
# Rapport de projet sécurité :
# Partage des fichiers avec authentification kerberos

Ichraf Ben Fadhel
Insaf Khorchani
Anwar Ghammam

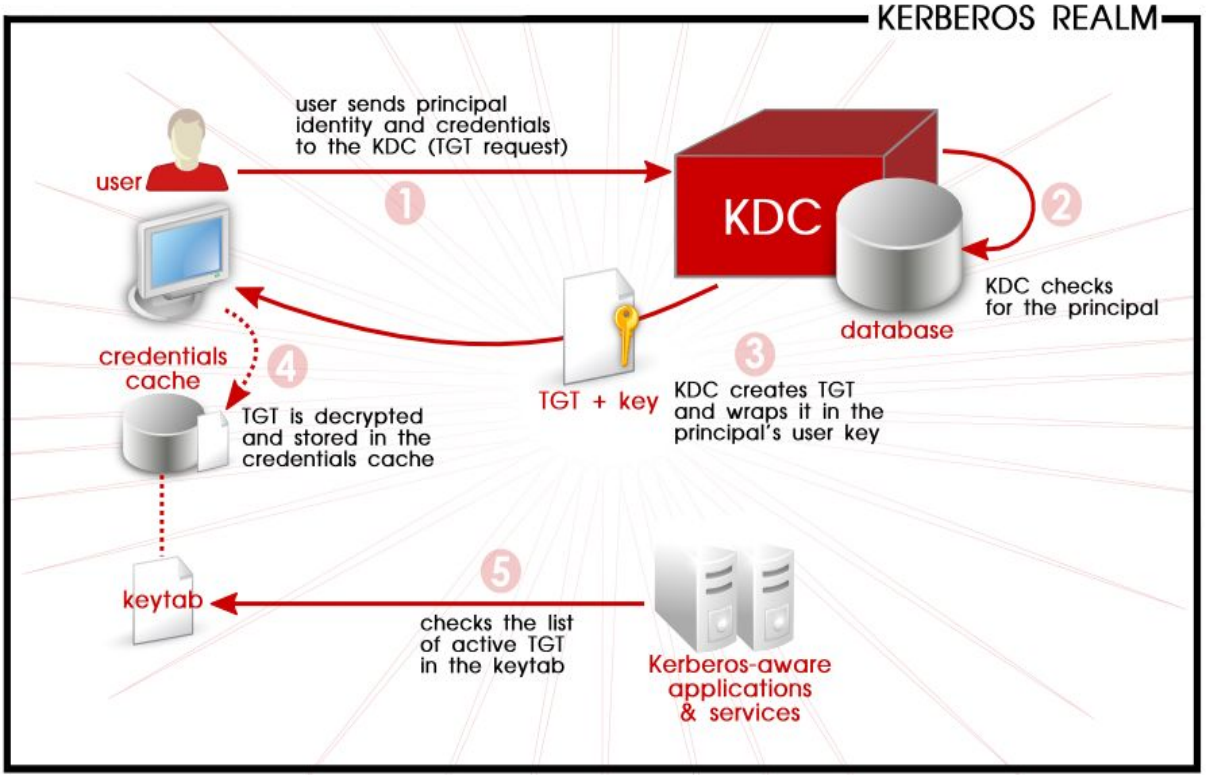
**13/05/2020**

# I. Introduction :

Dans ce projet on a réalisé un système de partage des fichiers sécurisé en utilisant le protocole NFS (Network file system) avec une authentification kerberos .

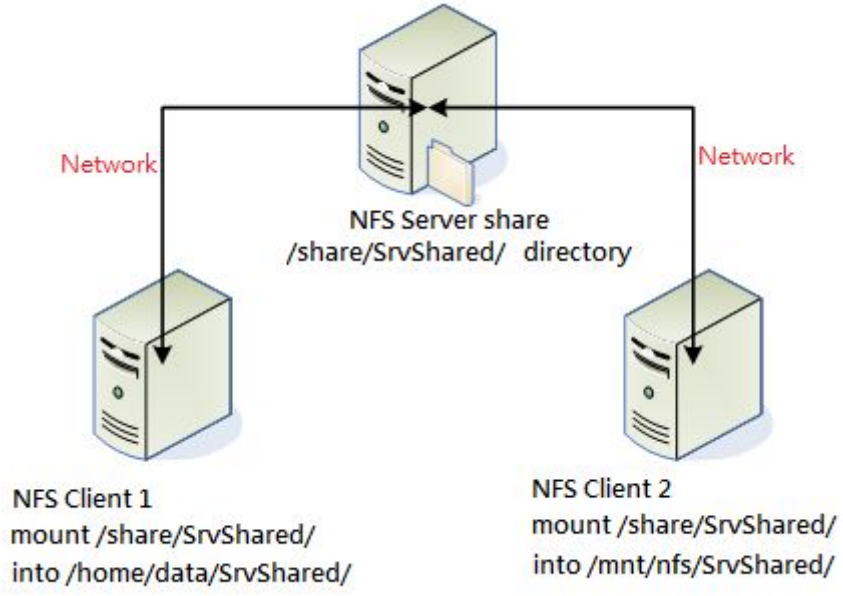
## 1. protocole kerberos :

Kerberos est un protocole d'authentification qui se base sur une tierce partie de confiance pour gérer l'authentification, le **KDC** (Key Distribution Center). Tous les utilisateurs et services du réseau font confiance à cette tierce partie. Pour réaliser la sécurité de l'authentification, Kerberos utilise un mécanisme de chiffrement basé sur des algorithmes à clef symétrique. Chaque sujet et service du réseau a une clé secrète partagée avec le KDC. Kerberos utilise un système de **ticket** pour réaliser l'authentification et introduit le principe de SSO (Single Sign On). L'utilisateur s'authentifie sur le **KDC** puis utilise un **ticket** pour s'authentifier sur chaque service demandé. L'utilisateur ne transmet jamais son mot de passe au service.



## 2. Le protocole Network file system

NFS, pour Network File System, est un protocole permettant à une machine d'accéder ,stocker et de mettre à jour des fichiers sur une machine distante .




# II. Travail réalisé :

## 1. Création et configuration des machine

Dans ce travail on a utilisé 3 instances EC2 fourni par aws

| Type d'instance | Zone de disponib | État de l'instanc | Contrôles des s | Statut des alarm | DNS public (IPv4) | IP publique IPv4 |
|---|---|---|---|---|---|---|
| t2.micro | us-east-1c | running | 2/2 contrôle... | *Aucun(e)* | ec2-3-88-174-244.com... | 3.88.174.244 |
| t2.micro | us-east-1c | running | 2/2 contrôle... | *Aucun(e)* | ec2-52-207-209-119.co... | 52.207.209.119 |
| t2.micro | us-east-1c | running | 2/2 contrôle... | *Aucun(e)* | ec2-54-172-27-106.co... | 54.172.27.106 |

il faut s'assurer que les protocoles TCP ,UDP sont actives car certains services qu'on va l'utiliser ont besoin de ces protocoles , donc on a configuré le groupe de sécurité comme ceci

**Toutes les règles de trafic entrant sélectionnées des groupes de sécurité**

| Type | Protocole | Plage de ports | Source |
|---|---|---|---|
| Tous les TCP | TCP | 0 - 65535 | 0.0.0.0/0 |
| SSH | TCP | 22 | 0.0.0.0/0 |
| Tous les UDP | UDP | 0 - 65535 | 0.0.0.0/0 |
| Règle ICMP personnalisée - IPv4 | Demande d'écho | N/A | 0.0.0.0/0 |

La règle ICMP personnalisée (demande d'écho) c'est pour pouvoir tester le ping par la suite .
Sur chacune de ces machine on modifie le fichier /etc/hosts on spécifie les hostname suivants :

```
3.88.174.244 kbserver.insat.tn kbserver
52.207.209.119 nfsserver.insat.tn nfsserver
54.172.27.106 nfsclient.insat.tn nfsclient
```

puis sur chaque machine on modifie sa propre hostname  puis en test via ping si la résolution est réussie

a. Machine kerberos

```
[root@ip-172-31-85-4 ec2-user]# hostnamectl set-hostname kbserver.insat.tn
[root@ip-172-31-85-4 ec2-user]# ping -c 3 $(hostname)
PING kbserver.insat.tn (3.88.174.244) 56(84) bytes of data.
64 bytes from kbserver.insat.tn (3.88.174.244): icmp_seq=1 ttl=254 time=0.420 ms
64 bytes from kbserver.insat.tn (3.88.174.244): icmp_seq=2 ttl=254 time=0.504 ms
64 bytes from kbserver.insat.tn (3.88.174.244): icmp_seq=3 ttl=254 time=0.492 ms

--- kbserver.insat.tn ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.420/0.472/0.504/0.037 ms
```

b. machine client NFS

```
[root@ip-172-31-83-229 ec2-user]# hostnamectl set-hostname nfsclient.insat.tn
[root@ip-172-31-83-229 ec2-user]# ping -c 3 $(hostname)
PING nfsclient.insat.tn (54.172.27.106) 56(84) bytes of data.
64 bytes from nfsclient.insat.tn (54.172.27.106): icmp_seq=1 ttl=254 time=0.375 ms
64 bytes from nfsclient.insat.tn (54.172.27.106): icmp_seq=2 ttl=254 time=0.483 ms
64 bytes from nfsclient.insat.tn (54.172.27.106): icmp_seq=3 ttl=254 time=0.466 ms

--- nfsclient.insat.tn ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.375/0.441/0.483/0.050 ms
```

c. machine serveur NFS

```
[root@ip-172-31-88-231 ec2-user]# hostnamectl set-hostname nfsserver.insat.tn
[root@ip-172-31-88-231 ec2-user]# hostname -f
nfsserver.insat.tn
[root@ip-172-31-88-231 ec2-user]# ping -c 3 $(hostname)
PING nfsserver.insat.tn (52.207.209.119) 56(84) bytes of data.
64 bytes from nfsserver.insat.tn (52.207.209.119): icmp_seq=1 ttl=254 time=0.453 ms
64 bytes from nfsserver.insat.tn (52.207.209.119): icmp_seq=2 ttl=254 time=0.443 ms
64 bytes from nfsserver.insat.tn (52.207.209.119): icmp_seq=3 ttl=254 time=0.486 ms

--- nfsserver.insat.tn ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.443/0.460/0.486/0.030 ms
```

## 2. configuration de KDC

La machine kbserver va jouer le rôle de serveur kerberos

### a. installer les packages nécessaires

```
[root@kbserver ec2-user]# yum install -y krb5-server krb5-workstation pam_krb5
```

### b. configuration de serveur kerberos

Dans le fichier /etc/krb5.conf , on definit “INSAT.TN” comme étant le nom de domain et le nom de realm par défaut , et on spécifie pour cette realm le serveur kerberos et le serveur administratif qui est ici kbserver.insat.tn

```
[libdefaults]
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 rdns = false
 pkinit_anchors = /etc/pki/tls/certs/ca-bundle.crt
 default_realm = INSAT.TN
 default_ccache_name = KEYRING:persistent:%{uid}

[realms]
 INSAT.TN = {
  kdc = kbserver.insat.tn
  admin_server = kbserver.insat.tn
 }

[domain_realm]
 .insat.tn = INSAT.TN
 insat.tn = INSAT.TN
```

### c. Configuration de fichier `/var/kerberos/krb5kdc/`kadm5.acl

Ce fichier accorde tous les privilèges à l'administrateur

```
[root@kbserver ec2-user]# cat /var/kerberos/krb5kdc/kadm5.acl
*/admin@INSAT.TN        *
```

d. Configuration de fichier `/var/kerberos/krb5kdc/`kdc.conf :

on spécifie notre realm “INSAT.TN”

```
[kdcdefaults]
 kdc_ports = 88
 kdc_tcp_ports = 88

[realms]
 INSAT.TN = {
  master_key_type = aes256-cts
  acl_file = /var/kerberos/krb5kdc/kadm5.acl
  dict_file = /usr/share/dict/words
  default_principal_flags = +preauth
  admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
  supported_enctypes = aes256-cts:normal aes128-cts:normal des3-hmac-sha1:normal
 arcfour-hmac:normal camellia256-cts:normal camellia128-cts:normal des-hmac-sha1
:normal des-cbc-md5:normal des-cbc-crc:normal
 }
```

e. Création de la base de donnée de KDC

Lors de création de la base de donnée KDC nous aurons besoin de pool d'entropie de données aléatoires donc il faut s’assurer que le service **rgnd** fonctionne .

```
[root@kbserver ec2-user]# kdb5_util create -s -r INSAT.TN
Loading random data
Initializing database '/var/kerberos/krb5kdc/principal' for realm 'INSAT.TN',
master key name 'K/M@INSAT.TN'
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:
```

f. Démarrer et activer Kerberos :

Notre serveur kerberos est prêt donc on démarre et on active le service

```
[root@kbserver ec2-user]# systemctl start krb5kdc kadmin
[root@kbserver ec2-user]# systemctl enable krb5kdc kadmin
Created symlink from /etc/systemd/system/multi-user.target.wants/krb5kdc.service
 to /usr/lib/systemd/system/krb5kdc.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/kadmin.service
to /usr/lib/systemd/system/kadmin.service.
[root@kbserver ec2-user]# systemctl status krb5kdc kadmin
● krb5kdc.service - Kerberos 5 KDC
   Loaded: loaded (/usr/lib/systemd/system/krb5kdc.service; enabled; vendor pres
et: disabled)
   Active: active (running) since ث 2020-05-12 20:46:27 UTC; 18s ago
 Main PID: 308 (krb5kdc)
   CGroup: /system.slice/krb5kdc.service
           └─308 /usr/sbin/krb5kdc -P /var/run/krb5kdc.pid

ماي 12 20:46:27 kbserver.insat.tn systemd[1]: Starting Kerberos 5 KDC...
ماي 12 20:46:27 kbserver.insat.tn systemd[1]: Started Kerberos 5 KDC.

● kadmin.service - Kerberos 5 Password-changing and Administration
   Loaded: loaded (/usr/lib/systemd/system/kadmin.service; enabled; vendor prese
t: disabled)
   Active: active (running) since ث 2020-05-12 20:46:27 UTC; 18s ago
 Main PID: 309 (kadmind)
   CGroup: /system.slice/kadmin.service
           └─309 /usr/sbin/kadmind -P /var/run/kadmind.pid
```

g. Création des principaux

on ajoute notre admin “root/admin@INSAT.TN”

```
[root@kbserver ec2-user]# kadmin.local
Authenticating as principal root/admin@INSAT.TN with password.
kadmin.local:  addprinc root/admin
WARNING: no policy specified for root/admin@INSAT.TN; defaulting to no policy
Enter password for principal "root/admin@INSAT.TN":
Re-enter password for principal "root/admin@INSAT.TN":
Principal "root/admin@INSAT.TN" created.
```

on ajoute un utilisateur pour tester après notre configuration

```
kadmin.local:  addprinc testuser
WARNING: no policy specified for testuser@INSAT.TN; defaulting to no policy
Enter password for principal "testuser@INSAT.TN":
Re-enter password for principal "testuser@INSAT.TN":
Principal "testuser@INSAT.TN" created.
```

Nous allons maintenant ajouter l'hôte kbserver.insat.tn afin qu'il puisse héberger des services Kerberose. Nous devons ajouter des principaux pour chaque utilisateur et serveur qui utilisent Kerberos.

```
kadmin.local:  addprinc -randkey host/kbserver.insat.tn
WARNING: no policy specified for host/kbserver.insat.tn@INSAT.TN; defaulting to no policy
Principal "host/kbserver.insat.tn@INSAT.TN" created.
kadmin.local:  ktadd host/kbserver.insat.tn
Entry for principal host/kbserver.insat.tn with kvno 2, encryption type aes256-cts-hmac-sha
1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kbserver.insat.tn with kvno 2, encryption type aes128-cts-hmac-sha
1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kbserver.insat.tn with kvno 2, encryption type des3-cbc-sha1 added
 to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kbserver.insat.tn with kvno 2, encryption type arcfour-hmac added
to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kbserver.insat.tn with kvno 2, encryption type camellia256-cts-cma
c added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kbserver.insat.tn with kvno 2, encryption type camellia128-cts-cma
c added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kbserver.insat.tn with kvno 2, encryption type des-hmac-sha1 added
 to keytab FILE:/etc/krb5.keytab.
Entry for principal host/kbserver.insat.tn with kvno 2, encryption type des-cbc-md5 added t
o keytab FILE:/etc/krb5.keytab.
```

h. Configuration SSH

On modifie le fichier /etc/ssh/ssh_config pour autoriser les clients ssh à utiliser l’authentification kerberos

```
    GSSAPIAuthentication yes
    GSSAPIDelegateCredentials yes
```

puis on redémarre le service sshd et autoriser l’authentification kerberos

```
[root@kbserver ec2-user]# systemctl reload sshd
[root@kbserver ec2-user]# authconfig --enablekrb5 --update
[root@kbserver ec2-user]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since ث 2020-05-12 19:31:33 UTC; 1h 50min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 576 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
 Main PID: 3409 (sshd)
   CGroup: /system.slice/sshd.service
           └─3409 /usr/sbin/sshd -D
```

i. Teste d'authentification kerberos :

on se connecte en tant que testuser pour pouvoir tester l'authentification , avec kinit on obtient un ticket de validité 24 h pour le principal testuser puis on accède avec ssh au serveur kerberos , puisqu'on a le ticket aucune mot de passe n'est demandé , on voit bien que notre configuration a réussi

```
[root@kbserver ec2-user]# su - testuser
[testuser@kbserver ~]$ kinit
Password for testuser@INSAT.TN:
[testuser@kbserver ~]$ klist
Ticket cache: KEYRING:persistent:1001:1001
Default principal: testuser@INSAT.TN

Valid starting       Expires              Service principal
05/12/2020 21:27:41  05/13/2020 21:27:31  krbtgt/INSAT.TN@INSAT.TN
```

```
[testuser@kbserver ~]$ ssh kbserver.insat.tn
The authenticity of host 'kbserver.insat.tn (3.88.174.244)' can't be established.
ECDSA key fingerprint is SHA256:wj3YOkq+grKqmWRgsyHdbHNnkKixrxmTZnQ6gH0jrEc.
ECDSA key fingerprint is MD5:39:70:82:c8:ef:34:f3:4b:f6:9a:42:2a:76:78:70:d0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'kbserver.insat.tn,3.88.174.244' (ECDSA) to the list of known ho
sts.
Last login: Tue May 12 21:27:24 2020

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 11 available
Run "sudo yum update" to apply all updates.
```

## 2. Configuration de nfs-server :

Dans cette partie on configure le serveur NFS , indépendamment de toute configuration de kerberos. Après installation des packages nécessaires on  démarre et on active nfs server

a. Démarrage et activation de serveur nfs

```
systemctl enable rpcbind nfs-server
systemctl start rpcbind nfs-server
```

b. Création de répertoire à exporter :

Cette répertoire va contenir les fichiers exportés

```
[root@nfsserver ec2-user]# mkdir -p /home/ichraf-docs
[root@nfsserver ec2-user]# chmod 0777 /home/ichraf-docs
```

c. Installation et configuration SELinux (Security Enhanced Linux) :

SELinux (Security Enhanced Linux) fournit un contrôle d'accès obligatoire au système d'exploitation Linux

```
[root@nfsserver ec2-user]#  curl https://rpmfind.net/linux/centos/7.8.2003/os/x86_64/Packages/setroubleshoot-server-3.2.30-8.el7
.x86_64.rpm --output setroubleshoot-server-3.2.30-8.el7.x86_64.rpm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  389k  100  389k    0     0   430k      0 --:--:-- --:--:-- --:--:--  429k
[root@nfsserver ec2-user]# wget http://mirror.centos.org/centos/7/os/x86_64/Packages/setroubleshoot-plugins-3.0.67-4.el7.noarch.
rpm
```

```
[root@nfsserver ec2-user]# yum install setroubleshoot-plugins-3.0.67-4.el7.noarch.rpm setroubleshoot-server-3.2.30-8.el7.x86_64.
rpm
```

on configure le répertoire qui va contenir les fichiers exportés en mode  lecture écriture pour pouvoir ajouter et accéder au fichier

```
[root@nfsserver ec2-user]# semanage fcontext -a -t public_content_rw_t
"/home/ichraf-docs(/.*)?"
[root@nfsserver ec2-user]#  restorecon -R /home/ichraf-docs
```

```
[root@nfsserver ec2-user]#  setsebool -P nfs_export_all_rw on
[root@nfsserver ec2-user]# setsebool -P use_nfs_home_dirs on
```

on redémarre le serveur NFS et on vérifie notre configuration

```
[root@nfsserver ec2-user]#  systemctl restart nfs-server
[root@nfsserver ec2-user]# showmount -e localhost
Export list for localhost:
/home/ichraf-docs nfsclient.insat.tn
```

## 3. configuration de client NFS

Il faut juste installer ce package sur la machine client nfs

```
[root@nfsclient ec2-user]# yum install -y nfs-utils
```

## 4. configuration de nfs-server en tant que client kerberos :

jusqu'à ici on a configuré le serveur kerberos un serveur nfs et un client nfs , maintenant on configure le client et le serveur nfs en tant que client kerberos .

a. Installation des packages

```
[root@nfsserver ec2-user]# yum install -y krb5-workstation pam_krb5
```

b. Configuration de fichier /etc/krb5.conf :

On configure ce fichier en précisant le nom de domain , le realm et le serveur kerberos , ce fichier est identique à celui de serveur kerberos

c. Création des principaux :

```
kadmin:  addprinc -randkey host/nfsserver.insat.tn
WARNING: no policy specified for host/nfsserver.insat.tn@INSAT.TN; defaulting to no policy
Principal "host/nfsserver.insat.tn@INSAT.TN" created.
kadmin:   ktadd host/nfsserver.insat.tn
Entry for principal host/nfsserver.insat.tn with kvno 2, encryption type aes256-cts-hmac-sha1-96 added
 to keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfsserver.insat.tn with kvno 2, encryption type aes128-cts-hmac-sha1-96 added
 to keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfsserver.insat.tn with kvno 2, encryption type des3-cbc-sha1 added to keytab
 FILE:/etc/krb5.keytab.
Entry for principal host/nfsserver.insat.tn with kvno 2, encryption type arcfour-hmac added to keytab
FILE:/etc/krb5.keytab.
Entry for principal host/nfsserver.insat.tn with kvno 2, encryption type camellia256-cts-cmac added to
 keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfsserver.insat.tn with kvno 2, encryption type camellia128-cts-cmac added to
 keytab FILE:/etc/krb5.keytab.
Entry for principal host/nfsserver.insat.tn with kvno 2, encryption type des-hmac-sha1 added to keytab
 FILE:/etc/krb5.keytab.
Entry for principal host/nfsserver.insat.tn with kvno 2, encryption type des-cbc-md5 added to keytab F
ILE:/etc/krb5.keytab.
kadmin:  quit
```

d. Configuration de SSH :

on configure le fichier /etc/ssh/ssh_config et on change

```
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
```

puis on redémarre le service sshd et on autorise l'authentification kerberos .

```
[root@nfsserver ec2-user]# systemctl reload sshd
[root@nfsserver ec2-user]#  authconfig --enablekrb5 --update
```

e. Test :

on teste l'authentification kerberos sur cette machine pour vérifier que le configuration est correct et fonctionne , avec l'utilisateur **testuser** qu'on le crée sur cette machine et il est déjà ajouter au principal on teste l'authentification , on peut accéder à la machine serveur kerberos avec le ticket crée par kinit et aucune mot de passe n'est demandé .

```
[root@nfsserver ec2-user]# su - testuser
[testuser@nfsserver ~]$ kinit
Password for testuser@INSAT.TN:
[testuser@nfsserver ~]$ ssh kbserver.insat.tn
The authenticity of host 'kbserver.insat.tn (3.88.174.244)' can't be established.
ECDSA key fingerprint is SHA256:wj3YOkq+grKqmWRgsyHdbHNnkKixrxmTZnQ6gH0jrEc.
ECDSA key fingerprint is MD5:39:70:82:c8:ef:34:f3:4b:f6:9a:42:2a:76:78:70:d0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'kbserver.insat.tn,3.88.174.244' (ECDSA) to the list of known hosts.
Last login: Tue May 12 23:47:57 2020

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 11 available
Run "sudo yum update" to apply all updates.
[testuser@kbserver ~]$
```

f. Ajout de NFS au principal :

on ajoute le serveur nfs au principal

```
kadmin:  addprinc -randkey nfs/nfsserver.insat.tn
WARNING: no policy specified for nfs/nfsserver.insat.tn@INSAT.TN; defaulting to no policy
Principal "nfs/nfsserver.insat.tn@INSAT.TN" created.
kadmin:   ktadd nfs/nfsserver.inast.tn
kadmin: Principal nfs/nfsserver.inast.tn does not exist.
kadmin:   ktadd nfs/nfsserver.insat.tn
Entry for principal nfs/nfsserver.insat.tn with kvno 2, encryption type aes256-cts-hmac-sha1-96 added
to keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfsserver.insat.tn with kvno 2, encryption type aes128-cts-hmac-sha1-96 added
to keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfsserver.insat.tn with kvno 2, encryption type des3-cbc-sha1 added to keytab
FILE:/etc/krb5.keytab.
Entry for principal nfs/nfsserver.insat.tn with kvno 2, encryption type arcfour-hmac added to keytab F
ILE:/etc/krb5.keytab.
Entry for principal nfs/nfsserver.insat.tn with kvno 2, encryption type camellia256-cts-cmac added to
keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfsserver.insat.tn with kvno 2, encryption type camellia128-cts-cmac added to
keytab FILE:/etc/krb5.keytab.
Entry for principal nfs/nfsserver.insat.tn with kvno 2, encryption type des-hmac-sha1 added to keytab
FILE:/etc/krb5.keytab.
Entry for principal nfs/nfsserver.insat.tn with kvno 2, encryption type des-cbc-md5 added to keytab FI
LE:/etc/krb5.keytab.
kadmin:
```

g. Configuration de fichier /etc/exports :

on modifie le fichier /etc:exports en ajoutant sec=krb5 pour que l'utilisateur doit être authentifié avec kerberos pour pouvoir envoyer les fichiers au serveur.

```
[root@nfsserver ec2-user]# cat /etc/exports
/home/ichraf-docs nfsclient.insat.tn(rw,no_root_squash,sec=krb5)
```

h. Redémarrage de serveur :

```
[root@nfsserver ec2-user]#  systemctl restart nfs-server
[root@nfsserver ec2-user]# showmount -e localhost
Export list for localhost:
/home/ichraf-docs nfsclient.insat.tn
```

## 5. Configuration de nfs-client en tant que client kerberos :

on repete les memes etapes (a=>f) que celle de nfs-server mais cette fois en tant que nfsclient.insat.tn

## 6. Test de l'application :

Finalement on test avec un client non authentifié donc n' a pas un ticket le client il ne peut pas créer un fichier et exporter , après authentification il obtient un ticket et maintenant il a peut créer et exporter des fichiers

```
[root@nfsclient ec2-user]# mount -t nfs4 -o sec=krb5 nfsserver.insat.tn:/home/ic
hraf-docs /mnt
[root@nfsclient ec2-user]# su - testuser
Last login: ي ام ر 13 00:55:43 UTC 2020 on pts/0
[testuser@nfsclient ~]$ klist                          testuser n' a pas une ticket
klist: Credentials cache keyring 'persistent:1002:1002' not found
[testuser@nfsclient ~]$ echo "hello" > /mnt/hello
-bash: /mnt/hello: Permission denied
[testuser@nfsclient ~]$ kinit
Password for testuser@INSAT.TN:
[testuser@nfsclient ~]$ echo "hello" > /mnt/hello
[testuser@nfsclient ~]$ ls /mnt
e  file  hello  i  ichraf  send.txt  tes2  test
[testuser@nfsclient ~]$
```

sur la machine serveur NFS on vérifie que les fichiers créés sont exporté