

OSUG-DC

03/10/2019

Présentation de l'architecture SSHADE

Philippe Bollard, IPAG

VM 1 : OSUG-SSHADE (Production)

- **Debian 8.11**
- **4 vCPU**
- **16G RAM**
- **~50Go alloués mais partitionnement à revoir**
 - 25Go /srv (21Go utilisés)
 - 'apps' + données: sshade, ghosst, wiki, autres sites, ...
 - Contenu 'déplacé' depuis /var trop petit
 - postgresql
 - autopostgresqlbackup
 - 5,4Go / (2,4Go utilisés)
 - 5,7Go /var (4Go utilisés)
 - Trop petit: pose problème (stockage BDD, ...)
 - 1,4Go /tmp
 - Trop petit: pose problème lors de l'upload de grosses archives à importer via le Web
 - 5,9Go /home (2,9Go utilisés)
 - Pas vraiment d'utilité...



VM 2 : OSUG-SSHADE-DEV (Développement)

- **Debian 8.11**
- **2 vCPU**
- **8G RAM**
- **~50Go alloués**
 - (même soucis de partitionnement)
- **Contenu**
 - `dev.sshade.eu`
 - Redmine (`forge.sshade.eu`) : Obsolète



SSHADE: Dépendances logicielles

- **Python 2.7 (migration vers 3 à faire...) + Virtualenv**
 - Pyramid
 - SQLAlchemy + Alembic
 - CLI via Click
 -
- **PostgreSQL**
 - autopostgresqlbackup
 - phppgadmin
 - pg_crypto
 -
- **Elasticsearch 5**
- **Redis**
- **Nginx**
 - uWSGI (emperor)
 - Letsencrypt
 - PHP-FPM (pour le DokuWiki)



SSHADE: Dépendances Python (requirements.txt)

SQLALCHEMY

psycopg2-binary
SQLAlchemy
alembic
SQLAlchemy-Utils
SQLAlchemy-Utc

PYRAMID

plaster_pastedeploy
pyramid >= 1.9a
pyramid_jinja2
pyramid_layout

WTFORMS

wtforms
wtforms-components
wtforms-alchemy
pyramid_wtforms

MAILER

premailer
pyramid_mailer

SESSION

pyramid_redis_sessions

ELASTICSEARCH

elasticsearch>=5.0.0,<6.0.0
elasticsearch-dsl>=5.0.0,<6.0.0

<https://github.com/rohanisaac/spc/>

TOOLS

numpy
enum34
click
lxml
passlib
bcrypt
json_tricks
rarfile
Babel
python-dateutil
StringGenerator
stringcase
matplotlib
datacite
pint
chardet
bibtextparser
astropy
BeautifulSoup4
python-slugify
WeasyPrint==0.42.3
pysitemap
requests
pylatexenc
shortuuid
pygeoip
pagecalc
pytz
filetype
datetimeutc



SSHADE: CLI

- **Database**

- Create
- Drop
- Upgrade/Downgrade (via Alembic)

- **User**

- Create

- **Import (de données)**

- **Export (VO)**

- **Elastic**

- Create/Drop index
- Create/Drop document
- Reindex

- **Backup**

- Save users
- Restore users
- Save data
- Restore data

- **Maintenance**

- Enable
- Disable

- **Cache**

- Nettoyage des preview de spectre/experiment
- Nettoyage des ressources orphelines

- **Sitemap**



SSHADE: Scripts

- **Update**

- sshade-cli maintenance enable
- Git pull
- Clean
- sshade-cli maintenance disable
- Reload uwsgi

- **db_upgrade**

- sshade-cli maintenance enable
- Git pull
- Clean
- Sshade-cli database upgrade
- sshade-cli maintenance disable
- Reload uwsgi

- **pip_upgrade**

-

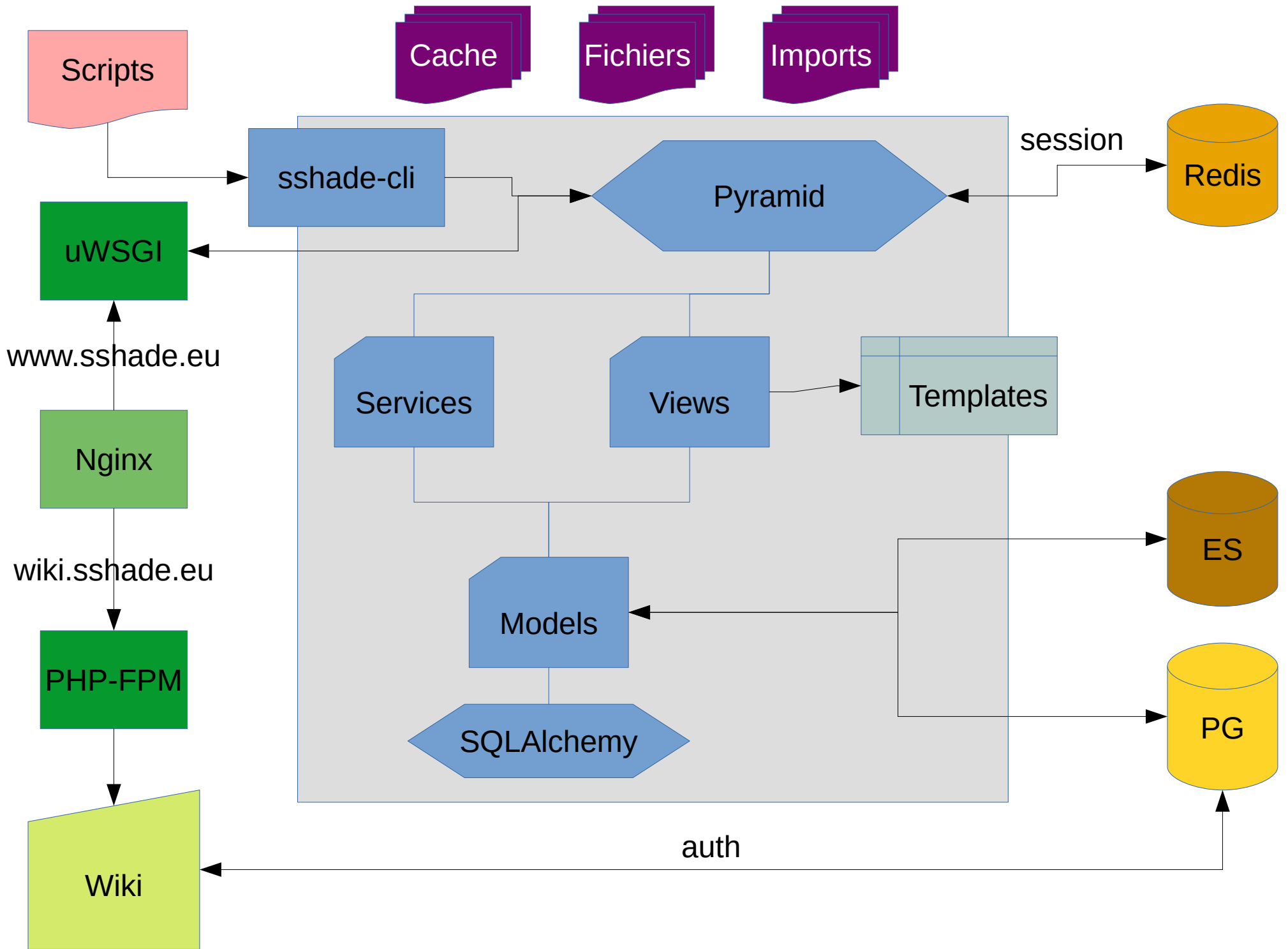
- **Reindex**

- Sshade-cli elastic reindex

- **Manuel**

- **CRON 1x/jour**





SSHADE: Mutualisation possible

- **Elasticsearch**

- 25 indexes / instance de SSHADE

- **PostgreSQL si...**

- Accès Web (phpPgAdmin)
- Dumps SQL (via autopostgresqlbackup)
- Droit de créer/supprimer une base
- Pas d'impact sur les nombreuses transactions

- **MySQL (inutilisé par SSHADE)**

- GhoSST
- Piwik / Matomo
- Autres sites



Amélioration d'arch.: VM

- **Instances VM**

- 1 VM prod / 1 VM dev (situation actuelle)
- Découpage VM par “service” ?
 - Web/App
 - Python
 - PHP
 - BDD
 - Elasticsearch
 - Stockage
- Découpage en mode Docker ?

- **Question VMWare:**

- Notion de dépendance entre VM ?



Amélioration d'arch.: Data

- **Backup**

- Archive générée par CLI
 - Comptes utilisateurs
 - XML+Données sources à importer
 - Log des opérations (import, validation, renommage, DOI, ...)
- Dumps SQL
- Fichiers

- **TODO:**

- CRON
- Copie sur NFS

