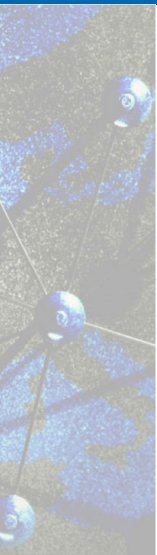


Managing virtual servers

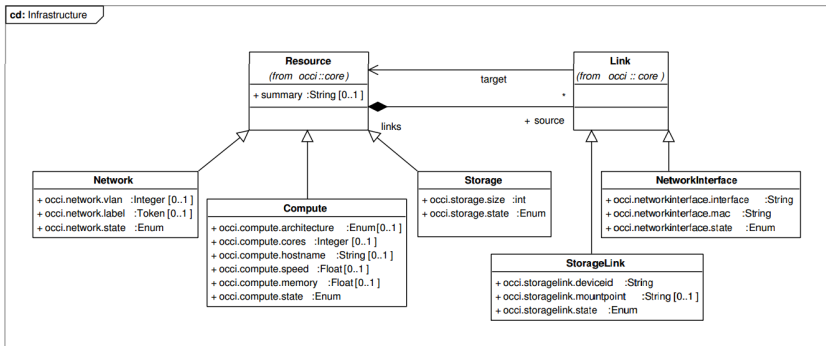
Enol Fernández, Boris Parák



- Brief introduction into OCCI
- Best practices for virtual machine management
- Hands-on part of the tutorial
 - Installation & configuration of rOCCI-cli
 - Basic life-cycle management tasks
 - Advanced management tasks
- Introduction to contextualization
- Using *cloud-init*
- Q & A

- OCCI → Open Cloud Computing Interface
- OGF standard; Core, Infrastructure and HTTP rendering (GFD.183 - 185)
- text-based protocol and API focusing on interoperability in the cloud
- designed with IaaS clouds in mind
- extensible; used for PaaS, SaaS, Brokering, ...

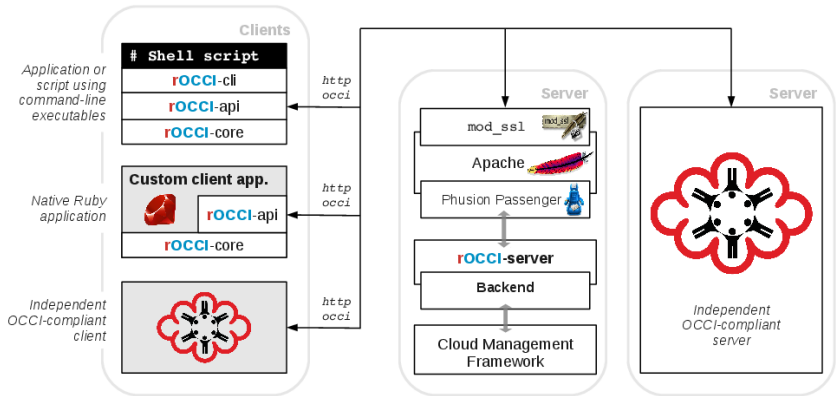




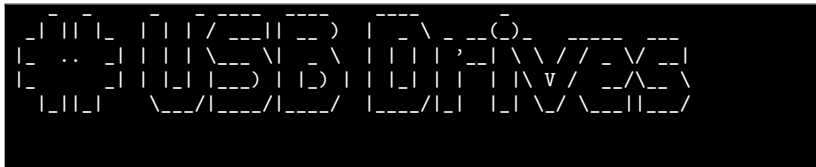
1. Creating VMs
2. Querying VMs
3. Destroying VMs
4. Triggering basic actions on VMs
5. Creating block storage
6. Destroying block storage
7. Attaching block storage to VMs
8. Detaching block storage from VMs
9. Attaching network interfaces to VMs
10. Detaching network interfaces from VMs
11. Using contextualization to modify VMs on boot

```
POST /compute/ HTTP/1.1

Category: compute;
  scheme="http://.../occi/infrastructure#";
  class="kind"
Category: debian7;
  scheme="http://.../infrastructure/os_tpl#";
  class="mixin"
Category: small;
  scheme="http://.../infrastructure/resource_tpl#";
  class="mixin"
X-OCCI-Attribute: occi.core.title="TestROCCI1"
X-OCCI-Attribute: occi.compute.cores=2
X-OCCI-Attribute: occi.compute.memory=1.7
```







root:roccitest
rocci:roccitest

`https://appdb.egi.eu/store/software/rocci.cli`

On CentOS or Scientific Linux:

```
~# EPEL=http://download.fedoraproject.org/pub/epel
~# wget $EPEL/6/x86_64/epel-release-6-8.noarch.rpm
~# yum install epel-release-6-8.noarch.rpm
~# yum install yum-priorities

~# UMD=http://repository.egi.eu/sw/production/umd/3
~# wget $UMD/sl6/x86_64/base/umd-release-3.0.0-1.el6.noarch.rpm
~# yum install umd-release-3.0.0-1.el6.noarch.rpm

~# APPDB=http://repository.egi.eu/community/software/rocci.cli
~# wget $APPDB/4.2.x/releases/repofiles/sl-6-x86_64.repo \
    -O /etc/yum.repos.d/rocci-cli-sl-6-x86_64.repo
```

`https://appdb.egi.eu/store/software/rocci.cli`

On Debian or Ubuntu:

```
~# IGTF=https://dist.eugridpma.info/distribution/igtf
~# apt-key adv --fetch-keys $IGTF/current/GPG-KEY-EUGridPMA-RPM-3
~# cat > /etc/apt/sources.list.d/eugridpma.list <<EOF
deb http://dist.eugridpma.info/distribution/igtf/current igtf accredited
EOF

~# KEY=http://repository.egi.eu/community/keys/APPDBCOMM-DEB-PGP-KEY.asc
~# apt-key adv --fetch-keys $KEY
~# APPDB=http://repository.egi.eu/community/software/rocci.cli
~# wget $APPDB/4.2.x/releases/repofiles/debian-wheezy-amd64.list \
    -O /etc/apt/sources.list.d/rocci-cli-wheezy.list

~# apt-get update
```

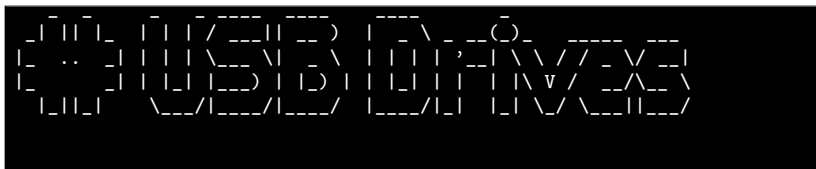
On CentOS or Scientific Linux:

```
~# yum install occi-cli  
~# /opt/occi-cli/bin/occi -d -v
```

On Debian or Ubuntu:

```
~# apt-get install occi-cli  
~# /opt/occi-cli/bin/occi -d -v
```

If you have **RVM** installed, disable it before running packaged
rOCCI-cli!



Copy your personal certificate into **\$HOME/.globus**

On CentOS or Scientific Linux:

```
~# yum install voms-clients lcg-CA fetch-crl
```

On Debian or Ubuntu:

```
~# apt-get install voms-clients ca-policy-igtf-classic \  
ca-policy-igtf-mics ca-policy-igtf-slcs \  
fetch-crl
```

```
~$ DNB="/DC=org/DC=terena/DC=tcs/OU=Domain Control Validated"
~$ VO="demo.fedcloud.egi.eu"
~$ cat >> ~/.voms/vomses <<EOF
"$VO" "voms1.egee.cesnet.cz" "15008" "$DNB/CN=voms1.egee.cesnet.cz" "$VO"
"$VO" "voms2.grid.cesnet.cz" "15008" "$DNB/CN=voms2.grid.cesnet.cz" "$VO"
EOF

~# VOMS_DIR=/etc/grid-security/vomsdir/demo.fedcloud.egi.eu
~# mkdir -p $VOMS_DIR

~# cat > $VOMS_DIR/voms1.egee.cesnet.cz.lsc <<EOF
/DC=org/DC=terena/DC=tcs/OU=Domain Control Validated/CN=voms1.egee.cesnet.cz
/C=NL/O=TERENA/CN=TERENA eScience SSL CA
EOF

~# cat > $VOMS_DIR/voms2.grid.cesnet.cz.lsc <<EOF
/DC=org/DC=terena/DC=tcs/OU=Domain Control Validated/CN=voms2.grid.cesnet.cz
/C=NL/O=TERENA/CN=TERENA eScience SSL CA
EOF
```

```
~$ voms-proxy-init -voms demo.fedcloud.egi.eu -rfc
~$ X509_USER_PROXY="..."

~$ occi --endpoint https://carach5.ics.muni.cz:11443 \
--auth x509 --voms --user-cred /tmp/x509up_u1000 \
--action list --resource resource_tpl
```

Successful execution of both commands is required!

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

More tomorrow morning, in
Best Practices for Cloud Application Architecture

Endpoints to choose from:

- <https://carach5.ics.muni.cz:11443> (CESNET)
- <https://cloud.ifca.es:8787> (CSIC)
- <https://cloud.cesga.es:3202> (CESGA)

```
~$ ENDPOINT=https://carach5.ics.muni.cz:11443
~$ occi --endpoint $ENDPOINT \
  --auth x509 --voms --user-cred $X509_USER_PROXY \
  --dump-model
```

```
~$ 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="os_tpl#uuid_metacloud_ubuntu_12_04_lts_x86_64_0001_fedcloud_dukan_72"  
~$ RESOURCE_TPL="resource_tpl#small"
```

OS templates to choose from:

- os_tpl#uuid_metacloud_ubuntu_12_04_lts_x86_64_0001_fedcloud_dukan_72 (CESNET)
- os_tpl#07c98683-8ccd-4001-80fd-3a8b83596a26 (CSIC)
- os_tpl#uuid_ubuntu_server12_04_193 (CESGA)

Resource templates to choose from:

- resource_tpl#small (CESNET)
- resource_tpl#m1-tiny (CSIC)
- resource_tpl#small (CESGA)

```
~$ occi --endpoint $ENDPOINT \  
  --auth x509 --voms --user-cred $X509_USER_PROXY \  
  --action create --resource compute \  
  --mixin $RESOURCE_TPL --mixin $OS_TPL \  
  --attribute occi.core.title="MyTutorialVM_$(date +%s)"  
~$ COMPUTE_ID="..."
```

```
~$ occi --endpoint $ENDPOINT \  
  --auth x509 --voms --user-cred $X509_USER_PROXY \  
  --action create --resource storage \  
  --attribute occi.storage.size="num(2)" \  
  --attribute occi.core.title="MyTutorialStorage_$(date +%s)"  
~$ STORAGE_ID="..."
```

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

Do **NOT** delete your VM right now!
You will need it in a few moments.

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

Networks to choose from:

- /network/6 (CESNET)
- /network/public (CSIC)
- /network/17 (CESGA)

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

```
~$ occi --endpoint $ENDPOINT \  
  --auth x509 --voms --user-cred $X509_USER_PROXY \  
  --action describe --resource $COMPUTE_ID  
  
~$ LINK_ID="..."  
  
~$ occi --endpoint $ENDPOINT \  
  --auth x509 --voms --user-cred $X509_USER_PROXY \  
  --action unlink --resource $LINK_ID
```



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

What?

Contextualization is the process of installing, configuring and preparing software upon boot time on a pre-defined virtual machine image (e.g. hostname, IP address, ssh keys, ...)

Why?

- Configuration not known until instantiation (e.g. data location).
- Private Information (e.g. host certs)
- Software that changes frequently or under development.
- Not practical to create a new VM image for every possible configuration.

How?

- Contextualization requires passing some data to the VMs on instantiation
- Standard OCCI API lacks such feature
- New mixins proposed and implemented in **rOCCI**-server, OCCI-OS and Synnefo (i.e. all current RPs of EGI's Fedcloud!)

User data:

```
Category: user_data;  
  scheme="http://schemas.openstack.org/compute/instance#";  
  class="mixin"  
  
X-OCCI-Attribute: org.openstack.compute.user_data="<base64 encoded data>"
```

Public SSH keys:

```
Category: public_key;  
  scheme="http://schemas.openstack.org/instance/credentials#";  
  class="mixin"  
  
X-OCCI-Attribute: org.openstack.credentials.publickey.name="key name"  
X-OCCI-Attribute: org.openstack.credentials.publickey.data="<the public key>"
```

Use `-T` option with rOCCI-cli:

```
~$ occi -e $ENDPOINT \  
  -n x509 -X -x $X509_USER_PROXY \  
  -a create -r compute \  
  -M $OS_TPL -M $RESOURCE_TPL \  
  -t occi.core.title="MyTutorialContextVM_$(date +%s)" \  
  -T public_key="<public key>" \  
  -T user_data="<user data>"
```

- OCCI extensions specifies how to pass data to the VM
- BUT not how the data will be available!
- Each RP has different mechanisms to provide the data:
 - metadata server at a fixed location
 - iso filesystem
 - file injection
 - ...
- Need a way to abstract this mess!

- Cloud-init abstracts these mechanisms and defines a format for the data
- It can:
 - configure network, users, ssh keys, filesystems, ...
 - install packages,
 - execute arbitrary commands,
 - execute user provided scripts,
 - configure VM with puppet or chef,
 - ...
- It also provides mechanisms for extension.

- Supports:
 - EC2 metadata server (used by OpenStack)
 - NoCloud data format as used by Synnefo.
 - OpenNebula context.sh format included in v0.7.3
 - Since v0.7.5 also supports base64 encoded data as provided in FedCloud.
- Packages available for most Linux distributions: ubuntu/debian, SL5/SL6 (in EPEL), SUSE, ...

Create key (if you don't have one):

```
~$ ssh-keygen -f test.key
```

Create simple script:

```
#!/bin/sh
echo "Hello World." > /root/context.txt
echo "The time is now $(date -R)!" >> /root/context.txt
```

Launch VM with rOCCI-cli:

```
~$ occi -e $ENDPOINT \  
-n x509 -X -x $X509_USER_PROXY \  
-a create -r compute \  
-M $OS_TPL -M $RESOURCE_TPL \  
-t occi.core.title="MyTutorialContextVM_$(date +%s)" \  
-T public_key="file://$PWD/test.key.pub" \  
-T user_data="file://$PWD/script.sh"
```

Check results:

```
~$ ssh -i test.key ubuntu@193.144.35.84  
ubuntu@ip-193-144-35-84:~$ sudo cat /root/context.txt  
Hello World.  
The time is now Mon, 19 May 2014 13:25:55 +0000!
```

```
#cloud-config

mounts:
  - [ xvdc, /mnt, "auto", "defaults,nobootwait", "0", "0" ]

users:
  - default
  - name: cloudy
    gecos: Cloud Foo Bar
    groups: users
    lock-passwd: false
    # created with mkpasswd --method=SHA-512 --rounds=4096
    passwd: $6$rounds=4096$cXbVrsbbRJvH$4hcoldR.RWR9mU21ewfSuEYau52Z4akm/Pbe/0vLP!

packages:
  - cowsay

runcmd:
  - echo "The time is now $(date -R)!" > /root/context.txt
  - [ wget, "http://slashdot.org", -O, /tmp/index.html ]
```

Launch VM with rOCCI-cli:

```
~$ occi -e $ENDPOINT \  
-n x509 -X -x $X509_USER_PROXY \  
-a create -r compute \  
-M $OS_TPL -M $RESOURCE_TPL \  
-t occi.core.title="MyTutorialContextVM_$(date +%s)" \  
-T public_key="file://$PWD/test.key.pub" \  
-T user_data="file://$PWD/cloud-init.cfg"
```

FedCloud

- FedCloud – <https://wiki.egi.eu/wiki/Fedcloud-tf:Main>
- FedCloud Users – <https://wiki.egi.eu/wiki/Fedcloud-tf:Users>
- AppDB Cloud – <https://appdb.egi.eu/browse/cloud>

OCCI

- OCCI – <http://occi-wg.org/>
- rOCCI-cli – <https://github.com/EGI-FCTF/rOCCI-cli>
- rOCCI-server – <https://github.com/EGI-FCTF/rOCCI-server>
- OCCI-OS – <https://wiki.openstack.org/wiki/Occi>

Contextualization

- contextualization @ FedCloud – <https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Contextualisation>
- cloud-init – <http://cloudinit.readthedocs.org/>

Packages in AppDB

- rOCCI-cli – <https://appdb.egi.eu/store/software/rocci.cli>
- cloud-init – <https://appdb.egi.eu/store/software/fedcloud.cloud.init>

?