Pour notre tp on va utiliser Jenkins pour automatiser la configuration d'un environnement virtuel Python, installer notre application Flask avec ses dépendances. on va installer un container jenkins via docker dans notre système centos 7

sudo docker pull jenkins/jenkins:lts

sudo docker run -d -p 8080:8080 -p 50000:50000 --name jenkinsAtelier jenkins/jenkins:lts

sudo docker exec jenkinsAtelier cat /var/jenkins\_home/secrets/initialAdminPassword

puis on va installer python dans cette container par :

on va commencer par configurer le dns  
**sudo touch /etc/docker/daemon.json**

**sudo nano /etc/docker/daemon.json**

et coller : {

"dns": ["8.8.8.8", "8.8.4.4"]

} et sauvgarder

**sudo systemctl restart docker**

sudo docker exec -u root -it jenkinsAtelier bash

**apt-get update**

**apt-get install -y python3 python3-pip**

**exit**

sudo docker restart jenkinsAtelier

pour la configuration:  
git : <https://github.com/Chacronemed/flask.git>

560f38f72d634a079f854e6a930ed9c0

H/5 \* \* \* \* : j'ai sélectionné l'option "Poll SCM" avec le planning "H/5 \* \* \* \*". Cela signifie que Jenkins vérifiera toutes les 5 minutes si des changements ont été apportés au dépôt source (SCM) configuré pour le projet. Si des changements sont détectés, Jenkins déclenchera une nouvelle construction du projet.

python3 -m venv venv

. venv/bin/activate

cd examples/tutorial

pip install .

pip install pytest

pytest

Crée un environnement virt

Navigue dans le répertoire où se trouve le code de l'application Flask.uel Python puis on l'active

in stalle l'application Flask et ses dépendances directement à partir des configurations dans pyproject.toml.

Exécute la suite de tests de l'application.