security/keytabs/tester.keytab.
Entry for principal tester@NIPAMTEST.COM with kvno 2, encryption type des-hmac-sha1 added to
keytab WRFILE:/etc/security/keytabs/tester.keytab.
Entry for principal tester@NIPAMTEST.COM with kvno 2, encryption type des-cbc-md5 added to keytab
WRFILE:/etc/security/keytabs/tester.keytab.
kadmin.local:  [root@hdp2 ~]# ls -ltrh /etc/security/keytabs/tester.keytab
-rw------- 1 root root 522 May 10 12:25 /etc/security/keytabs/tester.keytab
[root@hdp2 ~]# id tester
uid=5000(tester) gid=5000(tester) groups=5000(tester)
[root@hdp2 ~]# chown tester:tester /etc/security/keytabs/tester.keytab
[root@hdp2 ~]# ls -ltrh /etc/security/keytabs/tester.keytab
-rw------- 1 tester tester 522 May 10 12:25 /etc/security/keytabs/tester.keytab
[root@hdp2 ~]#
```

8. Test the Kerberos kinit and Hadoop command.

```
[root@hdp2 ~]# su - tester
Last login: Fri May 10 12:17:54 EDT 2019 on pts/0
[tester@hdp2 ~]$ klist
klist: No credentials cache found (filename: /tmp/krb5cc_5000)
[tester@hdp2 ~]$ kinit tester@NIPAMTEST.COM -kt /etc/security/keytabs/tester.keytab
[tester@hdp2 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_5000
Default principal: tester@NIPAMTEST.COM

Valid starting       Expires              Service principal
05/10/2019 12:28:03  05/11/2019 12:28:03  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
[tester@hdp2 ~]$ hadoop fs -ls /
Found 9 items
drwxrwxrwt   - yarn   hadoop          0 2019-04-26 14:39 /app-logs
drwxr-xr-x   - yarn   hadoop          0 2019-01-14 17:31 /ats
drwxr-xr-x   - hdfs   hdfs            0 2019-01-14 17:31 /atsv2
drwxr-xr-x   - hdfs   hdfs            0 2019-01-14 17:31 /hdp
drwxr-xr-x   - mapred hdfs            0 2019-01-14 17:32 /mapred
drwxrwxrwx   - mapred hadoop          0 2019-01-14 17:33 /mr-history
drwxr-xr-x   - hdfs   hdfs            0 2019-01-14 17:31 /services
drwxrwxrwx   - hdfs   hdfs            0 2019-05-07 18:16 /tmp
drwxr-xr-x   - hdfs   hdfs            0 2019-04-26 14:48 /user
[tester@hdp2 ~]$
```

9. Test the manual NFS mount and access and review the Kerberos tickets. This step is not mandatory, but you should make sure that Kerberos is configured properly.


```
[root@hdp2 ~]# mount -o sec=krb5p hadoop_svm_nfs:/nc_volume2 /tmp/nc_volume2/
[root@hdp2 ~]# mount | grep krb5p
hadoop_svm_nfs:/nc_volume2 on /tmp/nc_volume2 type nfs4
(rw,relatime,vers=4.0,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,s
ec=krb5p,clientaddr=10.63.150.176,local_lock=none,addr=10.63.150.128)
[root@hdp2 ~]# su testeruser1
[testeruser1@hdp2 root]$ kinit -k testeruser1@NIPAMTEST.COM -t
/etc/security/keytabs/testeruser1.keytab
[testeruser1@hdp2 root]$ klist -e
Ticket cache: FILE:/tmp/krb5cc_10071
Default principal: testeruser1@NIPAMTEST.COM

Valid starting       Expires              Service principal
08/21/2019 15:07:33  08/22/2019 15:07:33  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
        Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96
[testeruser1@hdp2 root]$ cd /tmp/nc_volume2/
[testeruser1@hdp2 nc_volume2]$ klist -e
Ticket cache: FILE:/tmp/krb5cc_10071
Default principal: testeruser1@NIPAMTEST.COM

Valid starting       Expires              Service principal
08/21/2019 15:07:33  08/22/2019 15:07:33  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
        Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96
08/21/2019 15:07:48  08/22/2019 15:07:33  nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM
        Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96

[testeruser1@hdp2 nc_volume2]$ ll
total 84
drwxrwxrwx 2 root        root                4096 Feb 14  2018 data
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:30 f2
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:28 f3
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:30 f4
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:28 f5
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:28 f6
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:30 f7
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:30 f8
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:28 f9
drwxrwxrwx 3 testeruser1 hadoopkerberosgroup 4096 Aug 20 17:00 kermkdir
-rw-r--r-- 1 testeruser1 hadoopkerberosgroup    0 Aug 21 14:20 newfile
-rw-r--r-- 1 hdfs        hadoop              1145 Jul  3 19:28 README
-rw-r--r-- 1 root        root                2865 Feb 20  2018 READMEdemo
drwxrwxrwx 3 root        root                4096 Jun 28 13:39 scancopyFromLocal
drwxrwxrwx 3 root        root                4096 Jun 28 13:38 scantg
drwxr-xr-x 3 nobody      tester1             4096 Jul 15 14:40 sdd1
drwxrwxrwx 2 root        root                4096 Jul  9 11:14 test
drwxrwxrwx 3 root        root                4096 Jul 10 16:35 warehouse
drwxrwxrwx 3 root        root                4096 Feb 14  2018 wcresult
drwxrwxrwx 3 root        root                4096 Feb 14  2018 wcresult1
drwxrwxrwx 2 root        root                4096 Feb 14  2018 wcresult2
drwxrwxrwx 2 root        root                4096 Feb 16  2018 wcresult3

[testeruser1@hdp2 nc_volume2]$ cd /
[testeruser1@hdp2 /]$ exit
exit
[root@hdp2 ~]# umount /tmp/nc_volume2/
[root@hdp2 ~]#
```

10. Create the home folder for the user before running the MapReduce job.

```
[root@hdp2 ~]# su - hdfs
Last login: Fri Apr 26 14:41:47 EDT 2019 on pts/1
[hdfs@hdp2 ~]$ hadoop fs -mkdir /user/tester
[hdfs@hdp2 ~]$ hadoop fs -chown tester:tester /user/tester
[hdfs@hdp2 ~]$ id tester
uid=5000(tester) gid=5000(tester) groups=5000(tester)
[hdfs@hdp2 ~]$ logout
```

11. If you see the following error, complete the following steps:

```
org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.yarn.exceptions.YarnException):
org.apache.hadoop.security.AccessControlException: User tester does not have permission to submit
application_1556303968308_0003 to queue default
```

a. Enable `anyone` or `tester` in the Yarn queue manager from Ambari.

b. Restart the required Yarn services.

12. Run the MapReduce job in HDFS to make sure that Kerberos is working for the user.

```
[tester@hdp2 ~]$ hadoop  jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-
examples.jar pi 10 100
Number of Maps  = 10
Samples per Map = 100
Wrote input for Map #0
Wrote input for Map #1
Wrote input for Map #2
Wrote input for Map #3
Wrote input for Map #4
Wrote input for Map #5
Wrote input for Map #6
Wrote input for Map #7
Wrote input for Map #8
Wrote input for Map #9
Starting Job
19/05/10 15:20:32 INFO client.RMProxy: Connecting to ResourceManager at
hdp2.rtpppe.netapp.com/10.63.150.176:8050
19/05/10 15:20:32 INFO client.AHSProxy: Connecting to Application History server at
hdp3.rtpppe.netapp.com/10.63.150.177:10200
19/05/10 15:20:32 INFO hdfs.DFSClient: Created token for tester: HDFS_DELEGATION_TOKEN
owner=tester@NIPAMTEST.COM, renewer=yarn, realUser=, issueDate=1557516032176,
maxDate=1558120832176, sequenceNumber=22, masterKeyId=6 on 10.63.150.176:8020
19/05/10 15:20:32 INFO security.TokenCache: Got dt for hdfs://hdp2.rtpppe.netapp.com:8020; Kind:
HDFS_DELEGATION_TOKEN, Service: 10.63.150.176:8020, Ident: (token for tester:
HDFS_DELEGATION_TOKEN owner=tester@NIPAMTEST.COM, renewer=yarn, realUser=,
issueDate=1557516032176, maxDate=1558120832176, sequenceNumber=22, masterKeyId=6)
19/05/10 15:20:32 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/user/tester/.staging/job_1557515977944_0002
19/05/10 15:20:32 INFO input.FileInputFormat: Total input files to process : 10
19/05/10 15:20:32 INFO mapreduce.JobSubmitter: number of splits:10
19/05/10 15:20:32 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1557515977944_0002
19/05/10 15:20:32 INFO mapreduce.JobSubmitter: Executing with tokens: [Kind:
HDFS_DELEGATION_TOKEN, Service: 10.63.150.176:8020, Ident: (token for tester:
HDFS_DELEGATION_TOKEN owner=tester@NIPAMTEST.COM, renewer=yarn, realUser=,
issueDate=1557516032176, maxDate=1558120832176, sequenceNumber=22, masterKeyId=6)]
19/05/10 15:20:32 INFO conf.Configuration: found resource resource-types.xml at
file:/etc/hadoop/3.0.1.0-187/0/resource-types.xml
19/05/10 15:20:33 INFO impl.TimelineClientImpl: Timeline service address:
hdp3.rtpppe.netapp.com:8188
19/05/10 15:20:34 INFO impl.YarnClientImpl: Submitted application application_1557515977944_0002
19/05/10 15:20:34 INFO mapreduce.Job: The url to track the job:
http://hdp2.rtpppe.netapp.com:8088/proxy/application_1557515977944_0002/
19/05/10 15:20:34 INFO mapreduce.Job: Running job: job_1557515977944_0002
19/05/10 15:20:59 INFO mapreduce.Job: Job job_1557515977944_0002 running in uber mode : false
19/05/10 15:20:59 INFO mapreduce.Job:  map 0% reduce 0%
19/05/10 15:21:03 INFO mapreduce.Job:  map 20% reduce 0%
19/05/10 15:21:06 INFO mapreduce.Job:  map 50% reduce 0%
19/05/10 15:21:07 INFO mapreduce.Job:  map 70% reduce 0%
19/05/10 15:21:11 INFO mapreduce.Job:  map 100% reduce 0%
19/05/10 15:21:13 INFO mapreduce.Job:  map 100% reduce 100%
19/05/10 15:21:13 INFO mapreduce.Job: Job job_1557515977944_0002 completed successfully
19/05/10 15:21:13 INFO mapreduce.Job: Counters: 53
        File System Counters
                FILE: Number of bytes read=226
                FILE: Number of bytes written=2605669
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2780
                HDFS: Number of bytes written=215
                HDFS: Number of read operations=45
```



```
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
        Job Counters
                Launched map tasks=10
                Launched reduce tasks=1
                Data-local map tasks=10
                Total time spent by all maps in occupied slots (ms)=2803930
                Total time spent by all reduces in occupied slots (ms)=606776
                Total time spent by all map tasks (ms)=38410
                Total time spent by all reduce tasks (ms)=4156
                Total vcore-milliseconds taken by all map tasks=38410
                Total vcore-milliseconds taken by all reduce tasks=4156
                Total megabyte-milliseconds taken by all map tasks=2871224320
                Total megabyte-milliseconds taken by all reduce tasks=621338624
        Map-Reduce Framework
                Map input records=10
                Map output records=20
                Map output bytes=180
                Map output materialized bytes=280
                Input split bytes=1600
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=280
                Reduce input records=20
                Reduce output records=0
                Spilled Records=40
                Shuffled Maps =10
                Failed Shuffles=0
                Merged Map outputs=10
                GC time elapsed (ms)=688
                CPU time spent (ms)=16000
                Physical memory (bytes) snapshot=25270697984
                Virtual memory (bytes) snapshot=791027789824
                Total committed heap usage (bytes)=27849129984
                Peak Map Physical memory (bytes)=2504773632
                Peak Map Virtual memory (bytes)=65998901248
                Peak Reduce Physical memory (bytes)=410890240
                Peak Reduce Virtual memory (bytes)=131058761728
        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=1180
        File Output Format Counters
                Bytes Written=97
Job Finished in 41.883 seconds
Estimated value of Pi is 3.14800000000000000000
[tester@hdp2 ~]$
```

13. For testing, Create a keytab for the testeruser1 user.

```
[root@hdp2 ~]# kinit root/admin@NIPAMTEST.COM
Password for root/admin@NIPAMTEST.COM:
kinit: Password read interrupted while getting initial credentials
[root@hdp2 ~]# kinit root/admin@NIPAMTEST.COM
Password for root/admin@NIPAMTEST.COM:
[root@hdp2 ~]# kadmin.local
Authenticating as principal root/admin@NIPAMTEST.COM with password.
kadmin.local:  add_principal testeruser1@NIPAMTEST.COM
WARNING: no policy specified for testeruser1@NIPAMTEST.COM; defaulting to no policy
Enter password for principal "testeruser1@NIPAMTEST.COM":
Re-enter password for principal "testeruser1@NIPAMTEST.COM":
Principal "testeruser1@NIPAMTEST.COM" created.
kadmin.local:  xst -k /etc/security/keytabs/testeruser1.keytab testeruser1@NIPAMTEST.COM
Entry for principal testeruser1@NIPAMTEST.COM with kvno 2, encryption type aes256-cts-hmac-sha1-
96 added to keytab WRFILE:/etc/security/keytabs/testeruser1.keytab.
```



```
Entry for principal testeruser1@NIPAMTEST.COM with kvno 2, encryption type aes128-cts-hmac-sha1-
96 added to keytab WRFILE:/etc/security/keytabs/testeruser1.keytab.
Entry for principal testeruser1@NIPAMTEST.COM with kvno 2, encryption type des3-cbc-sha1 added to
keytab WRFILE:/etc/security/keytabs/testeruser1.keytab.
Entry for principal testeruser1@NIPAMTEST.COM with kvno 2, encryption type arcfour-hmac added to
keytab WRFILE:/etc/security/keytabs/testeruser1.keytab.
Entry for principal testeruser1@NIPAMTEST.COM with kvno 2, encryption type camellia256-cts-cmac
added to keytab WRFILE:/etc/security/keytabs/testeruser1.keytab.
Entry for principal testeruser1@NIPAMTEST.COM with kvno 2, encryption type camellia128-cts-cmac
added to keytab WRFILE:/etc/security/keytabs/testeruser1.keytab.
Entry for principal testeruser1@NIPAMTEST.COM with kvno 2, encryption type des-hmac-sha1 added to
keytab WRFILE:/etc/security/keytabs/testeruser1.keytab.
Entry for principal testeruser1@NIPAMTEST.COM with kvno 2, encryption type des-cbc-md5 added to
keytab WRFILE:/etc/security/keytabs/testeruser1.keytab.
kadmin.local:
kadmin.local:  exit
[root@hdp2 ~]#
```

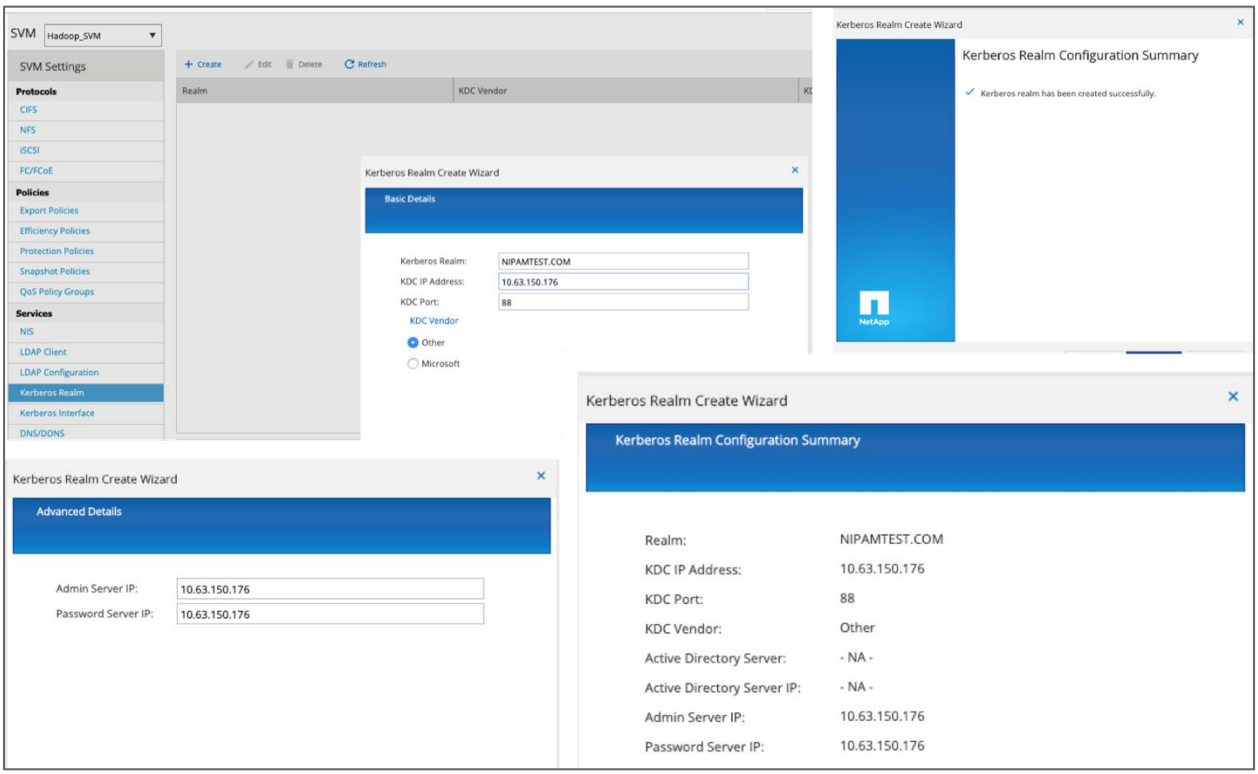
14. Create the same user for all Hadoop servers with the same UID and GUID. Alternatively, you can use use LDAP.

```
[root@hdp3 ~]# groupadd -g 10070 hadoopkerberosgroup
[root@hdp3 ~]# useradd -u 10071 -g 10070 testeruser1
[root@hdp3 ~]#
```

15. Copy the user keytab file to all Hadoop servers with the same permissions.

```
[root@hdp2 ~]# scp /etc/security/keytabs/testeruser1.keytab
hdp3:/etc/security/keytabs/testeruser1.keytab
testeruser1.keytab
100% 1122     1.6MB/s   00:00
[root@hdp2 ~]# ssh hdp3
[root@hdp3 ~]# id testeruser1
uid=10071(testeruser1) gid=10070(hadoopkerberosgroup) groups=10070(hadoopkerberosgroup)
[root@hdp3 ~]# chown testeruser1:hadoopkerberosgroup /etc/security/keytabs/testeruser1.keytab
[root@hdp3 ~]#
```

16. To perform storage configuration for the NetApp In-Place Analytics Module with Kerberos, complete the following steps:
    a. Configure the Kerberos realm.



b. Configure interfaces with Kerberos. In this configuration, we provide the admin username and password that are required the first time that ONTAP interacts with the KDC to create NFS service principals for the Kerberos operations.

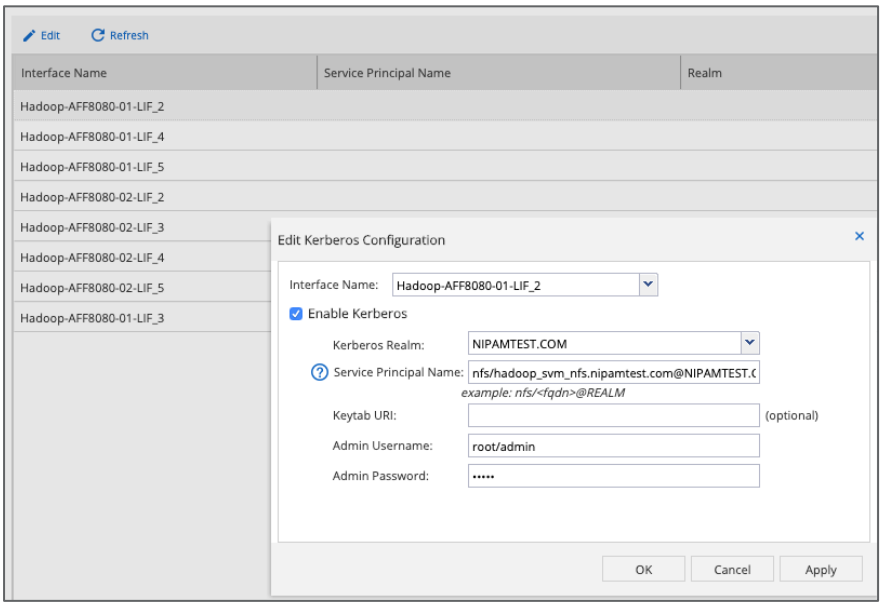
```
Hadoop-AFF8080::*> kerberos interface enable -vserver Hadoop_SVM -lif Hadoop-AFF8080-01-LIF_3 -
spn nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM -admin-username root/admin
  (vserver nfs kerberos interface enable)

Password:

Hadoop-AFF8080::*>
```

You do not need to provide the password for the following commands:

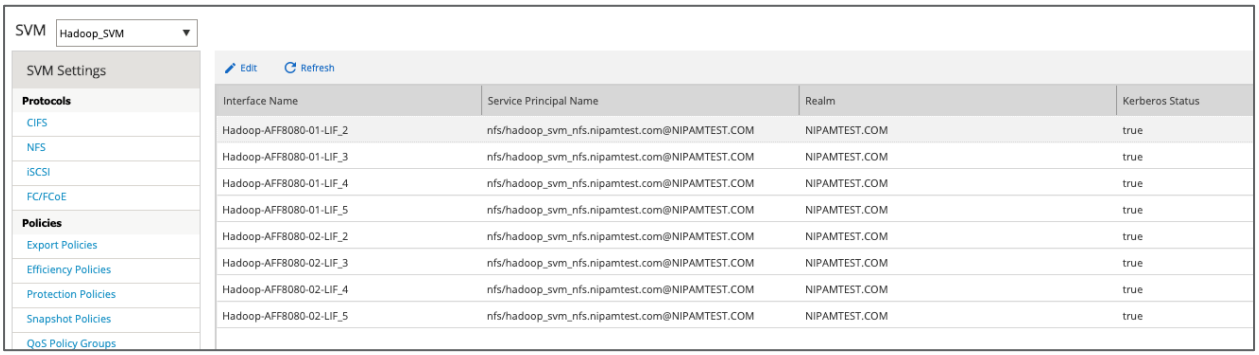
```
Hadoop-AFF8080::*> kerberos interface enable -vserver Hadoop_SVM -lif Hadoop-AFF8080-01-LIF_5 -
spn nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM -admin-username root/admin
  (vserver nfs kerberos interface enable)

Hadoop-AFF8080::*> kerberos interface enable -vserver Hadoop_SVM -lif Hadoop-AFF8080-01-LIF_4 -
spn nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM
  (vserver nfs kerberos interface enable)

Hadoop-AFF8080::*> kerberos interface enable -vserver Hadoop_SVM -lif Hadoop-AFF8080-02-LIF_2 -
spn nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM
  (vserver nfs kerberos interface enable)

Hadoop-AFF8080::*> kerberos interface enable -vserver Hadoop_SVM -lif Hadoop-AFF8080-02-LIF_3 -
spn nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM
  (vserver nfs kerberos interface enable)

Hadoop-AFF8080::*> kerberos interface enable -vserver Hadoop_SVM -lif Hadoop-AFF8080-02-LIF_4 -
spn nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM
  (vserver nfs kerberos interface enable)

Hadoop-AFF8080::*> kerberos interface enable -vserver Hadoop_SVM -lif Hadoop-AFF8080-02-LIF_5 -
spn nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM
  (vserver nfs kerberos interface enable)
```

```
Hadoop-AFF8080::*>
```

c. You can also use NetApp ONTAP System Manager to configure interfaces for Kerberos.



d. Check the status of the interfaces from ONTAP System Manager.



e. Create the user and group in the storage controller with same UID and GUID, or you can use LDAP. In the following output, the username called `host` is for krb-unix name mappings for the host SPN of the NFS client/Hadoop server. `nfs` is for krb-unix name mapping of the NFS service principal. The username `testuser1/hdfs` is for proper username/domain string authentication for NFSv4.x operations so that files get written for the proper users.

```
Hadoop-AFF8080::> unix-user show -vserver Hadoop_SVM
                  User            User   Group  Full
Vserver           Name            ID     ID     Name
----------------- --------------- ------ ------ --------------------------------
Hadoop_SVM        hdfs            5022   5005
Hadoop_SVM        hdp2            502    5005
Hadoop_SVM        host            503    5005
Hadoop_SVM        mapred          520    503
Hadoop_SVM        nfs             501    5005
Hadoop_SVM        nobody          65535  65535
Hadoop_SVM        pcuser          65534  65534
Hadoop_SVM        root            0      0
Hadoop_SVM        tester1         10039  10039
Hadoop_SVM        testeruser1     10071  10070
```

```
Hadoop_SVM     testeruser2     10072  10070
11 entries were displayed.

Hadoop-AFF8080::> unix-group  show -vserver Hadoop_SVM
Vserver        Name                ID
-------------- ------------------- ----------
Hadoop_SVM     daemon              1
Hadoop_SVM     hadoop              5005
Hadoop_SVM     hadoopkerberosgroup 10070
Hadoop_SVM     nobody              65535
Hadoop_SVM     pcuser              65534
Hadoop_SVM     root                0
Hadoop_SVM     tester1             10039
7 entries were displayed.

Hadoop-AFF8080::>
```

f. The following nfs-mapping.json file was used for testing.

```
[testeruser1@hdp2 ~]$ cat /etc/NetAppNFSConnector/conf/nfs-mapping.json
{
    "spaces": [
        {
            "endpoints": [
                {
                    "exportPath": "/nc_volume2",
                    "hosts": [
                            "nfs://10.63.150.213:2049/",
                             "nfs://10.63.150.211:2049/",
                             "nfs://10.63.150.123:2049/",
                             "nfs://10.63.150.124:2049/",
                             "nfs://10.63.150.125:2049/",
                             "nfs://10.63.150.126:2049/",
                             "nfs://10.63.150.127:2049/",
                             "nfs://10.63.150.128:2049/"
                    ],
                    "path": "/nc_volume2",
                      "nfsKerberosSpn": "nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM"
                },
                {
                    "exportPath": "/faiz_nfs_flexvol",
                    "hosts": [
                        "nfs://faiz_svm_nfs_lif1.example.com:2049/"
                    ],
                    "path": "/faiz_nfs_flexvol",
                    "nfsKerberosSpn": "nfs/faiz_svm_nfs_lif1.example.com@EXAMPLE.COM"
                }
            ],
            "name": "nowtest_scaning",
            "options": {
                "nfsAuthScheme1": "AUTH_SYS",
                "nfsDefaultKerberosSpn": "nfs/hadoop_svm_nfs.nipamtest.com@NIPAMTEST.COM",
                "nfsAuthScheme": "RPCSEC_GSS",
                "nfsGssServiceScheme": "RPCSEC_GSS_KRB5P",
                "nfsExportPath": "/nc_volume2",
                "nfsGid": 0,
                "nfsGroupname": "root",
                "nfsMountPort": -1,
                "nfsPort": 2049,
                "nfsReadMaxPoolThreads": 128,
                "nfsReadMinPoolThreads": 32,
                "nfsReadSizeBits": 20,
                "nfsRpcbindPort": 111,
                "nfsSplitSizeBits": 30,
                "nfsUid": 0,
                "nfsUserCacheTimeout": 15,
                "nfsUsername": "root",
                "nfsWriteMaxPoolThreads": 128,
                "nfsWriteMinPoolThreads": 32,
                "nfsWriteSizeBits": 20,
```



```
            "nfsIsRangerEnabled":false,
                 "nfsRangerConfig": {
                    "nfsRangerServiceName":"hdpntap_hadoop",
                    "nfsRangerAdminUrl":"http://10.63.150.161:6080/",
                    "nfsRangerAuditLogPath" : "/usr/nfs/ranger/logs",
                    "nfsRangerAuditEnabled" : true
                 }

             },
             "uri": "nfs://10.63.150.213:2049/"
        }
    ]
}
```

g. Set up Kinit for testeruser1.

```
[testeruser1@hdp2 ~]$ kinit testeruser1@NIPAMTEST.COM -kt
/etc/security/keytabs/testeruser1.keytab
[testeruser1@hdp2 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_10071
Default principal: testeruser1@NIPAMTEST.COM

Valid starting       Expires              Service principal
08/20/2019 16:43:21  08/21/2019 16:43:21  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
[testeruser1@hdp2 ~]$
```

h. Validate basic commands by testeruser1.

```
[testeruser1@hdp2 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/nc_volume2
Found 21 items
drwxrwxrwx   - root  root          4096 2019-08-20 16:05
nfs://10.63.150.213:2049/nc_volume2/.snapshot
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:28
nfs://10.63.150.213:2049/nc_volume2/README
-rw-r--r--   1 root  root          2865 2018-02-20 14:16
nfs://10.63.150.213:2049/nc_volume2/READMEdemo
drwxrwxrwx   - root  root          4096 2018-02-14 11:19 nfs://10.63.150.213:2049/nc_volume2/data
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:30 nfs://10.63.150.213:2049/nc_volume2/f2
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:28 nfs://10.63.150.213:2049/nc_volume2/f3
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:30 nfs://10.63.150.213:2049/nc_volume2/f4
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:28 nfs://10.63.150.213:2049/nc_volume2/f5
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:28 nfs://10.63.150.213:2049/nc_volume2/f6
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:30 nfs://10.63.150.213:2049/nc_volume2/f7
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:30 nfs://10.63.150.213:2049/nc_volume2/f8
-rw-r--r--   1 hdfs  hadoop        1145 2019-07-03 19:28 nfs://10.63.150.213:2049/nc_volume2/f9
drwxrwxrwx   - root  root          4096 2019-06-28 13:39
nfs://10.63.150.213:2049/nc_volume2/scancopyFromLocal
drwxrwxrwx   - root  root          4096 2019-06-28 13:38
nfs://10.63.150.213:2049/nc_volume2/scantg
drwxr-xr-x   - 10061 tester1       4096 2019-07-15 14:40 nfs://10.63.150.213:2049/nc_volume2/sdd1
drwxrwxrwx   - root  root          4096 2019-07-09 11:14 nfs://10.63.150.213:2049/nc_volume2/test
drwxrwxrwx   - root  root          4096 2019-07-10 16:35
nfs://10.63.150.213:2049/nc_volume2/warehouse
drwxrwxrwx   - root  root          4096 2018-02-14 13:56
nfs://10.63.150.213:2049/nc_volume2/wcresult
drwxrwxrwx   - root  root          4096 2018-02-14 14:53
nfs://10.63.150.213:2049/nc_volume2/wcresult1
drwxrwxrwx   - root  root          4096 2018-02-14 15:07
nfs://10.63.150.213:2049/nc_volume2/wcresult2
drwxrwxrwx   - root  root          4096 2018-02-16 10:24
nfs://10.63.150.213:2049/nc_volume2/wcresult3
[testeruser1@hdp2 ~]$ hadoop fs -mkdir nfs://10.63.150.213:2049/nc_volume2/kermkdir
[testeruser1@hdp2 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/nc_volume2
Found 22 items
drwxrwxrwx   - root        root                     4096 2019-08-20 16:05
nfs://10.63.150.213:2049/nc_volume2/.snapshot
-rw-r--r--   1 hdfs        hadoop                   1145 2019-07-03 19:28
nfs://10.63.150.213:2049/nc_volume2/README
-rw-r--r--   1 root        root                     2865 2018-02-20 14:16
nfs://10.63.150.213:2049/nc_volume2/READMEdemo
```



```
drwxrwxrwx   - root        root                       4096 2018-02-14 11:19
nfs://10.63.150.213:2049/nc_volume2/data
-rw-r--r--   1 hdfs        hadoop                     1145 2019-07-03 19:30
nfs://10.63.150.213:2049/nc_volume2/f2
-rw-r--r--   1 hdfs        hadoop                     1145 2019-07-03 19:28
nfs://10.63.150.213:2049/nc_volume2/f3
-rw-r--r--   1 hdfs        hadoop                     1145 2019-07-03 19:30
nfs://10.63.150.213:2049/nc_volume2/f4
-rw-r--r--   1 hdfs        hadoop                     1145 2019-07-03 19:28
nfs://10.63.150.213:2049/nc_volume2/f5
-rw-r--r--   1 hdfs        hadoop                     1145 2019-07-03 19:28
nfs://10.63.150.213:2049/nc_volume2/f6
-rw-r--r--   1 hdfs        hadoop                     1145 2019-07-03 19:30
nfs://10.63.150.213:2049/nc_volume2/f7
-rw-r--r--   1 hdfs        hadoop                     1145 2019-07-03 19:30
nfs://10.63.150.213:2049/nc_volume2/f8
-rw-r--r--   1 hdfs        hadoop                     1145 2019-07-03 19:28
nfs://10.63.150.213:2049/nc_volume2/f9
drwxrwxrwx   - testeruser1 hadoopkerberosgroup       4096 2019-08-20 16:43
nfs://10.63.150.213:2049/nc_volume2/kermkdir
drwxrwxrwx   - root        root                       4096 2019-06-28 13:39
nfs://10.63.150.213:2049/nc_volume2/scancopyFromLocal
drwxrwxrwx   - root        root                       4096 2019-06-28 13:38
nfs://10.63.150.213:2049/nc_volume2/scantg
drwxr-xr-x   - 10061       tester1                    4096 2019-07-15 14:40
nfs://10.63.150.213:2049/nc_volume2/sdd1
drwxrwxrwx   - root        root                       4096 2019-07-09 11:14
nfs://10.63.150.213:2049/nc_volume2/test
drwxrwxrwx   - root        root                       4096 2019-07-10 16:35
nfs://10.63.150.213:2049/nc_volume2/warehouse
drwxrwxrwx   - root        root                       4096 2018-02-14 13:56
nfs://10.63.150.213:2049/nc_volume2/wcresult
drwxrwxrwx   - root        root                       4096 2018-02-14 14:53
nfs://10.63.150.213:2049/nc_volume2/wcresult1
drwxrwxrwx   - root        root                       4096 2018-02-14 15:07
nfs://10.63.150.213:2049/nc_volume2/wcresult2
drwxrwxrwx   - root        root                       4096 2018-02-16 10:24
nfs://10.63.150.213:2049/nc_volume2/wcresult3
[testeruser1@hdp2 ~]$ hadoop fs -copyFromLocal /usr/share/doc/util-linux-2.23.2/README
nfs://10.63.150.213:2049/nc_volume2/kermkdir
```

i. This is a sample MapReduce job with the testeruser1 user.

```
[testeruser1@hdp2 ~]$ hadoop jar /usr/hdp/3.1.0.0-78/hadoop-mapreduce/hadoop-mapreduce-
examples.jar wordcount nfs://10.63.150.213:2049/nc_volume2/kermkdir/README
nfs://10.63.150.213:2049/nc_volume2/kermkdir/READMRE_result3
19/08/20 16:59:54 INFO client.RMProxy: Connecting to ResourceManager at
hdp2.rtpppe.netapp.com/10.63.150.176:8050
19/08/20 16:59:54 INFO client.AHSProxy: Connecting to Application History server at
hdp3.rtpppe.netapp.com/10.63.150.177:10200
19/08/20 16:59:55 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/user/testeruser1/.staging/job_1561743158956_0855
19/08/20 16:59:55 INFO input.FileInputFormat: Total input files to process : 1
19/08/20 16:59:55 INFO mapreduce.JobSubmitter: number of splits:1
19/08/20 16:59:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1561743158956_0855
19/08/20 16:59:55 INFO mapreduce.JobSubmitter: Executing with tokens: []
19/08/20 16:59:56 INFO conf.Configuration: found resource resource-types.xml at
file:/etc/hadoop/3.1.0.0-78/0/resource-types.xml
19/08/20 16:59:56 INFO impl.YarnClientImpl: Submitted application application_1561743158956_0855
19/08/20 16:59:56 INFO mapreduce.Job: The url to track the job:
http://hdp2.rtpppe.netapp.com:8088/proxy/application_1561743158956_0855/
19/08/20 16:59:56 INFO mapreduce.Job: Running job: job_1561743158956_0855
19/08/20 17:00:03 INFO mapreduce.Job: Job job_1561743158956_0855 running in uber mode : false
19/08/20 17:00:03 INFO mapreduce.Job:  map 0% reduce 0%
19/08/20 17:00:11 INFO mapreduce.Job:  map 100% reduce 0%
19/08/20 17:00:16 INFO mapreduce.Job:  map 100% reduce 100%
19/08/20 17:05:05 INFO mapreduce.Job: Job job_1561743158956_0855 completed successfully
19/08/20 17:05:05 INFO mapreduce.Job: Counters: 58
        File System Counters
                FILE: Number of bytes read=1339
                FILE: Number of bytes written=468719
```



```
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=116
                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)=453549
                Total time spent by all reduces in occupied slots (ms)=413764
                Total time spent by all map tasks (ms)=6213
                Total time spent by all reduce tasks (ms)=2834
                Total vcore-milliseconds taken by all map tasks=6213
                Total vcore-milliseconds taken by all reduce tasks=2834
                Total megabyte-milliseconds taken by all map tasks=464434176
                Total megabyte-milliseconds taken by all reduce tasks=423694336
        Map-Reduce Framework
                Map input records=47
                Map output records=83
                Map output bytes=1241
                Map output materialized bytes=1339
                Input split bytes=116
                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)=1476
                CPU time spent (ms)=38380
                Physical memory (bytes) snapshot=3186012160
                Virtual memory (bytes) snapshot=197225320448
                Total committed heap usage (bytes)=3860332544
                Peak Map Physical memory (bytes)=2640982016
                Peak Map Virtual memory (bytes)=66102480896
                Peak Reduce Physical memory (bytes)=545030144
                Peak Reduce Virtual memory (bytes)=131122839552
        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
[testeruser1@hdp2 ~]$
[testeruser1@hdp2 ~]$ hadoop fs -ls nfs://10.63.150.213:2049/nc_volume2/kermkdir/READMRE_result3
Found 2 items
-rw-r--r--   1 testeruser1 hadoopkerberosgroup          0 2019-08-20 17:00
nfs://10.63.150.213:2049/nc_volume2/kermkdir/READMRE_result3/_SUCCESS
-rw-r--r--   1 testeruser1 hadoopkerberosgroup       1025 2019-08-20 17:00
nfs://10.63.150.213:2049/nc_volume2/kermkdir/READMRE_result3/part-r-00000
[testeruser1@hdp2 ~]$ hadoop fs -cat
nfs://10.63.150.213:2049/nc_volume2/kermkdir/READMRE_result3/part-r-00000
"util-linux-ng".      1
```



```
(PO     1
2006-2010       1
<major>.<minor>-rc<N>  1
<major>.<minor>[.<maint>[.<bugfix>]]  1
=       4
CODE:   1
Checkout:       1
DOWNLOAD:       1
Development     1
E-MAIL: 1
LIST:   1
Linux   1
MAILING 1
NLS     1
Note    1
PO      1
SCHEMA: 1
SOURCE  1
Standard        1
TRANSLATIONS): 1
URL:    1
VERSION 1
Web     1
a       1
and     1
are     1
bug     1
bugfix  1
bugs    1
by:     1
changes 1
clone   1
collection      1
critical/security       1
deep    1
fatal   1
features        1
files   1
fixes   1
for     1
ftp://ftp.kernel.org/pub/linux/utils/util-linux/     1
git     1
git://git.kernel.org/pub/scm/utils/util-linux/util-linux.git        1
http://git.kernel.org/?p=utils/util-linux/util-linux.git    1
http://translationproject.org/domain/util-linux.html 1
http://vger.kernel.org/vger-lists.html#util-linux    1
https://github.com/karelzak/util-linux       1
in      1
interface:      1
is      1
maint   1
maintained      1
maintenance     1
major   1
minor   1
name    1
new     1
of      1
only    1
project 1
random  1
release 1
releases        1
releases:       2
releases;       1
that    1
the     1
this    1
typical 1
unplanned       1
used    1
```

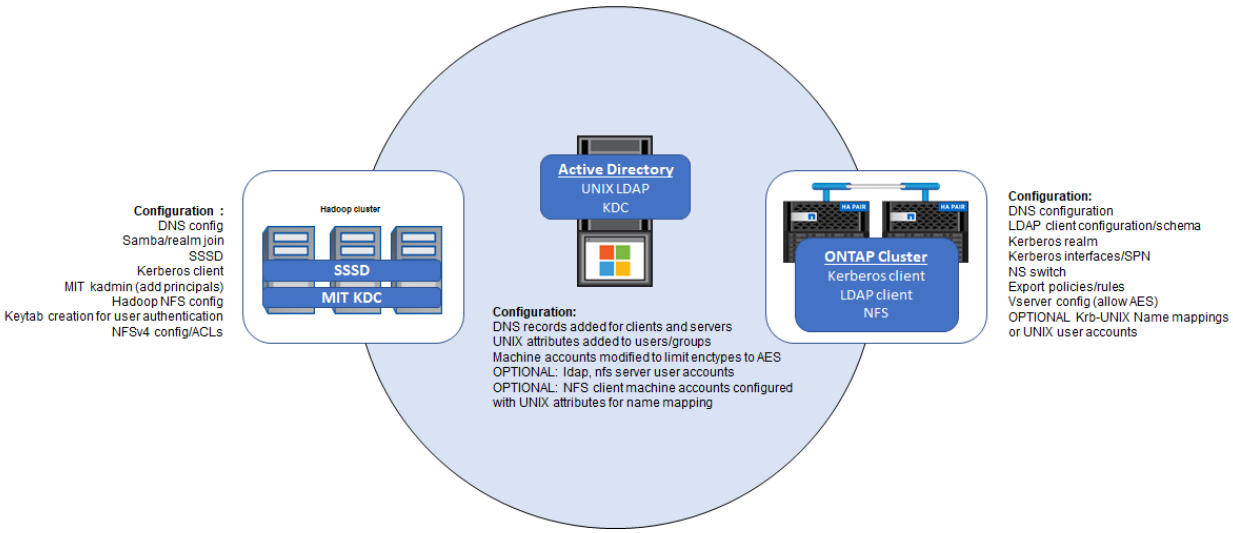


```
util-linux      3
util-linux@vger.kernel.org     1
utilities       1
with    1
years   1
[testeruser1@hdp2 ~]$
```

4 Kerberos with Active Directory and MIT

This document presents the steps needed to configure the environment, based on dependencies. Figure 2 depicts the general environment.

Figure 2) General environment.



4.1 Client/Hadoop Cluster Configuration – Kerberos Client: Active Directory

The Active Directory portion of Kerberos client connectivity is used for secure SSSD LDAP binds and lookups as well as for hosting the SPN of the NFS client. NFS mounts do not use the Active Directory KDC for Kerberos.

1. Configure DNS on the cluster to point to the desired DNS servers in Active Directory. DNS is needed to join the domain, as well as to find proper SPNs in the Active Directory KDC.
2. Install the Samba/realm packages.

```
yum -y install realmd sssd oddjob oddjob-mkhomedir adcli samba-common krb5-workstation ntp
```

3. Make sure that NTP is working and that the time is correct. The Kerberos time skew cannot exceed 5 minutes with the KDC. `pool.ntp.org` is a good NTP server to use.
4. Verify that the DNS points to the proper domains and name servers. In our case, we're using two search domains to allow for KDCs in Active Directory and MIT.

```
sh-4.2$ cat /etc/resolv.conf
# Generated by NetworkManager
search poctest.com
search nipamtest.com
nameserver 10.63.150.86
nameserver 10.61.173.186
nameserver 10.63.162.5
```

5. Make sure that the firewall allows the proper connectivity. See the Red Hat page on the proper use of firewalls.

6. Test discovery of the AD realm using the `realm discover domain.com`.
7. Join the domain using `realm join DOMAIN.COM`.
8. Enable secure NFS with the following commands:

```
systemctl enable nfs-client.target &&
systemctl start nfs-client.target
```

9. Test Kerberos ticket/login with `kinit administrator@DOMAIN.COM`.
10. Modify the `/etc/krb5.conf` file to make sure that only AES is used as well as for defining additional Kerberos realms (such as MIT).

```
[libdefaults]
default_realm = NIPAMTEST.COM
dns_lookup_kdc = false
dns_lookup_realm = false
ticket_lifetime = 86400
renew_lifetime = 604800
forwardable = true
default_tgs_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96
default_tkt_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96
permitted_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96
udp_preference_limit = 1
kdc_timeout = 3000
default_realm=NIPAMTEST.COM
[realms]
NIPAMTEST.COM = {
kdc = 10.63.150.85
admin_server = 10.63.150.85
default_domain = .nipamtest.com

}
POCTEST.COM = {
kdc = 10.63.150.86
admin_server = 10.63.150.86
}
[domain_realm]
.nipamtest.com = NIPAMTEST.COM
nipamtest.com = NIPAMTEST.COM
.poctest.com = POCTEST.COM
poctest.com = POCTEST.COM
```

11. After modifying `krb5.conf`, restart the service `service rpcgssd restart`.

4.2 Active Directory Configuration – LDAP

Active Directory has UNIX identity management/LDAP functionality support built in, but you need to configure users and groups that have valid UID/GID information. Active Directory uses the RFC2307bis standard for LDAP. For Kerberos, ONTAP performs name mappings (Kerberos-unix) for initial authentication for the NFS service SPN (defined as `nfs/fqdn@DOMAIN.COM` during Kerberos configuration in ONTAP) and the NFS client SPNs (generally `MACHINEFQDN$@DOMAIN.COM`). Natively, ONTAP can map these SPNs to UNIX users if UNIX users with the same name can be found either in local files (passwd) or in LDAP. For central management and access, the preferred method is adding a new user called `nfs`, defining its UNIX username as `nfs`, and populating the UID/GID information.

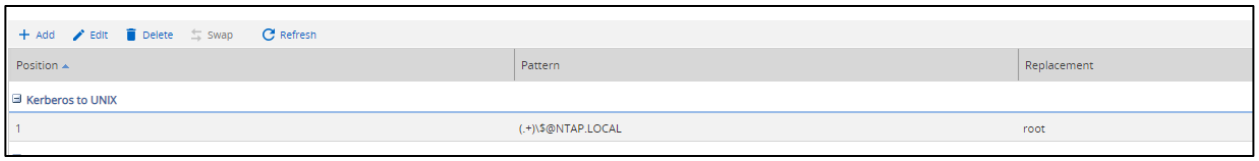
For machine accounts, we have the following options:

1. Modify the existing machine accounts created by the `realm join` command to populate the uid, uidnumber, and gidnumber fields so that the ONTAP LDAP client can find the necessary information to map the users.

-or-

2. Create a name-mapping rule in ONTAP to map all machine accounts requesting a Kerberos-Unix mapping to the same UNIX user (such as `nfs`); this mapping does not provide file access. Instead, it

allows Kerberos authentication to complete. This can be done in System Manager at Storage > SVMs > SVM Settings > Name Mapping.



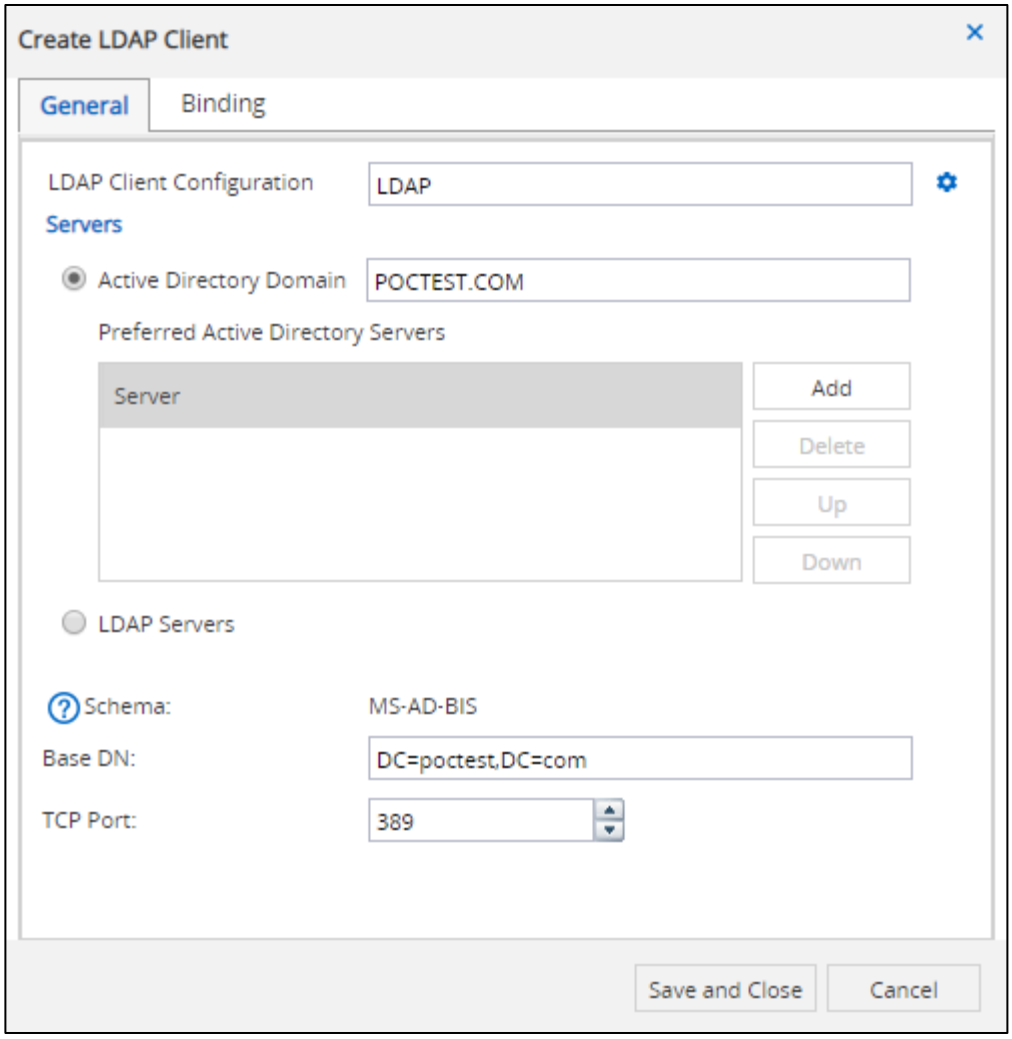
-or-

3. Create local UNIX users and groups with the same names as the machine accounts in the SVM at Storage > SVMs > SVM Settings > Host Users and Groups.

4.3 ONTAP Cluster Configuration – LDAP Client

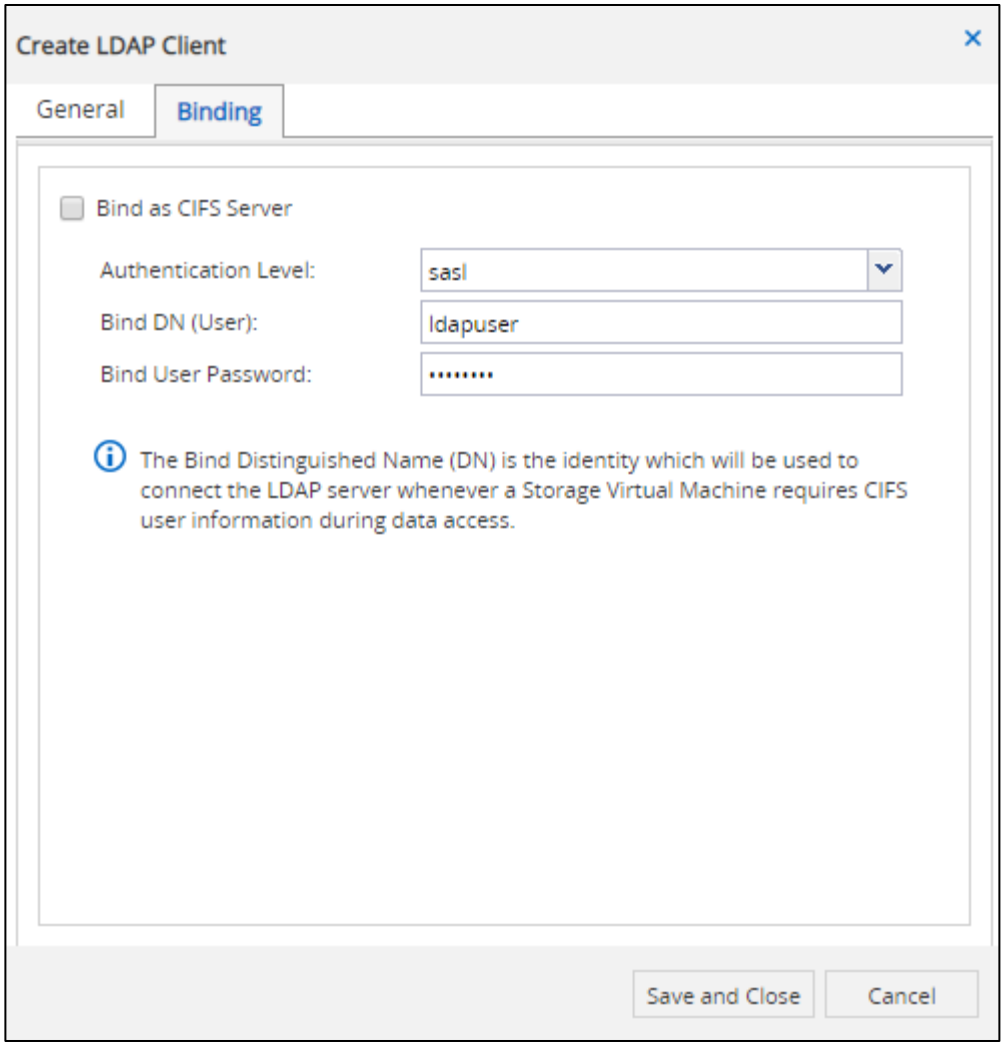
ONTAP storage virtual machines are able to act as LDAP clients to LDAP servers. This allows ONTAP to share the same identity management resources as NFS clients to ensure that all UIDs/GIDs are the same across deployments. To set up an SVM as an LDAP client, complete the following steps:

1. Determine the LDAP schema in your environment. In Active Directory, generally speaking, MS-AD-BIS is suitable. But for older or legacy AD schemas, you might need to choose a different schema or create a custom schema. This depends on the attributes that have been populated with UNIX identity information. Try MS-AD-BIS first if you are unsure, and, if necessary, reach out to parisi@netapp.com to help figure out which schema will be used.
2. Create the LDAP Client. This can be done in System Manager at Storage > SVMs > SVM Settings > LDAP Client. Available options include the following:
   - **LDAP client configuration.** The name of the config.
   - **Schema.** This defaults to MS-AD-BIS, but it can be changed using the gear icon.
   - **AD domain.** This is your domain name.
   - **Preferred AD servers.** Leaving this blank allows ONTAP to use all AD servers in the forest, regardless of their locations.
   - **LDAP servers.** You do not configure this when you configure the AD domain.
   - **Base DN.** This is the place where the LDAP client starts the search; leaving it blank defaults to the top-level domain DN.
   - **TCP port.**



In addition, you can configure how you bind (login) to the LDAP Server for searches. If you have an existing CIFS/SMB server, you can select Bind as CIFS Server to use the CIFS server's credentials to bind. Otherwise, binding depends on the LDAP server configuration. Generally speaking, anonymous binds in AD LDAP are disallowed. SASL is more secure, but, if you want fully encrypted binds (LDAP over SSL or LDAP signing and sealing), you must use the CLI. The bind user can be any domain user; they all have read access to LDAP. Creating an LDAP user as a service account might make the most sense here.

1. Create the LDAP configuration (same menu as the LDAP client). This creates the LDAP configuration for use with the SVM and specifies which client configuration are used with the SVM.
2. Configure `ns-switch` to use LDAP and files for `passwd` and `group`.



3. To test LDAP lookups, use `getxxbyyy` in the CLI in advanced privileges.

```
::> set advanced

::*> getxxbyyy
    getaddrinfo   getgrbygid    getgrbyname   getgrlist     gethostbyaddr
    gethostbyname getnameinfo   getpwbyname   getpwbyuid    netgrpcheck

::*> getxxbyyy getpwbyname -node [node] -vserver [svm] -username [name] -show-source true -show-
granular-err true
```

See the following example of working commands:

```
Local user:

ontap9-tme-8040::*> getxxbyyy getpwbyname -node ontap9-tme-8040-01 -vserver DEMO -username nfs -
show-source true
  (vserver services name-service getxxbyyy getpwbyname)
Source used for lookup: Files
pw_name: nfs
pw_passwd: *
pw_uid: 500
pw_gid: 500
pw_gecos:
pw_dir:
```

```
pw_shell:


User in LDAP:


ontap9-tme-8040::*> getxxbyyy getpwbyname -node ontap9-tme-8040-01 -vserver DEMO -username prof1
-show-source true
  (vserver services name-service getxxbyyy getpwbyname)
Source used for lookup: LDAP
pw_name: prof1
pw_passwd:
pw_uid: 1100
pw_gid: 1101
pw_gecos:
pw_dir:
pw_shell:
```

4.4 Client/Hadoop Cluster Configuration – LDAP Client: SSSD

Active Directory serve as our UNIX identity management server. We must point SSSD to the server and configure SSSD to use it for LDAP. Realm join (above) configures SSSD to use Active Directory authentication that creates UIDs based on SIDs. We configure this client to do standard UIDs for LDAP instead.

SSSD uses the client's SPN in the Active Directory KDC to bind (login) to perform GSSAPI encrypted LDAP queries.

To configure SSSD, complete the following steps:

1. Stop the SSSD service with `service sssd stop`.
2. Modify the `/etc/sssd/sssd.conf` file to look like this:

```
[sssd]
domains = poctest.com
config_file_version = 2
services = nss, pam

[domain/poctest.com]
auth_provider = krb5
chpass_provider = krb5
id_provider = ldap
cache_credentials = False
ldap_uri = ldap://10.63.150.86
ldap_search_base = dc=poctest,dc=com
ldap_schema = rfc2307bis
ldap_sasl_mech = GSSAPI
ldap_user_object_class = user
ldap_group_object_class = group
ldap_user_home_directory = unixHomeDirectory
ldap_user_principal = userPrincipalName
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
ldap_sasl_authid = STLRX2540M1-36$@POCTEST.COM
krb5_server = poctest.com
krb5_realm = POCTEST.COM
krb5_kpasswd = poctest.com
use_fully_qualified_names = false
```

3. Clear the SSSD cache with `rm -rf /var/lib/sss/db/*`.
4. Start the SSSD service with `service sssd start`.
5. Wait for the service to start up and come alive (a minute or so).
6. Test LDAP lookups with `id username`.

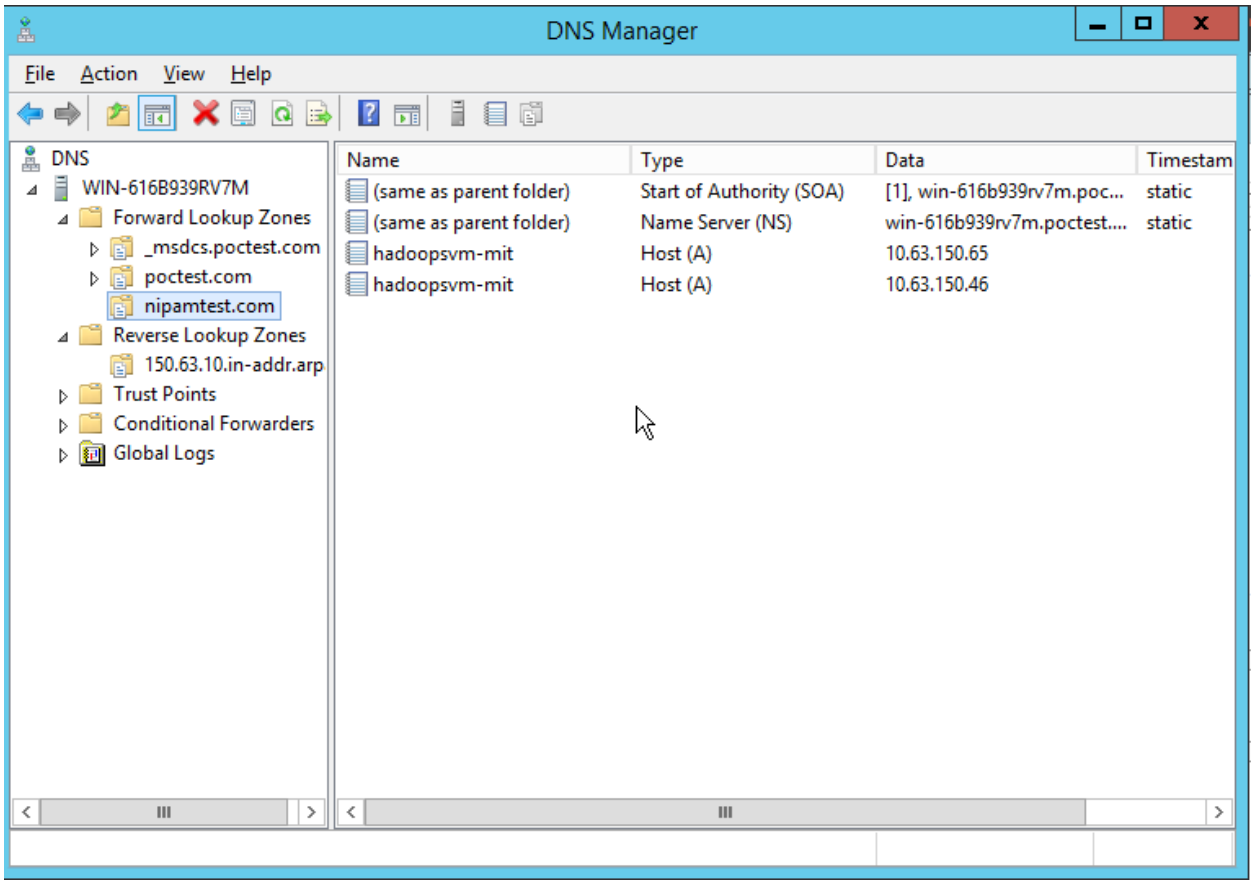
```
# id tester1
```



```
uid=10000(tester1) gid=10000(hadoopkerberosgroup) groups=10000(hadoopkerberosgroup)
```

4.5 Active Directory Configuration – DNS

You must verify that the client is using a proper SPN query for the NFS server. To do so, create a new primary zone in DNS called `nipamtest.com` that holds the A records for the data LIFs participating in the NFS Kerberos setup. Otherwise, you might see the client attempting to query the `POCTEST.COM` realm due to the `krb5.conf` configuration needed for SSSD.



On the NFS client, verify that nslookup works for the new hostname.

```
sh-4.2$ nslookup hadoopsvm-mit.nipamtest.com
Server:         10.63.150.86
Address:        10.63.150.86#53

Name:   hadoopsvm-mit.nipamtest.com
Address: 10.63.150.46
Name:   hadoopsvm-mit.nipamtest.com
Address: 10.63.150.65
```

4.6 ONTAP Cluster Configuration – Kerberos

When configuring ONTAP to interact with Kerberos KDCs, the commands create NFS service principals and transfer keytabs to the cluster SVM for use with Kerberized mounts. ONTAP supports the following ciphers for Kerberos authentication:

- DES
- 3DES

- AES-128 and AES-256

Active Directory still uses RC4-HMAC in Kerberos authentication, and ONTAP does not support RC4 (also referred to as ARCFOUR), so client attempts to use RC4 fail with the following error in the event logs (using `event log show –messagename *secd*`).

```
FAILURE: Failed to accept the context: Unspecified GSS failure.  Minor code may provide more
information (minor: Encryption type ArcFour with HMAC/md5 not permitted).
```

ONTAP provides a way to permit or disallow enctypes with the following NFS server option:

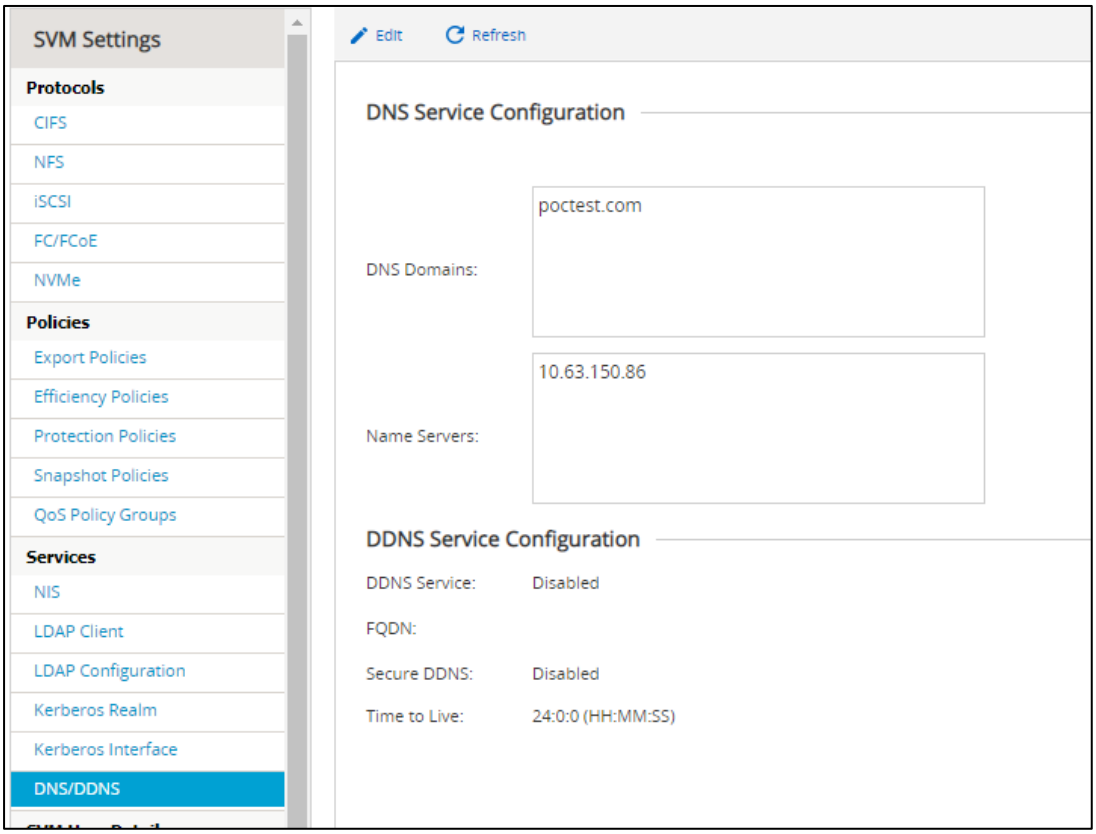
```
stlaff300-1and2::*> nfs server show -fields permitted-enc-types
vserver   permitted-enc-types
--------- ----------------------
hadoop-fg des,des3,aes-128,aes-256
hadoopsvm des,des3,aes-128,aes-256
2 entries were displayed.
```

AES is the preferred enctype, due to its superior security over DES and 3DES. To set AES only, run the following command before creating the Kerberos config:

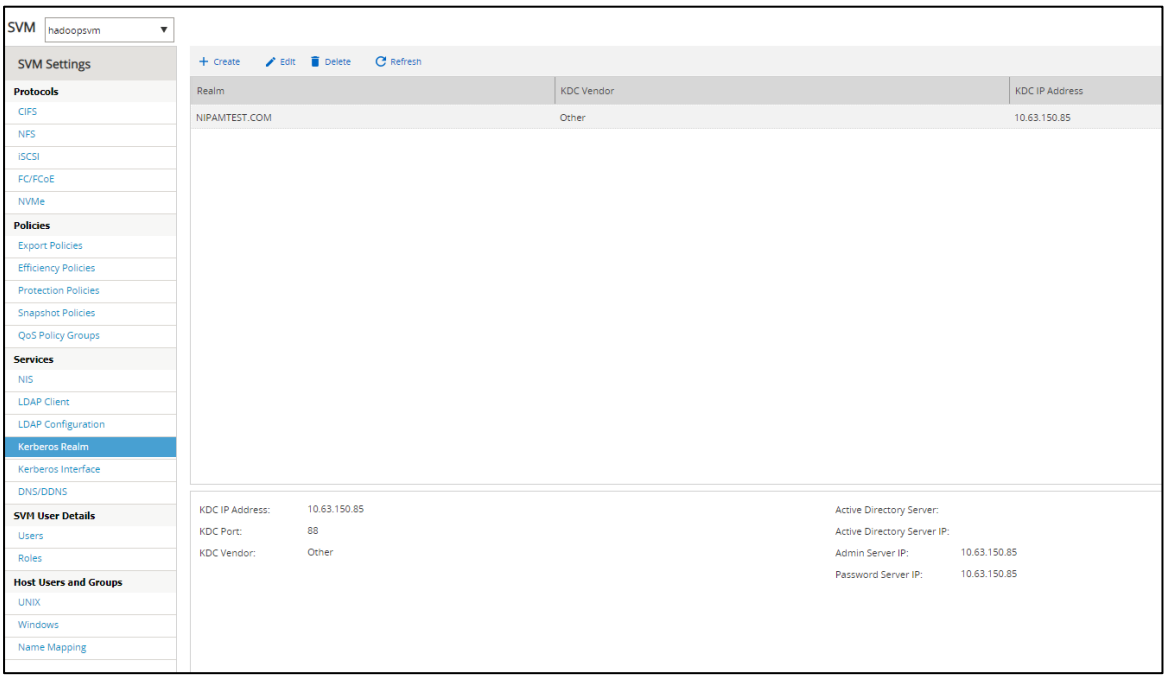
```
::> nfs server modify -vserver hadoopsvm -permitted-enc-types aes*
```

For cluster SVM Kerberos configuration, complete the following steps:

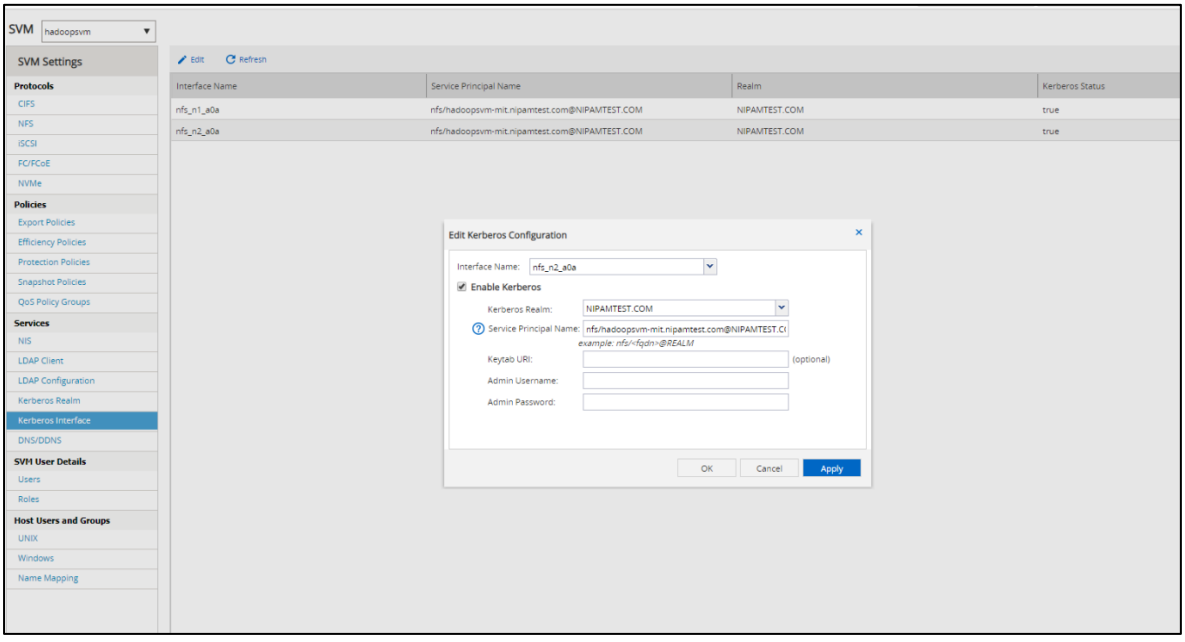
1. Configure DNS for the SVM.



2. Create the Kerberos realm for the MIT KDC.

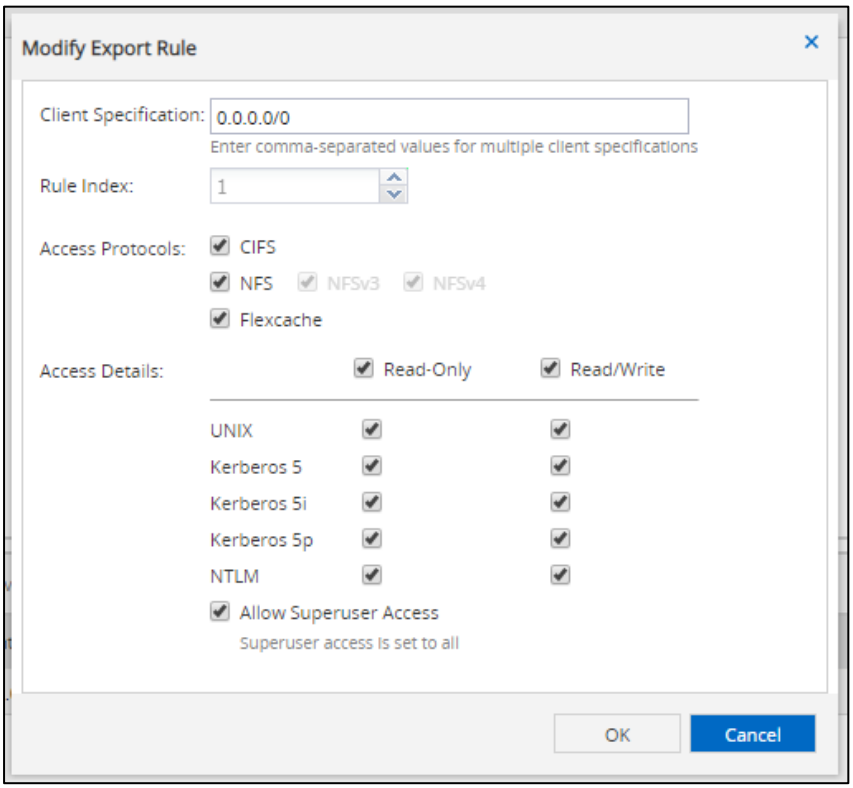


3. Create the Kerberos interface configuration on one or more data LIFs in the SVM. This interacts with the KDC and creates the SPN for the NFS server. The SPN should start with `nfs/` and include the FQDN that exists in the DNS server configuration specified (in our case, `nipamtest.com`). The SPN also ends with the realm in ALL CAPS. You do not need to specify a keytab; instead, supply the username and password for a Kerberos principal with the permissions necessary to create and delete principals (addprincs) in the MIT KDC. If you specify more than one data LIF for Kerberos, use A records (with PTRs) in DNS to provide proper name lookups.



4. Make sure that the export policy rule allows krb5* for ro, rw, and superuser access.

Note: krb5i and krb5p are more secure versions of krb5. Krb5p encrypts all traffic over the wire, but it also affects performance. The Kerberos flavor is defined at mount from the client. Allowing krb5i and krb5p lets clients choose those options.



5. If you didn't create an LDAP user for the NFS server and client, then create local UNIX users or name mapping rules as described in TR-4616.

- Test the export policy access for Kerberos in the CLI.

```
stlaff300-1and2::> export-policy check-access -vserver hadoopsvm -volume pocfg -client-ip
10.63.150.85 -authentication-method krb5 -protocol nfs3 -access-type read-write
                                        Policy    Policy       Rule
Path                            Policy  Owner     Owner Type  Index Access
------------------------------- ------- --------- ---------- ------ ----------
/                               default   mongodb_root
                                                  volume          1 read
/pocfg                          default   pocfg     volume          1 read-write
2 entries were displayed.
```

4.7 Windows AD KDC Configuration – Kerberos Machine Accounts

To prevent machine accounts from trying to use enctypes that are not supported (such as RC4), you must adjust the machine accounts to only use the AES enctype. You can do this with PowerShell.

For the NFS clients (the RHEL servers), run the following command:

```
PS C:\> Set-ADComputer CLIENT$ -KerberosEncryptionType AES256,AES128
```

4.8 Generate Keytab Files for Passwordless Access to Hadoop NFS Connector

The following command allows you to create keytab files for users to use in Hadoop NFS connector operations:

```
[root@stlrx2540m1-36 /]# kadmin.local
Authenticating as principal tester3/admin@NIPAMTEST.COM with password.
kadmin.local:  xst -k tester3-mit.keytab tester3@NIPAMTEST.COM
Entry for principal tester3@NIPAMTEST.COM with kvno 2, encryption type aes256-cts-hmac-sha1-96
added to keytab WRFILE:tester3-mit.keytab.
Entry for principal tester3@NIPAMTEST.COM with kvno 2, encryption type aes128-cts-hmac-sha1-96
added to keytab WRFILE:tester3-mit.keytab.
Entry for principal tester3@NIPAMTEST.COM with kvno 2, encryption type des3-cbc-sha1 added to
keytab WRFILE:tester3-mit.keytab.
Entry for principal tester3@NIPAMTEST.COM with kvno 2, encryption type arcfour-hmac added to
keytab WRFILE:tester3-mit.keytab.
Entry for principal tester3@NIPAMTEST.COM with kvno 2, encryption type camellia256-cts-cmac added
to keytab WRFILE:tester3-mit.keytab.
Entry for principal tester3@NIPAMTEST.COM with kvno 2, encryption type camellia128-cts-cmac added
to keytab WRFILE:tester3-mit.keytab.
Entry for principal tester3@NIPAMTEST.COM with kvno 2, encryption type des-hmac-sha1 added to
keytab WRFILE:tester3-mit.keytab.
Entry for principal tester3@NIPAMTEST.COM with kvno 2, encryption type des-cbc-md5 added to
keytab WRFILE:tester3-mit.keytab.
```

After you create the keytab for a user, you can pass it in kinit commands.

```
[root@stlrx2540m1-36 /]# kinit tester3@NIPAMTEST.COM -k -t tester3-mit.keytab
[root@stlrx2540m1-36 /]# klist -e
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: tester3@NIPAMTEST.COM

Valid starting       Expires              Service principal
06/06/2019 14:12:32  06/07/2019 14:12:32  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
        renew until 06/13/2019 14:12:32, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
```

4.9 Test the NFSv3 Kerberos Mount

Hadoop NFS connector can mount with Kerberos only if the NFSv3 client can mount with Kerberos. So, before you try Hadoop, try mounting with regular NFSv3 and troubleshoot any issues.

Root performs the mount and does not need to kinit. The mount uses the client's SPN to initiate the Kerberos mount. The following process happens:

1. The client initiates the mount command with `-o sec=krb5`.
2. ONTAP checks the export policy rules to ensure that a) the client IP has access to mount and b) the client IP has access to mount with krb5.
3. If the export policy rule check fails, we get permission or access denied for the mount.
4. The client interacts with AD/ONTAP to use the `MACHINE$@DOMAIN.COM` SPN for Kerberos tickets.
5. ONTAP attempts to perform a krb-unix mapping of `MACHINE$@DOMAIN.COM` to a UNIX user. This can be an LDAP user, a local user named `MACHINE$`, or a name-mapping rule.
6. If the mappings succeed, check the Kerberos encryption type of the ticket submitted by the client to make sure that ONTAP offers support for the enctype. If you don't, the mount fails with access or permission denied.

   On the cluster, you would get an event log message which can be viewed with `event log show -messages *secd*`; these messages offer details about why the error occurred.
7. If the enctype is supported, allow the mount to proceed. See the following example of a working mount:

```
[root@stlrx2540m1-36 /]# mount -o sec=krb5 10.63.150.65:/pocvol /pocvol
[root@stlrx2540m1-36 /]#
```

8. After the mount is established, to access it with a user, you must issue a kinit to get a Kerberos ticket to grant access to the mount. If you do not issue a kinit, then accessing a Kerberos mount fails like this:


```
[root@stlrx2540m1-36 /]# su tester1
sh-4.2$ id
uid=10000(tester1) gid=10000(hadoopkerberosgroup) groups=10000(hadoopkerberosgroup)
sh-4.2$ klist
klist: No credentials cache found (filename: /tmp/krb5cc_10000)
sh-4.2$ cd /pocvol
sh: cd: /pocvol: Not a directory
```

`Not a directory` means that you can't traverse the mount point to look for directory handles because there is no Kerberos ticket allowing you access. Trying to "ls" the directory results in denied permission.

```
sh-4.2$ ls -la /pocvol
ls: cannot access /pocvol: Permission denied
```

When you kinit to get a valid Kerberos ticket, you get access. The following process happens on an attempt to cd into a Kerberized NFS mount with a user that has a valid ticket:

1. The ticket is checked with the KDC to verify that it is valid and unexpired.
2. If it is valid, it is used to request an NFS service ticket from the KDC.
3. The NFS service ticket must perform a krb-unix mapping for the NFS server SPN (`nfs/fqdn.domain.com@DOMAIN.COM`) in ONTAP. This is a UNIX user named `nfs` that can exist in LDAP locally, or it can be a name-mapping rule (see the previous steps in this document or see TR-4616).

   If mapping fails for either SPN, then the cd fails on the client with the errors `access denied`, `permission denied`, or `not a directory`.

   If mapping succeeds, then the NFS service ticket is issued to the client and access is allowed (depending on the export policy rule and file/folder permissions).

```
sh-4.2$ kinit tester1@NIPAMTEST.COM -k -t /etc/security/keytabs/tester1-mit.keytab
sh-4.2$ klist -e
Ticket cache: FILE:/tmp/krb5cc_10000
Default principal: tester1@NIPAMTEST.COM

Valid starting       Expires              Service principal
06/11/2019 16:52:49  06/12/2019 16:52:49  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
        renew until 06/18/2019 16:52:49, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
sh-4.2$ cd /mnt
sh-4.2$ klist -e
Ticket cache: FILE:/tmp/krb5cc_10000
Default principal: tester1@NIPAMTEST.COM

Valid starting       Expires              Service principal
06/11/2019 16:52:49  06/12/2019 16:52:49  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
        renew until 06/18/2019 16:52:49, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
06/11/2019 16:53:00  06/12/2019 16:52:49  nfs/hadoopsvm-mit.nipamtest.com@NIPAMTEST.COM
        renew until 06/18/2019 16:52:49, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
```

4. Verify if files can be written to the mount and if they write as the proper user.

```
sh-4.2$ touch newfile_tester1
sh-4.2$ ls -la | grep newfile_tester1
-rw-r--r--   1 tester1 hadoopkerberosgroup    0 Jun 11 16:54 newfile_tester1
```

If the NFSv3 mount with Kerberos and subsequent navigation of the Kerberized mount succeeds, then Hadoop mounts with Kerberos should also succeed.

4.10 Hadoop NFS Connector Config File

The `nfs-mapping.json` file must be configured to provide Kerberos information and namespace information. The `nfs-mapping.json` file is as follows:


```
# cat /etc/NetAppNFSConnector/conf/nfs-mapping.json
{
"spaces": [{
"name": "santanderpoc1",
"uri": "nfs://hadoopsvm-mit.nipamtest.com:2049/",
"options": {
        "nfsDefaultKerberosSpn": "nfs/hadoopsvm-mit.nipamtest.com@NIPAMTEST.COM",
        "nfsAuthScheme": "RPCSEC_GSS",
        "nfsGssServiceScheme": "RPCSEC_GSS_KRB5P",
        "nfsAuthScheme1": "AUTH_SYS",
        "nfsExportPath": "/pocfg",
        "nfsIsRangerEnabled": false,
                                "nipamLogComponents":false,
                                "nfsIsStatsEnabled":false,
   "nfsRangerConfig": {
                    "nfsRangerAdminUrl": "http://master2.netapp.com:6080",
                    "nfsRangerAuditEnabled": false,
                    "nfsRangerAuditLogPath": "/usr/nfs/ranger/logs",
                    "nfsRangerServiceName": "hbase_nfspath"
                },

"nfsMountPort": -1,
"nfsPort": 2049,
"nfsRpcbindPort": 111,
"nfsReadSizeBits": 20,
"nfsWriteSizeBits": 20,
"nfsReadMinPoolThreads":64,
"nfsReadMaxPoolThreads":128,
"nfsWriteMinPoolThreads":64,
"nfsWriteMaxPoolThreads":128,
"nfsReceiveTCPBufferSizeBytes": 1048576,
"nfsSendTCPBufferSizeBytes": 1048576,
"nfsSplitSizeBits": 30
      },

      "endpoints": [
      {
        "hosts": [
                        "nfs://hadoopsvm-mit.nipamtest.com:2049/"
                ],
        "path": "/pocfg",
        "exportPath": "/pocfg",
        "nfsKerberosSpn": "nfs/hadoopsvm-mit.nipamtest.com@NIPAMTEST.COM"
      },
      {
        "hosts": [
                        "nfs://hadoopsvm-mit.nipamtest.com:2049/"
                ],
        "path": "/pocvol",
        "exportPath": "/pocvol",
        "nfsKerberosSpn": "nfs/hadoopsvm-mit.nipamtest.com@NIPAMTEST.COM"
      }
      ]
   }]
}
```

4.11 Test the NetApp In-Place Analytics Module(NIPAM)

After the config file is ready, test the NIPAM LDAP/Kerberos solution by changing to an LDAP user and issuing the kinit command using the generated keytab file:

1. Su as the user.

```
[root@stlrx2540m1-36 ~]# su tester1
sh-4.2$ id
uid=10000(tester1) gid=10000(hadoopkerberosgroup) groups=10000(hadoopkerberosgroup)
```

2. Kinit with the generated keytab file.

```
sh-4.2$ klist -e
```

```
klist: No credentials cache found (filename: /tmp/krb5cc_10000)
sh-4.2$ kinit tester1@NIPAMTEST.COM -k -t /etc/security/keytabs/tester1-mit.keytab
sh-4.2$ klist -e
Ticket cache: FILE:/tmp/krb5cc_10000
Default principal: tester1@NIPAMTEST.COM

Valid starting       Expires              Service principal
06/11/2019 17:01:33  06/12/2019 17:01:33  krbtgt/NIPAMTEST.COM@NIPAMTEST.COM
        renew until 06/18/2019 17:01:33, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-
hmac-sha1-96
```

3. Run the Hadoop command.

```
sh-4.2$ hadoop fs -ls nfs://hadoopsvm-mit.nipamtest.com:2049/pocvol
Found 9 items
drwxrwxrwx   - root    root                       4096 2019-06-11 16:05 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/.snapshot
-rw-r--r--   1 tester1 hadoopkerberosgroup          0 2019-06-11 16:54 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/newfile_tester1
drwxr-x---   - tester1 hadoopkerberosgroup       4096 2019-06-11 09:44 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/tester1_folder
drwxr-xr-x   - tester1 hadoopkerberosgroup       4096 2019-06-11 09:37 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/testuser1_folder2_jagnya
drwx------   - tester1 hadoopkerberosgroup       4096 2019-06-11 09:53 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/testuser1_folder3_jagnya
drwx------   - tester1 hadoopkerberosgroup       4096 2019-06-11 09:59 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/testuser1_folder4_jagnya_justin
drwxrwx---   - tester1 hadoopkerberosgroup       4096 2019-06-11 09:35 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/testuser1_folder_jagnya
drwxrwx---   - tester1 hadoopkerberosgroup       4096 2019-06-10 18:11 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/testuser1folder1
drwxrwx---   - tester1 hadoopkerberosgroup       4096 2019-06-10 19:22 nfs://hadoopsvm-
mit.nipamtest.com:2049/pocvol/testuser1folder1_noacl4tester2
```

If you run the command using any IP or hostname other than what you defined in the config file, the command fails, even if the IP addresses are valid.

```
sh-4.2$ hadoop fs -ls nfs://10.63.150.46:2049/pocvol
19/06/11 17:02:54 WARN topology.Mapping: NfsConnectorV3.1 No namespace available for
uri:nfs://10.63.150.46:2049/pocvol
ls: `nfs://10.63.150.46:2049/pocvol': No such file or directory
```

4.12 NFSv4 ACLs on NFSv3 Objects

NFSv3 does not have native standard ACLs. POSIX ACLs were added by some third-party development, but ONTAP does not support POSIX ACLs. NFSv3 permissions are basic – Read, Write, and Execute for Owner, Group, and Everyone. There is no native way in NFSv3 to assign ACLs to multiple users and groups.

As a result, to get granular ACLs, NFSv4 ACLs must be used. Fortunately, ONTAP supports adding NFSv4 ACLs to objects that are accessed via NFSv3 and it honors those ACLs. This provides a way to get granular permissions on NFSv3 objects.

You must provide the following prerequisites:

- NFSv4 configuration (ID domain on client and server)
- NFSv4 ACL support enabled in ONTAP
- NFSv4 ACL preserve enabled in ONTAP
- NFSv4 support on clients
- NetApp FlexVol® volumes (there is no current NFSv4.x support for FlexGroup volumes)
- NFSv4 control-plane mount to assign ACLs
- NFSv3 data-plane mount access for Hadoop

This blog post describes the process.

After the steps are completed, you can assign various ACLs to objects for different user permissions. As an example, we have three users: tester1, tester2, and tester3.

For the folder tester1_folder, we have the following ACLs:

```
[root@stlrx2540m1-36 mnt]# ls -la | grep tester1_folder
drwxr-x---   6 tester1 hadoopkerberosgroup 4096 Jun 11 09:44 tester1_folder

[root@stlrx2540m1-36 mnt]# nfs4_getfacl tester1_folder
# file: tester1_folder
A::tester2@NIPAMTEST.COM:rwax
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:tcy
```

Therefore, the owner (tester1) has full control, tester2 has read-write access, and tester3 has read-only access (by way of the GROUP ACL).

As a result, tester1 can write and chmod inside tester1_folder.

```
sh-4.2$ pwd
/mnt/tester1_folder
sh-4.2$ id
uid=10000(tester1) gid=10000(hadoopkerberosgroup) groups=10000(hadoopkerberosgroup)
sh-4.2$ touch tester1_file
sh-4.2$ ls -la | grep tester1_file
-rw-r--r-- 1 tester1 hadoopkerberosgroup     0 Jun 11 17:13 tester1_file
sh-4.2$ chmod 777 tester1_file
sh-4.2$ ls -la | grep tester1_file
-rwxrwxrwx 1 tester1 hadoopkerberosgroup     0 Jun 11 17:13 tester1_file
```

Tester2 can read and write.

```
[root@stlrx2540m1-36 mnt]# su tester2
sh-4.2$ cd /mnt/tester1_folder/
sh-4.2$ touch tester2_newfile
sh-4.2$ ls -la | grep tester2_newfile
-rw-r--r-- 1 tester2 hadoopkerberosgroup2    0 Jun 11 17:15 tester2_newfile
sh-4.2$ chmod 777 tester2_newfile
```

Tester3 can only read.

```
[root@stlrx2540m1-36 mnt]# su tester3
sh-4.2$ ls -la /mnt/tester1_folder/
total 24
drwxr-x--- 6 tester1 hadoopkerberosgroup  4096 Jun 11 17:15 .
drwxrwxrwx 9 root    root                 4096 Jun 11 16:54 ..
drwxr-xr-x 2 root    root                 4096 Jun 10 13:08 data
-rw-r--r-- 1 root    root                    0 Jun 10 16:08 file
-rwxrwxrwx 1 tester1 hadoopkerberosgroup     0 Jun 11 17:13 tester1_file
-rw-r--r-- 1 tester1 hadoopkerberosgroup     0 Jun 10 16:17 tester1file_acltest
drwx------ 3 tester1 hadoopkerberosgroup  4096 Jun 10 16:39 tester1folder_acltest
-rw-r--r-- 1 tester2 hadoopkerberosgroup2    0 Jun 11 09:44 tester2_file
-rw-r--r-- 1 tester2 hadoopkerberosgroup2    0 Jun 10 16:15 tester2file_afteracl
drwxr-xr-x 2 root    root                 4096 Jun 10 16:09 tester2folder
drwxr-xr-x 2 tester2 hadoopkerberosgroup2 4096 Jun 10 16:36 tester2folder1
-rwxrwxrwx 1 tester2 hadoopkerberosgroup2    0 Jun 11 17:15 tester2_newfile
sh-4.2$ touch /mnt/tester1_folder/tester3_newfile
touch: cannot touch ‘/mnt/tester1_folder/tester3_newfile’: Permission denied
```

Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- More detailed information about NFS Kerberos and LDAP can be found in
  https://www.netapp.com/us/media/tr-4616.pdf
  https://www.netapp.com/us/media/tr-4073.pdf
- NetApp In-Place Analytics Module best practice
  https://www.netapp.com/us/media/tr-4382.pdf
- For information about applying NFSv4 ACLS for use with NFSv3, see:
  https://whyistheinternetbroken.wordpress.com/2017/08/11/nfsv4acls-nfsv3mounts

Version History

Version	Date	Document Version History
Version 1.0	September 2019	Initial release

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