EGI Federated Cloud Integration Documentation

Enol Fernandez <enol.fernandez@egi.eu>

Mar 04, 2020

Contents:

This documentation covers how to join the EGI Cloud federation as a provider. If you are interested in joining please first contact EGI operations team (`operations _at_ egi.eu`), expressing interest and providing few details about:

- the projects you may be involved in as cloud provider
- the user communities you want to support (a.k.a. Virtual Organisations, VO). You can also support the 'long-tail of science' through the access.egi.eu VO.
- the technologies (Cloud Management Framework) you want to provide.
- details on the current status of your deployment (to be installed or already installed, already used or not, how it is used, who uses the services,...)

EGI will provide proper guidance through all the following steps until your service gets certified.

CHAPTER 1

Requirements

IaaS providers are very welcome to join the EGI Federated Cloud as a Resource Centres (RC) and joining the Federated Cloud Task Force to contribute to the design, creation and implementation of the federation.

Resource Centers are free to use any Cloud Management Framework (OpenNebula, OpenStack, etc...) as long as they are able to integrate with the EGI Federation components as described in the Federated Cloud Architecture. At the moment this compliance is guaranteed for the following CMFs:

- OpenStack (with/without OCCI)
- OpenNebula with OCCI
- Synnefo with OCCI

The general minimal requirements are:

- Hardware requirements greatly depend on your cloud infrastructure, EGI components in general do lightweigth operations by interacting with your services APIs.
  - `cloudkeeper` requires enough disk space to download and convert images before uploading into your local catalogue. The number and size of images which will be downloaded depends on the communities you plan to support. For the piloting VO `fedcloud.egi.eu`, 100GB of disk should be enough.
- Servers need to authenticate each other in the EGI Federated Cloud context using X.509 certificates. So a Resource Centre should be able to obtain server certificates for some services.
- User and research communities are called Virtual Organisations (VO). Resource Centres are expected to join:
  - `ops` and `dteam` VOs, used for operational purposes as per RC OLA
  - a community-VO that supports EGI users (e.g. `fedcloud.egi.eu` for piloting)
- EGI provides packages for the following operating systems (others may work but we are not providing packages):
  - CentOS 7 (and in general RHEL-compatible)
  - Ubuntu 16.04 (and in general Debian-based)

CHAPTER 2

Integration

Integration of cloud stacks into EGI FedCloud follows a well-defined path, with certain steps which need to be taken, depending on the cloud stack in question. By integration here, we refer to the proper interoperation with EGI infrastructure services such as accounting, monitoring, authentication and authorisation, *etc*. These configurations make your site discoverable and usable by the communities you wish to support, and allow EGI to support you in operational and technical matters.

Integration of these services implies specific configuration actions which you need to take on your site. These aim to be unintrusive and are mostly to facilitate access to your site by the communities you wish to support, without interfering with normal operations. This can be summarised essentially as :

1. Network configuration
2. Permissions configuration
3. AAI configuration
4. Accounting configuration
5. Information system integration
6. VM and appliance repository configuration

If at any time you experience technical difficulties or need support, please open a ticket or discuss the matter with us on the forum

You can follow dedicated integration guides for each cloud management frameworks:

2.1 OpenNebula

EGI Federated Cloud Site based on OpenNebula is an ordinary OpenNebula installation with some EGI-specific integration components. There are no additional requirements placed on internal site architecture. Follow OpenNebula documentation if you need advice on how to install and configure OpenNebula itself.

Supprted OpenNebula versions:

- OpenNebula v5.2.x

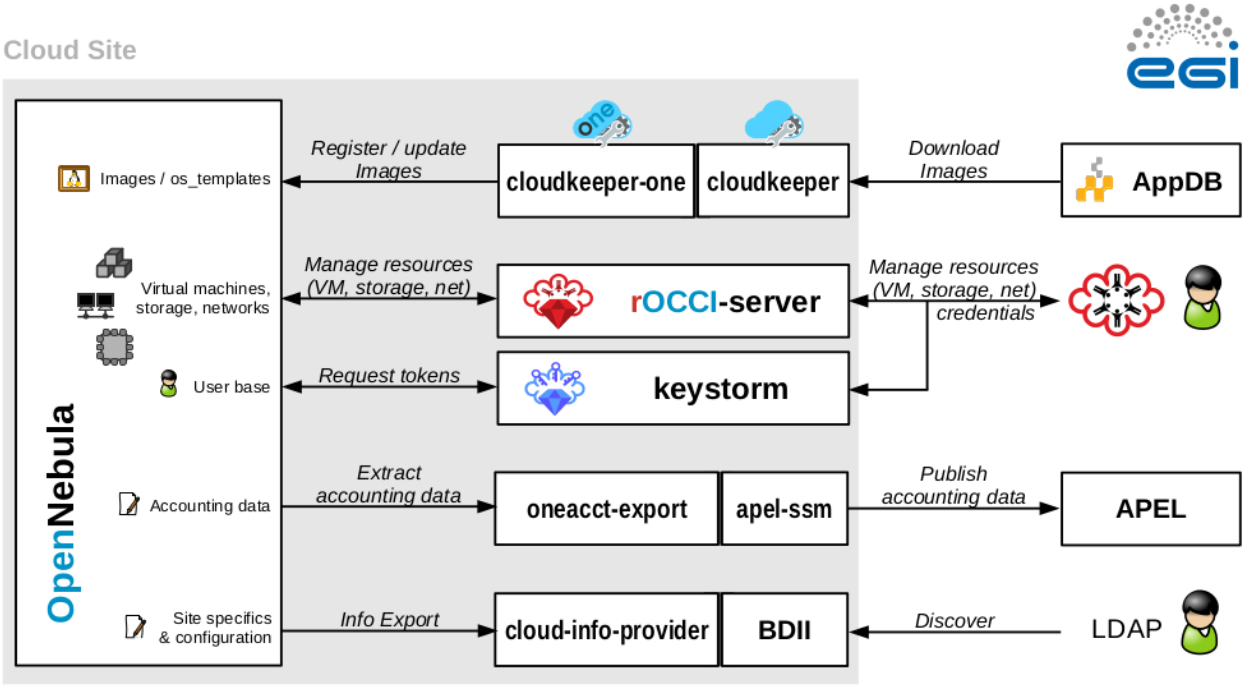
- OpenNebula v5.4.x

Integration Prerequisites:

- Working OpenNebula installation.
- Valid IGTF-trusted host certificates for selected hosts.

Please consider that:

- **CDMI** storage endpoints are currently **not supported** for OpenNebula-based sites.
- OpenNebula GUI integration is **not** supported.



The following **components** must be installed:

- **rOCCI-server** – provides a standard virtual machine management interface.
- **keystorm** – serves federated authentication and authorization.
- **cloudkeeper** and **cloudkeeper-one**, synchronize site with appliances from AppDB.
- **oneacct-export** and **apel-ssm** – collect accounting and publishe it into EGI's accounting database.
- **cloud-info-provider** and **BDII**, register site in the EGI Information System.

2.1.1 Open Ports

The following **ports** must be open to allow access to an OpenNebula-based FedCloud site:

Port	Application	Host	Note
2633/TCP	**OpenNebula**/XML-RPC	**OpenNebula**	Communication between integration components and OpenNebula.
2170/TCP	**BDII**/LDAP	**cloud-info-provider/BDII**	EGI Service Discovery/Information System.
11443/TCP	**OCCI**/HTTPS	**rOCCI-server**	EGI Virtual Machine Management.
5000/TCP	**keystorm**/HTTPS	**keystorm**	EGI User Management.
50505/TCP	**cloudkeeper**/HTTP	**cloudkeeper**	EGI Image Management, needs to be accessible from **cloudkeeper-one** node only
50051/TCP	**cloudkeeper-one**/gRPC	**cloudkeeper-one**	EGI Image Management, needs to be accessible from **cloudkeeper** node only

There are no additional requirements for **OpenNebula** hosts used to run virtual machines.

2.1.2 Service accounts

This is an overview of **service accounts** used in an OpenNebula-based site. The names are default and can be changed if required.

Type	Account name	Host	Use
System accounts	`rocci`	rOCCI-server	Service account for **rOCCI-server**. It is only a service account, no access required.
	`keystorm`	keystorm	Service account for **keystorm**. It is only a service account, no access required.
	`apel`	oneacct-export/APEL	Service account for **oneacct-export/APEL**. Just a service account, no access required.
	`openldap`	cloud-info-provider/BDII	Service account for **cloud-info-provider/BDII**. Just a service account, no access required.
	`cloudkeeper`	cloudkeeper	Service account for **cloudkeeper**. Just a service account, no access required.
	`cloudkeeper-one`	cloudkeeper-one	Service account for **cloudkeeper-one**. Just a service account, no access required.

2.1.3 EGI Virtual Machine Management

Prerequisites

Enable EPEL and install the following packages prior to installation:

```
yum install -y epel-release wget
```

Installation

rOCCI-server is distributed as package for multiple Linux distributions which is available in AppDB. This guide will expect CentOS 7 distribution but installation on any other supported distribution is very similar.

- Register `rOCCI-server` repositories

```
wget http://repository.egi.eu/community/software/rocci.server/2.x/releases/repofiles/
↪sl-7-x86_64.repo -O /etc/yum.repos.d/rocci-server.repo
```

- Install package

```
yum install -y occi-server
```

Configuration

- Make rOCCI-server listen on a public interface

```
mkdir -p /etc/systemd/system/occi-server.socket.d
cat > /etc/systemd/system/occi-server.socket.d/override.conf <<EOS
[Socket]
# lines below are NOT duplicated by mistake
ListenStream=
ListenStream=0.0.0.0:11443
EOS
```

```
sed -i 's/HOST=127.0.0.1/HOST=0.0.0.0/g' /etc/occi-server/variables
```

- Uncomment and configure optional parameters in */etc/occi-server/variables*

```
export HOST_CERT=/path/to/cert                                    # host certificate␣
↪readable by the rocci user
export HOST_KEY=/path/to/key                                      # host key␣
↪readable by the rocci user
```

```
export ROCCI_SERVER_KEYSTONE_URI=https://localhost:5000/          # URL pointing to␣
↪keystorm installation
```

```
export ROCCI_SERVER_OPENNEBULA_ENDPOINT=http://localhost:2633/RPC2 # URL pointing to␣
↪OpenNebula installation
```

```
export ROCCI_SERVER_ENCRYPTION_TOKEN_CIPHER=                      # crypto options␣
↪MUST MATCH keystorm's crypto options, see /etc/keystorm/variables
export ROCCI_SERVER_ENCRYPTION_TOKEN_KEY=                         # crypto options␣
↪MUST MATCH keystorm's crypto options, see /etc/keystorm/variables
export ROCCI_SERVER_ENCRYPTION_TOKEN_IV=                          # crypto options␣
↪MUST MATCH keystorm's crypto options, see /etc/keystorm/variables
```

- Enable and start the service

```
systemctl enable occi-server
systemctl start occi-server
```

Runtime

- Import resource templates to OpenNebula

```
/opt/occi-server/bin/oneresource create --endpoint http://one.example.org:2633/RPC2 #␣
↪--username PRIVILEGED_USER --password PASSWD
# re-run with `--resources /opt/occi-server/embedded/app/rOCCI-server/lib/resources/
↪gpu/` to enable GPU resource templates
```

(continues on next page)

(continued from previous page)

- In OpenNebula, set flags for groups by adding attributes:

```
DEFAULT_CLUSTER_ID="0"                # Default cluster for this group
DEFAULT_CONNECTIVITY="public"         # Default connectivity for this group:␣
↪public|nat|private
```

- In OpenNebula, set network type on networks used via OCCI by adding an attribute:

```
NETWORK_TYPE="public"                 # Supported types: public|nat|private
```

- In OpenNebula, set flag for networks that should be treated as public IP pools (for IP reservations) by adding an attribute:

```
FLOATING_IP_POOL="yes"
```

- In OpenNebula, set additional network attributes:

```
NETWORK_ADDRESS=""                    # e.g., "172.16.100.0"
NETWORK_MASK=""                       # e.g., "255.255.255.0"
GATEWAY=""                            # e.g., "172.16.100.1"
```

Migration from v1 to v2

In order to migrate from rOCCI-server v1 with Perun-managed user accounts, perform the following steps.

Preparation

- Disconnect direct propagation (slave scripts)
- Remove all user accounts that do not have any resource allocations

Migration

- Merge multiple single-group accounts into one account with multiple groups

```
Single-group accounts owned by the same person can be identified as having:

* `NAME` following the naming convention $VONAME_$ID where the same user always has␣
↪the same $ID
* `TEMPLATE/X509_DN` where the same user always has the same DN
```

```
Name of the merged user MUST be a SHA256 digest of the `TEMPLATE/X509_DN` attribute␣
↪value.

In ruby, SHA256 digest can be generated as:

require 'digest'
Digest::SHA256.hexdigest 'DN_STRING_HERE'
```

- Manually add user attributes

```
For each user, add the following attributes:

* TEMPLATE/ID
* TEMPLATE/NAME
* TEMPLATE/IDENTITY
* TEMPLATE/AUTHENTICATION

Where

* `TEMPLATE/ID` is a SHA256 digest of the `TEMPLATE/X509_DN` attribute value
* `TEMPLATE/IDENTITY` and `TEMPLATE/NAME` contain the old `TEMPLATE/X509_DN` value
* `TEMPLATE/AUTHENTICATION` is a static value 'voms'
```

- *chown* all user-owned resources to the new user

2.1.4 EGI User Management

Prerequisites

Enable EPEL and install the following packages prior to installation:

```
yum install -y epel-release wget
```

Installation

keystorm is distributed as package for multiple Linux distributions which is available in AppDB. This guide will expect CentOS 7 distribution but installation on any other supported distribution is very similar.

- Register `keystorm` repositories

```
wget http://repository.egi.eu/community/software/keystorm/1.x/releases/repofiles/sl-7-
↪x86_64.repo -O /etc/yum.repos.d/keystorm.repo
```

- Install package

```
yum install -y keystorm
```

Configuration

- Uncomment and configure optional parameters in */etc/keystorm/variables*

```
export KEYSTORM_OPENNEBULA_ENDPOINT=http://localhost:2633/RPC2      # URL pointing to␣
↪OpenNebula installation
export KEYSTORM_OPENNEBULA_SECRET=oneadmin:opennebula               # Privileged␣
↪OpenNebula credentials (with user and group management permissions)
```

- Enable and start the service

```
systemctl enable keystorm
systemctl start keystorm
```

- Configure Apache2/httpd

```
# on Ubuntu/Debian only
a2enmod ssl && \
  a2enmod headers && \
  a2enmod proxy && \
  a2enmod proxy_http && \
  a2enmod remoteip && \
  a2enmod auth_openidc && \
  a2enmod zgridsite
```

```
# make sure the following files exist
SSLCertificateFile /etc/grid-security/hostcert.pem
SSLCertificateKeyFile /etc/grid-security/hostkey.pem

# make sure the following directory exists
SSLCACertificatePath /etc/grid-security/certificates
```

- Enable and start Apache2/httpd

```
# on Ubuntu/Debian only
systemctl enable apache2
systemctl restart apache2
```

```
# on CentOS/SL only
systemctl enable httpd
systemctl start httpd
```

- Enable support for EGI VOs via VOMS: VOMS configuraton
- Enable support for EGI VOs via OIDC: *TBD*

Runtime

- In OpenNebula, create empty groups for *fedcloud.egi.eu*, *ops*, and *dteam* with group attribute:

```
KEYSTORM="YES"                  # Allow keystorm to manage membership for this group
```

2.1.5 EGI Accounting

Prerequisites

`oneacct-export` uses **Secure Stomp Messenger** to send accounting records to the central repository. Please, refer to `ssm` documentation for installation instructions. By default, accounting records are placed in `/var/spool/apel/outgoing/00000000`. You **have to** configure and run `ssmsend` periodically, this is not handled by oneacct-export.

Enable EPEL and install the following packages prior to oneacct-export installation:

```
yum install -y epel-release wget
```

Installation

oneacct-export is distributed as package for multiple Linux distributions which is available in AppDB. This guide will expect CentOS 7 distribution but installation on any other supported distribution is very similar.

- Register `oneacct-export` repositories

```
wget http://repository.egi.eu/community/software/oneacct.export/0.4.x/releases/
↪repofiles/sl-7-x86_64.repo -O /etc/yum.repos.d/oneacct-export.repo
```

- Install package

```
yum install -y oneacct-export
```

Configuration

- Edit `/etc/oneacct-export/conf.yml`

```
apel:
  site_name: Undefined                    # Usually a short provider name, e.g.␣
↪CESNET
  cloud_type: OpenNebula                  # CMF type, only OpenNebula is supported
  endpoint: https://localhost.edu:11443/ # Public URL of your OCCI endpoint

xml_rpc:
  secret: oneadmin:opennebula            # OpenNebula credentials, privileged
  endpoint: http://localhost:2633/RPC2 # OpenNebula XML RPC endpoint
```

- Add the following lines to `/etc/one/oned.conf` and restart OpenNebula

```
INHERIT_IMAGE_ATTR = "VMCATCHER_EVENT_AD_MPURI"
INHERIT_IMAGE_ATTR = "VMCATCHER_EVENT_DC_IDENTIFIER"
INHERIT_IMAGE_ATTR = "VMCATCHER_EVENT_IL_DC_IDENTIFIER"
INHERIT_IMAGE_ATTR = "VMCATCHER_EVENT_SL_CHECKSUM_SHA512"
INHERIT_IMAGE_ATTR = "VMCATCHER_EVENT_HV_VERSION"
```

- Set benchmark values on CLUSTERs (applies to all hosts in the cluster) or HOSTs (only for that host) in OpenNebula

```
BENCHMARK_TYPE  = "HEP-SPEC06" # benchmark type
BENCHMARK_VALUE = "84.46"      # represents a per-core measured value of said␣
↪benchmark
```

- Use `/etc/oneacct-export/groups.include` or `/etc/oneacct-export/groups.exclude` to control which information gets exported. Specify one group name per line.

Usage

- Enable and register service 'redis'

```
service redis start
chkconfig redis on
```

- Enable and register service 'oneacct-export-sidekiq'

```
service oneacct-export-sidekiq start
chkconfig oneacct-export-sidekiq on
```

- Perform the first export manually

```
# This process may take a long time, consider using **tmux** or **screen**
sudo -u apel /usr/bin/oneacct-export-cron --all
```

- Enable and register service 'oneacct-export-cron'

```
service oneacct-export-cron start
chkconfig oneacct-export-cron on
```

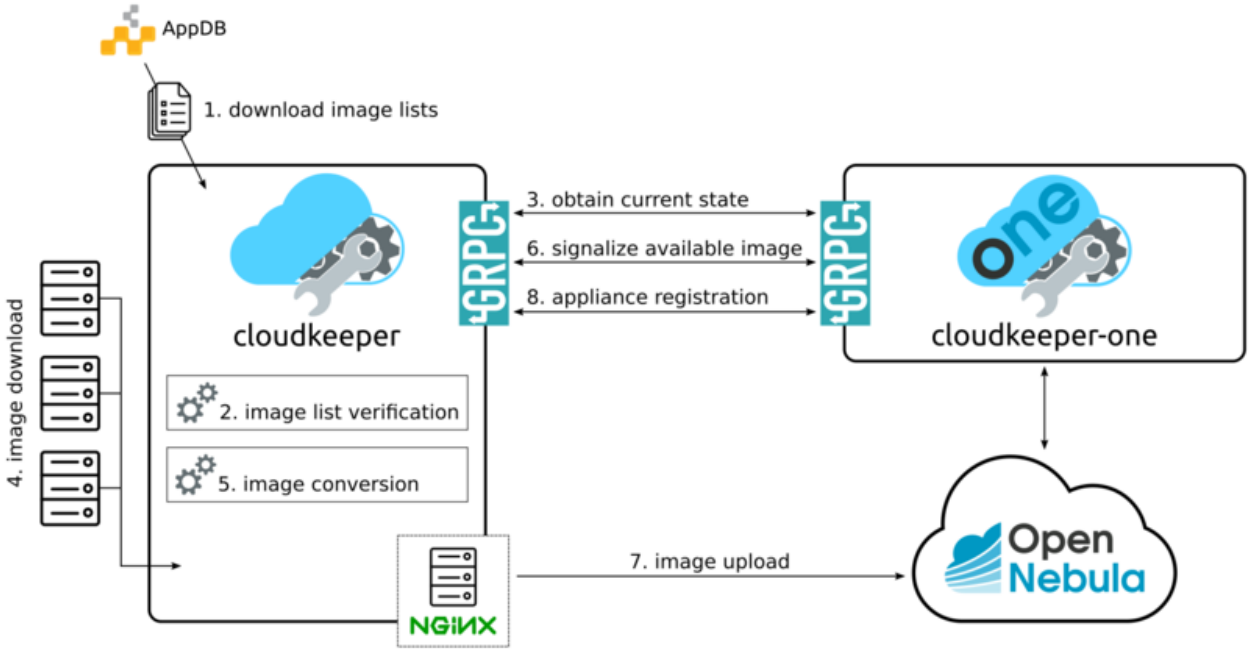
This service registers a cron job which will run oneacct-export every 2 hours.

2.1.6 EGI Information System

Sites must publish information to EGI information system which is based on BDII. There is a common bdii provider for all cloud management frameworks. Information on installation and configuration is available in the cloud-bdii-provider README.md and in the FedClouds BDII instructions, there is a specific section with OpenNebula details.

2.1.7 EGI VM Image Management

cloudkeeper and cloudkeeper-one are tools used to ensure synchronization of virtual appliances with an OpenNebula-based cloud.



Prerequisites

`cloudkeeper` uses VO-wide image lists provided by AppDB to synchronize virtual appliances to clouds. In order to use VO-wide image lists you need to have a valid access token to AppDB. Check how to access to VO-wide image lists and how to subscribe to a private image list documentation for more information.

- Install recent `qemu-img` and `wget`

```
yum install -y centos-release-qemu-ev wget sudo
```

Installation

Both `cloudkeeper` and `cloudkeeper-one` are distributed as packages for multiple Linux distributions which are available in AppDB. This guide will expect CentOS 7 distribution but installation on any other supported distribution is very similar.

- Register `cloudkeeper` and `cloudkeeper-one` repositories

```
wget http://repository.egi.eu/community/software/cloudkeeper/1.x/releases/repofiles/
↪sl-7-x86_64.repo -O /etc/yum.repos.d/cloudkeeper.repo
wget http://repository.egi.eu/community/software/cloudkeeper.one/1.x/releases/
↪repofiles/sl-7-x86_64.repo -O /etc/yum.repos.d/cloudkeeper-one.repo
```

- Install `cloudkeeper` and `cloudkeeper-one`

```
yum install -y cloudkeeper cloudkeeper-one
```

`cloudkeeper` configuration

`cloudkeeper` configuration file can be found in `/etc/cloudkeeper/cloudkeeper.yml`.

image-lists URLs of image lists containing appliances which you want to synchronize to your cloud. Must contain authentication token.

```
image-lists: # List of image lists to sync against
 - https://APPDB_TOKEN:x-oauth-basic@vmcaster.appdb.egi.eu/store/vo/somevo/image.
↪list
 - https://APPDB_TOKEN:x-oauth-basic@vmcaster.appdb.egi.eu/store/vo/othervo/image.
↪list
```

authentication Says whether `cloudkeeper` and `cloudkeeper-one` will communicate securely via TLS. This requires options `certificate`, `key` and `backend->certificate` to be properly set.

image-dir Directory where images will be downloaded and converted before uploading to OpenNebula. Directory is cleaned after each appliance registration/update nonetheless, it should provide sufficient free space (some runs may require up to 200GB of free space).

remote-mode Says whether to serve downloaded images via web server or to copy them locally. Should be `true` especially if OpenNebula is running on different machine than `cloudkeeper` and `cloudkeeper-one`.

nginx->ip-address IP address on which NGINX will serve images in remote mode. This address MUST be accessible from the machine hosting `cloudkeeper-one` and your OpenNebula installation.

formats List of image formats images can be converted to and are supported by the cloud.

`cloudkeeper-one` configuration

`cloudkeeper-one` configuration file can be found in `/etc/cloudkeeper-one/cloudkeeper-one.yml`.

authentication Says whether `cloudkeeper` and `cloudkeeper-one` will communicate securely via TLS. This requires options `certificate`, `key` and `core->certificate` to be properly set.

appliances->tmp-dir Directory images will be copied to before registration in OpenNebula when in non-remote mode.

appliances->template-dir Directory for ERB-enabled templates of OpenNebula images and templates used for registration. More information in the next section.

opennebula->datastores List of OpenNebula datastores images are uploaded to.

opennebula->allow-remote-source Allows OpenNebula to directly download images in remote mode.

Templates configuration

The directory specified by option `appliances->template-dir` contains templates for OpenNebula images and templates in files `image.erb` and `template.erb`. These files can be customized to register images and templates according to your needs. Files are using standard ERB templating mechanism. By default, these files can be found in `/etc/cloudkeeper-one/templates/`.

- `image.erb` available variables:

name Name, under which will the image be registered

appliance Appliance object. Contains following attributes: `identifier`, `title`, `description`, `mpuri`, `group`, `ram`, `core`, `version`, `architecture`, `operating_system`, `vo`, `expiration_date`, `image_list_identifier`, `attributes`.

image Image object. Contains following attributes: `format`, `uri`, `checksum`, `size`

- `template.erb` available variables:

name Name, under which will the template be registered

image_id ID of the previously registered image (same appliance)

appliance Appliance object. Same as for `image.erb`

image Image object. Same as for `image.erb`

For compatibility with other integration components, add the following lines to ‘‘image.rb‘‘:

```
VMCATCHER_EVENT_AD_MPURI="<%= appliance.mpuri %>"
VMCATCHER_EVENT_HV_VERSION="<%= appliance.version %>"
VMCATCHER_EVENT_DC_DESCRIPTION="<%= appliance.description %>"
VMCATCHER_EVENT_DC_TITLE="<%= appliance.title %>"
```

Usage

- Start and enable `cloudkeeper-one` service

```
systemctl enable cloudkeeper-one
systemctl start cloudkeeper-one
```

`cloudkeeper-one` will be now listening for communication from `cloudkeeper`.

- Perform the first synchronization manually

```
# This MAY take a long time, keep checking for successful exit with `systemctl status␣
↪cloudkeeper`
systemctl start cloudkeeper
```

- Start and enable systemd timer for `cloudkeeper`

```
systemctl enable cloudkeeper.timer
systemctl start cloudkeeper.timer
```

This service registers a systemd timer which will run `cloudkeeper` approx. every 2 hours.

2.2 OpenStack

This manual provides information on how to set up a Resource Centre providing cloud resources in the EGI infrastructure. Integration with FedCloud requires a *working OpenStack installation* as a pre-requirement (see http://docs.openstack.org/ for details). Support for OpenStack is provided for the following versions:

- OpenStack Mitaka – LTS under Ubuntu 16.04 (otherwise EOL)
- OpenStack Ocata
- OpenStack Pike
- OpenStack Queens (note that support for Keystone-VOMS is not available, only necessary for legacy VOs)

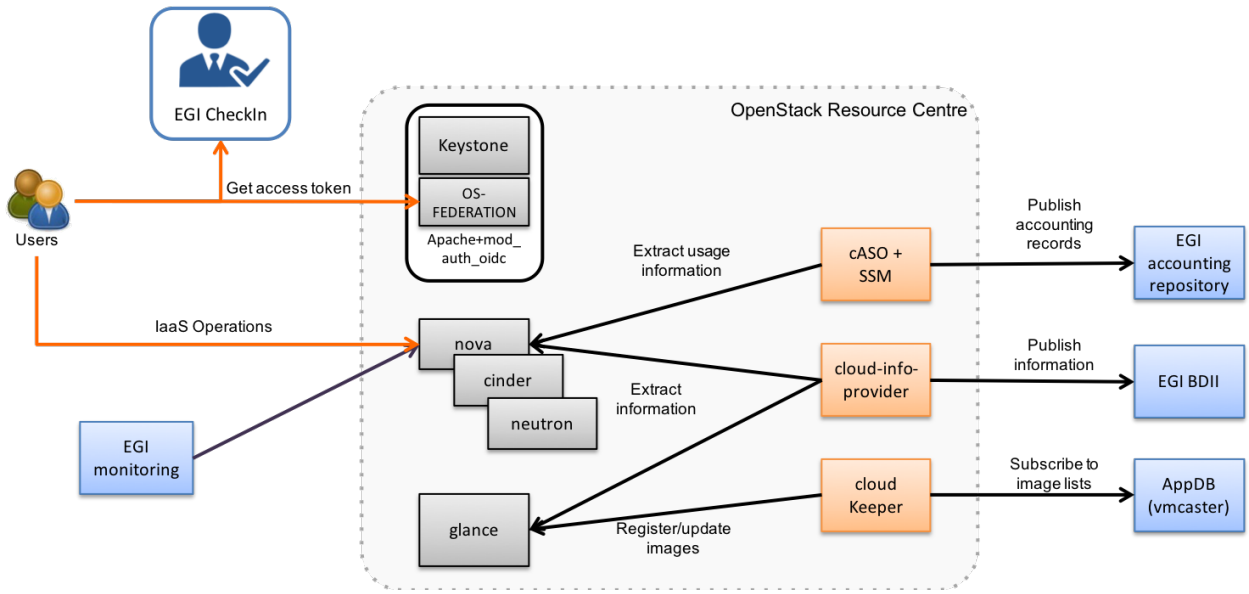
Support for other versions is not guaranteed and they are not recommended in production as they are EOL'd. See http://releases.openstack.org/ for more details on the OpenStack releases.

EGI expects the following OpenStack services to be available and accessible from outside your site:

- Keystone
- Nova
- Cinder
- Glance
- Neutron
- Swift (if providing Object Storage)

FedCloud components are distributed through CMD (Cloud Middleware Distribution) or docker container images available in dockerhub. These docker containers come pre-packaged and ready to use in the EGI FedCloud Appliance so you do not need to install any extra components on your site but just run a VM and configure it appropriately to interact with your services.

The integration is performed by a set of EGI components that interact with the OpenStack services APIs:



- Authentication of EGI users into your system is performed by configuring the native OpenID Connect support of Keystone. Support for legacy VOs using VOMS requires the installation of the **Keystone-VOMS Authorization plugin** to allow users with a valid VOMS proxy to obtain tokens to access your OpenStack deployment.
- **cASO** collects accounting data from OpenStack and uses **SSM** to send the records to the central accounting database on the EGI Accounting service (APEL)
- **cloud-info-provider** registers the RC configuration and description through the EGI Information System to facilitate service discovery
- **cloudkeeper** (and **cloudkeeper-os**) synchronises with EGI AppDB so new or updated images can be provided by the RC to user communities (VO).

Not all EGI components need to share the same credentials. They are individually configured, you can use different credentials and permissions if desired.

Optionally, **ooi (OpenStack OCCI Interface)** translates between OpenStack API and OCCI.

2.2.1 Installation options

EGI distributes the integration components as:

- A Virtual Appliance (VA) that uses Docker containers to bundle all of the components in a single VM and just needs minor configuration to get started
- RPM and DEB Packages in the CMD distribution

FedCloud Virtual Appliance

The EGI FedCloud Appliance is available at AppDB as an OVA file. You can easily extract the VMDK disk by untaring and optionally converting it to your preferred format with qemu-img:

```
# get image and extract VMDK
curl https://cephrgw01.ifca.es:8080/swift/v1/egi_endorsed_vas/FedCloud-Appliance.
↪Ubuntu.16.04-2017.08.09.ova | \
       tar x FedCloud-Appliance.Ubuntu.16.04-2017.08.09-disk001.vmdk
# convert to qcow2
qemu-img convert -O qcow2 FedCloud-Appliance.Ubuntu.16.04-2017.08.09-disk001.vmdk␣
↪fedcloud-appliance.qcow2
```

The appliance running at your OpenStack must:

- Be accessible via public IP with port 2170 open for external connections.
- Have a host certificate to send the accounting information to the accounting repository. DN of the host certificate must be registered in GOCDB with service type `eu.egi.cloud.accounting`. The host certificate and key in PEM format are expected in `/etc/grid-security/hostcert.pem` and `/etc/grid-security/hostkey.pem` respectively.
- Have enough disk space for handling the VM image replication (~ 100GB for `fedcloud.egi.eu` VO). By default these are stored at /image_data. You can mount a volume at that location.

CMD Packages

The CMD-OS repository provides packages that have gone through a quality assurance process for the supported distributions. Follow the the instructions for seting up the repos to install the packages.

2.2.2 Open Ports

The following **services** must be accessible to allow access to an OpenStack-based FedCloud site (default ports listed below, can be adjusted to your installation)

Port	Application	Note
5000/TCP	**OpenStack**/Keystone	Authentication to your OpenStack.
8776/TCP	**OpenStack**/cinder	Block Storage management.
8774/TCP	**OpenStack**/nova	VM management.
9696/TCP	**OpenStack**/neutron	Network management.
9292/TCP	**OpenStack**/glance	VM Image management.
2170/TCP	**BDII**/LDAP	EGI Service Discovery/Information System.
8787/TCP	**OpenStack**/ooi	EGI EGI Virtual Machine Management (optional).

2.2.3 Permissions

This is an overview of the expected account permissions used in an OpenStack site, these accounts can be merged as needed for your deployment:

Component	Permission
cloud-info	Member of all projects supporting EGI VOs
accounting	Member of all projects and able to list users (allowed to `identity:list_users` in keystone)
cloud-keeper	Permission to manage the images for all the projects supporting EGI VOs
Other users	Automatically created by Keystone and permission set as configured in the mappings

2.2.4 EGI AAI

OpenID Connect Support

The integration of OpenStack service providers into the EGI Check-in is a two-step process:

1. Test integration with the development instance of EGI Check-in. This will allow you to check complete the complete functionality of the system without affecting the production Check-in service.
2. Once the integration is working correctly, register your provider with the production instance of EGI Check-in to allow members of the EGI User Community to access your service.

Registration into Check-in development instance

Before your service can use the EGI Check-in OIDC Provider for user login, you must set up a client at https://aai-dev.egi.eu/oidc/manage/#admin/clients in order to obtain OAuth 2.0 credentials and register one or more redirect URIs.

Make sure that you fill in the following options:

- *Main* tab:
  - Set redirect URL to `https://<your keystone endpoint>/v3/auth/OS-FEDERATION/websso/openid/redirect`. Recent versions of OpenStack may deploy Keystone at `/identity/`, be sure to include that in the `<your keystone endpoint>` part of the URL if needed.
- *Access* tab:
  - Enable *openid*, *profile*, *email*, *eduperson_entitlement* and in the **Scope** field

- Enable *authorization code* in the **Grant Types** field
- Enable *Allow calls to the Introspection Endpoint?* in **Introspection** field

Once done, you will get a client id and client secret. Save them for the following steps

Keystone setup

Pre-requisites

1. Keystone must run as a WSGI application behind an HTTP server (Apache is used in this documentation, but any server should be possible if it has OpenID connect/OAuth2.0 support). Keystone project has deprecated eventlet, so you should be already running Keystone in such way.
2. Keystone must be run with SSL
3. You need to install mod_auth_openidc for adding support for OpenID Connect to Apache.

Note: EGI monitoring checks that your Keystone accepts clients with certificates from the IGTF CAs. Please ensure that your server is configured with the correct Certificate and Revocation path:

For Apache HTTPd HTTPd is able to use CAs and CRLs contained in a directory

```
SSLCACertificatePath    /etc/grid-security/certificates
SSLCARevocationPath     /etc/grid-security/certificates
```

For haproxy CA and CRLS have to be bundled into one file.

Client verification should be set as optional otherwise accepted CAs won't be presented to the EGI monitoring.

```
# crt: concatenated cert, key and CA
# ca-file: all IGTF CAs, concatenated as one file
# crl-file: all IGTF CRLs, concatenated as one file
# verify: enable optional X509 client authentication
bind XXX.XXX.XXX.XXX:443 ssl crt /etc/haproxy/certs/host-cert-with-key-and-ca.pem␣
↪ca-file /etc/haproxy/certs/igtf-cas-bundle.pem crl-file /etc/haproxy/certs/igtf-
↪crls-bundle.pem verify optional
```

For nginx CA and CRLS have to be bundled into one file.

Client verification should be set as optional otherwise accepted CAs won't be presented to the EGI monitoring.

```
ssl_client_certificate /etc/ssl/certs/igtf-cas-bundle.pem;
ssl_crl /etc/ssl/certs/igtf-crls-bundle.pem;
ssl_verify_client optional;
```

Managing IGTF CAs and CRLs IGTF CAs can be obtained from UMD, you can find repo files for your distribution at http://repository.egi.eu/sw/production/cas/1/current/

IGTF CAs and CRLs can be bundled using the examples command hereafter.

Please update CAs bundle after IGTF updates, and CRLs bundle after each CRLs update made by fetch-crl.

```
cat /etc/grid-security/certificates/*.pem > /etc/haproxy/certs/igtf-cas-bundle.pem
cat /etc/grid-security/certificates/*.r0 > /etc/haproxy/certs/igtf-crls-bundle.pem
# Some CRLs files are not ending with a new line
# Ensuring that CRLs markers are separated by a line feed
perl -pe 's/----------/-----\n-----/' -i /etc/haproxy/certs/igtf-crls-bundle.pem
```

Apache Configuration

Include this configuration on the Apache config for the virtual host of your Keystone service, using the client id and secret obtained above:

```
OIDCResponseType "code"
OIDCClaimPrefix "OIDC-"
OIDCClaimDelimiter ;
OIDCScope "openid profile email eduperson_entitlement"
OIDCProviderMetadataURL https://aai-dev.egi.eu/oidc/.well-known/openid-configuration
OIDCClientID <client id>
OIDCClientSecret <client secret>
OIDCCryptoPassphrase <some crypto pass phrase>
OIDCRedirectURI https://<your keystone endpoint>/v3/auth/OS-FEDERATION/websso/openid/
↪redirect

# OAuth for CLI access
OIDCOAuthIntrospectionEndpoint https://aai-dev.egi.eu/oidc/introspect
OIDCOAuthClientID <client id>
OIDCOAuthClientSecret <client secret>

# Increase Shm cache size for supporting long entitlements
OIDCCacheShmEntrySizeMax 65536

<Location ~ "/v3/auth/OS-FEDERATION/websso/openid">
        AuthType openid-connect
        Require valid-user
</Location>

<Location ~ "/v3/OS-FEDERATION/identity_providers/egi.eu/protocols/openid/auth">
        Authtype oauth20
        Require valid-user
</Location>
```

If you have multiple keystone hosts, configure an alternative caching mechanism as per https://github.com/zmartzone/mod_auth_openidc/wiki/Caching

For example, using memcache

```
OIDCCacheType memcache
OIDCMemCacheServers "memcache1 memcache2 memcache3"
```

Be sure to enable the mod_auth_oidc module in Apache, in Ubuntu:

```
sudo a2enmod auth_openidc
```

Note: If running Keystone behind a proxy, make sure to correctly set the X-Forwarded-Proto and X-Forwarded-Port request headers, e.g. for haproxy:

```
http-request set-header X-Forwarded-Proto https if { ssl_fc }
http-request set-header X-Forwarded-Proto http if !{ ssl_fc }
http-request set-header X-Forwarded-Port %[dst_port]
```

Keystone Configuration

Configure your `keystone.conf` to include in the `[auth]` section `openid` in the list of authentication methods:

```
[auth]

# This may change in your installation, add openid to the list of the methods you␣
↪support
methods = password, token, openid
```

Add a `[openid]` section as follows:

```
[openid]
# this is the attribute in the Keystone environment that will define the identity␣
↪provider
remote_id_attribute = HTTP_OIDC_ISS
```

Add your horizon host as trusted dashboard to the `[federation]` section:

```
[federation]
trusted_dashboard = https://<your horizon>/dashboard/auth/websso/
```

Finally copy the default template for managing the tokens in horizon to `/etc/keystone/sso_callback_template.html`. This template can be found in keystone git repo at `https://github.com/openstack/keystone/blob/master/etc/sso_callback_template.html`

```
curl -L https://raw.githubusercontent.com/openstack/keystone/master/etc/sso_callback_
↪template.html \
    > /etc/keystone/sso_callback_template.html
```

Now restart your Apache (and Keystone if running in uwsgi) so you can configure the Keystone Federation support.

Keystone Federation Support

First, create a new `egi.eu` identity provider with remote id `https://aai-dev.egi.eu/oidc/`:

```
$ openstack identity provider create --remote-id https://aai-dev.egi.eu/oidc/ egi.eu
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | None                             |
| domain_id   | 1cac7817dafb4740a249cc9ca6b14ea5 |
| enabled     | True                             |
| id          | egi.eu                           |
| remote_ids  | https://aai-dev.egi.eu/oidc/     |
+-------------+----------------------------------+
```

Create a group for users coming from EGI Check-in, usual configuration is to have one group per VO you want to support.

```
$ openstack group create ops
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
```

(continues on next page)

(continued from previous page)

```
| domain_id   | default                          |
| id          | 89cf5b6708354094942d9d16f0f29f8f |
| name        | ops                              |
+-------------+----------------------------------+
```

Add that group to the desired local project:

```
$ openstack role add member --group ops --project ops
```

Define a mapping of users from EGI Check-in to the group just created and restrict with the `OIDC-eduperson_entitlement` the VOs you want to support for that group. Substitute the group id and the allowed entitlements for the adequate values for your deployment:

```
$ cat mapping.egi.json
[
    {
        "local": [
            {
                "user": {
            "name": "{0}"
        },
                "group": {
                    "id": "89cf5b6708354094942d9d16f0f29f8f"
                }
            }
        ],
        "remote": [
            {
                "type": "HTTP_OIDC_SUB"
            },
            {
                "type": "HTTP_OIDC_ISS",
                "any_one_of": [
                    "https://aai-dev.egi.eu/oidc/"
                ]
            },
            {
                "type": "OIDC-eduperson_entitlement",
                "regex": true,
                "any_one_of": [
                    "^urn:mace:egi.eu:group:ops:role=vm_operator#aai.egi.eu$"
                ]
            }
        ]
    }
]
```

More recent versions of Keystone allow for more elaborated mapping, but this configuration should work for Mitaka and onwards

Create the mapping in Keystone:

```
$ openstack mapping create --rules mapping.egi.json egi-mapping
+-------+-------------------------------------------------------------------------------
↪-------------------------------------------------+
| Field | Value
↪                                                  |
```

(continues on next page)

(continued from previous page)

```
+-------+-------------------------------------------------------------------------------
↪-------------------------------------------------------+
| id    | egi-mapping
↪                                                       |
| rules | [{u'remote': [{u'type': u'HTTP_OIDC_SUB'}, {u'type': u'HTTP_OIDC_ISS', u
↪'any_one_of': [u'https://aai-dev.egi.eu/oidc/']},        |
|       | {u'regex': True, u'type': u'OIDC-eduperson_entitlement', u'any_one_of': [u'^
↪urn:mace:egi.eu:.*:ops:vm_operator@egi.eu$']}],      |
|       | u'local': [{u'group': {u'id': u'89cf5b6708354094942d9d16f0f29f8f'}, u'user
↪': {u'name': u'{0}'}}]}]                               |
+-------+-------------------------------------------------------------------------------
↪-------------------------------------------------------+
```

Finally, create the federated protocol with the identity provider and mapping created before:

```
$ openstack federation protocol create --identity-provider egi.eu --mapping egi-
↪mapping openid
+-------------------+-------------+
| Field             | Value       |
+-------------------+-------------+
| id                | openid      |
| identity_provider | egi.eu      |
| mapping           | egi-mapping |
+-------------------+-------------+
```

Keystone is now ready to accept EGI Check-in credentials.

Horizon Configuration

Edit your local_settings.py to include the following values:

```
# Enables keystone web single-sign-on if set to True.
WEBSSO_ENABLED = True

# Allow users to choose between local Keystone credentials or login
# with EGI Check-in
WEBSSO_CHOICES = (
    ("credentials", _("Keystone Credentials")),
    ("openid", _("EGI Check-in")),
)
```

Once horizon is restarted you will be able to choose "EGI Check-in" for login.

CLI Access

The OpenStack Client has built-in support for using OpenID Connect Access Tokens to authenticate. You first need to get a valid token from EGI Check-in (e.g. from https://aai-dev.egi.eu/fedcloud/) and then use it in a command like:

```
$ openstack --os-auth-url https://<your keystone endpoint>/v3 \
            --os-auth-type v3oidcaccesstoken --os-protocol openid \
            --os-identity-provider egi.eu \
            --os-access-token <your access token> \
            token issue
```

(continues on next page)

(continued from previous page)

```
+---------+-------------------------------------------------------------------------------
↪------------+
| Field   | Value
↪            |
+---------+-------------------------------------------------------------------------------
↪------------+
| expires | 2017-05-23T11:24:31+0000
↪            |
| id      | gAAAAABZJA3fbKX....nEMAPi-
↪IsFOCkU9QWGTISYElzYJsI3z0SJGs7QsTJv4aJQq0JDJUBz6uE85SqXDj3  |
| user_id | 020864ea9415413f9d706f6b473dbeba
↪            |
+---------+-------------------------------------------------------------------------------
↪------------+
```

Additional VOs

Configuration can include as many mappings as needed in the json file. Users will be members of all the groups matching the remote part of the mapping. For example this file has 2 mappings, one for members of `ops` and another for members of `fedcloud.egi.eu`:

```
[
    {
        "local": [
            {
                "user": {
            "name": "{0}"
        },
                "group": {
                    "id": "66df3a7a0c6248cba8b729de7b042639"
                }
            }
        ],
        "remote": [
            {
                "type": "HTTP_OIDC_SUB"
            },
            {
                "type": "HTTP_OIDC_ISS",
                "any_one_of": [
                    "https://aai-dev.egi.eu/oidc/"
                ]
            },
            {
                "type": "OIDC-eduperson_entitlement",
                "regex": true,
                "any_one_of": [
                    "^urn:mace:egi.eu:group:ops:role=vm_operator#aai.egi.eu$"
                ]
            }
        ]
    },
    {
        "local": [
            {
```

(continues on next page)

(continued from previous page)

```
                "user": {
            "name": "{0}"
        },
                "group": {
                    "id": "e1c04284718f4e19bb0516e5534a24e8"
                }
            }
        ],
        "remote": [
            {
                "type": "HTTP_OIDC_SUB"
            },
            {
                "type": "HTTP_OIDC_ISS",
                "any_one_of": [
                    "https://aai-dev.egi.eu/oidc/"
                ]
            },
            {
                "type": "OIDC-eduperson_entitlement",
                "regex": true,
                "any_one_of": [
                    "^urn:mace:egi.eu:group:fedcloud.egi.eu:role=vm_operator#aai.egi.
↪eu$"
                ]
            }
        ]
    }
]
```

Moving to EGI Check-in production instance

Once tests in the development instance of Check-in are successful, you can move to the production instance. You should open a GGUS ticket for the request. Besides you will need to update your configuration as follows:

- Update the `remote-id` of the identity provider:

```
$ openstack identity provider set --remote-id https://aai.egi.eu/oidc/ egi.eu
```

- Update the `HTTP_OIDC_ISS` filter in your mappings, e.g.:

```
$ sed -i 's/aai-dev.egi.eu/aai.egi.eu/' mapping.egi.json
$ openstack mapping set --rules mapping.egi.json egi-mapping
```

- Update Apache configuration to use *aai.egi.eu* instead of *aai-dev.egi.eu*:

```
OIDCProviderMetadataURL https://aai.egi.eu/oidc/.well-known/openid-configuration
OIDCOAuthIntrospectionEndpoint https://aai.egi.eu/oidc/introspect
```

Note: Changes in the production settings

If you want to make any changes to the client configuration of the production instance, first make the changes in the Check-in development environment and then open a GGUS ticket to sync the changes to production.

VOMS Support

VOMS with FEDERATION-OS (Keystone API v3)

Note: Configure VOMS with FEDERATION-OS if your site needs to support a legacy VO relying on VOMS for authorisation, check Keystone-VOMS below for older OpenStack versions.

Warning: Work in progress.

This is currently being tested and still misses components to be released into CMD!

VOMS authentication requires Keystone to be run as a WSGI application behind an Apache server with gridsite and SSL support. GridSite is a set of extensions to the Apache 2.x webserver, which support Grid security based on X.509 certificates.

Packages for gridsite can be obtained from CMD-OS-1. Follow the CMD-OS-1 guidelines for getting the packages for your distribution.

First install the `gridsite`, `fetch-crl` and `ca-policy-egi-core` for your distribution, ensuring that `gridsite` is at least version `2.3.2`. For Ubuntu 16.04:

```
apt-get install gridsite fetch-crl ca-policy-egi-core
```

Configure Apache to use gridsite module (this may differ in your distribution):

```
a2enmod zgridsite
```

Include these lines on your Apache config for the virtual host of your Keystone service:

```
# Use the IGTF trust anchors for CAs and CRLs
SSLCACertificatePath /etc/grid-security/certificates/
SSLCARevocationPath /etc/grid-security/certificates/

# Verify clients if they send their certificate
SSLVerifyClient         optional
SSLVerifyDepth          10
SSLOptions              +StdEnvVars +ExportCertData

# Adapt this URL if needed for your deployment
<Location /v3/OS-FEDERATION/identity_providers/egi.eu/protocols/mapped/auth>
    # populate ENV variables
    GridSiteEnvs on
    # turn off directory listings
    GridSiteIndexes off
    # accept GSI proxies from clients
    GridSiteGSIProxyLimit 4
    # disable GridSite method extensions
    GridSiteMethods ""

    Require all granted
    Options -MultiViews
</Location>
```

Make sure that `mapped` authentication method exists in your `keystone.conf` in the `[auth]` section:

```
[auth]

# This may change in your installation, add mapped to the list of the methods you␣
↪support
methods = password, token, openid, mapped
```

Create an `egi.eu` identity provider and any needed groups as described in *Keystone Federation Support* (do not forget to add roles to the new group). Use those groups to create appropriate mappings to the VOs you intend to support. You can use the `GRST_VOMS_FQANS` to match to the VOMS FQAN that you want to create the mapping for. The following is an example for the `fedcloud.egi.eu` VO:

```
$ openstack group create fedcloud.egi.eu
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | default                          |
| id          | fbccb5f81f9741fd8b84736cc10c1d34 |
| name        | fedcloud.egi.eu                  |
+-------------+----------------------------------+

$ cat mapping.voms.json
[
    {
        "local": [
            {
                "user": {
                    "name": "{0}",
                    "type": "ephemeral"
                },
                "group": {
                    "id": "fbccb5f81f9741fd8b84736cc10c1d34"
                }
            }
        ],
        "remote": [
            {
                "type": "GRST_CONN_AURI_0"
            },
            {
                "type": "GRST_VOMS_FQANS",
                "any_one_of": [
                    "^/fedcloud.egi.eu/.*"
                ],
                "regex": true
            }
        ]
    }
]
$ openstack mapping create --rules mapping.voms.json voms
+-------+--------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------------
↪--------------+
| Field | Value
↪                                                                                      ␣
↪                                                                                      ␣
↪              |
```

(continues on next page)

(continued from previous page)

```
+-------+-------------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------------
↪--------------+
| id    | voms
↪                                                                                      ␣
↪                                                                                      ␣
↪              |
| rules | [{u'remote': [{u'type': u'GRST_CONN_AURI_0'}, {u'regex': True, u'type': u
↪'GRST_VOMS_FQANS', u'any_one_of': [u'^/fedcloud.egi.eu/.*']}], u'local': [{u'group
↪': {u'id': u'7d9a21050cef48889f23eb9d5f7fef51'}, u'user': {u'type': u'ephemeral', u
↪'name': u'{0}'}}]}] |
+-------+-------------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------------
↪--------------+
```

Finally add the `mapped` protocol to your `egi.eu` identity provider with the mapping you have created:

```
$ openstack federation protocol create --identity-provider egi.eu --mapping voms␣
↪mapped
+-------------------+---------+
| Field             | Value   |
+-------------------+---------+
| id                | mapped  |
| identity_provider | egi.eu  |
| mapping           | voms    |
+-------------------+---------+
```

For every VO you support you should configure the corresponding `.lsc` files at `/etc/grid-security/vomsdir/<vo name>/`. Those files depend on each VO, check the Operations Portal for details. You can find below the `fedcloud.egi.eu` configuration:

```
$ cat /etc/grid-security/vomsdir/fedcloud.egi.eu/voms1.grid.cesnet.cz.lsc
/DC=cz/DC=cesnet-ca/O=CESNET/CN=voms1.grid.cesnet.cz
/DC=cz/DC=cesnet-ca/O=CESNET CA/CN=CESNET CA 3
$ cat /etc/grid-security/vomsdir/fedcloud.egi.eu/voms2.grid.cesnet.cz.lsc
/DC=cz/DC=cesnet-ca/O=CESNET/CN=voms2.grid.cesnet.cz
/DC=cz/DC=cesnet-ca/O=CESNET CA/CN=CESNET CA 3
```

You can test easily test the authentication is working using curl with your proxy:

```
$ curl -s --cert /tmp/x509up_u1000 https://<your keystone host>/v3/OS-FEDERATION/
↪identity_providers/egi.eu/protocols/mapped/auth | python -mjson.tool
{
    "token": {
        "audit_ids": [
            "wxB8VZeHSji0D57Z86PSTA"
        ],
        "expires_at": "2018-08-24T12:40:41.000000Z",
        "issued_at": "2018-08-24T11:40:41.000000Z",
        "methods": [
            "mapped"
        ],
        "user": {
            "OS-FEDERATION": {
```

(continues on next page)

(continued from previous page)

```
                "groups": [
                    {
                        "id": "fbccb5f81f9741fd8b84736cc10c1d34"
                    }
                ],
                "identity_provider": {
                    "id": "egi.eu"
                },
                "protocol": {
                    "id": "mapped"
                }
            },
            "domain": {
                "id": "Federated",
                "name": "Federated"
            },
            "id": "ea6520b3ad34400ba07115f7a3987a6b",
            "name": "dn:/DC=org/DC=terena/DC=tcs/C=NL/O=EGI/OU=UCST/CN=Enol+Fernandez"
        }
    }
}
```

Keystone-VOMS (Keystone API v2)

Caution: VOMS Support using Keystone-VOMS is no longer supported from OpenStack Queens onwards. You should use *VOMS with FEDERATION-OS (Keystone API v3)* or *OpenID Connect Support* instead.

Support for authenticating users with X.509 certificates with VOMS extensions is achieved with Keystone-VOMS extension. Documentation is available at https://keystone-voms.readthedocs.io/

Notes:

- **You need a host certificate from a recognised CA for your keystone server**.
- Take into account that using keystone-voms plugin will **enforce the use of https for your Keystone service**, you will need to update your URLs in the configuration of your services if your current installation is not using https:
  - you will probably need to include your CA to your system's CA bundle to avoid certificate validation issues: Check the Federated Cloud OpenStack Client guide on how to do it.
  - replace http with https in `auth_[protocol|uri|url]` and `auth_[host|uri|url]` in the nova, cinder, glance and neutron config files (`/etc/nova/nova.conf`, `/etc/nova/api-paste.ini`, `/etc/neutron/neutron.conf`, `/etc/neutron/api-paste.ini`, `/etc/neutron/metadata_agent.ini`, `/etc/cinder/cinder.conf`, `/etc/cinder/api-paste.ini`, `/etc/glance/glance-api.conf`, `/etc/glance/glance-registry.conf`, `/etc/glance/glance-cache.conf`) and any other service that needs to check keystone tokens.
  - Update the URLs of the services directly in the database:

```
mysql> use keystone;
mysql> update endpoint set url="https://<keystone-host>:5000/v2.0" where url="http://
↪<keystone-host>:5000/v2.0";
mysql> update endpoint set url="https://<keystone-host>:35357/v2.0" where url="http://
↪<keystone-host>:35357/v2.0";
```

(continues on next page)

(continued from previous page)

- Most sites should enable the `autocreate_users` option in the `[voms]` section of Keystone-VOMS configuration. This will enable new users to be automatically created in your local keystone the first time they login into your site.
- if (and only if) you need to configure the Per-User Subproxy (PUSP) feature, please follow the specific guide.

2.2.5 EGI Accounting

There are two different processes handling the accounting integration:

- cASO, which connects to the OpenStack deployment to get the usage information, and,
- ssmsend, which sends that usage information to the central EGI accounting repository.

They should be run by cron periodically, settings below run cASO every hour and ssmsend every six hours.

Using the VM Appliance

cASO configuration is stored at `/etc/caso/caso.conf`. Most default values should be ok, but you must set:

- `site_name` (line 12), with the name of your site as defined in GOCDB.
- `projects` (line 20), with the list of projects you want to extract accounting from.
- credentials to access the accounting data (lines 28-47, more options also available). Check the cASO documentation for the expected permissions of the user configured here.
- The mapping from EGI VOs to your local projects `/etc/caso/voms.json`, following this format:

```
{
   "vo name": {
       "projects": ["project A that accounts for the vo", "project B that␣
↪accounts for the VO"]
   },
   "another vo": {
       "projects": ["project C that accounts for the VO"]
   }
}
```

cASO will write records to `/var/spool/apel` from where ssmsend will take them.

SSM configuration is available at `/etc/apel`. Defaults should be ok for most cases. The cron file uses `/etc/grid-security` for the CAs and the host certificate and private keys (`/etc/grid-security/hostcert.pem` and `/etc/grid-security/hostkey.pem`).

Running the services

Both caso and ssmsend are run via the root user crontab. For convenience there are two scripts `/usr/local/bin/caso-extract.sh` and `/usr/local/bin/ssm-send.sh` that run the docker container with the proper volumes.

2.2.6 EGI Information System

Information discovery provides a real-time view about the actual images and flavors available at the OpenStack for the federation users. It has two components:

- Resource-Level BDII: which queries the OpenStack deployment to get the information to publish
- Site-Level BDII: gathers information from several resource-level BDIIs and makes it publicly available for the EGI information system.

Using the VM Appliance

Resource-level BDII

This is provided by container `egifedcloud/cloudbdii`. You need to configure:

- `/etc/cloud-info-provider/openstack.rc`, with the credentials to query your OpenStack. The user configured just needs to be able to access the lists of images and flavors.
- `/etc/cloud-info-provider/openstack.yaml`, this file includes the static information of your deployment. Make sure to set the `SITE-NAME` as defined in GOCDB.

Site-level BDII

The `egifedcloud/sitebdii` container runs this process. Configuration files:

- `/etc/sitebdii/glite-info-site-defaults.conf`. Set here the name of your site (as defined in GOCDB) and the public hostname where the appliance will be available.
- `/etc/sitebdii/site.cfg`. Include here basic information on your site.

Running the services

There is a `bdii.service` unit for systemd available in the appliance. This leverages docker-compose for running the containers. You can start the service with:

```
systemctl start bdii
```

Check the status with:

```
systemctl status bdii
```

And stop with:

```
systemctl stop bdii
```

You should be able to get the BDII information with an LDAP client, e.g.:

```
ldapsearch -x -p 2170 -h <yourVM.hostname.domain.com> -b o=glue
```

2.2.7 EGI VM Image Management

VM Images are replicated using *cloudkeeper*, which has two components:

- fronted (cloudkeeper-core) dealing the with image lists and downloading the needed images, run periodically with cron
- backend (cloudkeeper-os) dealing with your glance catalogue, running permanently.

Using the VM Appliance

Every 4 hours, the appliance will perform the following actions:

- download the configured lists in `/etc/cloudkeeper/image-lists.conf` and verify its signature
- check any changes in the lists and download new images
- synchronise this information to the configured glance endpoint

First you need to configure and start the backend. Edit `/etc/cloudkeeper-os/cloudkeeper-os.conf` and add the authentication parameters from line 117 to 136.

Then add as many image lists (one per line) as you would like to subscribe to `/etc/cloudkeeper/image-lists.conf`. Use URLs with your AppDB token for authentication, check the following guides for getting such token and URLs:

- how to access to VO-wide image lists, and
- how to subscribe to a private image list.

Running the services

cloudkeeper-os should run permanently, there is a `cloudkeeper-os.service` for systemd in the appliance. Manage as usual:

```
systemctl <start|stop|status> cloudkeeper-os
```

cloudkeeper core is run every 4 hours with a cron script.

2.2.8 EGI VM Management (optional)

Follow the installation and configuration manual of ooi.

Once the OCCI interface is installed, you should register it on your installation (adapt the region and URL to your deployment), e.g.:

```
$ openstack service create --name occi --description "OCCI Interface" occi
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description | OCCI Interface                   |
| enabled     | True                             |
| id          | 6dfd6a56c9a6456b84e8c86038e58f56 |
| name        | occi                             |
| type        | occi                             |
+-------------+----------------------------------+
```

(continues on next page)

(continued from previous page)

```
$ openstack endpoint create --region RegionOne occi --publicurl http://172.16.4.
↪70:8787/occi1.1

+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |           OCCI service           |
|      id     | 8e6de5d0d7624584bed6bec9bef7c9e0 |
|     name    |             occi_api             |
|     type    |               occi               |
+-------------+----------------------------------+
```

2.2.9 Post-installation

After the installation of all the needed components, it is recommended to set the following policies on Nova to avoid users accessing other users resources:

```
sed -i 's|"admin_or_owner":  "is_admin:True or project_id:%(project_id)s",|"admin_or_
↪owner":  "is_admin:True or project_id:%(project_id)s",\n    "admin_or_user":  "is_
↪admin:True or user_id:%(user_id)s",|g' /etc/nova/policy.json
sed -i 's|"default": "rule:admin_or_owner",|"default": "rule:admin_or_user",|g' /etc/
↪nova/policy.json
sed -i 's|"compute:get_all": "",|"compute:get": "rule:admin_or_owner",\n
↪"compute:get_all": "",|g' /etc/nova/policy.json
```

2.2.10 Upgrading the OpenStack Appliance

From 2017.08.09 to 2018.05.07

Configuration changes:

- This upgrade moves the `voms.json` file to the respective `caso` and `cloudkeeper-os` directories under `/etc/`
- No other changes in configuration are needed

From 20160403 to 2017.08.09

There are several major changes between these versions, namely:

- atrope has been deprecated and cloudkeeper is used instead. The configuration cannot be reused directly and the new services need to be configured as described above
- caso is upgraded to version 1.1.1, the configuration file has some incompatible changes.
- A new bdii.service is available for managing the process is available.

CHAPTER 3

Registration of services in GOCDB

Site endpoints must be registered in EGI Configuration Management Database (GOCDB). If you are creating a new site for your cloud services, check the PROC09 Resource Centre Registration and Certification procedure. Services can also coexist within an existing (grid) site.

These are the expected services for a working site:

- **Site-BDII**. This service collects and publishes site's data for the Information System. Existing sites should already have this registered.
- **eu.egi.cloud.accounting**. Register here the host sending the records to the accounting repository (executing SSM send).
- **eu.egi.cloud.vm-metadata.vmcatcher** for the VMI replication mechanism. Register here the host providing the replication (i.e. the host with cloudkeeper installation)

If offering OCCI interface, sites should register:

- **eu.egi.cloud.vm-management.occi** for the OCCI endpoint offered by the site. The endpoint URL must follow this syntax:

  ```
  https://hostname:port/?image=<image_name>&resource=<resource_name>
  ```

  where `<image_name>` and `<resource_name>` cannot contain spaces. These attributes map to os_tpl and resource_tpl respectively and will be the ones used for monitoring purposes.

If offering native OpenStack access (nova), register:

- **org.openstack.nova** for the Nova endpoint of the site. The endpoint URL must contain the Keystone v3 URL:

  ```
  https://hostname:port/url/v3
  ```

If offering native OpenStack access (swift), register:

- **org.openstack.swift** for the swift endpoint of the site. The endpoint URL field must contain Keystone v3 URL:

  ```
  https://hostname:port/url/v3
  ```

CHAPTER 4

VO Configuration

This section provides the needed steps for supporting a new VO in your infrastructure

4.1 EGI AAI

4.1.1 OpenStack

The usual method of supporting a VO is by creating a local project for it. You should assign quotas to this project as agreed in the OLA defining the support for the given VO.

Check-in VOs (OpenID Connect)

Follow these steps if you are using OpenID Connect to integrate with EGI:

1. Create a group where users belongig to the VO will be mapped to:

```
group_id=$(openstack group create -f value -c id <new_group>)
```

2. Add that group to the desired local project:

```
openstack role add member --group $group_id --project <your project>
```

3. Expand your mapping.json with the VO membership to the created group (substitute `group_id` and `vo_name` as appropriate):

```
[
    <existing mappings>,
    {
        "local": [
            {
                "user": {
                    "name": "{0}"
```

(continues on next page)

(continued from previous page)

```
                },
                "group": {
                    "id": "<group_id>"
                }
            }
        ],
        "remote": [
            {
                "type": "HTTP_OIDC_SUB"
            },
            {
                "type": "HTTP_OIDC_ISS",
                "any_one_of": [
                    "https://aai-dev.egi.eu/oidc/"
                ]
            },
            {
                "type": "OIDC-eduperson_entitlement",
                "regex": true,
                "any_one_of": [
                    "^urn:mace:egi.eu:group:<vo_name>:role=vm_operator#aai.egi.eu$
↪"
                ]
            }
        ]
    }
]
```

4. Update the mapping in your Keystone IdP:

```
openstack mapping set --rules mapping.json egi-mapping
```

Legacy VOs (VOMS)

When using the Keystone-VOMS module, you should follow these steps:

1. Configure your LSC files according to the VOMS documentation, e.g.:

```
mkdir -p /etc/grid-security/vomsdir/ops

cat > /etc/grid-security/vomsdir/ops/lcg-voms2.cern.ch.lsc << EOF
/DC=ch/DC=cern/OU=computers/CN=lcg-voms2.cern.ch
/DC=ch/DC=cern/CN=CERN Grid Certification Authority
EOF

cat > /etc/grid-security/vomsdir/ops/voms2.cern.ch.lsc << EOF
/DC=ch/DC=cern/OU=computers/CN=voms2.cern.ch
/DC=ch/DC=cern/CN=CERN Grid Certification Authority
EOF
```

2. Add the mapping to your `voms.json` mapping. It must be proper JSON (you can check its correctness online or with `python -mjson.tool /etc/keystone/voms.json`). Edit the file, and add an entry like this:

```
{
    "<voname|FQAN>": {
```

(continues on next page)

(continued from previous page)

```
        "tenant": "<project_name>"
    }
}
```

Note that you can use the FQAN from the incoming proxy, so you can map a group within a VO into a tenant, like this:

```
{
    "dteam": {
        "tenant": "dteam"
    },
    "/dteam/NGI_IBERGRID": {
        "tenant": "dteam_ibergrid"
    }
}
```

3. Restart Apache server, and it's done.

4.1.2 OpenNebula

TBC

4.2 EGI Accounting

4.2.1 OpenStack

Add the project supporting the VO to cASO:

1. `projects` in `/etc/caso/caso.conf`

```
projects = vo_project1, vo_project2, <your_new_vo_project>
```

2. as a new mapping in `/etc/caso/voms.json`

```
{
    "<your new vo>": {
        "projects": ["<your new vo project>"]
     }
}
```

Be sure to include the user running cASO as member of the project if it does not have admin privileges:

```
openstack role add member --user <your caso user> --project <your new vo project>
```

4.2.2 OpenNebula

Update `/etc/oneacct-export/groups.include` or `/etc/oneacct-export/groups.exclude` to allow extracting information from the new group. Specify one group name per line.

4.3 EGI Information System

4.3.1 OpenStack

Add the user configured in your cloud-info-provider as member of the new project:

```
openstack role add member --user <your cloud-info-provider user> --project <your new␣
↪vo project>
```

4.4 EGI VM Image Management

4.4.1 cloudkeeper-core

Add the new image list to the `cloudkeeper` configuration in `/etc/cloudkeeper/cloudkeeper.yml` (or `/etc/cloudkeeper/image-lists.conf` if using the appliance), new entry should look similar to:

```
https://<APPDB_TOKEN>:x-oauth-basic@vmcaster.appdb.egi.eu/store/vo/<your new vo>/
↪image.list:
```

4.4.2 OpenStack

Add the user configured in cloudkeeper-os as member of the new project:

```
openstack role add member --user <your cloudkeeper-os user> --project <your new vo␣
↪project>
```

Add the mapping of the project to the VO in `/etc/cloudkeeper-os/voms.json`:

```
{
    "<your new vo>": {
         "tenant": "<your new vo project>"
    }
}
```

CHAPTER 5

Installation Validation

Once the site services are registered in GOCDB (and flagged as "monitored") they will appear in the EGI service monitoring tools. EGI will check the status of the services (see Infrastructure Status for details). Check if your services are present in the EGI service monitoring tools and passing the tests; if you experience any issues (services not shown, services are not OK...) please contact back EGI Operations or your reference Resource Infrastructure.

Extra checks for your installation:

- Check in ARGO-Mon2 that your services are listed and are passing the tests. If all the tests are OK, your installation is already in good shape.
- Check that you are publishing cloud information in your site BDII:

```
ldapsearch -x -h <site bdii host> -p 2170 -b Glue2GroupID=cloud,Glue2DomainID=
↪<your site name>,o=glue
```

- Check that all the images listed in the AppDB for the VOs you support (e.g. AppDB page for fedlcoud.egi.eu VO) are listed in your BDII. This sample query will return all the template IDs registered in your BDII:

```
ldapsearch -x -h <site bdii host> -p 2170 -b Glue2GroupID=cloud,Glue2DomainID=
↪<your site name>,o=glue objectClass=GLUE2ApplicationEnvironment␣
↪GLUE2ApplicationEnvironmentRepository
```

- Try to start one of those images in your cloud. You can do it with *onetemplate instantiate* or OCCI commands, the result should be the same.
- Execute the site certification manual tests against your endpoints.
- Check in the accounting portal that your site is listed and the values reported look consistent with the usage of your site.

CHAPTER 6

FAQ

6.1 Why joining the EGI Cloud?

- To support international communities supported by EGI (e.g. these research communities and applications or these research infrastructures in EOSC-hub or these business pilots in the EOSC Digital Innovation Hub.
- To participate in e-Infrastructure projects (H2020, EOSC) as an EGI compliant IaaS cloud provider.
- To participate in resource allocation and in pay-for-use campaigns run by EGI.
- To align access policies and operational model of your cloud with international good practices.
- To adopt best practices of multi-cloud federation for the benefit of your local users.

6.2 Do I lose control on who can access my resources if I join federated cloud?

No

EGI uses the concept of Virtual Organisation (VO) to group users. The resource provider has complete control on which VOs he wants to allow on its resources and which quotas or restrictions to assign to each VO. In the case of OpenStack, each VO is mapped to a regular OpenStack project that can be managed as any other and are isolated to other projects you may have configured in your deployment. Although not recommended, you can even restrict the automatic access of users within a VO and manually enable individual members.

6.3 How many components do I have to install?

Depending on your cloud management framework and the kind of integration this will vary.

In general, the federation requires your cloud management framework to be configured to support Federated AAI with EGI Check-in. This may require changes in your current setup.

Other components are designed to access your cloud management framework public APIs and do not require modification of your deployment. For OpenStack, these components can be run on a single VM that encapsulates them for convenience.

6.4 Which components of my cloud will interact with the federated cloud components?

For OpenStack they are:

- Keystone
- Nova
- Glance
- Swift (optional)

Users will also interact with:

- Neutron
- Cinder

to perform their regular activities.

6.5 How will my daily operational activities change?

For the most part daily operations will not change.

A resource centre part of the EGI Federation, and supporting international communities, needs to provide support through the EGI channels. This means following up GGUS tickets. This includes requests from user communities and tickets triggered by failures detected by the monitoring infrastructure.

A resource centre needs to maintain the services federated in EGI properly configured with the EGI AAI.

The resource centre will have to comply with the operational and security requirements. All the EGI policies aim at implementing service provisioning best practices and common requirements. EGI operations may conduct campaigns targeted to mitigate security vulnerabilities and to update unsupported operating system and software. These activities are part of the regular activities of a resource centre anyways (also for the non-federated ones). EGI and the Operations Centres coordinate these actions in order to have them implemented in a timely manner.

In summary, most of the site activities that are coordinated by EGI and the NGIs are already part of the work plan of a well-maintained resource centre, the additional task for a site manager is to acknowledge to EGI that the task has been performed.