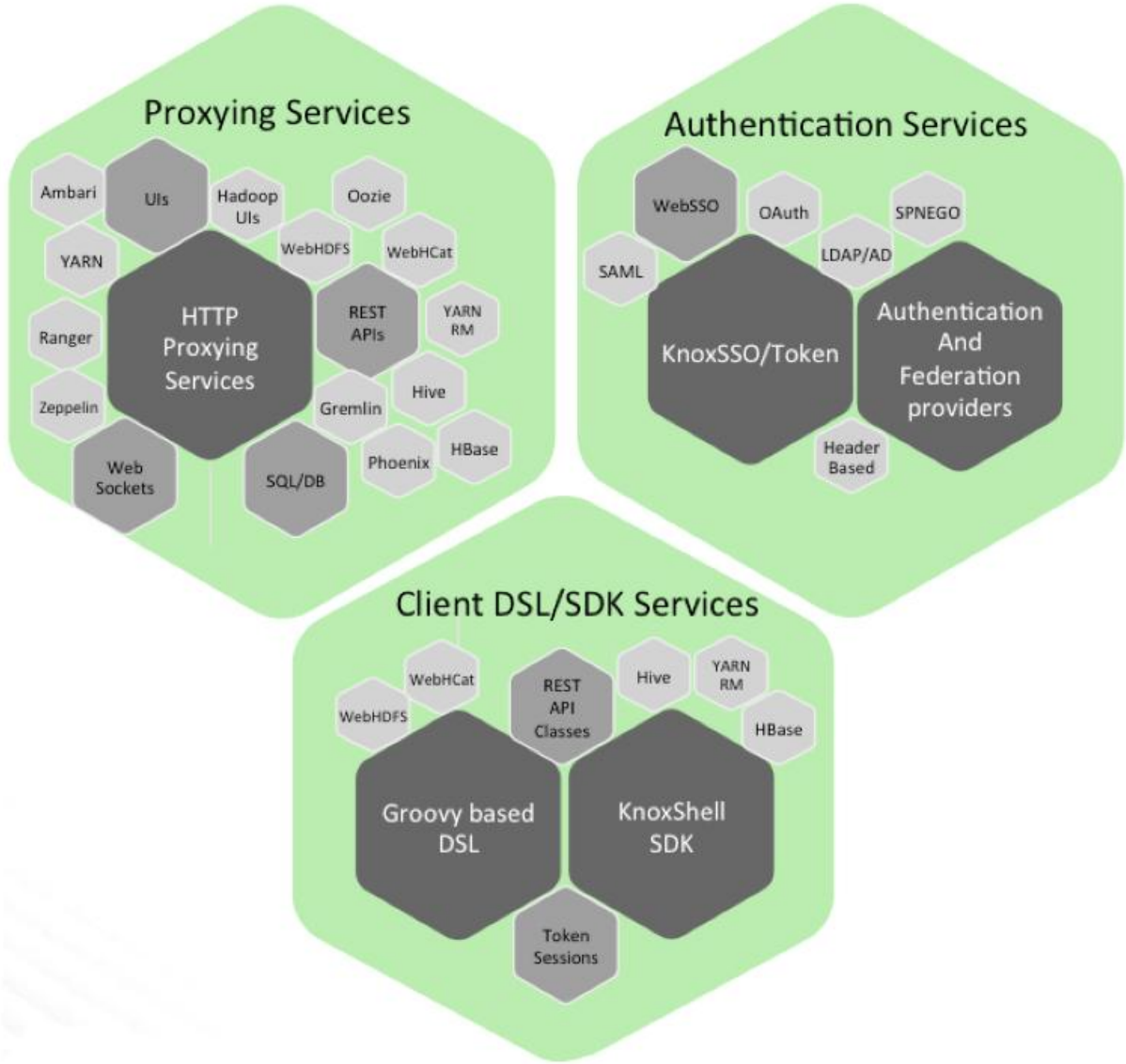
# Knox Implementation with AD/LDAP

## *Theory part*

### *Introduction*

***REST API and Application Gateway for the Apache Hadoop Ecosystem:*** *The Apache Knox™ Gateway is an Application Gateway for interacting with the REST APIs and UIs of Apache Hadoop deployments.* ***The Knox Gateway provides a single access point for all REST and HTTP interactions with Apache Hadoop clusters.***

*Knox delivers three groups of user facing services:*




- ✓ ***Proxying Services***

*Primary goals of the Apache Knox project are to provide access to Apache Hadoop via proxying of HTTP resources.*

- ✓ ***Authentication Services***

  *Authentication for REST API access as well as WebSSO flow for UIs. LDAP/AD, Header based PreAuth, Kerberos, SAML, OAuth are all available options.*

- ✓ ***Client Services***

*Client development can be done with scripting through DSL or using the Knox Shell classes directly as SDK.*

## ✓ *Supported Apache Hadoop Services*

*The following Apache Hadoop ecosystem services have integrations with the Knox Gateway:*

*Ambari*
*WebHDFS (HDFS)*
*Yarn RM*
*Stargate (Apache HBase)*
*Apache Oozie*
*Apache Hive/JDBC*
*Apache Hive WebHCat (Templeton)*
*Apache Storm*
*Apache Tinkerpop – Gremlin*
*Apache Avatica/Phoenix*
*Apache SOLR*
*Apache Livy (Spark REST Service)*
*Kafka REST Proxy*

## ✓ *Supported Apache Hadoop ecosystem UIs*

*Name Node UI*
*Job History UI*
*Yarn UI*
*Apache Oozie UI*
*Apache HBase UI*
*Apache Spark UI*
*Apache Ambari UI*
*Apache Ranger Admin Console*
*Apache Zeppelin*
*Apache NiFi*

## *Practical Part*

*The Knox Gateway supports one or more Hadoop clusters. Each Hadoop cluster configuration is defined in a topology deployment descriptor file in the* ***$gateway/conf/topologies*** *directory and is deployed to a corresponding WAR file in the* ***$gateway/data/deployments*** *directory. These files define how the gateway communicates with each Hadoop cluster.*

*The descriptor is an XML file contains the following sections:*

- *gateway/provider -- configuration settings enforced by the Knox Gateway while providing access to the Hadoop cluster.*
- *service -- defines the Hadoop service URLs used by the gateway to proxy communications from external clients.*

*The gateway automatically redeploys the cluster whenever it detects a new topology descriptor file, or detects a change in an existing topology descriptor file.*

*Configuring the Knox Topology to Connect to Hadoop Services*

*The Apache Knox Gateway redirects external requests to an internal Hadoop service using service name and URL of the service definition. Please find below is an example topology file; need to change the bold marked value as per your requirements.*

```
<topology>
    <gateway>
        <provider>
            <role>authentication</role>
            <name>ShiroProvider</name>
            <enabled>true</enabled>
     <param>
            <name>sessionTimeout</name>
            <value>30</value>
     </param>
     <param>
            <name>main.ldapRealm</name>
            <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm</value>
     </param>
      <!-- changes for AD/user sync -->
      <param>
            <name>main.ldapContextFactory</name>
            <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapContextFactory
            </value>
     </param>
            <!--main.ldapRealm.contextFactory needs to be placed before other
            main.ldapRealm.contextFactory* entries  -->
      <param>
            <name>main.ldapRealm.contextFactory</name>
            <value>$ldapContextFactory</value>
      </param>
      <param>
            <name>main.ldapRealm.contextFactory.url</name>
            <!-- need to put your AD FQDN with same port number -->
```

```
        <!--<value>ldap://ad2012.ansari.com:389</value> -->
        <value>ldaps:// ad2012.ansari.com:636</value>
    </param>
    <param>
        <name>main.ldapRealm.contextFactory.systemUsername</name>
        <value>cn=binduser,cn=users,dc=ansari,dc=com</value>
    </param>
    <param>
        <name>main.ldapRealm.contextFactory.systemPassword</name>
        <value>Faheem123</value> <!-- This is your binduser password -->
    </param>
    <param>
        <name>main.ldapRealm.contextFactory.authenticationMechanism</name>
        <value>simple</value>
    </param>
    <param>
        <name>urls./**</name>
        <value>authcBasic</value>
    </param>
    <param>
        <!-- please update your AD base DN -->
        <name>main.ldapRealm.searchBase</name>
        <value>DC=ansari,DC=com</value>
    </param>
    <param>
        <name>main.ldapRealm.userObjectClass</name>
        <value>person</value>
    </param>
    <param>
        <name>main.ldapRealm.userSearchAttributeName</name>
        <value>sAMAccountName</value>
    </param>
    <param>
        <name>main.ldapRealm.authorizationEnabled</name>
        <value>true</value>
    </param>
    <param>
         <!-- please update your AD base DN -->
        <name>main.ldapRealm.groupSearchBase</name>
        <value>dc=ansari,dc=com</value>
    </param>
    <param>
        <name>main.ldapRealm.groupObjectClass</name>
        <value>group</value>
    </param>
    <param>
        <name>main.ldapRealm.groupIdAttribute</name>
        <value>cn</value>
```

```
        </param>
           </provider>
        <provider>
              <role>identity-assertion</role>
              <name>Default</name>
              <enabled>true</enabled>
        </provider>
        <provider>
              <role>authorization</role>
              <name>AclsAuthz</name>
              <enabled>true</enabled>
        </provider>
        <provider>
              <role>ha</role>
              <name>HaProvider</name>
              <enabled>true</enabled>
        <param>
              <name>WEBHDFS</name>
              <value>maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=1
              000;enabled=true</value>
        </param>
        <param> <!-- please enter our management node FQDN>
              <name>HIVE</name>
              <value>maxFailoverAttempts=3;failoverSleep=1000;enabled=true;zookeeperEnsemble=hdp-
              node1.ansari.com:2181,hdp-node2.ansari.com:2181,hdp-
              node3.ansari.com:2181;zookeeperNamespace=hiveserver2</value>
        </param>
        <param>
               <name>OOZIE</name>
               <value>maxFailoverAttempts=3;failoverSleep=1000;enabled=true</value>
        </param>

        <param>
               <name>HBASE</name>
               <value>maxFailoverAttempts=3;failoverSleep=1000;enabled=true</value>
        </param>
        <param>
               <name>WEBHCAT</name>
               <value>maxFailoverAttempts=3;failoverSleep=1000;enabled=true</value>
        </param>
        <param>
               <name>RESOURCEMANAGER</name>
               <value>maxFailoverAttempts=3;failoverSleep=1000;maxRetryAttempts=300;retrySleep=
               3000;enabled=true</value>
        </param>
        </provider>
</gateway>
        <service>
```

```
        <role>NAMENODE</role>
        <url>hdfs://hdpafh</url> <!-- Your cluster name>
</service>
<service> <!-- please enter your Oozie Server FQDN>
        <role>OOZIEUI</role>
        <url>http://hdp-node2.ansari.com:11000/oozie</url>
        <!-- This is for is used when you have Oozie HA -->
        <url>http://hdp-node3.ansari.com:11000/oozie</url>
</service>
<service> <!-- please enter your Hbase master Server FQDN>
        <role>HBASEUI</role>
        <url>http://hdp-node3.ansari.com:16010/hbase</url>
        <!-- This is for is used when you have Hbase HA -->
        <url>http://hdp-node2.ansari.com:16010/hbase</url>
</service>
<!-- This section is for Hive server WebHcat UI's -->
<service>
        <!-- please enter your HCAT URL>
        <role>WEBHCAT</role>
        <url>http://hdp-node1.ansari.com:50111/webhcat</url>
        <url>http://hdp-node3.ansari.com:50111/webhcat</url>
</service>
<service> <!-- please enter your Namenode FQDN URL>
        <role>HDFSUI</role>
        <url>http://hdp-node1.ansari.com:50070</url>
        <url>http://hdp-node2.ansari.com:50070</url>
</service>

<service> <!-- please enter your Namenode FQDN URL>
        <role>WEBHDFS</role>
        <url>http://hdp-node1.ansari.com:50070/webhdfs</url>
        <url>http://hdp-node2.ansari.com:50070/webhdfs</url>
</service>
<service> <!-- please enter your Namenode FQDN URL>
        <role>NAMENODE</role>
        <url>hdfs://hdp-node1.ansari.com:8020</url>
        <url>hdfs://hdp-node2.ansari.com:8020</url>
</service>
<service> <!-- please enter your YARN url -- >
        <role>YARNUI</role> <!--  This is for YARN HA -->
        <url>http://hdp-node2.ansari.com:8088</url>
        <url>http://hdp-node1.ansari.com:8088</url>
</service>
<service> <!-- your ambari Server FQDN -->
        <role>AMBARIUI</role>
        <url>http://hdp-node1.ansari.com:8080</url>
</service>
<service> <!-- your zeppelin Server FQDN or URl -->
```

```
            <role>ZEPPELINUI</role>
            <url>http://hdp-node1.ansari.com:9995</url>
        </service>
        <service> <!-- your zeppelin Server FQDN or URl -->
            <role>ZEPPELINWS</role>
            <url>ws://hdp-node1.ansari.com:9995/ws</url>
        </service>
        <service>  <!-- please update your RM FQDN or URL -->
            <role>RESOURCEMANAGER</role>
            <url>http://hdp-node1.ansari.com:8088/ws</url>
            <url>http://hdp-node2.ansari.com:8088/ws</url>
        </service>
        <service> <!-- Please enter your Ranger Admin FDQN or URl -->
            <role>RANGERUI</role>
            <url>http://hdp-node3.ansari.com:6080</url>
        </service>
        <service>
            <role>HIVE</role>
        </service>
</topology>
```

***Note: All bold highlighted entries in this file are comments and for reference purpose.***

## *Verification of Knox Setup*

*Here are the steps to be followed to verify the knox configuration*

1. *Copy topology file create in last section on knox gateway nodes under* ***/etc/knox/conf/topologies/*** *directories.*

2. *First check the AD/Ldap connectivity with knox gateway node*

   ***ldapsearch -W -H ldap://ad2012.ansari.com -D binduser@ansari.com -b "dc=ansari,dc=com"***
   ***ldapsearch -W -H ldaps://ad2012.ansari.com -D binduser@ansari.com -b "dc=ansari,dc=com"***

3. ***Validate your topology file.***

   ***/usr/hdp/2.6.0.3-8/knox/bin/knoxcli.sh validate-topology --cluster hdpafh***

   ***Outputof above command:***
   ***File to be validated:***
   ***/usr/hdp/2.6.0.3-8/knox/bin/../conf/topologies/hdpafh.xml***

***Topology file validated successfully***

***Note: walhdp is topology file name and /usr/hdp/X.X.X.X-X/, our hdp version.***

4. ***Validate your auth users***
   ***/usr/hdp/2.6.4.0-91/knox/bin/knoxcli.sh --d system-user-auth-test --cluster hdpafh***

   ***Outputof above command: System LDAP Bind successful***

   ***Note: hdpafh is a topology file name and /usr/hdp/X.X.X.X-X/, our hdpversion.***

5. ***Make sure the below properties are set***

   ***HDFS: Core-site.xml***
   *hadoop.proxyuser.knox.groups=**
   *hadoop.proxyuser.knox.hosts=**

   ***HIVE: hive-site.xml***
   *webhcat.proxyuser.knox.groups=**
   *webhcat.proxyuser.knox.hosts=**
   *hive.server2.allow.user.substitution=true*
   *hive.server2.transport.mode=http*
   *hive.server2.thrift.http.port=10001*
   *hive.server2.thrift.http.path=cliservice*

   ***Oozie: oozie-site.xml***
   *oozie.service.ProxyUserService.proxyuser.knox.groups=**
   *oozie.service.ProxyUserService.proxyuser.knox.hosts=**

***Accessing Hadoop Cluster via Knox***

*Now let's check if we can access Hadoop Cluster via Apache Knox services. Using Knox means we can utilize the HTTPS protocol which utilizes SSL to encrypt our requests and makes using it much more secure.*
*Not only do we get the added benefit of the extra layer of protection with encryption, but we also get to plug in the LDAP server which many*

*organizations already utilize for authentication:*

***curl -vvv -i -k -u binduser -X GET https://namenodeFQDN:8443/gateway/hdpafh/webhdfs/v1?op=LISTSTATUS***

***Run Beeline Client***

*First we're going to need to login to knox gateway node then invoke a beeline shell.*

***#beeline***

***Connect to Hive Server 2***

*Copy our gateway.jks file from* ***/usr/hdp/current/knox-server/data/security/keystore/ to /home/afh/,*** *then try the following command in the beeline shell to access hive.*

***!connect jdbc:hive2://FQDNOFKNOX GatewayServer URl:8443/;ssl=true;sslTrustStore=/home/afh/gateway.jks ;trustStorePassword=bigdata;transportMode=http;httpPath=gateway/hdpafh/hive***

***Note: hdpafh is our topology file name and;ssl=sslTustStore=We have put the path of our knox gateway.jks file..***

*Enter username and password that Beeline will send to Knox over HTTPS to authenticate the user.*

***Connecting AD via secure ldap***

*If we want to access AD with secure ldap protocol then we needs to do below settings, Trust the Active Directories certificate.*

***Note:*** *This is required for self-signed certificates, as in our implementation of Active Directory*

i. *Login to AD then go* ***Server manager>> Tools >> open Certification Authority> click on issued Certificates:***

***double click on LdapOverSSL certificate >> then go to Details >> copy to File and Save it.***

ii. *Trust the CA certificate to execute the below commands on gateway node.*

***sudo update-ca-trust enable; sudo update-ca-trust extract; sudo update-ca-trust check***

iii. *Trust the CA certificate in Java.*

***$JAVA-HOME/bin/keytool -importcert -noprompt -storepasschangeit -file /etc/pki/ca-trust/source/anchors/ldap_ssl.pem -aliaseknos_ssl -keystore /etc/pki/java/cacerts***

iv. *Change your topology file under ldp to ldps and also port number of ldaps. ldaps is working on port 636 and ldap port is 389.*

*<param>*
*<name>main.ldapRealm.contextFactory.url</name>*
***<!--<value>ldap://ad2012.ansari.com:389</value> -->***
*<value>ldaps://ad2012.ansari.com:636</value>*
*</param>*

v. *Connect to Hive Server 2 via beeline.*

*!connect jdbc:hive2://hdp-node3.ansari.com:8443/;ssl=true;sslTrustStore=/home/afh/gateway.jks;trustStorePassword=bigdata;transportMode=http;httpPath=gateway/hdpafh /hive*

***Note: Hdpafh is our topology file name** and;**ssl=sslTustStore=We have put the path of our knox gateway.jks file.**.*

***To access different Web UI via knox using below command***

***Here is the steps to access web ui via knox gateway url.***

*i. To access Ambari web ui via below url:*

***https://knox-server-fqdn:8443/gateway/hdpafh /ambari/***

*ii. To access HDFS UI via below url:*

***https://knox-server-fqdn:8443/gateway/hdpafh /hdfs/***

*iii. To access HBase ui via below url:*

***https://knox-server-fqdn:8443/gateway/hdpafh /hbase/webui/***

*iv. To access YARN ui via below url:*

***https://knox-server-fqdn:8443/gateway/hdpafh /yarn/cluster/apps/RUNNING***

*v. To access Ranger web ui via below url:*

***https://knox-server-fqdn:8443/gateway/hdpafh /ranger/index.html***

*vi. To access Oozie web ui via below url:*

***https://knox-server-fqdn:8443/gateway/ hdpafh /oozie/***

*vii. To access Zeppline web ui, first need to update content of shiro_ini_content file in ambariui. Our shiro_ini_content file should be looks like below file, need to verify only bold entries.*

*# Sample LDAP configuration, for Active Directory user Authentication, currently tested for single Realm*
*[main]*
*ldapRealm=org.apache.zeppelin.realm.LdapRealm*
*ldapRealm.contextFactory.systemUsername=**cn=binduser,cn=users,dc=ansari,dc=com***
*ldapRealm.contextFactory.systemPassword=**Faheem123***
*ldapRealm.contextFactory.authenticationMechanism=simple*
*ldapRealm.contextFactory.url=ldaps**://ad2012.ansari.com:636***
*# Ability to set ldap paging Size if needed; default is 100*
*ldapRealm.pagingSize=200*
*ldapRealm.authorizationEnabled=true*

*ldapRealm.searchBase=**DC=ansari,DC=com***
*#ldapRealm.userSearchBase=OU=CorpUsers,DC=lab,DC=hortonworks,DC=net*
*ldapRealm.groupSearchBase=**dc=ansari,dc=com***
*ldapRealm.userObjectClass=**person***
*ldapRealm.groupObjectClass=**group***
*ldapRealm.userSearchAttributeName = **sAMAccountName***
*# Set search scopes for user and group. Values: subtree (default), onelevel, object*
*#ldapRealm.userSearchScope = **subtree***
*#ldapRealm.groupSearchScope = **subtree***
*#ldapRealm.userSearchFilter=**(&(objectclass=person)(sAMAccountName={0}))***
*#ldapRealm.memberAttribute=**member***
*# Format to parse & search group member values in 'memberAttribute'*
*#ldapRealm.memberAttributeValueTemplate=CN={0},OU=CorpUsers,DC=lab,DC=hortonwork s,DC=net*
*# No need to give userDnTemplate if memberAttributeValueTemplate is provided*
*#ldapRealm.userDnTemplate=*
*# Map from physical AD groups to logical application roles*
*#ldapRealm.rolesByGroup = "hadoop-admins":admin_role,"hadoop-users":hadoop_users_role*
*# Force usernames returned from ldap to lowercase, useful for AD*
*#ldapRealm.userLowerCase = true*
*# Enable support for nested groups using the LDAP_MATCHING_RULE_IN_CHAIN operator*
*#ldapRealm.groupSearchEnableMatchingRuleInChain = true*

*sessionManager = org.apache.shiro.web.session.mgt.DefaultWebSessionManager*
*### If caching of user is required then uncomment below lines*
*cacheManager = org.apache.shiro.cache.MemoryConstrainedCacheManager*
*securityManager.cacheManager = $cacheManager*

*securityManager.sessionManager = $sessionManager*
*securityManager.realms = $ldapRealm*
*# 86,400,000 milliseconds = 24 hour*
*securityManager.sessionManager.globalSessionTimeout = 86400000*
*shiro.loginUrl = /api/login*

*[urls]*
*# This section is used for url-based security.*
*# You can secure interpreter, configuration and credential information by urls. Comment or uncomment the below urls that you want to hide.*
*# anon means the access is anonymous.*
*# authc means Form based Auth Security*
*# To enfore security, comment the line below and uncomment the next one*
*#/api/version = anon*
*/api/interpreter/** = authcBasic, roles[admin_role,hadoop_users_role]*
*/api/configurations/** = authcBasic, roles[admin_role]*
*/api/credential/** = authcBasic, roles[admin_role,hadoop_users_role]*
*#/** = anon*
*/** = authcBasic*

*Once the file is set need to restart zeppline service. Then access zeppline web ui using below url.*

***https:// knox-server-fqdn: 8443/gateway/hdpafh/zeppelin/***