SAE-D011 CLOUD

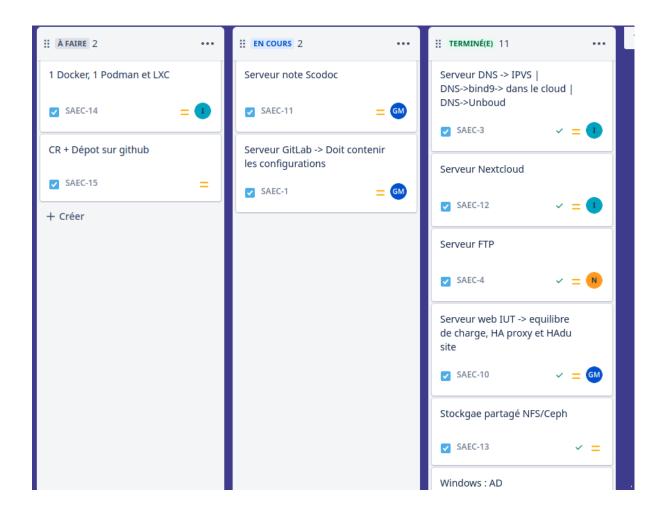
JM.Pouchoulon

Contexte:

On est une équipe de 4 personnes (diplômé bac+2), qui viennent d'obtenir le poste d'assistant ingénieur.La méthodologie de travail est en mode agile. De nombreux services sont à mettre en place chacun s'attribue une tâche et passe à une autre tâche une fois fini.

Partage de tâches:

Pour se partager les différentes tâches on à créé un tableau Jira en listant toute les différentes tâches.



I-Installation des services DNS en haute disponibilité

Service DNS local bind9:

Dans un premier temps, j'ai dû créer une vm ubuntu légère en tant que vm hôte pour containeriser le service, par la suite j'ai repris mes notes de l'année précédente et de début d'années pour revoir le format des fichiers configurations bind9. Après des premiers tests sans containerisation de bind9 j'ai récupérer les fichiers de configurations et transférées sur la vm hôte via un "git clone" (on peut avec scp aussi mais j'en ai profité pour sauvegarder mes fichiers) , j'ai créé un containers docker (j'ai build l'image d'abord avec "docker build") avec un volume avec l'options -v de la commande "docker run" ou j'ai indiquer le volume dossier contenant mes fichiers de configuration puis le dossiers de destinations

Mon Dockerfile:

J'installe les paquets (apt-utils et bind9) avec l'options -y pour qu'il accept les conditions d'installation automatique

J'ai rajouté des droits sur les trois fichiers car par défaut le container docker n'avait pas les droit d'écriture ce qui avait pour effet de faire planter au démarrage le container On lui ouvre le port 53 car c'est celui utilisé par le DNS L'options named -g permet de faire tourner le processus bind9 dès le démarrage car un container sans processus "meurt"

Puis on lance le build de l'image avec la commande : "docker build -t dnsi -f Dockerfile ."

```
root@dns:/home/dns# docker image ls
REPOSITORY
             TAG
                        IMAGE ID
                                        CREATED
                                                       SIZE
dnsi
                        8728b623b113
                                                       228MB
             latest
                                        5 days ago
debian
                        1ac99357ef21
                                                       124MB
             latest
                                        3 weeks ago
```

Puis on lance le container avec la commande suivante :

"docker run -d --restart=unless-stopped -v
/home/dns/docker-dns:/etc/bind -p 53:53/udp --name dnsserver dnsi"

Ici l'option restart=unless-stopped est très important car cela permet de redémarrer le container s'il s'arrête même au redémarrage de la vm hôte. On précise ici aussi le port 53:53/udp

```
root@dns:/home/dns# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
416bd80ca7e1 dnsi_ "named -g" 20 hours ago Up 20 hours 0.0.0.0:53→53/udp, :::53→53/udp dnsserver
```

Les fichiers de configuration bind9:

Fichier -> db.sae.jing.fr

```
$TTL
        604800
                 SOA
                         sae.jing.fr. root.sae.jing.fr. (
        IN
                                           ; Serial
                         604800
                                           ; Refresh
                         86400
                                           ; Retry
                         2419200
                                          ; Expire
                                           ; Negative Cache TTL
                         604800 )
                 IN
                         NS
                                  sae.jing.fr.
                 IN
                                  192.168.20.53
                         Α
webgui
                 IN
                                  192.168.20.54
git
ftp
                 IN
                                  192.168.20.55
                                  192.168.20.57
                 IN
                         Α
scodoc
                 IN
                                  192.168.20.59
                 IN
                         Α
                                  192.168.20.60
nc
glpi
                 IN
                                  192.168.20.250
                 IN
                                  192.168.20.251
portainer
                         Α
                 IN
                                  192.168.20.252
```

Le nom de notre domaine est sae.jing.fr (JING , initial de chaque membre du groupe)

Ici on indique l'ip du dns 192.168.20.53 Explications:

webgui représentera l'adresse webgui.sae.jing.fr
IN A permet d'indiquer une ipv4 → puis on entre l'ipv4

TTL:

durée de validité que le server fournit aux resolver Serial:

Le numéro de série est à incrémenter dès qu'une modification est effectuée

Refresh:

Nombre de secondes entre 2 demandes de mise à jour entre le serveur maître et esclave

Retry:

Nombre de secondes que le serveur esclave attend avant de réémettre une demande si la précédente a échoué

Expire:

Nombre de secondes qu'un serveur attend avant de considérer une donnée comme indisponible

Negative Cache TTL: obsolète

Fichier -> named.conf.local

```
zone "sae.jing.fr" IN {
type master;
file "/etc/bind/db.sae.jing.fr";
allow-update { none; };
allow-query{any;};
};
```

On nomme la zone , ici sae.jing.fr et indique le fichier /etc/bind/db.sae.jing.fr et on laisse le reste par défaut. Type correspond à server maître ici, le file correspond au chemin du fichier qui décrit la zone. Ce fichier permet de juste définir quel fichier nous allons utiliser pour décrire notre zone allow-query{any;}; est nécessaire pour éviter les problèmes de query-cache

Fichier -> named.conf.options

```
allow-query {any;};
allow-query {any;};
allow-query-cache {any;};
directory "/var/cache/bind";
forwarders {8.8.8.8;8.8.4.4;};
};
```

directory par défaut

forwarders on mais 8.8.8.8 et 8.8.4.4 cela permet de renvoyer toutes les demande de résolutions DNS inconnu au DNS google

<u>Service DNS cloud</u> unbound:

La démarche est presque pareil plutôt qu'avoir plusieurs fichier de configuration il nous suffit d'en avoir un seul unbound.conf

Fichier -> Dockerbuild

```
# Create the root anchoir file Install Unbound and its dependencies
RUN apt-get update & apt-get upgrade -y & apt-get install -y \
apt-utils\
unbound

EXPOSE 53
ADD root.key /var/lib/unbound/root.key

# Start Unbound when the container launches
CMD ["/usr/sbin/unbound", "-d"]
```

Pareil que tout à leur plutôt que bind9 on remplace par unbound Le port est toujours le même, cette fois si on ajoute un fichier dans /var/lib/unbound/ le fichier doit être créé au préalable avec la commande "unbound-anchor -a root.key" c'est la clef qui permet l'accès root.

Pour lancer unbound au démarrage du container ["/usr/sbin/unboud", "-d"]

Pour lancer crée l'image "docker build -t image_unbound -f Dockerfile ."

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
image_unbound	latest	a1cc7d059de8	2 days ago	182MB
debian	latest	1ac99357ef21	3 weeks ago	124MB

Fichier -> unbound.conf

```
# Unbound configuration file for Debian.
# See the unbound.conf(5) man page.
# See /usr/share/doc/unbound/examples/unbound.conf for a commented
# reference config file.
# The following line includes additional configuration files from the
# /etc/unbound/unbound.conf.d directory.
include-toplevel: "/etc/unbound/unbound.conf.d/*.conf"
server:
  interface: 0.0.0.0
  access-control: 0.0.0.0/0 allow
  do-ip6: no
  #control-enable: no
  #control-interface: ::1
  #control-port: 8955
local-zone: "sae.jing.fr" static
local-data: "webgui.sae.jing.fr IN A 192.168.20.54"
local-data: "git.sae.jing.fr IN A 192.168.20.55"
local-data: "ftp.sae.jing.fr IN A 192.168.20.57"
local-data: "scodoc.sae.jing.fr IN A 192.168.20.59"
local-data: "nc.sae.jing.fr IN A 192.168.20.60"
local-data: "glpi.sae.jing.fr IN A 192.168.20.250"
local-data: "ws.sae.jing.fr IN A 192.168.20.251"
local-data: "portainer.sae.jing.fr IN A 192.1689.20.252"
forward-zone:
  name: "."
  forward-addr: 8.8.8.8
```

server:

interface: 0.0.0.0

access-control: 0.0.0.0/0 allow

La machine écoutent sur tous les port et peut importe notre IP ont peut faire une demande de résolutions

do-ip6: no, désactive l'ipv6

local-zone: permet de définir le nom de domaine static "sae.jing.fr"

local-data: permet de définir les nom de domaine et l'est ip

correspondant

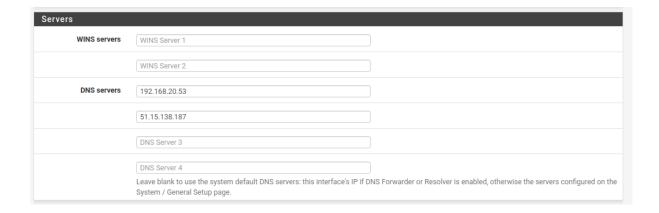
forward-zone: permet de renvoyer toutes les autres demandes inconnue

vers le dns google

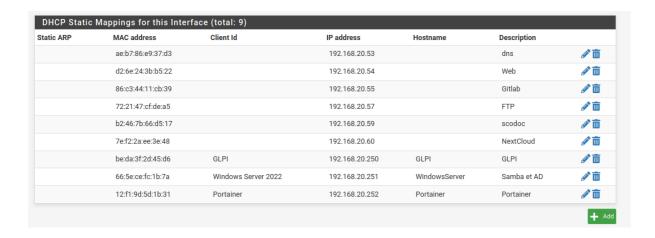
Une fois les configuration dns fini pour lancer le containers : "sudo docker run -d --restart=unless-stopped -v /root/conf:/etc/unbound -p 53:53/udp --name dns_unbound image_unbound"



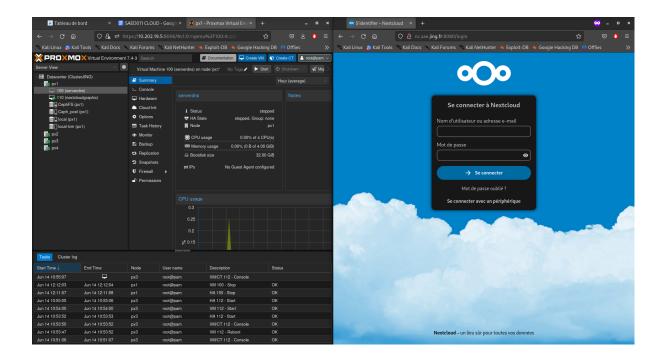
Une fois fini il faut ajouter dans le pfsense les ip des serveurs dns



Et fixer les ip selon la mac adresse pour par que le server dhcp change les ip des machines



On peut vérifier que quand on éteint un DNS ici le local le cloud reprend bien le relais.Ceci marque la haute disponibilité via Pfsense



II- Installation server nextcloud

Pour changer cette fois je vais utiliser un docker-compose, une docker-compose.yml est déjà disponible sur le github de nextcloud l'url est la suivante : "https://github.com/nextcloud"

Pour cette installation il faut savoir qu'il faut utiliser une VM hôte graphique car une requête SQL est faite depuis la page web mais le chemin spécifie le localhost donc depuis une machine graphique externe celle-ci ne marche pas

Donc j'ai installé une ubuntu graphique, fait un copier coller du fichier docker-compose.yml dans un dossier nommée NextCloud.

La première étape est de compléter ce fichier

Fichier → docker-compose.yml

```
version: '2'
   image: mariadb:10.6
   command: --transaction-isolation=READ-COMMITTED --log-bin=binlog --binlog-format=ROW
    - ./db:/var/lib/mysql
     - MYSQL_ROOT_PASSWORD=BJA2FJNfqa
     - MYSQL_PASSWORD=a4eJKJ8NwX
     - MYSQL_DATABASE=nextcloud
     - MYSQL_USER=nextcloud
   image: nextcloud
   restart: always
     - 8080:80
     - ./nextcloud:/var/www/html
     - MYSQL_PASSWORD=a4eJKJ8NwX
     - MYSQL_DATABASE=nextcloud
     - MYSQL_USER=nextcloud
     - MYSQL_HOST=db
     - TRUSTED_DOMAINS="nc.sae.jing.fr"
```

Premièrement dans les volumes rajouter au tout début du chemin un ./ pour que le volume soit persistant à l'emplacement de notre dossier NextCloud et stocker en local sur la VM hôte.

Ensuite compléter le mot de passe root et l'utilisateur normal, puis rajouter :

" - TRUSTED_DOMAINS="nc.sae.jing.fr" "

Cela permettra la connection depuis l'extérieur si on appartient au domains sae.jing.fr depuis le nom nc.sae.jing.fr

commande pour lancer les containers :

docker-compose up -d

commande pour arreter les containers :

docker-compose down -v
(départitionner les disques -v)

d_app_1	nextcl
7722e498dffc mariadb:10.6 "docker-entrypoint.s…" 25 hours ago Up 22 hours 3306/tcp _db_1 -ootûtest:/home/test#	nextcl

On peut voir que j'ai crée plusieurs compte utilisateurs pour tester :

- -Un pour chaque membre du groupe
- -Un pour l'enseignant
- -Un compte administrateur

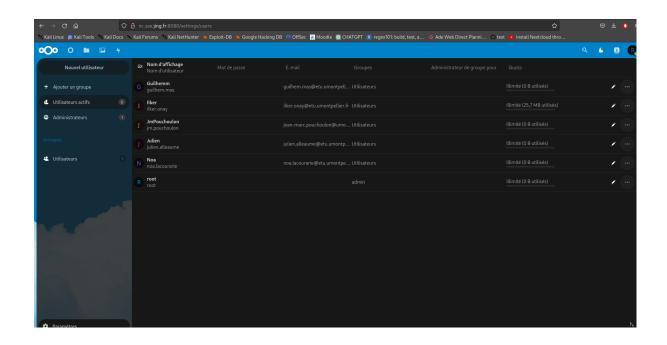


Schéma du réseaux :

