

Etherpad Lite

Etherpad est un éditeur de texte collaboratif en temps réel, permettant à plusieurs personnes d'éditer un même document en même temps.

Installation

On va voir comment l'installer et le rendre disponible rapidement.

Selon les versions utilisées, les différentes étapes peuvent varier

Étape 1 : Installation des dépendances et préparation du serveur

Mettre d'abord à jour les paquets existants :

```
apt update  
apt upgrade
```

Installer ensuite **git**, **mariadb** et **nginx**, qui seront nécessaires au bon fonctionnement d'**etherpad**:

```
apt install git nginx mariadb-server
```

Enfin, installer **Node.js** et **npm**:

```
curl -fsSL https://deb.nodesource.com/setup_20.x | -E bash -  
apt install -y nodejs  
npm install -g pnpm
```



Pour purger une ancienne version de **Node JS** utiliser les commande suivantes:

```
rm -rf /usr/local/bin/npm /usr/local/share/man/man1/node* ~/.npm  
rm -rf /usr/local/lib/node*  
rm -rf /usr/local/bin/node*  
rm -rf /usr/local/include/node*  
apt purge nodejs npm  
apt-get purge nodejs npm  
apt autoremove
```

Il est également recommandé de créer un utilisateur dédié à l'exécution d'**Etherpad-lite**:

```
useradd -mU etherpad
```

Le serveur est maintenant configuré, on peut passer à la création de la base de données.

Étape 2 : Préparation de la base de données

Etherpad-lite peut fonctionner sans base de données dans sa configuration par défaut. Cela reste cependant destiné à des tests ou à un usage ponctuel. Il est alors préférable d'utiliser une base de données telle que MySQL.

Créer la base de données qui sera utilisée par **Etherpad-lite**. On la nomme ici **db_etherpad**:

```
mysql

MariaDB [(none)]> CREATE DATABASE db_etherpad;
Query OK, 1 row affected (0.001 sec)

#Créer ensuite l'utilisateur qui se connectera à la base de données et
donnez-lui tous les droits sur la DB :

MariaDB [(none)]> GRANT ALL privileges ON db_etherpad.* TO
'etherpad'@'localhost' IDENTIFIED BY 'MotDePasse';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH privileges;
Query OK, 0 rows affected (0.001 sec)

#Quitter MySQL (MariaDB) :

MariaDB [(none)]> \q
Bye
```

La base de données est prête, il ne reste qu'à installer **Etherpad-lite**, et le rendre accessible à l'adresse du serveur.

Étape 3 : Installation et configuration d'Etherpad-lite dans le dossier de l'utilisateur etherpad

Passer à l'utilisateur **etherpad**:

```
su - etherpad
```

Et se rendre dans le dossier '/opt'

```
cd /opt
```

Récupérer le code d'Etherpad-lite à l'aide de git :

```
git clone https://github.com/ether/etherpad-lite.git
```

Il faut ensuite exécuter le script **run.sh** qui installe les dépendances requises automatiquement. Il faut donc l'exécuter en tant que **root** pour cette première exécution :

```
cd etherpad-lite  
npm config set https-proxy http://xx.xx.xxx.xxx:3128  
su -c bin/run.sh
```

Quitter à l'aide de Ctrl + C

Changer ensuite l'environnement de **Node.js** pour passer en production :

```
NODE_ENV=production  
export NODE_ENV
```

Cette modification peut également être exécuté en une seule commande :

```
export NODE_ENV=production
```

Lors de l'exécution du script **run.sh**, certains fichiers ont été créés et appartiennent à l'utilisateur root. Ces fichiers sont nécessaires au bon fonctionnement d'Etherpad, que l'on va exécuter avec l'utilisateur **etherpad**. Pour cette raison, il est nécessaire de redéfinir la propriété de l'ensemble des fichiers pour l'utilisateur **etherpad** en soit le propriétaire.

Retour à l'utilisateur root :

```
exit
```

Et modification de la propriété :

```
chown -R etherpad:etherpad /home/etherpad/etherpad-lite
```

Retour à l'utilisateur etherpad :

```
su - etherpad
```

Editer le fichier de configuration settings.json :

```
vim etherpad-lite/settings.json
```

Pour définir la base de données créée plus tôt, remplacer les lignes suivantes :

```
"dbType": "dirty",
"dbSettings": {
  "filename": "var/dirty.db"
},
```

Par :

```
"dbType" : "mysql",
"dbSettings" : {
  "user": "etherpad",
  "host": "localhost",
  "port": 3306,
  "password": "MotDePasse",
  "database": "db_etherpad",
  "charset": "utf8mb4"
},
```



Pour utiliser une base de données sqlite:

1-cr  er la base de donn  es sqlite3 etherpad.db puis .quit pour quitter sqlite

2-changer le propri  taire de la base chown etherpad:etherpad etherpad.db

3- Dans le fichier de configuration indiquer "dbType": "sqlite", puis "filename": "etherpad.db"

Toujours dans ce m  me fichier, changer les identifiants administrateurs, qui seront utiles pour se connecter    l'administration d'Etherpad-lite :

```
"users": {
  "admin": {
    "password": "SuperMotDePasse",
    "is_admin": true
  },
},
```

Cr  er un nouveau fichier etherpad.service dans /etc/systemd/system/ avec comme contenu :

```
[Unit]
Description=etherpad (  dition collaborative de documents en temps r  el)
After=network.target mariadb.service
Wants=mariadb.service

[Service]
Type=simple
WorkingDirectory=/opt/etherpad-lite
User=etherpad
```

```
Group=etherpad
Environment=NODE_ENV=production
ExecStart=/opt/etherpad-lite/bin/run.sh --root
ExecStop=/bin/kill -TERM $MAINPID
ExecReload=/bin/kill -HUP $MAINPID
Restart=on-failure
RestartSec=15
ExecStartPre=/bin/mkdir -p /run/etherpad
ExecStartPre=/bin/chown node:node /run/etherpad
ExecStartPre=/bin/chmod 755 /run/etherpad
PermissionsStartOnly=true
TimeoutStopSec=600
PIDFile=/run/etherpad.pid

[Install]
WantedBy=multi-user.target
```

Exécuter `systemctl daemon-reload` puis `systemctl enable etherpad` pour activer le service au démarrage enfin `systemctl start etherpad` pour démarrer le service.

Si tout est en ordre, **etherpad-lite** va démarrer correctement.

Une instance **etherpad** est maintenant en cours d'exécution et est accessible à l'adresse `http://127.0.0.1:9001`

Étape 4 : Configuration de Nginx pour assigner l'adresse de votre choix à Etherpad

En tant que root :

```
vi /etc/nginx/sites-available/etherpad.conf
```

Contenu :

```
upstream etherpad {
    server localhost:9001;
}

server {
    listen      443 ssl http2;
    listen      [::]:443 ssl http2;
    server_name _; # set domain name here if using one

    access_log  /var/log/nginx/access.log;
    error_log   /var/log/nginx/error.log;

    include /etc/nginx/snippets/ssl.conf;
```

```

location / {
    proxy_pass          http://etherpad;
    proxy_buffering     off; # attention, cette ligne ne remplace aucun
proxy_buffering défini dans un fichier conf.d/file.conf
    proxy_set_header    Host $host;
    proxy_pass_header    Server;

    # Note you might want to pass these headers etc too.
    proxy_set_header     X-Real-IP $remote_addr; #
https://nginx.org/en/docs/http/nginx_http_proxy_module.html
    proxy_set_header     X-Forwarded-For $remote_addr; # EP pour afficher
l'adresse IP distante réelle dans les logs
    proxy_set_header     X-Forwarded-Proto $scheme; # pour que EP
définisse un indicateur de cookie sécurisé lorsque https est utilisé
    proxy_http_version  1.1; # recommandé avec des connexions keepalive

    # WebSocket proxying - from
https://nginx.org/en/docs/http/websocket.html
    proxy_set_header     Upgrade $http_upgrade;
    proxy_set_header     Connection $connection_upgrade;
}
}

# on est ici dans le contexte http
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''          close;
}

# enforce HTTPS
server {
    listen      80;
    listen      [::]:80;
    return 302   https://$host$request_uri; # changer ceci en 301 lorsqu'on
utilise un domaine
}

```

Il ne reste plus qu'à activer le site :

```
ln -s /etc/nginx/sites-available/etherpad.conf /etc/nginx/sites-enabled/etherpad.conf
```

Et redémarrer **Nginx**:

```
systemctl restart nginx
```

Si tout s'est bien passé, **Etherpad** est disponible à l'adresse <https://pad.ceci.estun.example.com>, et on peut accéder à l'interface d'administration à partir de <https://pad.ceci.estun.example.com/admin>.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=centos:etherpad>

Last update: **2025/02/19 10:59**

