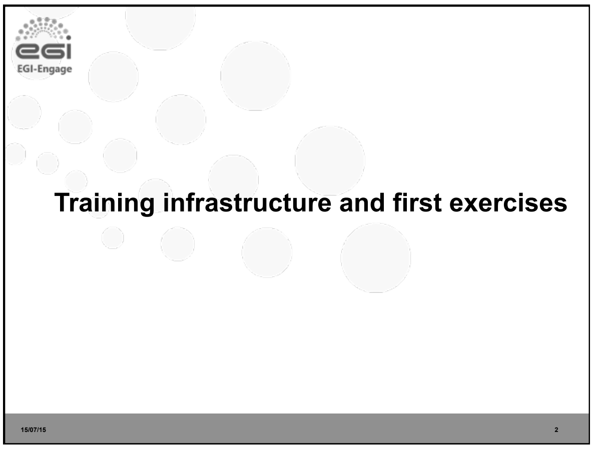
EGI Federated Cloud Tutorial

Enol Fernandez, Yin Chen
EGI User Community Support Team

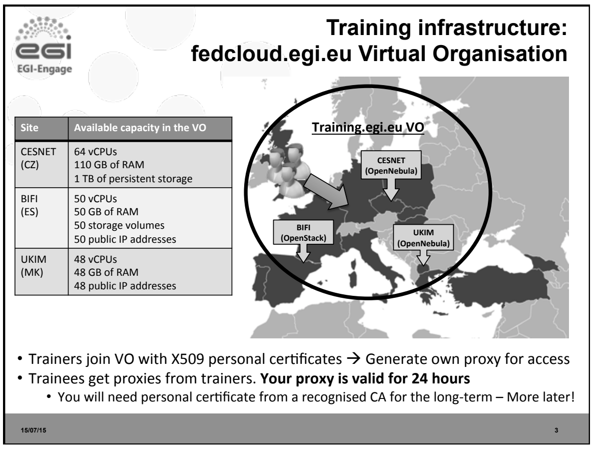
http://go.egi.eu/cloud

www.egi.eu

EGI-Engage is co-funded by the Horizon 2020 Framework Programme of the European Union under grant number 654142



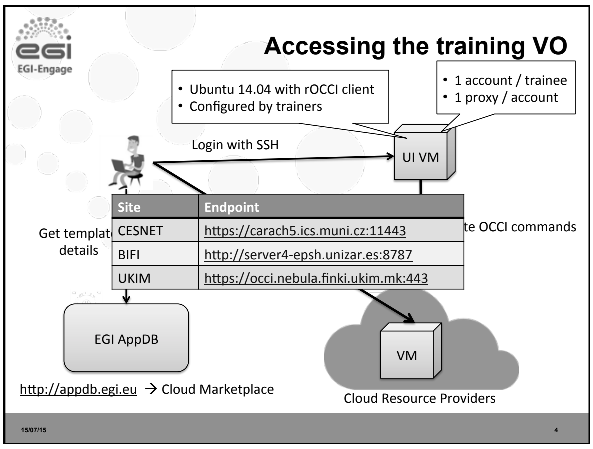
Training infrastructure and first exercises



Training infrastructure: fedcloud.egi.eu Virtual Organisation

Site	Available capacity in the VO
CESNET (CZ)	64 vCPUs 110 GB of RAM 1 TB of persistent storage
BIFI (ES)	50 vCPUs 50 GB of RAM 50 storage volumes 50 public IP addresses
UKIM (MK)	48 vCPUs 48 GB of RAM 48 public IP addresses

- Trainers join VO with X509 personal certificates → Generate own proxy for access
- Trainees get proxies from trainers. **Your proxy is valid for 24 hours**
  - You will need personal certificate from a recognised CA for the long-term – More later!



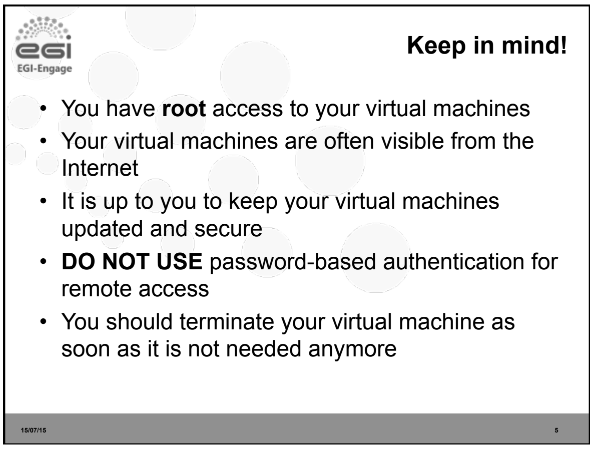
Accessing the training VO

- Ubuntu 14.04 with rOCCI client
- Configured by trainers

- 1 account / trainee
- 1 proxy / account

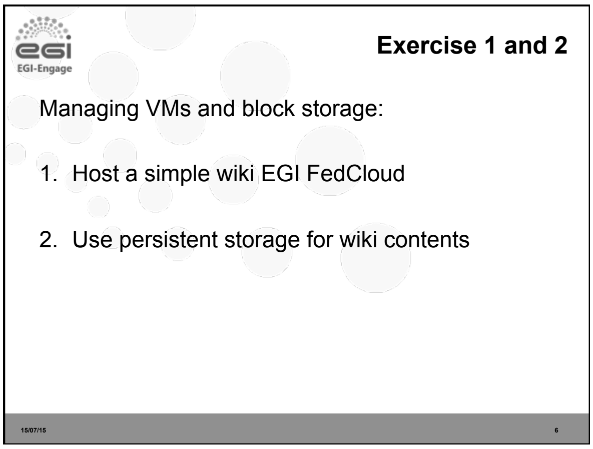
Site	Endpoint
CESNET	https://carach5.ics.muni.cz:11443
BIFI	http://server4-epsh.unizar.es:8787
UKIM	https://occi.nebula.finki.ukim.mk:443

http://appdb.egi.eu → Cloud Marketplace



Keep in mind!

- You have **root** access to your virtual machines
- Your virtual machines are often visible from the Internet
- It is up to you to keep your virtual machines updated and secure
- **DO NOT USE** password-based authentication for remote access
- You should terminate your virtual machine as soon as it is not needed anymore



Exercise 1 and 2

Managing VMs and block storage:

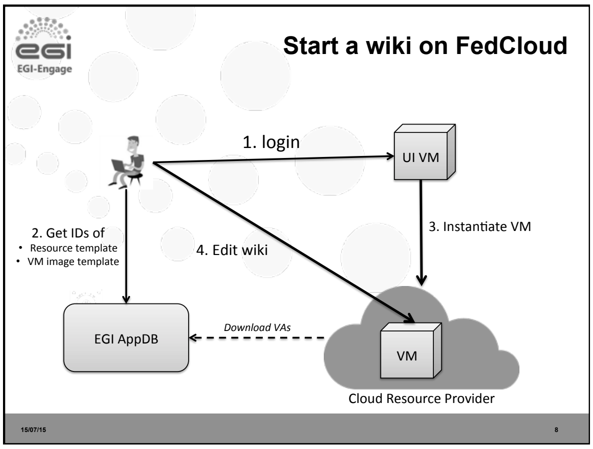
1. Host a simple wiki EGI FedCloud

2. Use persistent storage for wiki contents

Exercise sheet: Wiki setup

www.egi.eu

EGI-Engage is co-funded by the Horizon 2020 Framework Programme of the European Union under grant number 654142



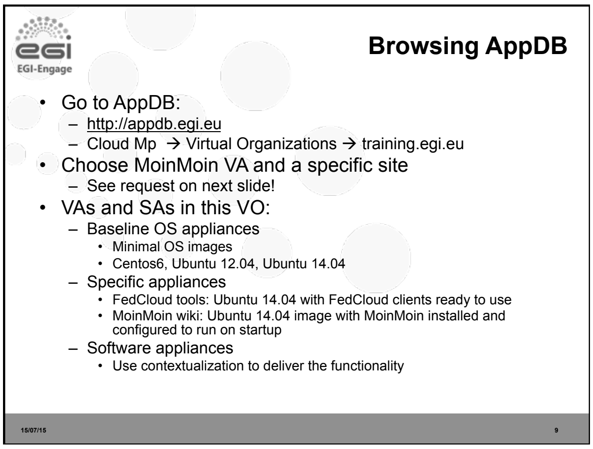
Start a wiki on FedCloud

1. login

2. Get IDs of
- Resource template
- VM image template

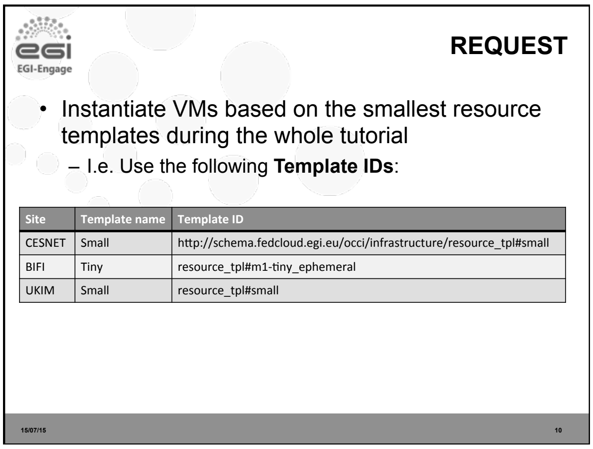
3. Instantiate VM

4. Edit wiki



Browsing AppDB

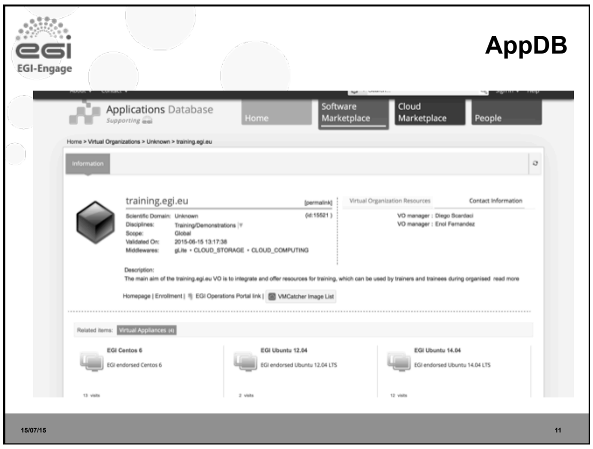
- Go to AppDB:
  - http://appdb.egi.eu
  - Cloud Mp → Virtual Organizations → training.egi.eu
- Choose MoinMoin VA and a specific site
  - See request on next slide!
- VAs and SAs in this VO:
  - Baseline OS appliances
    - Minimal OS images
    - Centos6, Ubuntu 12.04, Ubuntu 14.04
  - Specific appliances
    - FedCloud tools: Ubuntu 14.04 with FedCloud clients ready to use
    - MoinMoin wiki: Ubuntu 14.04 image with MoinMoin installed and configured to run on startup
  - Software appliances
    - Use contextualization to deliver the functionality



REQUEST

- Instantiate VMs based on the smallest resource templates during the whole tutorial
  - I.e. Use the following **Template IDs**:

Site	Template name	Template ID
CESNET	Small	http://schema.fedcloud.egi.eu/occi/infrastructure/resource_tpl#small
BIFI	Tiny	resource_tpl#m1-tiny_ephemeral
UKIM	Small	resource_tpl#small



AppDB

Home > Virtual Organizations > Unknown > training.egi.eu

training.egi.eu [permalink]

(id:15521)

Scientific Domain: Unknown
Disciplines: Training/Demonstrations
Scope: Global
Validated On: 2015-06-15 13:17:38
Middlewares: gLite • CLOUD_STORAGE • CLOUD_COMPUTING

Virtual Organization Resources | Contact Information

VO manager : Diego Scardaci
VO manager : Enol Fernandez

Description:
The main aim of the training.egi.eu VO is to integrate and offer resources for training, which can be used by trainers and trainees during organised read more

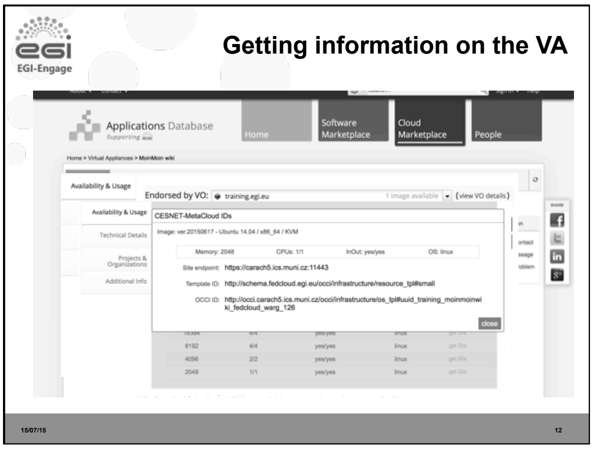
Homepage | Enrollment | EGI Operations Portal link | VMCatcher Image List

Related items: Virtual Appliances (4)

EGI Centos 6
EGI endorsed Centos 6
13 visits

EGI Ubuntu 12.04
EGI endorsed Ubuntu 12.04 LTS
2 visits

EGI Ubuntu 14.04
EGI endorsed Ubuntu 14.04 LTS
12 visits



Getting information on the VA

Home > Virtual Appliances > MoinMoin wiki

Endorsed by VO: training.egi.eu 1 image available (view VO details)

CESNET-MetaCloud IDs

Image: ver.20150617 - Ubuntu 14.04 / x86_64 / KVM

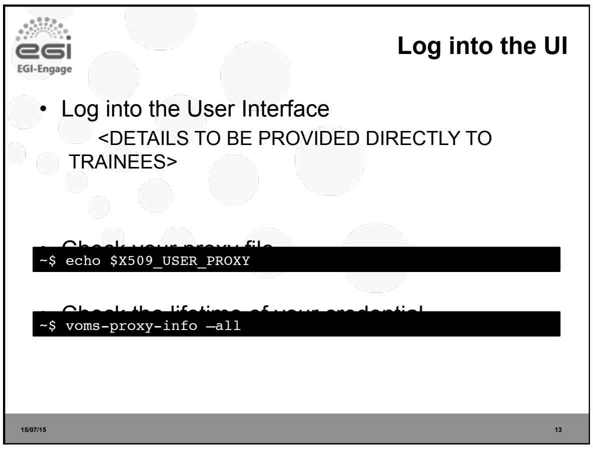
Memory: 2048 CPUs: 1/1 InOut: yes/yes OS: linux

Site endpoint: https://carach5.ics.muni.cz:11443

Template ID: http://schema.fedcloud.egi.eu/occi/infrastructure/resource_tpl#small

OCCI ID: http://occi.carach5.ics.muni.cz/occi/infrastructure/os_tpl#uuid_training_moinmoinwiki_fedcloud_warg_126

8192	4/4	yes/yes	linux	get IDs
4096	2/2	yes/yes	linux	get IDs
2048	1/1	yes/yes	linux	get IDs



Log into the UI

- Log into the User Interface

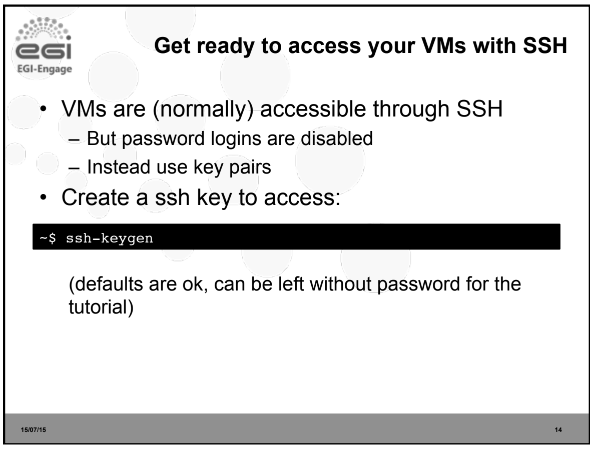
  <DETAILS TO BE PROVIDED DIRECTLY TO TRAINEES>

- Check your proxy file

```
~$ echo $X509_USER_PROXY
```

- Check the lifetime of your credential

```
~$ voms-proxy-info –all
```

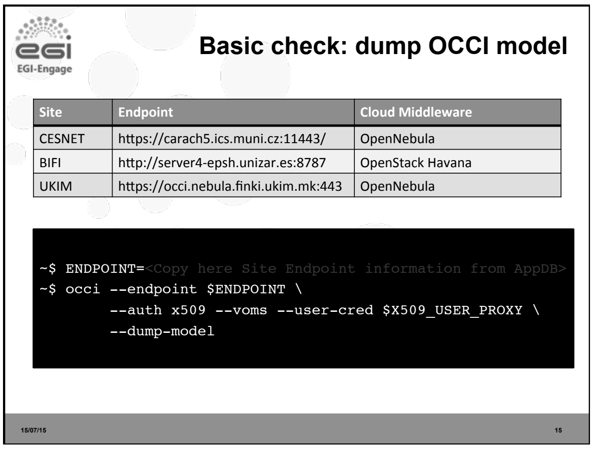


Get ready to access your VMs with SSH

- VMs are (normally) accessible through SSH
  - But password logins are disabled
  - Instead use key pairs
- Create a ssh key to access:

```
~$ ssh-keygen
```

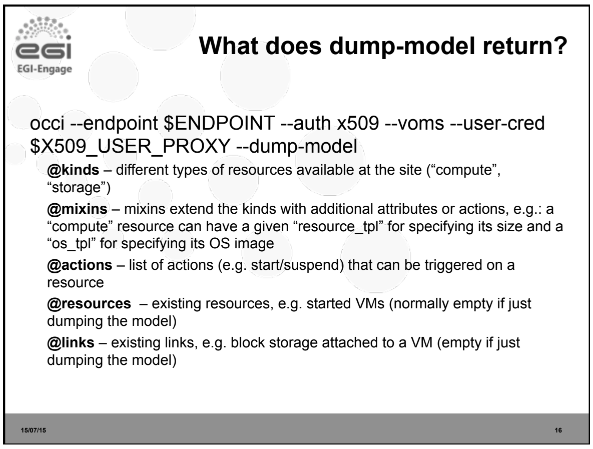
(defaults are ok, can be left without password for the tutorial)



Basic check: dump OCCI model

Site	Endpoint	Cloud Middleware
CESNET	https://carach5.ics.muni.cz:11443/	OpenNebula
BIFI	http://server4-epsh.unizar.es:8787	OpenStack Havana
UKIM	https://occi.nebula.finki.ukim.mk:443	OpenNebula

```
~$ ENDPOINT=<Copy here Site Endpoint information from AppDB>
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --dump-model
```



What does dump-model return?

occi --endpoint $ENDPOINT --auth x509 --voms --user-cred $X509_USER_PROXY --dump-model

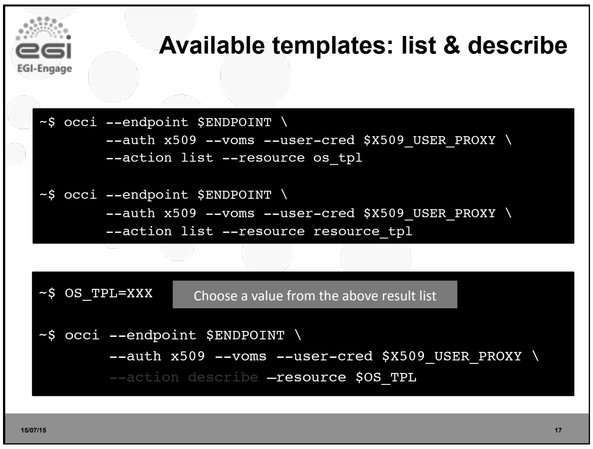
@kinds – different types of resources available at the site ("compute", "storage")

@mixins – mixins extend the kinds with additional attributes or actions, e.g.: a "compute" resource can have a given "resource_tpl" for specifying its size and a "os_tpl" for specifying its OS image

@actions – list of actions (e.g. start/suspend) that can be triggered on a resource

@resources – existing resources, e.g. started VMs (normally empty if just dumping the model)

@links – existing links, e.g. block storage attached to a VM (empty if just dumping the model)



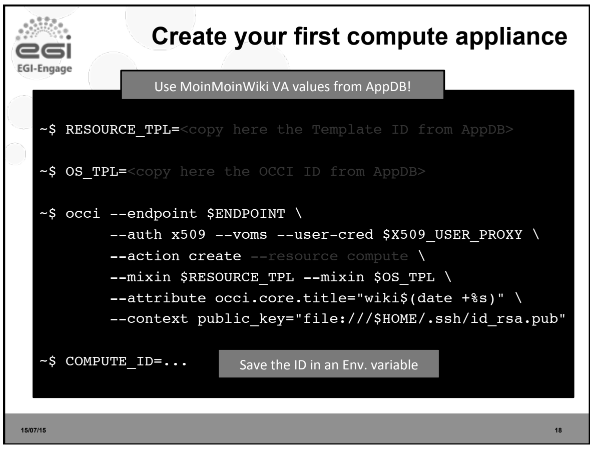
Available templates: list & describe

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action list --resource os_tpl

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action list --resource resource_tpl
```

```
~$ OS_TPL=XXX
```

Choose a value from the above result list

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe –resource $OS_TPL
```

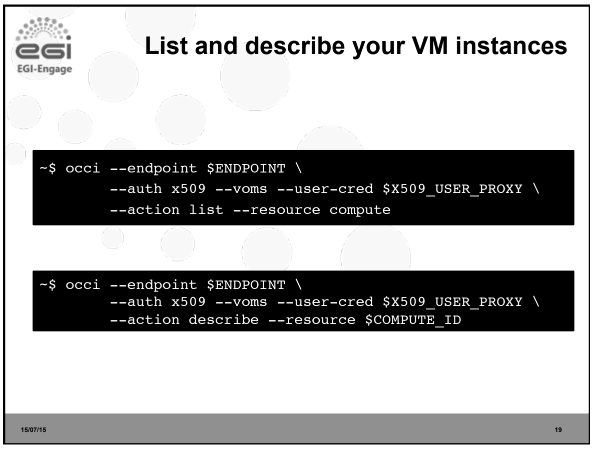


Create your first compute appliance

Use MoinMoinWiki VA values from AppDB!

```
~$ RESOURCE_TPL=<copy here the Template ID from AppDB>

~$ OS_TPL=<copy here the OCCI ID from AppDB>

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource compute \
        --mixin $RESOURCE_TPL --mixin $OS_TPL \
        --attribute occi.core.title="wiki$(date +%s)" \
        --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ COMPUTE_ID=...
```

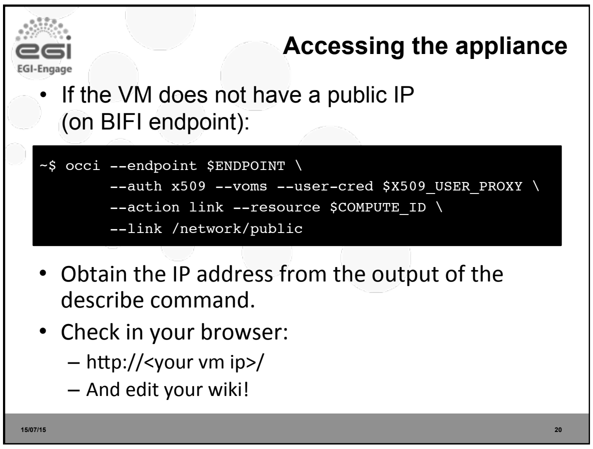
Save the ID in an Env. variable



List and describe your VM instances

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action list --resource compute
```

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe --resource $COMPUTE_ID
```

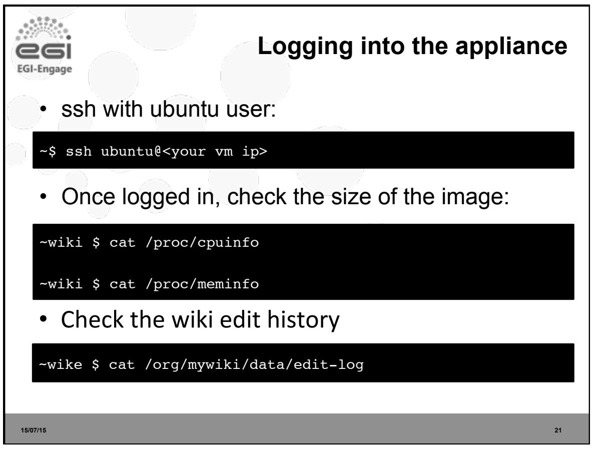


Accessing the appliance

- If the VM does not have a public IP (on BIFI endpoint):

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action link --resource $COMPUTE_ID \
        --link /network/public
```

- Obtain the IP address from the output of the describe command.
- Check in your browser:
  - http://<your vm ip>/
  - And edit your wiki!



Logging into the appliance

- ssh with ubuntu user:

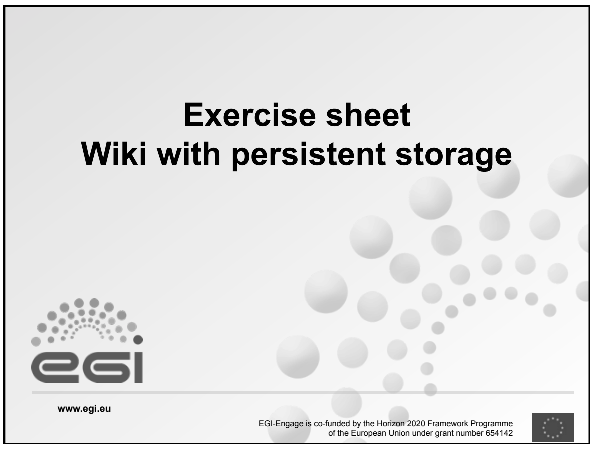
```
~$ ssh ubuntu@<your vm ip>
```

- Once logged in, check the size of the image:

```
~wiki $ cat /proc/cpuinfo

~wiki $ cat /proc/meminfo
```

- Check the wiki edit history

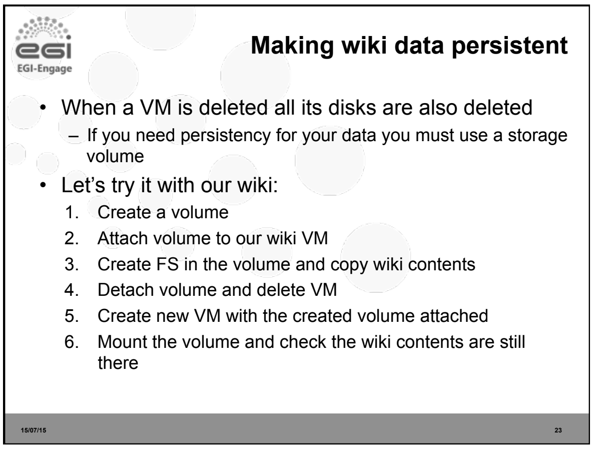
```
~wike $ cat /org/mywiki/data/edit-log
```



Exercise sheet
Wiki with persistent storage

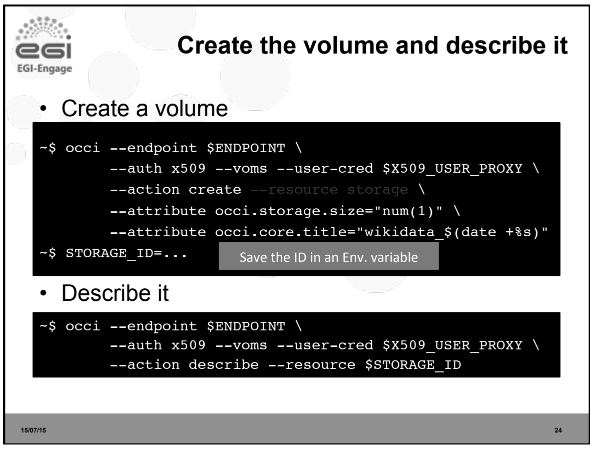
www.egi.eu

EGI-Engage is co-funded by the Horizon 2020 Framework Programme of the European Union under grant number 654142



Making wiki data persistent

- When a VM is deleted all its disks are also deleted
  - If you need persistency for your data you must use a storage volume
- Let's try it with our wiki:
  1. Create a volume
  2. Attach volume to our wiki VM
  3. Create FS in the volume and copy wiki contents
  4. Detach volume and delete VM
  5. Create new VM with the created volume attached
  6. Mount the volume and check the wiki contents are still there



Create the volume and describe it

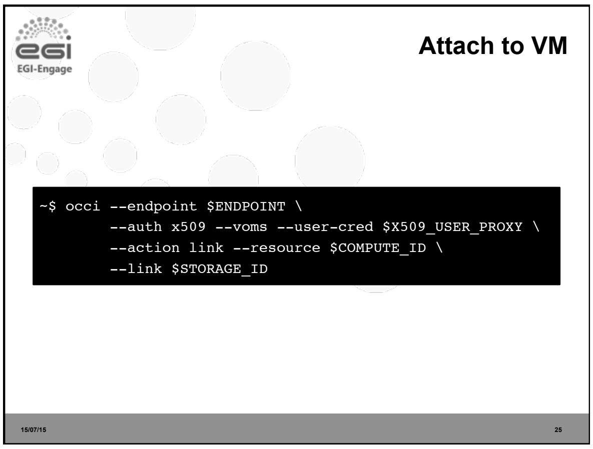
- Create a volume

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource storage \
        --attribute occi.storage.size="num(1)" \
        --attribute occi.core.title="wikidata_$(date +%s)"
~$ STORAGE_ID=...
```

Save the ID in an Env. variable

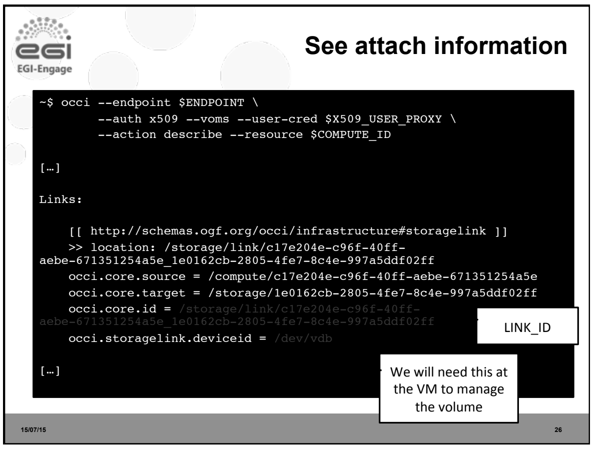
- Describe it

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe --resource $STORAGE_ID
```



Attach to VM

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action link --resource $COMPUTE_ID \
        --link $STORAGE_ID
```

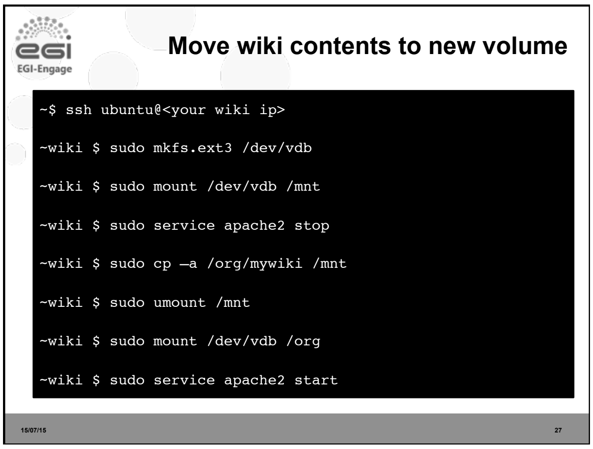


See attach information

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe --resource $COMPUTE_ID

[…]

Links:

    [[ http://schemas.ogf.org/occi/infrastructure#storagelink ]]
    >> location: /storage/link/c17e204e-c96f-40ff-
aebe-671351254a5e_1e0162cb-2805-4fe7-8c4e-997a5ddf02ff
    occi.core.source = /compute/c17e204e-c96f-40ff-aebe-671351254a5e
    occi.core.target = /storage/1e0162cb-2805-4fe7-8c4e-997a5ddf02ff
    occi.core.id = /storage/link/c17e204e-c96f-40ff-
aebe-671351254a5e_1e0162cb-2805-4fe7-8c4e-997a5ddf02ff
    occi.storagelink.deviceid = /dev/vdb

[…]
```

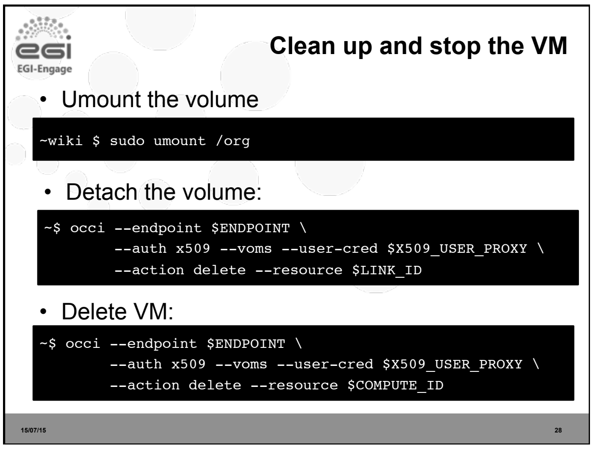
LINK_ID

We will need this at the VM to manage the volume



Move wiki contents to new volume

```
~$ ssh ubuntu@<your wiki ip>

~wiki $ sudo mkfs.ext3 /dev/vdb

~wiki $ sudo mount /dev/vdb /mnt

~wiki $ sudo service apache2 stop

~wiki $ sudo cp –a /org/mywiki /mnt

~wiki $ sudo umount /mnt

~wiki $ sudo mount /dev/vdb /org

~wiki $ sudo service apache2 start
```



Clean up and stop the VM

- Umount the volume

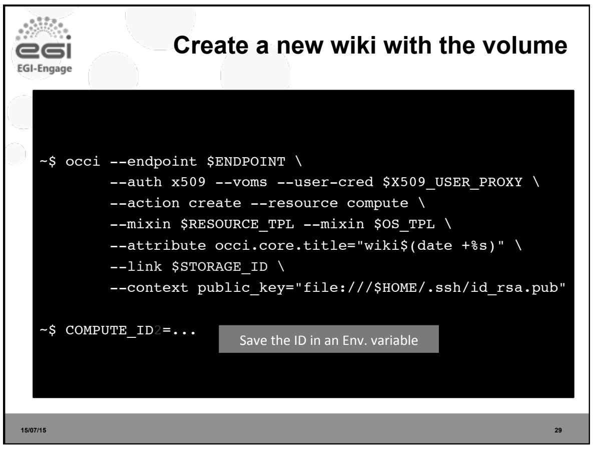
```
~wiki $ sudo umount /org
```

- Detach the volume:

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $LINK_ID
```

- Delete VM:

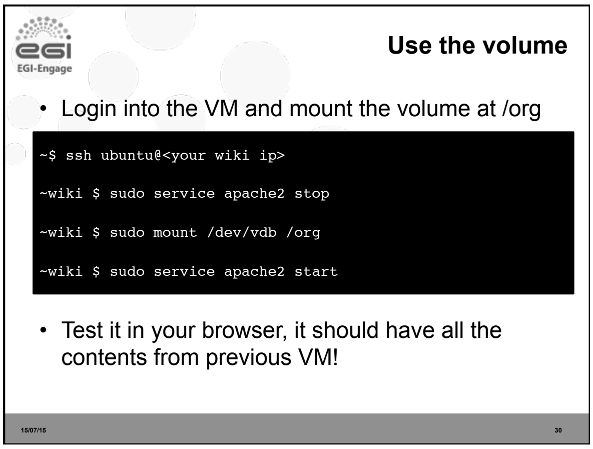
```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $COMPUTE_ID
```



Create a new wiki with the volume

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource compute \
        --mixin $RESOURCE_TPL --mixin $OS_TPL \
        --attribute occi.core.title="wiki$(date +%s)" \
        --link $STORAGE_ID \
        --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ COMPUTE_ID2=...
```

Save the ID in an Env. variable



Use the volume

- Login into the VM and mount the volume at /org

```
~$ ssh ubuntu@<your wiki ip>

~wiki $ sudo service apache2 stop

~wiki $ sudo mount /dev/vdb /org

~wiki $ sudo service apache2 start
```

- Test it in your browser, it should have all the contents from previous VM!

Once done, delete your instances

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $COMPUTE_ID2
```

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $STORAGE_ID
```

Exercise sheet

Fractal application with contextualisation

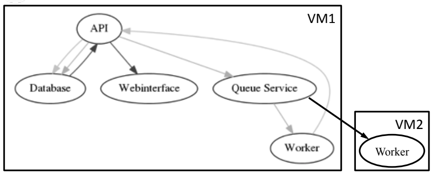
www.egi.eu

EGI-Engage is co-funded by the Horizon 2020 Framework Programme
of the European Union under grant number 654142

Fractal application on FedCloud

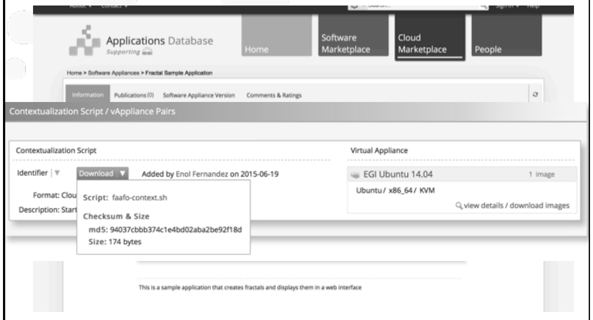
- Sample application borrowed from OpenStack “First Application For OpenStack” tutorial*
- Compute & display fractals using a set of worker nodes that get the tasks using a queue service

- A single binary includes all services.
- Configuration is controllable with command-line parameters



*Licensed under Creative Commons Attribution 3.0 License http://creativecommons.org/licenses/by/3.0/legalcode.

Where to find contextualised VAs?
Software Appliances in AppDB



Home > Software Appliances > Fractal Sample Application

Contextualization Script / vAppliance Pairs

Contextualization Script

Identifier

Added by Enol Fernandez on 2015-06-19

Format: Clou

Description: Start

Script: faafo-context.sh

Checksum & Size

md5: 94037cbbb374c1e4bd02aba2be92f18d

Size: 174 bytes

Virtual Appliance

EGI Ubuntu 14.04 1 image

Ubuntu / x86_64 / KVM

view details / download images

This is a sample application that creates fractals and displays them in a web interface

Workflow

1. Get the VA details from AppDB
2. Get the contextualization script from AppDB
3. Start a VM that holds the complete application and test it
4. Modify the contextualization script for a worker
5. Start a worker VM that listens on the queue service
6. Create some fractals
7. Check that the worker generates some fractals

Start Application

```
~$ OS_TPL=<OCCI ID from AppDB>

~$ RESOURCE_TPL=<Template ID from AppDB>

~$ curl <context script URL from AppDB> > context.sh

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource compute \
        --mixin $RESOURCE_TPL --mixin $OS_TPL \
        --attribute occi.core.title="fractal$(date +%s)" \
        --context user_data="file:///$PWD/context.sh" \
        --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ MASTER_ID=...
```

From the Fractal VA

Save the ID in an Env. variable

List and describe your VM instance

```
~$ occi --endpoint $ENDPOINT \
       --auth x509 --voms --user-cred $X509_USER_PROXY \
       --action list --resource compute
```

```
~$ occi --endpoint $ENDPOINT \
       --auth x509 --voms --user-cred $X509_USER_PROXY \
       --action describe --resource $MASTER_ID
```

Accessing the appliance

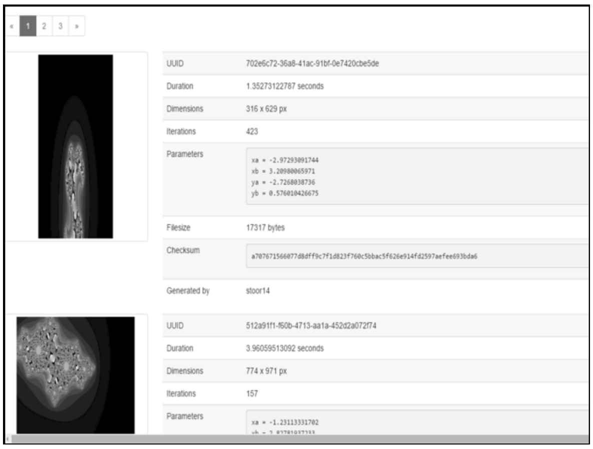
- If the VM does not have a public IP (BIFI endpoint):

```
~$ occi --endpoint $ENDPOINT \
       --auth x509 --voms --user-cred $X509_USER_PROXY \
       --action links --resource $MASTER_ID \
       --link /network/public
```

- Login with your ssh key and create some fractals:

```
~$ faafo --endpoint-url http://localhost --verbose create
```

- Check in your browser:
  - http://<your vm ip>/



UUID	702e6c72-36a8-41ac-91bf-0e7420cbe5de
Duration	1.35273122787 seconds
Dimensions	316 x 629 px
Iterations	423

Parameters	xa = -2.97293091744
	xb = 3.20980065971
	ya = -2.7268038736
	yb = 0.576010426675

Filesize	17317 bytes
Checksum	a707671566077d8dff9c7f1d823f760c5bbac5f626e914fd2597aefee693bda6
Generated by	stoor14
UUID	512a91f1-f60b-4713-aa1a-452d2a072f74
Duration	3.96059513092 seconds
Dimensions	774 x 971 px
Iterations	157

Parameters	xa = -1.23113331702
	xb = 2.82781037233

Add workers to the app

- Adapt the contextualization script
  - not start the whole application, but just the worker
  - connect to the existing VM for messaging queue
- Create worker.sh with the following content:

```
#!/usr/bin/env bash
curl -L -s http://git.openstack.org/cgit/stackforge/faafo/
plain/contrib/install.sh | bash -s -- \
    -i faafo -r worker -e 'http://<VM IP>' -m 'amqp://
guest:guest@<VM IP>:5672/'
```

Use here your previous VM IP

Start Worker – On the same, or on a different cloud site

```
~$ OS_TPL=<OCCI ID from AppDB>

~$ RESOURCE_TPL=<Template ID from AppDB>

~$ occi --endpoint $ENDPOINT \
       --auth x509 --voms --user-cred $X509_USER_PROXY \
       --action create --resource compute \
       --mixin $RESOURCE_TPL --mixin $OS_TPL \
       --attribute occi.core.title="worker$(date +%s)" \
       --context user_data="file:///$PWD/worker.sh" \
       --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ WORKER_ID="..."
```

Only if different cloud site is used for Master and Worker

Accessing the appliance

- Create some fractals at master VM (Don't change localhost!)

```
~fractal $ faafo --endpoint-url http://localhost \
                 --verbose create --tasks 20
```

- Check in your worker the activity:

```
~worker $ top
```

- Check in your browser the "creator" of each fractal

TIP: If you want to make sure that the Master gives the task to the second worker then disable the faafo-worker process in the first VM:
Comment out the [faafo-worker] in /etc/supervisor/conf.d/faafo.conf and restart the supervisor: sudo service supervisor restart.
This will make all the fractal generation go to VM2.

EGI-Engage

Remember to delete your VMs!

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $MASTER_ID

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $WORKER_ID
```

EGI-Engage

Next EGI Community Event

- EGI Community Forum
- Bari, Italy
- 2015, Nov 10-13.
- http://cf2015.egi.eu/
- Types of contributions
  - Presentations, tutorials, workshops, demos, posters
- Main topics:
  - Community Engagement and Innovation
  - Virtual Research Environments
  - Data and Computing
  - Identity provisioning, Authentication, Authorization and Accounting
  - Open Science Commons

Thank you for your attention.

Questions?

www.egi.eu