Dynamic DNS support for EGI Federated Cloud

Viet Tran
Institute of Informatics SAS
Slovakia
Requirements

• More and more services are deployed in Cloud
• In principle, any service hosted in cloud will need a valid DNS hostname
  – For security reasons: getting SSL host certificates
  – For simple deployment: setting hostnames in configuration files
  – For simple access: via hostnames instead of IP addresses

• Supports for DNS in EGI Federated Cloud are very limited and fragmented
  – Some sites have new hostname for every new VM
  – Some have fixed hostname for each IP
  – Some do not support at all
⇒ Random, encrypted and unpredictable hostnames for services hosted in clouds
⇒ Users need to reconfigure servers and clients for every start
FedCloud-wide Dynamic DNS support

- For users: can register hostnames and assign them to VMs:
  - Memorable, sensible hostnames for services hosted in FedCloud
  - Predictable, reusable hostnames for clients/servers setting, no reconfiguration of clients and servers for every start
  - Common domains for VMs in whole federation, independent from location of VMs in FedCloud

- For sites: no changes, no additional support required

- For FedCloud: promotion of federated approach
Architecture

Users

EGI CheckIn
Authentication

Register hostnames
Assign/update IPs

NS-update server

DDNS command-line client

BIND9 DNS server

updating DNS records

DNS server
Standards and compliance

• All communications between components are supported by standards:
  – OIDC authentication (NS-update – EGI CheckIn)
  – dyndns2 protocol (command-line client – NS-update)
  – RFC 2136 standard (NS-update – BIND9 server)
  ⇒ Implementation-independent

• All relationships are many-to-many, e.g.
  – One NS-update server – many OpenID identity providers
  – One NS-update server – many DNS servers (one portal for many domains)
  – Many NS-update servers – one DNS server (many portal for one domain)
  ⇒ Sustainable and extensible
Implementation

• Developed and hosted at IISAS

• Integrated with EGI CheckIn

• Can be used everywhere (VMs in EGI FedCloud, commercial clouds, local servers)

• No requirements on target clouds

• Simple, secure IP assignment/updating without user credentials
Usage

• Log into https://nsupdate.fedcloud.eu/ via EGI CheckIn and register a hostname

• Assign/update an IP address to the hostname
Registering hostnames

• Support multiple domains/subdomains
  – By default, subdomains from organization-neutral domain fedcloud.eu
  – Support for other domains/subdomain on requests

• Web portal with authentication via EGI CheckIn
  – Just log in and register a hostname in an available domain
  – Server will give error message if the hostname is occupied by other users
  – See instruction/demo at the end of presentation
Assigning/updating IPs

• Each registered hostname has its own secret (password)
  – Once hostname is registered, no user credential is needed for assigning/updating IP address
  ⇒ No user credential stored on VMs, just hostname and its own secret

• Very simple usage
  – For most users, just simple command on the host/VM
    “curl https://HOSTNAME:SECRET@nsupdate.fedcloud.eu/nic/update”
  – Can be included into any deployment tools (e.g. cloud-init)
  – Compatible with existing DDNS clients
  – Users can do it without knowing the IP address of the VM
Usage scenarios

• Testing and development
  – Register a hostname and assign it to local server
  – Develop/configure/test the service on local server using the hostname
  – Move the service to cloud and update hostname
  – No additional changes in configurations (both clients and servers)
Usage scenarios

• Configuring clients before deploying services
• Migrating services between different cloud sites without changing clients
• Getting host certificates
• Configuration of connected services
• ...
• Simply, no need to remember IP address anymore
NS-update portal (1)
NS-update portal (2)
NS-update portal (3)
Host Secret Generated

New secret generated for you. We store it hashed, so save it now, or you have to generate a new one again. Everytime you visit this page a new secret will be generated and the old one becomes invalid.

Secret: KZpvFEpDEm

How to configure automated dynamic DNS updates?

General configuration hints

Usually, you should configure one system on your network to update the DNS. This can be either your router or a host on your network and it will run some software called the update client.

We have some specific configuration hints for some devices and update clients, please see the tabs above. If you don't find yours there, use the generic hints below.

Important note about security and compatibility: For update URLs, we always give the https (not: http) URL as that will use an encrypted connection to transfer your data (including your update secret). Depending on your update client (router firmware or PC software) and the specific nspupdate info-based service, https might be not supported. So, if it does not work, you have the choice of using a different update client and/or a different service, or transmitting your data using an unencrypted connection (using http instead of https).

On fedcloud.eu https is supported.

Your update client needs to access the following URLs to update the DNS:

For IPv4 updates:

https://demo.test.fedcloud.eu.KZpvFEpDEm@nsupdate.fedcloud.eu/nic/update
Assigning/updating IP to the hostname

• Just single command on the VM
  curl https://HOSTNAME:SECRET@nsupdate.fedcloud.eu/nic/update

• Or anywhere else
  curl https://HOSTNAME:SECRET@nsupdate.fedcloud.eu/nic/update?hostname=HOSTNAME&myip=ipaddress

• No additional client, no user credential, no dependency on IP address if running on target server
Integration with deployment tools

• Easy to integrate with other deployment tools like Ansible, cloud-init, ...
• Example of cloud-init file

```bash
#cloud-config
users:
  - name: cloudadm
    sudo: ALL=(ALL) NOPASSWD:ALL
    lock-passwd: true
    ssh-import-id: cloudadm
    ssh-authorized-keys:
      - ssh-rsa AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
runcmd:
  - curl "https://HOSTNAME:SECRET@nsupdate.fedcloud.eu/nic/update"
```
Integration with deployment tools
Summary

- Dynamic DNS service is
  - Simple, easy to use
  - Easy to integrate with existing deployment tools/services
  - No additional requirements/supports from cloud sites
  - No user credential stored in cloud for updating IP

- For service developers/providers, Dynamic DNS service can
  - Simplify service configuration
  - Enable securing services with SSL host certificates
  - Enable migrating services from local servers to clouds and between clouds without reconfiguring service URL and client tools

- For users
  - No need to use of IP addresses for accessing services in cloud

- For cloud providers
  - Nothing need to do