

# Guide pour la version production

---

**Attention:** Les commandes doivent être adaptées selon votre cas spécifique.

## Mise en service

Générer le build de la version de développement

```
npm run build
```

Téléverser le contenu du dossier build vers le serveur

```
scp -r -P 1322 /path/to/your/project/build bob_tpiorganiser@blue.section-  
inf.ch:/home/bob_tpiorganiser/tpiorganiser2023
```

Téléversez également le contenu des différents dossiers disponibles à la racine de src :

```
scp -r -P 1322 /path/to/your/project/src/srvMongoDB.js  
bob_tpiorganiser@blue.section-  
inf.ch:/home/bob_tpiorganiser/tpiorganiser2023/server.js  
scp -r -P 1322 /path/to/your/project/.env bob_tpiorganiser@blue.section-  
inf.ch:/home/bob_tpiorganiser/tpiorganiser2023/.env  
scp -r -P 1322 /path/to/your/project/src/config bob_tpiorganiser@blue.section-  
inf.ch:/home/bob_tpiorganiser/tpiorganiser2023/config
```

Ces dossiers contiennent des variables et des sauvegardes nécessaires à l'application.

## Configuration du serveur

### Dépendances nécessaires

Installez les dépendances suivantes dans le répertoire `/home/bob_tpiorganiser/tpiorganiser2023` :

```
cd /home/bob_tpiorganiser/tpiorganizer2023/  
npm install express cors dotenv mongoose nodemailer date-fns
```

### Script de démarrage du serveur

Créez un script à la racine du projet (`start_server.sh`) pour lancer le serveur Node.js :

```
#!/bin/bash

# Démarrer le serveur Node.js
node server.js
```

## Configuration du service

Créez un service (`tpiorganiser.service`) dans le répertoire `/etc/systemd/system/` :

```
[Unit]
Description=Service TPIorganiser
After=network.target

[Service]
Type=simple
WorkingDirectory=/home/bob_tpiorganiser/tpiorganizer2023/
ExecStart=/home/bob_tpiorganiser/tpiorganizer2023/start_server.sh
Restart=always
User=bob_tpiorganiser
Group=bob_tpiorganiser

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

## Configuration de Nginx

Modifiez le comportement de Nginx pour rediriger les requêtes des visiteurs (experts, chefs de projet) vers nos applications (ports 6000 et 3000).

```
sudo nano /etc/nginx/sites-enabled/blue.section-inf.ch
```

```
server {
    listen 80;
    server_name blue.section-inf.ch;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    location /api {
        proxy_pass http://localhost:6000;
```

```
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
}
```

## Création d'un lien symbolique vers le dossier nginx/sites-available

```
sudo ln -s /etc/nginx/sites-available/blue.section-inf.ch /etc/nginx/sites-enabled/blue.section-inf.ch
```

## Quelques pistes en cas de problème

Lorsque vous rencontrez des problèmes avec votre application, voici quelques étapes à suivre pour résoudre les problèmes :

- Vérifiez l'état des différents services :
  - Assurez-vous que Nginx est opérationnel pour gérer toutes les requêtes HTTP(s).
  - Vérifiez que le service Node.js est en cours d'exécution pour gérer l'API et l'application. Vous devriez voir des messages de demande dans les journaux de Node.js lorsque des utilisateurs se connectent. Si aucun message n'apparaît, cela peut indiquer que le service n'est pas démarré correctement.
- Si l'application semble fonctionner mais que des données sont manquantes :
  - Il est possible que le serveur ou le service MongoDB soit arrêté. Vérifiez le statut de MongoDB et relancez le service si nécessaire en utilisant les commandes appropriées ci-dessous :

```
sudo service mongod status
sudo service mongod restart
```

En suivant ces étapes, vous pourrez identifier et résoudre les problèmes potentiels pour assurer le bon fonctionnement de votre application.

## Mongosh

Pour utiliser l'interpréteur de commandes MongoDB (mongosh) et interagir avec votre base de données, suivez ces étapes :

1. **Lancement de Mongosh** : Lancez l'interpréteur de commandes MongoDB en saisissant la commande suivante dans votre terminal :

```
mongosh
```

2. **Connexion à la base de données** : Connectez-vous à votre base de données en utilisant la commande `use` suivie du nom de votre base de données. Par exemple, si votre base de données est nommée `dbOrganizer`, vous pouvez vous connecter ainsi :

```
use dbOrganizer
```

3. **Affichage des collections** : Après avoir sélectionné votre base de données, vous pouvez afficher la liste des collections disponibles en utilisant la commande suivante :

```
show collections
```

Cela affichera toutes les collections présentes dans votre base de données `dbOrganizer`.

4. **Affichage du contenu d'une collection** : Une fois connecté à la base de données, vous pouvez afficher le contenu de la collection `tpiSoutenance_2023` en utilisant la commande `find()` sur cette collection. Voici deux exemples :

```
db.tpiSoutenance_2023.find()
```

ou

```
db.tpiSoutenance_2023.find().pretty()
```

La première commande affiche les documents de la collection sans mise en forme spéciale, tandis que la seconde les formate de manière plus lisible.