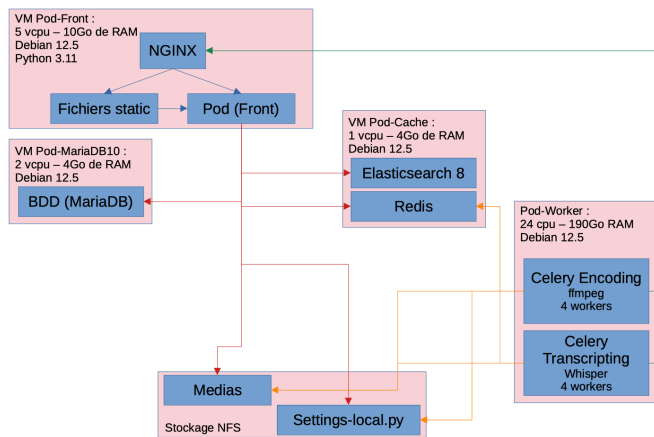


Installation Pod V3.6.0 ULille

Voici la nouvelle architecture mise en place à Lille en mai 2024 pour la version 3.6.0 de Pod (cliquer sur l'image pour l'agrandir)



Côté front, on a suivi l'installation classique de Pod présentée dans ce wiki.

Pour le fichier de settings local, on a posé le fichier sur le montage NFS pour la partager avec les encodeurs et les transpositeurs. On a fait de même pour les médias.

Organisation de fichier podv3

```
pod@pod-front:/usr/local/django_projects/podv3$ ll pod
total 288
authentication
bbb
chapter
completion
custom
cut
db_migrations
dressing
enrichment
import_video
__init__.py
live
locale
log
lti
main
media -> /srv/pod/media
meeting
node_modules
package.json
playlist
podfile
progressive_web_app
__pycache__
recorder
settings.py
static
urls.py
video
video_encode_transcript
video_search
wsgi.py
xapi
yarn.lock
pod@pod-front:/usr/local/django_projects/podv3$ ll pod/custom/
total 56
admin.py
apps.py
__init__.py
migrations
models.py
pod_nginx.conf
pod_uwsgi.ini
__pycache__
settings_local_docker_full_test.py
settings_local.py -> /srv/pod/settings_local.py
static
templates
tenants
tests.py
urls.py
views.py
pod@pod3:/usr/local/django_projects/podv3$
```

Pour le fichier de settings-local.py, voici son contenu

Settings-local.py

```
SECRET_KEY = 'XXXXXXXXXXXXXXXXXXXXX'

DEBUG = False
```

```

# Pour forcer le https
SECURE_SSL_REDIRECT = not DEBUG
SESSION_COOKIE_SECURE = not DEBUG
CSRF_COOKIE_SECURE = not DEBUG

ALLOWED_HOSTS = [
    'pod-front.univ-lille.fr', 'pod.univ-lille.fr'
]

ADMINS = (
    ('Nicolas', 'nicolas.can@univ-lille.fr'),
)
MANAGERS = ADMINS

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'pod3',
        'USER': 'pod',
        'PASSWORD': 'XXXXXXXXXXXXXXXXXXXX',
        'HOST': 'pod-bdd.univ.fr',
        'PORT': '3306',
        'OPTIONS': {
            'init_command': "SET storage_engine=INNODB, sql_mode='STRICT_TRANS_TABLES', innodb_strict_mode=1,
foreign_key_checks =1",
        },
    }
}

CACHES = {
    'default': {
        'BACKEND': 'django_redis.cache.RedisCache',
        'LOCATION': 'redis://default:XXXXXXXXXXXXXXXXXXXX@pod-cache:6379/3',
        'OPTIONS': {
            'CLIENT_CLASS': 'django_redis.client.DefaultClient',
        },
        'KEY_PREFIX': 'pod'
    },
    'session': {
        'BACKEND': 'django_redis.cache.RedisCache',
        'LOCATION': 'redis://default:XXXXXXXXXXXXXXXXXXXX@pod-cache:6379/3',
        'OPTIONS': {
            'CLIENT_CLASS': 'django_redis.client.DefaultClient',
        },
        'KEY_PREFIX': 'pod'
    },
    'select2': {
        'BACKEND': 'django_redis.cache.RedisCache',
        'LOCATION': 'redis://default:XXXXXXXXXXXXXXXXXXXX@pod-cache:6379/3',
        'OPTIONS': {
            'CLIENT_CLASS': 'django_redis.client.DefaultClient',
        },
    },
}

SESSION_CACHE_ALIAS = 'session'
SESSION_ENGINE = "django.contrib.sessions.backends.cache"
# les migrations des flatpages sont maintenant avec le code de pod
MIGRATION_MODULES = {'flatpages': 'pod.db_migrations.flatpages'}

TIME_ZONE = "Europe/Paris"

LANGUAGE_CODE = "fr"

LANGUAGES = (
    ('fr', 'Français'), ('en', 'English'),
)

FILE_UPLOAD_TEMP_DIR = "/srv/pod/tmp"

EMAIL_HOST = 'smtp.univ.fr'
EMAIL_PORT = 25

```

```
DEFAULT_FROM_EMAIL = 'no-reply@univ.fr'
SERVER_EMAIL = 'no@univ.fr'
SUPPORT_EMAIL = ['pod@univ.fr']
CONTACT_US_EMAIL = ['pod@univ.fr']

MENUBAR_HIDE_INACTIVE_OWNERS = True
MENUBAR_SHOW_STAFF_OWNERS_ONLY = True
HOMEPAGE_SHOWS_PASSWORDED = False
HOMEPAGE_SHOWS_RESTRICTED = False
FORCE_LOWERCASE_TAGS = True
MAX_TAG_LENGTH = 50

USE_PODFILE = True

THIRD_PARTY_APPS = ["enrichment", "live"]

HIDE_TAGS = True
HIDE_SHARE = True
HIDE_USER_TAB = False
HIDE_USER_FILTER = False
HIDE_USERNAME = False

LINK_SUPERPOSITION = True

USE_STATS_VIEW = True
VIEW_STATS_AUTH = False

MAX_DURATION_DATE_DELETE = 5

HIDE_LOCAL_LOGIN = True

ORGANIZE_BY_THEME = True

TEMPLATE_VISIBLE_SETTINGS = {
    'TITLE_SITE': 'Lille.Pod',
    'TITLE_ETB': 'Université de Lille',
    'LOGO_SITE': 'img/logoPod.svg',
    'LOGO_COMPACT_SITE': 'img/logoPod.svg',
    'LOGO_ETB': 'custom/images/Logo.sans.baseline-Horizontal-RVB-Blanc.svg',
    'LOGO_PLAYER': 'img/pod_favicon.svg',
    'FOOTER_TEXT': (
        '42, rue Paul Duez',
        '59000 Lille - France'
    ),
    'FAVICON': "custom/favicon.svg",
    'LINK_PLAYER': 'http://www.univ-lille.fr',
    'CSS_OVERRIDE': 'custom/override.css',
    'PRE_HEADER_TEMPLATE': 'custom/preheader.html',
    'POST_FOOTER_TEMPLATE': 'custom/footer.html',
    'TRACKING_TEMPLATE': 'custom/tracking.html',
}

ES_URL = ['https://pod-cache:9200']
ES_VERSION = 8
ES_INDEX = "pod"
ES_OPTIONS = {'verify_certs': False, 'basic_auth': ('elastic', 'XXXXXXXXXXXXXXXXXXXX')}

USE_CAS = True
CREATE_GROUP_FROM_AFFILIATION = True
CAS_LOGOUT_COMPLETELY = True
CAS_ADMIN_AUTH = False
CAS_GATEWAY = True
CAS_SERVER_URL = "https://cas.univ.fr/"
AUTH_CAS_USER_SEARCH = "user"
POPULATE_USER = "CAS"

AFFILIATION_STAFF = ('faculty', 'employee', 'staff')

CAS_FORCE_LOWERCASE_USERNAME = True

OEMBED = True
```

```

ACTIVE_VIDEO_COMMENT = True

USER_VIDEO_CATEGORY = True

LTI_ENABLED = True
PYLTI_CONFIG = {"consumers": {"univ.fr": {"secret": "XXXXXXXXXXXXXXXXXXXXX"}}}

# gestion de l'obsolescence des vidéos
USE_OBSOLESCENCE = True

WARN_DEADLINES = [60, 30, 7]

POD_ARCHIVE_AFFILIATION = ['faculty',
                             'staff',
                             'employee',
                             'affiliate',
                             'alum',
                             'library-walk-in',
                             'researcher',
                             'retired',
                             'emeritus',
                             'teacher',
                             'registered-reader',
                             'member']

ACCOMMODATION_YEARS = {
    'affiliate': 1
}
DEFAULT_YEAR_DATE_DELETE = 2

DARKMODE_ENABLED = True

DYSLEXIAMODE_ENABLED = True

COOKIE_LEARN_MORE = "https://www.univ.fr/dp/"

USE_OPENCAST_STUDIO = True

USE_MEETING = True
BBB_API_URL = "https://bbb.numerique-esr.fr"
BBB_SECRET_KEY = "XXXXXXXXXXXXXXXXXXXXX"
MEETING_DISABLE_RECORD = False
RESTRICT_EDIT_MEETING_ACCESS_TO_STAFF_ONLY = True

AFFILIATION_EVENT = ("event creator", )
EVENT_GROUP_ADMIN = "event administrator"
EMAIL_ON_EVENT_SCHEDULING = True

USE_NOTIFICATIONS = True
WEBPUSH_SETTINGS = {
    "VAPID_PUBLIC_KEY": "XXXXXXXXXXXXXXXXXXXXX",
    "VAPID_PRIVATE_KEY": "XXXXXXXXXXXXXXXXXXXXX",
    "VAPID_ADMIN_EMAIL": "nicolas.can@univ-lille.fr"
}

# LOGGING = {}

# Configuration Celery sur le frontal
USE_REMOTE_ENCODING_TRANSCODING = True
ENCODING_TRANSCODING_CELERY_BROKER_URL = "redis://default:XXXXXXXXXXXXXXXXXXXXX@pod-cache:6379/5"
POD_API_URL = "https://pod.univ-lille.fr/rest/"
POD_API_TOKEN = "XXXXXXXXXXXXXXXXXXXXX"
USE_TRANSCRIPTION = True
TRANSCRIPTION_TYPE = "WHISPER"
TRANSCRIPTION_MODEL_PARAM = {
    'WHISPER': {
        'fr': {
            'model': "small",
            'download_root': "/data/transcription/whisper/",
        },
        'en': {

```

```

        'model': "small",
        'download_root': "/data/transcription/whisper/",
    }
}
}

```

Ensuite, côté UWSGI, voici ce que ça donne

pod_uwsgi

```

# pod_uwsgi.ini file
[uwsgi]

# Django-related settings
# the base directory (full path)
chdir          = /usr/local/django_projects/podv3
# Django's wsgi file
module         = pod.wsgi
# the virtualenv (full path)
home          = /home/pod/.virtualenvs/django_pod3
# process-related settings
# master
master         = true
# maximum number of worker processes
processes      = 30
# the socket (use the full path to be safe)
socket         = /usr/local/django_projects/podv3/podv3.sock
# http
http           = :8000
# ... with appropriate permissions - may be needed
chmod-socket   = 666
# clear environment on exit
vacuum         = true
# In case of numerous/long cookies and/or long query string, the HTTP header may exceed default 4k.
# When it occurs, uwsgi rejects those requests with error "invalid request block size" and nginx returns HTTP
502.
# Allowing 8k is a safe value that still allows weird long cookies set on .univ-xxx.fr
buffer-size    = 8192

# To log to files instead of stdout/stderr, use 'logto',
# or to simultaneously daemonize uWSGI, 'daemonize'.
# daemonize     = /home/pod/django_projects/podv3/pod/log/uwsgi-pod.log
logto          = /home/pod/django_projects/podv3/pod/log/uwsgi-pod.log

# recommended params by https://www.techatbloomberg.com/blog/configuring-uwsgi-production-deployment/
strict         = true ; This option tells uWSGI to fail to start if any parameter in the configuration file
isn't explicitly understood.
die-on-term    = true ; Shutdown when receiving SIGTERM (default is respawn)
need-app       = true ; This parameter prevents uWSGI from starting if it is unable to find or load your
application module.

```

J'ai mis 30 processes... A voir !

pour l'encodage et la transcription, j'ai suivi cette doc <https://www.esup-portail.org/wiki/display/ES/Gestion+de+l'encodage%2C+de+la+transcription+et+de+l'API#Gestion+de+l'encodage,de+la+transcription+et+de+l'API-3.D%C3%A9port%C3%A9sur+un+machine+ou+docker+en+microservice>

Sur la même machine, j'ai 2 virtual env (**django_pod3_encodage** et **django_pod3_transcodage**) et 2 celery qui tournent : 1 pour l'encodage et 1 pour la transcription.

Voici un exemple pour le celery de l'encodage :

Config celery encodage

```
pod@pod-worker:/usr/local/django_projects/podv3$ cat /etc/default/celeryd-encodage
CELERYD_NODES="worker-encodage1 worker-encodage2 worker-encodage3 worker-encodage4" # Nom du/des worker(s).
Ajoutez autant de workers que de tache à executer en parallele.
CELERY_BIN="/home/pod/.virtualenvs/django_pod3_encodage/bin/celery" # répertoire source de
celery
CELERY_APP="pod.video_encode_transcript.encoding_tasks" # application où se situe
celery
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 -Q encoding -
n encode" # 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
```

et pour celui de la transcription :

Config celery transcription

```
pod@pod-worker:/usr/local/django_projects/podv3$ cat /etc/default/celeryd-transcodage
CELERYD_NODES="worker-transcodage1 worker-transcodage2 worker-transcodage3 worker-transcodage4" # Nom du/des
worker(s). Ajoutez autant de workers que de tache à executer en parallele.
CELERY_BIN="/home/pod/.virtualenvs/django_pod3_transcodage/bin/celery" # répertoire
source de celery
CELERY_APP="pod.video_encode_transcript.transcripting_tasks" # application
où se situe celery
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 -Q
transcripting -n transcript" # 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
```