

Virtualisation

Installer serveur DHCP et DNS

Semestre 2 2024_2025

Création et paramétrage du container

Lancer la console et vérifier que le container que l'on va créer n'existe pas (s'il existe, le supprimer) :

docker ps -a

Créer le réseau que va utiliser le serveur DHCP :

docker network create -d ipvlan --subnet=172.20.0/16 --gateway=172.20.100.1 -o parent=eno1 reseau_DHCP

<u>Créer</u> et <u>lancer</u> le container ubuntu en le nommant serveur_DHCP.

docker run --name=serveur_DHCP --net=reseau_DHCP --hostname=serveur_DHCP -it ubuntu

<u>Réaliser</u> les mises à jour du conteneur.

apt-get update && apt-get upgrade

<u>Entrer</u> le mot de passe administrateur.

passwd

<u>Installer</u> l'éditeur de texte nano. apt-get install nano

<u>Installer</u> le serveur ssh.

apt-get install ssh

Puis activer l'autorisation de se connecter en administrateur en décommentant et en complétant la ligne PermitRootLogin yes dans le fichier /etc/ssh/sshd_config.

Installer le service qui permet d'utiliser les services reseau :

apt-get install iproute2 rsync

Installer le service isc-dhcp-server : apt-get install isc-dhcp-server

Relever le nom de la carte réseau :

lp a

Paramétrage du service DHCP

Editer le fichier de configuration du service dhcp: nano /etc/default/isc-dhcp-server

Ajouter le nom de la carte réseau (indiquer par la commande précédente):

INTERFACESv4="eth0";

Editer le fichier de configuration du service dhcp: nano /etc/dhcp/dhcpd.conf

Entrer les lignes suivantes :

default-lease-time 600; max-lease-time 7200; allow bootp; allow booting;

option subnet-mask 255.255.0.0; option broadcast-address 172.20.255.255; option routers 172.20.100.1; option domain-name-servers 172.20.0.1;

EXPLORATION DU RESEAU

option domain-name "snir.jouvet";

subnet 172.20.0.0 netmask 255.255.0.0

{ range 172.20.100.10 172.20.100.50; }

Redémarrer le service dhcp:

service isc-dhcp-server restart

Demander, à partir des machines clientes, une adresse ip au serveur.

Visualisation des machines clientes

<u>Créer</u> et <u>lancer</u> un nouveau container ubuntu en le nommant page_web_DHCP.

docker run --name=page_web_DHCP --net=reseau_DHCP --hostname=page_web_DHCP -it ubuntu

<u>Réaliser</u> les mises à jour du conteneur.

apt-get update && apt-get upgrade

<u>Installer</u> les paquets nécessaires au fonctionnement. apt install nano rsync cron ssh apache2 php -y

Générer une clé SSH sur le serveur Apache (entrer une clé vide).

ssh-keygen -t rsa -b 4096

<u>Copier</u> la clé publique vers le serveur DHCP.

ssh-copy-id root@adresse_ip_serveur_DHCP

Ajouter la tache cron en lançant la commande suivante :

crontab -e

et *ajouter* dans le fichier la tache de synchronisation suivante (qui envoie la requête toutes les minutes): */1 * * * * rsync -avz root@adresse_ip_serveur_DHCP:/var/lib/dhcp/dhcpd.leases /var/www/html/dhcp.leases

<u>Créer</u> le fichier index.php :

nano /var/www/html/index.php

Copier le contenu suivant dans le fichier :

```
<?php
$file = "/var/www/html/dhcp.leases";

if (!file_exists($file)) {
    die("Fichier DHCP leases introuvable.");
}
$data = file($file);
$leases = [];

foreach ($data as $line) {
    if (preg_match('/lease (\d+\.\d+\.\d+\.\d+) {/', $line, $ip_match})) {
        $current_ip = $ip_match[1];
    } elseif (preg_match('/starts \d (\d{4}\/\d{2}\/\d{2} \d{2}:\d{2}:\d{2});/', $line, $start_match}) {
        $leases[$current_ip]['start'] = $start_match[1];
    } elseif (preg_match('/ends \d (\d{4}\/\d{2}\/\d{2} \d{2}:\d{2}:\d{2});/', $line, $end_match}) {
        $leases[$current_ip]['end'] = $end_match[1];
    } elseif (preg_match('/hardware ethernet ([0-9a-f:]+);/', $line, $mac_match)) {
        Sume ([0-9a-f:]+);/', $line, $mac_match}) {
        Sume ([0-9a-f:]+);/', $line, $mac_match)} {
        Sume ([0-9a-f:]+);/', $line, $mac_match}) {
        Sume ([0-9a-f:]+);/', $line, $mac_match)} {
        Sume ([0-9a-f:]+);/', $line, $mac_match)} {
        Sume ([0-9a-f:]+);/', $line, $mac_match}) {
        Sume ([0-9a-f:]+);/', $line, $mac_match})
```

```
$leases[$current_ip]['mac'] = $mac_match[1];
 } elseif (preg_match('/client-hostname "(.*?)";/', $line, $hostname_match)) {
   $leases[$current_ip]['hostname'] = $hostname_match[1];
 }
}
?>
<!DOCTYPE html>
<html lang="fr">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>DHCP Leases</title>
 <style>
   table { width: 100%; border-collapse: collapse; }
   th, td { border: 1px solid black; padding: 8px; text-align: left; }
   th { background-color: #f2f2f2; }
 </style>
</head>
<body>
 <h1>Liste des baux DHCP</h1>
 Adresse IP
     Nom de la machine
     Adresse MAC
     Début du bail
     Fin du bail
   <?php foreach ($leases as $ip => $info) : ?>
     <?= htmlspecialchars($ip) ?>
       <?= htmlspecialchars($info['hostname'] ?? 'Inconnu') ?>
       <?= htmlspecialchars($info['mac'] ?? 'Inconnu') ?>
       <?= htmlspecialchars($info['start'] ?? 'Inconnu') ?>
       <?= htmlspecialchars($info['end'] ?? 'Inconnu') ?>
     <?php endforeach; ?>
 </body>
</html>
```

Supprimer le fichier index.html :

rm /var/www/html/index.html

<u>Redémarrer</u> le service apache. service apache2 restart