

Formation Redis

Octobre 2025

Connexion à l'environnement de Labs (Machine virtuelle mutualisée)

Adresse : <http://nbstagevm.southafricanorth.cloudapp.azure.com/guacamole>

Premier login : student / Pa\$\$w0rd

Chaque personne utilisera toujours un profil utilisateur :

Marie : st01 = décalage de port = +1

Dominique : st02 = décalage de port = +2

Samba : st03 = décalage de port = +3

Abdoulaye : st04 = décalage de port = +4

Thierno : st05 = décalage de port = +5

Si économiseur d'écran : mot de passe = **password**

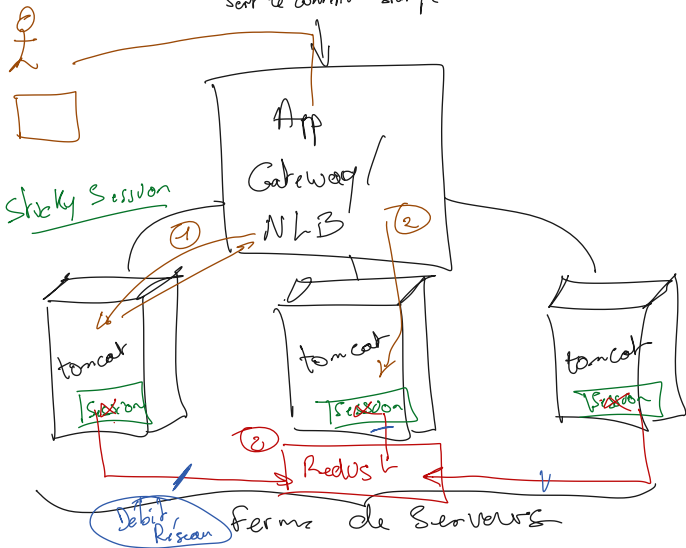
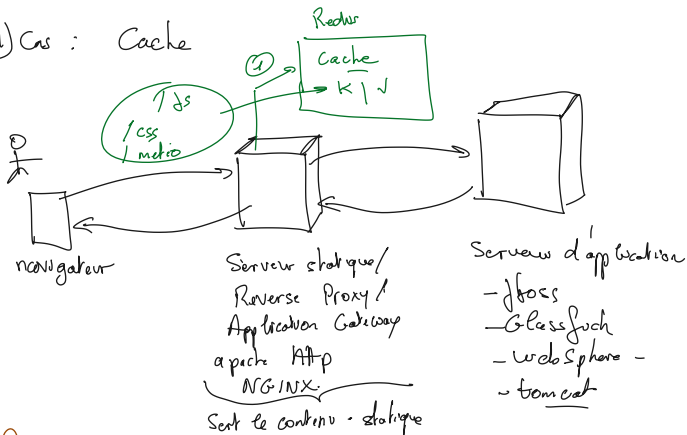
Télécharger le support de cours : <https://we.tl/t-qYlhUls6iN>

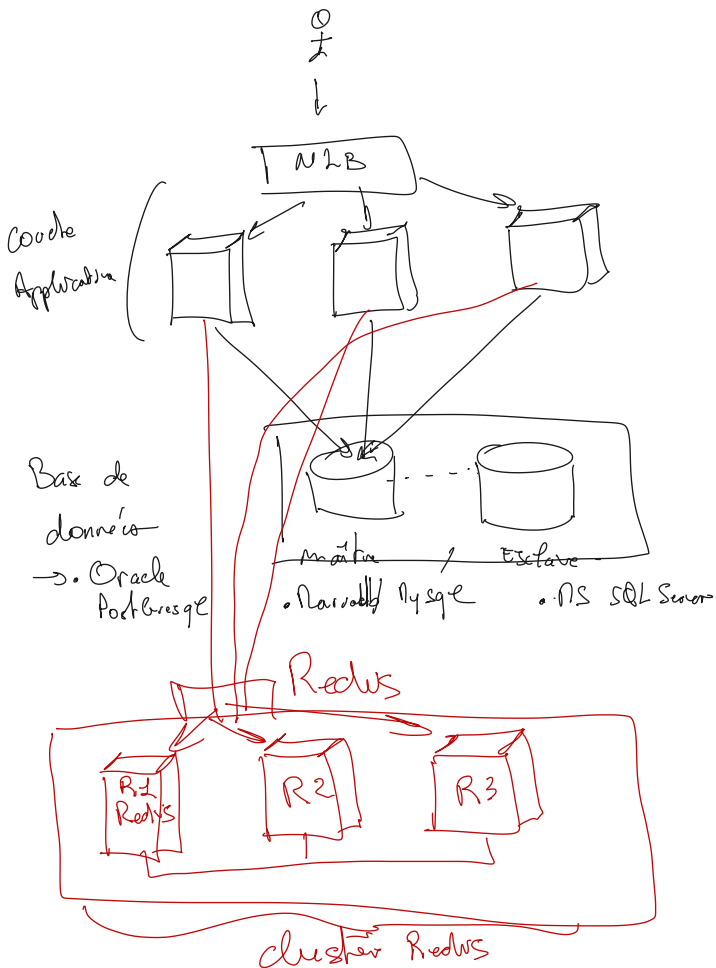
Téléchargez des livres pour Redis : <https://we.tl/t-xvpjOTpZnT>

Si besoin : philippe@nboost.eu

Utilisation de Redis.

1) Cas : Cache





Concept de base du dictionnaire :

clé $\rightarrow$ Valeur

Key/Value

sauf que ça pourra être beaucoup plus poussé....

$\rightarrow$ clé $\rightarrow$ collection

compteurs

Auto supprimable (Time To Live)

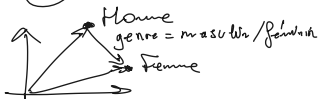
Type entier, chaîne, binaire, json,

Homme $\rightarrow$ mot français $\rightarrow$ 5 (FR) 1 (US)



femme $\rightarrow$

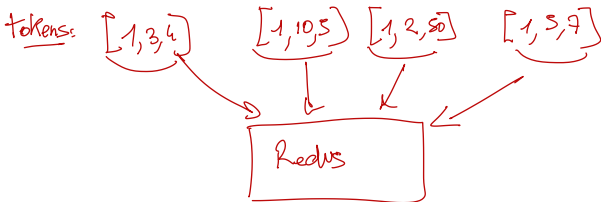
fémin $\rightarrow$ 2

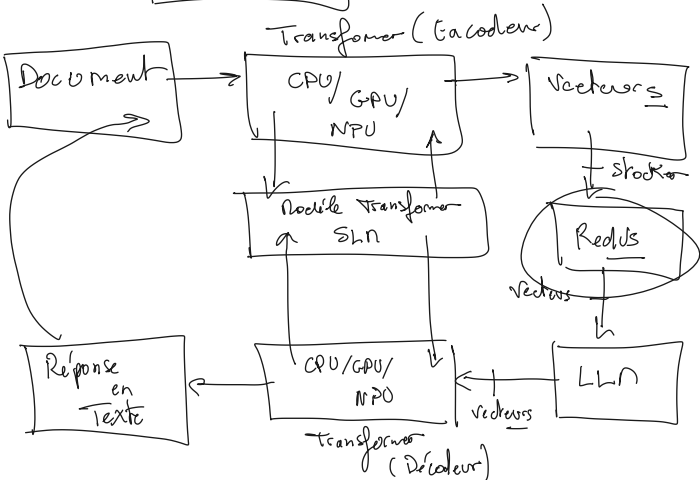
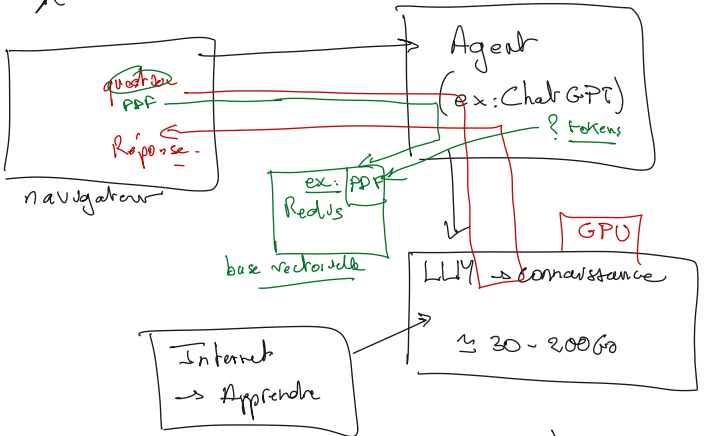


Tokenizers = associer un vecteur (= une collection de nombres)
à un mot

1 token = 1 vecteur = n nombres

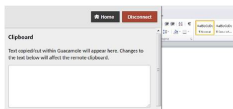
"Bonjour tout le monde"





Copier/Coller avec la VM

- Vous pouvez simplement copier depuis la VM, et coller sur votre machine.
- Il est plus compliqué de copier depuis votre machine, et coller dans la VM.
 - ça doit se faire en 4 étapes
 - a. Copier depuis votre machine
 - b. Appui sur "Ctrl"- "Alt"- "Shift" sous guacamole : une fenêtre s'ouvre
 - c. Coller dans cette fenêtre, puis "Ctrl"- "Alt"- "Shift" sous guacamole : la fenêtre se ferme
 - d. Collez dans votre machine



Lab : Installation de Redis :

1. Par les sources :
2. Procédure sur : https://redis.io/docs/latest/operate/oss_and_stack/install/build-stack/ubuntu-noble/
3. passez root :
 - a. `sudo su -`
4. Aller dans les sources :
 - a. `cd /usr/src`
5. Téléchargez les sources :
 - a. `wget -O redis-7.2.11.tar.gz https://github.com/redis/redis/archive/refs/tags/7.2.11.tar.gz`
6. Installer les prérequis :
 - a. Lancez la commande d'installation des prérequis
 - i. `apt update & apt-get install -y --no-install-recommends \`
`ca-certificates \`
`wget \`
`dpkg-dev \`
`gcc \`
`g++ \ ....`
7. décompressez avec
 - a. `tar xf (fichier gz de redis)`
8. Allez dans le dossier
9. Préconfigurez :
 - a. `export BUILD_TLS=yes`
`export BUILD_WITH_MODULES=yes`
`export INSTALL_RUST_TOOLCHAIN=yes`
`export DISABLE_WERRORS=yes`
10. Compilez :
 - a. Compilez en spécifiant sa destination : (par exemple, pour st01) :
 - i. `make -j "$(nproc)" all`
11. Pour chaque utilisateur :
 - a. Copiez dans son home :
 - i. `cp src/redis-server /home/st01/redis/7.2.11`
 - ii. `cp src/redis-cli /home/st01/redis/7.2.11`
 - iii. `cp redis.conf /home/st01/redis/7.2.11`
 - iv. `sudo chown -R st01:users /home/st01/redis/7.2.11`

Configuration de Redis en Démon :

- Activer daemonise à yes
- Spécifier le fichier PID dans un endroit écrivable
- Spécifier une destination du fichier log
- lancer le server
- se connecter avec le client
- Arrêt possible :
 - kill <process> : Je demande à Redis de s'arrêter proprement
 - kill -9 <process> : Je demande à l'OS de tuer Redis. Acceptable car par d'écriture de fichiers.
 - Recommandé : depuis le client : commande SHUTDOWN

Exercice :

Activez le mode daemonized comme il faut

Arrêtez le serveur :

1. via kill
2. via le client

Reparamétrez en mode interactif

Relancez redis, vérifiez.

Installation de RedisInsight par les sources :

<https://github.com/redis/RedisInsight/releases>

Prérequis :

sudo apt install git curl npm

Installer node.js : <https://nodejs.org/en/download>

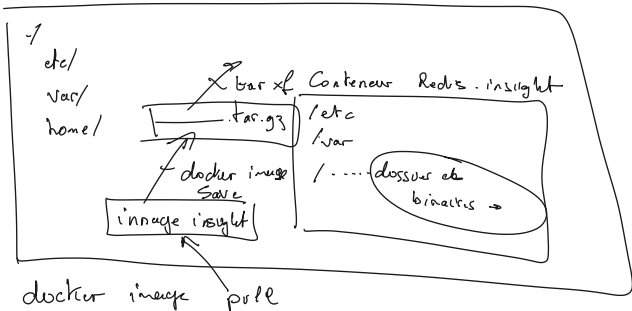
Installer yarn : npm i yarn

Télécharger les sources :

git clone <https://github.com/redis/RedisInsight.git>

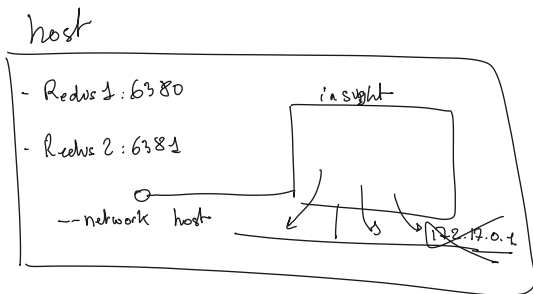
Récupérer les binaires de l'image Docker :

Machine



Procédure d'extraction des binaires d'une image docker :

1. Télécharger l'image
 - a. docker image pull redis/redisinsight
2. La sauvegarder au format .tar dans le home :
 - a. sudo docker image save -o ~/redisinsight.tar redis/redisinsight
3. l'extraire :
 - a. tar xf ~/redisinsight.tar



Lancer en conteneur sans redirection de port et ayant accès à tout localhost :

```
docker run -d --name redisinsight --restart always --network host redis/redisinsight:latest
```

Puis se connecter sur la console :

<http://localhost:5540/>

Utiliser l'équivalent des Transactions :

```
127.0.0.1:6381> multi
OK
127.0.0.1:6381(TX)> set test ok2
QUEUED
127.0.0.1:6381(TX)> discard
OK
127.0.0.1:6381> get test
(nil)
127.0.0.1:6381> multi
OK
127.0.0.1:6381(TX)> set test ok3
QUEUED
127.0.0.1:6381(TX)> exec
1) OK
127.0.0.1:6381> get test
"ok3"
```

Utilisation des sous dossiers/namespaces :
Créer la clef test dans racine/dossier/sousdossier :
set racine:dossier:sousdossier:test "ok dans le sous dossier"

rechercher les clefs d'un sous dossier / namespace :

Exercice :
Travailler dans des namespaces différents pour ces exercices :
Il existe un système de compteur.
Vous créez un compteur qui vaut 0
Vous l'incrémenter (mnémonique d'incrément)
Vous contrôlez.

Créez une liste
Ajouter 3 pays
Consultez la liste
Recherchez dans la liste pour savoir si un pays en fait partie.

Ces deux clefs sont dans deux namespaces différents.

Solutions :

Le compteur :

```
127.0.0.1:6381> SET compteur 0
OK
127.0.0.1:6381> INCR compteur
(integer) 1
127.0.0.1:6381> INCR compteur
(integer) 2
127.0.0.1:6381> GET compteur
"2"
```

La liste des pays :

127.0.0.1:6381> LPUSH pays "France"

(integer) 1

127.0.0.1:6381> LPUSH pays "Senegal"

(integer) 2

127.0.0.1:6381> LPUSH pays "Italie"

(integer) 3

127.0.0.1:6381> GET pays

(error) WRONGTYPE Operation against a key holding the wrong kind of value

127.0.0.1:6381> LRANGE pays 0 -1

1) "Italie"

2) "Senegal"

3) "France"

127.0.0.1:6381> # Il n'existe pas de fonction pour rechercher par valeur avec une liste. Il faut récupérer le contenu avec un range 0 -1 et chercher cote client.

Invalid argument(s)

127.0.0.1:6381> # Avec un SET par contre :

(error) ERR unknown command '#', with args beginning with: 'Avec' 'un' 'SET' 'par' 'contre' ':'

127.0.0.1:6381> DEL pays

(integer) 1

127.0.0.1:6381> SADD pays "France"

(integer) 1

127.0.0.1:6381> SADD pays "Senegal" "Italie"

(integer) 2

127.0.0.1:6381> SMEMBERS pays

1) "France"

2) "Senegal"

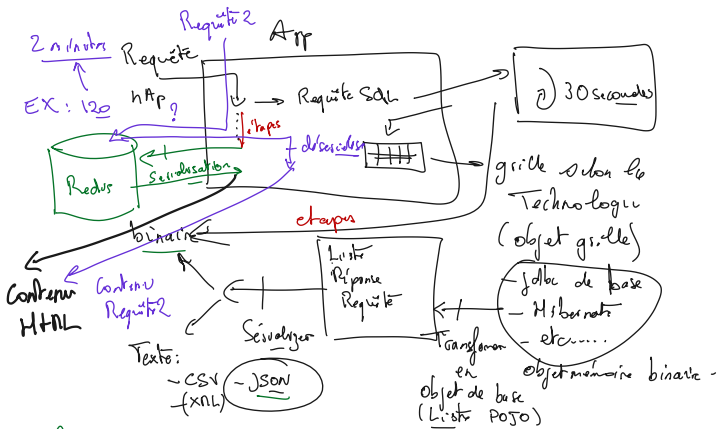
3) "Italie"

127.0.0.1:6381> SISMEMBER pays "Congo"

(integer) 0

127.0.0.1:6381> SISMEMBER pays "France"

(integer) 1



def:

ID de Session : dataset =

flux
m'importe

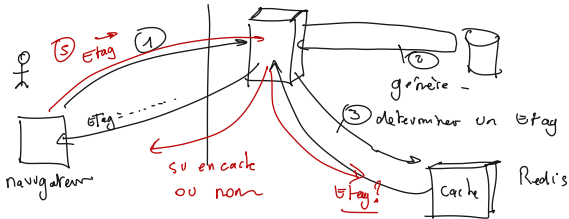
Mise en cache de Page:

- Par page (pour tous les clients) ex: Page mémo-
- Par client/page

ex.: La liste des commandes d'un client.

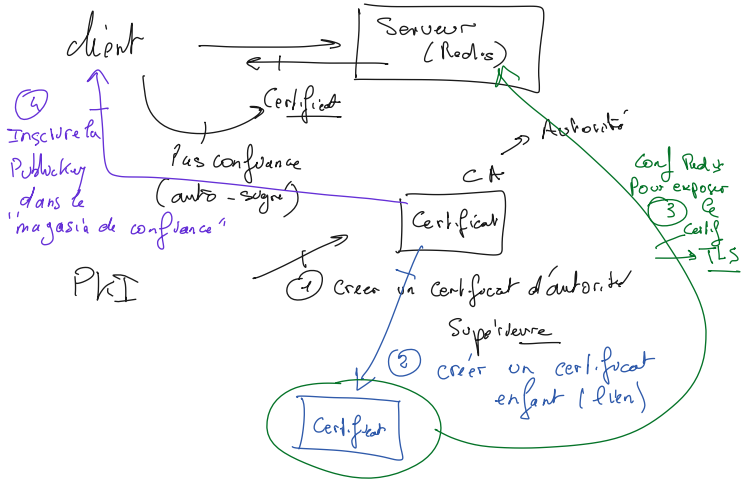
↳ "Ident" "Page"

E-Tag	URL	Contenu
1	Comande	Comde de X
2	Comde	" de Y

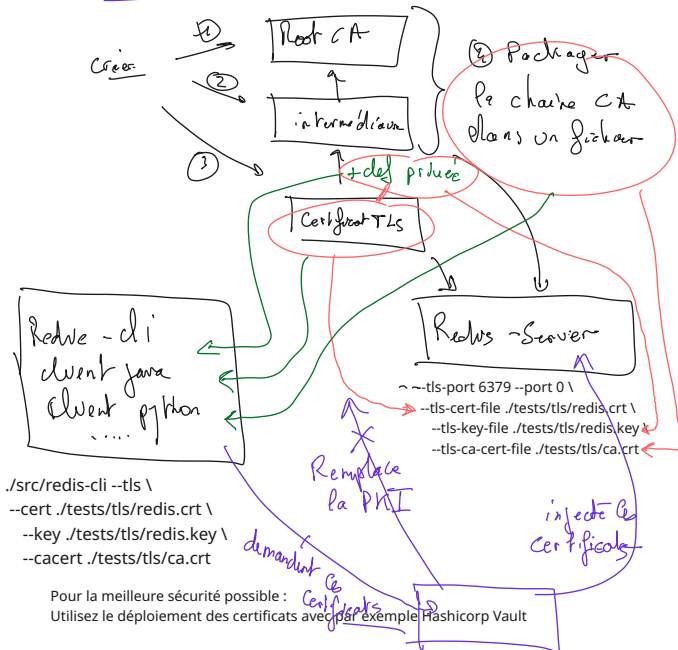


Par ~~server~~ app
ajoute un niveau de namespace = id du session
métadonnées → Etag
Données → Body
Data-headers

PKI



TLS Redis:



```
student@u24:~/redis/7.2.11/bin$ ./redis-cli -p 6379
```

```
127.0.0.1:6379> ping
```

```
Error: Connection reset by peer
not connected> exit
```

```
student@u24:~/redis/7.2.11/bin$ ./redis-cli --tls \
```

```
--cert ../redis.crt \
```

```
--key ../redis.key \
```

```
--cacert ../ca.crt
```

```
127.0.0.1:6379> ping
```

```
PONG
```

```
127.0.0.1:6379>
```

→ TLS act.f, crypté sur la ligne

Pour prendre en charge le TLS, vous devez compiler avec cette prise en charge :

- make clean destclean
 - export BUILD_TLS=yes
export BUILD_WITH_MODULES=yes
export INSTALL_RUST_TOOLCHAIN=yes
export DISABLE_WERRORS=yes
- make -j "\$(nproc)" all BUILD_TLS=yes

Création des certificats de test (alternative à openssl normal en prod) :

To run Redis test suite with TLS, you'll need TLS support for TCL (i.e. tcl-tls package on Debian/Ubuntu).

1. Run ./utils/gen-test-certs.sh to generate a root CA and a server certificate.
 - a. Les certificats sont dans /usr/src/redis/tests/tls/*
2. Run ./runtest --tls or ./runtest-cluster --tls to run Redis and Redis Cluster tests in TLS mode.

ACLs

- Créer un autre super admin :
 - 127.0.0.1:6379> aCL whoami
"default"
 - ACL SETUSER admin2 on >MonPassSuperFort123 allcommands allkeys
- Désactiver le user default
 - ACL SETUSER default off
- S'authentifier :
 - AUTH admin2 MonPassSuperFort123

Direct en ligne de commande :

```
./redis-cli --tls --cert ../redis.crt --key ../redis.key --cacert ../ca.crt --user admin2 --pass  
MonPassSuperFort123
```

Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.

```
127.0.0.1:6379> ping
```

```
PONG
```

```
127.0.0.1:6379> set foo bar3
```

```
OK
```

En cas de perte du mot de passe du dernier super admin

Redémarrez le serveur en mode non protégé :

Arrêt du serveur,

Fichier de conf :

protected-mode ~~yes~~ -> no

Redémarrer le serveur.

login en mode default sans mot de passe :

```
student@u24:~/redis/7.2.11/bin$ ./redis-cli -h 127.0.0.1 --tls --cert ../redis.crt --key ../redis.key --cacert ../ca.crt
```

```
127.0.0.1:6379> ping
```

```
PONG
```

```
127.0.0.1:6379> acl list
```

```
1) "user default on nopass sanitize-payload ~* &* +@all"
```

```
127.0.0.1:6379>
```

Rejouer toutes les commandes d'ACL pour tous les utilisateurs :

```
student@u24:~/redis/7.2.11/bin$ ./redis-cli -h 127.0.0.1 --tls --cert ../redis.crt --key ../redis.key --cacert ../ca.crt
```

```
127.0.0.1:6379> acl list
```

```
1) "user default on nopass sanitize-payload ~* &* +@all"
```

```
127.0.0.1:6379> ACL SETUSER admin2 on >MonPassSuperFort allcommands allkeys
```

```
OK
```

```
127.0.0.1:6379> exit
```

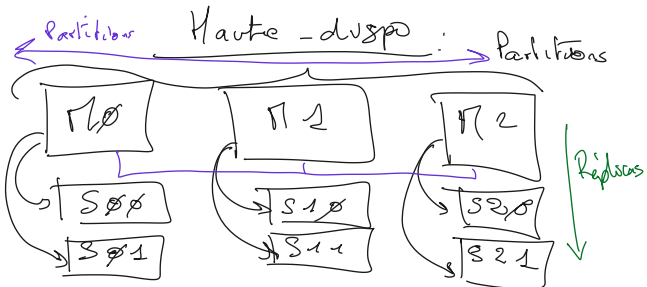
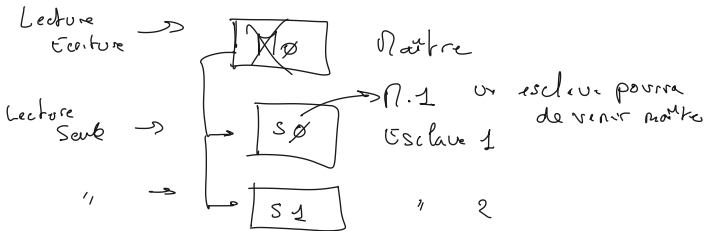
Topologies de HA :

x cluster (multi-maitre)



chaque def n'existe qu'une fois
(partitionnement)

x Haute-dispo (Réplication)



Il faut toujours au moins 3 master, et autant de nodes en réplica derrière :

```
student@u24:~/redis/7.2.11/bin$ ./redis-cli --cluster create 127.0.0.1:6379 127.0.0.1:6380 127.0.0.1:6381 --cluster-replicas 0
>>> Performing hash slots allocation on 3 nodes...
Master[0] -> Slots 0 - 5460
Master[1] -> Slots 5461 - 10922
Master[2] -> Slots 10923 - 16383
M: dd6b1bf2a904072624f7eddd77cdbc9ddfa31a9c 127.0.0.1:6379
  slots:[0-5460] (5461 slots) master
M: 289f2fd30e0181e7f4d23b26f5915bd296346a81 127.0.0.1:6380
  slots:[5461-10922] (5462 slots) master
M: ece331c5ae0820e4133493194557b4404436bb3f 127.0.0.1:6381
  slots:[10923-16383] (5461 slots) master
Can I set the above configuration? (type 'yes' to accept): ^C
student@u24:~/redis/7.2.11/bin$ ./redis-cli --cluster create 127.0.0.1:6379 127.0.0.1:6380 127.0.0.1:6381 --cluster-replicas 1
*** ERROR: Invalid configuration for cluster creation.
*** Redis Cluster requires at least 3 master nodes.
*** This is not possible with 3 nodes and 1 replicas per node.
*** At least 6 nodes are required.
student@u24:~/redis/7.2.11/bin$ ./redis-cli --cluster create 127.0.0.1:6379 127.0.0.1:6380 127.0.0.1:6381 --cluster-replicas 2
*** ERROR: Invalid configuration for cluster creation.
*** Redis Cluster requires at least 3 master nodes.
*** This is not possible with 3 nodes and 2 replicas per node.
*** At least 9 nodes are required.
student@u24:~/redis/7.2.11/bin$ ./redis-cli --cluster create 127.0.0.1:6379 127.0.0.1:6380 --cluster-replicas 1
*** ERROR: Invalid configuration for cluster creation.
*** Redis Cluster requires at least 3 master nodes.
*** This is not possible with 2 nodes and 1 replicas per node.
*** At least 6 nodes are required.
```

Re'plication

