AERO-WEBB
Technical Direction Docs

How to make aero webb server on ubuntu wsl visible on vpn

How to make aero webb server on ubuntu wsl visible on vpn

Cette note décrit la mise en place de **port forwarding** sur la machine Windows vers WSL afin de rendre un environnement Aero-Webb dans une distribution Ubuntu tournant sous WSL visible sur le VPN.

Abstract

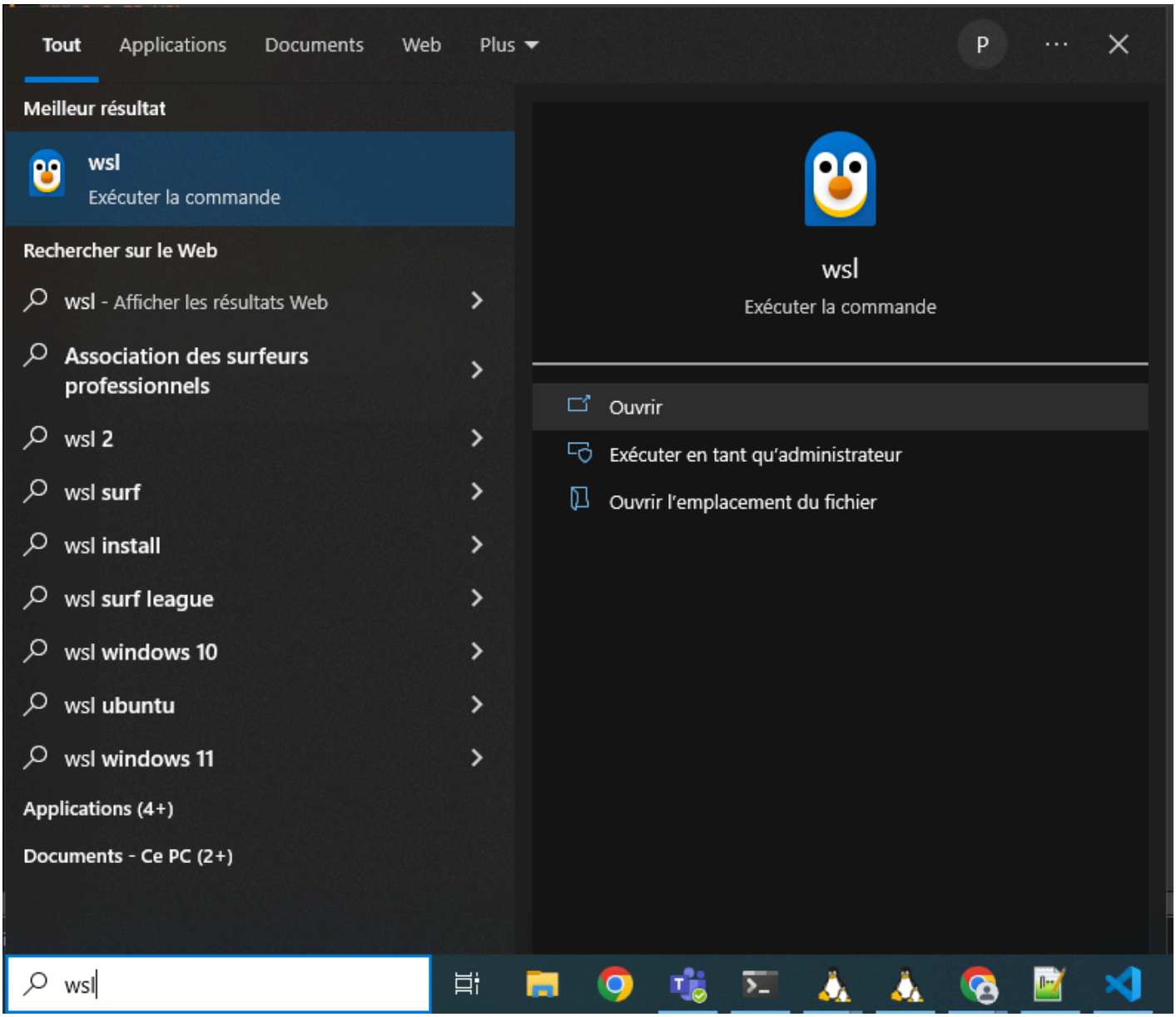
- Mise en œuvre du Backend Aero-Webb
- Mise en œuvre du Frontend Aero-Webb
- Connecter le VPN sur Windows (Mobile VPN)

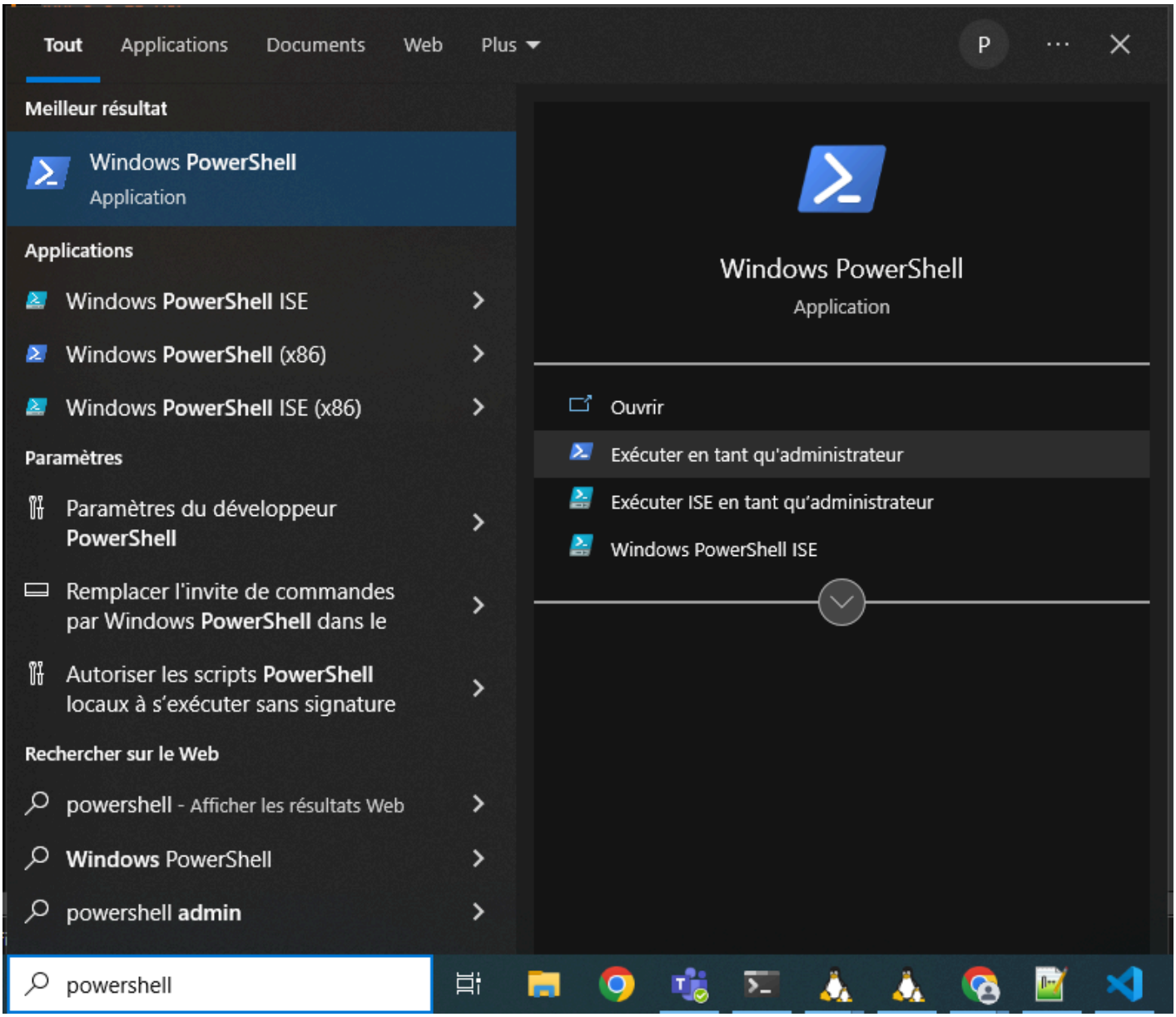
Introduction

Afin de rendre un environnement de developpement visible sur le réseau il faut :

- Récupérer les adresses IP de la machine Windows et de la machine WSL.
- Autoriser le pare-feu Windows à faire des redirections.
- Mettre en place la redirection.
- Configurer les serveur Backend et Frontend.

Au cours de cette installation il est nécéssaire d'ouvrir un terminal WSL ainsi qu'un terminal *PowerShell* en tant qu'administrateur.


Tout
Applications
Documents
Web
Plus
P
Meilleur résultat
wsl
Exécuter la commande
Rechercher sur le Web
wsl - Afficher les résultats Web
Association des surfeurs professionnels
wsl 2
wsl surf
wsl install
wsl surf league
wsl windows 10
wsl ubuntu
wsl windows 11
Applications (4+)
Documents - Ce PC (2+)
wsl
Exécuter la commande
Ouvrir
Exécuter en tant qu'administrateur
Ouvrir l'emplacement du fichier
wsl



Adresses IP

Cette section indique comment récupérer les différents IP nécéssaires à la mise en place de la redirection.

IP VPN

Dans le terminal *PowerShell* taper la commande suivante fin de lister les interfaces réseau de la machine Windows :

```
ipconfig
```

Voici un exemple de sortie

```
Configuration IP de Windows

Carte Ethernet Ethernet 2 :

Suffixe DNS propre à la connexion. . . : 2moro.lan
Adresse IPv6 de liaison locale. . . . .: fe80::6c4c:6cf4:d0ad:10b6%14
Adresse IPv4. . . . . . . . . . . . . .: 192.168.3.29
Masque de sous-réseau. . . . . . . . . : 255.255.255.0
Passerelle par défaut. . . . . . . . . :

Carte Ethernet vEthernet (Default Switch) :

Suffixe DNS propre à la connexion. . . :
Adresse IPv6 de liaison locale. . . . .: fe80::16d0:4250:ff26:c949%21
Adresse IPv4. . . . . . . . . . . . . .: 172.20.80.1
Masque de sous-réseau. . . . . . . . . : 255.255.240.0
Passerelle par défaut. . . . . . . . . :

Carte Ethernet vEthernet (WSL) :

Suffixe DNS propre à la connexion. . . :
Adresse IPv6 de liaison locale. . . . .: fe80::7999:cfeb:7cfb:ba2a%48
Adresse IPv4. . . . . . . . . . . . . .: 172.20.160.1
Masque de sous-réseau. . . . . . . . . : 255.255.240.0
Passerelle par défaut. . . . . . . . . :

Carte réseau sans fil Connexion au réseau local* 1:

Statut du média. . . . . . . . . . . . : Média déconnecté
Suffixe DNS propre à la connexion. . . :

Carte réseau sans fil Connexion au réseau local* 2:

Statut du média. . . . . . . . . . . . : Média déconnecté
Suffixe DNS propre à la connexion. . . :

Carte réseau sans fil Wi-Fi:

Suffixe DNS propre à la connexion. . . : dhcp.bida.fr.ssg
Adresse IPv6 de liaison locale. . . . .: fe80::7133:a9a0:6257:b62f%13
Adresse IPv4. . . . . . . . . . . . . .: 10.64.31.42
Masque de sous-réseau. . . . . . . . . : 255.255.254.0
Passerelle par défaut. . . . . . . . . : 10.64.30.1
```

L'IP de la machine Windows sur le VPN se trouve dans le bloc suivant (2moro.lan) :

```
Suffixe DNS propre à la connexion. . . : 2moro.lan
Adresse IPv6 de liaison locale. . . . .: fe80::6c4c:6cf4:d0ad:10b6%14
Adresse IPv4. . . . . . . . . . . . . .: 192.168.3.29
Masque de sous-réseau. . . . . . . . . : 255.255.255.0
Passerelle par défaut. . . . . . . . . :
```

L'IP de la machine Windows est dans cet exemple : 192.168.3.29

Important

Lorsque cette doc a été rédigée (Août 2023) le VPN pouvait fournir une adress IP dynamique, ce qui signifie qu'elle peut changer à chaque connexion. Si elle change il faudra verifier les configurations associées.

IP WSL

Dans le terminal WSL taper la commande suivante fin de lister les interfaces réseau de la machine WSL :

```
ip a
```

Voici un exemple de sortie

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
link/ether 00:15:5d:f7:3f:e3 brd ff:ff:ff:ff:ff:ff
inet 172.20.170.113/20 brd 172.20.175.255 scope global eth0
valid_lft forever preferred_lft forever
inet6 fe80::215:5dff:fef7:3fe3/64 scope link
valid_lft forever preferred_lft forever
```

L'IP de la machine WSL sur le sous-réseau de la machine Windows se trouve dans le bloc suivant à la ligne 'inet' (eth0) :

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
link/ether 00:15:5d:f7:3f:e3 brd ff:ff:ff:ff:ff:ff
inet 172.20.170.113/20 brd 172.20.175.255 scope global eth0
valid_lft forever preferred_lft forever
inet6 fe80::215:5dff:fef7:3fe3/64 scope link
valid_lft forever preferred_lft forever
```

L'IP de la machine WSL est dans cet exemple : 172.20.170.113

Notez que cette IP est statique est n'est pas sensible aux changements d'IP du VPN, en revanche elle change d'une machine à une autre, ainsi que d'une VM WSL à une autre (cas lorsqu'on a plusieurs VM WSL installées).

Autorisation de pare-feu

Cette section indique comment ajouter un ensemble de règles afin que le pare-feu Windows ne bloque pas les redirections.

Sur Aero-Webb les ports utilisé par ***Tomcat*** et ***Node.js*** sont respectivement **8080** et **4200**. Nous les retrouverons lors de la configuration des serveurs.

Nous devons créer une règle entrante et une règle sortante pour chaque port redirigés, soit un total de 4 règles. Nous utiliserons la commande suivante :

```
New-NetFireWallRule -DisplayName '<nom de la règle>' -Direction <Outbound|Inbound> -LocalPort "
<port>" -Action Allow -Protocol TCP
```

Voici les 4 règles à écrire dans PowerShell :

```
New-NetFireWallRule -DisplayName 'WSL 2 - AW Back - Out' -Direction Outbound -LocalPort "8080" -
Action Allow -Protocol TCP
New-NetFireWallRule -DisplayName 'WSL 2 - AW Back - In' -Direction Inbound -LocalPort "8080" -
Action Allow -Protocol TCP
New-NetFireWallRule -DisplayName 'WSL 2 - AW Front - Out' -Direction Outbound -LocalPort "4200" -
Action Allow -Protocol TCP
New-NetFireWallRule -DisplayName 'WSL 2 - AW Front - In' -Direction Inbound -LocalPort "4200" -
Action Allow -Protocol TCP
```

Port forwarding

Cette section indique comment mettre en place la redirection de port de la machine Windows vers la machine WSL.

Dans PowerShell nous utiliserons la commande suivante :

```
netsh interface portproxy add v4tov4 listenport=<port d'origine> listenaddress=<IP d'origine>
connectport=<port destination> connectaddress=<IP de destination>
```

L'IP d'origine sera **0.0.0.0** afin de rediriger les paquets qui viennent de toutes les IP (pas seulement une IP en particulier), l'IP de destination est l'IP de la machine WSL sur le sous-réseau de Windows (lue lors de l'étape 2.2).

Il y a 2 ports à rediriger, il faudra donc écrire 2 règles.

Dans cet exemple l'IP sera 172.20.170.113

```
netsh interface portproxy add v4tov4 listenport=4200 listenaddress=0.0.0.0 connectport=4200
connectaddress=172.20.170.113
netsh interface portproxy add v4tov4 listenport=8080 listenaddress=0.0.0.0 connectport=8080
connectaddress=172.20.170.113
```

Nous pouvons lister les règles de redirection avec la commande suivante :

```
netsh interface portproxy show all
```

Elle doit retourner le résultat suivant :

```
Écouter sur ipv4: Connecter à ipv4:

Adresse Port Adresse Port
----- ----- ----- -----
0.0.0.0 4200 172.20.170.113 4200
0.0.0.0 8080 172.20.170.113 8080
```

Cas particuliers

Tout les paquets sur ces ports sont redirigés, si un serveur sur la machine Windows essaie de se lancer sur ces ports cela ne marchera pas, il faudra donc soit changer de port pour le serveur sur Windows, soit retirer les redirections avec la commande suivante :

```
netsh interface portproxy reset
```

Configuration des serveurs

Cette section indique comment configurer les serveurs ***Tomcat*** et ***Node.js***.

Configuration Backend

Dans le fichier **server.xml**, situé dans le dossier *conf* de votre répertoire d'installation ***Tomcat***, il faut vérifier que le connecteur est bien configuré sur le port 8080 (port sur lequel la redirection du Backend est faite).

Le connecteur doit être le suivant :

```
<Connector port="8080"
   protocol="HTTP/1.1"
   connectionTimeout="20000"
   redirectPort="8443"
   maxParameterCount="1000"
/>
```

Dans le fichier **aerowebb.properties**, situé dans le dossier de configuration de l'appli Aero-Webb, il faut vérifier que la liste des CORS contienne bien les IP:PORT de la machine WSL et du VPN.

Dans notre exemple la ligne doit être la suivante (on retrouve bien l'IP de la machine WSL 172.20.170.113, VPN 192.168.3.29 et les ports 8080 et 4200) :

```
cors.allowed.origins=http://172.20.170.113:8080,http://172.20.170.113:4200,http://172.20.170.113,htt
```

Une fois ces fichiers modifiés il faut relancer le serveur Tomcat de la machine WSL avec la commande suivante (dans WSL):

Note

Attention à l'IP du VPN, si elle est dynamique elle peut changer à chaque connexion.

```
sudo systemctl restart tomcat
```

Configuration Frontend

> **Important**
>
> Ne pas commiter ces fichiers

Dans le fichier `environement.ts`, situé dans votre repo git `frontend_aerowebb/projects/aw-desktop/src/environments`, il faut configurer l'adresse du serveur Tomcat avec l'IP du VPN sur le port de redirection.

Dans notre exemple, comme ceci :

```
export const environment: AppEnvironment = {
    production: false,
    apiBaseUrl: '/aerowebb/remoting',
    serveurConfigFile: '',
    serveurUrl: 'http://192.168.3.29:8080'
};
```

Dans le fichier `angular.json`, situé dans votre repo git `frontend_aerowebb/projects/aw-desktop/src/environments`, il faut configurer l'adresse d'écoute du serveur Node.js avec l'IP de la machine WSL sur le port de redirection.

Dans notre exemple, comme ceci :

```
{
 "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
 "version": 1,
 "newProjectRoot": "projects",
 "projects": {
    "aw-desktop": {

        ...

        "architect": {

            ...

            "serve": {
                "builder": "@angular-devkit/build-angular:dev-serveur",
                "options": {
                    "browserTarget": "aw-desktop:build",
                    "port": 4200,
                    "host": "172.20.170.113"
                },

                    ...

                }

                ...

        }

        ...

    }
 }
}
```

Une fois ces fichiers modifiés il faut relancer le serveur ***Node.js*** de la machine WSL.