Serveur dédié PDB-SRV

Serveur physique dédié du SINP AURA, destiné à porter la base PostgreSQL de GeoNature.



Les accès d'administration, le port SSH et les empreintes des clés d'hôte sont sur le wiki Intranet : fiche privée de pdb-srv.

- Voir aussi : Infos services de l'hébergement SINP AURA
- Relevé du 2026-09-02.

Détails techniques

- Nom : **pdb-srv**
- Port SSH : contacter le CBNA → adminsys [at] cbn-alpin.fr
- Hébergeur / Localisation : OVH / Gravelines (GRA04)
- Type hébergement : Serveur dédié (Bare Metal)
- Modèle : ADVANCE-2
- Processeur : AMD EPYC 4344P, 8 cœurs et 16 threads
- Mémoire vive : 64 Go DDR5 avec ECC sur puce
- Stockage : 4 NVMe de 960 Go en RAID logiciel, 1,8 To utile
- Bande passante publique : 3 Gbps
- OS / Distribution : Debian 13 (Trixie)
- IP Public : 51.91.50.49
- IP Privée : 10.0.1.40
- IPv6 Public : aucune
- Réseau privé : sinp-aura-vpn (vRack)
- Prix : XXX € HT/mois
- Historique :
  - 2026-09-02 - Montage des archives Borg ajouté à la stack de sauvegarde.
  - 2026-08-27 - Sauvegarde Borgmatic complète, dépôt local et dépôt distant sur bkp-srv.
  - 2026-08-19 - Création du cluster PostgreSQL 18.
  - 2026-07-30 - Installation de Debian 13 avec le RAID logiciel.

Matériel

- Carte mère : MSI S3661
- BIOS : American Megatrends ES366AOC.10V
- BMC : ASPEED, IPMI présent
- Carte réseau : Intel E810-XXV, deux ports SFP28, pilote `ice`
  - Port public : `enp10s0f0np0`, lien 25 Gbps
  - Port privé : `enp10s0f1np1`, lien 25 Gbps, raccordé au vRack
- Deux emplacements mémoire libres sur quatre, 128 Go possibles

Les quatre disques

Relevé SMART du 2026-08-10, les quatre PASSED, 0 % d'usure et aucune erreur média.

Disque	Modèle	Série	Firmware	Heures
nvme0n1	Samsung PM9A3 MZQL2960HCJR-00A07	S64FNN0XB23793	GDC5A02Q	810 h
nvme1n1	Samsung PM9A3 MZQL2960HCJR-00A07	S64FNN0XB25358	GDC5A02Q	6 859 h
nvme2n1	Micron 7500 PRO MTFDKCC960TGP-1BK1DABYY	032510AA5CA0	E3MQ005	5 450 h
nvme3n1	Micron 7500 PRO MTFDKCC960TGP-1BK1DABYY	032510AA5C63	E3MQ005	5 450 h

Les températures courantes sont dans Grafana, elles ne sont pas reprises ici.

Les quatre disques sont formatés en secteurs logiques de 512 octets. Un disque de remplacement doit être formaté de la même façon avant d'être intégré à une grappe.

Stockage et RAID

Le RAID est logiciel, géré par `mdadm`. Il n'y a pas de carte RAID. Chaque grappe est faite d'une partition par disque et s'étend donc sur les quatre disques.

Le RAID 10 a été retenu contre le RAID 6. À quatre disques les deux donnent la même capacité utile et le RAID 10 conserve le TRIM que le noyau désactive sur les grappes à parité, reconstruit plus vite et donne de meilleures performances.

Grappe	Niveau	Métadonnées	Taille	Montage	Système
md1	RAID 1 sur 4 disques	**0.90**	512 Mo	`/boot/efi`	vfat
md2	RAID 1 sur 4 disques	1.2	4 Go	`/boot`	ext4
md3	RAID 10 sur 4 disques, near=2, chunk 512K	1.2	1,8 To	`/`	ext4

Chaque grappe utilise la partition de même rang sur les quatre disques, `p1` pour md1, `p2` pour md2 et `p3` pour md3. Il n'y a pas de `/var` séparé.

Ordre des membres de md3, tel que le donne `mdadm –detail /dev/md3`.

Rang	Partition	Ensemble
0	nvme2n1p3	set-A
1	nvme3n1p3	set-B
2	nvme0n1p3	set-A
3	nvme1n1p3	set-B

Diagnostiquer une grappe

```
sudo su -
cat /proc/mdstat
mdadm --detail /dev/md3
grep . /sys/block/md*/md/mismatch_cnt
```

```
lsblk -o NAME,SIZE,TYPE,MOUNTPOINT,MODEL
```

Les trois `mismatch_cnt` doivent valoir 0. Le contrôle mensuel des grappes est automatique, voir la section des contrôles périodiques.

Système

- Langue et clavier en français, fuseau Europe/Paris
- Mises à jour automatiques par `unattended-upgrades`. Un noyau mis à jour n'est chargé qu'au redémarrage suivant donc vérifier la version en cours avec `uname -r` avant de conclure qu'une mise à jour est appliquée.
- Mémoire compressée ZRAM à 25 % de la RAM, soit 16 Go, priorité 100
- Fichier d'échange `/swapfile01` de 64 Go, priorité 10
- Pages mémoire transparentes forcées à `never` pour PostgreSQL
- Postfix en envoi seul vers le relais OVH, Fail2ban et RKHunter actifs
- Les autres machines du SINP AURA sont connues par leur nom sur le réseau privé, `web-aura-sinp`, `db-aura-sinp` et `bkp-aura-sinp`, déclarées dans `/etc/hosts`

Comptes

Compte	Usage
debian	compte d'origine de l'installeur OVH
admin	administration courante
geonat	compte applicatif GeoNature

Connexion par clé uniquement, mot de passe refusé.

Pare-feu

Firewalld avec trois zones. SSH est ouvert sur un port personnalisé et PostgreSQL uniquement sur le réseau privé.

Zone	Interface	Ce qui est ouvert
public	enp10s0f0np0	SSH
sinp-aura	enp10s0f1np1	SSH, PostgreSQL 5432
docker	interfaces des conteneurs	géré par Docker

Points de vigilance

Cinq configurations volontaires qui s'écartent du comportement attendu par défaut.

Constat	Explication
Pas d'IPv6	désactivée sur les deux interfaces physiques, la boucle locale `::1` est conservée
`ssh root@` est refusé	un `command=` dans l'`authorized_keys` de root renvoie vers le compte `debian`, c'est la configuration d'origine OVH

Constat	Explication
`borg` n'existe pas en ligne de commande	Borg est dans le conteneur et pas sur l'hôte, comme sur les autres serveurs du SINP
md1 est en métadonnées 0.90	ce format place le superbloc en fin de partition donc le firmware UEFI lit le FAT32 sans connaître le RAID. Recréer cette grappe en 1.2 empêcherait la machine de démarrer
PostgreSQL écoute sur 10.0.1.40 dès le démarrage	`net.ipv4.ip_nonlocal_bind = 1` dans `/etc/sysctl.d/30-postgresql.conf`, sans lui il démarre avant le réseau privé et n'écoute que sur la boucle locale

Docker

- Docker CE 29.7.2, Docker Compose 5.4.0
- Les stacks sont dans `/home/admin/docker/`

Stack	Image	Rôle
borgmatic	biodivaura/borgmatic:2026.09.02	sauvegarde
telegraf	biodivaura/telegraf:1.39.3	supervision

PostgreSQL

- Serveur et client 18.6, PostGIS 3.6.4, depuis le dépôt PGDG avec une priorité APT de 200
- Cluster `18/main` en UTF-8, locale `fr_FR.UTF-8`
- Sommes de contrôle des données activées (`data_checksums`)
- Écoute sur la boucle locale et sur l'adresse privée 10.0.1.40

Sauvegarde

Borgmatic sauvegarde le système tous les jours à 6 h sur deux dépôts, un local et un distant sur bkp-srv. La sauvegarde des bases PostgreSQL n'est pas encore activée.

Élément	Valeur
Chemins sauvegardés	`/etc`, `/home`, `/opt`, `/root`, `/usr/local`
Dépôts	deux, nommés `local` et `bkp-srv` dans la configuration
Chiffrement	repokey-blake2
Compression	zstd, niveau 6
Nom des archives	pdb-srv-{date}
Rétention	7 quotidiennes, 4 hebdomadaires, 6 mensuelles, 3 annuelles
Contrôles	dépôt et trois dernières archives après chaque sauvegarde
Planification	cron du conteneur, tous les jours à 6 h
Notifications	Telegram au démarrage, à la réussite et à l'échec

La clé SSH utilisée vers le serveur de sauvegarde est enfermée dans le seul dépôt de pdb-srv par une commande forcée. Elle n'ouvre aucun shell et ne donne pas accès aux dépôts des autres machines.

Les commandes ci-dessous utilisent les noms de dépôt `local` et `bkp-srv`, il n'y a donc pas besoin de connaître l'adresse du serveur de sauvegarde pour les exécuter.

Voir ce qui est sauvegardé

```
cd /home/admin/docker/borgmatic
docker exec borgmatic sh -c '. /etc/environment; borgmatic repo-info'
docker exec borgmatic sh -c '. /etc/environment; borgmatic repo-list --
repository local'
docker exec borgmatic sh -c '. /etc/environment; borgmatic repo-list --
repository bkp-srv'
docker exec borgmatic sh -c '. /etc/environment; borgmatic list --repository
local --archive latest'
```

`repo-list` donne les archives d'un dépôt, `list` donne le contenu d'une archive. Dans les exemples qui suivent, `latest` désigne la dernière archive et peut être remplacé par un nom d'archive donné par `repo-list`.

Récupérer un fichier par extraction

C'est la méthode la plus simple. Elle ne demande ni FUSE ni capacité particulière au conteneur. Elle se fait dans un conteneur jetable donc on écrit dans un dossier partagé avec l'hôte.

```
sudo su -
cd /home/admin/docker/borgmatic
rm -rf /tmp/restore
mkdir -p /tmp/restore
docker compose run --rm --no-deps -v /tmp/restore:/tmp/restore borgmatic
borgmatic --verbosity 1 extract --repository local --archive latest --path
mnt/source/etc/hostname --destination /tmp/restore
ls -l /tmp/restore/mnt/source/etc/
```

Le chemin s'écrit sans le slash du début. Les fichiers ressortent sous `/tmp/restore/mnt/source/…` puisque la racine du serveur est montée sur `/mnt/source` dans le conteneur.

Pour le dépôt distant, remplacer `--repository local` par `--repository bkp-srv` .

Le bloc s'exécute depuis un shell root pour une seule raison, le conteneur écrit en tant que `root` donc les fichiers produits lui appartiennent et le nettoyage du dossier temporaire échoue sans ces droits.

Parcourir une archive par montage

Utile quand on ne sait pas exactement quel fichier chercher. Le montage utilise FUSE et demande donc le conteneur de restauration.

```
sudo su -
cd /home/admin/docker/borgmatic
rm -rf /tmp/restore
mkdir -p /tmp/restore
docker compose -f docker-compose.yml -f docker-compose.restore.yml run --rm
```

```
--no-deps -v /tmp/restore:/tmp/restore borgmatic
```

Puis, dans le shell du conteneur :

```
mkdir -p /tmp/repo
borgmatic mount --repository bkp-srv --archive latest --mount-point
/tmp/repo
ls /tmp/repo/mnt/source/etc
mkdir -p /tmp/restore/mnt/source/etc
cp -a /tmp/repo/mnt/source/etc/hostname /tmp/restore/mnt/source/etc/
borgmatic umount --mount-point /tmp/repo
exit
```

Le fichier est copié dans la même arborescence que celle produite par l'extraction, pour que la remise en place ci-dessous fonctionne après l'une ou l'autre méthode.

Remettre un fichier en place

Toujours dans le shell root ouvert plus haut. Rien ne doit être remplacé sans contrôle. Un fichier de configuration restauré peut être plus ancien que la configuration en cours et le service concerné peut avoir besoin d'être rechargé ensuite.

```
diff /tmp/restore/mnt/source/etc/hostname /etc/hostname
ls -l /tmp/restore/mnt/source/etc/hostname /etc/hostname
cp -a /etc/hostname /etc/hostname.back
cp -a /tmp/restore/mnt/source/etc/hostname /etc/hostname
ls -l /etc/hostname
rm -rf /tmp/restore
```

Vérifier dans l'ordre le contenu du fichier extrait, ses droits et son propriétaire, puis l'incidence sur le service qui l'utilise. Recharger ou redémarrer ce service si nécessaire.

Supervision et alertes

- Telegraf envoie vers l'InfluxDB de la stack AURA, avec un jeton propre à pdb-srv en écriture seule
- Les tableaux de bord sont sur monitor.biodiversite-aura.net
- La disponibilité générale, l'état du RAID et l'état SMART des disques sont contrôlés par healthcheck.biodiversite-aura.net. Une tâche cron appelle les trois contrôles, il n'y a pas d'agent installé sur la machine.

Source	Ce qui déclenche	Où ça arrive
mdadm	grappe dégradée ou en échec	courriel, alias d'alertes
smartd	usure ou erreur d'un disque	courriel, alias d'alertes
RKHunter	fichier suspect au contrôle quotidien	courriel, alias d'alertes
Borgmatic	démarrage, réussite et échec de la sauvegarde	Telegram
Healthchecks	contrôle non reçu dans le délai attendu	Telegram

Contrôles périodiques

Quoi	Fréquence	Mécanisme
Contrôle des grappes RAID	mensuel	`mdcheck_start.timer`
Surveillance des grappes	continue	`mdmonitor`
TRIM des disques	hebdomadaire	`fstrim.timer`
Contrôle RKHunter	quotidien	tâche cron
Mise à jour des signatures RKHunter	hebdomadaire	tâche cron
Sauvegarde Borgmatic et contrôle des dépôts	quotidien à 6 h	cron du conteneur
Mises à jour de sécurité	quotidienne	`unattended-upgrades`

Performances

Comparatif Fio entre le RAID 10 de pdb-srv et les deux stockages du serveur db-srv.

Test	pdb-srv RAID 10	db-srv, racine	db-srv, Block Storage /data
Écriture séquentielle 1 Mo	2 749 IOPS, 2 883 MB/s	594 IOPS, 623 MB/s	350 IOPS, 368 MB/s
Lecture séquentielle 1 Mo	15 500 IOPS, 16 300 MB/s	6 593 IOPS, 6 913 MB/s	350 IOPS, 368 MB/s
Écriture aléatoire 4 Ko	694 000 IOPS, 2 844 MB/s	77 900 IOPS, 319 MB/s	20 500 IOPS, 84,0 MB/s
Lecture aléatoire directe 4 Ko	1 951 000 IOPS, 7 991 MB/s	79 300 IOPS, 325 MB/s	20 500 IOPS, 83,9 MB/s
Lecture aléatoire historique direct=0	212 000 IOPS, 869 MB/s	55 400 IOPS, 227 MB/s	20 200 IOPS, 82,7 MB/s

Latence au 99e percentile.

Test	pdb-srv RAID 10	db-srv, racine	db-srv, Block Storage /data
Écriture aléatoire directe	2,606 ms	30,278 ms	190 ms
Lecture aléatoire directe	0,453 ms	8,291 ms	50 ms

Mesuré le 2026-08-24 avec Fio, 16 tâches et 16 fichiers de 1 Go par cible, profondeur 32 en séquentiel et en écriture aléatoire, 16 en lecture aléatoire. Le Block Storage plafonne au même chiffre en lecture et en écriture séquentielles, c'est la limite du volume.

From:
https://sinp-wiki.cbn-alpin.fr/ - **CBNA SINP**

Permanent link:
https://sinp-wiki.cbn-alpin.fr/serveurs/sinp-aura/pdb-srv

Last update: **2026/09/24 15:47**