CentOS

- Compiler Kea 1.5 via les sources sur CentOS 7.5

Compiler Kea 1.5 via les sources sur CentOS 7.5

Les prérequis

- Mettre à jour le serveur pour avoir les dernières versions des différents packages installés sur le serveur et avoir les derniers patchs de sécurité
  `# yum update -y`
- Premièrement installer le dépôt EPEL pour avoir accès aux dépendances spécifiques des packages
  `#yum install epel-release -y`
- Installer toutes les dépendances requises. Vous devez séparer l'installation des dépendances, car une des dépendances installe un dépôt mandataire pour gcc-g++
  `# yum -y install epel-release cmake bison flex pcre-devel libev-devel protobuf-c-devel protobuf-c-compiler make rpm-build doxygen swig autoconf automake libtool gtest-devel openssl-devel git wget mariadb-devel mariadb-server`
  `# yum -y install log4cplus-devel ccache boost-devel centos-release-scl`
  `# yum -y install devtoolset-7-gcc*`
  NOTE IMPORTANTE: Exécuter les commandes dans cet ordre ! Quelques packages installent un mini dépôt pour récuperer des dépendances C++. Elles ne peuvent pas être concaténer en une seule commande
- La commande suivante est requise pour compiler la librairie Boost (une des principales dépendences) et Kea DHCP Server. Il lance une console bash avec toutes les variable d'environement de départ pour utiliser la version correcte de gcc-g++
  `# scl enable devtoolset-7 bash`
  NOTE IMPORTANTE: Si vous avez besoin de reconnecter ou de reboot le serveur, n'oubliez pas de retaper cette commande pour éviter les problèmes durant la compilation ou l'installation

Installation de Kea

- Maintenant, télécharger la dernière version (1.5) de Kea (1.4-P1 ou moins ne fonctionnera pas a cause des dépendances spécifique de la librairie Boost)
  `# wget -nd https://ftp.isc.org/isc/kea/1.5.0/kea-1.5.0.tar.gz`
  `# tar zxvf kea-1.5.0.tar.gz`
  `# cd kea-1.5.0`

- Lancer le script de configuration avec le flag "--with-mysql" (vous pouvez en rajouter si vous avez besoin d'autres fonctionnalités)
  `# ./configure --with-mysql`
- Tout devrait être bon, Kea peut être compilé en utilisant make. Regarder si tout est bon après la compilation avec la commande make check
  `# make`
  `# make check`
  NOTE IMPORTANTE: Si vous voyez une erreur d'initialisation MySQL, c'est normal, le script de test essaye de se connecter a une base de donnée qui n'existe pas sur le serveur (Car la base de données a besoin d'etre configuré avant l'installation de Kea pour tester l'interaction entre Kea et la base, mais ce n'a pas d'impact si la base est configuré après l'installation)
- Enfin, installer le logiciel
  `# make install`
- Vous pouvez regarder si le logiciel fonctionne bien avec la commande keactrl
  LRYyg1CtiKHfAsaI-image-1558614256671.png
- Essayer de lancer le serveur DHCP en utilisant la commande keactrl start et revérifier le statu

WZfTlvYaDVJRUOhM-image-155861429

Configuration de la base de donnée

Cette section va concerner la configuration de Kea

- Activer et démarrer le service MySQL/MariaDB
  `# systemctl enable —-now mariadb`
  NOTE IMPORTANTE: La commande ci-dessus est une astuce pour lancer et activer un service systemd en même temps
- Configuration de MySQL/MariaDB
  `# mysql_secure_installation`
  Suivez les indications et configurez les paramètres à votre grès
- Configuration de la base
  - Connectez vous a la base en utilisant le client MySQL
    `$ mysql -u root —p`
  - Créez la base que Kea va utilisez, utilisez le nom de votre choix
    `mysql> CREATE DATABASE keadb;`
  - Créer un user dédié et donnez lui les bon accès
    `mysql> CREATE USER 'keauser'@'localhost' IDENTIFIED BY 'passkea';`
    `mysql> GRANT ALL ON keadb.* TO 'keauser'@'localhost';`
  - Quittez MySQL et tapez la commande suivante pour que Kea crée le schéma de la base de données
    `$ kea-admin lease-init mysql -u keauser -p passkea -n keadb`

- Configurez l’intégration MySQL sur le serveur DHCP Kea
  La configuration de l’intégration de la base de donnée sur Kea est faite en utilisant ce genre de configuration. La configuration suivante autoriser le stockage du bail dans la base de donnée. Vous pouvez stocker les hôtes aussi dans la base de donnée (voir l'image suivante).
  HonGrJJ5KNxEgf2P-image-1558615186649.png
- -> DB Backend : MySQL, Postgre ou Cassandra
  -> Nom de la base de donnée
  -> User pour accéder a la base
  -> Mot de passe de l'user
  -> L'hote ou est stocké la base
  -> Le port pour atteindre la base
- Configurez les classes Kea
  La définition des classe DHCP est défini par un tableau JSON appelé "client-classes". Chaque propriété de ce tableau définit une classe
  MbaChiR2vWkcqPXs-image-1558615453879.png
  Pour chaque objet dans le tableau, il est possible de donner un nom et un test (équivalent de "if match" sur ISC DHCP) a la classe
- Configurez les plages de sous réseau
  La définition du sous réseau DHCP est défini par un tableau JSON appelé "subnet". Chaque propriété de ce tableau spécifie toutes les données qui vont être envoyé au client, excepté la propriété "client-class" qui va définir pour quel classe Kea Doit appliquer une IP de cette plage
  QG42EsSgeutYyUe5-image-1558615743847.png

Configuration de la haute disponibilité (Failover)

Cette partie va couvrir la configuration de la fonctionnalité HA Failover entre deux Kea DHCP serveurs. Pour configurer le HA, vous avez besoin de deux hooks spécifique pour étendre la capacité de Kea, mais ces hooks sont installé avec Kea
6FFg7v1YeuFkMMhV-image-1558679836700.png

- Pour utiliser cette fonctionnalité, vérifiez que les deux fichier suivant sont présent sur l'installation de ISC Kea
  - libdhcp_ha.so
  - libdhcp_lease_cmds.so
    Le chemin par défaut de ces deux fichier est /usr/local/lib/hooks/
    Si vous avez spécifié un autre chemin durant l'installation de Kea (./configure step), vérifiez en fonction des parametres que vous avez défini. Si les fichiers ne sont pas présent, vous devez recompiler le logiciel en entier.
- L'étape suivante est de configurer le fichier de configuration des deux serveurs. La configuration de ces deux serveur est presque la même, juste quelques détails qui changent. Vous devez créer la section suivante avant la section "subnet"
  - Le premier hook/librairie à charger:
    KCIrmtW2TWIc6a9D-image-1558680241857.png

- La deuxième librairie à charger et à configurer
  VubtcfYbOb1bfcqN-image-1558680273499.png
  - this-server-name : Le nom du serveur actuellement configuré
    mode : the High-Availability mode (load-balancing or hot-standby).
  - heartbeat-delay : Le délai en millisecondes entre les messages heartbeat (Pour vérifier que l'autre serveur est encore en vie).
  - max-response-delay : Le délai en millisecondes depuis la dernière communication avec l'autre serveur, après lequel le serveur considère que la communication est interrompu
  - max-ack-delay : Ce paramètre spécifie le temps maximum en millisecondes pour le client d'essayer de communiquer avec le serveur DHCP, après lequel le serveur considère que le client n'a pas réussi a communiquer avec le serveur DHCP (il est "unacked").
  - max-unacked-clients : Ce paramètre défini le nombre maximum de client qui ne peuvent pas être servi avant que ce serveur considère que l'autre est hors-ligne et devient le serveur qui sert le client
    Un exemple en pratique est avec la valeur défini a 5. Si le premier serveur est hors-ligne, le serveur en stand-by va attendre que 5 client demande un bail sans avoir de réponse du premier server avant de changer son état et commencer a servir des clients
    B12uTSYNzGAS8YYx-image-1558680755514.png
  - name : Le nom du serveur qui fait partie du peering
  - url : L’adresse HTTP du serveur. La communication entre les serveurs est faites avec une API RESTful qui utilise le protocole HTTP
  - role : Le role de ce serveur (primaire, stand-by ou backup)
  - auto-failover : Si le serveur détecte une défaillance, il devrait automatiquement démarrer a servir des clients
- La prochaine étape est de configurer l'agent de contrôle Kea. Cet agent de contrôle expose L’API RESTful sur le port 8080 par défaut. Il y a juste besoin de configurer l'interface d'écoute, car par défaut l'API écoute seulement sur l'interface loopback
  Le chemin par défaut pour configurer l'agent de contrôle est: /usr/local/etc/kea/kea-ctrl-agent.conf
  Dans cet exemple, j'ai configuré l'agent de controle pour exposer l'API sur tout les réseau sur le port 8080
  kiSNDXYwPLR9gLGX-image-1558681135874.png
- Avant de lancer le serveur DHCP, il est important d'ouvrir un port dans le firewall
  `# firewalld-cmd --permanent –add-port=8080/tcp`
  `# firewalld-cmd --permanent –add-port=3306/tcp`
  `# firewalld-cmd --reload`
  J'ai choisi d'ouvrir le port de l'API (8080) et le port MySQL (3306). L'API a besoin d'étre ouverte car par défaut le port est bloqué par le Firewall.
- On peux vérifier que le système HA est fonctionnel
  - Premier serveur:
    2HWbTofHJ5tbHVIZ-image-1558681260986.png
  - Le serveur en Stand-by:
    9oe1LKxpjme3FEw7-image-1558681279398.png

- Si le premier serveur tombe, la transition est faite (le deuxième serveur réponds au DHCP Discover):
  6qUZJVfC0yiUQB1c-image-1558681323190.png