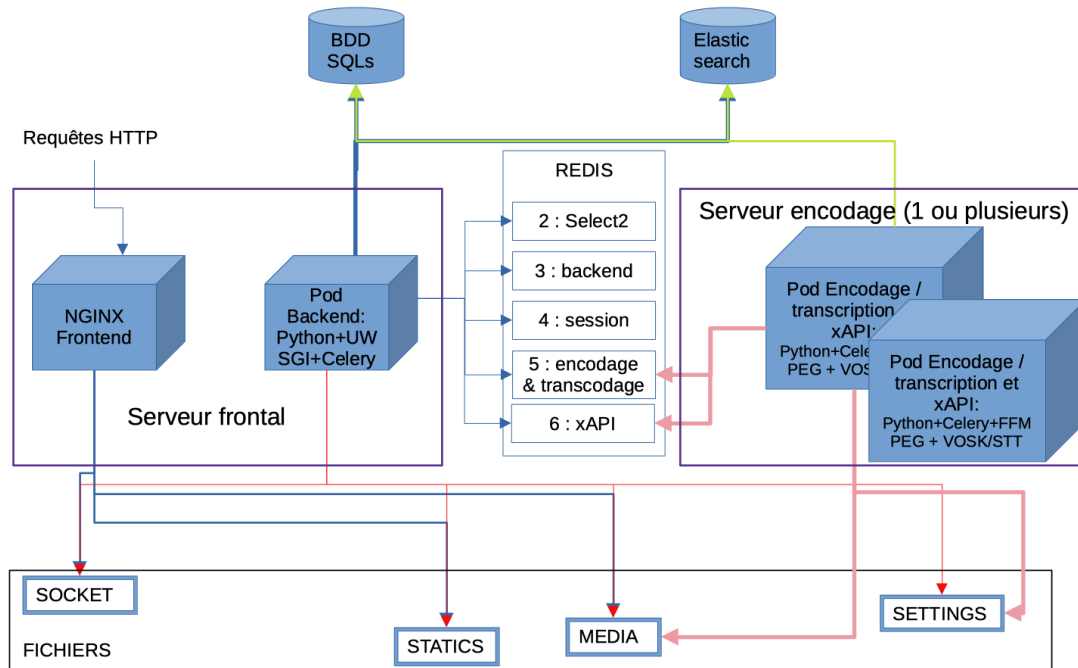


Mise en place de l'xAPI sur le serveur d'encodage



Nous appellerons dans la suite de cette documentation, **serveur frontal** le serveur où la partie web serveur est installée et **serveur encodage** le serveur où est déporté la tâche xAPI

Schéma de principe de fonctionnement :



Activation sur le serveur frontal :

Rajouter la configuration Celery/Redis dans le fichier `settings_local.py`

```
(django_pod) pod@pod:/usr/local/django_projects/podv3$ vim pod/custom/settings_local.py
```

```
# Configuration Celery sur le frontal

USE_XAPI = True
XAPI_ANONYMIZE_ACTOR = False
XAPI_LRS_LOGIN = "XXX"
XAPI_LRS_PWD = "XXX"
XAPI_LRS_URL = "http://monserveurLRS/xAPI/statements/"
USE_XAPI_VIDEO = True
XAPI_CELERY_BROKER_URL = "redis://redis:6379/6" # on utilise la db6 comme espace file d'attente sur redis
```

Installation sur le serveur d'encodage :

Il faut installer Pod sans réinitialiser la base et sans nginx/uwsgi/Elasticsearch. Vous pouvez suivre la doc [Installation de la plateforme Pod](#).

Rajouter la configuration de tout ça dans le fichier de configuration

Il faut maintenant dire au serveur d'encodage :

- Que l'on souhaite utiliser CELERY
- Donner l'adresse du serveur front de CELERY BROKER

```
(django_pod) pod@pod-encodage:/usr/local/django_projects/podv3$ vim pod/custom/settings_local.py
```

```
USE_XAPI = True
XAPI_ANONYMIZE_ACTOR = False
XAPI_LRS_LOGIN = "XXX"
XAPI_LRS_PWD = "XXX"
XAPI_LRS_URL = "http://monserveurLRS/xAPI/statements/"
USE_XAPI_VIDEO = True
XAPI_CELERY_BROKER_URL = "redis://redis:6379/6" # on utilise la db6 comme espace file d'attente sur redis
```

Activer Celery sur le serveur d'encodage

Mettre le contenu de <https://raw.githubusercontent.com/celery/celery/main/extra/generic-init.d/celeryd> dans /etc/init.d/celeryd

```
(django_pod) pod@pod-enc:~/django_projects/podv3$ sudo vim /etc/init.d/celeryd-xapi
(django_pod) pod@pod-enc:~/django_projects/podv3$ sudo chmod u+x /etc/init.d/celeryd-xapi
```

Créer le fichier default associé :

```
(django_pod) pod@pod-enc:/usr/local/django_projects/podv3$ sudo vim /etc/default/celeryd-xapi
```

```
CELERYD_NODES="worker-xapi" # Nom du/des worker(s). Ajoutez
autant de workers que de tache à executer en parallele.
DJANGO_SETTINGS_MODULE="pod.settings" # settings de votre Pod
CELERY_BIN="/home/pod/.virtualenvs/django_pod/bin/celery" # répertoire source de celery
CELERY_APP="pod.xapi.xapi_tasks" # application où se situe
celery
CELERY_ROUTES = {"pod.xapi.xapi_tasks.*": {"queue": "xapi"}}
CELERYD_CHDIR="/usr/local/django_projects/podv3" # répertoire du projet Pod (où se trouve
manage.py)
CELERYD_OPTS="--time-limit=86400 --concurrency=1 --max-tasks-per-child=1 --prefetch-multiplier=1" # options à
appliquer en plus sur le comportement du/des worker(s)
CELERYD_LOG_FILE="/var/log/celery/%N.log" # fichier log
CELERYD_PID_FILE="/var/run/celery/%N.pid" # fichier pid
CELERYD_USER="pod" # utilisateur système utilisant celery
CELERYD_GROUP="pod" # groupe système utilisant celery
CELERY_CREATE_DIRS=1 # si celery dispose du droit de création
de dossiers
CELERYD_LOG_LEVEL="INFO" # niveau d'information qui seront
inscrit dans les logs
```

Démarrer Celeryd

```
(django_pod) pod@pod-enc:~/django_projects/podv3$ sudo /etc/init.d/celeryd-xapi start
```

Pour vérifier si Celery fonctionne bien :

```
celery -A pod.xapi.xapi_tasks -l INFO -Q xapi --concurrency 1 -n xapi
```