Documentation sur l'évaluation docker et reverse proxy





Raphaël Bousquet—Cadena ; SIO2

# Sommaire

1. Installation Docker	3
2. Script Conteneur	4
3. Reverse Proxy	6
4. DNS	8

#### 1. Installation Docker

On met à jour et supprime apache2

```
root@dockernginx:~# apt update && apt upgrade -y && apt autoremove --purge ap
ache2
```

On installe des modules dont on aura besoin

```
root@dockernginx:~# apt install -y \
apt-transport-https \
ca-certificates \
curl \
gnupg \
lsb-release
```

On modifie le source.list

```
root@dockernginx:~# echo \
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.
gpg] https://download.docker.com/linux/debian \
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
tee /etc/apt/sources.list.d/docker.list > /dev/null
```

On génère la clé gpg

```
root@dockernginx:~# curl -fsSL https://download.docker.com/linux/debian/gpg
   gpg --dearmor -o /etc/apt/keyrings/docker.gpg
root@dockernginx:~# chmod a+r /etc/apt/keyrings/docker.gpg
```

On installe docker

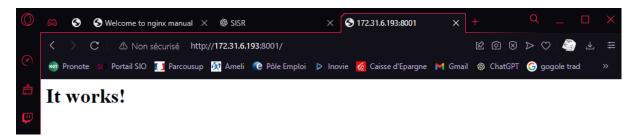
```
root@dockerssh:~# apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y
```

On modifie les droits

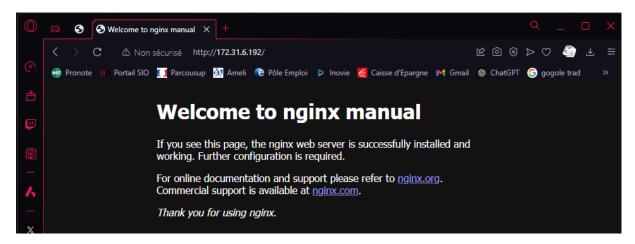
```
root@dockernginx:~# gpasswd -a sio docker
Ajout de l'utilisateur sio au groupe docker
root@dockernginx:~#
```

```
ot@dockernginx:~# systemctl status docker
 docker.service - Docker Application Container Engine
    Loaded: loaded (/lib/systemd/system/docker.service; enabled; preset: en>
    Active: active (running) since Mon 2024-12-09 09:52:36 CET; 21s ago
TriggeredBy: • docker.socket
      Docs: https://docs.docker.com
  Main PID: 27691 (dockerd)
     Tasks: 9
    Memory: 23.0M
    CGroup: /system.slice/docker.service
               27691 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/>
déc. 09 09:52:33 dockernginx systemd[1]: Starting docker.service - Docker Ap
déc. 09 09:52:34 dockernginx dockerd[27691]: time="2024-12-09T09:52:34.0850"
déc. 09 09:52:34 dockernginx dockerd[27691]: time="2024-12-09T09:52:34.77404
déc. 09 09:52:36 dockernginx dockerd[27691]: time="2024-12-09T09:52:36.11058
déc. 09 09:52:36 dockernginx dockerd[27691]: time="2024-12-09T09:52:36.16641
déc. 09 09:52:36 dockernginx dockerd[27691]: time="2024-12-09T09:52:36.16658
déc. 09 09:52:36 dockernginx dockerd[27691]: time="2024-12-09T09:52:36.16664
déc. 09 09:52:36 dockernginx dockerd[27691]: time="2024-12-09T09:52:36.1667
déc. 09 09:52:36 dockernginx dockerd[27691]: time="2024-12-09T09:52:36.2817
    09 09:52:36 dockernginx systemd[1]: Started docker.service - Docker App
lines 1-22/22 (END)
```

On vérifie sur un navigateur le serveur web avec docker :



#### Et voici le serveur web manuel :



# 2. Script Conteneur

echo "Bienvenue dans le gestionnaire de serveurs Docker." echo "Choisissez le type de serveur à déployer :" # Affichage des options disponibles echo "1) Serveur Web (Apache)" echo "2) Serveur DNS (Bind)"	Permet de présenter ce qu'il y a sur le script
echo "3) Serveur Ubuntu de base" echo "4) Serveur SSH (OpenSSH)" read -p "Entrez le	
numéro correspondant à votre choix (1-4) :	
" server_choice	
default_ip=\$(hostname -I   awk '{print \$1}')	Récupérer l'adresse IP principale de la machine
<pre>if [ -z "\$default_ip" ]; then echo "Erreur : Impossible de détecter une adresse IP par</pre>	Vérifier si l'adresse IP a été détectée
défaut. Vérifiez votre configuration réseau."	
exit 1 fi	
echo "Interfaces disponibles :" ip a   grep 'inet '   awk '{print \$2}'   cut -d'/' -f1 sleep 5 # Pause de 5 secondes # Demander une IP et fournir une valeur par défaut read -p "Entrez l'adresse IP pour exposer le serveur (par défaut : \$default_ip) : " machine_ip machine_ip=\${machine_ip:\$default_ip}	Afficher les interfaces disponibles
if! ip a   grep -qw "\$machine_ip"; then echo	Vérifier si l'IP est configurée sur la machine
"Erreur : L'adresse IP \$machine_ip n'est pas	
configurée sur cette machine." echo	
"Veuillez choisir une adresse parmi les	

interfaces disponibles ou utiliser l'adresse	
détectée : \$default_ip" exit 1 fi sleep 5 #	
Pause de 5 secondes	
read -p "Entrez le port à utiliser pour le	Demander le port à utiliser
serveur (par défaut : 8001) : " port	
port=\${port:-8001}	
if ss -tuln   grep -q "\$machine_ip:\$port";	Vérifier si le port est déjà utilisé
then echo "Le port \$port est déjà utilisé sur	
\$machine_ip. Veuillez choisir un autre	
port." read -p "Entrez un nouveau port (par	
défaut : 8080) : " port port=\${port:- 8080} fi	
sleep 5 # Pause de 5 secondes	
start_ssh_server() { echo "Lancement du	Lancer les différents serveurs en fonction
serveur SSH" container_name="servssh"	du choix de l'utilisateur
if docker ps -aformat '{{.Names}}'   grep	Vérifier si un conteneur avec ce nom existe
-qw "\$container_name"; then echo "Un	déjà
conteneur avec le nom \$container_name	
existe déjà. Suppression" docker rm -f	
"\$container_name" fi	
docker run -dname "\$container_name"	Lancer le conteneur Ubuntu et installer
ubuntu bash -c "apt update && apt install	OpenSSH
-y openssh-server && mkdir /var/run/sshd	
&& echo 'PermitRootLogin yes' >>	
/etc/ssh/sshd_config && /usr/sbin/sshd -D"	
if [\$? -eq 0]; then echo "Serveur SSH prêt.	Vérification du succès
Accédez-y via SSH avec : ssh	
root@\$machine_ip -p \$port." else echo	
"Erreur : Impossible de démarrer le serveur	
SSH." exit 1 fi sleep 5 # Pause de 5	
secondes } start_web_server() { echo	
"Lancement du serveur Web Apache"	
container_name="servweb"	
if docker ps -aformat '{{.Names}}'   grep	Vérifier si un conteneur avec ce nom existe
-qw "\$container_name"; then echo "Un	déjà
conteneur avec le nom \$container_name	
existe déjà. Suppression" docker rm -f	
"\$container_name" fi	
docker run -d -p "\$machine_ip:\$port:80"	Lancer le conteneur Apache Web Server
name "\$container_name" httpd	
if [\$?-eq 0]; then echo "Serveur Web	Vérification du succès
Apache prêt. Accédez-y via :	
http://\$machine_ip:\$port" else echo "Erreur	
: Impossible de démarrer le serveur Web."	
exit 1 fi sleep 5 # Pause de 5 secondes }	
start_dns_server() { echo "Lancement du	
serveur DNS Bind"	
container_name="servdns"	
if docker ps -aformat '{{.Names}}'   grep	Vérifier si un conteneur avec ce nom existe
-qw "\$container_name"; then echo "Un	déjà
conteneur avec le nom \$container_name.	
Suppression" docker rm -f	
"\$container_name" fi	
docker run -d -p	Lancer le conteneur Bind DNS Server avec
"\$machine_ip:\$port:53/udp"name	une version spécifique de l'image
"\$container_name"	
internetsystemsconsortium/bind9:9.18	

if [\$? -eq 0]; then echo "Serveur DNS Bind prêt. Accédez-y via : \$machine_ip:\$port (port DNS)" else echo "Erreur : Impossible de démarrer le serveur DNS." exit 1 fi sleep 5 # Pause de 5 secondes }	Vérification du succès
case "\$server_choice" in 1) start_web_server ;; 2) start_dns_server ;; 3) echo "Démarrage d'un conteneur Ubuntu de base" container_name="ubuntu-base" docker run -itname "\$container_name" ubuntu bash sleep 5 # Pause de 5 secondes ;; 4) start_ssh_server ;; *) echo "Option invalide. Veuillez choisir un numéro entre 1 et 4." exit 1 ;; esac echo "Script terminé."	Décider quel serveur démarrer en fonction du choix de l'utilisateur

Grace à ce script, nous avons créé un serveur web et un serveur reverse proxy.

## 3. Reverse Proxy

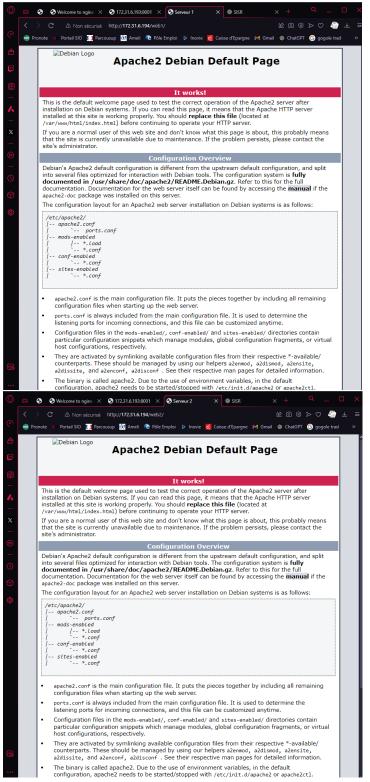
apt install nginx -y

nano /etc/nginx/sites-available/reverse-proxy

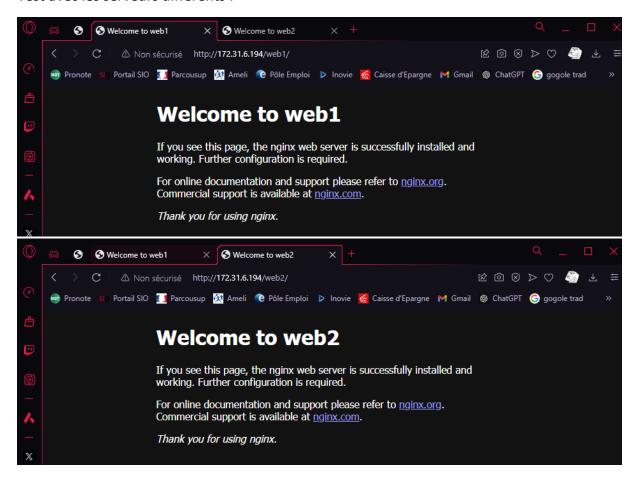
nano /etc/nginx/sites-enabled/reverse-proxy

```
GNU nano 7.2
                                      /etc/nginx/sites-available/reverse-proxy
 Définir le groupe de serveurs backend pour l'équilibrage de charge
upstream backend servers {
   server {
   listen 80;
   server_name tortilla.fr;
   # Redirection vers le premier serveur Apache
   location /webl/ {
      proxy_pass http://172.31.6.101/;
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
       proxy set header X-Forwarded-For $proxy add x forwarded for;
      proxy_set_header X-Forwarded-Proto $scheme;
   # Redirection vers le second serveur Apache
   location /web2/ {
      proxy_pass http://172.31.6.102/;
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
       proxy_set_header X-Forwarded-Proto $scheme;
       proxy_pass http://backend_servers;
      proxy_set_header Host $host;
       proxy set header X-Real-IP $remote addr;
       proxy set header X-Forwarded-For $proxy add x forwarded for;
       proxy_set_header X-Forwarded-Proto $scheme;
```

#### Test avec les deux servers identiques :



Test avec les serveurs différents :



### 4. DNS

#### Apt install bind9

Il faut mettre le domaine dans le hosts du dns et du reverse proxy :

```
GNU nano 7.2 /etc/hosts
127.0.0.1 dnsproxy.tortilla.fr dnsproxy

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 p6-allnodes
ff02::2 ip6-allrouters
```

#### named.conf.local (zones du dns):

```
GNU nano 7.2

// Do any local configuration here
// Consider adding the 1918 zones here, if they are not used in your
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "tortilla.fr" {
    type master;
    file "/etc/bind/db.tortilla.fr";
};

zone "6.31.172.in-addr.arpa" {
    type master;
    file "/etc/bind/db.reverse";
```

db.tortilla.fr (zone principale du dns):

```
GNU nano 7.2
tortilla.fr. IN SOA dnsproxy.tortilla.fr. root.tortilla.fr. (
1 ; Serial
604800 ; Refresh
86400 ; Retry
2419200 ; Expire
604800 ) ; Negative Cache TTL

tortilla.fr. IN NS dnsproxy.tortilla.fr.
dnsproxy.tortilla.fr. IN A 172.31.6.194
tortilla.fr. IN A 172.31.6.194
```

db.reverse (zone inverse du dns) :

```
GNU nano 7.2
6.31.172.in-addr.arpa. IN SOA dnsproxy.tortilla.fr. root.tortilla.fr. (
2000102402
21600
3600
604800
86400
)
6.31.172.in-addr.arpa. IN NS dnsproxy.tortilla.fr.
194_IN PTR dnsproxy.tortilla.fr
```

Pour tester, nous allons prendre une VM windowsd, aller dans le panneau de configuration pour mettre en DNS l'adresse de notre serveur DNS, ensuite aller dans un naviguateur et taper les adresses :

