# Installation de la dépendance Erlang

Ajout de la dépendance Erlang utilisé pour installer couchDB :

echo "deb http://packages.erlang-solutions.com/debian jessie contrib" > /etc/apt/sources.list.d/erlang-solutions.list

Installation de Erlang :

wget -qO - http://packages.erlang-solutions.com/debian/erlang\_solutions.asc | apt-key add -

Installation des dépendances :

aptitude update

aptitude install build-essential curl libmozjs185-1.0 libmozjs185-dev libcurl4-openssl-dev libicu-dev wget curl

Installation de la bonne version d’Erlang (CouchDB étant incompatible avec les nouvelles versions d’Erlang) :

aptitude install erlang-dev=1:17.5.3 erlang-base=1:17.5.3 erlang-crypto=1:17.5.3 \

erlang-nox=1:17.5.3 erlang-inviso=1:17.5.3 erlang-runtime-tools=1:17.5.3 \

erlang-inets=1:17.5.3 erlang-edoc=1:17.5.3 erlang-syntax-tools=1:17.5.3 \

erlang-xmerl=1:17.5.3 erlang-corba=1:17.5.3 erlang-mnesia=1:17.5.3 \

erlang-os-mon=1:17.5.3 erlang-snmp=1:17.5.3 erlang-ssl=1:17.5.3 \

erlang-public-key=1:17.5.3 erlang-asn1=1:17.5.3 erlang-ssh=1:17.5.3 \

erlang-erl-docgen=1:17.5.3 erlang-percept=1:17.5.3 erlang-diameter=1:17.5.3 \

erlang-webtool=1:17.5.3 erlang-eldap=1:17.5.3 erlang-tools=1:17.5.3 \

erlang-eunit=1:17.5.3 erlang-ic=1:17.5.3 erlang-odbc=1:17.5.3 \

erlang-parsetools=1:17.5.3

Mettre les package dans hold pour empêcher les upgrade (Et des problèmes d’incompatibilité) :

aptitude hold erlang-dev erlang-base erlang-crypto erlang-nox erlang-inviso erlang-runtime-tools \

erlang-inets erlang-edoc erlang-syntax-tools erlang-xmerl erlang-corba \

erlang-mnesia erlang-os-mon erlang-snmp erlang-ssl erlang-public-key \

erlang-asn1 erlang-ssh erlang-erl-docgen erlang-percept erlang-diameter \

erlang-webtool erlang-eldap erlang-tools erlang-eunit erlang-ic erlang-odbc \

erlang-parsetools

# Set up de l’environnement CouchDB

useradd -d /var/lib/couchdb couchdb

mkdir -p /usr/local/{lib,etc}/couchdb /usr/local/var/{lib,log,run}/couchdb /var/lib/couchdb

chown -R couchdb:couchdb /usr/local/{lib,etc}/couchdb /usr/local/var/{lib,log,run}/couchdb

chmod -R g+rw /usr/local/{lib,etc}/couchdb /usr/local/var/{lib,log,run}/couchdb

# Téléchargement, compilation et installation de CouchDB

wget http://apache.panu.it/couchdb/source/1.6.1/apache-couchdb-1.6.1.tar.gz

tar xzf apache-couchdb-1.6.1.tar.gz

cd apache-couchdb-1.6.1

./configure --prefix=/usr/local --with-js-lib=/usr/lib --with-js-include=/usr/include/js --enable-init

make && make install

# Configuration des fichiers CouchDB et lancement du CouchDB

chown couchdb:couchdb /usr/local/etc/couchdb/local.ini

ln -s /usr/local/etc/init.d/couchdb /etc/init.d/couchdb

ln -s /usr/local/etc/couchdb /etc

update-rc.d couchdb defaults

nano /usr/local/etc/couchdb/local.ini

« bind address 192.168.25.80

port 5984 » -> Premier serveur

« bind address 192.168.25.90

port 5984 » -> Deuxième serveur

/etc/init.d/couchdb start

# Création de la base et création de la réplication

Sur l’interface futon (« 192.168.25.X » les deux serveurs) :

Créer une nouvelle database « hermes »

Dans \_replicator serveur 192.168.25.80 :

* Créer un nouveau fichier

|  |  |
| --- | --- |
| Field | Valeur |
| source | 192.168.25.90:5984/hermes |
| target | 192.168.25.80:5984/hermes |
| continuous | true |

Dans \_replicator serveur 192.168.25.90 :

* Créer un nouveau fichier

|  |  |
| --- | --- |
| Field | Valeur |
| source | 192.168.25.80:5984/hermes |
| target | 192.168.25.90:5984/hermes |
| continuous | true |

# Création du nginx et du load balancing (serveur 192.168.25.80)

Installation de nginx :

apt-get install nginx

nano /home/olaf/nginx.conf

Mettre la configuration minimale dans nginx :

user www-wata;

worker\_processes 4;

pid /run/nginx.pid;

events {

worker\_connections 768;

}

http {

upstream cluster\_couchdb {

server 192.168.25.80:5984;

server 192.168.25.90:5984;

}

server {

listen 8080;

location / {

proxy\_pass http://cluster\_couchdb;

}

}

}

Lancement du nginx:

nginx –c /home/olaf/nginx.conf