



xBatch: Extending the CERN Batch Service into the Public Cloud

Pedro Andrade, Nacho Barrientos, Jérôme Belleman, Miguel Coelho dos Santos, Cristovao Cordeiro, Laurence Field, Domenico Giordano, Manuel Guijarro, Ben Jones, Gavin McCance, David Moreno Garcia, Iain Steers, Steve Traylen, Jan van Eldik

Outline

Context

Strategy

Provisioning

Configuration

Context

Current Capacity

- LSF: 80k cores for grid and local
- Condor: 15k cores for grid
 - 4 LHC VOs
 - ILC
 - COMPASS

(Condor) Capacity to increase further

Condor Setup

- Submit via HTCondor-CEs, local schedds next
- Virtual nodes
- Shared membership secret in the pool using Serf
- cloud-init to bootstrap Puppet
- Puppet master
- PuppetDB/Foreman for life cycle and inventory

Strategy

Goals

- Deploy Condor worker nodes in external clouds
- Transparent to the user, same entry point
- Flat capacity first, opportunistic spot-style later
- Batch, then other services?

Components

- Provisioning with life cycle
- Monitoring
- Accounting
- Configuration management

Requirements

- Separate Puppet instance
- Separate instance for secrets
- State management
- Host name registration
- Distinguishable VMs and bare metal servers
- Software repositories
- Data access

Non-Requirements

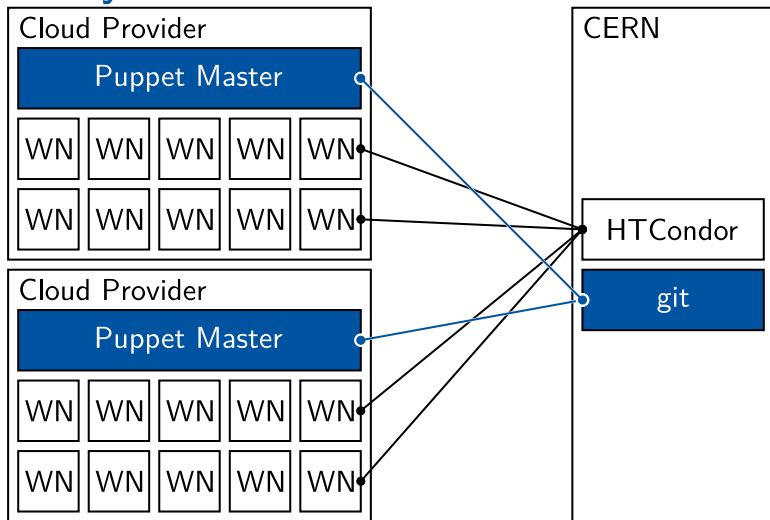
- Local storage
- Public IP addresses
- Active Directory

Approaches

- Same tools at CERN and in clouds
- Opportunity to simplify at CERN
- Deal with differences between cloud providers
- Try all components with each provider
- First try with Puppet master

Lean is good: try, fail, iterate

Visually...



Provisioning

Options

- Cloudify
- Terraform
- HTCondor (condor_annex, Condor-G)
- Home scripts

Plumbing

- Plumb into our existing setup
- Many plug-ins available in open solutions
- Choose one that makes plumbing easy

Providers

IBM SoftLayer:

- Bare metal and VMs
- For a few months

But also:

- Tested [Microsoft Azure](#)
- Deutsche Börse Cloud Exchange marketplace
- [HNSciCloud](#) marketplace
- ...

Cloudfify

Some deployment difficulties as:

- RPMs not in EPEL
- Requires dependencies to be installed manually
- Installs into /opt
- Missing Python modules
- Documentation not up-to-date?

Terraform

Tested with a few hundreds of nodes:

- Works fine with SoftLayer
- Easy to use
- Easy to integrate, e.g.:
 - Add resources to Foreman with local-exec provisioner
 - Run cloud-init/Puppet with a remote-exec provisioner
 - Destroy actions not yet ready, though
 - API differences may make using standard tools harder
- Slow since synchronous, but can run in parallel

HTCondor

- Strategic direction
- Well suited to public clouds
- Gaining expertise
- Fantastic community

Monitoring

- Lot of automation hanging off our existing one
- Start with Lemon
- Aggregate in the cloud?

Accounting (I)

Two views:

- Provider side, you get a bill
- Consumer side, you check what you used

Critical to reconcile the two, e.g.:

- VMs losing contact
- Failed starts
- Provider bugs

Accounting (II)

- Systematic benchmark during machine life cycle
- Can't measure physical machines up-front
- Also important to validate provider SLA
- Expect high variance in performance

Configuration

Configuration Management

- Distinct Puppet masters because of the trust
- Separate Puppet DB instance
- Shared Foreman?
- Write minimal ENC?

Authentication

- How to get certificates to nodes?
- Simple untrusted Puppet CA?
- Workers don't need grid certs → No CERN CA
- Untrusted CA?
- No Active Directory → No Kerberos, but X.509?

Host Name Registration

- After-boot registration?
- Something similar to adding nodes with Serf?

Outlook

Conclusion

- Terraform promising, let's see other providers
- Same tools at CERN and in clouds
- Opportunity to simplify

Other Services to Provision?

- CVMFS Squids
- Monitoring aggregators
- Storage
- ...



www.cern.ch