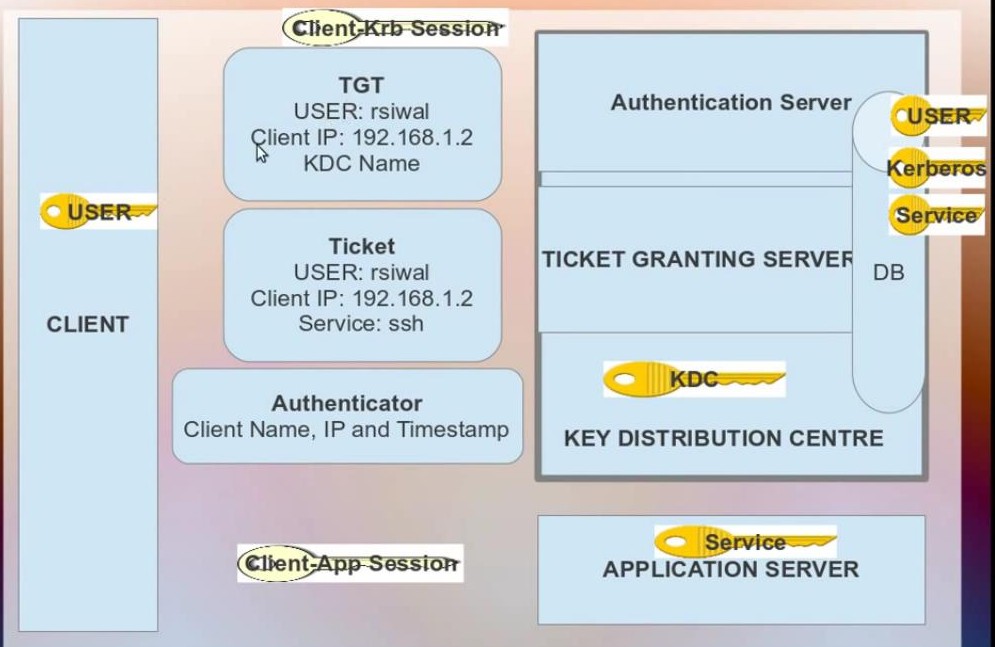
**Kerberos**

- Users have to authenticate themselves before using any service in Hadoop.

- Hosts & Services also have to identify each other for security reasons. Else some malicious service will act as reliable component of cluster to gain access.

Hadoop uses Kerberos as the basis for strong authentication and identity propagation for both user and services. Kerberos is a third party authentication mechanism, in which users and services rely on a third party - the Kerberos server to authenticate each to the other.

**User ←----- Kerberos -----→ service**



**AIM:**

i. Credentials doesn’t flow through network. - Kerberos

-> flow through network – LDAP/ with algorithm encrypted format.

ii. user pswd should never be stored in network/system.

iii. User pswd should not be kept un-encrypted format in kerberos db/authentic db.

iv. SSO -

**Terms:-**

Kerberos DB will store, User key, KDC key & Application server key.

KDC – Authentication Server

TGS →

Kerberos uses symmetric key algorithm.

User tries to login, user receives TGT for authentication. Username → Request goes to KDC (Authentication server) & it checks user’s identity across DB. With the key from DB, it generates TGT (username+ip+KDC name) is encrypted with its private key & shares to user.

KDC creates a session key, TGT encrypted with private key & sends details to the client. Now, when the user enters its password, it gets decrypted & accepts the session key. Then session is open.

While User requests the service access, Authenticator (client name, ip & timestamp) + Encrypted with session key will be created.

Application server will create a session key (user-service)

**User ←-------Authenticator------- -------Session Key-------→ Application/Service**

**Setting up Kerberos:**

- install KDC, **→** then install kerberos

* Install Kerberos using the below command
  + $ yum install krb5-server krb5-libs krb5-auth-dialog krb5-workstation
* Take a back up of krb5.conf file and modify as per your server details. In this demonstration the REALM = **HDP.COM**. The kdc and admin server = **master.hdp.com**. These values change as per your environment.
* Use the utility kdb5\_util to create the Kerberos database. Provide a password when prompted.

$ kdb5\_util create -s

* Start the KDC server and the KDC admin server.

$ Service krb5kdc start

$ service kadmin start

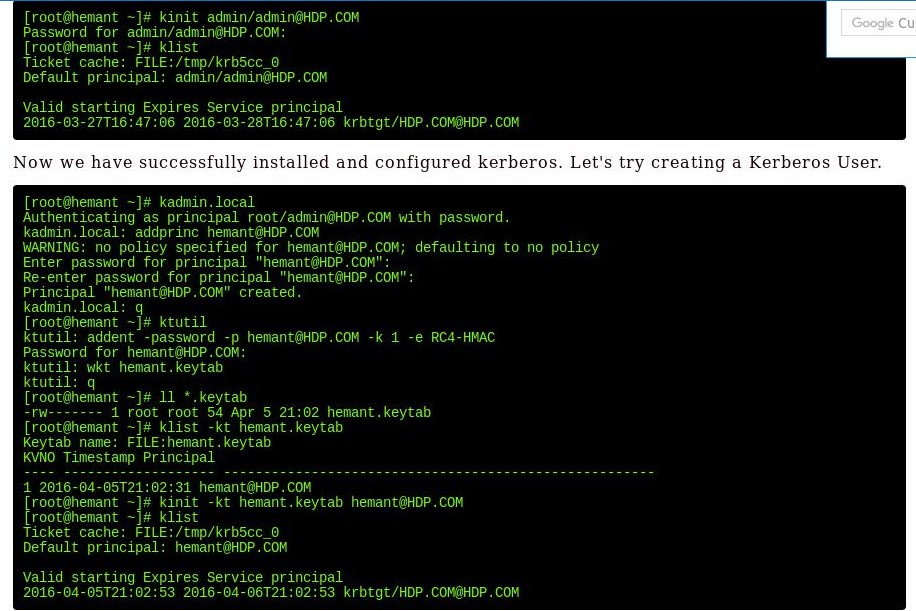
* Create Kerberos Admin:

$ kadmin.local -q "addprinc admin/admin"

* + Provide permissions to admin to administer the KDC for your specific realm. Modify /var/kerberos/krb5kdc/kadm5.acl

\*/admin@HDP.COM \*

* + - * + $ kinit [admin/admin@HDP.COM](mailto:admin/admin@HDP.COM)
        + $ klist
        + Now try to do a kinit as admin.







**Create DB for Kerberos.**

$ Yum install yum-utils => provides a pckg – repoquery

$ repoquery -f \*/kdb5-util

$ kdb5-util create -s -r LABS.LOCAL

<master – password>

$ vi /etc/kerberso/krb5kdc/kadm5.acl

$ vi /etc/kerberso/krb5kdc/kdc.conf

$ /etc/init.d/krb5kdc restart

$ chkconfig krb5kdc on

$ kadmin.local (k<user> to login to kdc db)

kadmin.local : listcp

kadmin.local : listprincs

kadmin.local : addprinc [root/admin@LABS.LOCAL](mailto:root/admin@LABS.LOCAL)

**Ranger: Centralized security, auditing,**

**Daily Hadoop Admin Activities**

<https://www.edureka.co/blog/hadoop-admin-responsibilities/>

**i) add/delete user spaces →**

\* User space in Hadoop is, similar to having users in linux.

\* hdfs dfs -ls /user

/user/hive

/user/hbase

/user/root

\* for each ddepartments, we create user space.

$ su – root

$ useradd etl\_user -g hadoop

$ id etl\_user

$ sudo su – etl\_user

$ hdfs dfs -mkdir -p /user/etl\_user; hdfs dfs -chown etl\_user:hadoop /user/etl\_user

\* Name quota:- control number of files.

\* Space quota:- controls amount of data stored & takes replication into account.

$ hdfs dfsadmin -setQuota 1000 /user/etl\_user (Name)

$ hdfs dfsadmin -clrQuota (Name)

$ hdfs dfsadmin -setSpaceQuota 10G /user/etl\_user

$ hdfs dfsadmin -clrSpaceQuota

$ hdfs dfs -count -q /user/etl\_user → shows the quota count. How many files left to be created.

**2. data migration from 1 hadoop cluster to another**

\* it will use all resources. So better to do it during the GreenZone.

\*

$ hadoop distcp hdfs://ip-192.142.100.02:8080/user/etl\_user <target\_cluster\_directory>

**-overwrite**

**-update**

**-delete**

**3. Taking data backup**

- Script for Database (hive metastore, oozie-db, ranger-db, ambari-db) backup.

- HDFS metadata Backup

Hdfs dfsadmin –fetchImage fsimage.backup.ddmmyyyy

* Cleaning up older log files.

**4. add/remove services**

- upgrading software

- add/remove host from cluster.

- troubleshooting issues

- changing/overriding parameters @ runtime

- Phinix query server ….

- updating parameters

- setting up alarms.

- Clean up activity:

- Archival

- Balancer **<hdfs balancer –threshold 1 >**

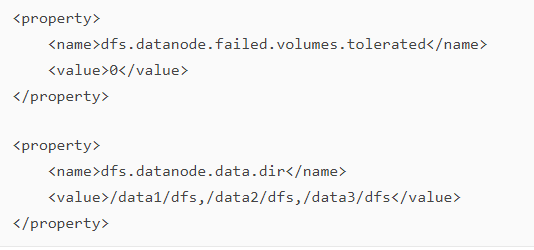
- Regular job scheduling/monitoring the failure/troubleshoot

- Datanode Volume Failure:

Multiple disk failures on single Datanode could cause shutdown of Datanode daemon.

**Hdfs-site.xml** 🡪 dfs.datanode.failed.volumes.tolerated

**Root Cause:**



**FIX and Workaround:**

1. replace failed hardware
2. restore data volume using OS utilities to recreate the file system and mount.
3. mkdir /data2/dfs
4. chown hdfs:hadoop /data2/dfs
5. service hadoop-hdfs-datanode start

<https://discuss.pivotal.io/hc/en-us/articles/202400998-DataNode-is-shutting-down-reported-due-to-DataNode-failed-volumes->

* Verify dfs is healthy with "sudo -u hdfs hdfs dfsadmin -report"

- Refresh Nodes after config changes:

**<hdfs dfsadmin –refreshNodes>**

**< sudo -u yarn yarn rmadmin -refreshNodes>**

**Activities:**

<http://www.hadoopadmin.co.in/hadoop-administration-and-maintenance/>

**Fine Tuning:**

<https://community.hortonworks.com/articles/131670/ambari-server-performance-tuning-troubleshooting-c.html>

**Ambari UI is slow:-**

* No of hosts added to ambari cluster –
* No of concurrent users/user views created –
* How old the cluster is – operational logs & alert history would be consuming space.
* Ambari DB & Ambari Server regions – N/W issue.
* Ambari heap –
* Resource availability in Ambari server.

# free -m

# top

# lsof -p $AMBARI\_PID

# netstat -tnlpa | grep $AMBARI\_PID

* Collect Ambari thread dumps to see any stuck/blocked threads—

**# su -l hdfs -c "$JAVA\_HOME/bin/jcmd -l"**

1. 5752 org.apache.hadoop.hdfs.server.namenode.NameNode
2. 5546 org.apache.hadoop.hdfs.tools.DFSZKFailoverController
3. 5340 org.apache.hadoop.hdfs.server.datanode.DataNode
4. 4991 org.apache.hadoop.hdfs.qjournal.server.JournalNode

* Ambari heap statistics

$**JAVA\_HOME/bin/jmap -heap $AMBARI\_SERVER\_PID**

## **Hadoop Admin Responsibilities:**

* Responsible for implementation and ongoing administration of Hadoop infrastructure.
* Aligning with the systems engineering team to propose and deploy new hardware and software environments required for Hadoop and to expand existing environments.
* Working with data delivery teams to setup new Hadoop users. This job includes setting up Linux users, setting up Kerberos principals and testing HDFS, Hive, Pig and Map Reduce access for the new users.
* Cluster maintenance as well as creation and removal of nodes using tools like Ganglia, Nagios, Cloudera Manager Enterprise, Dell Open Manage and other tools.
* Performance tuning of Hadoop clusters and Hadoop Map Reduce routines.
* Screen Hadoop cluster job performances and capacity planning
* Monitor Hadoop cluster connectivity and security
* Manage and review Hadoop log files.
* File system management and monitoring.
* HDFS support and maintenance.
* Diligently teaming with the infrastructure, network, database, application and business intelligence teams to guarantee high data quality and availability.
* Collaborating with application teams to install operating system and Hadoop updates, patches, version upgrades when required.
* Point of Contact for Vendor escalation

b. My Project Architecture.

c. transition from Tibco to Hadoop : Reaosn,

d. nifi – Hortonworks DataFlow