Author	Ilia Kebets
Reviewer	Thiébaud Modoux
Date	04.06.2019
Version	1

Pryv.io register migration

Procedure

Summary

The register migration procedure only takes into account the master registers. We copy the data from the old master register to the new master register, set the old register to proxy to the new one and enable replication between the 2 so they are synchronized during the DNS propagation phase.

Deploy and launch services on the destination machine

We assume that register is already deployed (config present, docker images downloaded) on the *dest* machine.

Launch services by running `${PRYV_CONF_ROOT}/run-reg-master` and verify that all containers are started using `docker ps` and check logs on `register` and `dns` containers.

(optional) backup Redis data

As we will use Redis replication, it is recommended to backup the database. Make a copy of the data located in `${PRYV_CONF_ROOT}/reg-master/redis/data/`.

Transfer user data

User data migration has a down time which we'll call *cold* migration. To limit its duration, we transfer the bulk of the data from *source* to *dest* prior to the *cold* migration using `rsync`. The *cold* migration consists of syncing the most recent data changes. After this, services will be started on *dest* and the `nginx` process on *source* will proxy calls while DNS entries are updated.

1. Create an SSH key pair using `ssh-keygen -t rsa -b 4096 -C "migration@remote"` and copy the private one to `${PATH_TO_PRIVATE_KEY}` in *dest* and add the public one in `authorized_keys` on *source*.
2. Transfer Redis data: on *dest*, run `time rsync --verbose --copy-links --archive --compress --delete -e "ssh -i ${PATH_TO_PRIVATE_KEY}" ${USERNAME}@${SOURCE_MACHINE}:${PRYV_CONF_ROOT}/reg-master/redis/data/ ${PRYV_CONF_ROOT}/reg-master/redis/data`
3. If needed, Repeat step 2 to sync the biggest bulk of the data prior to the *cold* migration
4. Shutdown services on *source*: `${PRYV_CONF_ROOT}/stop-containers`
5. Make last sync by executing steps 2

If you wish to reactivate service on the *source* machine, simply reboot the stopped services: `${PRYV_CONF_ROOT}/run-reg-master`

Set NGINX proxying

Since the DNS changes will take some time to come into effect, the NGINX process on *source* will be set to proxy to the *dest* machine. The following steps describe the configuration changes to make NGINX proxy calls to the *dest* register. It is advised to comment out the old setting inline using # in order to rollback easily in case of need.

- In `${PRYV_CONF_ROOT}/reg-master/nginx/conf/site-443.conf`, Replace the following:

```
upstream register_server { server register:9000 max_fails=3 fail_timeout=30s; }
```

with

```
upstream register_server { server ${DEST_CORE_IP_ADDRESS}:443; }
```

Change proxy protocol from `http` to `https`

- Change: `http://register_server` to `https://register_server`

Run `${PRYV_CONF_ROOT}/run-reg-master`

As we are currently using docker-compose to specify the mounted volumes (containing the NGINX config), we just boot all services, even if they will be ignored as NGINX is proxying to the *dest* machine.

Set the *source* register as replica of the *dest* register through a SSH tunnel

1. On the *dest* machine, open the Redis container port 6379 to localhost: Add `- "127.0.0.1:6379:6379"` to the `ports` section of the `redis` service in the `${PRYV_CONF_ROOT}/reg-master.yml` docker-compose file and reboot it running `${PRYV_CONF_ROOT}/stop-containers` and `${PRYV_CONF_ROOT}/run-reg-master`
2. Generate SSH key pair `ssh-keygen -t rsa -b 4096 -C "migration@remote"`

3. Copy the public key to `~/.ssh/authorized_keys` of the *dest* register.
4. Copy the private key to the *source* register in `${PRYV_CONF_ROOT}/reg-master/redis/conf` so it is mounted in the container upon startup (this can be any other directory that is mounted in the redis container, this is defined in the `reg-master.yml` docker-compose file)
5. On the *source* register, enter the redis container (`docker exec -ti ${REDIS_CONTAINER_NAME} /bin/bash`), open a SSH tunnel: run `ssh -i ${PATH_TO_PRIVATE_KEY} -L 4567:127.0.0.1:6379 root@${DEST_REG_HOSTNAME} -N`.
6. Set *source* register as replica of *dest* register and add the following to *source* register's redis config file `${PRYV_CONF_ROOT}/reg-master/redis/conf/redis.conf`: `replicaof localhost 4567`
7. Boot services on *source*: `${PRYV_CONF_ROOT}/run-reg-master`

Update Name servers

In your hosting provider (or your own system), set the name servers to the domain name associate to your Pryv.io platform as the *dest* register machines.

Verify

Run a DNS query on the *dest* Register machines and verify that they contain the same data as the *source* ones.

Run `dig @{DEST_REG_MASTER_IP_ADDRESS} USERNAME.DOMAIN` and `dig @{DEST_REG_SLAVE_IP_ADDRESS} USERNAME.DOMAIN`

Finalize

After some time, all DNS requests will be directed to the *dest* register machines. To verify this, take a look at the logs on the *sources* of the `dns` and `register` containers and ensure that they have served no request in ~24 hours.