## Google Cloud

Créer votre espace Google Cloud en utilisant votre compte google.

Une version gratuite existe pour 3 mois avec 300€ de crédit. Elle ne permet pas la création des VM Windows, uniquement des versions Linux avec la plupart des fonctionnalités.

Suivre cet article pour créer une VM Debian et configurer le bureau accès à distance chrome

<https://medium.com/@selvamraju007/install-and-launch-ubuntu-22-04-desktop-on-google-cloud-1fba8c0f9585>

# Suivre ce document pour l’installation de remote desktop et un environnement X window (graphique)

<https://cloud.google.com/architecture/chrome-desktop-remote-on-compute-engine?hl=fr>

# Install and Launch Ubuntu 22.04 Desktop on Google Cloud

# [Selvam Raju](https://medium.com/@selvamraju007?source=post_page-----1fba8c0f9585--------------------------------)

In this blog, we are going to learn how to install Desktop (GUI) and Launch on an Ubuntu 22.04 LTS.

Usually GUI has been disabled by default on all Linux machines, so if you need desktop for some reasons this blog will help you.

We need below four steps to setup VM with graphical user interface mode.

1. Create Ubuntu instance in Google Cloud
2. Install and configure Chrome remote desktop service on the VM instance
3. Install Ubuntu desktop packages on the VM instance
4. Connect from your Chrome web browser to the desktop environment on the VM instance

**Create Ubuntu VM instance:**

We will launch VM instance in google cloud. The default E2-medium (2vCPU and 4GB memory)

1. In the <https://console.cloud.google.com/> go to the VM instance under the Compute Engine.
2. Click create instance for the VM creation.
3. Set the instance name, region and zone.

4. Scroll down to **boot disk** and click **change.**

5. In the**boot disk** you should select Operation system, select ubuntu from the drop down window and select version ubuntu 22.04 LTS and keep the rest options as default.

6. Click to create the instance.

7. Instance will create 2–3 mins and come up with running state.

# Install Chrome Remote Desktop on the VM instance

Cette installation permettra l’accès distant par interface graphique.

The next step is to install Chrome Remote Desktop on the VM instance which you created. Download and install the Debian Linux Chrome Remote Desktop installation package:

wget <https://dl.google.com/linux/direct/chrome-remote-desktop_current_amd64.deb>

Vérifier les droits sur le fichier \*.deb est r-xr-xr-x sinon

chmod 555 chrome-remote-desktop\_current\_amd64.deb

sudo apt-get install --assume-yes ./chrome-remote-desktop\_current\_amd64.deb

Si cette commande ne fonctionne pas essayer de récupérer les dépendances par :

sudo apt-get -f install

**Setup Ubuntu desktop environment in the VM instance:**

1. Connect your instance using SSH, refresh the repository and package lists, perform the necessary upgrade using the below commands.

sudo apt update && sudo apt upgrade

2. Install and setup display manager. Here i used SLiM for its lightweight feature

sudo apt install slim

3. Install Ubuntu desktop environment (the installation process may need around 15-20 minutes):

sudo apt install ubuntu-desktop

4. Once finishes the installation, reboot the machine.

sudo reboot

5. Once instance is back to online after the reboot, ssh the connection and start the slim using below command.

sudo service slim start

# Configure the Chrome Remote Desktop service and Connect to your Ubuntu Desktop

To start the remote desktop connection, you need to have an authorization key for your Google account.

1. On your local computer, using the Chrome browser, go to the Chrome Remote Desktop command line setup page:  
<https://remotedesktop.google.com/headless>

2. On the Set up another computer page, click Begin.

3. Click **next.**You already installed the Chrome Remote Desktop on the remote computer in STEP2.

4. Click **Authorise**

5. Now you get the command to set up and start the Chrome Remote Desktop service on your VM instance. Copy the command for Debian Linux.

6. Copy and Paste the command to the SSH window that connects to your VM instance. Run the command.

7. Enter a 6-digit PIN when prompted. This PIN will be used when you log into the VM instance from your Chrome.

PIN=123456 🡺 Ce code sera demande par Chrome Remote Desktop pour se connecter à la VM

8. On your local computer, go to the [Chrome Remote Desktop](https://remotedesktop.google.com/) web site. You will find your Ubuntu Desktop shows up in the portal. Le type de la connection est SSH !

9. Click the remote desktop instance. In this case, it is “worker-node1”. And you will be prompted to input the 6-digit PIN you set in step 7.

10. Now you can see the ubuntu desktop and fully functional.

Hope this helps to everyone who wants to setup desktop (GUI). Happy learning!

# Installer Oracle Express

**Attention**: avec denedo Express 8 la base oracle pris en charge est Oracle Express 11g

pm2cpio oracle-database-xe-21c-1.0-1.ol8.x86\_64.rpm | cpio –idmv

Oracle est installé dans le repertoire: ..../opt/oracle/….

Suivre ce lien pour l’installation :

<https://www.baeldung.com/linux/debian-oracle-database-express-edition-xe>

<https://docs.oracle.com/en/database/oracle/oracle-database/21/xeinl/installing-oracle-database-free.html#GUID-46EA860A-AAC4-453F-8EEE-42CC55A4FAD5>

**Installation de Oracle-xe-client**

<https://doc.ubuntu-fr.org/oracle>

**Démarrer le service Oracle**

systemctl start oracle-xe-21c.service

**Comptes Oracle**

SYS, SYSTEM, PDBADMIN 🡺 password : Lauboudou

Le container ou la base des données pour se connecter

**XEPDB1**

**Remarque**

Après un reboot de l’instance VM Gougle Cloud, la connection à Oracle est perdue.

**Vérifier si le listener est démarré**

lsnrctl start listener

**Vérifier si systemctl est démarré**

systemctl start oracle-xe-21c.service

Si le démarrage des deux ne fait pas fonctionner Oracle, reconfigurer le service oracle

1. Remove la configuration

/etc/init.d/oracle-xe-21c remove

1. Reconfigurer le service, puis assurer que la base est démarrée

/etc/init.d/oracle-xe-21c configure

1. Voici les traces de la fin de configuration,

………

………

Connect to Oracle Database using one of the connect strings:

Pluggable database: instance-20240303-003315/**XEPDB1**

Multitenant container database: instance-20240303-003315

Use **https://localhost:5500/em** to access Oracle Enterprise Manager for Oracle Database XE

Unexpected error creating a connection: IO Error: The Network Adapter could not establish the connection Received exception with message 'IO Error: The Network Adapter could not establish the connection' Received exception with message 'The Network Adapter could not establish the connection' Received exception with message 'data-server: Temporary failure in name resolution'

# Désinstaller ORACLE

Consulter le site:

<https://askubuntu.com/questions/740203/how-to-uninstall-oracle-database-from-ubuntu-14-04>

$ sudo su

Enter the following:

**Arrêter le service oracle**

# /etc/init.d/oracle-xe stop

**Supprimer le service oracle**

# /etc/init.d/oracle-xe delete

**Nettoyer l’installation**

# dpkg --purge oracle-xe

**Supprimer le répertoire oracle**

#rm -r /opt/oracle

**Exécuter les updates suivants**

# update-rc.d -f oracle-xe remove

# update-rc.d -f oracle-mount remove

# update-rc.d -f oracle-shm remove

# Installer Oracle 11g

Télécharger la version 11g

**Préparer l’environnement pour Oracle**

apt-get install libaio1 unixodbc

Unzip le package

**Passer le package en \*\*\*.deb**

alien --scripts --to-deb oracle-xe-11.2.0-1.0.x86\_64.rpm

**Exécuter le package deb**

dpkg --install oracle-xe-11g\*\*\*\*\*.deb

**Exécuter la configuration d’Oracle**

/etc/init.d/oracle-xe configure

**Renseigner les informations suivantes :**

Specify the HTTP port that will be used for Oracle Application Express [8080]:8085 (different recommandé pour denodo\_training)

Specify a port that will be used for the database listener [1521]:1521

Mot de passé pour SYS et SYSTEM : Lauboudou

**Configurer les paramètres de l’environnement d’exécution**

ORACLE\_SID=XE;

export ORACLE\_SID

ORACLE\_HOME=/u01/app/oracle/product/11.2.0/xe/

export ORACLE\_HOME

LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib:$LD\_LIBRARY\_PATH;

export LD\_LIBRARY\_PATH

PATH=$ORACLE\_HOME/bin:$LD\_LIBRARY\_PATH:$PATH;

export PATH

NLS\_LANG=AMERICAN\_AMERICA.AL32UTF8;

export NLS\_LANG

**Démarrer le service Oracle**

systemctl start oracle-xe

# Installer MYSQL Server

Télécharger Mysql sur le site official

Suivre cette page

<https://ubuntu.com/server/docs/databases-mysql>

To install MySQL, run the following command from a terminal prompt:

sudo apt mysql-server

Once the installation is complete, the MySQL server should be started automatically. You can quickly check its current status via systemd:

sudo service mysql status

Which should provide an output like the following:

● mysql.service - MySQL Community Server

Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)

Active: active (running) since Tue 2019-10-08 14:37:38 PDT; 2 weeks 5 days ago

Main PID: 2028 (mysqld)

Tasks: 28 (limit: 4915)

CGroup: /system.slice/mysql.service

└─2028 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

Oct 08 14:37:36 db.example.org systemd[1]: Starting MySQL Community Server...

Oct 08 14:37:38 db.example.org systemd[1]: Started MySQL Community Server.

The network status of the MySQL service can also be checked by running the ss command at the terminal prompt:

sudo ss -tap | grep mysql

When you run this command, you should see something similar to the following:

LISTEN 0 151 127.0.0.1:mysql 0.0.0.0:\* users:(("mysqld",pid=149190,fd=29))

LISTEN 0 70 \*:33060 \*:\* users:(("mysqld",pid=149190,fd=32))

If the server is not running correctly, you can type the following command to start it:

sudo service mysql restart

A good starting point for troubleshooting problems is the systemd journal, which can be accessed from the terminal prompt with this command:

sudo journalctl -u mysql

Configuration

You can edit the files in /etc/mysql/ to configure the basic settings – log file, port number, etc. For example, to configure MySQL to listen for connections from network hosts, in the file /etc/mysql/mysql.conf.d/mysqld.cnf, change the bind-address directive to the server’s IP address:

bind-address = 192.168.0.5

Note:

Replace 192.168.0.5 with the appropriate address, which can be determined via the ip address show command.

After making a configuration change, the MySQL daemon will need to be restarted with the following command:

sudo systemctl restart mysql.service

**Installe mysql-workbench-community**

Télécharger le package du site

A partir du répertoire de téléchargement, enlever l’extension \*\*\*\*.deb

dpkg –i package\*\*\*\*\*\*.deb

Lancer la commande

Apt get install mysql-workbench-community\*\*\*\*\*\*\*

Lancer le workbench

**Si vous rencontrer un compte root@localhost non accessible, taper les commandes suivantes**

<https://phoenixnap.com/kb/access-denied-for-user-root-localhost>

sudo mysql

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY **'lauboudou'**;

mysql> Query ok, 0 rows affected

mysql> FLUSH PRIVILEGES;

mysql> Query ok, 0 rows affected

mysql> exit

Tapper cette commande et se connecter avec le nouveau mot de passe,

mysql -u root -p

**Donner des privileges à root**

mysql -u root

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost' WITH GRANT OPTION;

FLUSH PRIVILEGES;

exit

**Utiliser l’adresse IP de localhost**

mysql -u root -p -h 127.0.0.1

# Installer Denodo Express

unzip denodo-express-install-8\_0-linux64.zip

Denodo est installé dans le répertoire: **/home/dlaubo/denodo-platform-8.0**

Se connecter avec dlaubo pour assurer le lancement de ./**install.sh** sous **X11 Display** (le mode graphique)

Lancer ;/installer.sh

# Data Virtualisation Basics Tutoriel

Suivre ce lien,

<https://community.denodo.com/tutorials/browse/basics/index#3>

Fixer Java sur ubuntu

Installer Java

Les installations seront dans le répertoire Downloads

L’installation se trouve dans le répertoire :

/usr/lib/jvm/\*

Fixer le PATH sur ubuntu dans le fichier **~/.bashrc**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

export JAVA\_HOME=**/usr/lib/jvm/jdk-11-oracle-x64**

export ORACLE\_HOME=**/u01/app/oracle/product/11.2.0/xe**

export ORACLE\_SID=**XE**

export ORACLE\_BASE=**/u01/app/oracle**

export LD\_LIBRARY\_PATH=**$ORACLE\_HOME/lib:$LD\_LIBRARY\_PATH**

export PATH**=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin**

export PATH=**$ORACLE\_HOME/bin:$JAVA\_HOME/bin:$PATH**

export NLS\_LANG=**AMERICAN.US7ASCII**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Lancement du service pour le tutoriel 🡺 Go to /home/dlaubo/Downloads/jetty and run:

 java -jar start.jar

# Ajouter un Adapter MYSQL à Denodo

Choisir une librairie avec une extension jar

**Important**

Toute clé primaire trouvée dans la table source est marquée dans la vue de base de Denodo.

**Note:**Any **primary key** found in the source table gets marked in the Denodo base view. This is important because this information will be exposed to Denodo clients (for example, reporting tools).

**Web Services**

**Note:**The Web service needs a mandatory input parameter and Denodo is automatically taking care of providing this parameter in the join view using the data coming from the **personal\_data\_crm** view.

Dans la jointure, on peut tester la nouvelle vue sans passer le parameter. Denodo prendra en charge le paramètre pendant l’exécution de la requête.

## ****Aggregation Operation****

Faite sur une vue permet de préparer une colonne avec Group By pour représenter les résultats sous format d’une ligne avec la somme de billings dus par le client

## **Se connecter à Denodo via une application externe par JDBC**

Installer **DbVisualizer**  sur le lien <https://www.dbvis.com/>

Exécuter le script ./dbvis\_linux\_24\_1\_3.sh

Suivre les instructions, DbVisualizer sera installé dans :

**/home/dlaubo/DbVisualizer**

Lancer la commande avec le user dlaubo, si la fenêtre n’est pas ouverte automatiquement

**./dbvisgui.sh**

Si DISPLAY n’est pas initialisé

Fixer le avec : **set DISPLAY= :20.0 🡺 (chercher comment le fixer)**

**La différence entre Bases JDBC et ODBC**

**Installer ODBC**

Les tutos :

<https://community.denodo.com/tutorials/browse/basics/index#6>

<https://community.denodo.com/docs/html/browse/8.0/en/vdp/developer/access_through_odbc/configuration_of_the_odbc_driver_in_linux_and_other_unix/configuration_of_the_odbc_driver_in_linux_and_other_unix>

Aller dans

**/home/dlaubo/denodo-platform-8.0/tools/client-drivers/odbc**

Exécuter la commande

tar -xf denodo-vdp-odbcdriver-linux.tar.gz

Le répertoire suivant sera créé

**denodo-vdp-odbcdriver-linux**

Aller dans ce répertoire pour installer unixodbc

**/home/dlaubo/denodo-platform-8.0/tools/client-drivers/odbc/denodo-vdp-odbcdriver-linux/lib**

ls –la

drwxrwxr-x 6 dlaubo dlaubo 4096 Apr 26 2023 .

drwxrwxr-x 5 dlaubo dlaubo 4096 Mar 19 17:49 ..

drwxrwxr-x 2 dlaubo dlaubo 4096 Apr 26 2023 iodbc\_x64

drwxrwxr-x 2 dlaubo dlaubo 4096 Apr 26 2023 iodbc\_x86

drwxrwxr-x 2 dlaubo dlaubo 4096 Apr 26 2023 unixodbc\_x64

drwxrwxr-x 2 dlaubo dlaubo 4096 Apr 26 2023 unixodbc\_x86

Vérifier si unixodbc est installé

**which isql**

/usr/bin/isql

**which odbcinst**

rien

sudo apt-get install unixodbc

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

unixodbc is already the newest version (2.3.12-1).

0 upgraded, 0 newly installed, 0 to remove and 39 not upgraded.

Execute this to verify that unixODBC was installed correctly:

odbcinst -j

Command 'odbcinst' not found, but can be installed with:

**apt install odbcinst**

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following NEW packages will be installed:

odbcinst

0 upgraded, 1 newly installed, 0 to remove and 39 not upgraded.

Need to get 9720 B of archives.

After this operation, 43.0 kB of additional disk space will be used.

Get:1 http://us-west1.gce.archive.ubuntu.com/ubuntu mantic/main amd64 odbcinst amd64 2.3.12-1 [9720 B]

Fetched 9720 B in 0s (104 kB/s)

Selecting previously unselected package odbcinst.

(Reading database ... 220225 files and directories currently installed.)

Preparing to unpack .../odbcinst\_2.3.12-1\_amd64.deb ...

Unpacking odbcinst (2.3.12-1) ...

Setting up odbcinst (2.3.12-1) ...

Processing triggers for man-db (2.11.2-3) ...

Scanning processes...

Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

**Enable ODBC Data Source on Denodo Virtual DataPort**

Aller dans Administration / Server Configuration / ODBC Data Sources

Cliquer sur On puis OK

## Register the Denodo ODBC Driver in UnixODBC

After installing unixODBC, register the ODBC driver. Follow these steps:

Create a file denodoODBCDriver.template with this content:

[DenodoODBCDriver]

Description=ODBC driver of Denodo

Driver=/home/dlaubo/denodo-platform-8.0/tools/client-drivers/odbc/denodo-vdp-odbcdriver-linux/lib/unixodbc\_x86/denodoodbc.so

UsageCount=1

1. Modify line #3 so it points to the flavor of the ODBC driver you want to use (i.e. unixodbc\_x86, unixodbc\_x64,…).
2. Execute the following command to register the Denodo driver in the ODBC Driver Manager:
3. sudo odbcinst -install -driver -file denodoODBCDriver.template

To list the ODBC drivers registered in the driver manager, execute this:

sudo odbcinst -query -driver

The result should list the new driver: DenodoODBCDriver.

To uninstall the driver, execute:

sudo odbcinst -uninstall -driver -name DenodoODBCDriver

# Annexes

Installer Chrome Desktop

<https://cloud.google.com/architecture/chrome-desktop-remote-on-compute-engine?hl=fr>

Configuration KDE

<https://phoenixnap.com/kb/how-to-install-a-gui-on-ubuntu>

**Ajout Chrome Remote Desktop à Debian**

sudo DEBIAN\_FRONTEND=noninteractive apt-get install --assume-yes ./chrome-remote-desktop\_current\_amd64.deb

Installer docker desktop sur ubuntu

<https://docs.docker.com/desktop/install/ubuntu/>

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04>

Installer docker step by step

<https://shape.host/resources/installing-docker-on-ubuntu-23-10-step-by-step-guide>

Install VMware sur ubuntu

<https://phoenixnap.com/kb/install-vmware-workstation-ubuntu>

AWS Cloud VM

L’inscription est faite sur ce site : business.whizlabs.com 🡺 3 mois gratuits

Suite à la journée de Cloud 2024 / Inscription via Yacoubou

[LMS | Whizlabs](https://business.whizlabs.com/learn/my-training/)

L’url d’accès

<https://business.whizlabs.com/learn/my-training/>

Les credentials

[dlaubo@gmail.com](mailto:dlaubo@gmail.com) / S\*\*\*\*\*\*3

La VM1 temporaire :

Nom sandbox : Whiz\_User\_213680.58948949

Password : 5b16e72f-8bfb-4cd0-ad0b-850d4a56701d

Access key : AKIAT5AY7H7ANCWQGSOC

Secret key : sLyvSnuMlLaoG9zexslBTqP3HKD3RNcRUr2qQzSC

Créer une instance VM-Debian-Docker

Nom de la paire de clé : Pair-Cle-VM-Linux-Docker

Aller sur AWS, dans les instances créer une VM

Vérifier dans la liste des services AWS autorisés de votre compte business.whizlabs.com

VM Windows est autorisée. Accès RDP et password déchiffrer depuis la pare de clé.

**Journée Cloud 28/06/2024**

Aliou BA, entrepreneur / Cloudoor

**09/07/2024**

L’offre whizlabs 3 mois gratuit sandbox amazone (AWS)

<https://business.whizlabs.com/login>

AWS-VM-Windows-080724

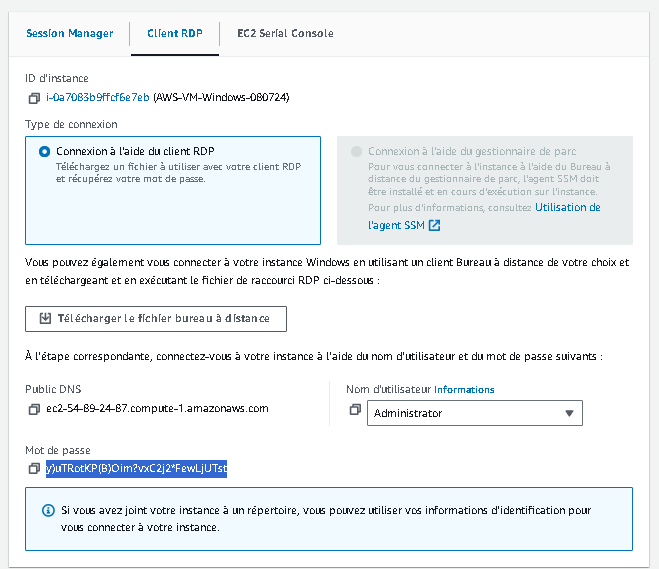
Pour accéder à cette VM, cliquer sur se connecter et lancer la machine et attendre que le credential s’affiche.

Utiliser les username et passeword fournis pour se connecter

Revenir sur la page <https://us-east-1.console.aws.amazon.com/>



Cliquer sur Obtenir le mot de passe



Configurer l’accès local en SSH à la VM Ubuntu

**Configurer le Groupe de Sécurité**

Les groupes de sécurité d'AWS agissent comme des pare-feu virtuels qui contrôlent le trafic entrant et sortant de vos instances EC2.

**a. Accéder au Groupe de Sécurité**

1. Connectez-vous à votre console AWS.
2. Accédez à la section **EC2** dans le panneau de services.
3. Dans le menu de gauche, cliquez sur **Security Groups**.

**b. Modifier les Règles de Sécurité**

1. Sélectionnez le groupe de sécurité associé à votre instance Ubuntu.
2. Cliquez sur l'onglet **Inbound rules** (Règles entrantes) et cliquez sur **Edit inbound rules** (Modifier les règles entrantes).
3. Ajoutez une règle pour le type de trafic que vous souhaitez autoriser. Voici quelques exemples courants :
   * **SSH (port 22)** : Pour l'accès SSH à la VM.
     + Type : SSH
     + Protocol : TCP
     + Port Range : 22
     + Source : Spécifiez l'adresse IP de votre machine locale ou mettez 0.0.0.0/0 pour autoriser l'accès depuis n'importe où (ceci est moins sécurisé).
   * **HTTP (port 80)** : Pour l'accès à un serveur web.
     + Type : HTTP
     + Protocol : TCP
     + Port Range : 80
     + Source : 0.0.0.0/0
   * **HTTPS (port 443)** : Pour l'accès sécurisé à un serveur web.
     + Type : HTTPS
     + Protocol : TCP
     + Port Range : 443
     + Source : 0.0.0.0/0
4. Cliquez sur **Save rules** pour enregistrer vos modifications.

**2. Vérifier l'Adresse IP Publique de l'Instance**

Votre instance doit avoir une adresse IP publique pour être accessible depuis l'extérieur.

**a. Vérifier l'IP Publique**

1. Toujours dans la console EC2, cliquez sur **Instances** dans le menu de gauche.
2. Sélectionnez votre instance Ubuntu.
3. Dans la section **Description** en bas, cherchez l'**IPv4 Public IP**.

### Installer docker sur windows server

Pour installer Docker sur Windows Server, vous pouvez suivre ces étapes. Docker peut être installé en tant que Docker Desktop (pour les environnements de développement) ou Docker Enterprise Edition (pour les environnements de production). Voici les instructions pour installer Docker sur Windows Server.

### Prérequis

* Windows Server 2016 ou plus récent.
* Rôle Hyper-V installé.
* Accès administrateur à la machine.

### Étapes d'installation

#### 1. **Installer Hyper-V et les fonctionnalités conteneurs**

Avant d'installer Docker, vous devez installer Hyper-V et les fonctionnalités de conteneurs :

1. Ouvrez PowerShell en tant qu'administrateur.
2. Exécutez les commandes suivantes :

Install-WindowsFeature -Name containers -IncludeAllSubFeature -Restart

Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart

Ces commandes installeront les fonctionnalités nécessaires et redémarreront votre serveur.

#### 2. **Installer Docker**

Après avoir installé Hyper-V et les fonctionnalités de conteneurs, vous pouvez installer Docker :

1. **Télécharger Docker** :

Téléchargez Docker pour Windows Server depuis Docker Hub.

1. **Installer Docker** :

Ouvrez PowerShell en tant qu'administrateur et exécutez les commandes suivantes :

Invoke-WebRequest -UseBasicParsing -OutFile docker-20.10.6.zip https://download.docker.com/win/static/stable/x86\_64/docker-20.10.6.zip

Expand-Archive -DestinationPath $Env:ProgramFiles -Path docker-20.10.6.zip

[Environment]::SetEnvironmentVariable("Path", $Env:Path + ";$Env:ProgramFiles\Docker", [EnvironmentVariableTarget]::Machine)

1. **Démarrer Docker** :

Toujours dans PowerShell, exécutez la commande suivante pour démarrer Docker :

dockerd --run-service

Cette commande démarre le démon Docker en tant que service.

1. **Vérifier l'installation** :

Pour vérifier que Docker a été installé correctement, exécutez :

docker --version

Vous devriez voir la version de Docker installée.

#### 3. **Configurer Docker pour démarrer au démarrage**

Pour s'assurer que Docker démarre automatiquement au démarrage du serveur, créez un service :

1. **Créer un fichier de service** :

Créez un fichier nommé DockerService.ps1 avec le contenu suivant :

New-Service -Name Docker -BinaryPathName "$Env:ProgramFiles\Docker\dockerd.exe --run-service" -DisplayName "Docker Engine" -Description "Docker Daemon" -StartupType Automatic

Si le service est mal installé, vous pouvez le supprimer avec les commandes suivantes :

$serviceName = "Docker"

sc.exe delete $serviceName

1. **Exécuter le script de service** :

Ouvrez PowerShell en tant qu'administrateur et exécutez le script :

powershell

Copier le code

.\DockerService.ps1

### Utilisation de Docker

Maintenant que Docker est installé, vous pouvez commencer à l'utiliser pour gérer des conteneurs. Voici quelques commandes de base :

* **Télécharger une image Docker** :

docker pull hello-world

* **Exécuter un conteneur Docker** :

docker run hello-world

* **Lister les conteneurs Docker en cours d'exécution** :

docker ps

* **Lister toutes les images Docker téléchargées** :

docker images

Avec ces étapes, vous devriez être capable d'installer et de configurer Docker sur Windows Server, et de commencer à utiliser Docker pour gérer vos conteneurs.

**Installer Docker sur Ubuntu**

# Update package list

sudo apt-get update

# Install prerequisites

sudo apt-get install -y ca-certificates curl gnupg lsb-release

# Add Docker’s official GPG key

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

# Set up the stable repository

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Update package list again

sudo apt-get update

# Install Docker Engine

sudo apt-get install -y docker-ce docker-ce-cli containerd.io

# Verify installation

sudo docker run hello-world

# Optional: Manage Docker as a non-root user

sudo usermod -aG docker $USER

newgrp docker

**Gérer les volumes en local**

[Utiliser les volumes dans le docker-compose.yml (nicelydev.com)](https://www.nicelydev.com/docker/volume-docker-compose-yml)

Installer Git

[Git for Windows](https://gitforwindows.org/)

Sdu9.goziamass.io

Une VM de stage

Installer tuleap

Récupérer la dernière image sur docker hub

Créer un dossier /home/dlaubo/projet/services/tuleap

Préparer un fichier docker-compose.yml

**Dans le fichier .env**

# L'URL d'accès à Tuleap

TULEAP\_FQDN=tuleap.vogoumpaga.org

# Nom de la base des données de l'application Tuleap

TULEAP\_SYS\_DBPASSWD=tuleap

# Nom du Tuleap Host

TULEAP\_SYS\_DBHOST=db

# Le port d'accès à Tuleap

VIRTUAL\_PORT=8081

# Mot de passe de l'utilisateur admin de l'application Tuleap

SITE\_ADMINISTRATOR\_PASSWORD=tuleapadmin

# Le mot de passe de l'administrateur de la base de données

DB\_ADMIN\_PASSWORD=tuleapdb

# Utilisateur de l'utilisateur root

DB\_ADMIN\_USER=root

# Mot de passe de l'utilisateur root

MYSQL\_ROOT\_PASSWORD=tuleapdb

# Nom du conteneur base des données

PROTEUS\_DB\_CONTAINER\_NAME=tuleap-db

# Le nom du volume utilisé par le service Tuleap

PROTEUS\_VOL\_APP\_NAME=tuleap\_tuleap-data

# Le nom du volume utilisé par le conteneur base des données du service Tuleap

PROTEUS\_VOL\_DB\_NAME=tuleap\_db-data

# Le nom des conteneurs de dépendances

PROTEUS\_DEPENDS\_ON=db,db2

# Le nom du hostname du conteneur

PROTEUS\_HOST\_NAME=

**Configurer le fichier .env**

Eviter des lignes vides dans le fichier .env

**Exporter les variables :**

vi export\_env.sh

copier/coller

#!/bin/bash

# Vérifier si le fichier .env existe

if [ -f .env ]; then

# Lire chaque ligne du fichier .env

while IFS= read -r line || [ -n "$line" ]; do

# Ignorer les lignes vides et les commentaires

if [ ! -z "$line" ] && [ "${line:0:1}" != "#" ]; then

# Exporter la variable d'environnement

export "$line"

fi

done < .env

else

echo ".env file not found"

fi

chmod +x export\_env.sh

source ./export\_env.sh

**Vérifier l’export**

printenv MYSQL\_ROOT\_PASSWORD

echo $MYSQL\_ROOT\_PASSWORD

Lancer la commande : **docker-compose.yml up -d**

version: '3.8'

services:

tuleap:

container\_name: tuleap

image: tuleap/tuleap-community-edition

hostname: ${TULEAP\_FQDN}

restart: always

ports:

- "8081:80" # Map container port 80 to host port 8081 (HTTP)

- "4443:443" # Map container port 443 to host port 4443 (HTTPS)

- "2222:22" # Map container port 22 to host port 2222 (SSH)

volumes:

- tuleap-data:/data

depends\_on:

- db

environment:

- TULEAP\_FQDN=${TULEAP\_FQDN}

- TULEAP\_SYS\_DBHOST=db

- TULEAP\_SYS\_DBPASSWD=${TULEAP\_SYS\_DBPASSWD}

- SITE\_ADMINISTRATOR\_PASSWORD=${SITE\_ADMINISTRATOR\_PASSWORD}

- DB\_ADMIN\_USER=root

- DB\_ADMIN\_PASSWORD=${MYSQL\_ROOT\_PASSWORD}

- VIRTUAL\_HOST=${TULEAP\_FQDN}

- VIRTUAL\_PORT=8081

db:

image: mysql:8.0

command: ["--character-set-server=utf8mb4", "--collation-server=utf8mb4\_unicode\_ci", "--sql-mode=NO\_ENGINE\_SUBSTITUTION"]

environment:

- MYSQL\_ROOT\_PASSWORD=${MYSQL\_ROOT\_PASSWORD}

- MYSQL\_DATABASE=tuleapdb

- MYSQL\_USER=tuleap

- MYSQL\_PASSWORD=tuleap

volumes:

- db-data:/var/lib/mysql

volumes:

tuleap-data:

db-data:

**Vérifier les conteneurs déployés**

docker logs tuleap

docker logs tuleap\_db\_1

**verifier l’accès de tuleap à mysql**

docker exec -it tuleap ping tuleap\_db\_1

Si changement dans les paramètres d’environnement (Fichier .env)

Par exemple : le nom du domain 🡺 sdu9.gozaimass.io à tuleap.vogoumpaga.org

Docker-compose down

Docker-compose up -d

Les commandes au-dessus ne suffiront pas pour modifier les volumes montés par le déploiement des conteneurs.

Pour vérifier les volumes

docker volume ls

Exécuter la commande suivante pour obtenir un shell root et accéder à la liste des volumes

sudo -i

cd /data/docker/volumes/

ls

**Forcer la récréation des volumes**

docker-compose up -d --force-recreate -V

**Supprimer les volumes**

docker-compose down -v

**Compte tuleap**

Admin/tuleapadmin

Dlaubo/gozaimass

**Lundi 15/07/2024**

[Tutoriel installation de Nginx Proxy Manager (sparwan.com)](https://sparwan.com/tutoriel-installation-de-nginx-proxy-manager.htm)

**Le compte admin**

Email address : [**admin@example.com**](mailto:admin@example.com)

Password : **changeme**

NPM vous les fera changer dès la première connexion.

Ynx : un browser pour voir à l’intérieur dans une application 🡺 conteneur wordpress

Selenium : tester automatiquement les pages

**Mardi 16/07/2024**

Point discord 🡺

Tuleap

Compte tuleap dlaubo/gozaimass

GitLab

dlaubo/S\*\*\*\*\*\*3

Les projets sur GitLab devrait avoir la structure suivante :

Projet / **tuleap**

<https://www.youtube.com/watch?v=WZQmm7P68ac>

deploiement

Readme.md

Le dossier deploiement contient les fichiers suivants,

.env

docker-compose.yml

start.sh

stop.sh

L’administrateur donne le droit de faire push sur le repository en passant par les settings et unlock sur la branche.

Puis les commandes suivantes fonctionneront sur repository serveur – gitlib

Sur le serveur sdu9.gozaimass.io

ssh-keygen

laisser options par défaut

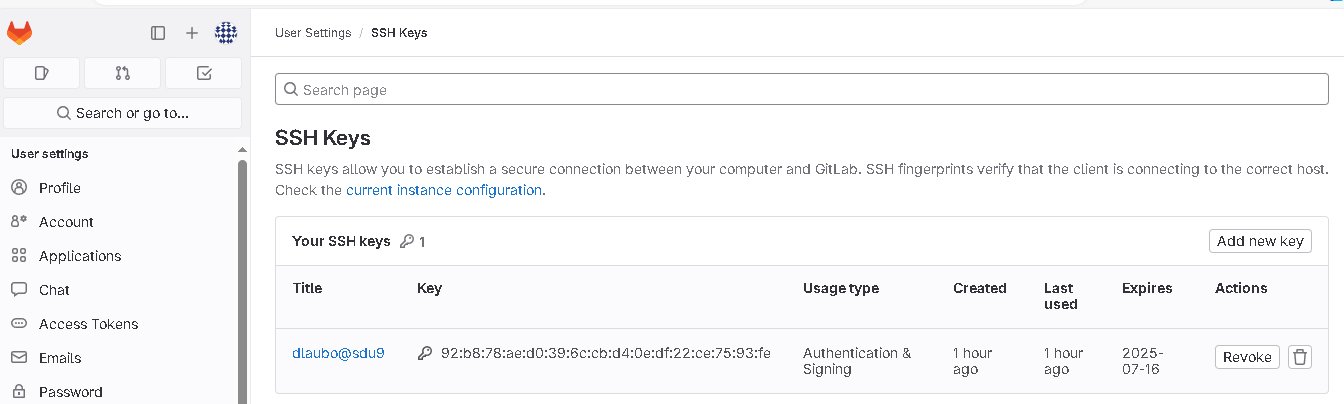
dans /home/dlaubo/.ssh

clés : id\_rsa et id\_ras.pub

copier la clé pub dans GitLab

Aller dans Références / SSH Keys

Ajouter la clé



cd **tuleap**

git remote add origin

https://gitlab.gozaimass.io/proteus/services/contributions/**tuleap.git**

[git@gitlab.gozaimass.io:proteus/services/contributions/tuleap.git](mailto:git@gitlab.gozaimass.io:proteus/services/contributions/tuleap.git)

git branch -M main

git push -uf origin main

Pour ignorer la vérification ssh

ssh -o StrictHostKeyChecking=no" git clone [git@github.com:votre\_utilisateur/votre\_repository.git](mailto:git@github.com:votre_utilisateur/votre_repository.git)

**Les submodules**

[Git - Sous-modules (git-scm.com)](https://git-scm.com/book/fr/v2/Utilitaires-Git-Sous-modules)

git clone git@gitlab.gozaimass.io:dlaubo/test-submodules.git

git clone git@gitlab.gozaimass.io:dlaubo/submodules.git

ls -la

**Les dossiers suivants sont créés**

test-submodules

submodules

cd test-submodules/

la commande suivante clone le reporistory submodules dans submodules (qui peut être un nom différent)

git submodule add git@gitlab.gozaimass.io:dlaubo/submodules.git submodules

**le fichier .gitmodules et le dossier submodules seront crées**

ls -la

cat .gitmodules

[submodule "submodules"]

path = submodules

url = git@gitlab.gozaimass.io:dlaubo/submodules.git

git status

[submodule "submodules"]

path = submodules

url = git@gitlab.gozaimass.io:dlaubo/submodules.git

dlaubo@sd10:~/test-submodules$ git status

On branch main

Your branch is up to date with 'origin/main'.

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: .gitmodules

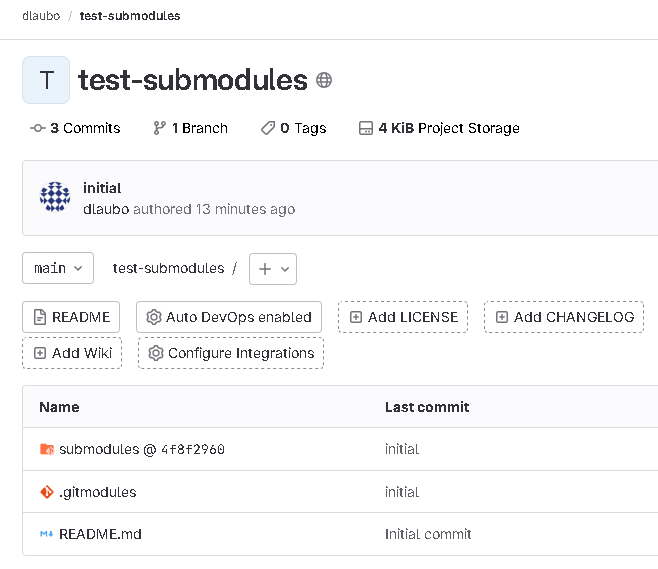
new file: submodules

git add .

git commit -m "initial"

git push

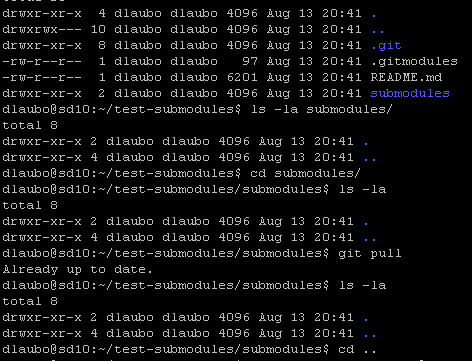
**Vérifier le repository test-submodules sur gitlab**



**Cloner un repository avec des submodules**

git clone git@gitlab.gozaimass.io:dlaubo/test-submodules.git

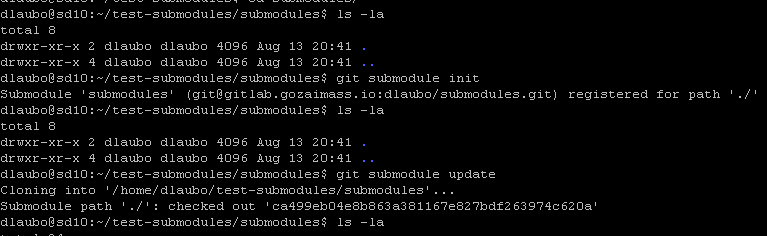
Après le clone, le projet test-submodules est créé avec ses submodules mais vides



Mettez-vous dans le répertoire submodules et exécuter les 2 commandes suivantes

git submodule init

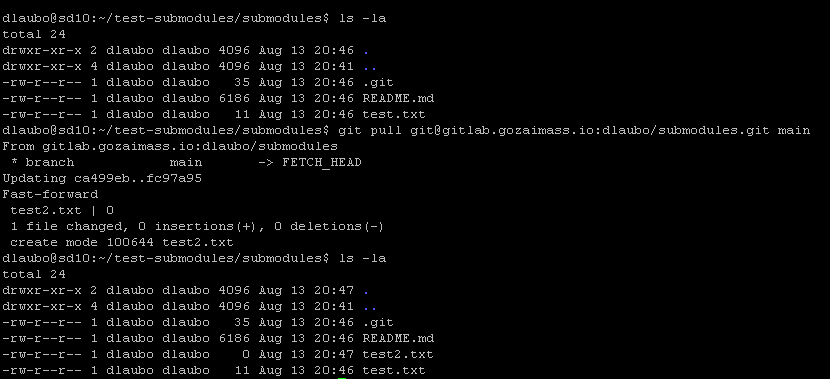
git submodule update



Le contenu du submodule connu par test-submodules sera récupérer

Pour récupérer et se synchroniser avec le repository du submodule exécuter la commande git pull suivante

git pull git@gitlab.gozaimass.io:dlaubo/submodules.git main



Faire cette commande à l’intérieur du projet git contributions/prestashop

git submodule add git@gitlab.gozaimass.io:proteus/services/deploiements/prestashop.git deploiement

Cela créer un répertoire deploiement qui point sur le projet git deploiements/prestashop

Note :

[submodule "deploiement"]

url = git@gitlab.gozaimass.io:proteus/services/deploiements/prestashop.git

active = true

**Modification labels Tuleap**

New User 🡺 Nouvel Utilisateur

Advanced User 🡺 Utilisateur Avancé

Acceptance Criteria 🡺 Critères d’acceptation

Attachements 🡺 Pièces Attachées

Progress 🡺 Progression

Initial Effort 🡺 Effot Initial

In order to 🡺 Afin

As a 🡺 En tant que

I want to 🡺 Je souhaite pouvoir

Comment tuleap associe Jalon

Release 🡺 Jalons 🡺

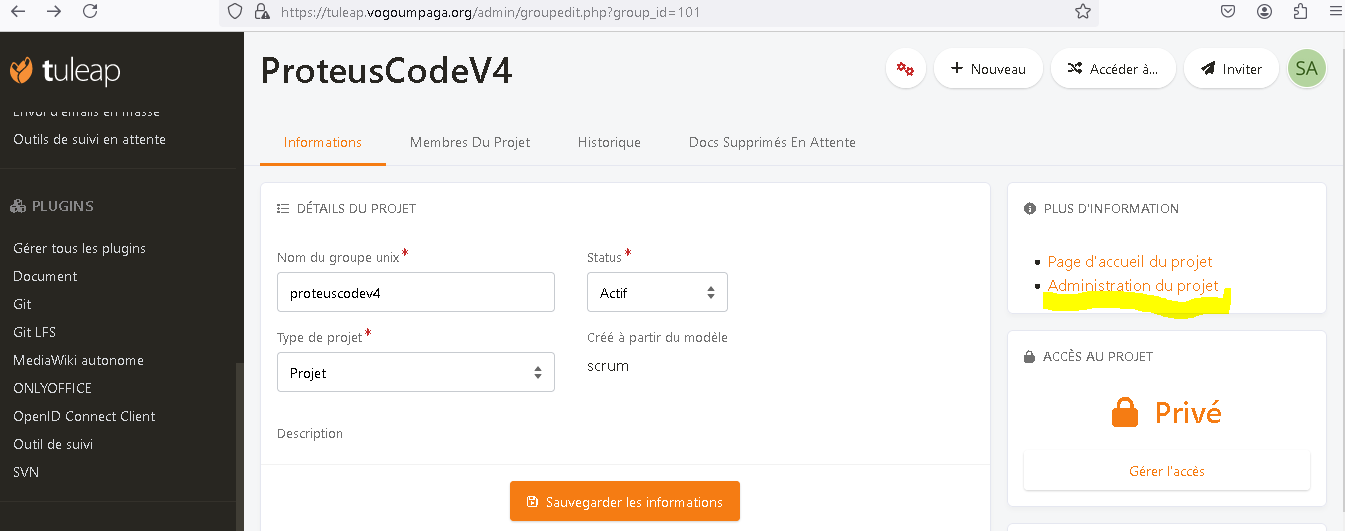
Jalon 🡸> Milestones

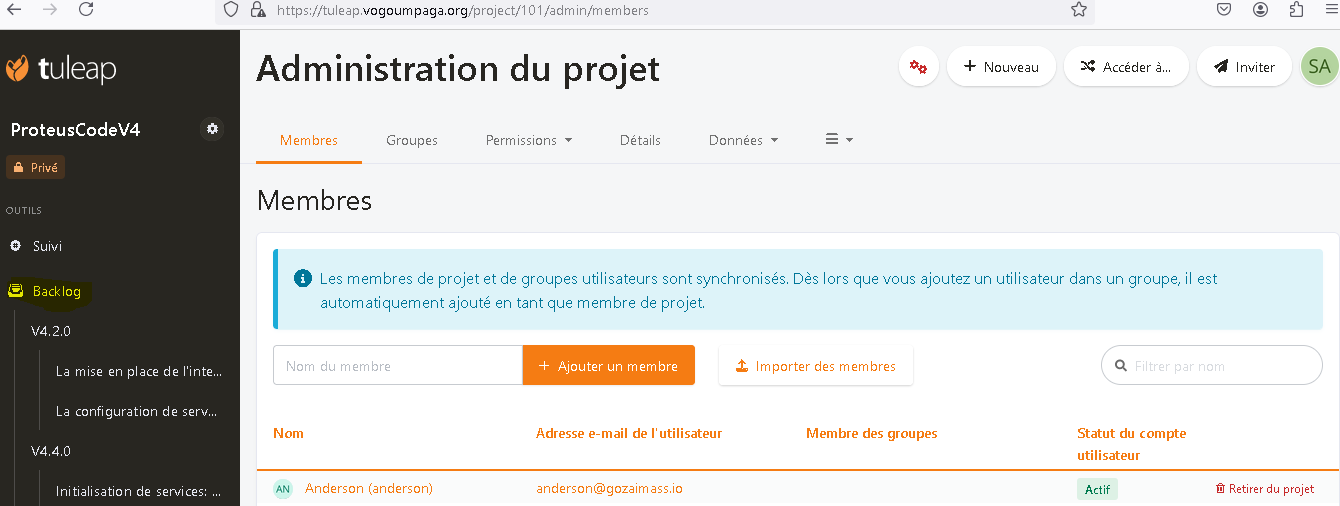
Changement des libellés dans Tuleap

Se connecter en tant qu’admin / tuleap

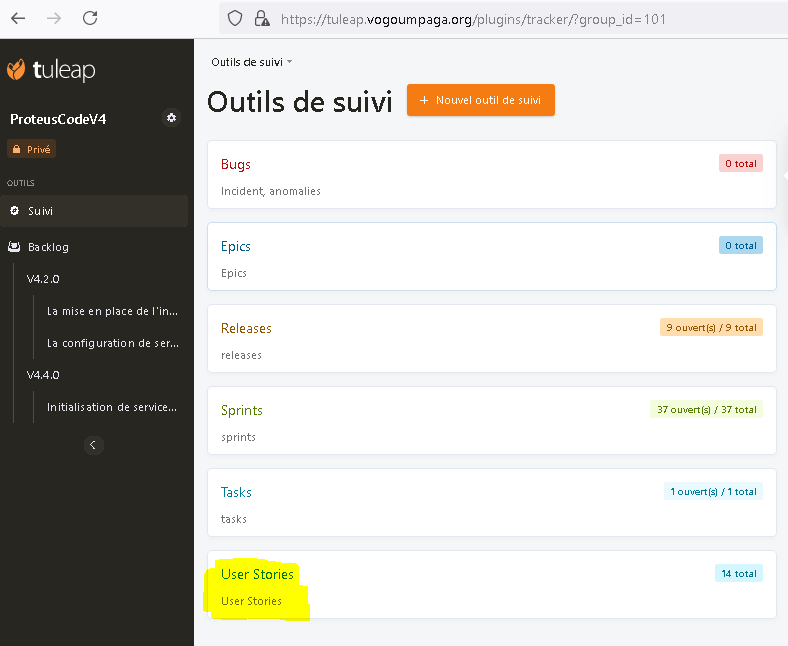
Aller dans le projet ProteusCodeV4

Aller dans la fénêtre Administration du projet

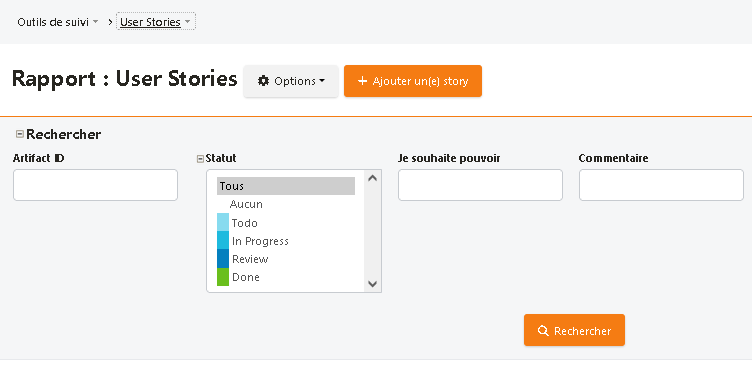




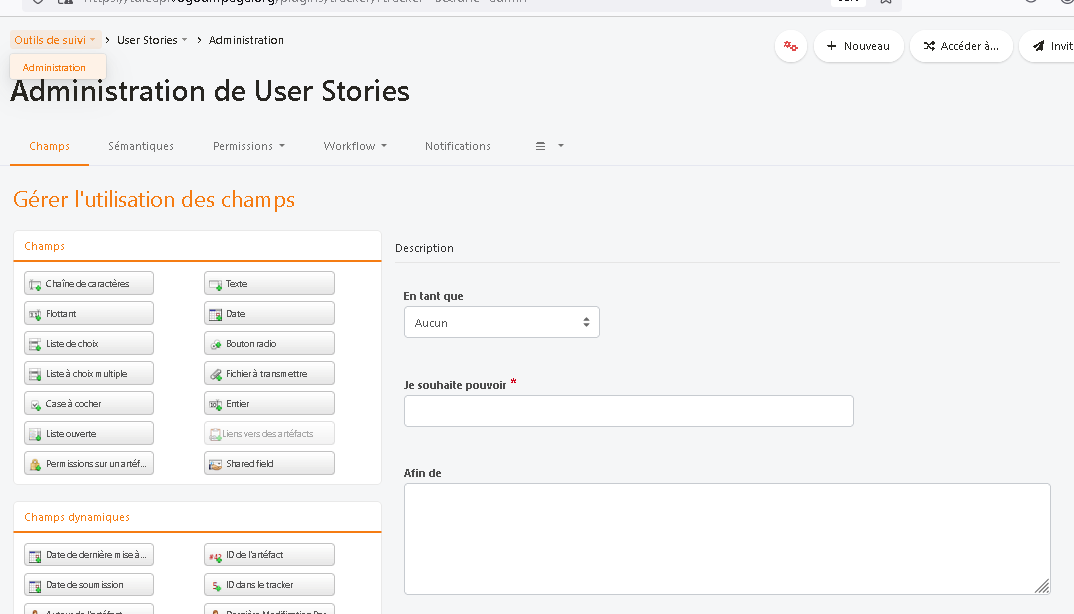
Aller sur User Stories



En haut de la page, Aller dans Outils de suivi > User Stories développer la flèche > Administration



La page Administration User Stories s’affiche



NectCloud : outil collaboratif

Sdu9.gozaimass.io:8080

[S’identifier – Nextcloud (gozaimass.io)](http://sdu9.gozaimass.io:8080/login)

Découpage releases

**Objectif de service:**

F1 - Importer une image (1 minute)

F2 - Ajouter un nouveau service

F3 - Déployer\* très rapidement un ou plusieurs services (1 minute)

F4 - Acceder à un service

F5 - Mise à jour d’un service

F6 - Sauvegarder\* données métiers d’un ou plusieurs services

F7 - Restaurer\* les données métiers d’un ou plusieurs services

F8 - Migrer\* un ou plusieurs services d’une plateforme vers une autre

F9 - Démarrer\* un ou plusieurs services

F10 - Arrêter\* un ou plusieurs services

F11 - Désinstaller\* un ou plusieurs services

F12 - Ouvrir un ticket

Vendredi 19/07/2024

Créer une user story DevSecOps pour sécuriser, supervisé et monitoré

Sprint : installation et configuration de l’environnement

**Tâches :**

Installation nginx, routage, proxy pour chaque service

Installation registery

Installation onlyoffice

installation de portenair : interface graphique de gestion de contenaires

Installation et paramètrage tuleap : ldia

\*\*\*\*\*\* nextcloud

\*\*\*\*\*\* onlyoffice

installation lemonldap et lurl potail.vogoumpaga.org

installer kubernetes et son dasboard

installer terraform

La migration des volumes docker : création de script change-docker

Docker-compose, volume, Changement de nom de domain

planification des tâches et le roadmap sur tuleap : ldia

paramètrage lemonldap

paramètrage registry avec le certificat SSL

configurer portenair

**Les scripts user\_data**

/var/lib/cloud/instance/scripts/part-001

openoffice

lemonldap

Automatisation des pipelines

### **Etapes pour réinitialiser le mot de passe root de MySQL dans un conteneur Docker**

1. **Arrêtez le conteneur MySQL** :

Arrêtez votre conteneur MySQL en cours d'exécution :

docker stop <mysql\_container\_id>

1. **Démarrez le conteneur MySQL en mode sans échec** :

Démarrez le conteneur MySQL en mode sans échec pour contourner la vérification du mot de passe :

docker run --name tuleap-db-test -e MYSQL\_ALLOW\_EMPTY\_PASSWORD=yes -d mysql:8.0 --skip-grant-tables

docker start tuleap-db-test

Remplacez mysql:8.0 par la version appropriée de votre image MySQL.

1. **Accédez au conteneur MySQL** :

Accédez au conteneur en cours d'exécution :

**docker exec -it tuleap-db-test mysql**

1. **Réinitialisez le mot de passe root** :

Une fois dans le shell MySQL, exécutez les commandes suivantes pour réinitialiser le mot de passe root :

FLUSH PRIVILEGES;

ALTER USER 'root'@'localhost' IDENTIFIED BY 'tuleapdb';

ALTER USER 'root'@'%' IDENTIFIED BY 'tuleapdb';

Remplacez new\_password par le nouveau mot de passe que vous souhaitez utiliser.

1. **Arrêtez et supprimez le conteneur temporaire** :

Quittez le shell MySQL :

exit

Arrêtez et supprimez le conteneur temporaire :

docker stop mysql\_temp

docker rm mysql\_temp

1. **Redémarrez votre conteneur MySQL avec le nouveau mot de passe** :

Modifiez votre fichier docker-compose.yml ou la commande docker run pour inclure le nouveau mot de passe. Si vous utilisez Docker Compose, cela pourrait ressembler à ceci :

version: '3.8'

services:

db:

image: mysql:5.7

environment:

MYSQL\_ROOT\_PASSWORD: new\_password

volumes:

- db\_data:/var/lib/mysql

ports:

- "3306:3306"

restart: always

Assurez-vous de remplacer new\_password par le mot de passe root que vous avez défini.

Ensuite, redémarrez votre conteneur MySQL :

docker-compose up -d

### Vérification

Une fois le conteneur redémarré, vous pouvez vérifier que le nouveau mot de passe fonctionne en vous connectant à MySQL avec :

docker exec -it <mysql\_container\_id> mysql -u root -p new\_password

Cela devrait vous permettre de vous connecter à la base de données MySQL avec le nouveau mot de passe root.

**Samedi 20/07/2024**

Sur SDU9.GOZAIMASS.IO

Créer les images dlaubo/tuleapdb (mysql) et dlaubo/tuleap (tuleap)

docker commit tuleap **dlaubo/dlaubo-tuleap:latest**

docker commit tuleap-db **dlaubo/dlaubo-tuleapdb:latest**

Se connecter sur dockerhub avec le compte dlaubo

docker login

username : dlaubo

password : S\*\*\*\*\*\*3

docker push **dlaubo/dlaubo-tuleap:latest**

docker push **dlaubo/dlaubo-tuleap:latest**

Vérifier sur DockerHub la présence de deux images

Extrait de docker-compose.yml

services:

tuleap:

container\_name: tuleap

image: dlaubo/dlaubo-tuleap:latest

#hostname: ${TULEAP\_FQDN}

hostname: localhost

restart: always

ports:

- "8081:80" # Map container port 80 to host port 8081 (HTTP)

- "4443:443" # Map container port 443 to host port 4443 (HTTPS)

- "2222:22" # Map container port 22 to host port 2222 (SSH)

volumes:

#- /var/run/docker.sock:/var/run/docker.sock

- tuleap-data:/data

depends\_on:

- db

environment:

#- TULEAP\_FQDN=${TULEAP\_FQDN}

- TULEAP\_FQDN=localhost

- TULEAP\_SYS\_DBHOST=db

#- TULEAP\_SYS\_DBPASSWD=${TULEAP\_SYS\_DBPASSWD}

- TULEAP\_SYS\_DBPASSWD=tuleap

#- SITE\_ADMINISTRATOR\_PASSWORD=${SITE\_ADMINISTRATOR\_PASSWORD}

- SITE\_ADMINISTRATOR\_PASSWORD=tuleap

- DB\_ADMIN\_USER=root

#- DB\_ADMIN\_PASSWORD=${MYSQL\_ROOT\_PASSWORD}

- DB\_ADMIN\_PASSWORD=tuleapdb

**Lundi 22/07/2024**

Planification des tâches sur Tuleap

Modification des champs sur un widget (task, release, user stories….) fare save rapport en haut de la page pour que le tableau de bord le prend en compte.

Exemple champs Description dans Release 🡺 le champs était ajouté et sauvegarde rapport le fait apparaitre dans le tableau de bord du projet

Tuleap & Onlyoffice

Token= D2voHsN54NABLxCDvJamm8Flsr5XUCxt

URL = <https://51.159.18.105:8088>

Sdu6 cherche des infos

On veut

Sdu6 exécute les playbooks sur Sdu9 qui lance les scripts

**Refaire l’agent (proteus-agent)**

Ranger les services dans /home/installer/services

Apache

Tuleap

…

…

Apache catégorie http : déploiement

Sudo Python3 -m http.server 90 &

**Proteus-agent (sdu9)**

**Installation des services : OK**

Save et restore pour tuleap

Installer Kubernetes, openstack,terraform… sur SDU8

Compte SDU8 (dlaubo/gozaimaa)

**Pour créer une roadmap dans Tuleap, vous aurez besoin de 2 choses :**

activer le widget « Agile Dashboard » dans votre projet et le configurer pour Scrum,

le widget « Roadmap »

**Mettre à jour les plugins de tuleap**

**Suivre la doc :** [Plugins installation — Tuleap latest version documentation](https://docs.tuleap.org/installation-guide/step-by-step/install-plugins.html)

Les commandes seront exécutées à l’intérieur du conteneur

Se connecter au conteneur tuleap

docker exec -it tuleap bash

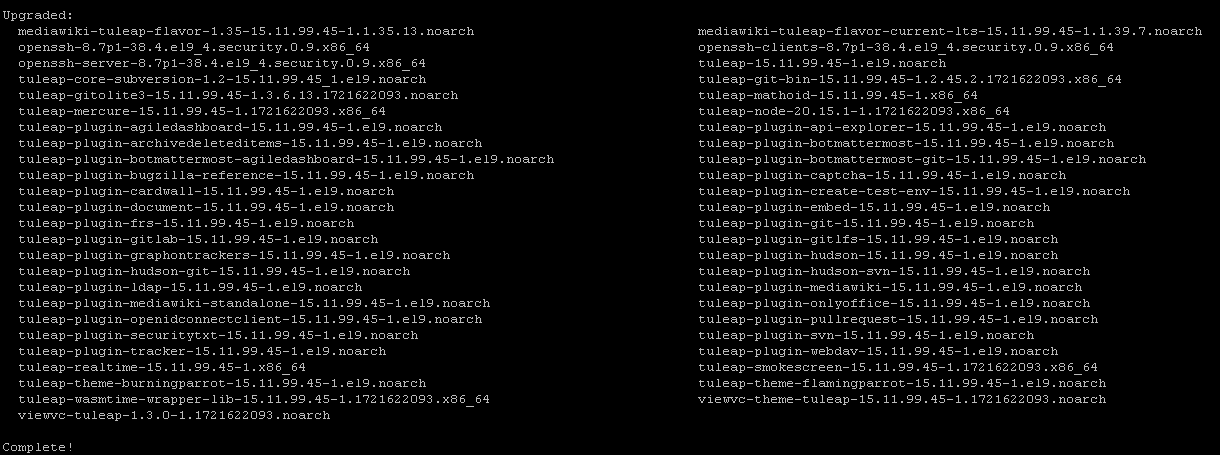
executer les commandes suivantes

dnf update

Les plugisn se trouvent

ls -la /usr/share/tuleap/plugins/

nous aurions cette liste des plugins mise à jour.



Chercher les fichier tuleap-cfg pour recharger les plugins

Find / -name ‘’tuleap-cfg’’

Afficher le contenu de fichier trouvé

cat /usr/bin/tuleap-cfg

Recharger les sites et les plugins

/usr/bin/tuleap-cfg **site-deploy**

Faire exit du conteneur et redemarrer le

Docker restart tuleap

Se connecter sur le site Admi tuleap, vérifier les plugins

Les commandes pour l’installation des plugins dans le conteneur de tuleap

dnf install <https://ci.tuleap.net/yum/tuleap/rhel/9/dev/x86_64/tuleap-community-release.rpm>

* **Install Tuleap** by running the following command:

dnf install **-**y \

mysql**-**server \

tuleap \

tuleap**-**theme**-**burningparrot \

tuleap**-**theme**-**flamingparrot \

tuleap**-**plugin**-**agiledashboard \

tuleap**-**plugin**-**graphontrackers \

tuleap**-**plugin**-**git \

tuleap**-**plugin**-**hudson**-**git \

tuleap**-**plugin**-**pullrequest \

tuleap**-**plugin**-**gitlfs \

tuleap**-**plugin**-**document \

tuleap**-**plugin**-**onlyoffice \

tuleap**-**plugin**-**embed \

tuleap**-**plugin**-**gitlab \

tuleap**-**plugin**-**openidconnectclient \

tuleap**-**plugin**-**ldap

**Vendredi 26/07/2024**

**volumes** : Le volume à utiliser pour la persistance des données. Le chemin /home/installer/services/tuleap/tuleap-data sur l'hôte est monté sur /data dans le conteneur.

Sudo ./proteus-save.sh tuleap

Pour tuleap: un fichier tar sera créer avec les conteneurs tuleap et tuleap-db

Et une image du conteneur sera créée

# commit(snaptshot) (docker commit emet un sha256 a l'execution de la commande)

docker commit "$app\_name" "${image\_name\_f}-${date\_time}" > /dev/null

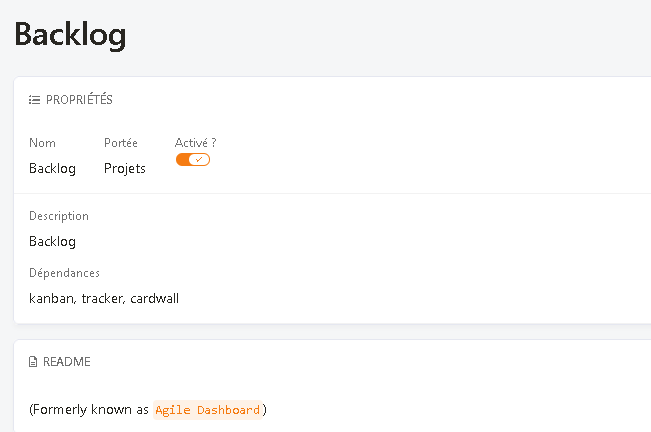
**Lundi 29/07/2024**

Installer Selinium

Faire un pipeline de tests des applications

Reprise configuration tuleap avec les plugins pour accéder à Agile Dashboard

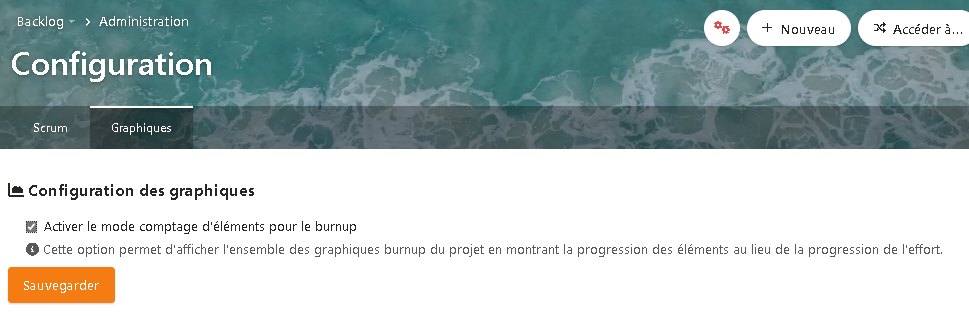
Dans les plugins nous retrouvons Backlog (formely knomw as Agile Dashbord)



Configurer l’onglet Backlog

Aller dans Backlog > Administration

Cocher la case Activer le mode comptage d’élément pour le burmup



Mardi 30/07/2024

Backup\_dir = /home/installer/allbackup

Backup\_file\_name= tuleap-${date\_time}.tar

**service\_backup\_sh =** /home/installer/proteus-agent/configurations/services/tuleap/deploiement/backup.sh

si existe $**service\_backup\_sh**

alors

**$service\_backup\_sh**  /home/installer/allbackup

sinon

**generic\_backup.sh**  /home/installer/allbackup

fin si

sinon on continue

lance le generic

/home/installer/proteus-agent/scripts/proteus-backup.sh **service**

Appel **backup**-service.sh

Appel /home/installer/proteus-agent/configurations/backup/backup.sh **service**

**Save dans** /home/installer/allbackup

service-date.time-service.tar 🡺 **dossier du service**

service-date.time.tar 🡺 le conteneur du service

**Le contenu du proteus-install.sh**

ansible-playbook -i /home/installer/proteus-agent/configurations/ansible/inventory.yml “/home/installer/proteus-agent/configurations/ansible-playbook/ansible-install.yml”--extra-vars "service=$service host=$host" --ask-become-pass

La commande pour récupérer le nom de la base des données

docker exec -it tuleap env | grep PROTEUS\_DB\_NAME | cut -d '=' -f 2

La commande pour récupérer le nom du volume

docker exec -it tuleap env | grep PROTEUS\_VOLUME\_NAME | cut -d '=' -f 2

docker-compose stop db

docker run --name mysql\_recovery -e MYSQL\_ALLOW\_EMPTY\_PASSWORD=yes -v tuleap\_tuleap\_db-data:/var/lib/mysql -p 3306:3306 --rm -d mysql:8.0 --skip-grant-tables --skip-networking

**Chercher les dépendances et les volumes d’un service**

PROTEUS\_DEPENDS\_ON=tuleap-db,….

**Configuration de volumes et des networks**

**volumes:**

tuleap\_tuleap-data:

name: tuleap\_tuleap-data

#external: true # A decommenter après la création du volume, external: true utilise le volume existant

tuleap\_db-data:

name: tuleap\_db-data

#external: true # A decommenter après la création du volume, external: true utilise le volume existant

**networks:**

proteus-network:

driver: bridge

external: true # utilisation du réseau existant pour le projet proteus

tuleap\_default:

name: tuleap\_default # définir le nom du réseau par default, à commenter après la création du réseau

driver: bridge

**commandes shell**

Cat st.cfg | substitute

echo "Dépendances du conteneur $container\_name dans $compose\_file :"

grep -A 10 "tuleap:" "docker-compose.yml" | grep -E 'depends\_on|services'

Pour récupérer les informations de la directive depends\_on d'un conteneur Docker en utilisant docker container inspect, vous devrez inspecter la configuration du conteneur et extraire les informations pertinentes. Toutefois, il est important de noter que la directive depends\_on est une fonctionnalité de docker-compose et ne fait pas partie de la configuration d'un conteneur Docker de base.

La commande docker container inspect renvoie des informations détaillées sur un conteneur Docker, mais elle ne comprend pas directement les dépendances depends\_on telles qu'elles sont définies dans un fichier docker-compose.yml. Néanmoins, vous pouvez vérifier les liens et les réseaux qui peuvent donner une idée des dépendances entre conteneurs.

**Étape 1: Inspecter le Conteneur**

Exécutez la commande suivante pour inspecter le conteneur et obtenir les informations en format

docker container inspect tuleap > tuleap\_inspect.json

**Analyser le contenu du fichier Json**

cat container\_inspect.json | jq .

Connaitre le nombre de volumes dans un docker

docker inspect tuleap --format '{{json .Mounts}}' | jq length

Annexes

[Blog Eleven Labs - Comprendre et Personnaliser son environnement Docker (eleven-labs.com)](https://blog.eleven-labs.com/fr/comprendre-et-personnaliser-son-environnement-docker/)

Copiier entre les serveurs

scp /\*\*\*\*\*\*\*\*\*\*\*/\*.tar [installer@sdu8.gozaimaa.io:/home/installer/backup](mailto:installer@sdu8.gozaimaa.io:/home/installer/backup)

scp /\*\*\*\*\*\*\*\*\*\*\*/\*.tar [installer@sdu8.gozaimaa.io:/home/installer/backup](mailto:installer@sdu8.gozaimaa.io:/home/installer/backup)

scp tuleap\_2024-08-08-16h58\_dossier.tar [dlaubosdu10.gozaimass.io:/home/installer/backaup/tuleap](mailto:dlaubo@sdu10.gozaimass.io:/home/installer/backaup/tuleap)

@

**Lundi 05/08/2024**

Vérifier si -it est nécessaire pendant la récupération de paramètre d’environnement ?

Factoriser le code fonction echo ‘’

Vérifier les éléments archivés si type=bind ?

Sdu10

Dlaubo / gozaimass@2025

**Containerisation de Proteus-agent sur sdu10.gozaimass.io**

**Création d’un user proteus**

Sudo adduser proteus

**Pour supprimer un utilisateur**

Sudo usermod -rG nom\_grope user

**Pas besoin d’ajouter proteus dans le groupe sudoers**

Dans /home/proteus

**Générer les clé ssh**

Ssh-keygen

Prendre la clé public et copier la dans votre compte gitlab

Git clone proteus-agent

Créer le Dockerfile pour contenairiser proteus-agent à la racine de proteus-agent

**Mardi 06/08/2024**

Domain sur sdu10 : avemandela.io

IP sud10 : 51.15.180.11

Dans les scripts utiliser **source vars.bash ??**

**Utilisation de la commande dirname dans le proteus-restore.sh**

**Création du network proteus-network manuellement**

**La première fois**

Sdu10 🡺 /var/lib/docker/volumes

Sdu9 🡺 /data/docker/volumes/

Restorer les volumes

Sdu10 🡺

cd /var/lib/docker/volumes

dans /home/installer/backup

**data/docker/volumes**/tuleap\_tuleap-data/\_data\*\*\*\*\*\*\*

**data/docker/volumes**/tuleap\_db-data/\_data\*\*\*\*\*\*\*

volumes:

- ./volumes/tuleap\_db-data:/var/lib/mysql

# =====================================================================

**Pour le restore.sh**

Docker-compose down

Deploiement-old en deploiement-2024-08-07-HH

Les droits sur le fichier .tar et les volumes ? root ou pas

Chown les dossier /home/installer à installer :installer

Ajouter l’utilisateur X au groupe installer

sudo usermod -aG installer dlaubo

sudo usermod -aG installer djoan

sudo usermod -aG installer pascal

sudo usermod -aG installer proteus

Backup & restore : à exécuter avec sudo

./proteus-restore.sh fichier\* tar » service 🡺 si tar pas présent chercher le dernier tar

Lancer le service tuleap

Utiliser les variables de .env

Première étape de la cybersécurité : backup & restore (service existant)

Utiliser les services :

/home/installer/services/**tuleap**

Dans le tar de backup : pas le

Build.sh

Setup.sh

Installation de docker-compose : à Djoan 🡺 utiliser docker compose sans tirait entre les deux

**Réinitialiser les logs via la suppression du fichier de logs**

**Troncature du fichier**

sudo truncate -s 0 $(docker inspect --format='{{.LogPath}}' tuleap)

**Supprimer le fichier**

sudo rm $(docker inspect --format='{{.LogPath}}' tuleap)

**vendredi 09/08/2024**

DCIT: réseau?

Le server STUN: faciliter la communication entre machines en passant par Firewall (ex pc perso et free)

[Qu’est-ce qu’un serveur STUN •• Explications | 3CX](https://www.3cx.fr/voip-sip/stun-server/)

**Se connecter à tuleap-db via tuleap**

mysql -h tuleap-db -u tuleapadm -p 🡺 user applicatif

mysql -h tuleap-db -u root -p

**Connaitre le conteneur qui utilise le volume**

In case you find a volume you don’t know which container is using, simply use

**Git clone**

to find details about the container(s) that use the image.

**Effacer les logs d’un conteneur**

sudo truncate -s 0 /data/docker/containers/4563ceb487e89f7bc5a6f3240c9bb301c1f80392ac17973f49fdf85e6075d4d1/tuleap-json.log

**Penser à definer le nome du domaine dans le fichier**

Modifier ce fichier

../tuleap\_tuleap-data/etc/tuleap/conf/database.inc

OSRM : un outil?

**Remplacement d’une valeur dans un fichier**

sed -i 's/vogoumpage.org/avemandela/g' local.inc

find /chemin/vers/repertoire -name "local.inc"

grep -ri "chaine\_de\_caracteres" /chemin/vers/repertoire

sed -i 's/vogoumpaga.org/avemandela.org/g' $(grep -rl "vogoumpaga.org" .)

docker logs tuleap|grep tail -f

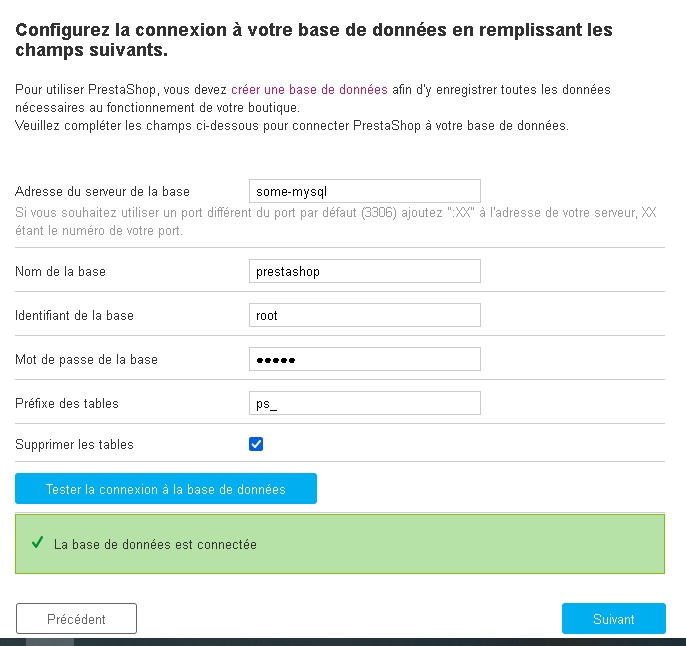
sdu9: password temporaire 🡺 gozaimass@2025

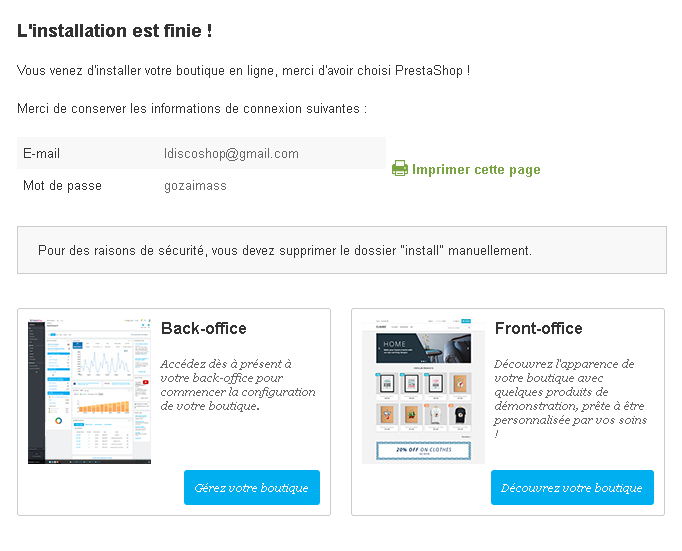
**Vendredi 16/08/2024**

Installation de prestashop sur sdu10

Le compte gmail : [ldiscoshop@gmail.com](mailto:ldiscoshop@gmail.com)

Passe= gozaimass





**Penser à mettre une structure pour initialiser un projet sur gitlab**

Git init

Git template :

[Création d’un modèle de dépôt - Documentation GitHub](https://docs.github.com/fr/repositories/creating-and-managing-repositories/creating-a-template-repository)

Version gitlab de gozaimass.io[**v16.8.2**](https://gitlab.com/gitlab-org/gitlab-foss/-/tags/v16.8.2)

Ou script proteus-init

**Lundi 19/08/2024**

Tuleap Gantt : la version community ne prend pas tuleap-plugin-roadmap

L’écriture de script proteus.sh pour le code générique des scrpts

**Mardi 20/08/2024**

Reprise de script proteus.sh

Nettoyage des codes

Sur sdu10

Création d’un compte proteus-deploiement

Création d’un compte proteus-installer

Suppression du répertoire proteus-agent/configurations

Création d’un compte proteus-agent

Se connecter à su proteus-agent

Création d’une clé ssh pour accéder à gitlab

Déposer la clé ssh avec le compte proteus-agent sur gitlab

Admin donnera les droits sur les repository des projets en tant que reporter sur proteus-agent et proteus/services/deploiements

Aller dans le dossier utilisateur /home/proteus-agent

Mkdir deploiements

Cd deploiements

Git clone gti.\*\*\*\*\*\* .proteus/services/deploiements/prestashop.git

Cd ..

Git clone gti.\*\*\*\*\*\* .proteus-agent.git

Penser au tag pendant le commit dans deploiements

git tag -a v1.0.0 <commit\_hash> -m "Message du tag"

git clone --branch v1.0 --single-branch [git@git\*\*\*\*\*\*.repository.git](mailto:git@git******.repository.git)

**Mercredi 21/08/2024**

**Installer glab sur sdu10**

**Exemple : Installer glab**

Si vous avez téléchargé un fichier .deb pour glab, l'installation se ferait comme suit :

**Téléchargement du fichier .deb** :

wget https://gitlab.com/gitlab-org/cli/-/releases/v1.45.0/downloads/glab\_1.45.0\_Linux\_x86\_64.deb

**Installation du package** :

sudo dpkg -i glab\_1.45.0\_Linux\_x86\_64.deb

sudo apt-get install -f # pour résoudre les dépendances

**Vérification** :

glab --version

Une fois glab installé, créer un token sur gitlab

Exporter les paramètres : gitlab\_home et gitlab\_token

export GITLAB\_HOST=https://gitlab.gozaimass.io

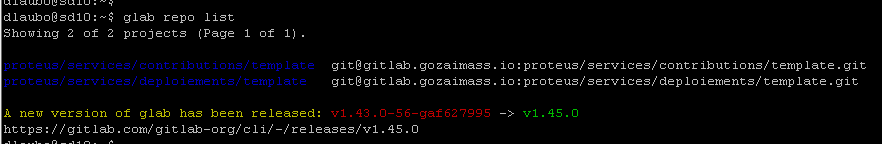
export GLAB\_TOKEN=glpat-fZn7kgobFSV46FySRqN8

Se connecter via glab à gitlab

glab auth login



glab repo list



**Afficher les groups dans gitlab**

curl --header "PRIVATE-TOKEN: glpat-fZn7kgobFSV46FySRqN8" <https://gitlab.gozaimass.io/api/v4/groups>

**Créer un groupe**

curl --request POST --header "PRIVATE-TOKEN: glpat-fZn7kgobFSV46FySRqN8" --data "name=subgrp2&path= testdlaubo" <https://gitlab.gozaimass.io/api/v4/groups>

**Créer un repo sur gitlab, dans un groupe existant**

glab repo create testdlaubo myrepo

glab repo create **--group testdlaubo/subgrp** myrepo

**Supprimer un depot**

glab repo delete testdlaubo/subgrp/myrepo -y

**Déconnecter glab de gitlab**

glab auth login

Chercher l’endroit où se trouve la matrice des droits

Les droits en fonction de rôle dans gitlab.

Lancer docker compose enindiquant le fichier docker-compose.yml

docker-compose -f chemin/vers/docker-compose.yml up -d

**Jeudi 22/08/2024**

Openproject

admin / gozaimass@2025

Users Open project

Lauboudou DIA dlaubo/ gozaimass123

Vlad

Anderson

Djoan

Pascal

elastic search

elastic / 2=j5dQVuV-97\*+4bLB0k

OpenProject API Token

5e6c842ba039096e6602d8f2b35f96a714a7fcf0ce642aad02314e4b0b96ee47

Affciher la totalité des work packages

GET <https://projects.vogoumpaga.org/api/v3/work_packages>

Affichage des work packages du projet proteuscode

GET <https://projects.vogoumpaga.org/api/v3/projects/proteuscode/work_packages>

**Créer un work package via POST request**

POST <https://projects.vogoumpaga.org/api/v3/projects/proteuscode/work_packages>

**Authentification**

Auth Basic, token

**Body**

 "subject":"us1",

    "description": {

        "format": "textile",

        "raw": "My raw @textile@ formatted description. \_Bye guys!\_"

    },

    "\_links": {

        "type": {"href":"/api/v3/types/1"},

        "status":{"href":"/api/v3/statuses/1"},

        "priority":{"href":"/api/v3/priorities/8"}

    }

**Postman Token à passer dans l’outil pour,**

postman://auth/callback?code=6c43298b6dee2fceb566506742f5c580905e3e84d534d257de40c995fed53820

**La documentation de l’API REST OpenProject**

[API: Work Packages (openproject.org)](https://www.openproject.org/fr/docs/api/endpoints/work-packages/#work-package-create-form-for-project)

Interface de communication REST avec openproject

[Step by step guide to synchronize your Excel Sheet with OpenProject](https://www.openproject.org/blog/synchronize-excel-openproject/)

Le projet sur github : [opf/OpenProjectExcel (github.com)](https://github.com/opf/OpenProjectExcel)

Débloquer les macros du fichier excel :

[Une macro potentiellement dangereuse a été bloquée - Support Microsoft](https://support.microsoft.com/fr-fr/topic/une-macro-potentiellement-dangereuse-a-%C3%A9t%C3%A9-bloqu%C3%A9e-0952faa0-37e7-4316-b61d-5b5ed6024216)

Utiliser Ctrl+b ou Ctrl+n

[Déployer Elasticsearch, Kibana et Logstash (ELK Stack) avec Docker Compose | Douleur moyenne (medium.com)](https://medium.com/@kayvan.sol2/deploy-elasticsearch-kibana-logstash-with-docker-compose-df518d68731d)

Voir l’article pour comprendre

**Lundi 26/08/2024**

Import éléments de tuleap dans openproject

Terminologie

|  |  |
| --- | --- |
| **Tuleap** | **OpenProject** |
| Sprint | Version |
| User story | User story |
| Epic : new web site | Epic |
| **Release** | Version |
| Release | Milistone |
| Dev, Prod, … | Phase |
|  | Feature |

A **milestone** is a specific point within a project's life cycle used to measure the progress toward the ultimate goal. Milestones in project management are used as signal posts for a project's start or end date, external reviews or input, budget checks, submission of a major deliverable, etc

User story

The project team knows how to solve a customer’s problem. A user story is part of how to solve the customer’s problem, i.e. something that the customer wants. In order to make it a workable unit, the user story should have a scope that can be processed within one sprint. User stories are often used in the product backlog,

Epic

An epic describes the needs of a customer or an insight. The epic will get broken down into smaller, workable user stories that can be processed during a sprint. So you could call an epic also a sum of user stories.

Feature

A feature is a functionality that the customer values and that will deliver business value.

Bug

A bug is an unexpected issue. A project management software can help you perform [bug tracking](https://www.openproject.org/collaboration-software-features/task-management/#bug-tracking).

Story points

Story points are assigned values (points) to items (tasks, features, user stories etc.) to give them a priority. They show the estimated effort to fully complete the item. In order to make a good estimation about the effort, it is important to agree as a team on how to use the set scale. At the same time, you need to make sure that you do the estimation in the team where everyone involved in the item is taking part.

**Préparer le build.sh**

Il sera utilisé par le pipeline de vlad 🡺 pas de fichier build.sh

Setup.sh 🡺 à réviser également

**Mardi 27/08/2024**

Ce lien explique les différents concepts de OpenProject

[Boards for Agile Project Management (openproject.org)](https://www.openproject.org/docs/user-guide/agile-boards/#create-a-new-board)

[Time tracking and cost reporting (openproject.org)](https://www.openproject.org/docs/user-guide/time-and-costs/)3

**Stage**

Outil de supervision : Prometheus

Injecter les données d’un agent Zabbix dans elastic search : zabbic monitoring des ressources (machines, parc des postes). Peut faire un doublon avec Logstach ??

Installer les agents zabbix sur les postes. Exemple : une attaque sur un site, l’avantage de zabbix est de l’installer pour identifier les machines vulnérables.

Regarder les indicateurs que Zabbix envoi et les points communs avec ELK (elastic LogStach kavana)

Voir les metrics des zabbix

[Handy Tips #36: Collecting custom metrics with Zabbix agent user parameters - Zabbix Blog](https://blog.zabbix.com/handy-tips-36-collecting-custom-metrics-with-zabbix-agent-user-parameters/22850/)

Suivre l’installation de Zabbix docker compose

[6 Installation from containers (zabbix.com)](https://www.zabbix.com/documentation/current/en/manual/installation/containers)

Récupérer le git repository suivant :

git clone <https://github.com/zabbix/zabbix-docker.git>

**Installer zabbix par le docker compose**

docker compose -f ./docker-compose\_v3\_ubuntu\_mysql\_local.yaml up -d

**Commandes pour installer zabbix sur debian par des commandes en ligne**

docker pull zabbix/zabbix-server-mysql

docker run --name zabbix-server -e DB\_SERVER\_HOST='sdu8.gozaimass.io' -e MYSQL\_USER='zabbix-mysql' -e MYSQL\_PASSWORD='zabbix-pass' -e MYSQL\_DATABASE='zabbix-db' -d zabbix/zabbix-server-mysql

docker run --name zabbix-web -e DB\_SERVER\_HOST='sdu8.gozaimass.io' -e MYSQL\_USER='zabbix-mysql' -e MYSQL\_PASSWORD='zabbix-pass' -e MYSQL\_DATABASE='zabbix-db' -p 8280:80 -d zabbix/zabbix-web-nginx-mysql

<http://sdu8.gozaimass.io:8280>

sdu8.gozaimass.io : réinstaller et reconfigurer

Aller étape par étape pour déployer tuleap via Dockerfile

Setup.sh

date > ./last\_build.txt

Cat last\_build.txt

Date du jour

Installer le docker compose sur sdu8

sudo apt install docker-compose

Création de l’image tuleap\_gozaimass :latest

Lancement de start.sh du service tuleap avec l’image tuleap\_gozaimass :latest

**Mercredi 28/08/2024**

Lancer un script via le docker-compose.yml

version: '3'

services:

app:

image: votre\_image

volumes:

- ./script.sh:/usr/local/bin/script.sh

command: ["/bin/bash", "-c", "/usr/local/bin/script.sh"]

**Jeudi 29/08/2024**

Avancement avec les scripts

La commande healthcheck : à regader

Préparation d’un script pour tester l’état d’un conteneur (démarré et healthy)

Se connecter à WIZLABS

Création d’une VM Ubuntu 20Go + 1 CPU + 1 G0 de RAM

Adresse IP publique

52.87.229.116

Nom utilisateur : ubuntu

**Vendredi 30/08/2024**

La machine sdu8 a changé des comportements suite à des manipulation et un crash hier.

Supprimer les volumes sans passer par rm -fR volumes

Voici la commande, monter un conteneur temporaire comme suit,

docker stop vgp-tuleap

docker stop vgp-tuleap-db

docker run --name vgp-tuleap-db-test -v ./volumes/tuleap\_db-data:/var/lib/mysql -e MYSQL\_ALLOW\_EMPTY\_PASSWORD=yes -d mysql:8.0 --skip-grant-tables

Démarrer le conteneur si pas e cas

docker start vgp-tuleap-db-test

se connecter à l’intérieur du conteneur

docker exec -it vgp-tuleap-db-test bash

Aller dans le répertoire du volume

cd /var/lib/mysql

Se connecter au volume de tuleap-data

docker run --name vgp-tuleap-test -v ./volumes/tuleap\_tuleap-data:/data -e MYSQL\_ALLOW\_EMPTY\_PASSWORD=yes -d vgp/tuleap:latest

Démarrer

docker start vgp-tuleap-test

Se connecter au conteneur

docker exec -it vgp-tuleap-test bash

docker network connect --link vgp-tuleap-db deploiement\_default vgp-tuleap

Remplacer une chaine dans un fichier :

:%s/echo/log/g

**Transformation des données automatiquement via un outil ETL open source**

Penser à **LogStach** pour faire la transformation et la migration des données

**Principaux Outils ETL Open Source**

1. **Apache Nifi**
   * **Description** : Apache NiFi est un outil ETL puissant conçu pour automatiser le flux de données entre systèmes. Il permet le transfert, la transformation et la gestion des données en temps réel.
   * **Caractéristiques** :
     + Interface utilisateur intuitive pour concevoir les flux de données.
     + Prise en charge des flux de données complexes.
     + Forte capacité de gestion des erreurs et de reprise.
     + Évolutivité et gestion des flux distribués.
   * **Cas d'utilisation** : Collecte et transformation de données en temps réel, intégration de données entre différents systèmes.
2. **Talend Open Studio for Data Integration**
   * **Description** : Talend Open Studio est un outil ETL complet qui permet d'effectuer des opérations d'extraction, de transformation et de chargement de données. Il est très populaire pour sa facilité d'utilisation et son large éventail de connecteurs.
   * **Caractéristiques** :
     + Interface glisser-déposer pour la conception des flux de données.
     + Large support de connecteurs pour bases de données, systèmes cloud, et applications.
     + Possibilité de générer du code Java à partir des jobs créés.
   * **Cas d'utilisation** : Intégration de données, migration de données, synchronisation de données entre systèmes.
3. **Pentaho Data Integration (Kettle)**
   * **Description** : Pentaho Data Integration, également connu sous le nom de Kettle, est un outil ETL puissant et polyvalent qui fait partie de la suite Pentaho. Il est utilisé pour la manipulation de données, la gestion des données volumineuses, et l'intégration des données.
   * **Caractéristiques** :
     + Interface graphique pour concevoir les flux de données.
     + Large support de sources de données, y compris les bases de données relationnelles, NoSQL, et les services web.
     + Exécution de transformations en mémoire, gestion de gros volumes de données.
   * **Cas d'utilisation** : Intégration de données d'entreprise, chargement de data warehouses, traitement de données volumineuses.
4. **Apache Airflow**
   * **Description** : Apache Airflow est un orchestrateur de flux de travail (workflow) qui permet de gérer des processus ETL complexes. Bien qu'il ne soit pas strictement un outil ETL, il est couramment utilisé pour orchestrer les processus ETL.
   * **Caractéristiques** :
     + Programmation de tâches ETL sous forme de DAGs (Directed Acyclic Graphs).
     + Forte intégration avec les services cloud, bases de données, et autres outils.
     + Possibilité de surveiller, gérer et dépanner les workflows via une interface web.
   * **Cas d'utilisation** : Orchestration de pipelines ETL complexes, automatisation des flux de travail de données.
5. **Apache Spark**
   * **Description** : Apache Spark est un moteur de traitement distribué qui prend en charge le traitement des données en mémoire. Il est particulièrement adapté aux grandes quantités de données et peut être utilisé pour des tâches ETL massives.
   * **Caractéristiques** :
     + Traitement en mémoire pour des performances élevées.
     + Prise en charge du traitement par lots et en temps réel.
     + Écosystème riche avec des modules pour SQL, machine learning, graphes, et flux de données.
   * **Cas d'utilisation** : Traitement de données massives, ETL en temps réel, traitement de données de streaming.
6. **Hevo Data**
   * **Description** : Hevo Data est une plateforme ETL basée sur le cloud qui propose une version communautaire. Bien qu'il ne soit pas entièrement open source, il mérite d'être mentionné pour ses capacités à gérer des pipelines ETL sans avoir besoin d'écriture de code.
   * **Caractéristiques** :
     + Interface simple et intuitive pour configurer les flux de données.
     + Prise en charge d'une large gamme de sources de données.
     + Synchronisation en temps réel des données.
   * **Cas d'utilisation** : Intégration rapide des données, pipelines ETL sans code.

**Conclusion**

Chacun de ces outils ETL open source présente des avantages spécifiques selon les besoins de votre organisation. Apache Nifi et Talend Open Studio sont parfaits pour une intégration de données facile et intuitive, tandis que Pentaho et Apache Airflow offrent plus de puissance et de flexibilité pour des pipelines de données plus complexes. Apache Spark est idéal pour des tâches de traitement de données à grande échelle. Choisissez l'outil en fonction de la complexité de votre flux de données, des volumes de données à traiter, et des exigences spécifiques de votre projet.

==================================================================================

**Lundi 02/09/2024**

Installation de Prestashop avec Docker Compose

Consulter le site : <https://github.com/PrestaShop/PrestaShop/blob/develop/README.md>

Créer un réseau

docker network create proteus-network

Créer un fichier d’env pour déclarer les paramètres d’environnement

docker-compose.yml

services:

mysql:

container\_name: diamond-mysql

image: mysql:5.7

restart: unless-stopped

environment:

MYSQL\_ROOT\_PASSWORD: ldiashop

MYSQL\_DATABASE: prestashop

networks:

- proteus-network

volumes:

- ./volumes/mysql-data:/var/lib/mysql

prestashop:

container\_name: prestashop

image: prestashop/prestashop:latest # Latest stable version of the PrestaShop, to see all available images go to ...

restart: unless-stopped

depends\_on:

- mysql

ports:

- 8011:80

environment:

DB\_SERVER: diamond-mysql

DB\_NAME: prestashop

DB\_USER: root

DB\_PASSWD: ldiashop

networks:

- proteus-network

volumes:

- ./volumes/prestashop-data:/data

networks:

proteus-network:

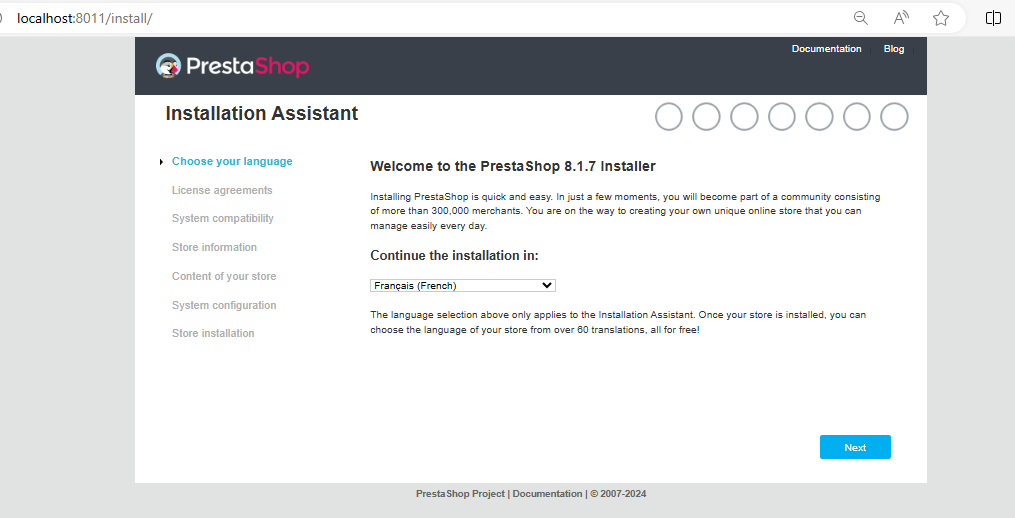
driver: bridge

external: true

Démarrer prestashop

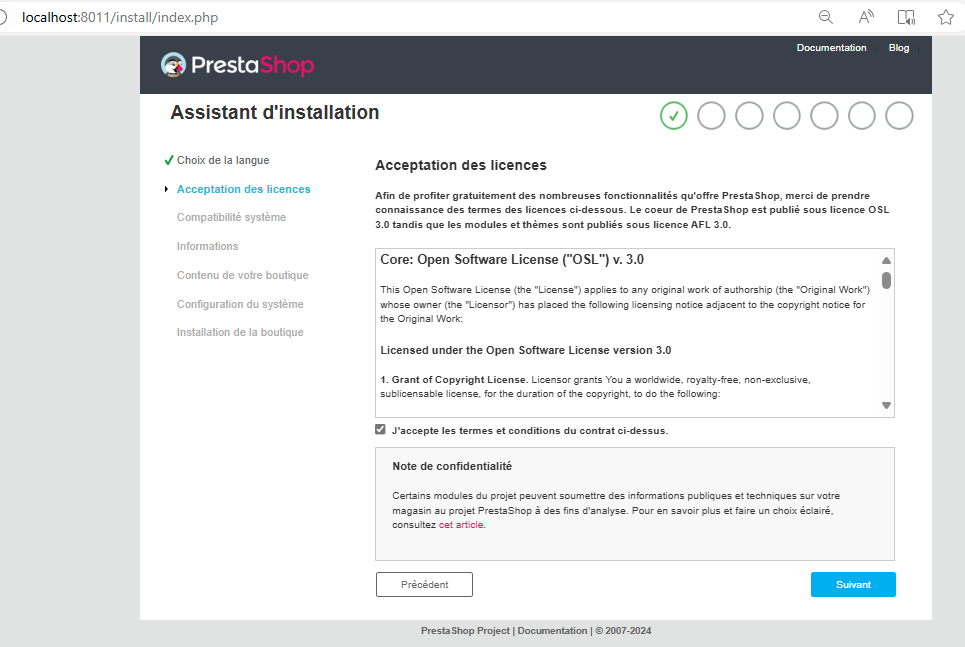
./start.sh

Accéder à la l’interface <http://localhost:8011>



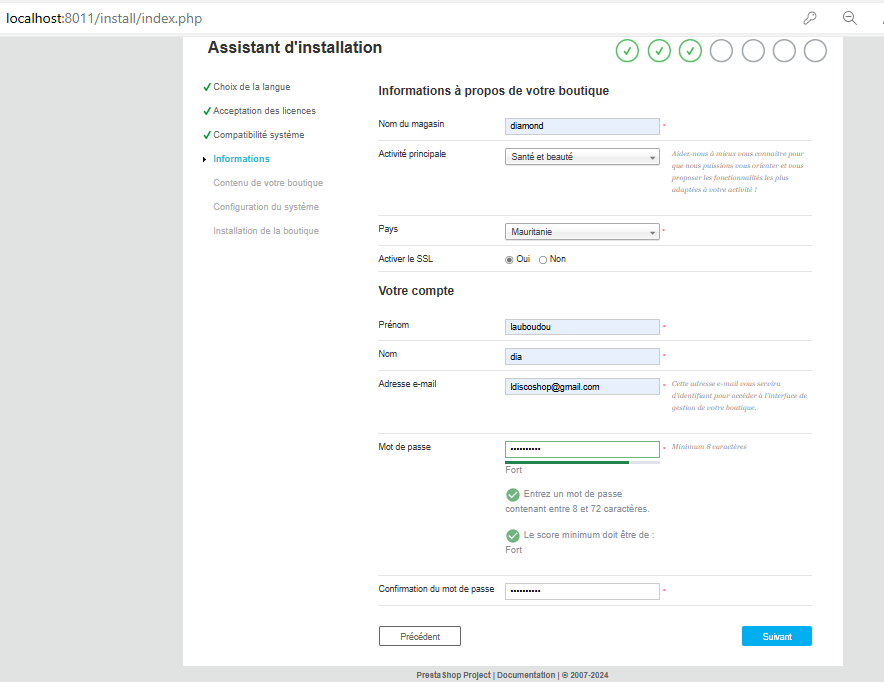
Commencer la configuration,

Accepter les termes et conditions du contrat (licence open source)

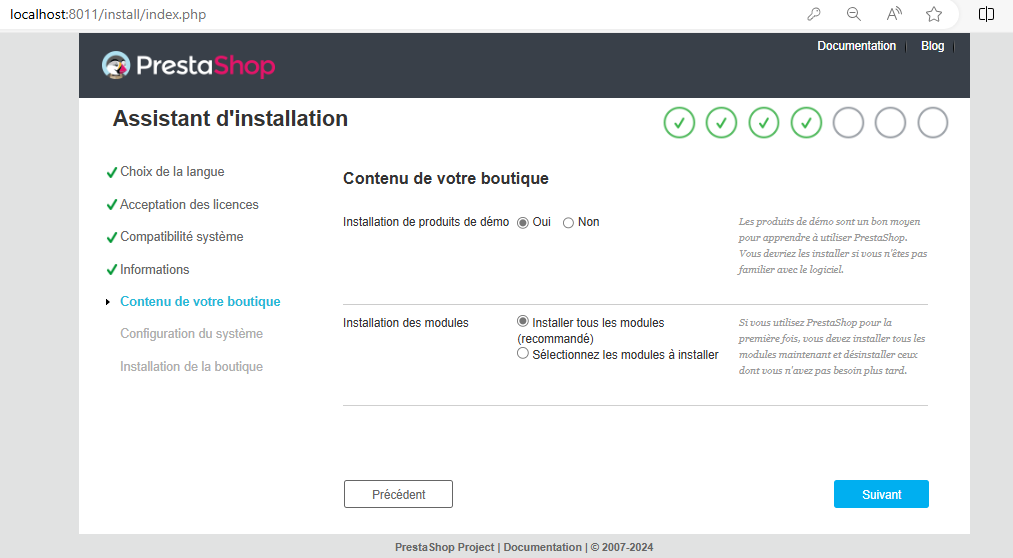


Configurer l’accès au site prestashop, compte : [ldiscoshop@gmail.com](mailto:ldiscoshop@gmail.com) passeword : S\*\*\*\_\*\*\*456

Compte client lauboudou identifiant : [ldiscoshop@gmail.com](mailto:ldiscoshop@gmail.com) passeword : lauboudou

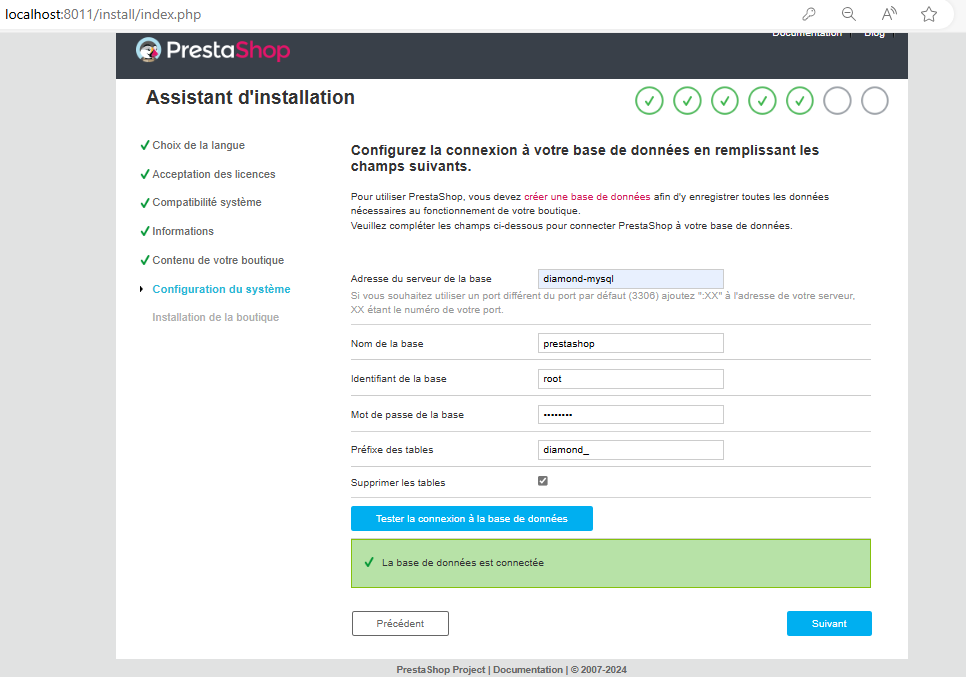


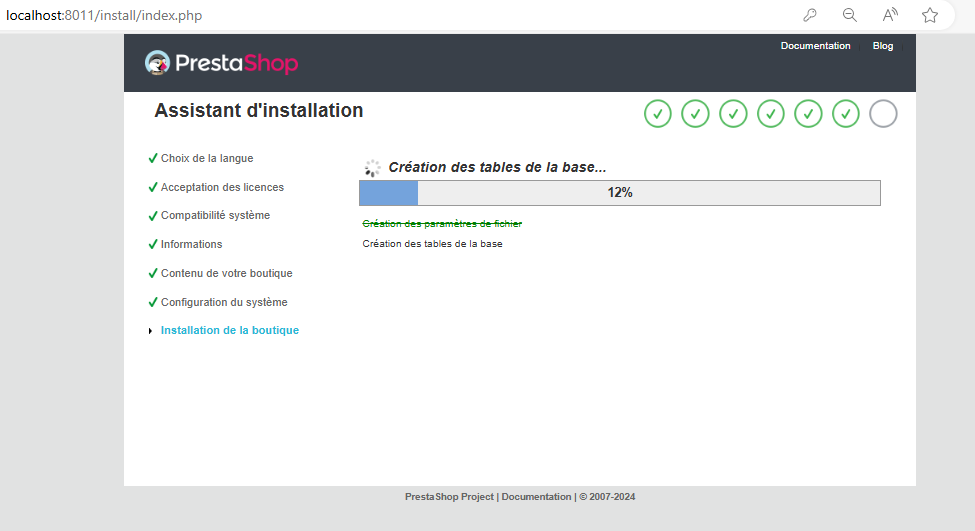
Choisir les modules à installer ainsi qu’une installation de démo ou pas



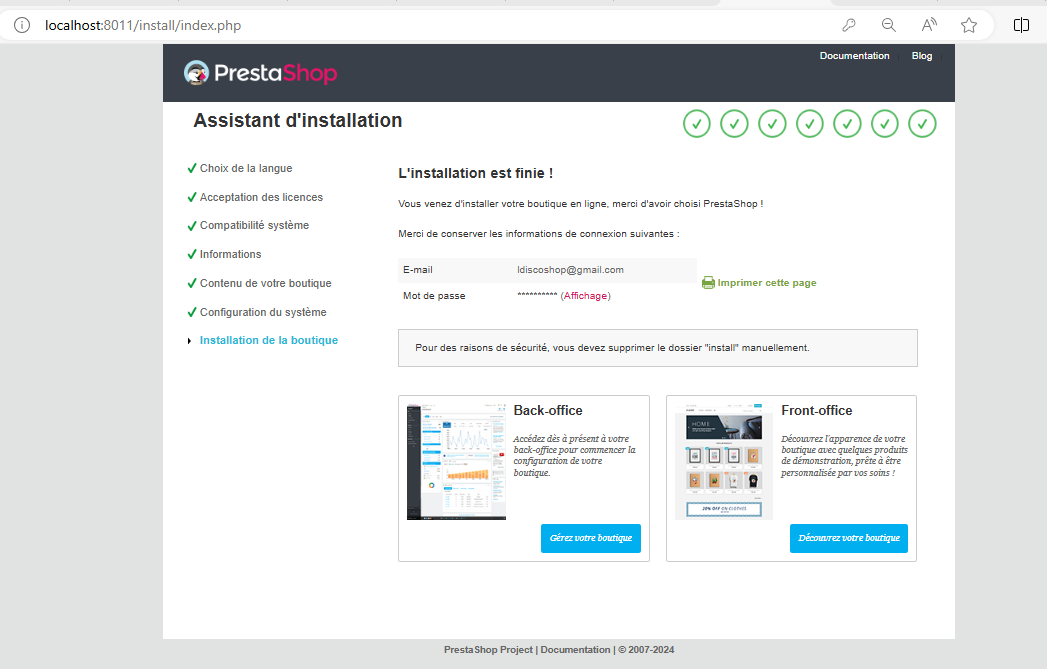
Configurer l’utilisation de prestashop avec la base des données mysql

Reprendre la configuration de la base mysql dans docker-compose.yml (fichier .env)





Fin de l’installation

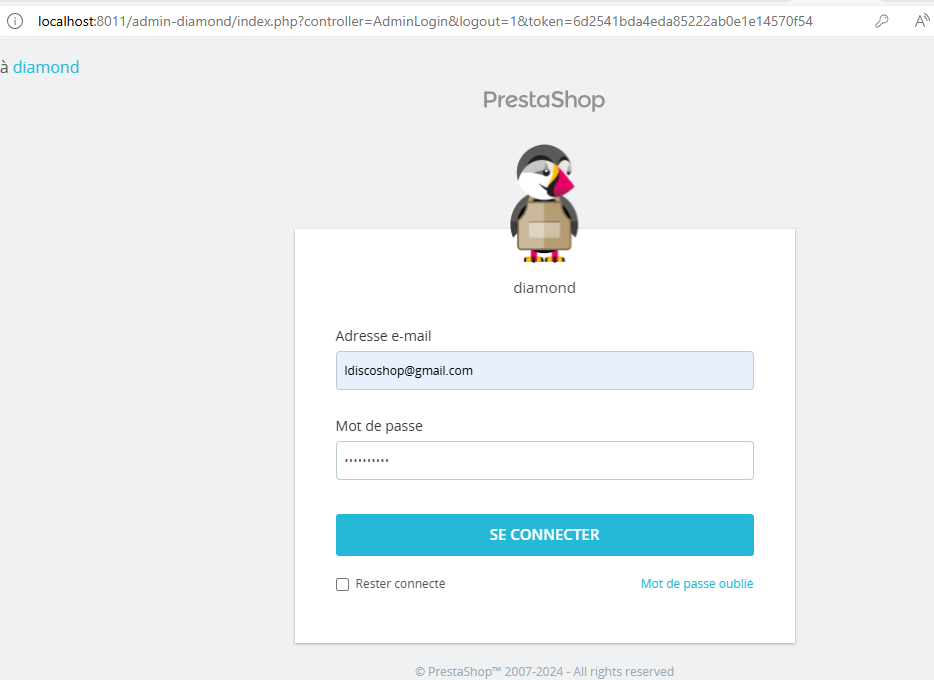




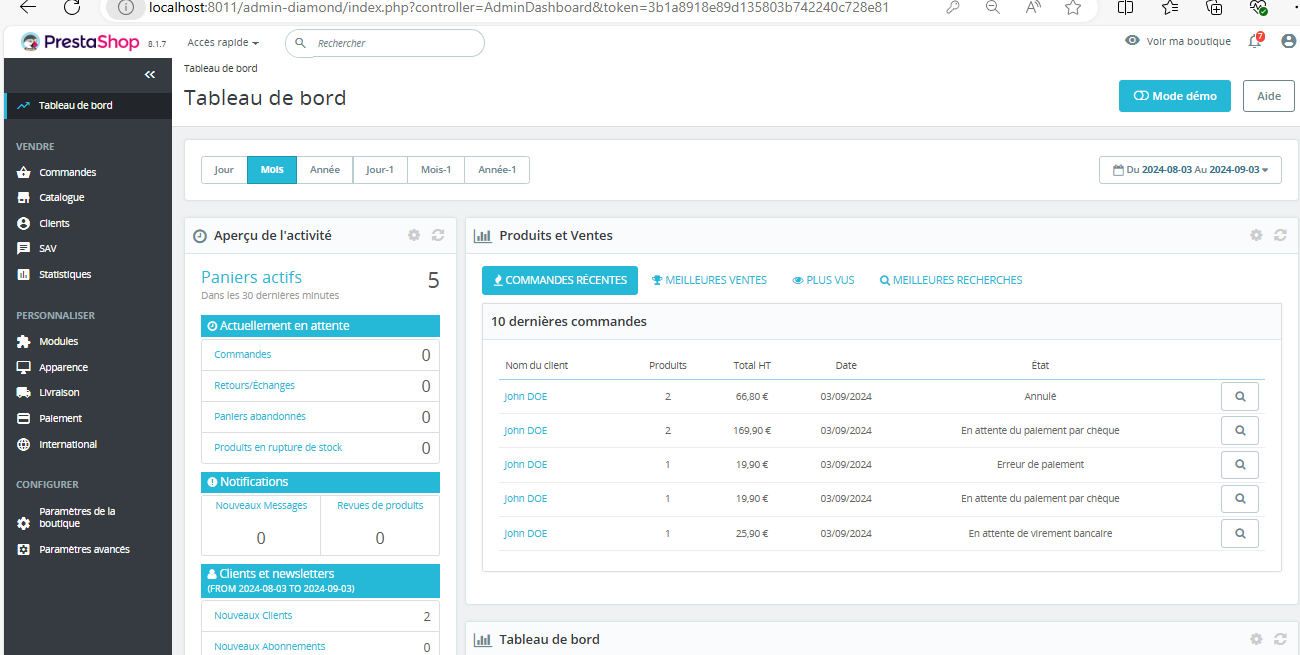
Pour des raisons de sécurité, penser à supprimer le dossier install (dans le dossier /var/www/html du conteneur prestashop)

Renommer le dossier /var/www/html/admin en /var/www/html/admin\_diamond

Se connecter à l’interface admin du site diamond <http://localhost:8011/admin_diamond>



La page de l’interface admin



Activer l’accès SSL sur Prestashop

[Tuto complet : https / ssl avec PrestaShop - Freelance Expert PrestaShop - WordPress - WooCommerce : Arnaud Merigeau (arnaud-merigeau.fr)](https://www.arnaud-merigeau.fr/tuto-complet-https-ssl-avec-prestashop/)

Le tuto présenté ici est effectué sur la version 1.5.3.1 de PrestaShop car moins claire que la version 1.6 et 1.7 de PrestaShop. IMPORTANT : **ce tuto peut aussi fonctionner avec les versions de PrestaShop 1.6 et supérieures**.  
Pour rappel, vous avez besoin d’un certificat SSL à commander auprès de votre hébergeur pour **activer le HTTPS sur votre boutique**.  
Pas besoin d’être un **expert PrestaShop** mais un peu de prudence, je vous conseille de faire une sauvegarde complète des fichiers et de la base de données. Une fois la sauvegarde terminée, rendez-vous dans le back office > préférences > général et activez le https en cliquant sur la phrase noire « Please click here to use HTTPS protocol before enabling SSL » (CF capture d’écran).

Générer un certificat csr

openssl req -nodes -newkey rsa:2048 -sha256 -keyout /home/diamond.key -out /home/diamond.csr

Une clé \*\*\*.key et \*\*\*\*.csr seront créés.

docker cp /home/diamond.key prestashop:/var/www/html/admin-diamond/.

docker cp /home/diamond.csr prestashop:/var/www/html/admin-diamond/.

Copier la clé publique SSH du serveur sur la racine du site admin-diamond

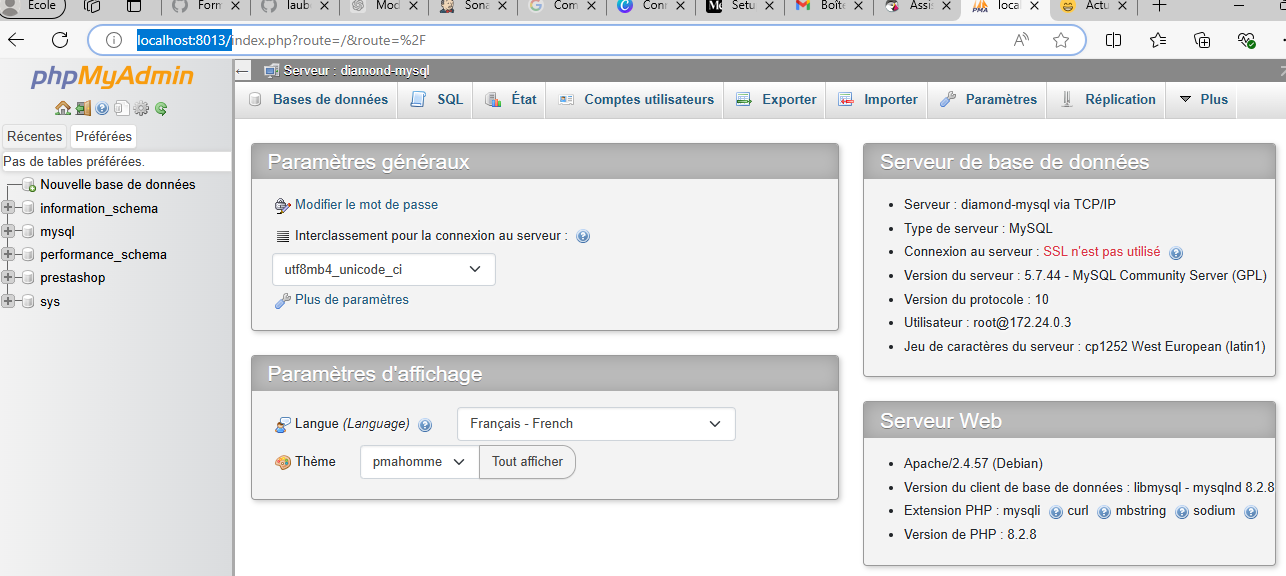
docker cp /home/dlaubo/.ssh/id\_rsa.pub prestashop:/var/www/html/admin-diamond/.

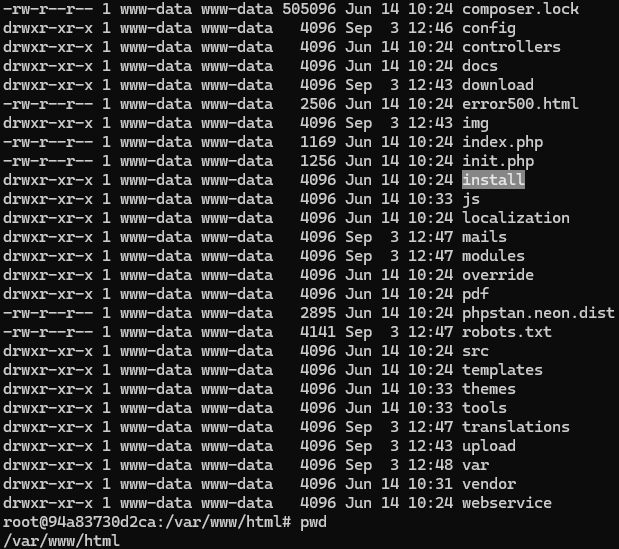
Site à consulter : [Comment sécuriser un site PrestaShop - PrestaSafe](https://www.prestasafe.com/comment-securiser-un-site-prestashop)

Phpmyadmin

<http://localhost:8013/>

compte root/ldiashop





Vérifier pourquoi ubuntu.22-04 ne fonctionne pas avec mysql :5.7 : à partir d’un shell ubuntu-22.04 ça ne fonctionne pas , sur un powershell cela fonctionne.

**Mercredi 04/09/2024**

Installation de zabbix

Suivre le projet git de zabbix : zabbix-docker

Installer la version ubuntu\_mysql\_Latest

Exécuter le docker compose en question

Accéder à l’interface zabbix sur l’url : <https://localhost> (le port 10051 peut n’est pas être renseigné)

**Installation de zabbix-agent**

zabbix-docker/blob/7.0/Dockerfiles/agent2/alpine/Dockerfile

Pour la récupération de l’image zabbix-build-mysql Changer la ligne numéro 4

ARG BUILD\_BASE\_IMAGE=zabbix-build-mysql:alpine-${ZBX\_VERSION}

par

ARG BUILD\_BASE\_IMAGE= **zabbix/**zabbix-build-mysql:alpine-${ZBX\_VERSION}

Maintenant aller dans ce répertoire

zabbix-docker/blob/7.0/Dockerfiles/agent2/alpine/

Lancer le build.sh

./build.sh

Le zabbix agent sera installé.

========================================================================

Vérification des packages installés sur ubuntu

dpkg --get-selections

**Vendredi 06/09/2024**

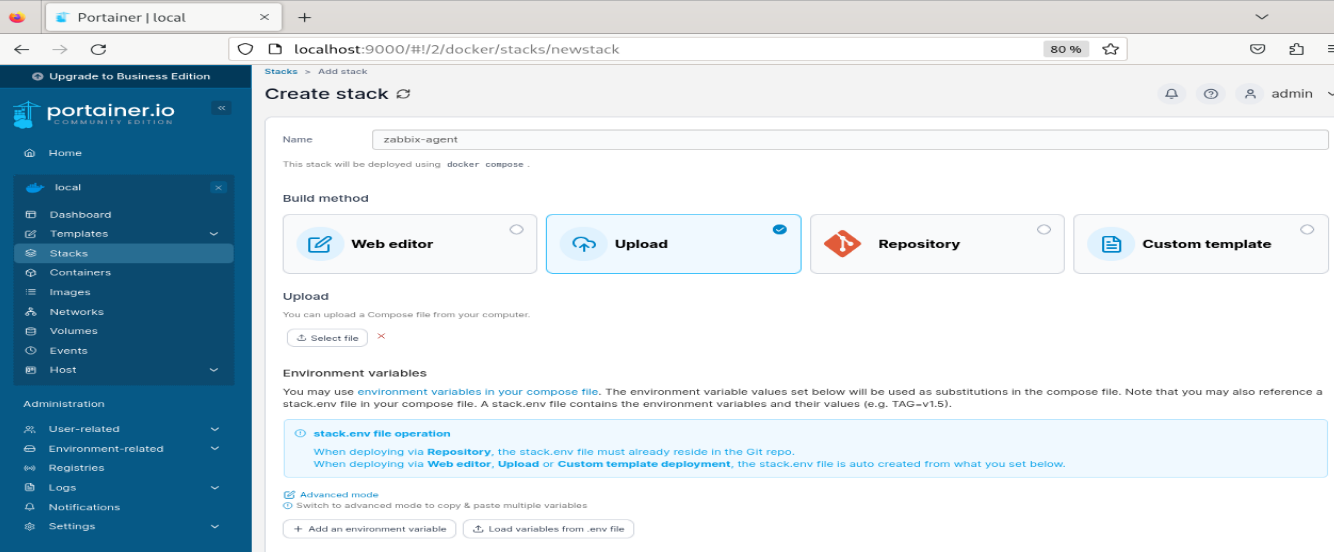
**Sur la VM Debian 22 : SRD-DevOps**

Prestashop installé

Zabbix téléchargé

Portainer compte : admin / portainer2024

Déployer zabbix via portainer



Charger le fichier docker-compose.yml et le fichier .env (variables)

Puis deployer zabbix

Ou

Déployer classiquement

docker compose -f docker-compose\_v3\_alpine\_mysql\_latest.yaml up -d

Et déployer zabbix-agent

docker compose -f docker-compose\_v3\_alpine\_mysql\_local.yaml build zabbix-build-base zabbix-build-mysql zabbix-agent

Regarder cette vidéo pour l’installation de zabbix

[Monitor Docker Containers with Zabbix - Easy Setup and Configuration Guide (youtube.com)](https://www.youtube.com/watch?v=QNdsWp_X9-c)

**Installation de prometheus-grafana**

[**Qu’est-ce que Prométhée ? | Documentation Grafana**](https://grafana.com/docs/grafana/latest/fundamentals/intro-to-prometheus/)

*Grafana : surveillance, observabilité*

*Le logiciel open source (OSS) de Grafana* vous permet d’interroger, de visualiser, d’alerter et d’explorer vos métriques, journaux et traces, où qu’ils soient stockés. Les plug-ins de source de données Grafana vous permettent d’interroger des sources de données, **notamment des bases de données de séries chronologiques comme Prometheus** et CloudWatch, des outils de journalisation comme Loki et Elasticsearch, des bases de données NoSQL/SQL comme Postgres, des outils CI/CD comme GitHub, et bien d’autres. Grafana OSS vous fournit des outils pour afficher ces données sur des tableaux de bord en direct avec des graphiques et des visualisations perspicaces.

Prometheus n’est pas seulement un format de données ; Il est également considéré comme une [boîte à outils de surveillance et d’alerte des systèmes open source](https://prometheus.io/docs/introduction/overview/). C’est parce que Prometheus est un logiciel, pas seulement des données.

Les [trois piliers de l’observabilité sont les](https://www.oreilly.com/library/view/distributed-systems-observability/9781492033431/ch04.html) métriques, les logs et les traces. Prometheus prend en charge le pilier des métriques. Lorsque le logiciel d’un ordinateur s’exécute lentement, l’observabilité peut vous aider à déterminer si le processeur est saturé, si le système n’a plus de mémoire ou si le disque écrit à la vitesse maximale, ce qui vous permet de réagir de manière proactive.

**Prometheus : monitoring**

Pour obtenir des métriques dans Grafana, vous pouvez utiliser le logiciel Prometheus ou [Grafana Alloy](https://grafana.com/docs/alloy/latest/) pour extraire les métriques. Grafana Alloy collecte et transmet les données de télémétrie aux déploiements open source de Grafana Stack, Grafana Cloud ou Grafana Enterprise, où vos données peuvent être analysées. Par exemple, vous pouvez configurer Grafana Alloy pour extraire les données de « MyApp » toutes les cinq secondes et envoyer les résultats à Grafana Cloud.

Les données de métriques ne sont qu’un type de données de télémétrie ; Les autres types sont les journaux et les traces. L’utilisation de Grafana Alloy peut être une excellente option pour envoyer des données de télémétrie, car à mesure que vous mettez à l’échelle vos pratiques d’observabilité pour inclure des journaux et des traces, que Grafana Alloy prend également en charge, vous disposez déjà d’une solution en place.

**Prometheus to minitor it self**

<https://prometheus.io/docs/prometheus/latest/getting_started/>

prometheus utilise le fichier de configuration prometheus.yml pour configurer le service (serveur ou autre) à surveiller.

Note : scrapes = égratignures (problème, non fonctionnement)

Voici une configuration de monitoring sur prometheus sur lui même

global:

scrape\_interval: 15s

# By default, scrape targets every 15 seconds.

# Attach these labels to any time series or alerts when communicating with

# external systems (federation, remote storage, Alertmanager).

external\_labels:

monitor: 'codelab-monitor'

# A scrape configuration containing exactly one endpoint to scrape:

# Here it's Prometheus itself.

scrape\_configs:

# The job name is added as a label `job=<job\_name>` to any timeseries scraped from this config.

- job\_name: 'prometheus'

# Override the global default and scrape targets from this job every 5 seconds.

scrape\_interval: 5s

static\_configs:

- targets: ['localhost:9090']

Une autre configuration de monitoring sur prometheus

global:

scrape\_interval: 15s

scrape\_timeout: 10s

evaluation\_interval: 15s

alerting:

alertmanagers:

- static\_configs:

- targets: []

scheme: http

timeout: 10s

api\_version: v1

scrape\_configs:

- job\_name: prometheus

honor\_timestamps: true

scrape\_interval: 15s

scrape\_timeout: 10s

metrics\_path: /metrics

scheme: http

static\_configs:

- targets:

- localhost:9090

Ce lien montre les metrics prometheus collectés et disponibles

<http://localhost:9090/metrics>

penser à vérifier les métrics récupérer par node\_exporter

<https://www.stackhero.io/fr-fr/services/Prometheus/documentations/Utilisation-de-Node-Exporter>

Comment récupérer les métriques de serveurs Linux dans Prometheus en utilisant Node Exporter

Pour résumer, Node Exporter est ce que vous devez installer si vous souhaitez monitorer un système basé sur Linux.

Pour les systèmes sous Windows, vous pouvez utiliser Windows Exporter.

Utilisez le navigateur d’expressions intégré de Prometheus, accédez à <http://localhost:9090/graph> et choisissez la vue « Table » dans l’onglet « Graphique ».

**Le projet prometheus-grafana2**

**Assurer que les conteneurs tournent sur le même réseau que prestashop (diamond\_network)**

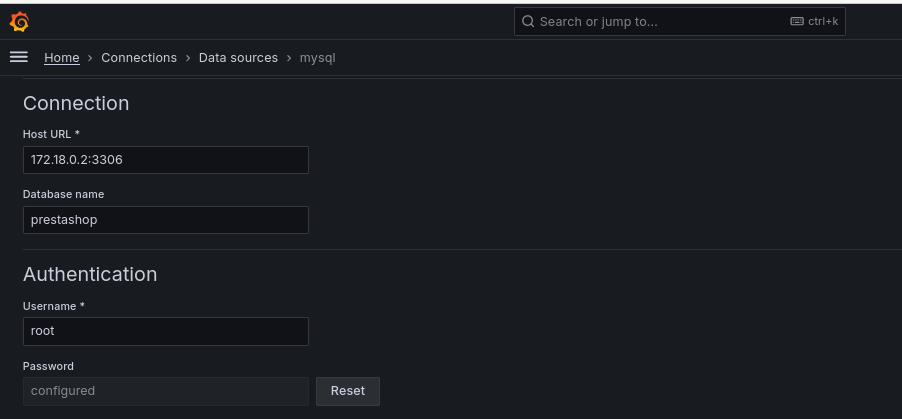
Testé et fonctionne avec les metrics et les datasource prestashop et prometheus

Configurer le fichier prometheus.yml

Configurer la datasource avec l’adresse IP du conteneur prestashop-mysql 172.10.0.2 :3306 et compte root / ldiashop

Compte grafana <http://localhost:3000>

admin/admin ou passe=grafana



**Tuto à consulter**

[**Building a Monitoring Stack with Prometheus, Grafana, and Alerting: A Docker Compose | by Ravi Patel | Medium**](https://medium.com/@ravipatel.it/building-a-monitoring-stack-with-prometheus-grafana-and-alerting-a-docker-compose-ef78127e4a19)

…./projets/prometheus-grafana

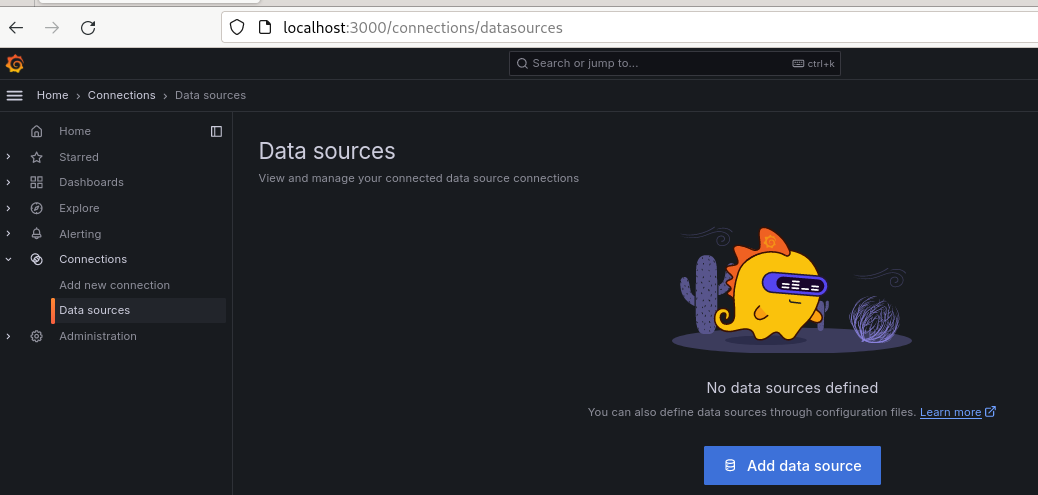
Vérifier que les conteneurs sont up et tout est OK

Aller sur l’interface de grafana : http://localhost:3000

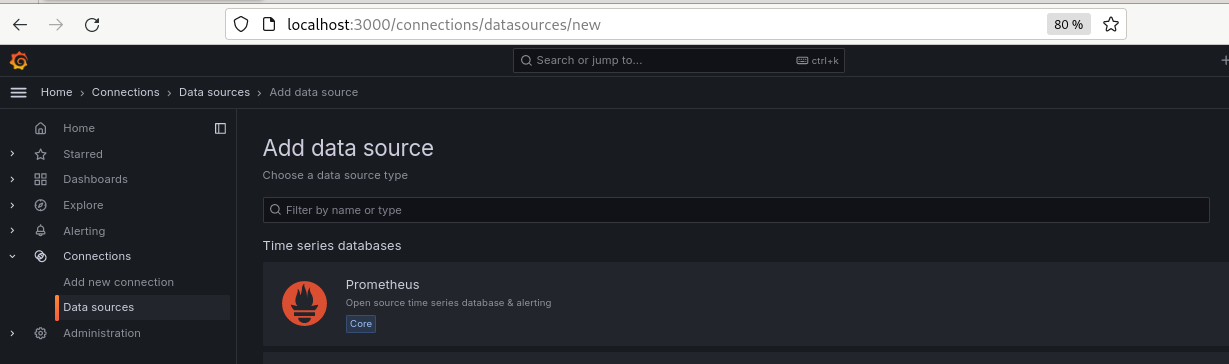
compte grafana admin/admin

**Configure Grafana Data Source**

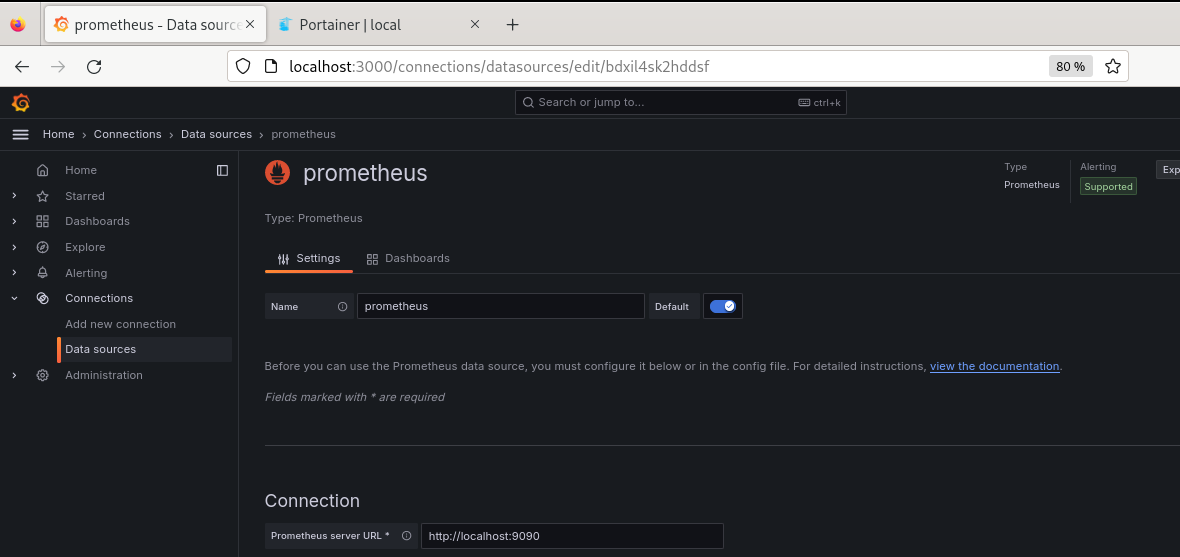
* Aller dans Grafana “Connections” > “Data Sources” > “Add Data Source”



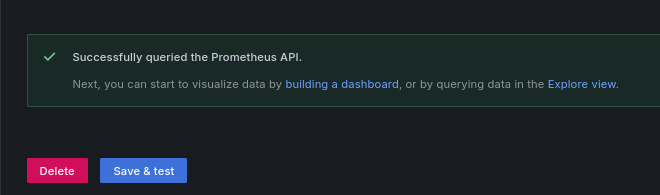
* Sélectionner “Prometheus” comme type



* Set the URL to <http://prometheus:9090>

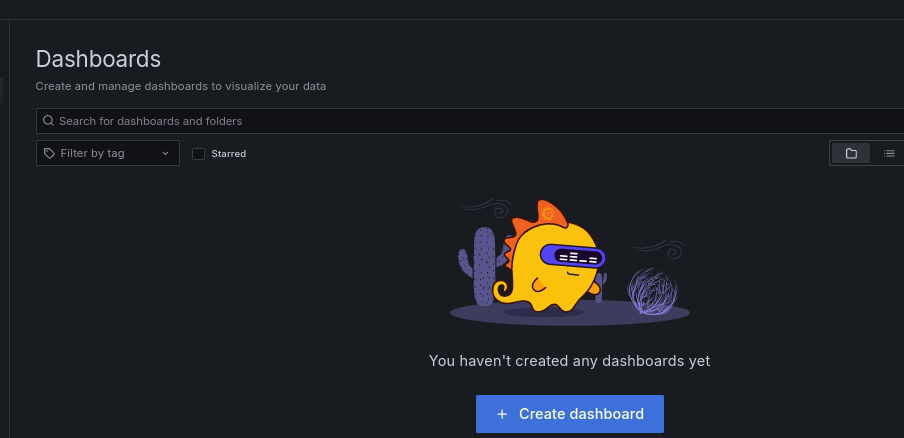


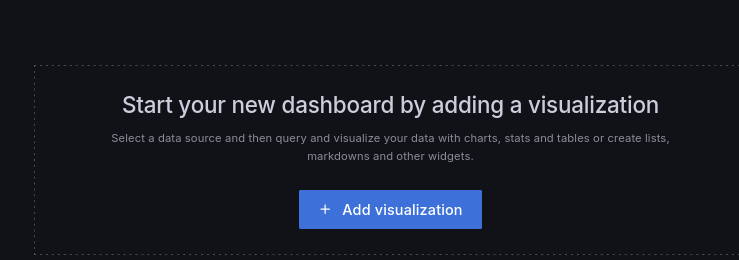
* Click “Save & Test” to validate the connection.

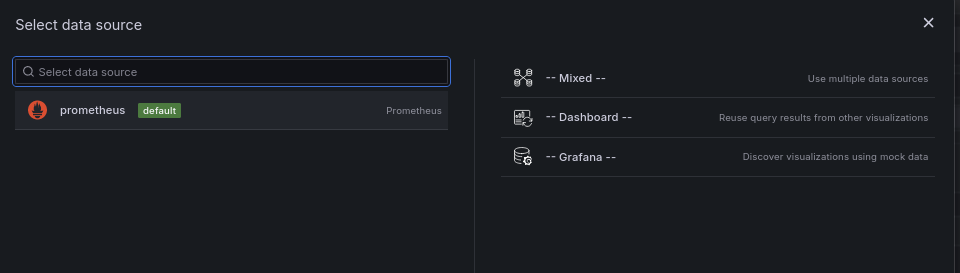


Create a Dashboard:

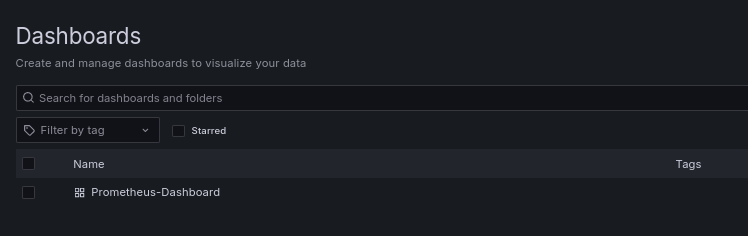
* Establish a new dashboard and incorporate panels to visualize metrics







* Utilize Prometheus queries to retrieve pertinent metrics such as request count, response time, and server status
* Customize the dashboard layout and visualizations to meet specific preferences.



Apply et sauve la configuration.

**Configure Alerting in Prometheus:**

Configuring alerting in Prometheus involves defining alerting rules using the Prometheus Alerting Rule Language (PromQL), specifying **alertmanager.yml**configuration for routing and managing alerts, and restarting Prometheus to apply the changes. Below are the step-by-step instructions with example code:

**Step 1: Define Alerting Rules**

Create a file named alerts.rules.yml with your alerting rules. Here's an example:

groups:  
- name: example\_alerts  
 rules:  
 - alert: HighRequestRate  
 expr: rate(http\_requests\_total{job="web\_server"}[5m]) > 100  
 for: 1m  
 labels:  
 severity: critical  
 annotations:  
 summary: High request rate on web server  
 description: '{{ $labels.instance }} has high request rate (>100 req/min) for the last 5 minutes.'  
 - alert: HighCPULoad  
 expr: node\_load1 > 0.8  
 for: 5m  
 labels:  
 severity: warning  
 annotations:  
 summary: High CPU load detected  
 description: '{{ $labels.instance }} has CPU load >0.8 for the last 5 minutes.'

Faire backup de volume

docker run --rm -v my\_volume:/volume -v $(pwd):/backup alpine \

tar cvf /backup/backup.tar /volume

Faire restaure de volume

docker volume create my\_volume

docker run --rm -v my\_volume:/volume -v $(pwd):/backup alpine \

tar xvf /backup/backup.tar -C /

tuto pour prometheus-grafana2

: [awesome-compose/prometheus-grafana/README.md at master · docker/awesome-compose (github.com)](https://github.com/docker/awesome-compose/blob/master/prometheus-grafana/README.md)

=========================================================================

Pour accéder à l’interface zabbix-web changer cette configuration dans le conteneur

cat /etc/apache2/httpd.conf

Chercher le Directory suivant et faites le changement,

# Deny access to the entirety of your server's filesystem. You must

# explicitly permit access to web content directories in other

# <Directory> blocks below.

#

<Directory />

AllowOverride none

Require all denied

</Directory>

Remplacer denied par granted

<Directory />

AllowOverride none

Require all granted

</Directory>

Ajouter le chemin suivant dans le fichier

/etc/nginx/http.d/nginx.conf

Webroot=’ /usr/share/zabbix/ui’

Sauvegarder et redémarrer le conteneur zabbix-web

Suivre ce tuto pour comprendre les agents zabbix

<https://www.digitalocean.com/community/tutorials/how-to-monitor-docker-using-zabbix-on-ubuntu-20-04>

Ce tuto met en place un zabbix agent

<https://www.heyvaldemar.com/install-zabbix-using-docker-compose/>

Ce tuto installe zabbix et zabbix agent

<https://medium.com/@fredrik.maxfield/simplified-zabbix-deployment-step-by-step-with-docker-and-portainer-19e85c08a65b>

Résoudre un probleme de certificat auto signé

<https://aboutssl.org/how-to-fix-mozilla-pkix-self-signed-cert-error/>

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Compléter le dossier professionnel | 19/09/2024 | | Finaliser l’installation des outils et l’automatisation | 23/09/2024 | | L’installation sur les VM Cloud AWS | 25/09/2024 | | Compléter le dossier projet et annexes | 26/09/2024 | |

Le dossier projet accompagnera les autres actions, au fur et à mesure il sera rempli.

Compléter les plans de visualisation sur grafana

**28/09/2024**

Recherche des solutions, à intégrer dans le rapport de stage

**Créer un agent node automatiquement**

1. **Récupérer jenkins API Token manuellement**

Générer ce token en allant dans :

* Jenkins > Manage Jenkins > Manage Users > <ton utilisateur> > API Token > Add new Token

Script **get-jenkins-secret.sh**

#!/bin/bash

# Variables à configurer

JENKINS\_URL="http://<adresse-du-serveur-jenkins>:8080"

JENKINS\_USER="<ton-utilisateur>"

JENKINS\_API\_TOKEN="<ton-token-api>"

AGENT\_NAME="jenkins-agent-node1"

AGENT\_WORKDIR="/home/jenkins/agent"

AGENT\_LABELS="docker linux"

# Créer l'agent via l'API Jenkins

create\_agent\_response=$(curl -s -X POST "${JENKINS\_URL}/computer/doCreateItem?name=${AGENT\_NAME}" \

--user "${JENKINS\_USER}:${JENKINS\_API\_TOKEN}" \

-H "Content-Type:application/x-www-form-urlencoded" \

--data "type=hudson.slaves.DumbSlave&name=${AGENT\_NAME}&nodeDescription=&numExecutors=1&remoteFS=${AGENT\_WORKDIR}&labelString=${AGENT\_LABELS}&mode=EXCLUSIVE&retentionStrategy=Always&launcher=hudson.slaves.JNLPLauncher")

# Extraire le JENKINS\_SECRET de la réponse

JENKINS\_SECRET=$(curl -s "${JENKINS\_URL}/computer/${AGENT\_NAME}/jenkins-agent.jnlp" \

--user "${JENKINS\_USER}:${JENKINS\_API\_TOKEN}" | grep -oP '(?<=<argument>)[^<]+')

# Vérification si le secret a été récupéré

if [ -z "$JENKINS\_SECRET" ]; then

echo "Erreur: Impossible de récupérer le JENKINS\_SECRET."

exit 1

else

echo "Le JENKINS\_SECRET pour l'agent '${AGENT\_NAME}' est : $JENKINS\_SECRET"

fi

# Lancer l'agent Jenkins avec Docker

docker run -d \

--name ${AGENT\_NAME} \

-e JENKINS\_URL=${JENKINS\_URL} \

-e JENKINS\_SECRET=${JENKINS\_SECRET} \

-e JENKINS\_AGENT\_NAME=${AGENT\_NAME} \

-e JENKINS\_AGENT\_WORKDIR=${AGENT\_WORKDIR} \

-e JENKINS\_LABELS="${AGENT\_LABELS}" \

-v /var/run/docker.sock:/var/run/docker.sock \

jenkins/inbound-agent

1. **Récupérer jenkins API Token automatiquement**

#!/bin/bash

# Variables

JENKINS\_URL="http://localhost:8080"

JENKINS\_USER="jenkins"

JENKINS\_PASSWORD="jenkins"

NEW\_TOKEN\_NAME="diamond-jenkins-token"

AGENT\_NAME="jenkins-agent-node1"

AGENT\_WORKDIR="/home/jenkins/agent"

AGENT\_LABELS="docker linux"

# Par sécurité, débloquer la protection CSRF en récupérant le jenkins CRUMB

JENKINS\_CRUMB=$(curl -u "jenkins:jenkins" -s 'http://localhost:8080/crumbIssuer/api/json' | jq -r .crumb)

echo "JENKINS\_CRUMB : $JENKINS\_CRUMB"

# Requête pour générer un nouveau token

RESPONSE=$(curl -s -X POST "$JENKINS\_URL/me/descriptorByName/jenkins.security.ApiTokenProperty/generateNewToken" \

--user "$JENKINS\_USER:$JENKINS\_PASSWORD" \

--data "newTokenName=$NEW\_TOKEN\_NAME" \

#-H "Content-Type: application/x-www-form-urlencoded")

-H "Jenkins-Crumb:$JENKINS\_CRUMB"

# Extraire le nouveau jeton de la réponse JSON

echo "RESPONSE= $RESPONSE"

JENKINS\_API\_TOKEN =$(echo $RESPONSE | grep -o '"tokenValue":"[^"]\*' | grep -o '[^"]\*$')

# Afficher le nouveau token

if [ -n "$JENKINS\_API\_TOKEN" ]; then

echo "Nouveau jeton API généré : $JENKINS\_API\_TOKEN"

else

echo "Échec de la génération du jeton API"

exit 1

fi

# Créer l'agent via l'API Jenkins

create\_agent\_response=$(curl -s -X POST "${JENKINS\_URL}/computer/doCreateItem?name=${AGENT\_NAME}" \

--user "${JENKINS\_USER}:${JENKINS\_API\_TOKEN}" \

-H "Content-Type:application/x-www-form-urlencoded" \

--data "type=hudson.slaves.DumbSlave&name=${AGENT\_NAME}&nodeDescription=&numExecutors=1&remoteFS=${AGENT\_WORKDIR}&labelString=${AGENT\_LABELS}&mode=EXCLUSIVE&retentionStrategy=Always&launcher=hudson.slaves.JNLPLauncher")

# Extraire le JENKINS\_SECRET de la réponse

JENKINS\_SECRET=$(curl -s "${JENKINS\_URL}/computer/${AGENT\_NAME}/jenkins-agent.jnlp" \

--user "${JENKINS\_USER}:${JENKINS\_API\_TOKEN}" | grep -oP '(?<=<argument>)[^<]+')

# Vérification si le secret a été récupéré

if [ -z "$JENKINS\_SECRET" ]; then

echo "Erreur: Impossible de récupérer le JENKINS\_SECRET."

exit 1

else

echo "Le JENKINS\_SECRET pour l'agent '${AGENT\_NAME}' est : $JENKINS\_SECRET"

fi

**Déclencher un job automatiquement**

curl -u <username>:<password> \

-H "Jenkins-Crumb:$CRUMB" \

-X POST [http://localhost:8080/job/Test\_Zodiac\_ReactJS/build](http://localhost:8080/job//build)

Déclencher un pipeline automatiquement

jenkins lancer pipeline si github projet modifé

**ChatGPT a dit :**

ChatGPT

Pour configurer un pipeline Jenkins afin de détecter automatiquement les modifications dans un projet GitHub et déclencher une exécution du pipeline (CI/CD), vous pouvez suivre ces étapes :

**1. Configurer Jenkins :**

Assurez-vous d'avoir Jenkins installé et que les plugins nécessaires sont configurés :

* **Git Plugin** : pour interagir avec des dépôts Git.
* **GitHub Plugin** : pour intégrer GitHub et Jenkins.
* **Pipeline Plugin** : pour créer des pipelines CI/CD.

**2. Créer un Webhook dans GitHub :**

1. Accédez à votre dépôt GitHub.
2. Allez dans **Settings** > **Webhooks**.
3. Cliquez sur **Add webhook**.
4. Dans **Payload URL**, saisissez l'URL de votre Jenkins : http://<votre-serveur-jenkins>/github-webhook/.
5. Sélectionnez **Content type** comme application/json.
6. Sous **Which events would you like to trigger this webhook?**, choisissez **Just the push event** pour déclencher le pipeline lors de chaque commit.

**3. Créer un Pipeline dans Jenkins :**

1. Connectez-vous à Jenkins.
2. Allez à **New Item** et sélectionnez **Pipeline**.
3. Configurez le projet en utilisant les paramètres suivants :
   * **GitHub Project** : Entrez l'URL de votre projet GitHub.
   * **Source Code Management (SCM)** : Sélectionnez **Git** et fournissez l'URL du dépôt GitHub.
   * **Branch to build** : Sélectionnez la branche à surveiller, par exemple \*/main.

**4. Définir le Jenkinsfile :**

Le fichier Jenkinsfile dans le dépôt GitHub contient le script qui définit le pipeline. Voici un exemple de Jenkinsfile de base pour un projet simple :

groovy

Copier le code

pipeline {

agent any

stages {

stage('Checkout') {

steps {

// Récupérer le code du dépôt GitHub

git url: 'https://github.com/votre-utilisateur/votre-depot.git', branch: 'main'

}

}

stage('Build') {

steps {

// Par exemple, pour un projet Java Maven

sh 'mvn clean install'

}

}

stage('Test') {

steps {

// Exécuter les tests

sh 'mvn test'

}

}

stage('Deploy') {

steps {

// Déployer ou toute autre action post-build

echo 'Déploiement en cours...'

}

}

}

}

**5. Configurer le déclencheur dans Jenkins :**

1. Allez dans les paramètres du pipeline.
2. Sous **Build Triggers**, cochez la case **GitHub hook trigger for GITScm polling**.

**6. Tester :**

Faites un commit ou un push dans le dépôt GitHub. Le webhook devrait envoyer une notification à Jenkins, qui déclenchera automatiquement l'exécution du pipeline.

**Résultat :**

Chaque fois qu'une modification est poussée dans votre projet GitHub, Jenkins récupérera les changements et exécutera automatiquement le pipeline défini dans le Jenkinsfile.

Si vous avez besoin d'une configuration plus complexe ou d'une intégration spécifique, n'hésitez pas à préciser.

**Dimanche 30/09/2024**

**Suivre l’exécution d’un script en arrière plan**

nohup ./start.sh &

tail -f nohup.out

**Installer Minikube**

[**Installing Minikube on Linux. Install Minikube on Linux , Ubuntu, and… | by Suresh yadav | Cypik | Medium**](https://medium.com/cypik/installing-minikube-on-ubuntu-22-04-lts-77f5abaf3d39)

Les traces de cette installation ne se voit pas après la commande newgrp docker

Rediriger les logs de l’exécution du script dans un fichier out.log

./start.sh > start.log 2>&1

**1.Installing Docker**

sudo apt-get install docker.io -y  
sudo usermod -aG docker $USER && newgrp docker

**2.Updating system packages and installing Minikube dependencies**

sudo apt update & sudo apt upgrade   
sudo apt install -y curl wget apt-transport-https

**3. Installing Minikube**

Use the following curl command to download the latest Minikube binary:

sudo curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

Once the binary is downloaded, copy it to the path /usr/local/bin and set the executable permissions on it

sudo install minikube-linux-amd64 /usr/local/bin/minikube

Verify the minikube version

minikube version  
 #output  
minikube version: v1.32.0  
commit: 8220a6eb95f0a4d75f7f2d7b14cef975f050512d

Note: At the time of writing this tutorial, the latest version of minikube was v1.32.0

**4. Installing kubectl utility**

**kubectl** is a command line used to interact with Kubernetes cluster. It is used for managing deployments, replica sets, services, etc. Use the following command to download the latest version of kubectl.

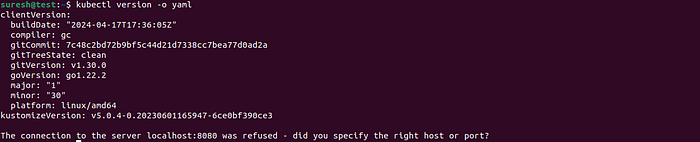
sudo curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

Once Kubectl is downloaded, set the executable permissions on the Kubectl binary and move it to the path /usr/local/bin.

chmod +x kubectl  
sudo mv kubectl /usr/local/bin/

Now verify the kubectl version

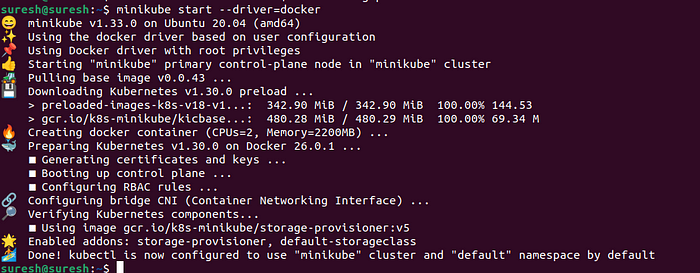
kubectl version -o yaml



5.**Start minikube**

As we stated in the beginning, we would be using docker as base for minikue, so start Minikue with the docker driver and run

minikube start --driver=docker



Perfect! The above picture confirms that the Minikube cluster has been confluenced and started successfully.

**6. Verifying Installation**

Run the below minikube command to check the status.

minikube status  
 #output  
minikube  
type: Control Plane  
host: Running  
kubelet: Running  
apiserver: Running  
kubeconfig: Configured

Run the following kubectl command to verify the Kubernetes version, node status, and cluster info.

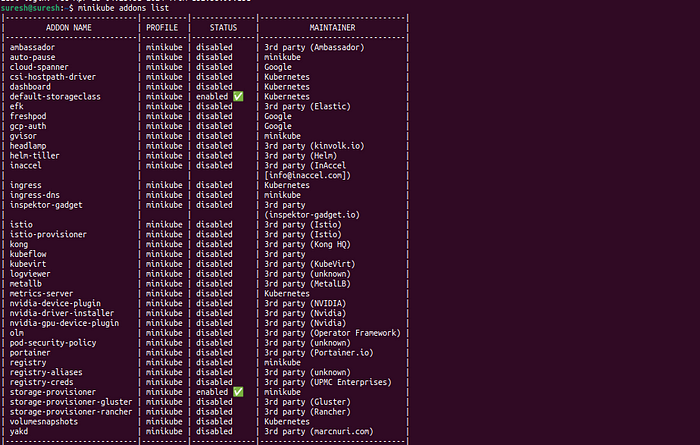
kubectl cluster-info  
  
Kubernetes control plane is running at **https://192.168.49.2:8443**  
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy  
  
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

kubectl get nodes  
NAME STATUS ROLES AGE VERSION  
minikube Ready control-plane 3m27s v1.30.0

**7. Managing Addons on minikube**

By default, only a couple of addons are enabled during minikube installation. To see the addons in Minikube, run the below command.

minikube addons list



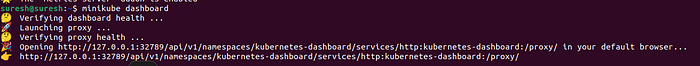
If you wish to enable any addons, run the below minikube command.

minikube addons enable metrics-server

Let’s assume we want to enable and access the Kubernetes dashboard. Run

minikube dashboard

Use localhost output



Server use this commands

minikube dashboard --url



Follow next order

Run the command in background

kubectl proxy --address='0.0.0.0' --disable-filter=true &  
 #output   
W0423 04:59:24.539492 15291 proxy.go:177] Request filter disabled, your proxy is vulnerable to XSRF attacks, please be cautious  
Starting to serve on [::]:8001

Follow next order

Use server IP and port [::]:8001 and use **url /api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/**

http://server\_ip:8001/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/workloads?namespace=default

**Lundi 30/09/2024**

**Installation de Openproject**

https://www.openproject.org/docs/installation-and-operations/installation/docker/#:~:text=Overview.%20OpenProject%E2%80%99s%20docker%20setup%20can%20be

Récupérer le projet git

Dans le fichier .env

**Garder que les paramètres**

**OPENPROJECT\_HOST\_\_NAME**=localhost:8080

**PORT**=8080

**Note si besoin**: Make sure to replace secret with a random string. One way to generate one is to run head /dev/urandom | tr -dc A-Za-z0-9 | head -c 32 ; echo '' if you are on Linux.

**Déploiement par docker-compose.yml**

# Start OpenProject with docker compose

OPENPROJECT\_HTTPS=false docker-compose up -d --build --pull always

**Déploiement par docker run**

docker run -d -p 8080:80 --name openproject \

-e OPENPROJECT\_HOST\_\_NAME=localhost :8080 \

-e OPENPROJECT\_SECRET\_KEY\_BASE=L5475IONSh3pdKBBdwPTgNUsSB9nFa9q \

-v ./volumes/var/lib/openproject/pgdata:/var/openproject/pgdata \

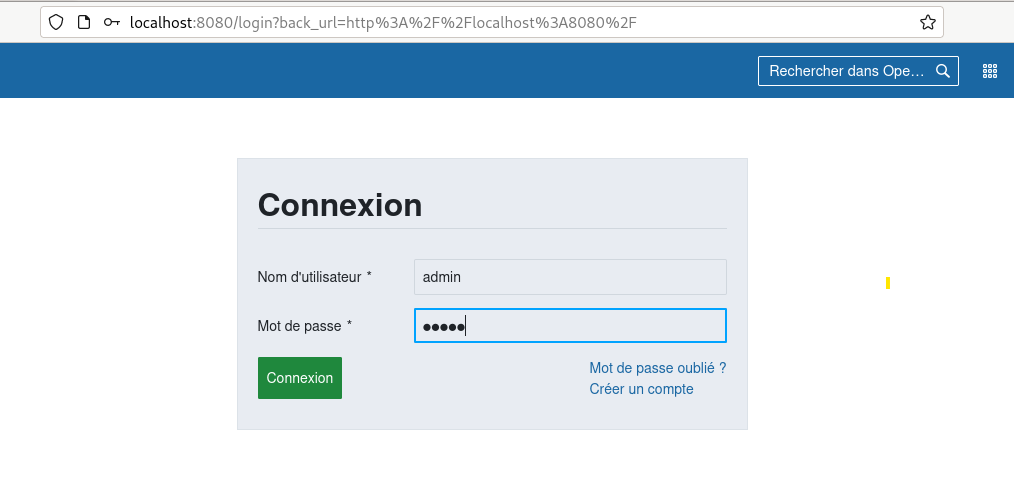
-v ./volumes/var/lib/openproject/assets:/var/openproject/assets \

openproject/openproject:14

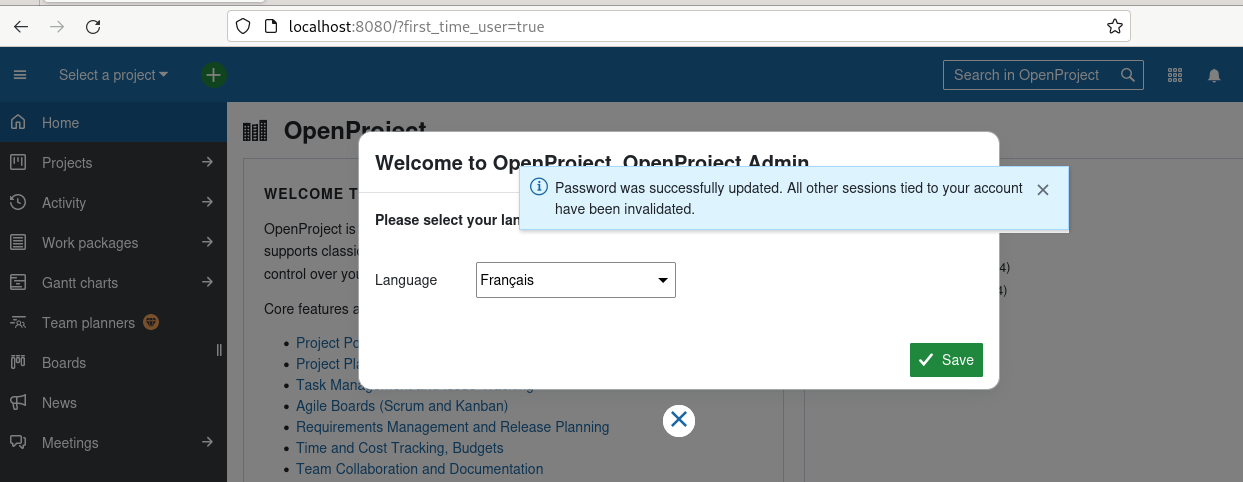
**Se connecter**

localhost:8080

Compte par défaut admin/admin



Il faut changer le mot de passe après la première connexion 🡺 openproject2024



**AWS différence entre ec2-instance-connect et cloudshell**

**ChatGPT a dit :**

ChatGPT

Amazon Web Services (AWS) propose plusieurs moyens d'interagir avec des ressources en ligne, dont **EC2 Instance Connect** et **CloudShell**. Bien que les deux permettent l'accès aux environnements en ligne via des interfaces en ligne de commande, ils sont conçus pour des usages différents. Voici un comparatif de ces deux services :

**1. EC2 Instance Connect**

* **Objectif** : EC2 Instance Connect permet de se connecter directement à des instances Amazon EC2 (machines virtuelles) via SSH, sans avoir besoin d'une clé SSH pré-configurée sur l'instance.
* **Utilisation** :
  + Il s'agit d'une méthode simple pour se connecter à une instance EC2 depuis la console AWS, sans avoir à gérer manuellement des clés SSH.
  + Fonctionne via la console AWS ou en utilisant le client EC2 Instance Connect.
  + Utile pour se connecter à une instance en SSH quand il n'y a pas de clé SSH partagée ou quand l'utilisateur n'a pas accès direct à la clé privée.
* **Sécurité** :
  + Le service génère temporairement une clé publique qui est envoyée à l'instance EC2 pour authentification.
  + Ce mécanisme est sécurisé par IAM (Identity and Access Management) et limite les accès non autorisés.
  + Ne fonctionne que pour les instances Amazon Linux 2, Ubuntu, et certaines distributions avec EC2 Instance Connect installé.
* **Limites** :
  + Ce n'est pas une solution universelle, car toutes les instances ne prennent pas en charge EC2 Instance Connect. Vous devez vous assurer que la configuration SSH sur l'instance est compatible.

**2. AWS CloudShell**

* **Objectif** : AWS CloudShell est un environnement shell basé sur un navigateur qui permet d'exécuter des commandes AWS sans avoir à configurer de machine virtuelle ou de client SSH.
* **Utilisation** :
  + Il offre un accès direct à un terminal préconfiguré dans le navigateur, avec les outils AWS CLI, Git, et autres outils de gestion déjà installés.
  + Idéal pour exécuter des scripts, interagir avec des services AWS (comme S3, Lambda, EC2) ou tester des commandes rapidement sans devoir configurer un environnement local.
  + Chaque utilisateur reçoit un stockage persistant de 1 Go pour les fichiers et les scripts.
  + L'utilisateur n'a pas besoin de configurer une instance EC2 ou de maintenir une infrastructure.
* **Sécurité** :
  + Il utilise les mêmes permissions IAM que celles assignées à votre utilisateur pour déterminer l'accès aux ressources AWS.
  + Chaque session est temporaire et les fichiers téléchargés dans l'environnement de travail sont stockés de manière persistante sur 1 Go d'espace alloué.
* **Limites** :
  + AWS CloudShell ne permet pas d'accéder directement à des instances EC2 comme SSH. Il est destiné à l'exécution de commandes sur l'infrastructure AWS, pas à la gestion d'instances spécifiques.
  + Limité à certaines régions AWS.

**Résumé des différences :**

| **Caractéristique** | **EC2 Instance Connect** | **AWS CloudShell** |
| --- | --- | --- |
| **Objectif** | Connexion SSH aux instances EC2 | Accès terminal à AWS CLI et outils de gestion via un navigateur |
| **Accès** | Connecte uniquement aux instances EC2 compatibles | Accède à toute l'infrastructure AWS via CLI |
| **Sécurité** | Utilise des clés SSH temporaires et IAM | Utilise IAM pour accéder aux ressources AWS |
| **Configuration requise** | Nécessite une instance EC2 compatible avec EC2 Connect | Aucune configuration, environnement prêt à l’emploi |
| **Stockage** | N/A | 1 Go d’espace persistant pour chaque utilisateur |
| **Limite d’utilisation** | Utilisable uniquement pour les connexions SSH EC2 | Limité à des commandes et scripts AWS dans le navigateur |

**Quand utiliser :**

* **EC2 Instance Connect** est utile lorsque vous avez besoin d'accéder directement à une instance EC2 via SSH, surtout si vous n'avez pas de clé SSH à portée de main.
* **AWS CloudShell** est idéal pour gérer vos services AWS via la CLI sans avoir à configurer une machine locale ou une instance EC2.

**Sauvegarder conteneur docker avec volumes et network**

ChatGPT

Pour sauvegarder un conteneur Docker avec ses volumes et son réseau, vous devez effectuer plusieurs étapes, incluant la sauvegarde de l'image du conteneur, des volumes associés et des informations de réseau. Voici un guide complet pour sauvegarder et restaurer un conteneur Docker avec ses volumes et son réseau.

**Sauvegarde du conteneur**

**1. Sauvegarder l'image du conteneur**

1. **Identifier l'image du conteneur** :

sh

Copier le code

container\_name=<container\_name\_or\_id>

image\_name=$(docker inspect --format '{{.Config.Image}}' $container\_name)

1. **Sauvegarder l'image du conteneur** :

sh

Copier le code

docker save -o ${image\_name//[:\/]/\_}.tar $image\_name

**2. Sauvegarder les volumes associés**

1. **Identifier les volumes associés au conteneur** :

sh

Copier le code

volumes=$(docker inspect --format '{{ range .Mounts }}{{ if eq .Type "volume" }}{{ .Name }} {{ end }}{{ end }}' $container\_name)

1. **Sauvegarder chaque volume** :

sh

Copier le code

for volume in $volumes; do

docker run --rm -v $volume:/volume -v $(pwd):/backup alpine tar czf /backup/${volume}.tar.gz -C /volume .

done

**3. Sauvegarder les informations réseau**

1. **Identifier le réseau associé au conteneur** :

sh

Copier le code

network=$(docker inspect --format '{{range $k,$v := .NetworkSettings.Networks}}{{$k}}{{end}}' $container\_name)

1. **Sauvegarder les informations réseau** :

sh

Copier le code

echo $network > network\_info.txt

**Script de sauvegarde complet**

sh

Copier le code

#!/bin/bash

container\_name=$1

if [ -z "$container\_name" ]; then

echo "Usage: $0 <container\_name\_or\_id>"

exit 1

fi

# Sauvegarder l'image du conteneur

image\_name=$(docker inspect --format '{{.Config.Image}}' $container\_name)

docker save -o ${image\_name//[:\/]/\_}.tar $image\_name

echo "Image sauvegardée : ${image\_name//[:\/]/\_}.tar"

# Sauvegarder les volumes associés

volumes=$(docker inspect --format '{{ range .Mounts }}{{ if eq .Type "volume" }}{{ .Name }} {{ end }}{{ end }}' $container\_name)

for volume in $volumes; do

docker run --rm -v $volume:/volume -v $(pwd):/backup alpine tar czf /backup/${volume}.tar.gz -C /volume .

echo "Volume sauvegardé : ${volume}.tar.gz"

done

# Sauvegarder les informations réseau

network=$(docker inspect --format '{{range $k,$v := .NetworkSettings.Networks}}{{$k}}{{end}}' $container\_name)

echo $network > network\_info.txt

echo "Informations réseau sauvegardées : network\_info.txt"

**Restauration du conteneur**

**1. Restaurer l'image du conteneur**

1. **Charger l'image du conteneur** :

sh

Copier le code

docker load -i ${image\_name//[:\/]/\_}.tar

**2. Restaurer les volumes associés**

1. **Restaurer chaque volume** :

sh

Copier le code

for volume in $volumes; do

docker volume create $volume

docker run --rm -v $volume:/volume -v $(pwd):/backup alpine sh -c "cd /volume && tar xzvf /backup/${volume}.tar.gz --strip 1"

done

**3. Restaurer le réseau et redéployer le conteneur**

1. **Lire les informations réseau** :

sh

Copier le code

network=$(cat network\_info.txt)

1. **Redéployer le conteneur** :

sh

Copier le code

docker run -d --name $container\_name --network $network $(for volume in $volumes; do echo -n "--mount source=$volume,target=/path\_in\_container "; done) $image\_name

**Script de restauration complet**

sh

Copier le code

#!/bin/bash

container\_name=$1

if [ -z "$container\_name" ]; then

echo "Usage: $0 <container\_name\_or\_id>"

exit 1

fi

# Restaurer l'image du conteneur

image\_name=$(docker load -i ${image\_name//[:\/]/\_}.tar | awk '/Loaded image: /{print $3}')

echo "Image restaurée : $image\_name"

# Restaurer les volumes associés

volumes=$(ls \*.tar.gz | sed 's/\.tar\.gz//')

for volume in $volumes; do

docker volume create $volume

docker run --rm -v $volume:/volume -v $(pwd):/backup alpine sh -c "cd /volume && tar xzvf /backup/${volume}.tar.gz --strip 1"

echo "Volume restauré : $volume"

done

# Restaurer le réseau et redéployer le conteneur

network=$(cat network\_info.txt)

docker run -d --name $container\_name --network $network $(for volume in $volumes; do echo -n "--mount source=$volume,target=/path\_in\_container "; done) $image\_name

echo "Conteneur restauré : $container\_name"

**Exécution des scripts**

1. Sauvegarder un conteneur :

sh

Copier le code

chmod +x backup\_container.sh

./backup\_container.sh <container\_name\_or\_id>

1. Restaurer un conteneur :

sh

Copier le code

chmod +x restore\_container.sh

./restore\_container.sh <container\_name>

**Notes**

* **Permissions** : Assurez-vous que l'utilisateur exécutant les scripts a les permissions nécessaires pour lire/écrire dans le répertoire de sauvegarde et exécuter des commandes Docker.
* **Compatibilité des chemins** : Les chemins des montages (/path\_in\_container) doivent correspondre à ceux utilisés dans le conteneur original.
* **Docker Desktop** : Si vous utilisez Docker Desktop (Windows/macOS), les chemins de fichiers peuvent nécessiter des ajustements pour fonctionner correctement avec le système de fichiers de l'hôte.

Ces scripts fournissent une méthode complète pour sauvegarder et restaurer un conteneur Docker avec ses volumes et son réseau, facilitant ainsi la gestion des environnements Docker.

**Vacances 2024**

[Camping familial proche des Sables d’Olonne - Camping le Pavillon](https://www.campinglepavillon.com/)

[contact@campinglepavillon.com](mailto:contact@campinglepavillon.com)

Nom

Prénom

Date de naissance

Date du séjour, location identifiée

Numéro vacaf

WIZLABS

S’inscrire via le mail de la journée Cloud, scan le QR code et remplir le formulaire

Nouveau compte :

[sdia1950@gmail.com](mailto:sdia1950@gmail.com)

bolo/bolo/

<https://www.linkedin.com/in/khady-dia/>

github pat= github\_pat\_11BHX2UZA0qgUcJK7lGReZ\_kIHaMTGvuXO8lxEWLgKYQjOuvPUtnOzFXN3fN6hD07h7JBTV7IMjYOBPBUY