

# DemocracyOS sur Ubuntu 16.04

## Caractéristiques des serveurs

| Serveur de développement | Serveur de production |
|--------------------------|-----------------------|
| 1 ou 2 CPU               | 4 CPU                 |
| 1GO de RAM               | 4GO de RAM            |
| Espace disque 10GO       | Espace disque 10GO    |

Ces caractéristiques de serveurs restent relatives à la base de données et au nombre de participants attendus (connexions sur le serveur).

## Procédure d'installation

*Tout se passe dans le terminal.*

### Configuration initiale d'un serveur Ubuntu 16.04

1. Première étape - Connexion avec root

```
# ssh root@your_server_ip
```

2. Deuxième étape - Créer un nouvel utilisateur

```
# adduser marouen
```

### 3. Troisième étape - Privilèges de root : ajoutez votre nouvel utilisateur au groupe sudo

```
# usermod -aG sudo marouen
```

### 4. Étape 4 - Ajouter une authentification par clé publique (recommandé)

Générer une paire de clés avec :

```
# ssh-keygen
```

Copiez la clé publique :

```
# ssh-copy-id marouen@your_server_ip
```

Après avoir fourni votre mot de passe dans le terminal, votre clé publique sera ajoutée à l'utilisateur distant `.ssh/authorized_keys`. La clé privée correspondante peut maintenant être utilisée pour se connecter au serveur.

### 5. Étape 5 - Désactiver l'authentification par mot de passe (recommandé)

```
# sudo nano /etc/ssh/sshd_config
```

Trouvez la ligne qui spécifie **PasswordAuthentication**. Décommentez-le en supprimant le `#` précédent, puis changez sa valeur en "non". Cela devrait ressembler à ceci après que vous ayez fait le changement:

```
# PasswordAuthentication no
```

Lorsque vous avez terminé vos modifications, enregistrez et fermez le fichier en utilisant la méthode que nous avons utilisée plus tôt.

Tapez ceci pour recharger le SSH daemon:

```
# sudo systemctl reload sshd
```

## 6. Sixième étape - Tester la Connexion

```
# ssh marouen@your_server_ip
```

Rappelez-vous, si vous avez besoin d'exécuter une commande avec les privilèges root, tapez "sudo" avant cela comme ceci:

```
# sudo command_to_run
```

## 7. Étape sept - Configurer un pare-feu de base

```
# sudo ufw app list  
# sudo ufw allow OpenSSH  
# sudo ufw enable  
# sudo ufw status
```

Pour plus de détails, vous pouvez visiter :

<https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-16-04>

# Installer Nginx sur Ubuntu 16.04

## 1. Étape 1: Installer Nginx

```
# sudo apt-get update
# sudo apt-get install nginx
```

## 2. Étape 2: Ajustez le pare-feu

```
# sudo ufw app list
# sudo ufw allow 'Nginx HTTP'
# sudo ufw status
```

## 3. Étape 3: Vérifiez votre serveur Web

```
# systemctl status nginx
```

Entrez l'adresse IP ou le domaine de votre serveur dans la barre d'adresse de votre navigateur :

Vous devriez voir la page Nginx par défaut, qui devrait ressembler à ceci :

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

Pour plus de détails, vous pouvez visiter :

<https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-16-04>

# Sécurisez Nginx avec Let's Encrypt sur Ubuntu 16.04

## 1. Étape 1 - Installation de Certbot

```
# sudo add-apt-repository ppa:certbot/certbot
# sudo apt-get update
# sudo apt-get install python-certbot-nginx
```

## 2. Étape 2 - Configuration de Nginx

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

Recherchez la ligne **server\_name** existante et remplacez le trait de soulignement   par votre nom de domaine :

```
server_name example.com www.example.com;
```

Enregistrez le fichier et quittez votre éditeur.

Ensuite, vérifiez la syntaxe de vos modifications de configuration :

```
# sudo nginx -t
```

Une fois la syntaxe de votre configuration correcte, rechargez Nginx pour charger la nouvelle configuration :

```
# sudo systemctl reload nginx
```

## 3. Étape 3 - Autoriser HTTPS à travers le pare-feu

```
# sudo ufw allow 'Nginx Full'
# sudo ufw delete allow 'Nginx HTTP'
# sudo ufw status
```

## 4. Étape 4 - Obtenir un certificat SSL

```
# sudo certbot --nginx -d example.com -d www.example.com
```

## 5. Vérification du renouvellement automatique de Certbot

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

Pour plus de détails, vous pouvez visiter :

<https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-16-04>

## Installer et sécuriser MongoDB sur Ubuntu 16.04

### Première partie: Configuration du serveur

#### 1. Étape 1 - Ajout du référentiel MongoDB

```
# sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv  
0C49F3730359A14518585931BC711F9BA15703C6  
  
# echo "deb [ arch=amd64,arm64 ] http://repo.mongodb.org/apt/ubuntu  
xenia/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-  
org-3.4.list  
  
# sudo apt-get update
```

#### 2. Étape 2 - Installation de MongoDB

```
# sudo apt-get install mongodb-org  
# sudo systemctl start mongod  
# sudo systemctl status mongod  
# sudo systemctl enable mongod
```

## Deuxième partie: sécuriser MongoDB

### 1. Étape 1 - Ajout d'un utilisateur administrateur

```
# mongo
# use admin
# db.createUser(
  {
    user: "AdminMarouen",
    pwd: "AdminMarouen'sSecurePassword",
    roles: [ { role: "userAdminAnyDatabase", db: "admin" } ,
  { "role" : "readWrite", "db" : "DemocracyOS-dev" }
  ]
}
)
```

Tapez 'exit' et appuyez sur ENTRÉE ou utilisez CTRL + C pour quitter le client.

### 2. Étape 2 - Activation de l'authentification

```
# sudo nano /etc/mongod.conf
```

Dans la section **#security**, nous allons supprimer le hachage devant la sécurité pour activer la strophe. Ensuite, nous allons ajouter le paramètre d'autorisation. Lorsque nous aurons terminé, les lignes devraient ressembler à l'extrait ci-dessous :

```
. . .
security:
  authorization: "enabled"
. . .
```

Une fois que nous avons sauvegardé et quitté le fichier, nous redémarrons le daemon:

```
# sudo systemctl restart mongod
```

Si nous avons fait une erreur dans la configuration, le daemon ne démarre pas. Puisque systemctl ne fournit pas de sortie, nous utiliserons son option status pour être sûr que c'est bien le cas:

```
# sudo systemctl status mongod
```

### 3. Étape 3 - Vérification de la restriction des utilisateurs non authentifiés

```
# mongo
# show dbs

Output
2017-02-21T19:20:42.919+0000 E QUERY [thread1] Error: listDatabases failed:{"ok" : 0, "errmsg" : "not authorized on admin to execute command { listDatabases: 1.0 }", "code" : 13, "codeName" : "Unauthorized"
. . .
```

Quittons le shell pour continuer :

```
# exit
```

### 4. Étape 4 - Vérification de l'accès de l'utilisateur administrateur

```
# mongo -u AdminMarouen -p --authenticationDatabase admin
# show dbs

Output
admin 0.000GB
local 0.000GB
```

Tapez exit ou appuyez sur CTRL + C pour quitter.

Pour plus de détails, vous pouvez visiter :

<https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-mongodb-on-ubuntu-16-04>

# Configurer DemocracyOS en mode production sur Ubuntu 16.04

## 1. Installer Node.js

```
# cd ~  
# curl -sL https://deb.nodesource.com/setup_6.x -o nodesource_setup.sh  
# sudo bash nodesource_setup.sh
```

Le PPA sera ajouté à votre configuration et votre cache de package local sera mis à jour automatiquement. Après l'exécution du script d'installation à partir de nodesource, vous pouvez installer le package Node.js de la même manière que ci-dessus:

```
# sudo apt-get install nodejs
```

installer le paquet build-essential :

```
# sudo apt-get install build-essential
```

## 2. Cloner DemocracyOS de GitHub

```
# git clone -b alt/agora-dev https://github.com/democracy-os-fr/democracyos.git
```

## 3. Configurer DemocracyOS

Pour installer les dépendances run :

```
# sudo make packages
```

Configurer DemocracyOS :

Toutes les variables de configuration se trouvent dans le fichier “/ config / defaults.json”.

Si vous voulez surcharger l'un d'entre eux, vous devez créer un nouveau fichier appelé “development.json” dans le même répertoire et réécrire le nom et la valeur que vous voulez.

Il faut ajouter des membres à la variable “staff”. Ce sont les futurs Super Administrateurs de notre application.

Il faut ajouter l'utilisateur à qui on a donné les privilèges sur la base DemocracyOS-dev pour pouvoir communiquer avec mongoDB.

Pour la partie "notifications" et "nodemailer" Il faut avoir un client SMTP configuré et autoriser l'adresse ip du serveur à communiquer avec ce client.

Exemple de fichier de configuration :

```
{
  "locale": "fr",
  "protocol": "https",
  "warnHTTP": true,
  "host": "example.com",
  "publicPort": 443,
  "https": {
    "port": 443,
    "redirect": "no-redirect"
  },
  "mongoUrl": "mongodb://<login>:<password>@127.0.0.1:27017/DemocracyOS-dev?
authSource=admin",
  "mongoUsersUrl": "",
  "jwtSecret":
"fde8cfed56bc2bd4e259b8f4841e9ac8b50b904a0728ae07a9fc6994eba5801f",
  "corsDomains": [],
  "signinUrl": "",
  "signupUrl": "",
  "settingsUrl": "",
  "staff": ["marouen@democracyos.eu"],
  "allowEmailAliases": true,
  "facebookSignin": false,
  "googleSignin": false,
  "ldapSignin": false,
  "publicEmails": false,
  "moderatorEnabled": false,
  "generatorEnabled": false,
  "checkNodeVersion": true,
  "auth": {
    "basic": {
      "username": "",
      "password": ""
    },
    "facebook": {
      "clientID": "",
      "clientSecret": "",
      "permissions": [ "email" ]
    },
    "google": {
      "clientID": "",
      "clientSecret": ""
    },
    "ldap": {
      "options": {
        "server": {
          "url": "",
          "bindDN": "",
          "bindCredentials": "",
          "searchBase": "",
          "searchFilter": ""
        }
      }
    }
  }
}
```

```
    },
    "domainWhiteList": []
  }
},
"certbot": {
  "key": "",
  "token": ""
},
"notifications": {
  "mailer": {
    "name": "DemocracyOS",
    "email": "example@democracyos.eu"
  },
  "nodemailer": {
    "host": "smtp-relay.gmail.com",
    "port": 587,
    "logger": true,
    "debug": true,
    "auth": {
      "user": "<votre login SMTP>",
      "pass": "<votre mot de passe SMTP>"
    }
  }
}
},
"slack": {
  "apiErrorWebhookUri": ""
},
"recaptchaSite": "",
"recaptchaSecret": "",
"blackListEmails": [],
"client": [
  "protocol",
  "host",
  "publicPort",
  "env",
  "locale",
  "logo",
  "logoMobile",
  "favicon",
  "organizationName",
  "organizationUrl",
  "fontFamily",
  "fontFamilyLink",
  "learnMoreUrl",
  "googleAnalyticsTrackingId",
  "frequentlyAskedQuestions",
  "termsOfService",
  "moderatorEnabled",
  "generatorEnabled",
  "privacyPolicy",
  "legalTerms",
  "glossary",
  "signinUrl",
  "signupUrl",
  "settingsUrl",
  "homeLink",
  "headerContrast",
  "headerBackgroundColor",
  "headerFontColor",
  "facebookSignin",
  "googleSignin",
```

```
"visibility",
"usersWhitelist",
"domainWhitelist",
"multiForum",
"restrictForumCreation",
"tweetText",
"segmentKey",
"spamLimit",
"availableLocales",
"warnHTTP",
"recaptchaSite",
"extra",
"euCookieCompliance",
"sorts",
"piwik",
"socialshare",
"defaultCommentsLength",
"maxCommentsLength",
"showReplies"
],
"socialshare": {
  "siteName": "DemocracyOS",
  "siteDescription": "DemocracyOS voting system",
  "image": "https://2.gravatar.com/avatar/a8b9176bd0d042db078bf38500727671?
d=https%3A%2F%2Fidenticons.github.com
%2F14bae9f2564556f8eb23cc263a779f59.png&s=400",
  "domain": "democracyos.org",
  "twitter": {
    "username": "@democracyos"
  }
}
},
"logo": "/lib/boot/logo.png",
"logoMobile": "/lib/boot/logo-mobile.png",
"favicon": "/lib/boot/favicon.ico",
"fontFamily": "'Roboto', 'Helvetica Neue', Helvetica, Arial, sans-serif",
"fontFamilyLink": "https://fonts.googleapis.com/css?
family=Roboto:300,400,700&subset=cyrillic,cyrillic-ext,latin,greek-
ext,greek,latin-ext",
"organizationName": "Titre de la consultation",
"organizationEmail": "email@domain.net",
"organizationUrl": "https://domaine.net",
"learnMoreUrl": "",
"googleAnalyticsTrackingId": "",
"spamLimit": 5,
"defaultCommentsLength": 4096,
"maxCommentsLength": 10000,
"showReplies": false,
"frequentlyAskedQuestions": false,
"termsOfService": false,
"privacyPolicy": false,
"legalTerms": false,
"glossary": false,
"homeLink": "/",
"headerBackgroundColor": "#fff",
"headerFontColor": "#000",
"headerContrast": true,
"googleAPIKey": "",
"visibility": "visible",
"usersWhitelist": true,
"strictWhitelist": false,
"domainWhitelist": [],
"multiForum": true,
```

```

"restrictForumCreation": false,
"defaultForum": "",
"tweetText": "",
"forceSafeImageURLs": true,
"segmentKey": "",
"availableLocales": ["en","fr"],
"enforceLocale": true,
"extra": {
  "user": {
    "age": false,
    "job": false,
    "org": false,
    "postal": false
  }
},
"euCookieCompliance": true,
"sorts": {
  "home": {
    "default": "by-title-asc",
    "random": false
  },
  "multi": {
    "default": "newest-first",
    "random": false
  },
  "sidebar": {
    "default": "closing-soon",
    "random": false
  },
  "comment": {
    "default": "score",
    "random": false
  },
  "admin": {
    "default": "recently-updated",
    "random": false
  }
},
"piwik": {
  "url": "",
  "id": ""
}
}

```

#### 4. Configurer Nginx en tant que serveur proxy inverse

```

# cd /etc/nginx/sites-available/
# mkdir democracyos
# sudo nano /etc/nginx/sites-available/democracyos

```

Exemple de fichier de configuration pour le Reverse Proxy :

```
server {  
  
    root /home/marouen/democracyos/public;  
  
    # Add index.php to the list if you are using PHP  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name democracyos.example.com;  
  
    location / {  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-Host $host:$server_port;  
        proxy_set_header X-Forwarded-Server $host;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-NginX-Proxy true;  
        proxy_set_header X-Forwarded-Ssl on;  
        proxy_set_header X-Forwarded-Port 443 ;  
        proxy_pass http://localhost:3000;  
        proxy_ssl_session_reuse off;  
        proxy_set_header Host $http_host;  
        proxy_cache_bypass $http_upgrade;  
        proxy_redirect off ;  
    }  
}
```

Assurez-vous que vous n'avez pas introduit d'erreurs de syntaxe en tapant:

```
# sudo nginx -t
```

Ensuite, redémarrez Nginx:

```
# sudo systemctl restart nginx
```

Créer un lien symbolique de `/etc/nginx/sites-available/democracyos`  
vers `/etc/nginx/sites-enabled/`

```
# ln -s /etc/nginx/sites-available/democracyos /etc/nginx/sites-enabled/
```

## 5. Créer un fichier pour lancer DemocracyOS

Exemple de script de lancement : "ecosystem.config.json"

```
{
  "apps": [{
    "name": "democracyos",
    "cwd": "/home/marouen/democracyos",
    "script": "./index.js",
    "watch": false,
    "exec_mode": "fork",
    "instances": 0,
    "env": {
      "NODE_ENV": "development",
      "NODE_PATH": ".",
      "DEBUG": "democracyos*",
      "PORT": "3000",
    },
    "node_args": "--max-old-space-size=4096"
  }
]
```

## 6. Installer PM2

Nous utiliserons npm, un gestionnaire de paquets pour les modules Node qui s'installe avec Node.js, pour installer PM2 sur notre serveur. Utilisez cette commande pour installer PM2:

```
# sudo npm install -g pm2
```

Manage Application with PM2 :

```
# pm2 start ecosystem.config.json
```

La sous-commande startup génère et configure un script de démarrage pour lancer PM2 et ses processus gérés sur les amorçages du serveur :

```
# pm2 startup systemd
```

La dernière ligne de la sortie résultante inclura une commande que vous devez exécuter avec les privilèges super-utilisateur :

```
# sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd  
-u marouen --hp /home/marouen
```

Cela va créer une unité systemd qui exécute pm2 pour votre utilisateur au démarrage. Cette instance pm2, à son tour, exécute ecosystem.config.json. Vous pouvez vérifier l'état de l'unité systemd avec systemctl :

```
# systemctl status pm2-marouen
```

Pour plus de détails, vous pouvez visiter :

<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-16-04>