



# How to install aero-Webb on Ubuntu WSL

# How to install aero-Webb on Ubuntu WSL

Cette note décrit la mise en œuvre des outils et services nécessaires pour mettre en œuvre le backend Aero-Webb dans une distribution Ubuntu tournant sous WSL.

## Prerequisites

Mise en œuvre de [Ubuntu WSL](#)

## Introduction

Les services mis en œuvre dans ce document suivent les prérequis de version inscrits dans le document [DEVELOPER MANUAL, AERO-WEBB 7.x](#) enregistré sur SharePoint.

## Git

Cette section décrit l'installation de Git dans un environnement WSL Ubuntu.

**Avant** de procéder aux manipulations qui suivent, Git doit préalablement être installé sur le système hôte (Windows). Le nom d'utilisateur, son email et sa clé publique doivent être convenablement configurés. Ils seront communs aux deux systèmes.

## Installation

Vérifiez que Git est installé sur votre système à l'aide de la commande suivante :

```
git --version
```

Si le système vous indique que git n'a pas été trouvé, utilisez les commandes qui suivent pour l'installer :

```
sudo apt update  
sudo apt install git
```

## Apt ne marche pas

Pour toute installation de paquet dans WSL, désactivez le VPN 2MoRO, celui-ci bloque apt sans raison.

## Java OpenJDK 11 LTS

Cette section décrit l'installation de Java OpenJDK-11 LTS dans un environnement Ubuntu.

## Installation

Installer OpenJDK-11 :

```
sudo apt update
sudo apt install -y openjdk-11-jdk
```

Vérifier que tout est bien installé :

```
java -version
```

## Configuration

Créer un lien symbolique vers la version de java par défaut :

```
sudo ln -s /usr/lib/jvm/java-11-openjdk-amd64 /usr/lib/jvm/default-java
```

## Apache Tomcat 9

Cette section décrit l'installation d'Apache Tomcat 9 dans un environnement Ubuntu.

### Installation

Créez un nouvel utilisateur et un groupe en exécutant la commande suivante:

```
sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat
```

Cet utilisateur sera utilisé pour exécuter le service Tomcat.

Télécharger Tomcat 9 :

```
cd /tmp
wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.71/bin/apache-tomcat-9.0.71.tar.gz
```

Au moment de rédiger ces lignes, la dernière version de Tomcat 9 est la v9.0.71. Si la version change, le lien ci-dessus ne fonctionnera plus ; un lien fonctionnel peut alors être obtenu en explorant les fichiers de [cette URL](#). Il faudra également adapter le nom des dossiers à la nouvelle version de Tomcat.

#### Unable to establish SSLConnection

Add `--no-check-certificate` to your `wget`

Extraire les fichiers de l'archive :

```
tar -xvzf apache-tomcat-9.0.71.tar.gz
```

Déplacer le dossier vers sa destination :

```
sudo mv apache-tomcat-9.0.71 /opt/tomcat/
```

Créer un lien symbolique vers Tomcat :

```
sudo ln -s /opt/tomcat/apache-tomcat-9.0.71 /opt/tomcat/latest
```

Ce lien symbolique permet d'avoir plusieurs installations de Tomcat ; il est particulièrement pratique en cas de mise à jour, il suffira alors de décompresser la version plus récente et de modifier le lien symbolique pour qu'il pointe vers cette version.

Modifier le propriétaire du dossier afin que Tomcat puisse accéder aux fichiers :

```
sudo chown -R tomcat:tomcat /opt/tomcat
```

## Configuration

### Créer un fichier systemd

#### Prerequisites

Activer `systemd` sur WSL : cf. [§ 2.4 Activation de systemd](#)

Créer le fichier `/etc/systemd/system/tomcat.service` :

```
sudo nano /etc/systemd/system/tomcat.service
```

avec le contenu suivant :

```
[Unit]
Description=Tomcat 9.0 servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/default-java"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom"

Environment="CATALINA_BASE=/opt/tomcat/latest"
Environment="CATALINA_HOME=/opt/tomcat/latest"
Environment="CATALINA_PID=/opt/tomcat/latest/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"
EnvironmentFile=/home/<utilisateur>/config/tomcat/env

ExecStart=/opt/tomcat/latest/bin/catalina.sh $PARAM1 $PARAM2
ExecStop=/opt/tomcat/latest/bin/catalina.sh stop

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

 Remark

Dans le fichier ci-dessus et dans les commandes qui suivent, veiller à remplacer `<utilisateur>` par le nom de l'utilisateur WSL qui a été créé lors de la création de la VM Ubuntu.

Créer le répertoire `/home/<utilisateur>/.config/tomcat` :

```
mkdir -p /home/<utilisateur>/.config/tomcat
```

Puis créer et éditer le fichier `/home/<utilisateur>/.config/tomcat/env` :

```
nano /home/<utilisateur>/.config/tomcat/env
```

Et y mettre le contenu suivant :

```
PARAM1=jpda  
PARAM2=start
```

 Remark

Le premier paramètre permet d'activer le mode *debug* de Tomcat. C'est à dire que lors du démarrage du service, Tomcat lancera un thread supplémentaire qui écoutera sur le port 8000 (par défaut) afin de permettre le débogage d'une application web, comme Aero-Webb, à partir d'un EDI, comme Eclipse ou IntelliJ par exemple.

Pour désactiver le mode *debug* de Tomcat, il suffit de commenter la ligne en ajoutant le caractère `"#"` en tout début, puis redémarrer le service Tomcat.

Donner les permissions correctes au fichier du service :

```
sudo chmod 0644 /etc/systemd/system/tomcat.service
```

Informers systemd qu'un nouveau service existe :

```
sudo systemctl daemon-reload
```

Démarrer Tomcat et vérifier qu'il est bien lancé :

```
sudo systemctl start tomcat  
sudo systemctl status tomcat
```

Si Tomcat est bien démarré, activer le démarrage automatique du service :

```
sudo systemctl enable tomcat
```

En cas de besoin, il est possible d'utiliser `sudo systemctl start|stop|restart tomcat` pour démarrer, arrêter ou relancer Tomcat manuellement, sans avoir à désactiver le démarrage automatique.

## Configurer l'interface Web de Tomcat

Éditer le fichier `/opt/tomcat/latest/conf/tomcat-users.xml` comme suit :

```
<tomcat-users>
<!--
  Comments
-->
<role rolename="admin-gui"/>
<role rolename="manager-gui"/>
<user username="admin" password="admin_password" roles="admin-gui,manager-gui"/>
</tomcat-users>
```

Avec cette configuration, les identifiants d'accès à l'interface Web de Tomcat sont :

Donnée	Valeur
username	admin
password	admin_password

## Tester l'installation

Redémarrer Tomcat :


```
sudo systemctl restart tomcat
```

Ouvrez votre navigateur à l'adresse : [localhost:8080](http://localhost:8080)


Si tout est bon l'interface de Tomcat doit apparaitre :

Home Documentation Configuration Examples Wiki Mailing Lists
Find Help

## Apache Tomcat/9.0.68



**If you're seeing this, you've successfully installed Tomcat. Congratulations!**



**Recommended Reading:**

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Server Status

Manager App

Host Manager

**Developer Quick Start**

<a href="#">Tomcat Setup</a>	<a href="#">Realms &amp; AAA</a>	<a href="#">Examples</a>	<a href="#">Servlet Specifications</a>
<a href="#">First Web Application</a>	<a href="#">JDBC DataSources</a>		<a href="#">Tomcat Versions</a>

**Managing Tomcat**

For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users. [Read more...](#)

- [Release Notes](#)
- [Changelog](#)
- [Migration Guide](#)
- [Security Notices](#)

**Documentation**

- [Tomcat 9.0 Documentation](#)
- [Tomcat 9.0 Configuration](#)
- [Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

- [Tomcat 9.0 Bug Database](#)
- [Tomcat 9.0 JavaDocs](#)
- [Tomcat 9.0 Git Repository at GitHub](#)

**Getting Help**

**FAQ and Mailing Lists**

The following mailing lists are available:

**tomcat-announce**  
Important announcements, releases, security vulnerability notifications. (Low volume).

- [tomcat-users](#)  
User support and discussion
- [taglibs-user](#)  
User support and discussion for [Apache Taglibs](#)
- [tomcat-dev](#)  
Development mailing list, including commit messages

## Apache Maven 3.5

Cette section décrit l'installation d'Apache Maven 3.5 dans un environnement Ubuntu.

### Installation

Télécharger l'archive de Maven dans le dossier `/tmp` :

```
cd /tmp
wget https://archive.apache.org/dist/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.tar.gz
```

**Unable to establish SSLConnection**

Add `--no-check-certificate` to your `wget`

Le lien de téléchargement des versions de Maven peut s'obtenir à partir de [cette page](#).

How to install aero-Webb on Ubuntu WSL - Page 7 of 11

**i Info**

Si possible télécharger bien la version 3.5.4 même si d'autres plus récentes sont disponible

Décompresser le contenu de l'archive dans le dossier `/opt` :

```
sudo tar xvzf /tmp/apache-maven-3.5.4-bin.tar.gz -C /opt
```

Créer un lien symbolique vers Maven :

```
sudo ln -s /opt/apache-maven-3.5.4 /opt/maven
```

Ce lien symbolique permet d'avoir plusieurs installations de Maven ; il est particulièrement pratique en cas de mise à jour, il suffira alors de décompresser la version plus récente et de modifier le lien symbolique pour qu'il pointe vers cette version.

Créer le fichier `/etc/profile.d/maven.sh` comme suit ; ce fichier contient les variables d'environnement de Maven :

```
export JAVA_HOME=/usr/lib/jvm/default-java
export M2_HOME=/opt/maven
export MAVEN_HOME=/opt/maven
export PATH=${M2_HOME}/bin:${PATH}
```

Rendre le fichier de variables d'environnement Maven exécutable :

```
sudo chmod +x /etc/profile.d/maven.sh
```

Exécuter le fichier afin de charger les variables d'environnement de Maven :

```
sudo /etc/profile.d/maven.sh
```

Vérifier que l'installation est correcte en exécutant la commande suivante :

```
mvn -version
```

**💡 Tip**

Si après cette commande il vous est indiqué que maven n'est pas installé éteignez la machine Ubuntu et rallumez la

```
exit

wsl --shutdown Ubuntu
wsl -d Ubuntu
```

Puis vérifiez à nouveau la version de maven

## Configuration

L'installation de Maven ne nécessite pas de configuration particulière.

## PostgreSQL 12

Cette section décrit l'installation de PostgreSQL 12 dans un environnement Ubuntu.

### Installation

Importer la clé GPG utilisée pour signer les packages PostgreSQL :

```
curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc|sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/postgresql.gpg
```

Après avoir importé la clé GPG, ajouter le référentiel PostgreSQL :

```
echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" |sudo tee /etc/apt/sources.list.d/pgdg.list
```

Installer PostgreSQL sur le système :

```
sudo apt update
sudo apt -y install postgresql-12 postgresql-client-12
```

Démarrer PostgreSQL et vérifier qu'il est bien lancé :

```
sudo systemctl start postgresql
sudo systemctl status postgresql
```

Utiliser `sudo systemctl start|stop|restart|status postgresql` pour démarrer, arrêter ou relancer PostgreSQL.

### Configuration

Définir le mot de passe de l'utilisateur PostgreSQL créé par défaut lors de l'installation :

```
sudo su - postgres
psql -c "ALTER USER postgres WITH PASSWORD 'postgres'"
exit
```

Configurer la directive `listen_addresses` du fichier `/etc/postgresql/12/main/postgresql.conf` afin d'autoriser les connexions à travers le réseau :

```
# Listen on all interfaces
listen_addresses = '*'
```

Modifier le type d'authentification pour l'utilisateur postgres en éditant le fichier

`/etc/postgresql/12/main/pg_hba.conf` et en remplaçant la méthode `peer` par la méthode `md5` :

```
# Database administrative login by Unix domain socket
local all postgres md5
```

Redémarrer le service :

```
sudo systemctl restart postgresql
```

Configurer l'encodage par défaut des bases de données :

```
psql -U postgres -c "UPDATE pg_database SET datistemplate = FALSE WHERE datname = 'template1';"
psql -U postgres -c "DROP DATABASE template1;"
psql -U postgres -c "CREATE DATABASE template1 WITH TEMPLATE = template0 ENCODING = 'UNICODE';"
psql -U postgres -c "UPDATE pg_database SET datistemplate = TRUE WHERE datname = 'template1';"
psql -U postgres -d template1 -c "VACUUM FREEZE;"
```

### Mes commandes psql ne passent pas

our vos commandes psql je vous conseille de basculer direct sur le user postgres dans votre terminal, cela vous évitera d'avoir à taper votre mdp postgres à chaque commande (sudo su, su -l postgres).

### Paramètres de connexion à la base de données

Donnée	Valeur
Hôte	localhost
Port	5432
Utilisateur	postgres
Mot de passe	postgres

## NPM (Node Package Manager)


Cette section décrit l'installation de NPM et de Node.js dans un environnement Ubuntu, en utilisant NVM (Node Version Manager).

### Installation de Node Version Manager

#### Note

Dans les exemples qui suivent, choisir une version des outils compatible avec la version d'Aero-Webb utilisée (cf. [la matrice de compatibilité](#)).

Installer Node Version Manager (nvm) :


 **Note**

De préférence, installer la version la plus récente de l'outil `nvm` (cf. [releases](#)).

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
source ~/.bashrc
```

Listez les versions de Node.js disponibles à l'installation (la liste est conséquente) :

```
nvm list-remote
```

 **list-remote → N/A**

Pour `nvm` dans WSL, ajoutez un fichier `.curlrc` dans votre répertoire `/home//` contenant uniquement `-k` (il y a un souci avec les db de Node, cela vous permettra de le contourner).

Ou

Connecter votre ordinateur sur un autre wifi (ex : Partage de connection avec votre téléphone)

Installer une version de Node.js spécifique :

```
nvm install <version>
```

Vérifier que Node.js et NPM fonctionnent :

```
node -v
npm -v
```

Listez les versions de Node.js installées via `nvm`

Il est possible de lister les versions de Node.js déjà installées sur le système à l'aide de la commande suivante :

```
nvm list
```

La première ligne retournée par cette commande correspond à la version actuellement active de Node.js.

## Configuration

### Node.js

L'installation de Node.js ne nécessite aucune configuration particulière.

### NPM

Éditez ou créez le fichier de configuration `/home/<utilisateur>/.npmrc`, afin que son contenu soit le suivant :

```
registry=http://nexus-ext.2moro.fr:8081/repository/npm-all/
```