

GUIDE DES PRÉREQUIS

TP Middleware API Gateway et Microservices

Préparation de l'environnement Docker et Git avant la séance

Pour suivre la simulation en temps réel, sans temps morts

Préparé par :

**Mouad BENCAID
Raissouni Abdellah**

Année universitaire : 2025/2026

Pourquoi ce guide ?

Ce document est **un guide de préparation** à utiliser **avant** la séance de Travaux Pratiques sur les middlewares et l'API Gateway.

Son objectif est simple : faire en sorte que, le jour de la démonstration, **tout le monde puisse suivre la simulation en temps réel**, sans perdre de temps à télécharger des images Docker ou à installer des dépendances.

Objectifs de ce guide

Ce guide vous permet de :

- ✓ Préparer un environnement technique stable pour pouvoir **observer concrètement le rôle des middlewares** dans une architecture microservices ;
- ✓ Vérifier que votre environnement **supporte Docker Desktop et Git** pour simuler des **topologies de middleware** proches d'un contexte d'entreprise engagée ;
- ✓ **Installer et tester Docker** (daemon en marche, commandes de base) afin de pouvoir démarrer et arrêter facilement les différentes topologies ;
- ✓ **Cloner les deux repositories** utilisés pendant le TP : architecture **sans** middleware d'entrée et architecture **avec** middleware (API Gateway) ;

Ce que ce guide n'est pas

Ce document **ne remplace pas** :

- le contenu théorique de l'atelier sur les middlewares ;
- le support TP principal (« Guide de Travaux Pratiques ») que vous utiliserez pendant la séance ;
- les slides de présentation.

Il est uniquement destiné à **préparer votre machine** pour éviter les problèmes techniques de dernière minute.

Résultat attendu avant de venir en TP

À l'issue de ce guide, sur votre poste :

- Docker Desktop doit être installé, lancé et testé ;
- Git doit être installé et opérationnel ;
- les deux projets doivent être clonés en local ;
- les images Docker des microservices doivent être **déjà construites** pour les deux architectures.

Ainsi, pendant la séance, vous n'aurez plus qu'à **lancer les conteneurs** à partir d'images déjà prêtes.

1 Préparation de l'environnement

1.1 Prérequis matériels et système

- Connexion Internet **temporaire** pour télécharger Docker, Git et les repositories (non nécessaire le jour du TP si tout est déjà fait);
- Système compatible Docker Desktop (Windows 10/11 64 bits ou macOS, ou bien une distribution GNU/Linux avec Docker Engine);
- Droits administrateur sur la machine (installation de logiciels).

1.2 Étape 1 – Installation et vérification de Docker

1.1) Rendez-vous sur le site officiel :

— Windows / macOS : <https://www.docker.com/products/docker-desktop/>

1.2) Installez Docker Desktop en suivant l'assistant d'installation (sur Windows, activez WSL2 si nécessaire).

1.3) Redémarrez votre machine si l'installateur le demande, puis lancez Docker Desktop et attendez que le daemon soit « Up and running ».

1.4) Ouvrez un terminal (PowerShell, CMD ou terminal Linux/macOS) et tapez :

```
1 docker --version  
2 docker compose version # ou: docker-compose --version
```

Listing 1 – Vérification de l'installation Docker

1.5) Vérifiez qu'aucune erreur de type « command not found » n'apparaît.

En cas de problème Docker

Si Docker ne démarre pas ou si les commandes ci-dessus échouent, résolvez le problème avant de passer à la suite (mise à jour de Windows, activation de la virtualisation, réinstallation de Docker Desktop, etc.).

1.3 Étape 2 – Vérification de Git

2.1 Dans le même terminal, vérifiez la présence de Git :

```
1 git --version
```

Listing 2 – Vérification de Git

2.2 Si la commande échoue, installez Git depuis : <https://git-scm.com/downloads>, puis recommencez la vérification.

1.4 Étape 3 – Clonage des deux repositories

Choisissez ou créez un dossier de travail (par exemple : C:\middleware-lab ou /middleware-lab). Puis exécutez les commandes suivantes :

```
1 # Se placer dans votre dossier de travail
2 cd chemin/vers/votre/dossier
3
4 # 1) Architecture microservices SANS API Gateway
5 git clone https://github.com/mouadd9/demo1-microservices-without-a-gateway.git
6
7 # 2) Architecture microservices AVEC API Gateway
8 git clone https://github.com/mouadd9/demo2-microservices-with-a-gateway.git
```

Listing 3 – Clonage des projets du TP

Pour consultation directe du code source et de la documentation en ligne, vous pouvez aussi accéder aux dépôts GitHub :

- <https://github.com/mouadd9/demo1-microservices-without-a-gateway>
- <https://github.com/mouadd9/demo2-microservices-with-a-gateway>

Vous devez obtenir deux dossiers :

- demo1-microservices-without-a-gateway
- demo2-microservices-with-a-gateway

Pour être cohérent avec les chemins utilisés dans le **guide TP principal**, vous pouvez **renommer manuellement** ces dossiers (clic droit → Renommer) en :

- microservices-without-a-gateway
- microservices-with-a-gateway

À ce stade

Les sources des deux architectures sont téléchargées en local sur votre machine et prêtes à être utilisées par Docker Compose.

1.5 Étape 4 – Construction des images Docker (sans Gateway)

Dans cette étape, nous construisons toutes les images nécessaires pour **l'architecture sans middleware d'entrée (sans API Gateway)**. Cette première version sert de point de comparaison pour illustrer les **limites d'un système distribué sans middleware**, avant d'introduire une topologie régulée.

4.1 Placez-vous dans le dossier du premier projet (en adaptant le chemin en fonction de votre organisation) :

```
1 cd microservices-without-a-gateway
```

4.2 Construisez les images :

```
1 # Utilisez l'une des deux syntaxes selon votre version
2 Docker compose build
3 # ou
4 Docker-compose build
```

Listing 4 – Construction des images pour l'architecture sans Gateway

4.3 Patientez jusqu'à la fin de la construction (les premières fois, Docker télécharge les couches de base et installe les dépendances).

4.4 Vérifiez la présence des images dans votre cache Docker :

```
1 docker images
```

Astuce

Si vous avez le temps, vous pouvez lancer une fois l'architecture complète pour vérifier que tout démarre correctement :

```
1 Docker compose up
2 # ou: docker-compose up
```

Puis arrêtez et nettoyez :

```
1 Docker compose down
2 # ou: docker-compose down
```

Le jour du TP, le redémarrage sera alors quasi immédiat, car toutes les images sont déjà construites.

1.6 Étape 5 – Construction des images Docker (avec Gateway)

Répétez la même logique pour **l'architecture avec middleware d'entrée (API Gateway)**. Dans le cadre de cet atelier, la Gateway est utilisée comme **exemple concret de middleware** permettant de réguler une architecture microservices. Un **autre groupe** proposera un atelier entièrement dédié à l'API Gateway elle-même (patterns avancés, configuration détaillée, etc.).

5.1 Revenez au dossier parent puis placez-vous dans le deuxième projet :

```
1 cd ..
2 cd microservices-with-a-gateway
```

5.2 Construisez les images correspondantes :

```
1 Docker compose build  
2 # ou: docker-compose build
```

Listing 5 – Construction des images pour l'architecture avec Gateway

5.3 Optionnel : testez un démarrage complet, puis arrêtez :

```
1 Docker compose up  
2 Docker compose down
```

Bilan de la préparation

Si toutes les étapes précédentes sont complétées :

- Docker Desktop fonctionne et peut exécuter des conteneurs ;
- Git est installé et les deux repositories sont clonés ;
- les images Docker des deux architectures (sans middleware d'entrée et avec middleware d'entrée) sont déjà construites ;
- le jour de la séance, vous n'aurez plus qu'à exécuter les commandes de lancement indiquées dans le **guide TP principal** (« docker-compose up »), sans attendre le téléchargement ni la construction des images, et vous pourrez vous concentrer sur la **comparaison des topologies de middleware** (API Gateway n'étant qu'un exemple parmi d'autres).

Vous êtes donc prêt(e) à suivre la **simulation en temps réel** dans des conditions optimales.