

Serveur Web Raspberry Pi avec Nginx

Étape 1 : Connexion SSH

Définitions

Nginx → Un serveur web rapide. Il sert tes pages HTML/CSS aux visiteurs.

Nom de domaine → Une adresse lisible, genre monsite.duckdns.org, au lieu d'une IP.

Certificat SSL → Ce qui active le HTTPS (le cadenas). Ça chiffre les échanges entre ton site et les visiteurs.

Cron → Un petit programme qui lance des tâches automatiquement à un horaire régulier.

Activer SSH sur le Raspberry

```
sudo systemctl enable ssh  
sudo systemctl start ssh
```

Trouver l'IP du Raspberry

```
hostname -I
```

Note : Exemple `192.168.1.100`

Se connecter depuis Linux

```
ssh pi@192.168.1.100
```

Mot de passe par défaut : `raspberry`

Changer le mot de passe

```
passwd
```

Étape 2 : Installation Nginx

Mise à jour système

```
sudo apt update  
sudo apt upgrade -y
```

Installer Nginx

```
sudo apt install nginx -y
```

Vérifier

```
sudo systemctl status nginx
```

Tester dans navigateur : `http://192.168.1.100`

Étape 3 : Créer votre site HTML/CSS

Structure des dossiers

```
sudo mkdir -p /var/www/html/css
sudo mkdir -p /var/www/html/js
sudo mkdir -p /var/www/html/images
```

Page HTML

```
sudo nano /var/www/html/index.html
```

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mon Serveur</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="/">Accueil</a></li>
        <li><a href="/about.html">À propos</a></li>
        <li><a href="/contact.html">Contact</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <h1>Bienvenue sur mon serveur Raspberry Pi</h1>
    <p>Serveur web propulsé par Nginx</p>
  </main>
</body>
</html>
```

Feuille de style CSS

```
sudo nano /var/www/html/css/style.css
```

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}  
  
body {  
  font-family: Arial, sans-serif;  
  line-height: 1.6;  
  color: #333;  
}  
  
header {  
  background: #333;  
  color: #fff;  
  padding: 1rem;  
}  
  
nav ul {  
  list-style: none;  
  display: flex;  
  gap: 2rem;  
}  
  
nav a {  
  color: #fff;  
  text-decoration: none;  
}  
  
nav a:hover {  
  text-decoration: underline;  
}  
  
main {  
  max-width: 800px;  
  margin: 2rem auto;  
  padding: 0 1rem;  
}
```

```
h1 {  
    color: #333;  
    margin-bottom: 1rem;  
}
```

Permissions

```
sudo chown -R www-data:www-data /var/www/html  
sudo chmod -R 755 /var/www/html
```

Étape 4 : Configuration Nginx

Éditer la configuration

```
sudo nano /etc/nginx/sites-available/default
```

Configuration basique :

```
server {  
    listen 80;  
    listen [::]:80;  
  
    root /var/www/html;  
    index index.html index.htm;  
  
    server_name _;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
  
    # Compression  
    gzip on;  
    gzip_types text/plain text/css application/javascript;  
}
```

Tester et recharger

```
sudo nginx -t  
sudo systemctl reload nginx
```

Étape 5 : Nom de domaine

Option A : DuckDNS (gratuit)

1. Créer compte sur duckdns.org
2. Choisir sous-domaine : `monsite.duckdns.org`

Installer le client DuckDNS

```
mkdir ~/duckdns  
nano ~/duckdns/duck.sh
```

```
#!/bin/bash  
echo url="https://www.duckdns.org/update?domains=monsite&token=VOTRE-TOKEN&ip=" | curl -k -o  
~/duckdns/duck.log -K -
```

```
chmod +x ~/duckdns/duck.sh
```

Automatiser avec cron

```
crontab -e
```

Ajouter :

```
*/* * * * * ~/duckdns/duck.sh >/dev/null 2>&1
```

Configurer Nginx avec le domaine

```
sudo nano /etc/nginx/sites-available/default
```

Modifier `server_name` :

```
server_name monsite.duckdns.org;
```

Étape 6 : Certificat SSL avec Let's Encrypt

Installer Certbot

```
sudo apt install certbot python3-certbot-nginx -y
```

Obtenir le certificat

```
sudo certbot --nginx -d monsite.duckdns.org
```

- Email
- Accepter conditions (A)
- Choisir option 2 (redirect HTTP to HTTPS)

Vérifier renouvellement auto

```
sudo certbot renew --dry-run
```

Étape 7 : Sécurité basique

Pare-feu

```
sudo apt install ufw -y  
sudo ufw allow 22/tcp
```

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw enable
```

Redémarrer pour appliquer

```
sudo systemctl restart nginx
```

Commandes utiles

```
# Statut Nginx
sudo systemctl status nginx

# Logs
sudo tail -f /var/log/nginx/error.log
sudo tail -f /var/log/nginx/access.log

# Test config
sudo nginx -t

# Redémarrer
sudo systemctl restart nginx

# Température Raspberry
vcgencmd measure_temp
```

Résultat final

- **Site accessible** : `https://monsite.duckdns.org`
- **SSH** : `ssh pi@192.168.1.100`
- **Certificat SSL** : Renouvelé automatiquement
- **Pare-feu** : Actif

Revision #4

Created 2025-11-14 12:58:21 UTC by Flavien

Updated 2025-11-14 13:10:52 UTC by Flavien