Garage

A garage server is used for backups.

Summary

The idea of garage is to distribute data within nodes of a cluster and expose an S3 API to populate it. Our node, `marsouin` (a secondary computer made up of old hard drives) is connected to the Rhizome's cluster, comprised of 6 nodes totalling a few TB at the time of writting.

Routing

Garage requires a public IP adress on port 3901 by default, but `marsouin` does not own one. Instead, the port is forwarded from `sagouin` which has one, to `marsouin` thanks to the following iptable rules on `sagouin`:

```
iptables -A PREROUTING -t nat -p tcp -d <PUBLIC_IP> --dport <PORT> -j DNAT --to-destination
<PRIVATE_IP_MARSOUIN>:<PORT>
iptables -A OUTPUT -t nat -p tcp -d <PUBLIC_IP> --dport <PORT> -j DNAT --to-destination
<PRIVATE_IP_MARSOUIN>:<PORT>
iptables -A POSTROUTING -t nat -p tcp --dport <PORT> -j SNAT --to-source=<PRIVATE_IP_SAGOUIN>
iptables -t nat -A POSTROUTING -j MASQUERADE
sysctl net.ipv4.ip_forward=1
```

Basically what they do is preroute incomming packets towards `marsouin` and masquerade its response (the postroute) as if it came from `sagouin`. In addition, the output is for cases where sagou.in itself wants to talk to marsouin. Notice how the PREROUTING and OUTPUT lines are the same after the table name.

The iptables rules are made persistent by putting them in a Yunohost specific bash script that is executed at startup, named

```
/etc/yunohost/hooks.d/post_iptable_rules/99-specific_rules
```

Marsouin storage

`marsouin` is another computer located in the same private network as `sagouin` whose sole utility is to store old, semi-broken hard drives. As such it will be slow and prone to hardware failures. To limit the impact of these issues, garage will be distributed accross all hard drives. To prevent catastrophic failure if one dies, an additionnal SSD stores the OS and garage's metadata.

Old BTRFS way

DO NOT USE, prefer the above method and only keep this section as reference of what used to be done.

BTRFS is used to setup a software RAID :

```
sudo mkfs.btrfs -d raid1c3 -m raid1c3 /dev/sd{a,b,d,e,f}
sudo mount /dev/sda /mnt
```

meaning that a RAID1 with 3 duplications is distributed across 5 hard drives of varying capacities, both for data and metadata storage. As such, 1 to 2 hard drives can fail without much of an impact outside.

The state of the RAID can be checked with

```
btrfs filesystem usage /mnt
```

Garage setup

After installing garage and starting it with a

```
garage server
```

with the following configuration

```
metadata_dir = "/mnt/ssd/garage/meta"
data_dir = "/mnt/ssd/garage/data"
db_engine = "lmdb"

replication_mode = "3"
```

```
rpc_bind_addr = "0.0.0.0:<RPC_PORT>"
rpc_public_addr = "<PUBLIC_IP>:<RPC_PORT>"
rpc_secret = "<SECRET>"
bootstrap_peers = [
    "<ANOTHER_NODE_ID>"
]

[s3_api]
s3_region = "garage"
api_bind_addr = "0.0.0.0:3900"
```

with `ANOTHER_NODE_ID` that come from another node doing `garage node id` and automatically makes the wole cluster available to us. The connection can be verified with

```
garage status
```

Our node then needs to be described to others with

```
garage layout assign OUR_NODE_ID -z ZONE -t TAG -c CAPACITY
garage layout show # To check that everything is right
garage layout apply --version XXX
```

with `ZONE` being vaguely a geographical zone to favor distributing accross zones, `TAG` a simple tag for the user and `CAPACITY` being a vague description of how much space is available. Our current rule of thumb for the capacity is the number of GB in the node, but it could be any integer. The `--version` flag in the `apply` command is simply incremented every time the cluster layout is modified.

S3 Bucket

In order to expose the storage as S3, the following have been done on marsouin:

- Create a new bucket

```
garage bucket create BUCKETNAME
garage bucket set-quotas BUCKETNAME --max-size 200GiB
```

- Create a new key to access the bucket

```
garage key new --name KEYNAME
garage bucket allow --read --write --owner BUCKETNAME --key KEYNAME
```

Révision #3
Créé 2023-09-04 14:46:42 UTC par Antoine Lima
Mis à jour 2023-11-14 23:39:59 UTC par Antoine Lima