

RETOUR D'EXPÉRIENCE CLIENT

SOCLE DALIBO



INTRODUCTION

PRÉSENTATION



Clément Paillier

-
- DBA (Oracle[©]) depuis 2016
 - PostgreSQL depuis 2019
 - MAIF depuis 2022

INTRODUCTION

OBJET DE CE « REX »

- Historique « MAIF + Dalibo »
- L'arrivée du « socle v2 »
- Sa mise en place
- Son utilisation aujourd'hui

HISTORIQUE

2016 : MAIF souhaite une alternative à Oracle comme RDBMS.

- Pas de compétences en interne
- Optique de mise en œuvre simple et industrialisable

HISTORIQUE (SUITE)

2016 : Choix de **Dalibo** et le socle « essential »

- Scripts pour implémentation simplifiée :
 - création d'instance,
 - sauvegarde...
- Uniformisation du parc
- Support des instances où le socle est déployé.
- Accompagnement, personnalisation et documentation

HISTORIQUE (SUITE)

2017 : Arrivée d'**Ansible** comme outil d'automatisation

- Problème : les scripts sont interactifs
- Début du paramétrage en « json »
- Haute-disponibilité : passage au « socle advanced »

HISTORIQUE (SUITE)

2018 à 2022 : Utilisation massive d'Ansible pour l'exploitation du socle

- A évolué au fil des besoins
- Progression par itérations
- Solution à la carte
- ... Qui devient difficile à maintenir

SOCLE V2

Annonce fin 2022 de l'arrivée d'un socle entièrement repensé :

- Plus modulaire et évolutif
- CLI : pglift
- Forte intégration avec Ansible (*collections*)
- pgbackrest comme outil de sauvegarde principal
- Compatible RHEL8 / Debian

SOCLE V2

Contexte Maif :

- Négociation d'un contrat « groupe » avec nos filiales
 - Uniformisation autour d'un socle commun
- ⇒ Va nécessiter une importante adaptation de l'industrialisation Maif

2023 : ARRIVÉE DU SOCLE V2

PREMIER TRIMESTRE

- Définition des normes
- Collecte des besoins
 - ⇒ Livraison d'un dossier d'architecture

2023 : ARRIVÉE DU SOCLE V2

SECOND TRIMESTRE

- Démarrage du projet
- Première livraison du socle fin Avril
 - ⇒ Début des travaux de recette / ré-écriture Ansible

2023 : ARRIVÉE DU SOCLE V2

TROISIÈME TRIMESTRE

- Recette intensive
- Nombreux échanges sur notre implémentation via tickets
- Points de situation hebdomadaires jusqu'en fin de recette

2023 : ARRIVÉE DU SOCLE V2

PREMIERS DÉPLOIEMENTS « OFFICIELS » EN NOVEMBRE

- sortie de pglift 1.0.0
- les collections ouvertes au public courant 2024 :
 - extras : 01/2024 (*v2.0.0*)
 - essential : 05/2024 (*v4.0.0*)
 - advanced : 08/2024 (*v2.0.0*)

2023 : ARRIVÉE DU SOCLE V2

BILAN

Groupe Maif parmi les premiers clients à passer au socle v2

- avec une infrastructure complexe :
 - Patroni
 - ETCD mutualisés
 - pgbackrest centralisé (stockage "S3")
 - Sécurisation des échanges : TLS

FOCUS SUR L'IMPLEMENTATION

Rationalisation du code :

- Partir d'une page blanche
- Code plus structuré
- Commenté et documenté dès le départ !
- Bonnes pratiques Ansible : rôles / variables / syntaxe
- et de développement : DRY (conception modulaire / réutilisation)

FOCUS SUR L'IMPLÉMENTATION

Rationalisation du code :

- Projet Ansible Socle v1

```
1          363 text files.
2          314 unique files.
3          120 files ignored.
4
5 -----
6 Language                files          blank          comment
7 code -----
8 YAML                    196           1864           2549
9 SQL                      28            420            490
10 Python                   7            318            445
11 Markdown                  5             96              0
12 XML                       6              0              0
13 Korn Shell                 1              6              41
14 Bourne Shell               1              1              0
15 -----
16 SUM:                      244           2705           3525
17 15030
18 -----
```


FOCUS SUR L'IMPLEMENTATION

Rationalisation du code :

- Projet Ansible Socle v2

```
1          368 text files.
2          259 unique files.
3          141 files ignored.
4
5 -----
6 Language                files          blank          comment
7 code
8 -----
8 YAML                    185           1540           1749
9 Python                   4             266            391
10 Markdown                24            405             0
11 SQL                     14             0               0
12 -----
13 SUM:                    227           2211           2140
14 -----
```

FOCUS SUR L'IMPLÉMENTATION

Extrait de playbook de préparation d'une VM :

```
- hosts: database
  gather_facts: yes
  become: yes
  become_user: root
  pre_tasks:
    - name: Activation des dépôts de Dalibo et autres dépôts système
      requis:
        community.general.rhsm_repository:
          name: "{{ mailf_specific_repo_list }}"
          state: enabled
  roles:
    # Pour enregistrer le produit postgresql lié à la machine
    - role: register_atu
      tags: register_atu,system
    # Role de gestion du user postgres
    - role: crea_user
      tags: crea_user,system
    # Création des FS et arborescences utiles à l'installation du socle
    - role: crea_arbo
      tags: crea_arbo,system
    # Préparation supplémentaire (packages : acl/langpack-fr,
    désactivation module postgresql)
    - role: init_vm
      tags: init_vm,system
    # Gestion certificats
    - role: gestion_certificat
      tags: gestion_certificat,system
    # Roles Dalibo
    - role: dalibo.essential.postgresql
      tags: dalibo-postgresql,postgresql,dalibo
    - role: post install
```

```
tags: post_install,postgresql
- role: dalibo:essential.pgbackrest
tags: dalibo-pgbackrest,postgresql,dalibo
- role: dalibo:essential.prometheus_postgres_exporter
tags: dalibo-postgres_exporter,postgresql,dalibo
- role: dalibo:advanced.patroni
tags: dalibo-patroni,postgresql,dalibo
when: haute_dispo == OUI
- role: dalibo:advanced.powa
tags: dalibo-powa,postgresql,dalibo
- role: dalibo:advanced.pgaudit
tags: dalibo-pgaudit,postgresql,dalibo
- role: dalibo:extras.rsyslog
tags: dalibo-rsyslog,postgresql,dalibo
- role: dalibo:extras.logrotate
tags: dalibo-logrotate,postgresql,dalibo
- role: dalibo:essential.pglift
tags: dalibo-pglift,postgresql,dalibo
```

FOCUS SUR L'IMPLEMENTATION

Extrait de playbook, gestion d'une instance PostgreSQL

```
- name: Instance Standalone
  dalibo:pglift:instance:
    name: "{{ pglift_instance.name | default('main') }}"
    state: "{{ pglift_instance.state | default('started') }}"
    port: "{{ pglift_instance.port | default(5432) }}"
    version: "{{ pglift_instance.version | default(version) }}"
    surote_password: "{{ postgres_postgres_pwd }}"
    replrote_password: "{{ postgres_replication_pwd }}"
    powa:
      password: "{{ postgres_powa_pwd }}"
    pgbackrest:
      stanza: "{{ pglift_instance.stanza }}"
      password: "{{ postgres_backup_pwd }}"
    prometheus:
      port: "{{ pg_exporter_port | default('9876') }}"
      password: "{{ postgres_postgres_exporter_pwd }}"
  environment: "{{ pg_xdg_env }}"
```

LE CLI PGLIFT

```
[postgres@srvpg ~]$ pglift instance list
```

name	version	port	datadir	status
main	16	5432	/var/lib/pgsql/data/16/main/data	running

```
[postgres@srvpg ~]$ pglift instance list -o json
```

```
[  
  {  
    "name": "main",  
    "version": "16",  
    "port": 5432,  
    "datadir": "/var/lib/pgsql/data/16/main/data",  
    "status": "running"  
  }  
]
```

UN SOCLE EN ÉVOLUTION

RÉCEMMENT AJOUTÉ :

- temboard
- PostGIS
- HAProxy
- Pg Anonymizer

DE NOUVELLES PERSPECTIVES ?

- Conteneurisation
- Open Telemetry
- ...

MERCI

- À Dalibo pour l'invitation

QUESTION ?