

**Reconnaissance automatique des matricules et
compilation des notes pour optimiser le travail des
enseignants**

Guide d'installation

Version 1.0

Guide d'installation	3
Code Source	3
Docker	3
Minikube	4
Keda	4
Nginx	5

Guide d'installation

Ce guide d'installation présente les étapes à suivre pour une configuration à long terme du logiciel, désignée pour un environnement en production.

Code Source

Le code source du logiciel est accessible à partir de Gitlab. Il est disponible à l'adresse suivante:

<https://gitlab.com/polytechnique-montr-al/log89xx/22-3/equipe-1/LOG89XX-1>

Pour passer à la prochaine étape, il est requis de récupérer le code source du logiciel à l'aide de *Git*:

```
git clone https://gitlab.com/polytechnique-montr-al/log89xx/22-3/equipe-1/LOG89XX-1.git
```

Configuration à long terme (environnement de production)

La configuration à long terme est désignée pour une installation du logiciel dans un environnement de production faisant usage des conteneurs sous *Docker* orchestré par *Kubernetes*.

Docker

Tel que mentionné, pour configurer l'environnement pour l'exécution du logiciel, il faut préalablement installer Docker. Si ce n'est pas déjà fait, suivez le guide à partir du lien suivant:

<https://docs.docker.com/engine/install/ubuntu/>

Assurez-vous également d'avoir un compte Docker Hub pour la suite de l'installation:

<https://hub.docker.com/>

Ouvrez ensuite le fichier *docker-compose.yml* dans le répertoire */LOG89XX-1/* et remplacez le nom d'utilisateur pour les images du *executor*, *server* et *socketio*.

Par la suite, veuillez créer les images en exécutant les commandes suivantes:

```
docker compose up
```

Une fois les images créées, il faut les téléverser sur *Docker Hub*. Pour ce faire, il faut exécuter les commandes suivantes:

```
docker push <nom usager>/executor:latest  
docker push <nom usager>/server:latest  
docker push <nom usager>/socketio:latest
```

Déplacez-vous ensuite dans le répertoire *LOG89XX-1/Projet/cadriciel* pour générer l'image de l'application web en exécutant les commandes suivantes:

```
docker build -t <nom usager>/rmn-webapp:latest .  
docker push <nom usager>/rmn-webapp:latest
```

Minikube

Minikube sert d'orchestrateur des différentes images docker. L'installation de *Minikube* est requise à partir du lien suivant:

<https://minikube.sigs.k8s.io/docs/start/>

Démarrer ensuite un *cluster Minikube* en exécutant la commande suivante:

```
minikube start --driver=docker --memory=<memoire désirée> --cpus=<nombre  
de cpu désiré>
```

Keda

L'automatisation et la gestion de la file de tâches est faite par Keda. Keda s'occupe de surveiller la file de tâches et de déployer un cluster d'exécution de tâches dès qu'il y a une tâche dans la file et que les ressources disponibles le permettent.

```
kubectl apply -f  
https://github.com/kedacore/keda/releases/download/v2.9.3/keda-2.9.3.yaml
```

Déplacez-vous dans le répertoire *LOG89XX-1/deployment* et modifiez le nom d'utilisateur *Docker Hub* pour chacun des fichiers *yaml*. Veuillez créer ensuite les *Pods* avec la commande suivante:

```
kubectl apply -f .
```

Nginx

Veuillez récupérer par la suite les adresses *URL* de chacun des services en exécutant les commandes suivantes:

```
minikube service rmn-webapp --url  
minikube service server --url  
minikube service socketio --url
```

Veuillez noter les adresses *IP* ainsi que les ports associés. Ils seront nécessaires pour l'étape suivante.

Voici un exemple d'*URL* retourné par l'exécution de la commande ci-haut:

```
projet@pi4-log01-a:~$ projet@pi4-log01-a:~$ minikube service rmn-webapp --url  
http://192.168.49.2:32206
```

Une fois l'application web déployée sur *Minikube*, elle sera assignée à une adresse IP. Cette adresse est le point d'entrée de l'application. Il est possible d'exposer l'adresse et le port assigné automatiquement pour accéder à l'application de l'externe.

Il est possible que le serveur ne permette pas d'exposer tous les ports à l'externe. Ainsi, il faut passer par un proxy inverse. *Nginx* est un proxy inverse; il capte le trafic d'un port (Ex: 80) pour le rediriger vers un port voulu (Ex: 32206). Cette façon de faire permet un contrôle sécuritaire des requêtes venant de l'externe.

Récupérez également l'adresse IP publique de votre système. Sous Linux, vous pouvez exécuter la commande suivante:

```
curl https://ipinfo.io/ip
```

L'installation de *Nginx* peut se faire en exécutant les commandes suivantes:

```
sudo apt update
sudo apt install nginx
```

Veillez ouvrir le fichier `/etc/nginx/sites-available/default` afin de configurer votre serveur *Nginx* et copier la configuration ci-dessous en prenant soin de remplacer les adresses *IP* (et les ports associés).

```
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    include snippets/self-signed.conf;
    include snippets/ssl-params.conf;

    root /var/www/html;

    index index.html index.htm index.nginx-debian.html;

    server_name _;
    location / {
        proxy_pass <adresse IP rmn-webapp>:<PORT rmn-webapp>;
    }

    location /new-correction {
        proxy_pass <adresse IP rmn-webapp>:<PORT rmn-webapp>;
    }

    location /api/ {
        proxy_pass <adresse IP server>:<PORT server>;
    }

    location /socket.io/ {
        proxy_pass <adresse IP socketio>:<PORT
socketio>/socket.io/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}

server {
    listen 80;
```

```
listen [::]:80;

server_name <adresse IP de votre système>;

return 302 https://$server_name$request_uri;
}
```

Veuillez noter qu'il faut également un certificat SSL pour exécuter le logiciel sous le serveur *Nginx*. Si vous ne possédez pas de certificat SSL, suivez le guide au lien suivant pour créer un certificat auto signé:

<https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-nginx-in-ubuntu-16-04>

Par la suite, veuillez redémarrer le serveur *Nginx* pour que les changements prennent effet.

```
sudo systemctl restart nginx
```

Vous pouvez maintenant accéder au logiciel à partir d'un navigateur en entrant l'adresse *IP* de votre système comme lien.

