

# TP atelier Big DATA

## Partie 1 : installation des composant

### **Docker**

*Linux* : <https://docs.docker.com/desktop/install/linux-install/>

*Windows* : <https://docs.docker.com/desktop/install/windows-install/>

*mac* : <https://docs.docker.com/desktop/install/mac-install/>

### **Conduktor**

*all os* : <https://www.conduktor.io/get-started/#desktop>

### **Spark**

spark: <https://cedric.cnam.fr/vertigo/Cours/RCP216/installationSpark.html#installationspark>

*attention pre-requis : JDK ou openJDK 1,8 (8)*

*Faites les installations mais les versions de spark ayant évoluer faites en sorte de manipuler le spark que vous avez téléchargé.*

### ***Python avec miniconda :***

*Personnellement j'utilise Python avec Miniconda mais faites comme vous le sentez*

*all os* : <https://docs.conda.io/projects/miniconda/en/latest/miniconda-install.html>

### ***suite jetbrain :***

*avec l'école vous avez le droits a Pycharm et IntelliJ de la suite Jetbrain :*

*Faites un compte etudiant avec le mail de l'école qui vous offrira un compte gratuit :*

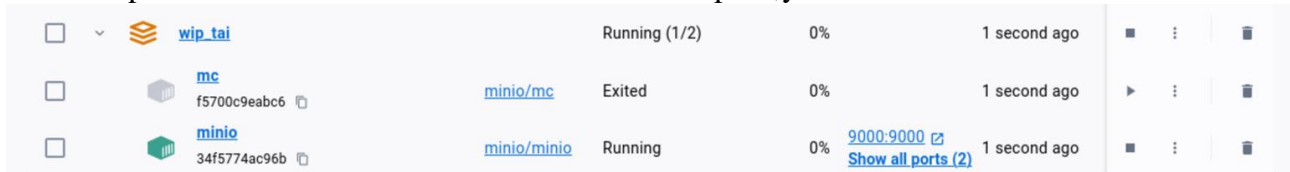
<https://www.jetbrains.com/>

## Partie 2 : Mise en place

### **docker :**

Lancer la commande : `docker compose up -d (linux)`  
ou `docker-compose up -d (windows)`

dans le repertoire ou sont stocké les fichier docker-compose.yml



The screenshot shows the Docker Desktop interface with three containers listed:

Container Name	Image	Status	Restart Policy	Uptime	Ports
wip_tai	wip_tai	Running (1/2)	0%	1 second ago	
mc	minio/mc	Exited	0%	1 second ago	
minio	minio/minio	Running	0%	1 second ago	9000:9000, Show all ports (2)

Vous verrez ce type d'instance ce Mettre en route sur le docker engine

(petit tips : installer git bash sur windows pour pouvoir manipuler comme du linux )

### **Code :**

Aller sur GitHub et faite un git clone pour recuperer le projet\*

Vous avez le choix de programmer en SCALA  
ou en PYTHON au choix.

## Partie 3 : consigne

### Partie 1 :

si python finir le producteur pour envoyer la donnée dans kafka  
Si scala , consumer déjà tout prêt

### Partie 2 :

Recuperer les donnée via spark streaming ou kafka-stream  
stocker cette donner en dataframe

puis faire des agregats dessus :

```
# convertir USD en EUR
# ajouter le TimeZone
# remplacer la date en string en une valeur date
# supprimer les transaction en erreur
# supprimer les valeur en None ( Adresse )
```

Partie 3 :

ecrire le nouveaux DATAFRAME au format parquet sur Minio

Partie 4 option

Lire le fichier nouvellement cree sur Minio avec spark