

Supporting Native Cloud Tools in the EGI Federated Cloud via the FedCloud Client

Viet Tran viet.tran@savba.sk
Institute of Informatics, Slovak Academy of Sciences
Slovakia

EGI Federated Cloud

- Federation of 27 OpenStack sites across Europe
- Unified authentication/authorization infrastructure:
 - Login via OpenID Connect from EGI Checkin
 - Authorization via virtual organizations (VO)
- But very heterogeneous resources among sites
 - Different VM image names/IDs
 - Different project names/IDs
 - Different flavor names/IDs
 - ...
- Writing a script that works over federation is difficult

Motivation of FedCloud client

- Most native cloud tools (OpenStack CLI, Terraform, rclone, ...) cannot work on the federation directly:
 - Designed for single site
 - Not able to deal with service registry
 - Not familiar with the site/VO concept
 - Site have different configurations (flavors, images, ...)
- FedCloud client make thing simple:
 - Creating working environment
 - Making the tool federation-aware
 - Selecting resources

Very quick introduction of FedCloud client

Setting up FedCloud client

- Installing FedCloud client

```
$ pip install fedcloudclient
```

- Setting OIDC access token (or mytoken, oidc-agent)

```
$ export OIDC_ACCESS_TOKEN=xxxxxxxxxxxxxxxxxxxx
```

(or

```
$ export FEDCLOUD_MYTOKEN=xxxxxxxxxxxxxxxxxxxx)
```

- That is all

Listing sites in EGI Federated Cloud

```
$ fedcloud site list
```

```
100IT
```

```
BIFI
```

```
CESGA-CLOUD
```

```
CESGA
```

```
...
```

Listing VO memberships

```
$ fedcloud token list-vos
```

```
biomed
```

```
demo.fedcloud.egi.eu
```

```
fedcloud.egi.eu
```

```
msswss.ui.savba.sk
```

```
vo.access.egi.eu
```

```
vo.ai4eosc.eu
```

Executing Openstack commands

```
$ fedcloud openstack image list --site IFCA-LCG2 --vo vo.ai4eosc.eu
```

```
Site: IFCA-LCG2, V0: vo.ai4eosc.eu, command: image list
```

| ID | Name | Status |
|--------------------------------------|---------------|--------|
| 9082297e-d1cf-46a2-baaa-c09ea040ab75 | AlmaLinux-9.1 | active |
| 3da4053f-34cb-4266-bb02-5f30987d9a91 | ... | |

Execute a command on all sites

```
$ fedcloud openstack server list -i -c ID -c Name --site ALL_SITES --vo vo.ai4eosc.eu
```

```
Site: IISAS-FedCloud, VO: vo.ai4eosc.eu, command: server list -c ID -c Name
```

```
+-----+-----+
| ID                | Name                |
+-----+-----+
| 44a93c0c-7e0b-401e... | nomad-ai4eosc-wn-a00f2202-4d85-11ee-b620-9aee49d2abae |
| 007fa7f6-39d4-4096 ... | nomad-ai4eosc-front-98b8c04e-4d85-11ee-b620-9aee49d2abae |
```

```
Site: IFCA-LCG2, VO: vo.ai4eosc.eu, command: server list -c ID -c Name
```

```
+-----+-----+
| ID                | Name                |
+-----+-----+
| 36d89087-6b0e-429a-... | host-ifca-dc1-server2 |
| 472f4bb3-ca66-4227-... | host-ifca-dc1-cli2   |
```

Scripting and programming

FedCloud client is designed for automation:

- Native JSON outputs
- Support for different shells (Windows, Linux)
- Customizable configuration files
- Directly usable as a Python library

Setting up working environment for external tools

Setting up working environment (1)

- Setting a working environment for external tools is easy

```
$ fedcloud site env --site IFCA-LCG2 --vo vo.ai4eosc.eu
export OS_AUTH_URL="https://api.cloud.ifca.es:5000/v3/";
export OS_AUTH_TYPE="v3oidcaccessstoken";
export OS_IDENTITY_PROVIDER="egi.eu";
export OS_PROTOCOL="openid";
export OS_PROJECT_ID="f44e296a9ea441548456d25fb5b467c9";
export OS_ACCESS_TOKEN="..."
```

Setting up working environment (2)

- In scripts

```
# Setting environment
```

```
eval `fedcloud site env --site IFCA-LCG2 --vo vo.ai4eosc.eu`
```

```
# Then executing some native commands
```

```
openstack image list
```

Selecting resources

Selecting resources

- Deployment tools like OpenStack CLI, Terraform need different kind of parameters: flavors, images, networks in their configurations
- The names of the resources are different from sites to sites

=> Need to change configuration when moving to another site

FedCloud client can help to automate that

Selecting flavors

```
$ fedcloud select flavor --site IFCA-LCG2 --vo  
vo.access.egi.eu --flavor-specs "VCPU>=2" --output-format  
list
```

cm14-compute-2

cm14-compute-4

cm14-compute-8

cm14-compute-16

cm14-compute-32

Selecting flavors (2)

```
$ fedcloud select flavor --site IFCA-LCG2 --vo  
vo.access.egi.eu --flavor-specs "VCPU<=2" --output-format  
first  
cm14-compute-2
```

Selecting flavors (3)

```
# Setting environment
```

```
$ TF_VAR_FLAVOR = `fedcloud select flavor --site IFCA-LCG2  
--vo vo.access.egi.eu --flavor-specs "VCPU<math>\geq 2</math>"  
--output-format first`
```

```
# And update configuration files if needed
```

```
$ sed -i "s/FLAVOR/$TF_VAR_FLAVOR/" terraform.tfvars
```

Selecting images

```
$ fedcloud select image --site IFCA-LCG2 --vo vo.ai4eosc.eu  
--image-specs "Name =~ '22.04'" --output-format list
```

```
IFCA Ubuntu 22.04 [2023-08-29]
```

```
IFCA Ubuntu 22.04 [2023-07-29]
```

```
...
```

Summary

Summary

FedCloud client simplifies the configuration and operation of native cloud tools on EGI Federated Cloud

- Powerful scripting capabilities
- Setting working environment
- Selecting resources

Some real examples of using FedCloud client

<https://github.com/EGI-Federation/fedcloud-dashboard/blob/main/deployment/site-config.sh#L15-L19>

<https://github.com/EGI-Federation/fedcloud-catchall-operations/blob/main/.github/workflows/deploy.yml#L47-L54>

(Credit: EGI team)

More information

<https://fedcloudclient.fedcloud.eu/usage.html>

<https://fedcloudclient.fedcloud.eu/cheat.html>