A modular (almost) automatic set-up for elastic multi-tenants cloud (micro)infrastructures

Physical hosts: servers hosting the Virtual Machines
- often called “hypervisors” (like the software KVM (OpenNebula supports also vCenter and Xen)
- monitoring daemons
- sshd for system connection

Networks
- Service and Storage network:
  - monitoring and control information
  - image transfers

Networks used by the Virtual Machines
- Private Network:
  - private IPs
  - intra-cloud communications

Public Network:
- public IPs
- incoming connectivity to VMs

OpenNebula

"VM director" (vs DIRAC “Pilot director”)
- starts VMs (vs DIRAC pilot jobs)
- VMs at boot time start "pilot jobs"
- instantiated VMs behave just as other WNs
- w1.1 DIRAC WMS

Integrate Federated cloud into DIRAC
- OCCI compliant clouds
  - OpenStack, OpenNebula
  - CloudStack
  - Amazon EC2

VMDIRAC architecture and components
- DIRAC server side
  - VM Scheduler – gets job status from TQ and match it with the proper cloud site, submit requests of VMs to Director
  - VM Manager – takes statistics of VM status and decides if needing new VMs
  - VM Director – connects with cloud manager to start VMs
  - Image context manager – contextualizes VMs to be WNs

VM side
- VM monitor Agent – periodically monitors the status of the VM and shutdows VMs when no need
- Job Agent – just like "pilot jobs", pulling jobs from TQ

Configuration and load management
- is used to configure the joined cloud and the image
- starts VMs
- runs jobs on VMs

VMDIRAC

VM scheduler
- manages dynamic virtual machines according to job status

Main functions
- Check Task queue and start VMs
- Contextualize VMs to be WNs to the DIRAC WMS
- Pull jobs from central task queue
- Centrally monitor VM status
- VMs automatic shutdown according to jobs queues

Storage
- Service datastores don’t necessarily need to be shared across VMs:
  - images can be transferred to the hypervisors’ disk through ssh and started locally
- Image Repository Datastore:
  - holds the OS images
- System Datastore:
  - holds the running instances
  - if it’s a shared FS, VMs can be “live-migrated”

Control node: runs the OpenNebula stack
- oned (the main daemon)
- sched (the VM scheduler)
- Sunstone (the web-based GUI)
- MySQL DB backend (can be separate)
- API services (OCCI or EC2)
- advanced services (OneFlow, OneGate, …)
- control node unavailability does not affect running VMs
- only control on them (start & stop, monitoring…)