

GUIDE DE DÉPLOIEMENT

Ce document explique la procédure à suivre afin de déployer l'application GestionParcAuto créée dans le cadre du TPI Gestion de parc automobile d'entreprise, sur un serveur Ubuntu 24.04.

INSTALLATION DE .NET 9 PAR UBUNTU PPA

.NET 9 n'étant pas disponible sur Ubuntu 24.04, il est nécessaire de l'ajouter via PPA avec la commande suivante.

```
sudo add-apt-repository ppa:dotnet/backports
```

Il faut ensuite ajouter le sdk et les runtimes

```
sudo apt install dotnet9
sudo apt-get install -y dotnet-sdk-9.0
sudo apt-get install -y aspnetcore-runtime-9.0
sudo apt-get install -y dotnet-runtime-9.0
```

INSTALLATION DE NGINX

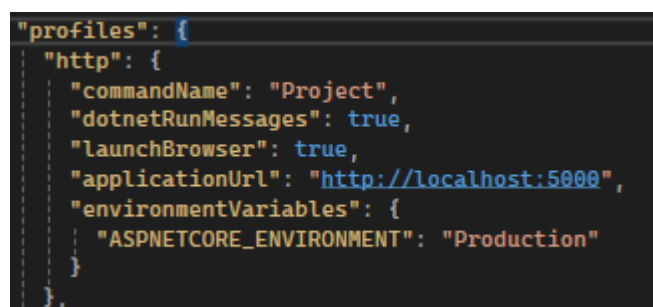
Il faut l'installer avec les commandes suivantes :

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

PARAMÉTRAGES

PARAMÉTRAGE DE L'APPLICATION

Dans le fichier **launchSettings.json** dans l'applicationUrl utilisez le port 5000.



```
"profiles": {
  "http": {
    "commandName": "Project",
    "dotnetRunMessages": true,
    "launchBrowser": true,
    "applicationUrl": "http://localhost:5000",
    "environmentVariables": {
      "ASPNETCORE_ENVIRONMENT": "Production"
    }
  }
},
```

FIGURE 1 PROFILE DE LANCEMENT

PARAMÉTRAGE DE NGINX

Ouvrir en édition le fichier suivant => /etc/nginx/sites-available/default et ajouter la config suivante :

```
server {  
    listen      80;  
    root /var/www/publish;  
    location / {  
        proxy_pass      http://localhost:5000;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection keep-alive;  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
}
```

Rafraichir le service

```
sudo service nginx restart
```

PUBLIER L'APPLICATION

En faisant clic droit sur le projet cliquer sur publier

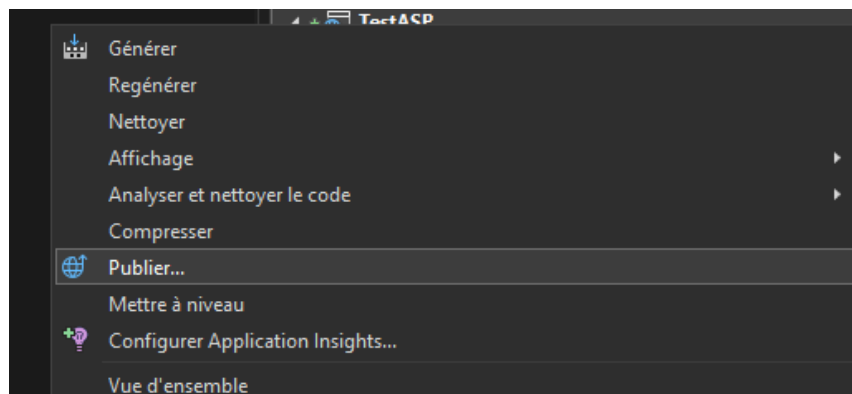


FIGURE 2 PUBLIER L'APPLICATION

Créer un profil dossier. Publier et récupérer le dossier généré appelé publish. Créer une archive zip de ce dossier et l'ajouter à une release.

Sur le serveur aller sous /var/www et exécuter la commande :

```
sudo wget -N https://github.com/ederiazl/TPI-Gestion-de-parc-automobile-d-entreprise/releases/download/{Nom de la release}/publish.zip
```

Exécuter la commande :

```
sudo unzip publish.zip
```

Pour dézipper le dossier. Se déplacer dans le dossier dézippé et exécuter la commande suivante :

```
dotnet {nom du projet}.dll
```

Pour lancer l'application.

Contrôler que l'application se lance bien.

CRÉER LE SERVICE

Si le service existe déjà, il n'est pas nécessaire de le recréer mais il faudra le redémarrer avec la commande :

```
Sudo systemctl restart GestionParcAuto.service
```

Sous /etc/systemd/system/ créer un fichier GestionParcAuto.service avec le contenu suivant :

```
[Unit]
Description=Application de gestion de parc automobile d'entreprise
[Service]
WorkingDirectory=/var/www/publish
ExecStart=/usr/bin/dotnet /var/www/publish/GestionParcAuto.dll
Restart=always
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=dotnet-example
User=www-data
Environment=ASPNETCORE_ENVIRONMENT=Production
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false
[Install]
WantedBy=multi-user.target
```

Puis ajouter et démarrer le service.

```
sudo systemctl enable project_name.service
sudo systemctl start project_name.service
```