

# **Disposable elastic clusters**

with CernVM

Dario Berzano, George Lestaris  
CERN PH-SFT

ALICE Offline Week - Genève, 20.03.2014



- Many resources nowadays available in **cloud form**
- CERN has **OpenStack** where every user can get a quota of VMs

## Paradigm shift

Website:  
[openstack.cern.ch](http://openstack.cern.ch)

- From **pet computing** to **cattle computing**  
→ *VMs are unimportant: you don't care if you lose one of them*
- From **centrally managed** clusters to **personal clusters**  
→ *Admins provide resources: easy for users to do self-servicing*

*Build cloud-aware applications for opportunistic resources usage*



- Ensures a **consistent environment** for your software
- Clear **separation of administrative domains**
- Support different use cases on the same hardware infrastructure, and rapidly move resources between them: **multi-tenancy**
- **Opportunistic exploitation** of some resources instead of leaving them idle: a good example are **HLT farms**



- Virtualization exposes to the VMs the **lowest common denominator** of hardware features (such as CPU specs): no architecture-specific **optimization** possible
- However loss of performances is **invisible** in most cases:
  - **near zero** loss on CPU bound tasks
  - Grid jobs slowed down by **remote I/O**
- Virtualization is **appropriate** for some use cases (Grid-like) and **not suitable** for others (real-time applications, triggers, etc.)

*Grid-like applications: cloud has more benefits than drawbacks*

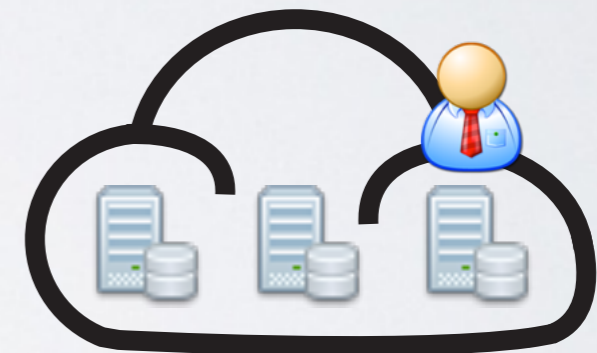
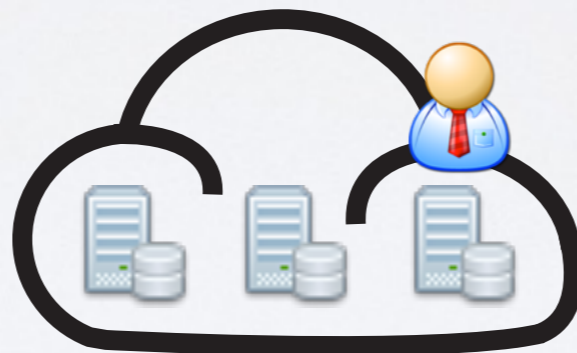
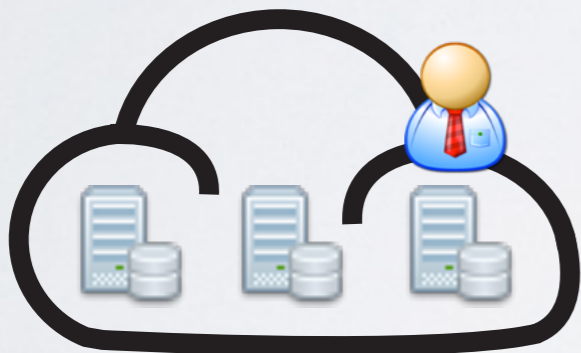
# Administrative domains in the cloud



virtual infrastructure administrator  
manages the VM configuration, does  
not see hardware failures



user  
uses the services as if they were  
physical, unaware of virtualization



administrators of distributed and independent clouds  
manage the hardware, replace disks when broken, monitor resources  
usage, coordinate multiple tenants (local and remote users)

# From Grid to clouds

- Local sites expose **standard\*** interfaces, such as the **EC2 API**  
*\* as in "industry standard" and not "HEP standard"*
- Experiments or users submit **fully configured** VMs
- Users can keep using the **same workflow** (e.g. Grid submission) and have **in addition** the possibility of **launching VMs directly**
- Only **generic** requirements for local cloud sites
- Experiments can **centrally** ensure **environment consistency** in a non-invasive manner for the local sites

*Clouds provide true abstraction of resources (CPU, RAM, disk...)*



## Cloud taxonomy: Everything-as-a-Service

- **Infrastructure as a Service:** ensemble of virtual machines exposing no particular service (you use them by **logging in**)
- **Cluster as a Service:** runs a batch system and exposes to users a job submission interface, *i.e. HTCondor, Work Queue, PROOF...*

## Virtual clusters:

- Can be **shared** or **personal**
- Can **elastically grow and shrink** based on the usage, in a completely transparent way for the user



- **CernVM-FS**: experiments software downloaded on demand, featuring aggressive caching with HTTP proxies
- **CernVM 3**: tiny (< 20 MB) virtual machine image, immediate to deploy: an OS on demand with root filesystem from CernVM-FS
- **Long Term Data Preservation**: can run **any snapshot** of the OS from the past like OSX **Time Machine** (*see Jakob's talk*)
- **Embedded elasticity**: no external services needed, works on any cloud
- **CernVM Online** → [cernvm-online.cern.ch](http://cernvm-online.cern.ch): web interface for VM and cluster creation

*Cluster creation truly for dummies with CernVM Online*





Your context definitions

Name	Operations	WebAPI
Condor Head Node	Clone Publish	Launch now
Condor Worker Node	Clone Publish	Launch now

**Create new context**    New context based on abstract

Users

Define the users your configuration will have

Username	Group	Home	Password	Operations
user		/home/user		Add user

Create new context from CernVM Online dashboard

Configure HTCondor

Define user accounts

Configure CernVM use the "devel" branch for now

CernVM Preferences

While Basic CernVM is sufficient to run a typical experiment software framework, you can optionally download or use extra tools and libraries to support software development) or full Desktop Edition which add a full X desktop environment and platforms where native X is not available).

Configuration URL:

CernVM Version:

CernVM Edition:

Expand root partition:  Use full root partition instead of the first 20 GB only

µCernVM branch:

Condor Batch ON

Setup Condor batch system.

Master hostname:

Shared secret:

Optional information:

Collector name:

HTCondor user:

HTCondor group:

HTCondor directory:

HTCondor admin:

Use IP addresses:

HTCondor UID domain:

HTCondor lowport:

HTCondor highport:

Create a context for the head node



Your context definitions

Name	Operations	WebAPI
Condor Head Node	Clone  Publish	Launch now ▾
Condor Worker Node	Clone  Publish	Launch now ▾

Create new context New context based on abstract ▾

Specify %ipv4% as HTCondor master will be substituted with the correct IP

Clone head node context many options are the same

### Condor Batch

Setup Condor batch system.

Master hostname:

Shared secret:

**Optional information:**

Collector name:

HTCondor user:

HTCondor group:

HTCondor directory:

HTCondor admin:

Use IP addresses:

HTCondor UID domain:

HTCondor lowport:

HTCondor highport:

Leave all other options untouched they are inherited from the head node

Create a context for the worker nodes



### Your cluster definitions

Name	Contexts	Operations
No cluster definition created yet		

**Create new cluster**

Create a new cluster definition

### EC2 API

API URL:

API Version:

AWS Access key:

AWS secret key:

---

### Virtual machines profile

Worker nodes' image:

Worker nodes' flavor:

SSH key name:

Amt. of jobs per VM:  Estimated VM deployment time (in sec.):

Batch system:

---

### Quota configuration

Min. workers:  Max. workers:

Insert your EC2 credentials, VM image, min/max number of workers...

### Master node context

No context selected

**Condor Head Node dberzano**

Condor Worker Node dberzano

---

No context selected

Select contexts of the head node and workers

*Create a cluster definition*

### Your cluster definitions

Name	Contexts	Operations
Condor Cluster for ALICE	<b>Master:</b> Condor Cluster for ALICE: head node	Clone
	<b>Worker:</b> Condor Cluster for ALICE: worker node	<b>Deploy cluster</b>

Just click the  
deploy button

Copy-paste the given command to  
spawn the whole cluster

### Deploy cluster

Using euca2ools from command-line    Using user-data field

```
euca-run-instances -t m1.medium -k 'CernVM-VAF' -d "$(echo W2FtaWVubmZpZ10KcGx1Z2luc21jZXJudm0KW2Nlcm52bV0KY29udGV4dHVhbGl6YXRpb25fa2V5PTY4MmE0MzI1YjY1YjQ0MmE4ZmM0YjQ2ZTIzZWVjN2ZjC1t1Y2VybnZtLWJlZ2luXQpjdmc1mc19odHRwX3Byb3h5PSJESVJFQ1QiCnJlc2l6ZV9yb290ZnM9dHJlZQpjdmc1mc19icmFuY2g9Y2VybnZtLWRldmVsLmNlc m4uY2gKY3ZtZnNfc2VydmVpPWhlCHZtLmNlcm4uY2gKW3VjZXJudm0tZW5kXQo=|base64 --decode)" ami-00000207
```

Paste this in the command line

If you don't have euca2ools installed,  
do it from [lxplus.cern.ch](http://lxplus.cern.ch)

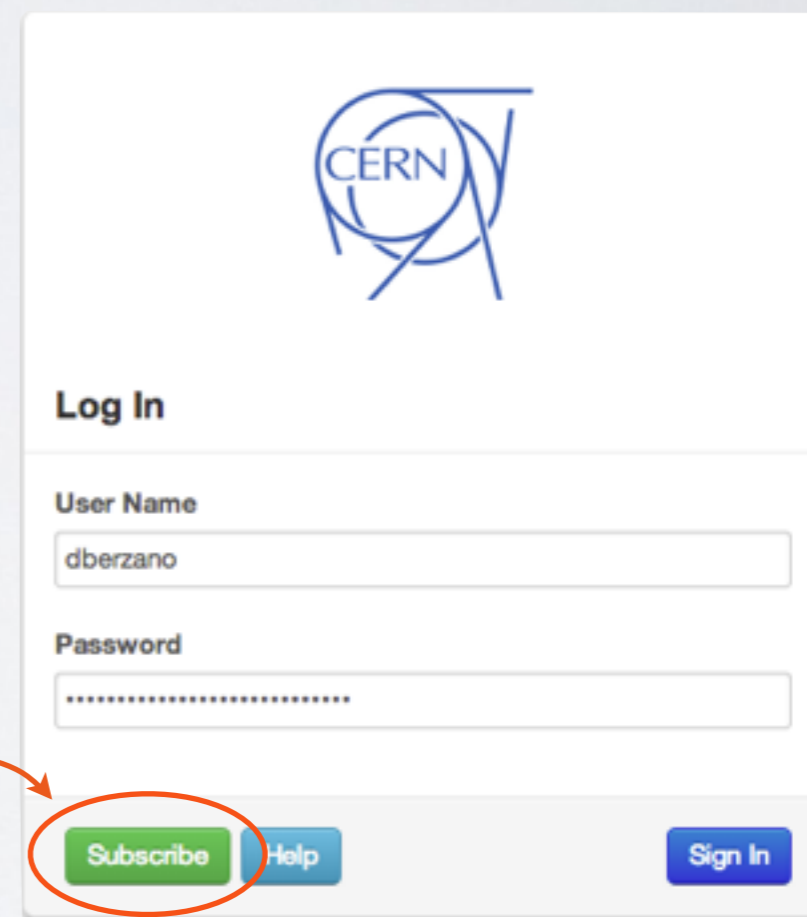
*Deploy the whole cluster by copying-pasting a single command*

- QA cluster: see Stefan Roiser's talk
- Personal cluster for running your batch jobs with AliRoot versions not available on your laptop
- Run PROOF on top of the elastic cluster via PROOF on Demand: can be a sustainable replacement of current AAF model

*Sensible usage of computing resources via embedded elasticity*

- On **CERN's OpenStack** → [openstack.cern.ch](http://openstack.cern.ch): every CERN user has her own quota
- On **your institute's cloud** (i.e. at INFN Torino you can → [chep2013.org/contrib/474](http://chep2013.org/contrib/474))
- On **public clouds**, such as Amazon EC2
- On **opportunistic clouds**, i.e. HLT farms when they are idle

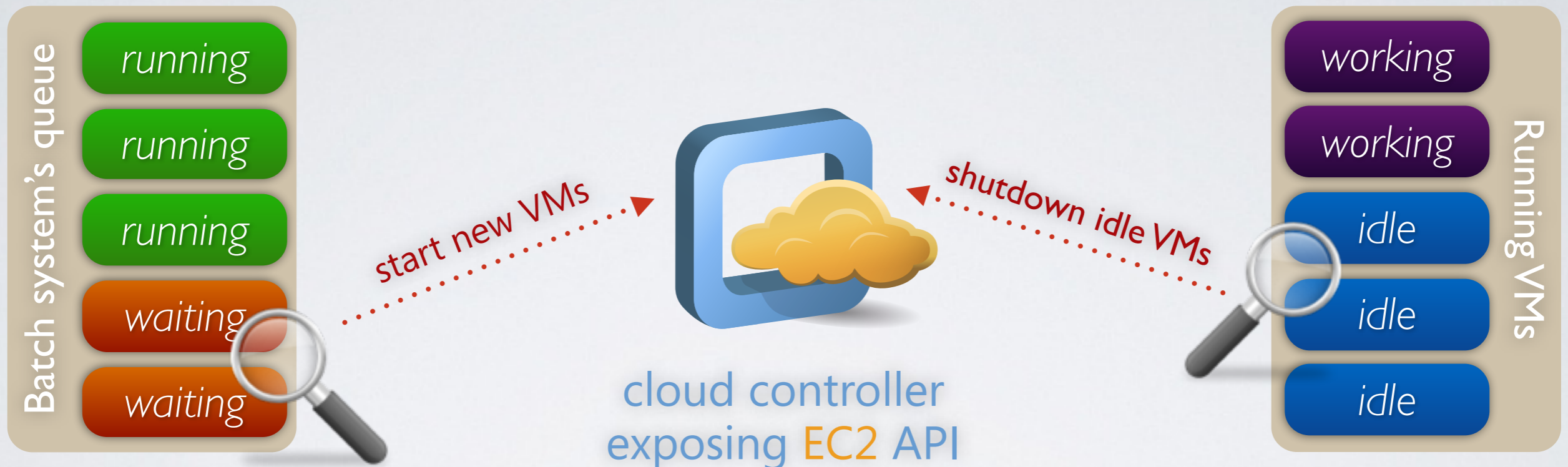
You have to subscribe to CERN's OpenStack before use



The screenshot shows the CERN OpenStack login interface. At the top is the CERN logo. Below it is the heading "Log In". There are two input fields: "User Name" with the text "dberzano" and "Password" with a masked password. At the bottom, there are three buttons: "Subscribe" (highlighted with a red circle and an arrow from the text box), "Help", and "Sign In".



*elastiq is a Python app monitoring the queue to bring elasticity*



Jobs waiting too long will trigger a **scale up**

Supports minimum and maximum quota of VMs

You deploy only the master node: minimum quota immediately launches VMs automatically

Integrated in CernVM 3  
source: [github.com/dberzano/elastiq](https://github.com/dberzano/elastiq)

- **Zero configuration:** once the VMs are launched, no further configuration needs to be performed
- **Sandboxing:** if a user's PROOF server crashes, others are not affected
- **Self-servicing:** user restarts her own PROOF server without bothering the administrators

Virtual Analysis Facility

☰ ON

*Configure the authentication method and the experiment settings for using the CernVM Virtual Analysis Facility.*

### Authentication

**Authentication method:** ALICE LDAP ▾

**Enable HTTPS+SSH authentication:** Yes ▾

### PROOF and PoD

**Client settings:** ALICE ▾

**URL template (i.e. root://server/<path>) or Storage Element (i.e. ALICE::CERN::EOS):** alien://<path>

Additional configuration for PROOF on CernVM Online





- **Users scheduling delegated to HTCondor:** no more assignment of resources that are not available for real
- **External storage:** on CAF if a host is down its data is unavailable
- **No more datasets:** list of files created dynamically from the AliEn file catalog and cached for faster subsequent requests

*It can be a sustainable alternative to current static AAFs*

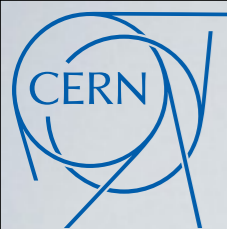
# Issues and solutions

## EC2 credentials need to be embedded in the head node

- context is **stored and transferred encrypted** by CernVM Online
- It is possible to create **per-cluster EC2 credentials** with OpenStack:  
`keystone ec2-credentials-create`
- If compromised, such credentials can be easily **revoked**:  
`keystone ec2-credentials-delete --access <created_access_key>`

## Some VMs might never boot and go to error state

- elastiqa will take care of **cleaning them up** (from release v0.9.3 due next week)



# Please Try This At Home (or here)



- **How to create a CernVM virtual cluster**  
*cernvm.cern.ch/portal/elasticclusters*
- **Use PROOF on the CernVM virtual cluster**  
*www.to.infn.it/~berzano/cloud/vaf\_guide.html*
- **Get access to CERN's OpenStack**  
*openstack.cern.ch*
- **Create your cluster on CernVM Online**  
*cernvm-online.cern.ch*
- **Report issues**  
*github.com/dberzano/elastiq/issues*
- **Subscribe to the CernVM discussion list**  
*cernvm-talk e-group*