

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 a créé un tableau Jira en listant toutes les différentes tâches.

À FAIRE 2	EN COURS 2	TERMINÉ(E) 11
1 Docker, 1 Podman et LXC <input checked="" type="checkbox"/> SAEC-14 = I	Serveur note Scodoc <input checked="" type="checkbox"/> SAEC-11 = GM	Serveur DNS -> IPVS DNS->bind9-> dans le cloud DNS->Unbound <input checked="" type="checkbox"/> SAEC-3 ✓ = I
CR + Dépôt sur github <input checked="" type="checkbox"/> SAEC-15 =	Serveur GitLab -> Doit contenir les configurations <input checked="" type="checkbox"/> SAEC-1 = GM	Serveur Nextcloud <input checked="" type="checkbox"/> SAEC-12 ✓ = I
+ Créer		Serveur FTP <input checked="" type="checkbox"/> SAEC-4 ✓ = N
		Serveur web IUT -> équilibre de charge, HA proxy et HAdu site <input checked="" type="checkbox"/> SAEC-10 ✓ = GM
		Stockage partagé NFS/Ceph <input checked="" type="checkbox"/> SAEC-13 ✓ =
		Windows : AD

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:

```
FROM debian:latest

# Install necessary packages
RUN apt-get update && apt-get upgrade -y && apt-get install -y \
    apt-utils\
    bind9
RUN chmod o+w /etc/bind /etc/bind/named.conf.options
RUN chmod o+w /etc/bind /etc/bind/named.conf.local
RUN chmod o+w /etc/bind /etc/bind/named.conf

# Expose DNS port
EXPOSE 53/udp

# Start the DNS server
CMD ["named", "-g"]
```

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'option 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	latest	8728b623b113	5 days ago	228MB
debian	latest	1ac99357ef21	3 weeks ago	124MB

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

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

```
options {  
    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

Service DNS cloud 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

```
FROM debian:latest  
  
# Create the root anchor 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

ONAY Ilker

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 192.168.20.252"
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"
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
aca40c66536e	image_unbound	"/usr/sbin/unbound -d"	39 seconds ago	Up 38 seconds	53/tcp, 0.0.0.0:53→53/udp	dns_unbound

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

Servers

WINS servers

WINS Server 1

WINS Server 2

DNS servers

192.168.20.53

51.15.138.187

DNS Server 3

DNS Server 4

Leave blank to use the system default DNS servers: this interface's IP if DNS Forwarder or Resolver is enabled, otherwise the servers configured on the System / General Setup page.

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

DHCP Static Mappings for this Interface (total: 9)					
Static ARP	MAC address	Client Id	IP address	Hostname	Description
	ae:b7:86:e9:37:d3		192.168.20.53		dns
	d2:6e:24:3b:b5:22		192.168.20.54		Web
	86:c3:44:11:cb:39		192.168.20.55		Gitlab
	72:21:47:cf:de:a5		192.168.20.57		FTP
	b2:46:7b:66:d5:17		192.168.20.59		scodoc
	7e:f2:2a:ee:3e:48		192.168.20.60		NextCloud
	be:da:3f:2d:45:d6	GLPI	192.168.20.250	GLPI	GLPI
	66:5e:ce:fc:1b:7a	Windows Server 2022	192.168.20.251	WindowsServer	Samba et AD
	12:f1:9d:5d:1b:31	Portainer	192.168.20.252	Portainer	Portainer

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

The screenshot shows two side-by-side windows. The left window is the Proxmox VE interface, displaying the 'Summary' tab for a virtual machine named 'serverdns'. The status is 'stopped'. The right window shows the Nextcloud login page with the title 'Se connecter à Nextcloud'. It includes fields for 'Nom d'utilisateur ou adresse e-mail' and 'Mot de passe', a 'Se connecter' button, and links for 'Mot de passe oublié ?' and 'Se connecter avec un périphérique'.

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'

volumes:
  nextcloud:
  db:

services:
  db:
    image: mariadb:10.6
    restart: always
    command: --transaction-isolation=READ-COMMITTED --log-bin=binlog --binlog-format=ROW
    volumes:
      - ./db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=BJA2FJNfqa
      - MYSQL_PASSWORD=a4eJKJ8NwX
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud

  app:
    image: nextcloud
    restart: always
    ports:
      - 8080:80
    links:
      - db
    volumes:
      - ./nextcloud:/var/www/html
    environment:
      - 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)

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
54530d75bc42	nextcloud	"/entrypoint.sh apac..."	25 hours ago	Up 22 hours	0.0.0.0:8080->80/tcp, :::8080->80/tcp	nextcloud
d_app_1						
a722e498dffc	mariadb:10.6	"docker-entrypoint.s..."	25 hours ago	Up 22 hours	3306/tcp	nextcloud
d_db_1						

root@test:/home/test#

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

Nom d'affichage	Mot de passe	E-mail	Groupes	Administrateur de groupe pour	Quota
Guilhem guilhem.mas		guilhem.mas@etu.umontpell...	Utilisateurs		Illimité (0 B utilisés)
Iiker iiker.onay		iiker.onay@etu.umontpellier.fr	Utilisateurs		Illimité (25,7 MB utilisés)
JmPouchoulon jm.pouchoulon		jean-marc.pouchoulon@umo...	Utilisateurs		Illimité (0 B utilisés)
Julien julien.alleaume		julien.alleaume@etu.umontp...	Utilisateurs		Illimité (0 B utilisés)
Noa noa.lacourarie		noa.lacourarie@etu.umontpe...	Utilisateurs		Illimité (0 B utilisés)
root root			admin		Illimité (0 B utilisés)

Schéma du réseaux :

