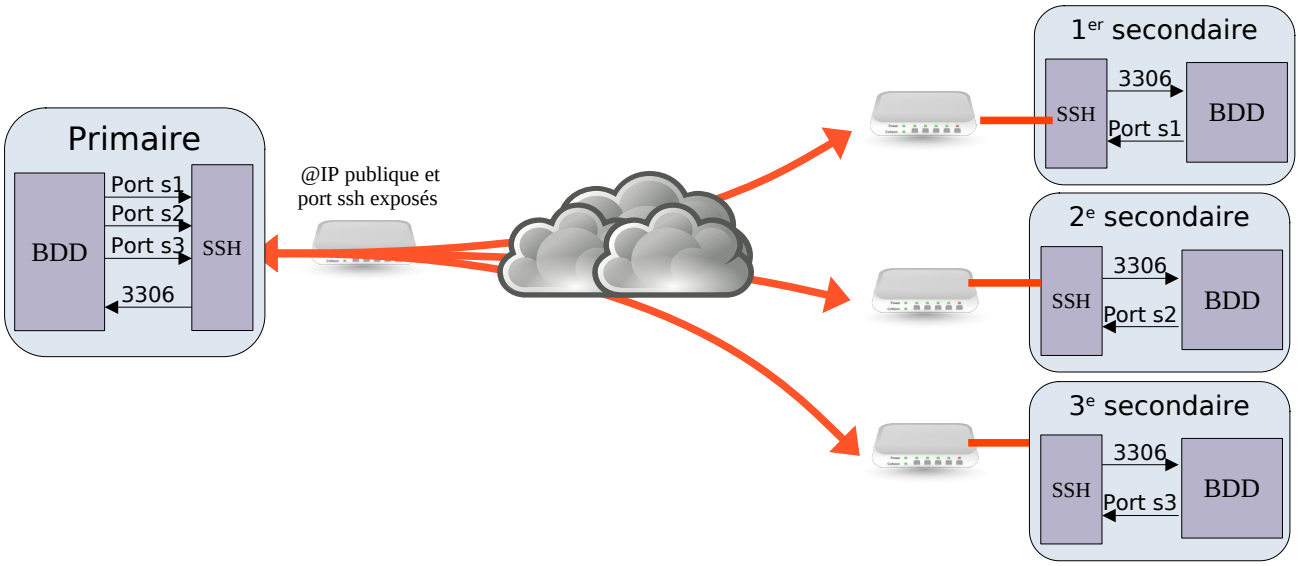
8.10 - Fédération

Le principe de la fédération est de créer des redirections de ports bidirectionnelles dans des tunnels SSH afin de permettre aux serveurs « mariadb » de gérer des répliqua ensemble de manière transparente en exposant leur port d'écoute 3306 uniquement sur l'interface loopback (127.0.0.1). Afin de limiter l'exposition sur Internet, seul l'ALCASAR primaire expose un port SSH.



Un exemple de commande permettant d'activer ce type de tunnelling à partir d'un secondaire : /usr/bin/ssh -NT -4 -o ServerAliveInterval=60 -o ExitOnForwardFailure=yes -p port_ssh_exposé -L port_s1:localhost:3306 -R port_s1:localhost:3306 replication@IP secondaire

En prenant l'exemple du schéma ci-dessus :

1. Lorsque la réplication est configurée, le secondaire se connecte via SSH de manière persistante sur le primaire via le port s1. Le primaire reçoit cette connexion sur le port 3306.
2. Lors de l'activation de la réplication via le tunnel SSH (START SLAVE), le secondaire reçoit le flux des évènements (transactions) depuis les logs binaires du primaire, au fur et à mesure qu'ils sont écrits. Dès qu'ils sont reçus, le secondaire les écrit dans ses « relay logs » locaux.
3. Le secondaire lit ses « relay logs » et applique les modifications à ses propres données.

8.10.1 - Installation/désinstallation de la fédération sur primaire ou secondaire (sur chaque ALCASAR)

- **alcasar-replication-install.sh**
  - Crée l'utilisateur BDD « db_replication ». ajoute « db_replication » + mdp dans « /root/ALCASAR-passwords.txt » ;
  - Crée le fichier de configuration Mariadb : « /etc/my.cnf.d/replication.cnf » ;
  - Autorise Mariadb à écouter en réseau sur 127.0.0.1 (modif « /etc/my.cnf.d/server.cnf ») ;
  - Crée le fichier de configuration du logging binaire et du REPLICA (/etc/my.cnf.d/replication.cnf) ;
  - Relance Mariadb
  - Crée l'utilisateur Linux « replication » Ajoute « replication » + mdp dans « /root/ALCASAR-passwords.txt » ;
  - Crée le fichier « .ssh/authorized_keys » dans « /home/replication » ;
  - Crée une biclé RSA dans « /root/.ssh » : id_rsa et id_rsa.pub ;
  - Crée le fichier /home/replication/local-db_replication-pwd.txt contenant le pwd de « db_replication »
  - Fixe la clé « REPLICATION=on » dans « alcasar.conf ».

- **alcasar-replication-uninstall.sh**
  - Stop les réplications en cours (/usr/local/bin/alcasar-replication-stop.sh –all) ;
  - Supprime toutes les conf de réplication (/usr/local/bin/alcasar-replication-delete.sh –all) ;
  - Supprime les entrées du fichier « /root/ALCASAR-passwords.txt » (replication et db_replication) ;
  - Supprime l'utilisateur BDD « db_replication » ;
  - Supprime le fichier de configuration du REPLICA ;
  - Supprime l'écoute de Mariadb sur 127.0.0.1 ;
  - Relance Mariadb
  - Supprime l'utilisateur Linux « replication » et son homedirectory. Tue ses processus actifs ;
  - Fixe la clé « REPLICATION=off » et la clé « REPLICATION_TO= » dans « alcasar.conf »
  - Relance le parefeu (fermeture des sorties SSH)

8.10.2 - Copie de la clé publique SSH d'un secondaire vers le primaire.

- **alcasar-replication-ssh-keys-management.sh --add -file …**
  - importe la clé publique à partir d'un fichier dans « /home/replication/.ssh/authorized_keys »
- **alcasar-replication-ssh-keys-management.sh --delete -regex …**
  - supprime la clé publique de l'hôte distant (regex) de « /home/replication/.ssh/authorized_keys »
- **alcasar-replication-ssh-keys-management.sh --list|-l**
  - montre les clés publiques importées des hôtes distants dans « /home/replication/.ssh/authorized_keys »
- **alcasar-replication-ssh-keys-management.sh –show-pubkey**
  - montre la clé publique locale stockée dans « /root/.ssh/id_rsa.pub »

8.10.3 - Ajout/suppression d'une réplication (à partir d'un secondaire)

- **alcasar-replication-add.sh --to-primary --name=nom_primaire --address=192.168.182.10 --port=22 --user=replication --db-user=db_replication –db-password=(à récupérer sur primaire dans '/root/ALCASAR-passwords.txt', attribut 'db_replication_pwd')**
  - Récupère les infos des réplications actives (alcasar-replication-list.sh –all) et vérifie qu'il n'y a pas doublon ;
  - Ouvre temporairement le port SSH en sortie du parefeu vers le primaire (pour test) ;
  - Teste la connexion SSH vers le primaire avec le compte « replication » ;
  - Copie le fichier « local-db_replication_pwd.txt » à partir du primaire. Extraction du mdp et suppression de la copie ;
  - Teste la connexion sur la BDD du primaire avec le compte « db_replication » + mdp récupéré ;
  - Récupère la liste des ports réseau déjà en écoute sur le primaire, puis localement
  - Récupère la plage des ports éphémères locaux (inutile sur le primaire, car même distrib)
  - Détermine un port éphémère libre commun
  - Crée une unité systemd (/etc/systemd/system/replication-nom_primaire.service) permettant de maintenir un tunnel SSH vers le primaire avec une redirection de port (port_libre local vers port 3386 distant et vice versa)
  - Active et lance cette unité
  - Crée le REPLICA (SQL) : CHANGE MASTER 'primary_name' TO MASTER_HOST='127.0.0.1', MASTER_PORT=$bind_port, MASTER_USER='$remote_db_user', MASTER_PASSWORD='$remote_db_pwd', MASTER_USE_GTID=replica_pos
  - Fixe la clé « REPLICATION_TO=@IP:port » dans « alcasar.conf »
  - Relance le parefeu (prise en compte de cette sortie SSH)
  - Se connecte au primaire et lance la réplication inverse (alcasar-replication-add.sh --to_secondary)

CHECK :

- *Unité systemd de maintien du canal ssh vers le primaire active et enabled (« systemctl status replication-remote_host »)*
- *Tunnel ssh actif : ss -ntaup |grep ssh*
- *« iptable -nvL OUTPUT » avec une entrée ssh vers remote_host*
- *Entrée « REPLICATION_TO » avec le remote_hote:port dans « /usr/local/etc/alcasar.conf »*
- *Replication configurée, mais non active (alcasar-replication-list.sh --all (Slave_IO_running=No et Slave_SQL_Running=No)*

- **alcasar-replication-delete.sh** --name=nom_primaire|all
  - Arrête et supprime le repliqua
  - Arrête, désactive et supprime l'unité systemd de maintien du tunnel SSH
  - retire « @IP:port, » de la clé « REPLICATION_TO= dans « alcasar.conf »
  - relance le parefeu (*prise en compte de ce retrait)*

- **alcasar-replication-start.sh –name=nom_primaire|all**
  - *SQL : START REPLICA $connexion_name (remote_host)*

CHECK : Replication configurée, et active (alcasar-replication-list.sh --all (Slave_IO_running=Yes et Slave_SQL_Running=Yes)

- **alcasar-replication-stop.sh --name=nom_primaire|all**
  - *SQL : STOP REPLICA $connexion_name (remote host)*

8.10.4 - Ajout /suppression d'une réplication à partir du primaire

- **alcasar-replication-add.sh --to-secondary --name=nom_secondaire --bind-port=port_négocié_phase1 –db-user=db_replication --db-password=(à récupérer sur secondaire dans '/root/ALCASAR-passwords.txt', attribut 'db_replication_pwd')**
  - *Crée le REPLICA (SQL) : CHANGE MASTER 'primary_name' TO MASTER_HOST='127.0.0.1', MASTER_PORT=$bind_port, MASTER_USER='$remote_db_user', MASTER_PASSWORD='$remote_db_pwd', MASTER_USE_GTID=replica_pos*

CHECK : Replication configurée, mais non active (alcasar-replication-list.sh --all (Slave_IO_running=No et Slave_SQL_Running=No)

- **alcasar-replication-delete.sh** --name=nom_secondaire|all
  - Arrête et supprime le repliqua