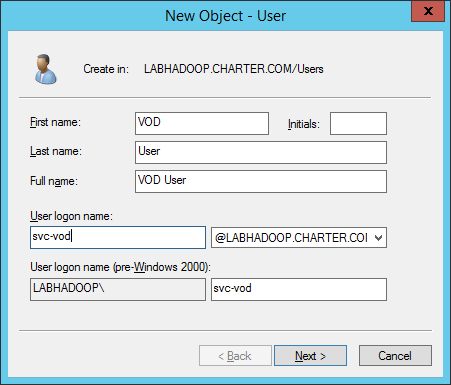
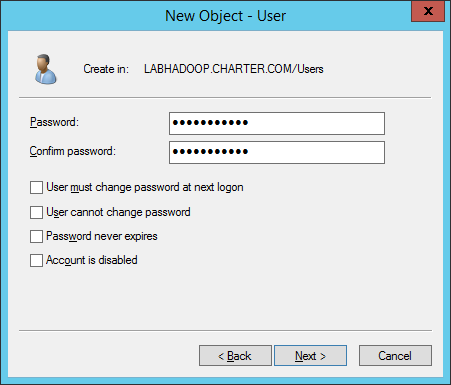
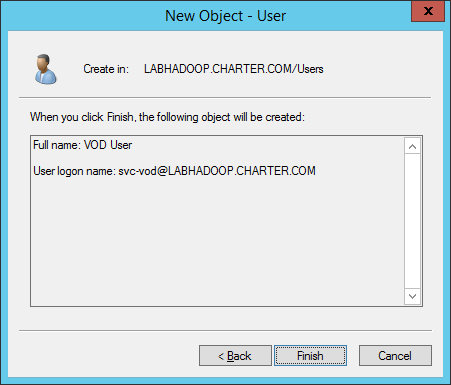
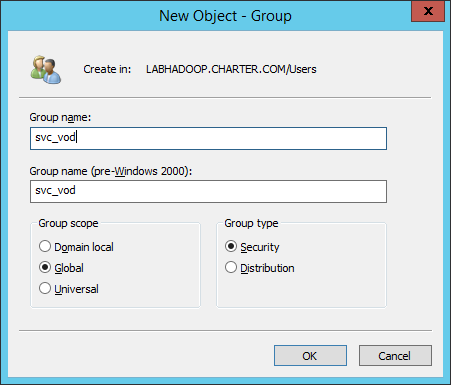
Create user:



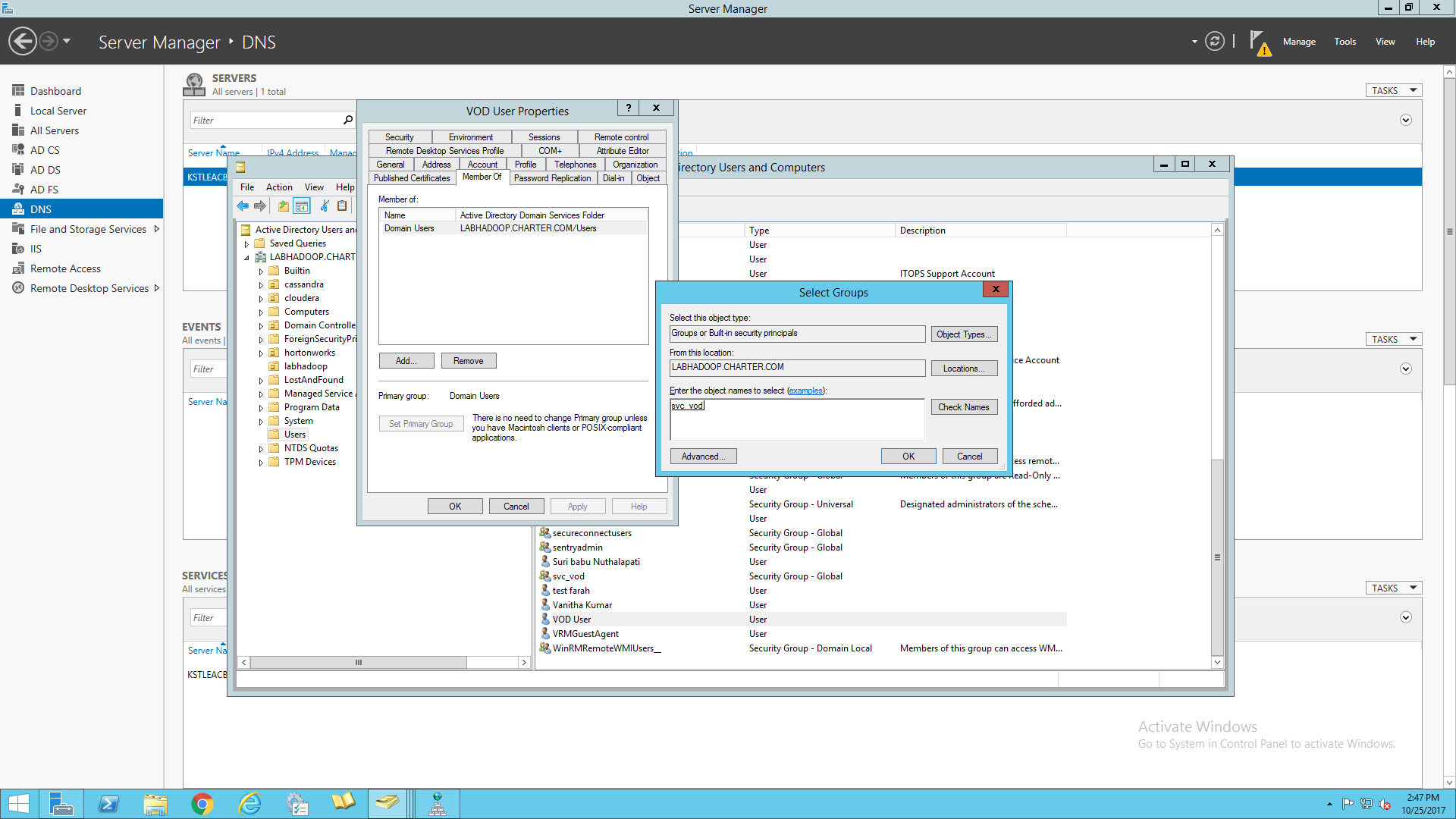




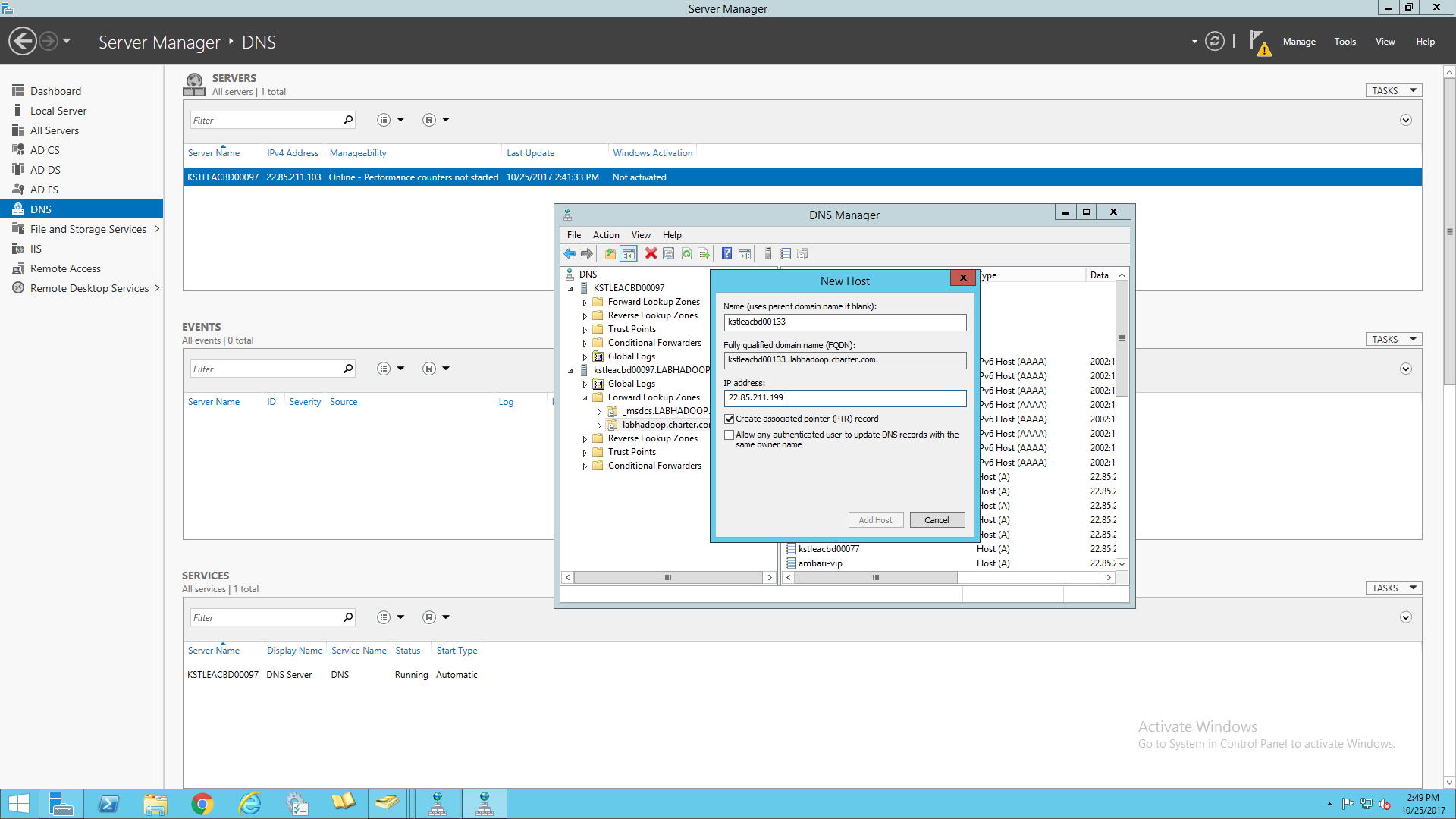
New Group:



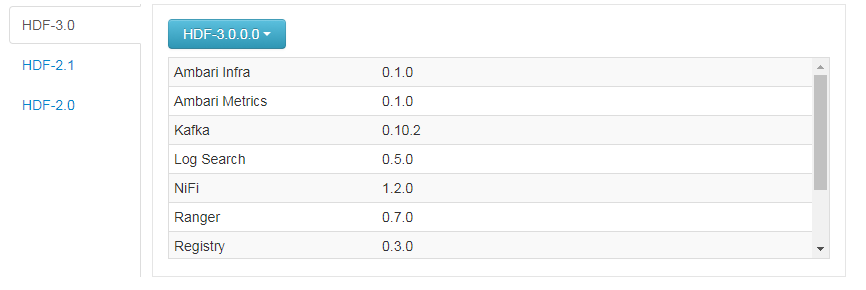
Add user to group

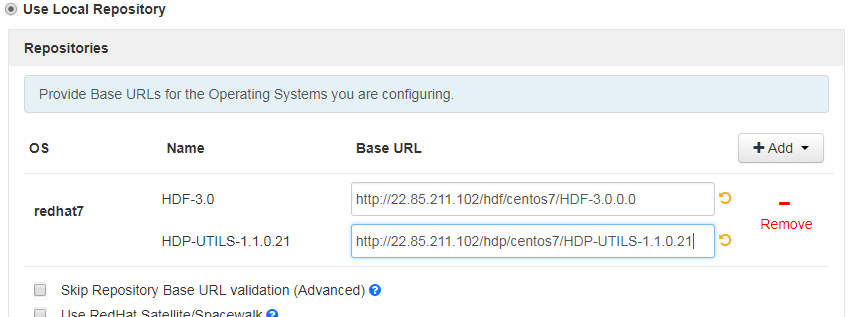


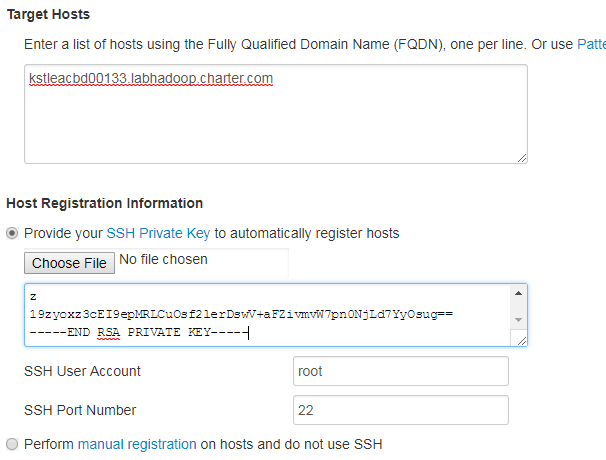
Add DNS



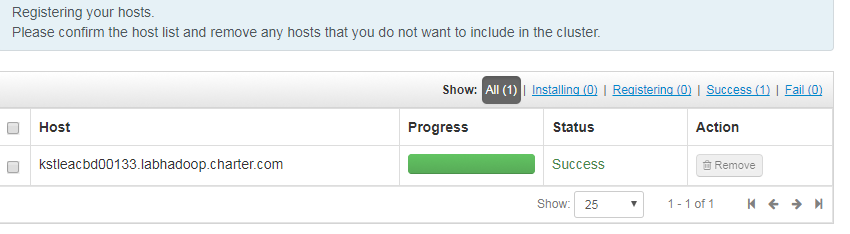
Kafka Installation:

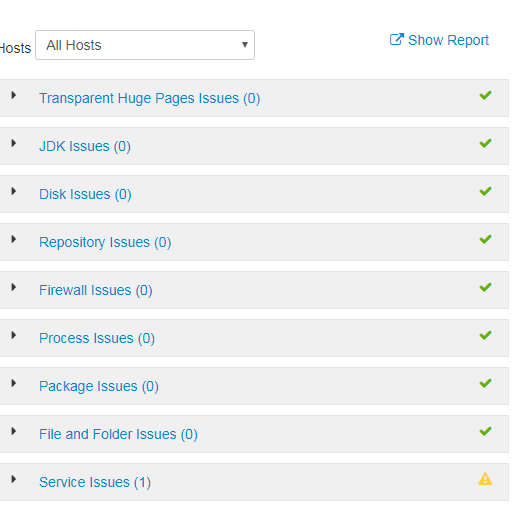




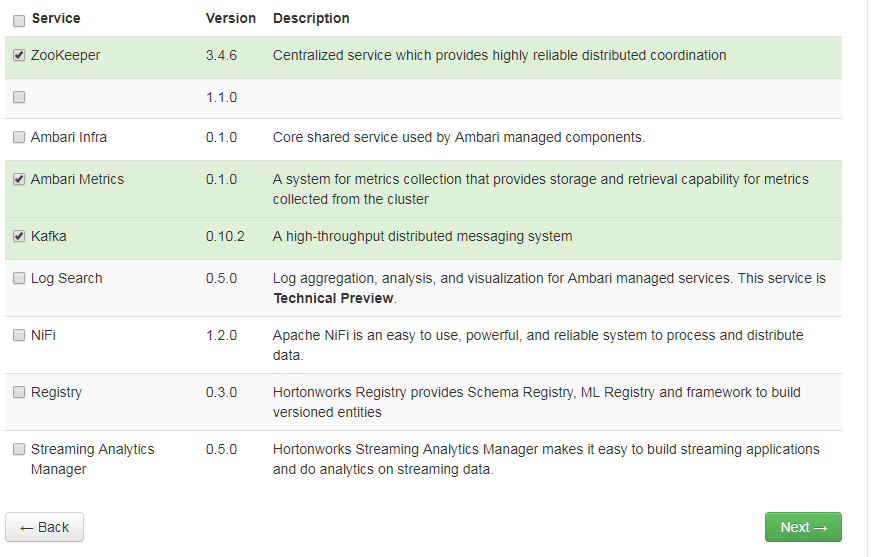


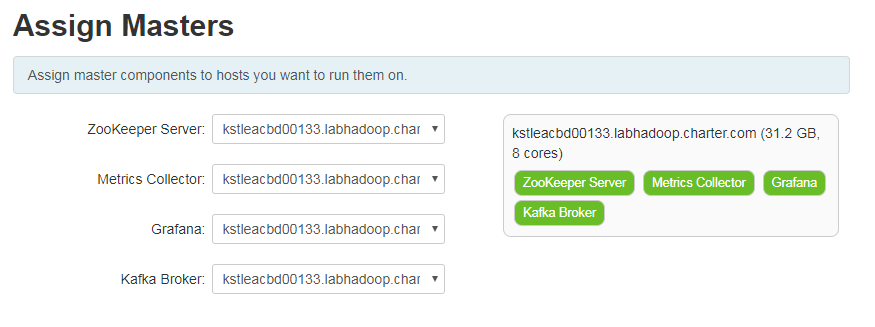
Register and Confirm

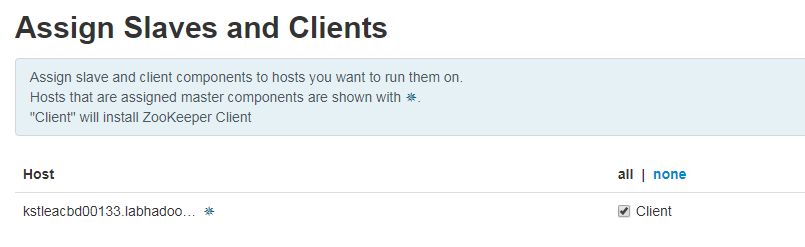






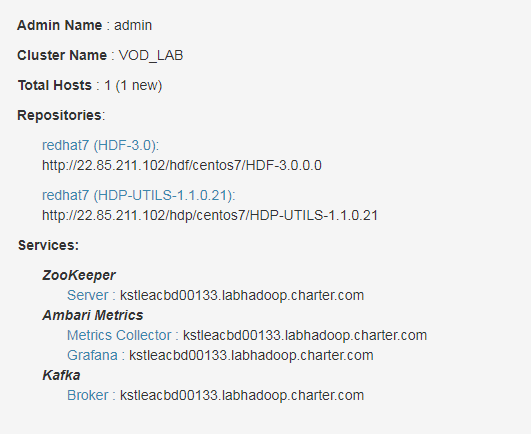






https://community.hortonworks.com/questions/109103/ambari-25-create-cluster-registration-host-failed.html

https://access.redhat.com/articles/2039753#controlling-certificate-verification-7



Ranger:

[root@kstleacbd00133 ambari-server]# mysql -u root -p

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 260

Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database ranger\_hdf;

Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> grant all on ranger\_hdf.\* to 'ranger\_hdf'@'%' identified by 'horton123';

Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> flush privileges;

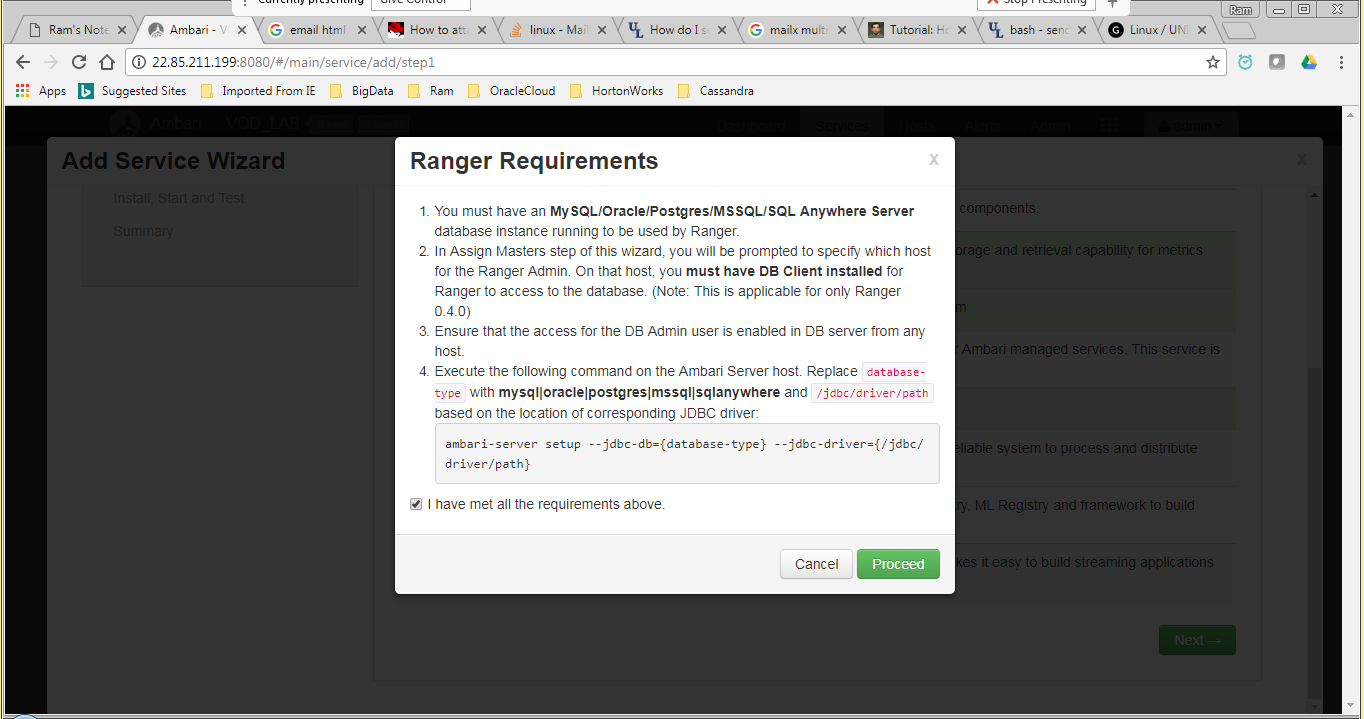
Query OK, 0 rows affected (0.00 sec)

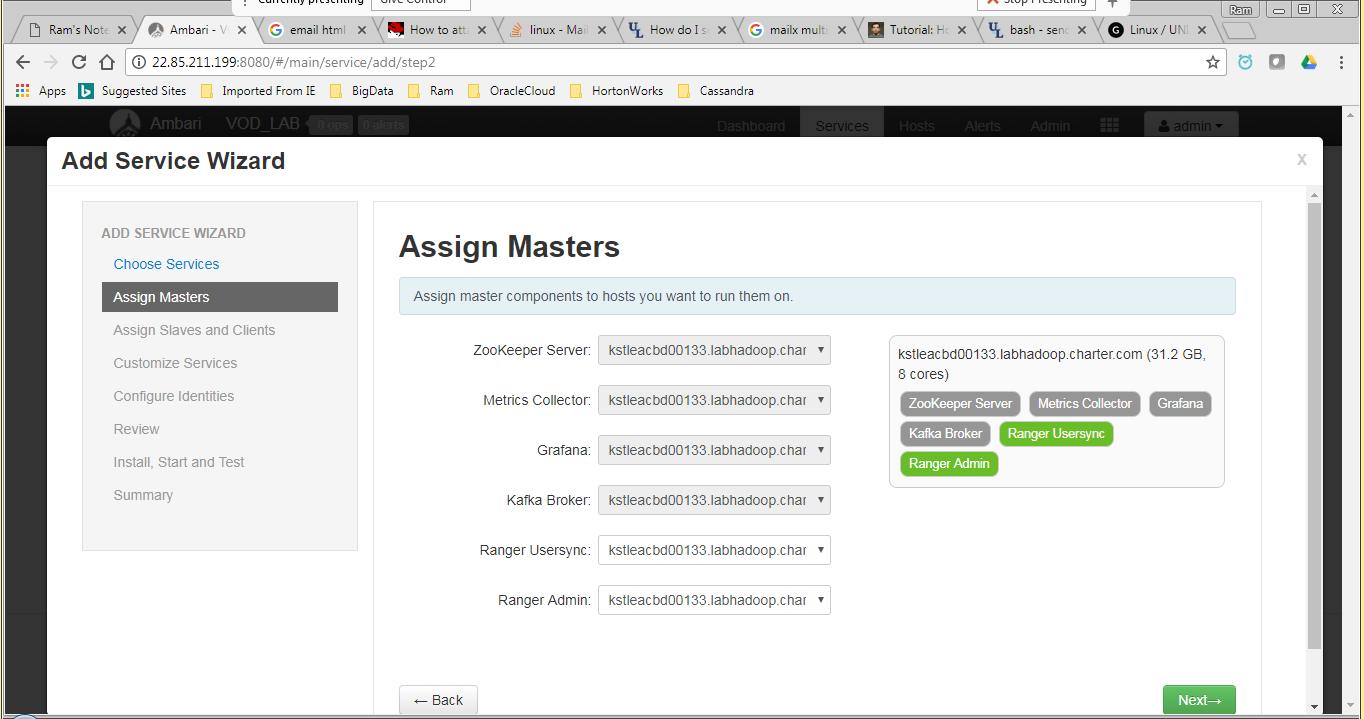
MariaDB [(none)]> exit

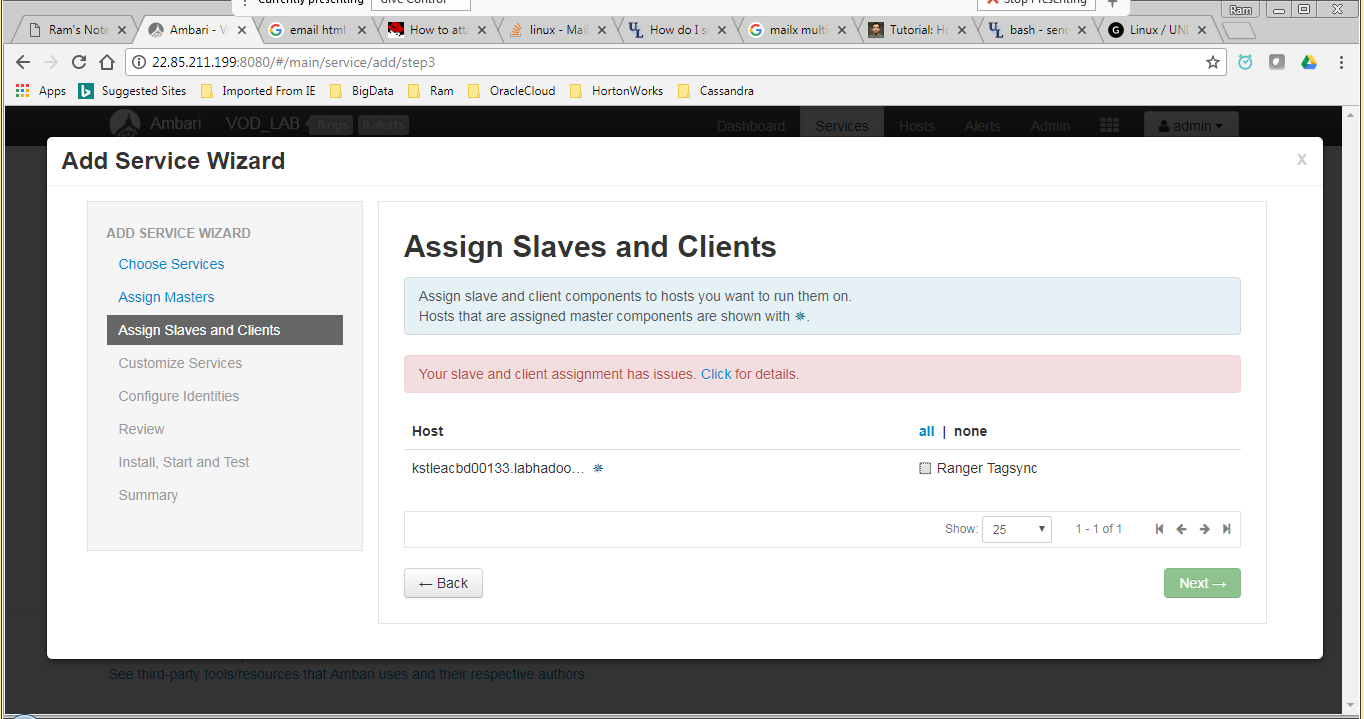
Bye

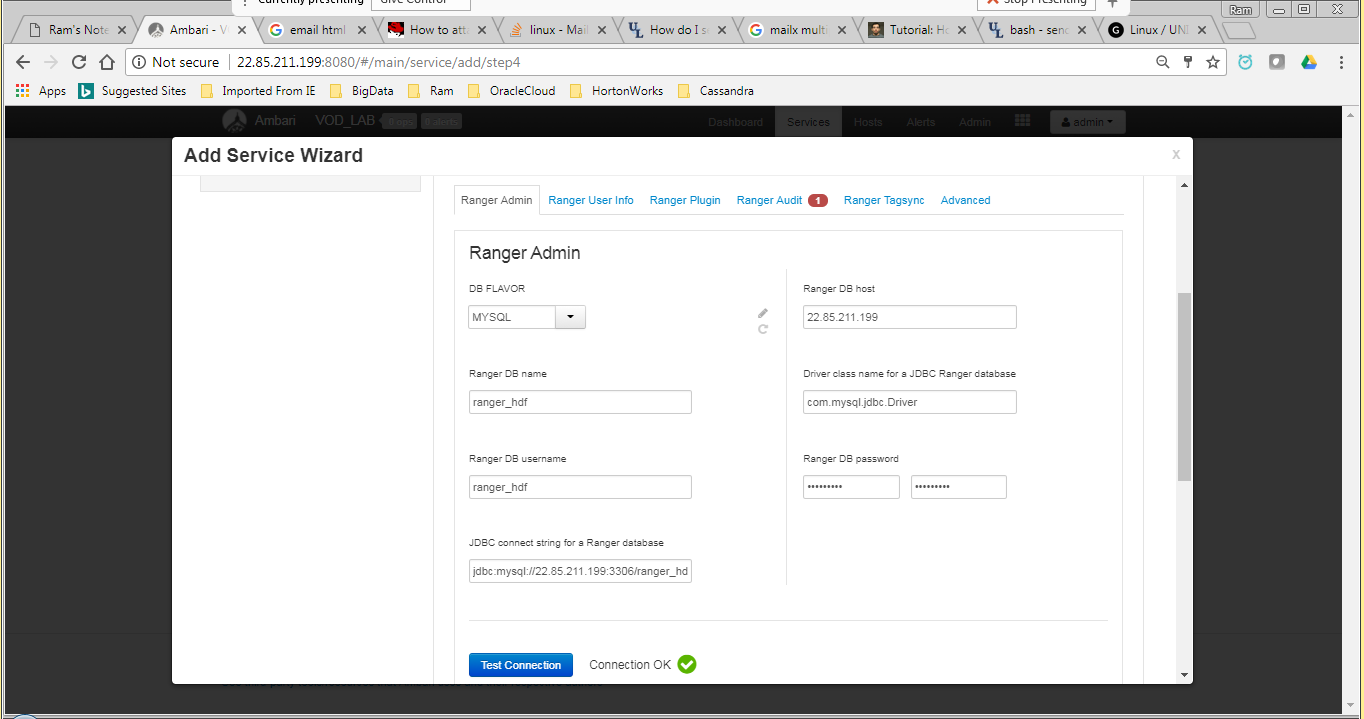
[root@kstleacbd00133 ambari-server]#

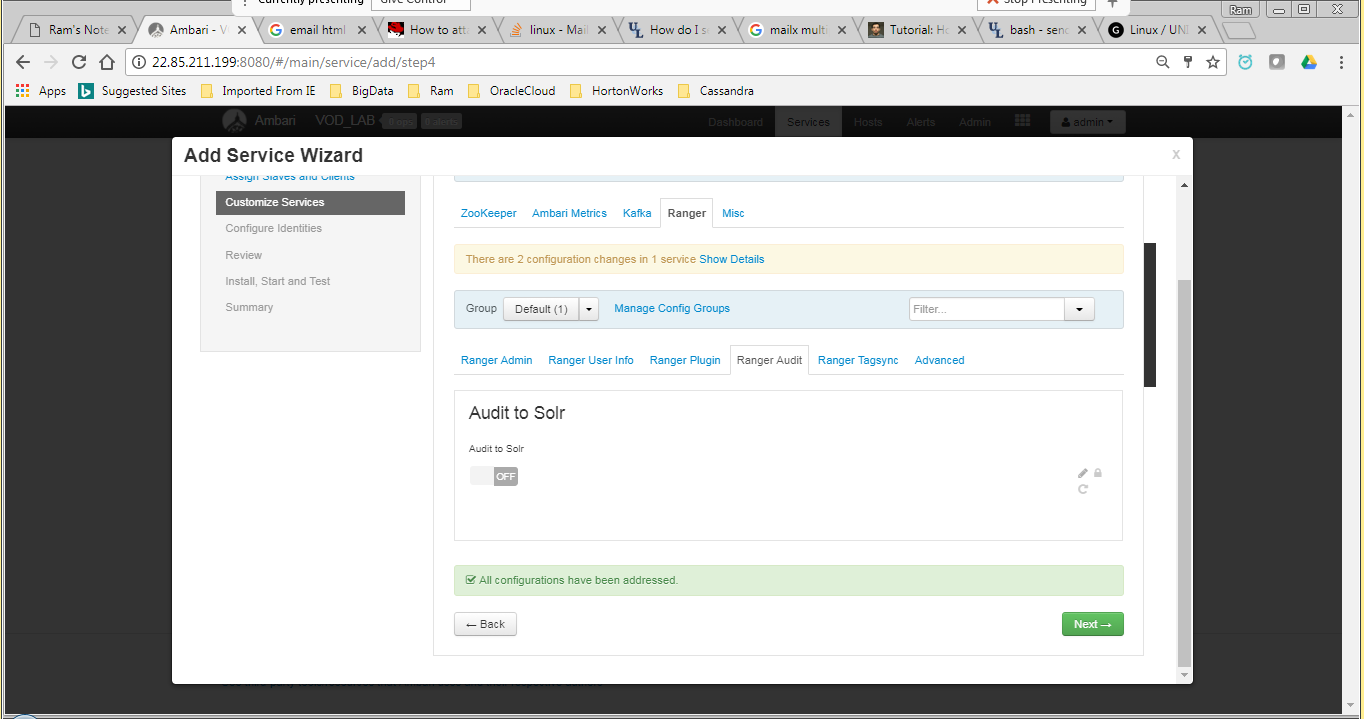
Actions -> Add Service -> Ranger -> Next ->

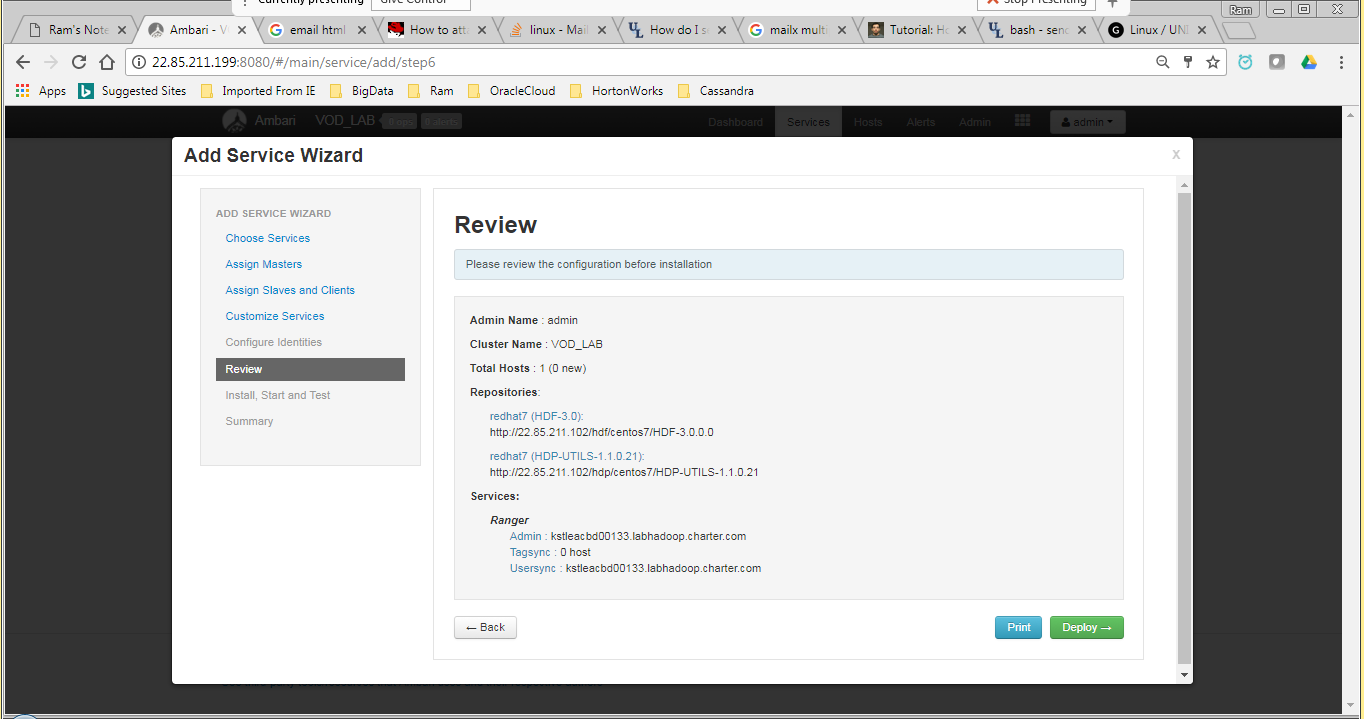


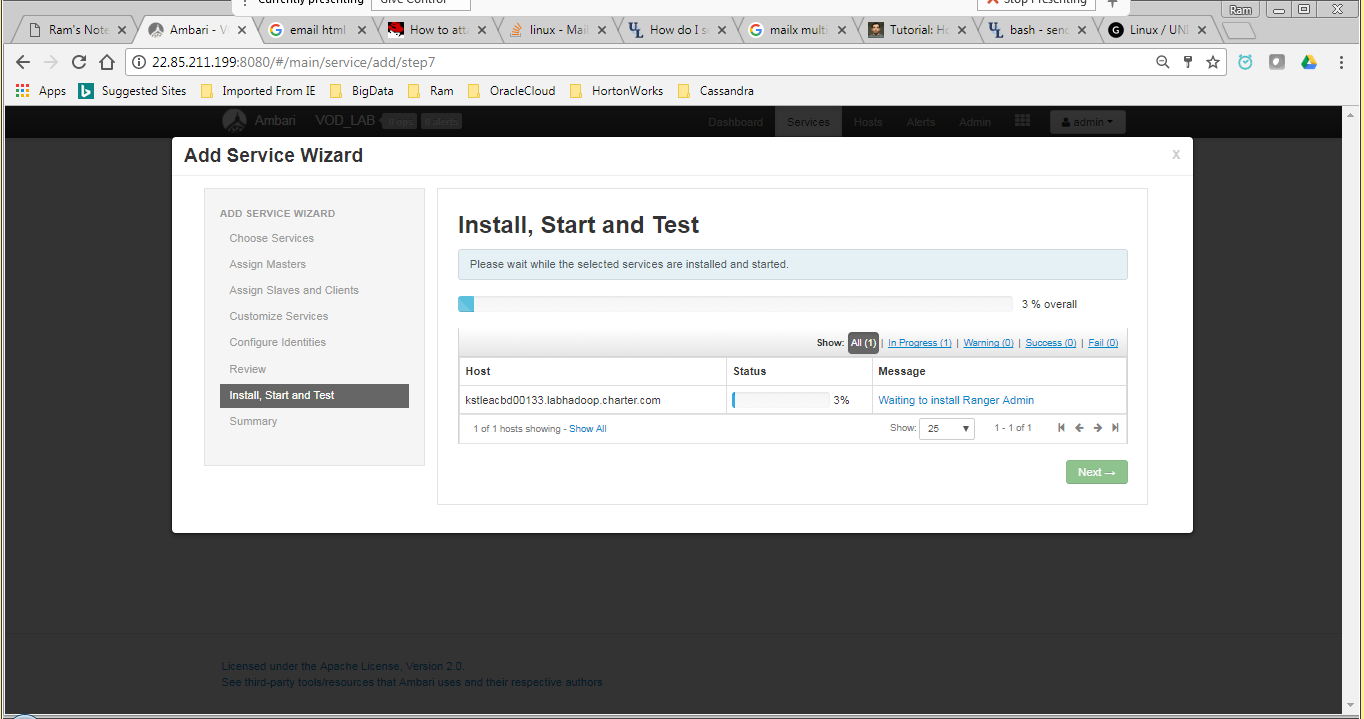


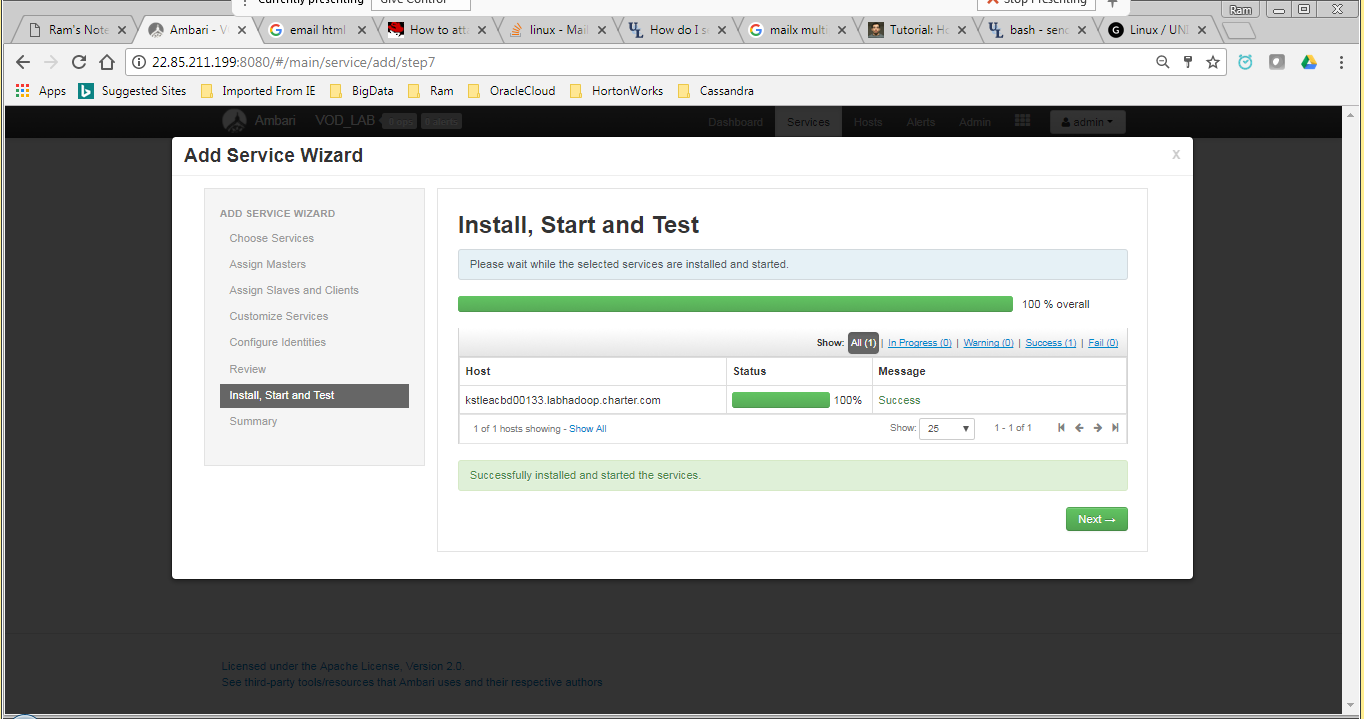


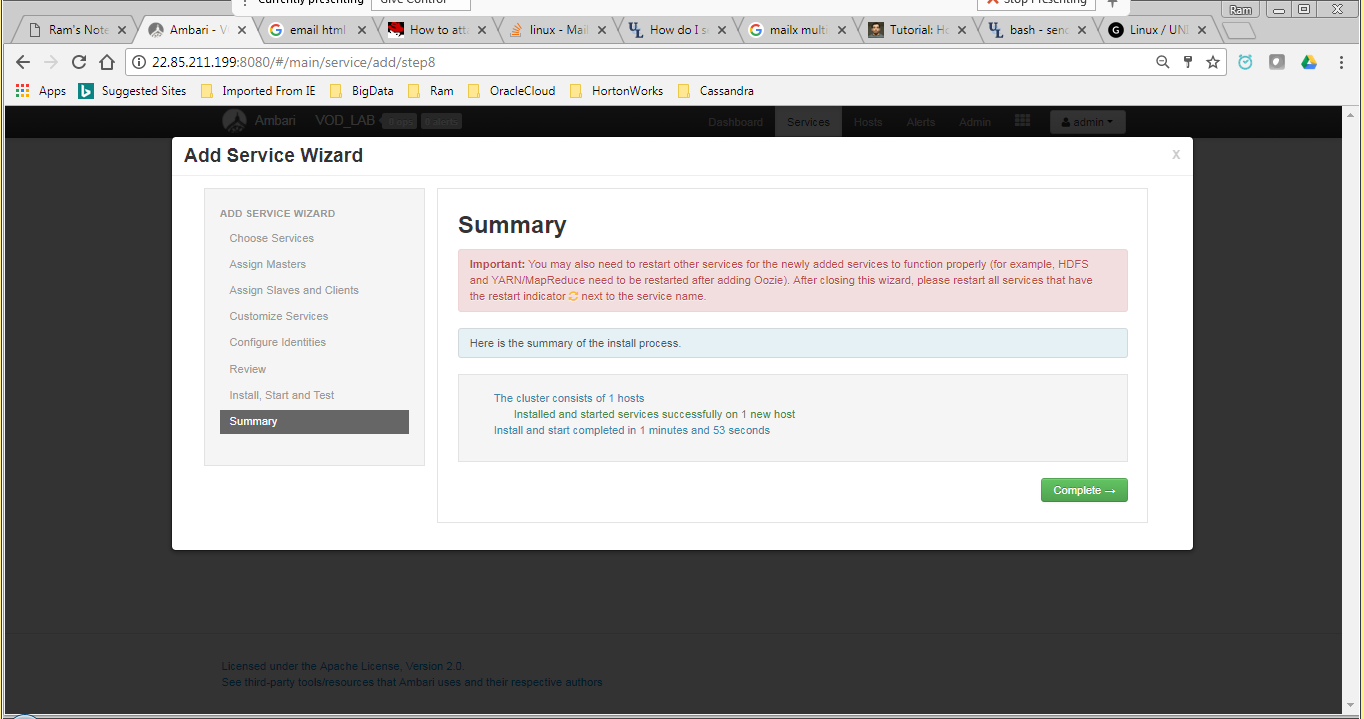








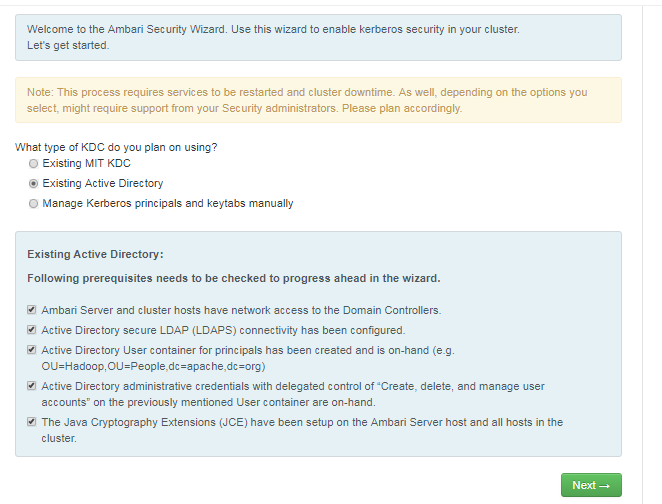


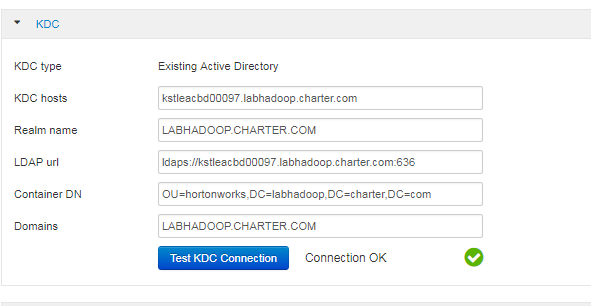


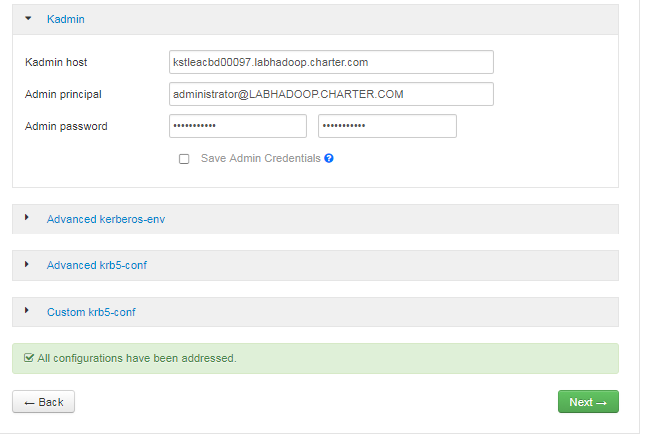
Kerberos

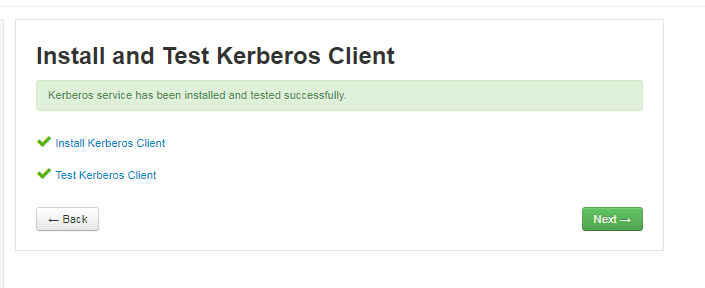
Admin -> Kerberos -> Enable Kerberos

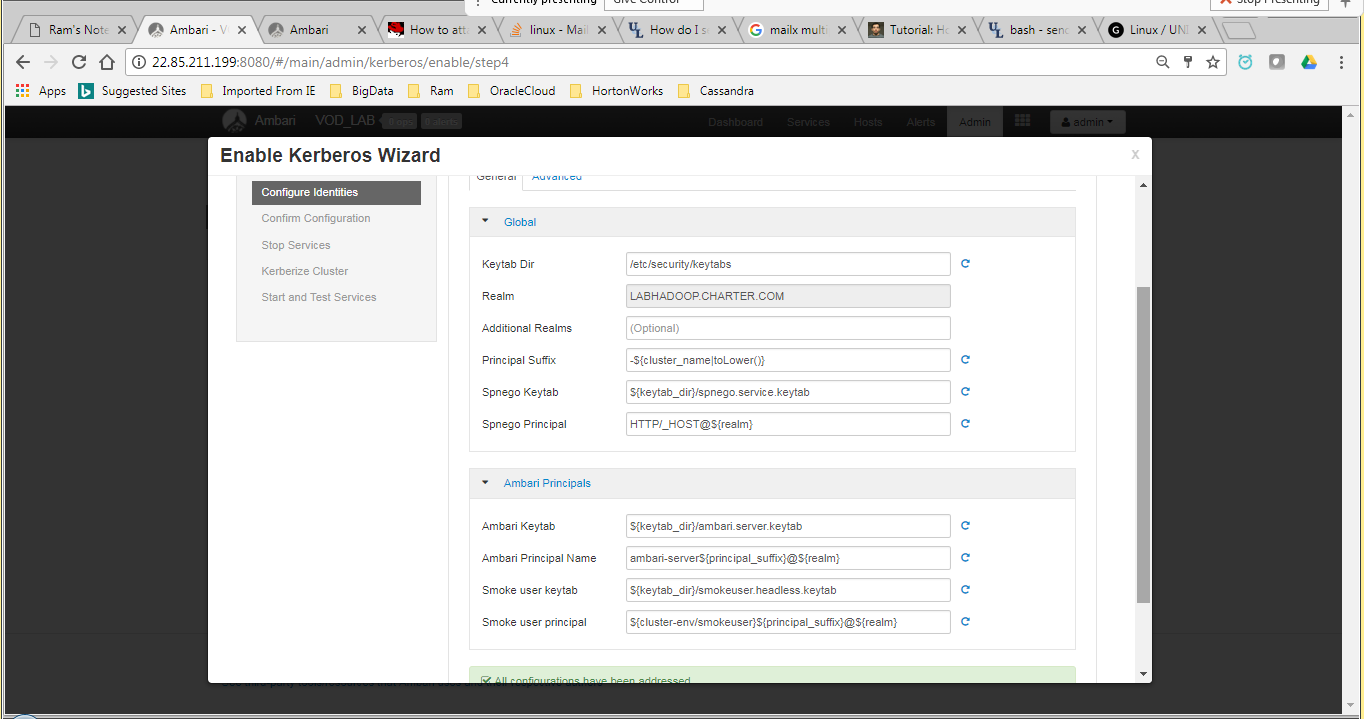
[root@kstleacbd00133 ambari-server]# yum install openldap-clients



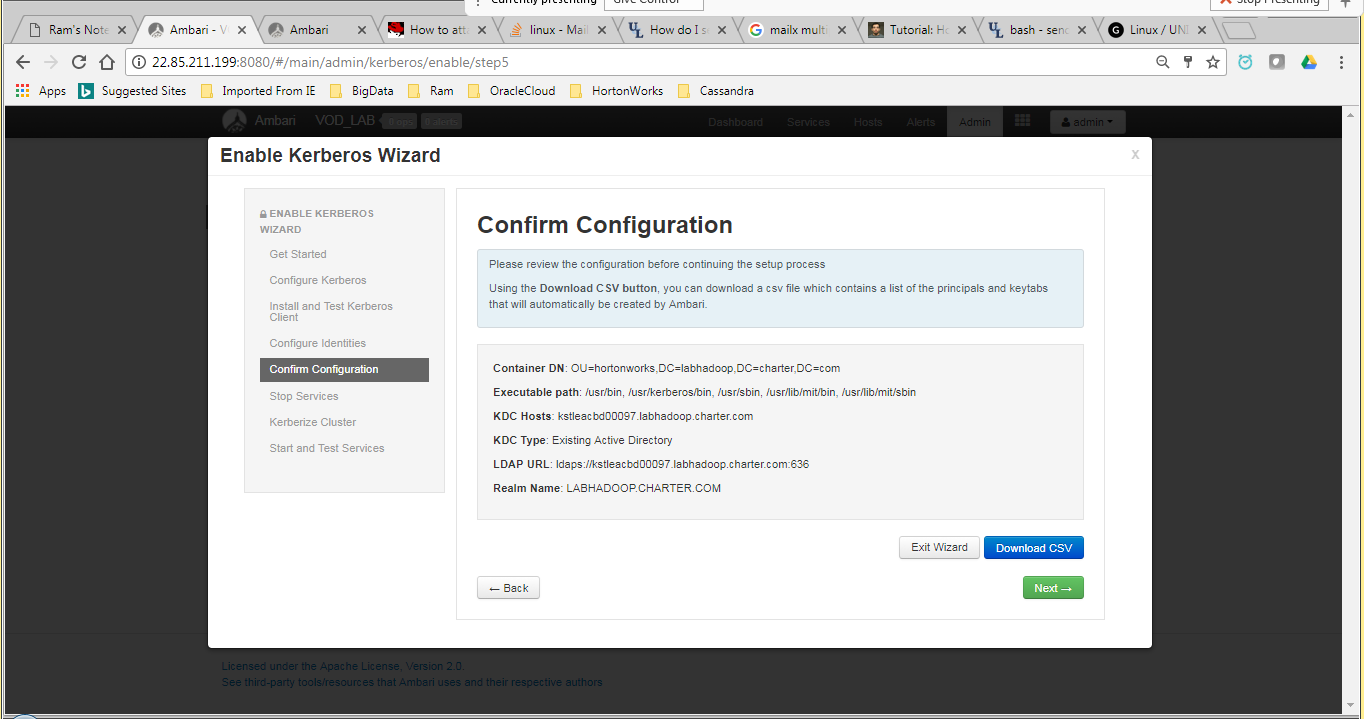


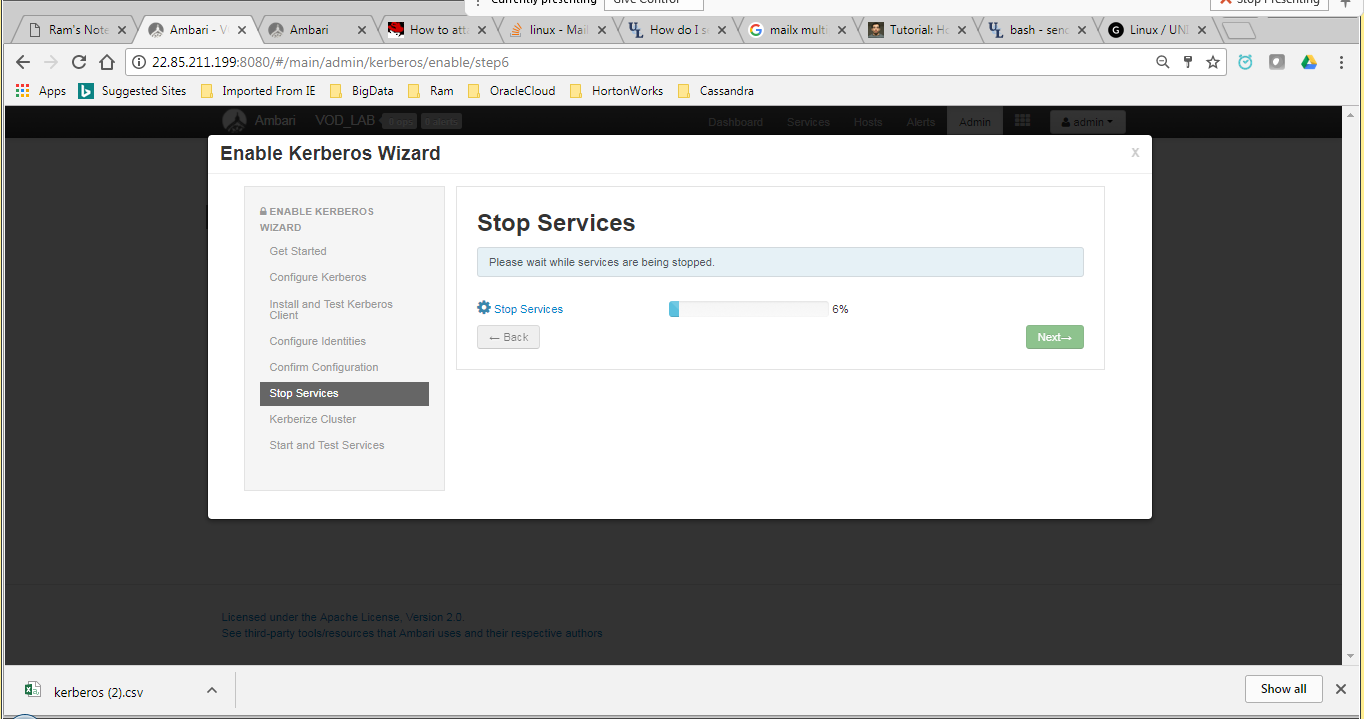


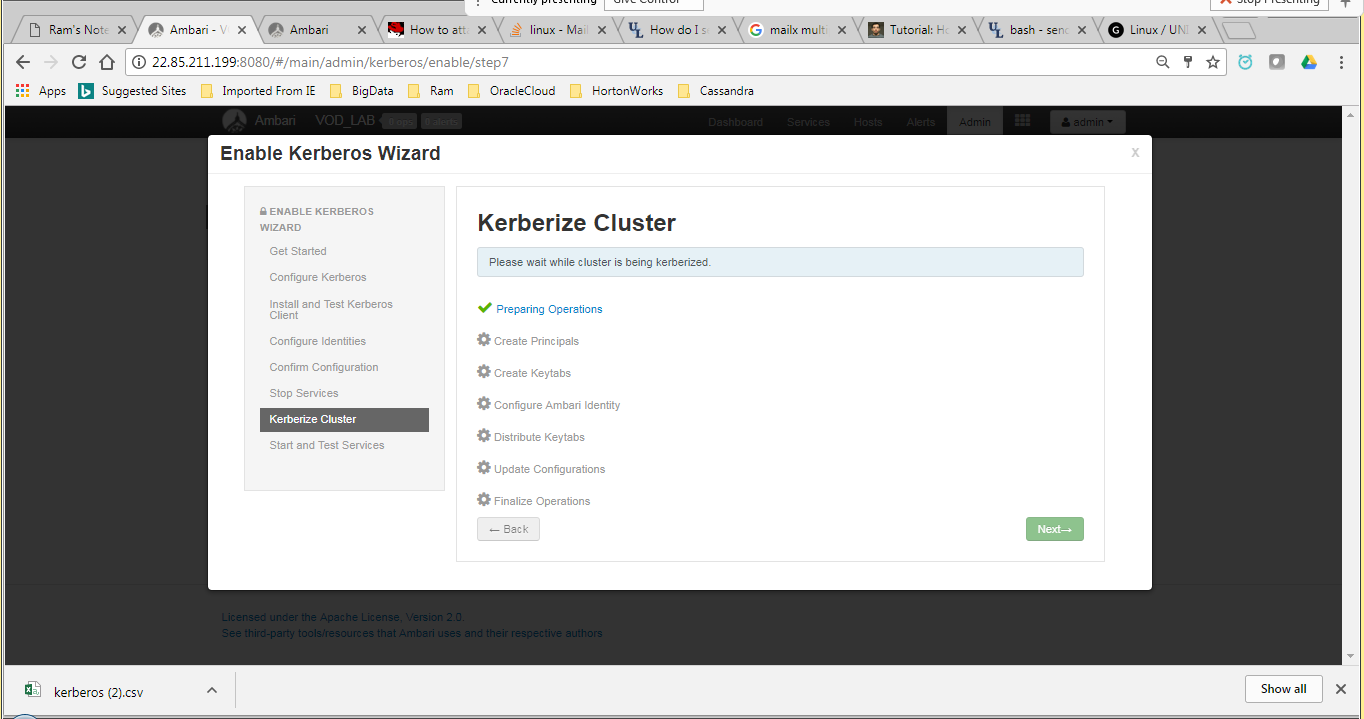


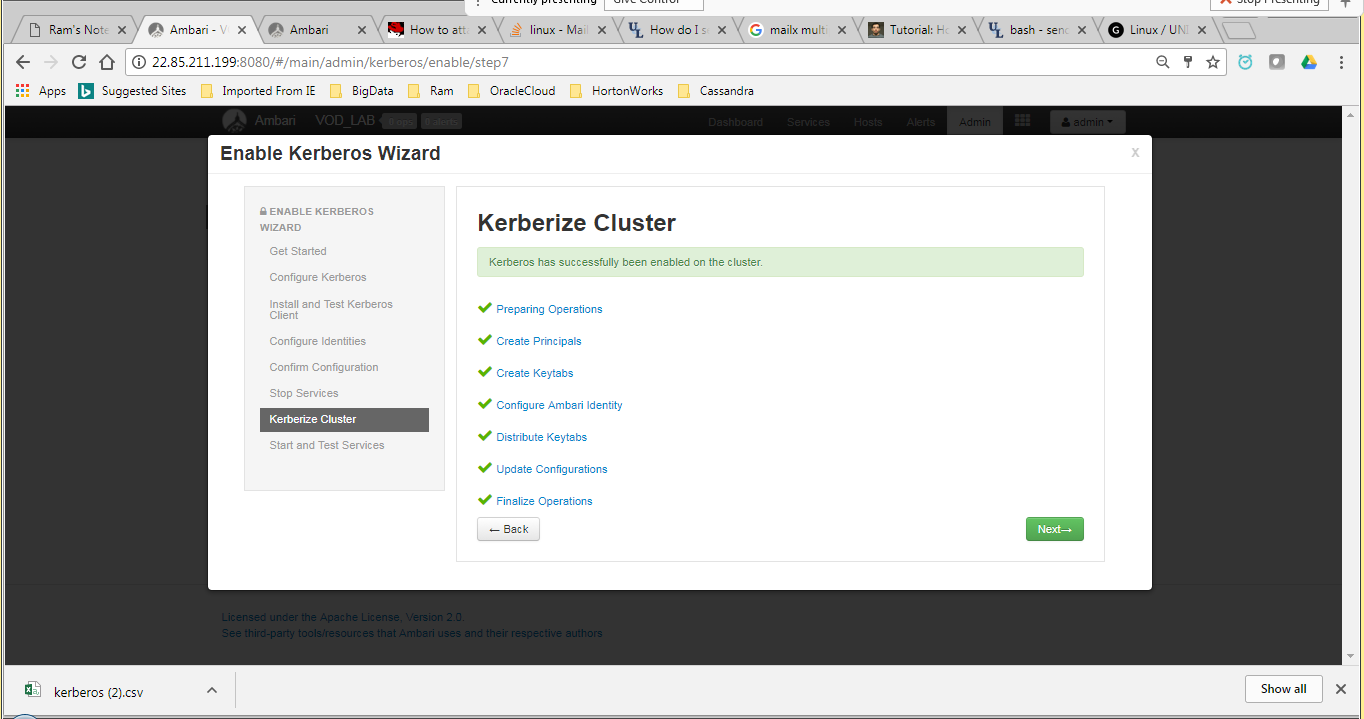


All defaults









SSL:

[root@kstleacbd00133 ~]# mkdir -p /opt/hadoop/security/ca-certs /opt/hadoop/security/jks /opt/hadoop/security/tmp /opt/hadoop/security/certs /opt/hadoop/security/truststore /opt/hadoop/security/x509

[root@kstleacbd00133 ~]# openssl genrsa -out /opt/hadoop/security/ca-certs/rootCA.key -aes256 -passout pass:Charter2017 2048

Generating RSA private key, 2048 bit long modulus

..........................................+++

.+++

e is 65537 (0x10001)

[root@kstleacbd00133 ~]#

[root@kstleacbd00133 ~]# openssl req -x509 -new -nodes -key /opt/hadoop/security/ca-certs/rootCA.key -days 4000 -out /opt/hadoop/security/ca-certs/rootCA.pem -passin pass:Charter2017 -passout pass:Charter2017 -subj '/C=US/ST=Missouri/L=STL/O=Hortonworks/OU=labhadoop.charter.com/CN=Admin\/emailAddress=DLCORPITITSDBigData@charter.com'

[root@kstleacbd00133 ~]#

[root@kstleacbd00133 ~]# /usr/jdk64/jdk1.8.0\_112/bin/keytool -genkey -alias kstleacbd00133.labhadoop.charter.com -keyalg RSA -keystore /opt/hadoop/security/jks/kstleacbd00133.labhadoop.charter.com-keystore.js -keysize 2048 \

> -dname "CN=kstleacbd00133.labhadoop.charter.com, OU=labhadoop,O=charter.com,L=STL, S=Missouri, C=US" -storepass Charter2017 -keypass Charter2017

[root@kstleacbd00133 ~]#

[root@kstleacbd00133 ~]# /usr/jdk64/jdk1.8.0\_112/bin/keytool -importkeystore -srckeystore /opt/hadoop/security/jks/kstleacbd00133.labhadoop.charter.com-keystore.js -srcstorepass Charter2017 -srckeypass Charter2017 -destkeystore /opt/hadoop/security/tmp/kstleacbd00133.labhadoop.charter.com-keystore.p12 -deststoretype PKCS12 -srcalias kstleacbd00133.labhadoop.charter.com -deststorepass Charter2017 -destkeypass Charter2017

[root@kstleacbd00133 ~]# openssl pkcs12 -in /opt/hadoop/security/tmp/kstleacbd00133.labhadoop.charter.com-keystore.p12 -passin pass:Charter2017 \

> -nocerts -out /opt/hadoop/security/x509/kstleacbd00133.labhadoop.charter.com-unsignedkey.pem -passout pass:Charter2017

MAC verified OK

[root@kstleacbd00133 ~]# /usr/jdk64/jdk1.8.0\_112/bin/keytool -certreq -alias kstleacbd00133.labhadoop.charter.com \

> -keystore /opt/hadoop/security/jks/kstleacbd00133.labhadoop.charter.com-keystore.jks -file /opt/hadoop/security/certs/kstleacbd00133.labhadoop.charter.com-keystore.csr -storepass Charter2017 -keypass Charter2017

[root@kstleacbd00133 ~]#

[root@kstleacbd00133 ~]# openssl x509 -req -in /opt/hadoop/security/certs/kstleacbd00133.labhadoop.charter.com-keystore.csr -CA /opt/hadoop/security/ca-certs/rootCA.pem -CAkey /opt/hadoop/security/ca-certs/rootCA.key -CAcreateserial -out /opt/hadoop/security/certs/kstleacbd00133.labhadoop.charter.com-keystore.pem -days 4000 -passin pass:Charter2017

Signature ok

subject=/C=US/ST=Missouri/L=STL/O=charter.com/OU=labhadoop/CN=kstleacbd00133.labhadoop.charter.com

Getting CA Private Key

[root@kstleacbd00133 ~]#

[root@kstleacbd00133 ~]# cp /usr/jdk64/jdk1.8.0\_112/jre/lib/security/cacerts /opt/hadoop/security/truststore/jssecacerts

[root@kstleacbd00133 ~]# /usr/jdk64/jdk1.8.0\_112/bin/keytool -noprompt -importcert -trustcacerts -alias rootCA -file /opt/hadoop/security/ca-certs/rootCA.pem -keystore /opt/hadoop/security/truststore/jssecacerts -storepass changeit

Certificate was added to keystore

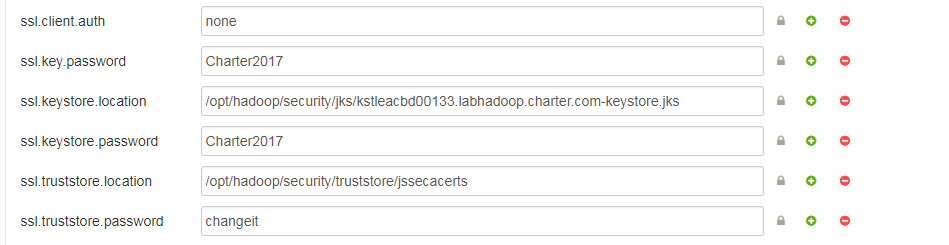
[root@kstleacbd00133 ~]# /usr/jdk64/jdk1.8.0\_112/bin/keytool -noprompt -importcert -trustcacerts -alias rootCA -file /opt/hadoop/security/ca-certs/rootCA.pem -keystore /opt/hadoop/security/jks/kstleacbd00133.labhadoop.charter.com-keystore.jks -storepass Charter2017 -keypass Charter2017

Certificate was added to keystore

Kafka settings:







security.inter.broker.protocol=SASL\_PLAINTEXT

ssl.client.auth=none

ssl.key.password=Charter2017

ssl.keystore.location=/opt/hadoop/security/jks/kstleacbd00133.labhadoop.charter.com-keystore.jks

ssl.keystore.password=Charter2017

ssl.truststore.location=/opt/hadoop/security/truststore/jssecacerts

ssl.truststore.password=changeit

WATCHER::

WatchedEvent state:SyncConnected type:None path:null

2017-10-27 16:37:09,567 - ERROR [main-SendThread(localhost:2181):ZooKeeperSaslClient@384] - An error: (java.security.PrivilegedActionException: javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Server not found in Kerberos database (7))]) occurred when evaluating Zookeeper Quorum Member's received SASL token. Zookeeper Client will go to AUTH\_FAILED state.

2017-10-27 16:37:09,568 - ERROR [main-SendThread(localhost:2181):ClientCnxn$SendThread@1059] - SASL authentication with Zookeeper Quorum member failed: javax.security.sasl.SaslException: An error: (java.security.PrivilegedActionException: javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Server not found in Kerberos database (7))]) occurred when evaluating Zookeeper Quorum Member's received SASL token. Zookeeper Client will go to AUTH\_FAILED state.

WATCHER::

WatchedEvent state:AuthFailed type:None path:null

[root@kstleacbd00133 ~]# pwd

/root

[root@kstleacbd00133 ~]# cat client\_jaas.conf

[root@kstleacbd00133 ~]# cat client\_jaas.conf

Client {

com.sun.security.auth.module.Krb5LoginModule required

useKeyTab=true

keyTab=/etc/security/keytabs/zk.service.keytab

storeKey=true

useTicketCache=false

principal=zookeeper/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM;

};

[root@kstleacbd00133 ~]# export JVMFLAGS="-Djava.security.auth.login.config=/root/client\_jaas.conf"

[root@kstleacbd00133 ~]# /usr/hdf/current/zookeeper-server/bin/zk

zkCleanup.sh zkCli.sh zkEnv.sh zkServer-initialize.sh zkServer.sh

[root@kstleacbd00133 ~]# /usr/hdf/current/zookeeper-server/bin/zkCli.sh

[root@kstleacbd00133 ~]# /usr/hdf/current/kafka-broker/bin/kafka-topics.sh --zookeeper kstleacbd00133:2181 --create --topic TestTopic --partitions 1 --replication-factor 1

Created topic "TestTopic".

[root@kstleacbd00133 ~]# /usr/hdf/current/kafka-broker/bin/kafka-topics.sh --zookeeper kstleacbd00133:2181 --list

TestTopic

ambari\_kafka\_service\_check

[root@kstleacbd00133 ~]# /usr/hdf/current/kafka-broker/bin/kafka-topics.sh --zookeeper kstleacbd00133:2181 --describe TestTopic

Topic:TestTopic PartitionCount:1 ReplicationFactor:1 Configs:

Topic: TestTopic Partition: 0 Leader: 1001 Replicas: 1001 Isr: 1001

Topic:ambari\_kafka\_service\_check PartitionCount:1 ReplicationFactor:1 Configs:

Topic: ambari\_kafka\_service\_check Partition: 0 Leader: 1001 Replicas: 1001 Isr: 1001

[root@kstleacbd00133 ~]# klist -l

Principal name Cache name

-------------- ----------

mmanchala@LABHADOOP.CHARTER.CO FILE:/tmp/krb5cc\_0

[root@kstleacbd00133 ~]# klist -l kafka.service.keytab

Principal name Cache name

-------------- ----------

mmanchala@LABHADOOP.CHARTER.CO FILE:/tmp/krb5cc\_0

[root@kstleacbd00133 ~]# klist -l /etc/security/keytabs/kafka.service.keytab

Principal name Cache name

-------------- ----------

mmanchala@LABHADOOP.CHARTER.CO FILE:/tmp/krb5cc\_0

[root@kstleacbd00133 ~]# destroy

-bash: destroy: command not found

[root@kstleacbd00133 ~]# kdestroy

[root@kstleacbd00133 ~]# kinit -k -l /etc/security/keytabs/kafka.service.keytab

Bad lifetime value /etc/security/keytabs/kafka.service.keytab

Usage: kinit [-V] [-l lifetime] [-s start\_time]

[-r renewable\_life] [-f | -F | --forwardable | --noforwardable]

[-p | -P | --proxiable | --noproxiable]

-n [-a | -A | --addresses | --noaddresses]

[--request-pac | --no-request-pac]

[-C | --canonicalize]

[-E | --enterprise]

[-v] [-R] [-k [-i|-t keytab\_file]] [-c cachename]

[-S service\_name] [-T ticket\_armor\_cache]

[-X <attribute>[=<value>]] [principal]

options:

-V verbose

-l lifetime

-s start time

-r renewable lifetime

-f forwardable

-F not forwardable

-p proxiable

-P not proxiable

-n anonymous

-a include addresses

-A do not include addresses

-v validate

-R renew

-C canonicalize

-E client is enterprise principal name

-k use keytab

-i use default client keytab (with -k)

-t filename of keytab to use

-c Kerberos 5 cache name

-S service

-T armor credential cache

-X <attribute>[=<value>]

[root@kstleacbd00133 ~]# klist -k -l /etc/security/keytabs/kafka.service.keytab

Usage: klist [-e] [-V] [[-c] [-l] [-A] [-d] [-f] [-s] [-a [-n]]] [-k [-t] [-K]] [name]

-c specifies credentials cache

-k specifies keytab

(Default is credentials cache)

-i uses default client keytab if no name given

-l lists credential caches in collection

-A shows content of all credential caches

-e shows the encryption type

-V shows the Kerberos version and exits

options for credential caches:

-d shows the submitted authorization data types

-f shows credentials flags

-s sets exit status based on valid tgt existence

-a displays the address list

-n do not reverse-resolve

options for keytabs:

-t shows keytab entry timestamps

-K shows keytab entry keys

[root@kstleacbd00133 ~]# klist -l /etc/security/keytabs/kafka.service.keytab

Principal name Cache name

-------------- ----------

[root@kstleacbd00133 ~]# klist -K

Usage: klist [-e] [-V] [[-c] [-l] [-A] [-d] [-f] [-s] [-a [-n]]] [-k [-t] [-K]] [name]

-c specifies credentials cache

-k specifies keytab

(Default is credentials cache)

-i uses default client keytab if no name given

-l lists credential caches in collection

-A shows content of all credential caches

-e shows the encryption type

-V shows the Kerberos version and exits

options for credential caches:

-d shows the submitted authorization data types

-f shows credentials flags

-s sets exit status based on valid tgt existence

-a displays the address list

-n do not reverse-resolve

options for keytabs:

-t shows keytab entry timestamps

-K shows keytab entry keys

[root@kstleacbd00133 ~]# klist -K /etc/security/keytabs/kafka.service.keytab

Usage: klist [-e] [-V] [[-c] [-l] [-A] [-d] [-f] [-s] [-a [-n]]] [-k [-t] [-K]] [name]

-c specifies credentials cache

-k specifies keytab

(Default is credentials cache)

-i uses default client keytab if no name given

-l lists credential caches in collection

-A shows content of all credential caches

-e shows the encryption type

-V shows the Kerberos version and exits

options for credential caches:

-d shows the submitted authorization data types

-f shows credentials flags

-s sets exit status based on valid tgt existence

-a displays the address list

-n do not reverse-resolve

options for keytabs:

-t shows keytab entry timestamps

-K shows keytab entry keys

[root@kstleacbd00133 ~]# klist -t -K /etc/security/keytabs/kafka.service.keytab

Usage: klist [-e] [-V] [[-c] [-l] [-A] [-d] [-f] [-s] [-a [-n]]] [-k [-t] [-K]] [name]

-c specifies credentials cache

-k specifies keytab

(Default is credentials cache)

-i uses default client keytab if no name given

-l lists credential caches in collection

-A shows content of all credential caches

-e shows the encryption type

-V shows the Kerberos version and exits

options for credential caches:

-d shows the submitted authorization data types

-f shows credentials flags

-s sets exit status based on valid tgt existence

-a displays the address list

-n do not reverse-resolve

options for keytabs:

-t shows keytab entry timestamps

-K shows keytab entry keys

[root@kstleacbd00133 ~]# klist -k /etc/security/keytabs/kafka.service.keytab

Keytab name: FILE:/etc/security/keytabs/kafka.service.keytab

KVNO Principal

---- --------------------------------------------------------------------------

0 kafka/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM

0 kafka/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM

0 kafka/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM

0 kafka/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM

0 kafka/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM

[root@kstleacbd00133 ~]# pwd

/root

[root@kstleacbd00133 ~]# vi client\_jaas.conf

[root@kstleacbd00133 ~]# cat producer.properties

security.protocol=SASL\_PLAINTEXT

sasl.kerberos.service.name=kafka

ssl.truststore.location=/opt/hadoop/security/truststore/jssecacerts

ssl.truststore.password=changeit

[root@kstleacbd00133 ~]# cat consumer.properties

security.protocol=SASL\_PLAINTEXT

#security.protocol=SASL\_SSL

sasl.kerberos.service.name=kafka

ssl.truststore.location=/opt/hadoop/security/truststore/jssecacerts

ssl.truststore.password=changeit

group.id=TestGroup

[root@kstleacbd00133 ~]# cat client\_jaas.conf

Client {

com.sun.security.auth.module.Krb5LoginModule required

useKeyTab=true

keyTab=/etc/security/keytabs/zk.service.keytab

storeKey=true

useTicketCache=false

principal=zookeeper/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM;

};

KafkaClient {

com.sun.security.auth.module.Krb5LoginModule required

doNotPrompt=true

useTicketCache=false

useKeyTab=true

storeKey=true

serviceName="kafka"

keyTab="/etc/security/keytabs/kafka.service.keytab"

principal="kafka/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM"

client=true;

};

Client {

com.sun.security.auth.module.Krb5LoginModule required

doNotPrompt=true

useTicketCache=false

useKeyTab=true

storeKey=true

serviceName="kafka"

keyTab="/etc/security/keytabs/kafka.service.keytab"

principal="kafka/kstleacbd00133.labhadoop.charter.com@LABHADOOP.CHARTER.COM"

client=true;

};

[root@kstleacbd00133 ~]#