Service migration and high availability via Dynamic DNS service

Viet Tran
Institute of Informatics SAS
Slovakia

EGI-ACE receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 101017567.

Motivation of Dynamic DNS service

- VMs in EGI Federated Cloud are usually accessed only via IP addresses
- But hostnames are often required
  - for getting SSL certificates
  - for user-friendly access
  - for hiding complexity of clouds
- Dynamic DNS service was developed to address the issue

Dynamic DNS service

- With Dynamic DNS service, users can:
  - register sensible, memorable hostnames in supported domains
  - attach the hostnames to hosts/VMs in Cloud
  - then access services deployed in the VMs via the hostnames
- Full automation, self-service, immediately available, easy to use
- Independent, no additional requirements (software, support from Cloud providers or site admins)

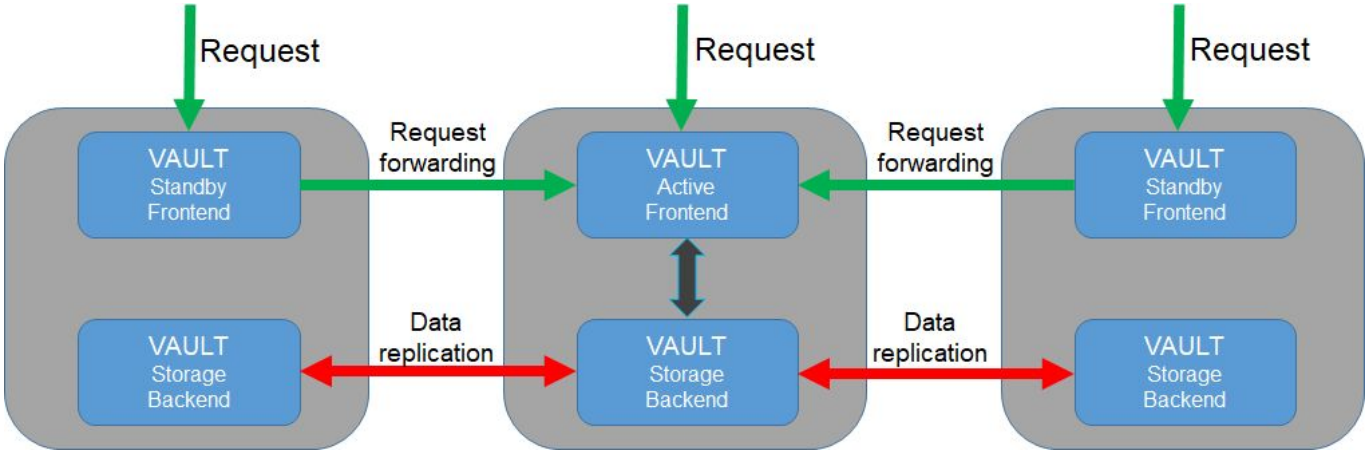
Service migration via Dynamic DNS

- Dynamic DNS can be used for service migration
- Let's see the demo how it works

High availability via Dynamic DNS

A real use case: Secret management service:

- Three servers hosted on different providers: IISAS, INFN and IFCA
- Data replicated automatically for high availability
- But: how users know which server is the health one?
    - Solution: to use Dynamic DNS to assign generic hostname to the healthy one



Code example

https://github.com/tdviet/vault-ha-check-public/blob/main/vault-ha-check.py

```
GENERIC_HOSTNAME = "vault.services.fedcloud.eu"
INSTANCE_HOSTNAMES = ("vault-infn.services.fedcloud.eu", "vault-ifca.services.fedcloud.eu")

   # First, check the health of generic endpoint
    if check_server_health(generic):

        # If OK, nothing to do, print OK message and return OK
        return 0

    else:
        # Instance at generic endpoint is faulty

        # Looking for a healthy instance among the instance list
        for instance in instances:
            if check_server_health(instance):

                # Found a healthy one, updating generic endpoint to it
                return update_generic_endpoint(generic, instance, update_secret)

        # No healthy instance found, print error message and return CRITICAL
        return 2
```

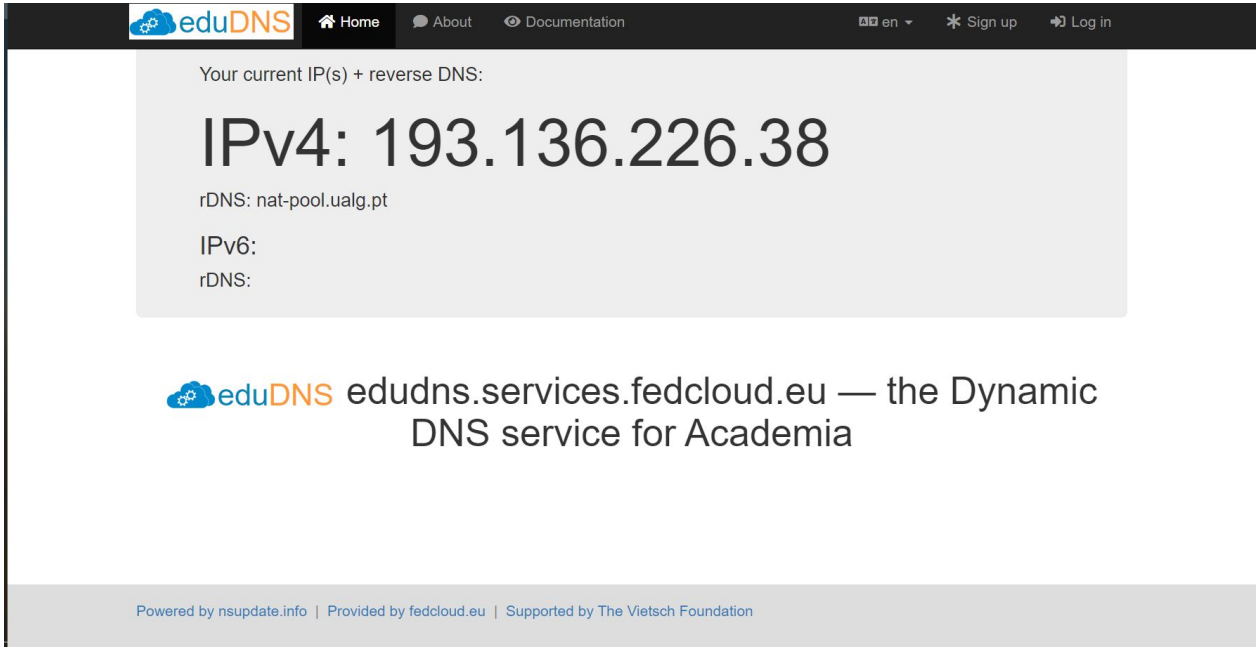
Summary

- Dynamic DNS service is designed for assigning memorable hostnames to services deployed in Cloud

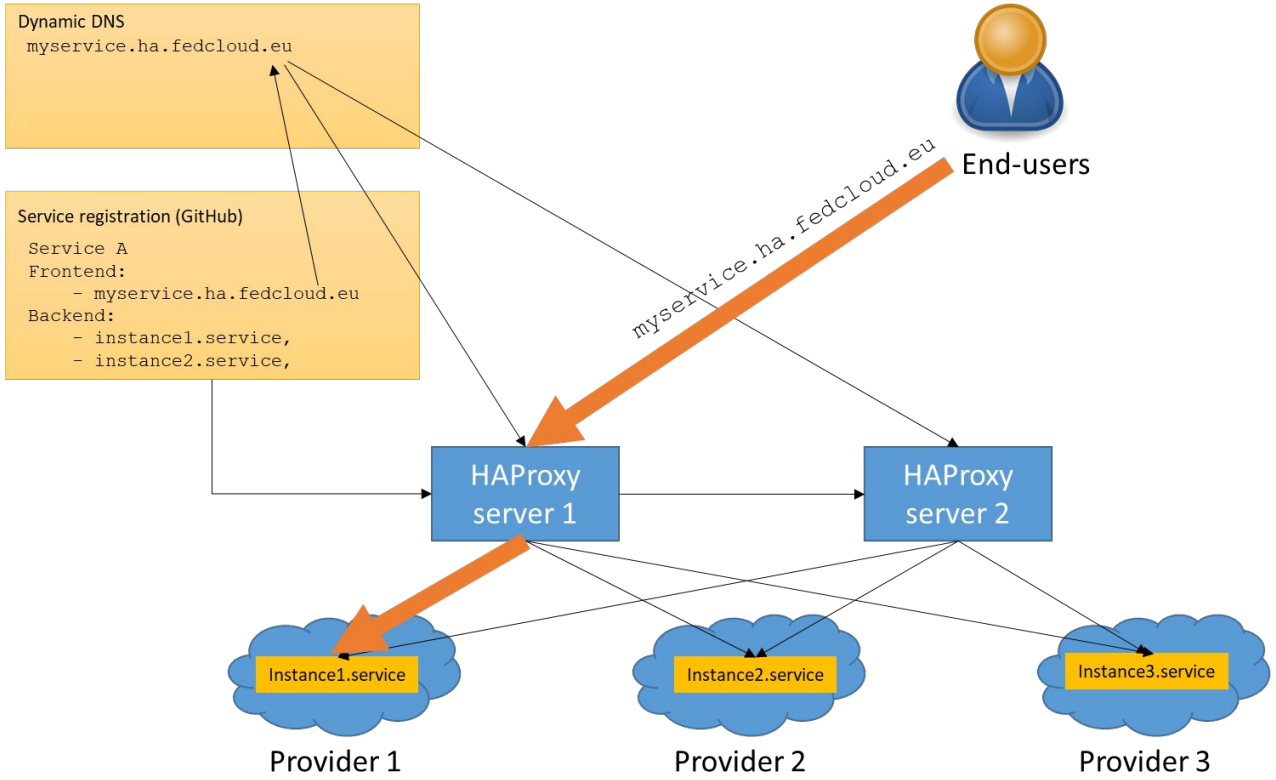
- But there are more useful and interesting uses of Dynamic DNS
  - Testing and development of services locally with the same hostnames
  - Migrating services from local/testing infrastructure to Cloud
  - Migrating services from a Cloud provider to another
  - High availability of services

New deployment: eduDNS - Dynamic DNS for academia

- Available at https://edudns.services.fedcloud.eu/
- Authentication via eduTEAMS



Ongoing work: FedCloud load balancer



Links

- Dynamic DNS service: https://nsupdate.fedcloud.eu/
- Design of secret management service for HA: https://vault.docs.fedcloud.eu/design.html
- Source code for HA checking: https://github.com/tdviet/vault-ha-check-public/blob/main/vault-ha-check.py
- Video demonstration of Dynamic DNS: https://www.youtube.com/watch?v=dk4VYT2VFmU

Thank you for your attention