Authentication, Authorization, and Contextualization in FermiCloud

S. Timm, D. Yocum, F. Lowe, K. Chadwick, G. Garzoglio, D. Strain, D. Dykstra, T. Hesselroth
Fermi National Accelerator Laboratory, Batavia, IL, United States of America

FermiCloud Production Hardware
23 nodes
Dual Intel Xeon E5640 “Westmere” quad core CPU
24GB of RAM
2 x 300GB SAS system disks
6 x 2TB SATA data disk in RAID5 configuration
High capacity LSI–1078 RAID controller,
can source 300MByte/sec of data per machine.
Dual GB Ethernet, 1 public network, one private.
Mellanox “ConnectX2” Infiniband adapter.

What is FermiCloud

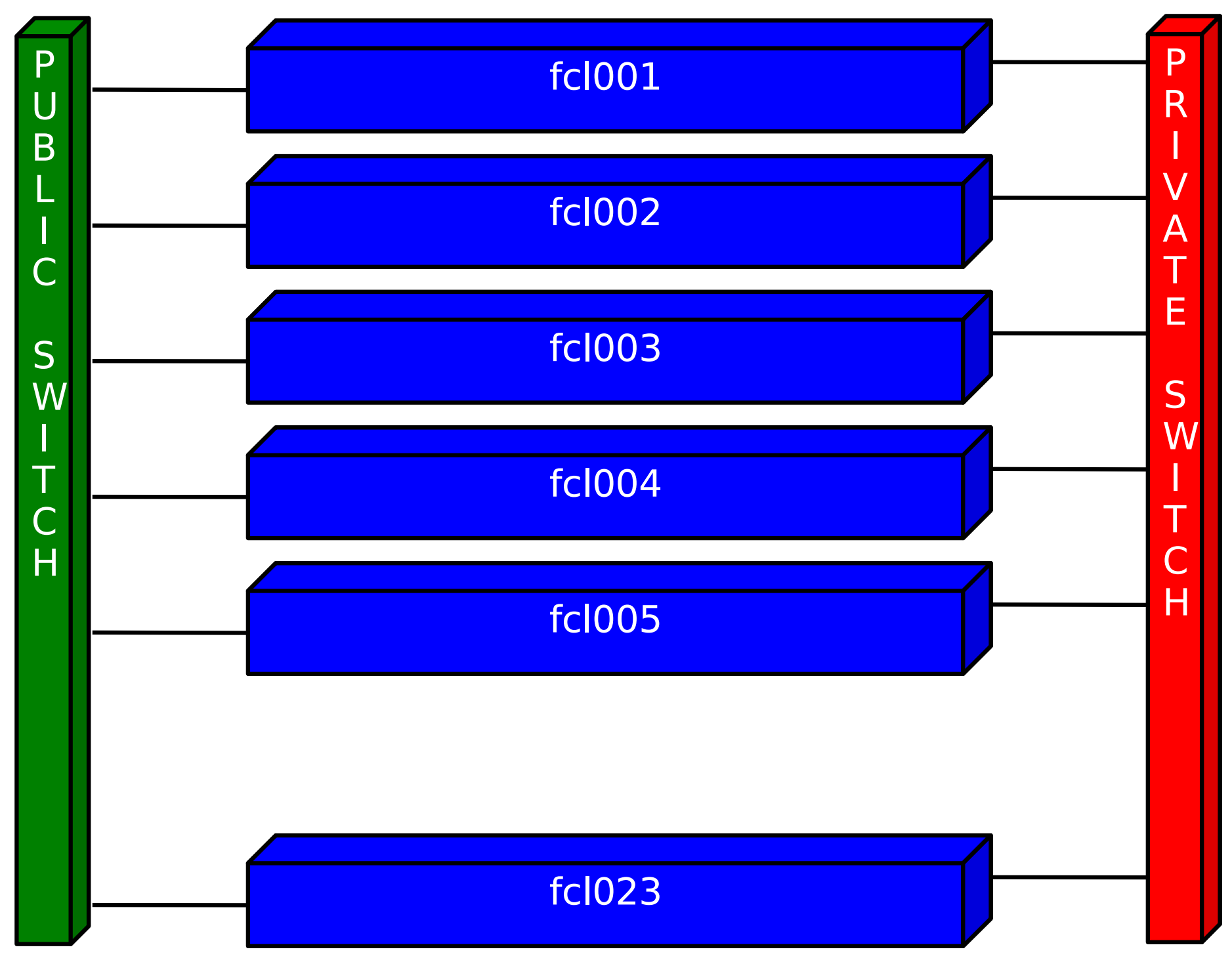
FermiCloud is a private cloud at Fermilab providing Infrastructure–as–a–service to our grid and storage developers, integrators, and testers, and also for production services. With the capacity to create virtual machines on demand, developers and integrators can use machines for just as long as they are needed, and the excess capacity can then be used by opportunistic scientific computing.

The FermiCloud Project has been evaluating open–source cloud software systems. One key feature being evaluated is the Authentication and Authorization that these systems use. Since these systems live on the Fermilab network they have to meet all security requirements that normal Fermilab machines do.

In our pilot service we are running OpenNebula and Eucalyptus, both of which are using the KVM hypervisor as shipped with Scientific Linux Fermi 5.5.

FermiCloud Network Topology

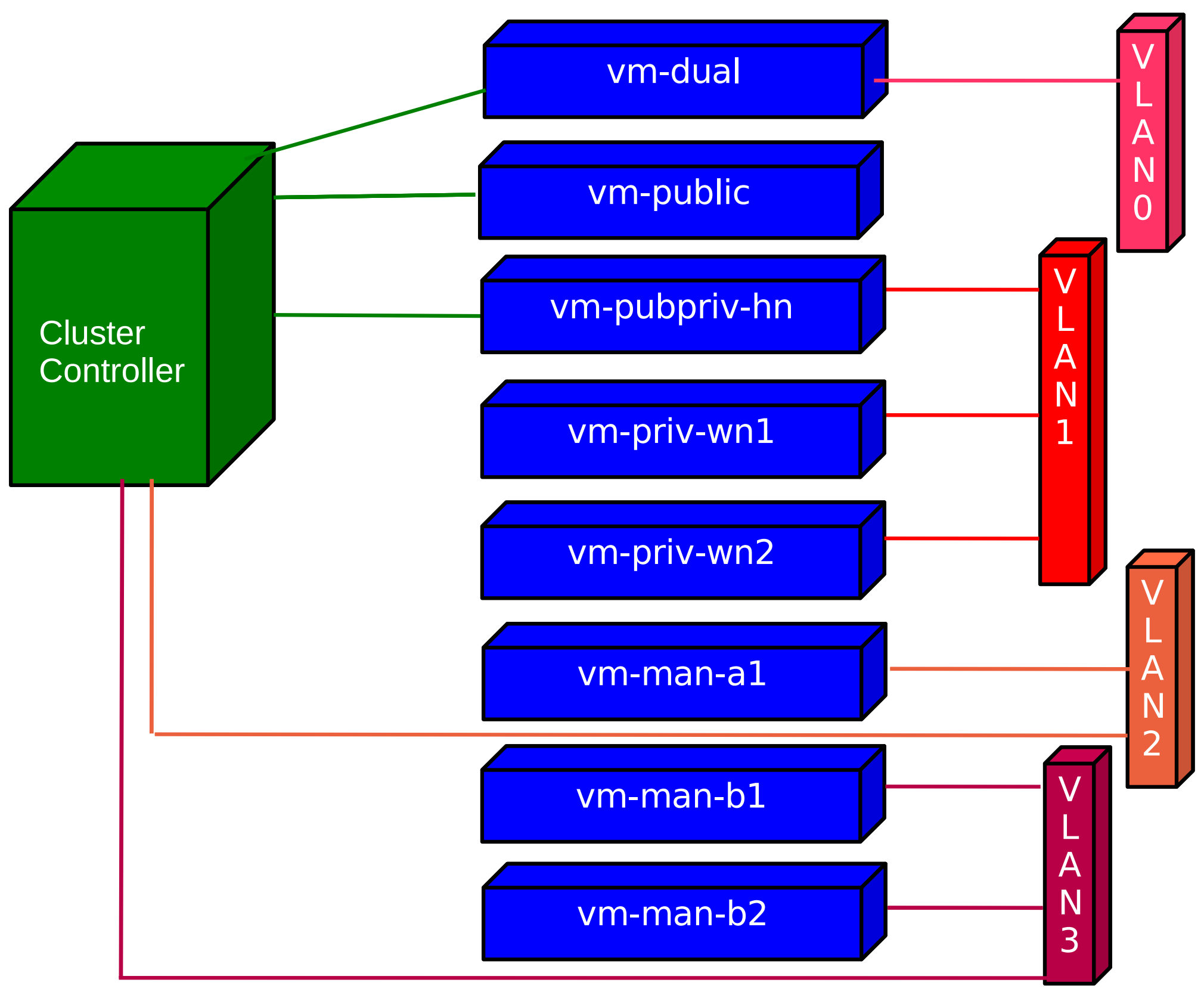
Physical Network Topology



All cloud host machines are connected to the main Fermilab public network with publicly–resolvable IP Addresses.

All cloud host machines also share a common private management network.

Logical Network Topology



Dual–bridged network private mgmt, public data

Fixed public IP every time, public–only
Virtual cluster, One VM with public and private, many VM's private only

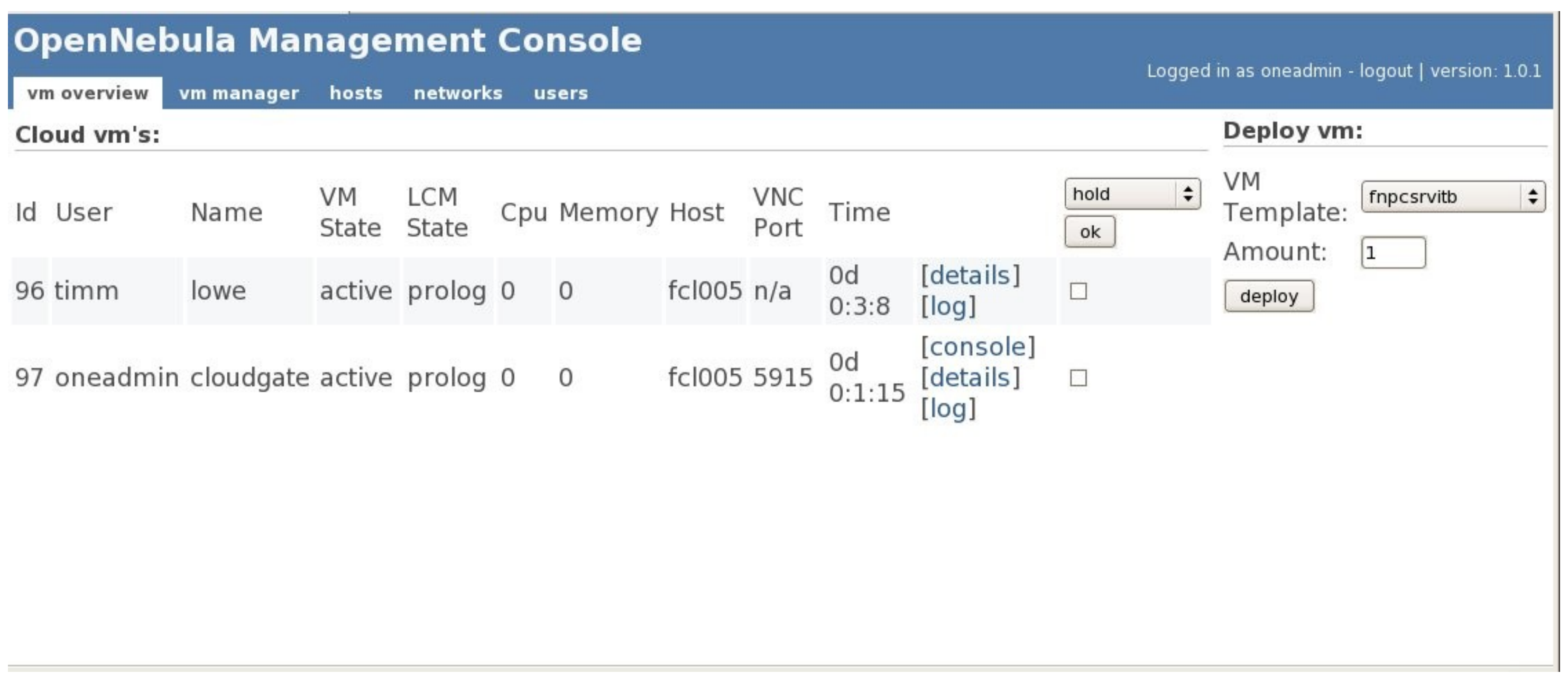
Amazon EC2 “Elastic IP” Managed on private network, cluster controller attaches public IP to instance and routes it via NAT. Different groups have independent private VLANS.

Cloud Authorization/Authentication

Authorization and Authentication in Cloud Computing has two basic categories. Cloud API's that create and manage virtual machines, and logging into virtual machines once they are running. For authentication, commercial clouds rely on having the billing information to identify the customer and prevent abuse. Private clouds need to rely on some other form of authentication such as X509 certificates or PKI/ssh infrastructure. The Amazon EC2 API is the *de facto* standard. They use a SOAP API which relies on X509 certificates to authenticate the user and the daemons to each other. They also have a REST API, also known as the query API, which uses a access key / secret key combination for authentication to launch and manage machines.

Authorization is accomplished by defining users and granting them privileges and priorities. These users are then given a X509 cert/key pair and an Access/secret key pair. Some clouds also have the feature of “security groups” in which the group can see each other's virtual images and share an isolated private network segment so that other customers can't interfere with their network traffic.

For logging into virtual cloud machines, the standard method is to generate an ssh public/private key pair based on the user's key, and insert it into authorized_keys on the virtual machine so that the user can access it as root.



OpenNebula Management Console

Logged in as oneadmin - logout | version: 1.0.1

Cloud vm's:

Id	User	Name	VM State	LCM State	Cpu	Memory	Host	VNC Port	Time		
96	timm	lowe	active	prolog	0	0	fcl005	n/a	0d 0:3:8	[details] [log]	□
97	oneadmin	cloudgate	active	prolog	0	0	fcl005	5915	0d 0:1:15	[console] [details] [log]	□

hold ok

Deploy vm:

VM Template: fnpcsrvitb
Amount: 1
deploy

FermiCloud Contextualization

Fermilab machines are required to use Kerberos 5 authentication for user logins. This means there is a machine–dependent secret (a kerberos host principal) which is stored on each machine. This can't be stored in the virtual machine repository. Machines that are grid hosts also have a X509 host certificate and private key, which also need to be stored independently.

We have written a small startup script which can be used with any cloud mechanism to detect the cloud machine's IP at startup, and fetch the machine–dependent secrets for that IP via SSL–encrypted wget.

We have also leveraged the contextualization methods for open–source clouds. For Eucalyptus we have used their Instance Metadata feature which makes information internally available on the http://169.254.169.254 URL.
OpenNebula allows attachment of a small ISO image to the main disk image at launch time. Machine–specific files can be loaded into this image, as can instructions about how to choose the IP address. We modified their stock scripts to make scripts that launched a virtual machine with the same IP address every time. This is a crucial feature for grid gatekeepers, which have the IP address hardwired in dozens of configuration files.

We provide users with pre–built OS images which they can then modify as needed. Only approved kernels and operating systems (Scientific Linux Fermi 4,5, STS (Fedora) and Windows) are allowed to be run. Authentication for login is restricted to be Kerberos 5, not the default ssh–keypair. Security patches are delivered via the normal Fermilab site–wide patching mechanism.

Default Authentication/Authorization of Common Open Source Clouds

CLOUD SYSTEM	Upload Image	Launch VM CLI	Launch VM API	Login
Eucalyptus	X509	X509	X509, EC2_ACCESS_KEY	ssh–keypair
Nimbus	X509	X509	X509 EC2_ACCESS_KEY	ssh–keypair
OpenNebula	user/pass	user/pass	user/pass EC2_ACCESS_KEY	ssh–keypair

X509 cert/key pairs are used in the cloud software, but not consistently. Self–signed certificates are used frequently, and the cloud software also relies on SimpleCA certificate authority. The user cert/key pairs that are generated are passwordless and thus vulnerable to being picked up by intruders. The EC2_ACCESS_KEY allows a user to perform many functions without X509 authentication at all.

OpenNebula is shown as username/password authentication above. Version 2.0 advertises pluggable authentication mechanism that allows X509 or Kerberos authentication among other methods.

FermiCloud Project Coming Enhancements in Authorization/Authentication

When new virtual machine is first run, scan for vulnerabilities and viruses like visiting laptop coming on site before giving it network access.

The cloud software must have a feature to periodically wake up dormant virtual machines to make sure they get their patches on a regular basis.

Make sure all GUI and CLI use X509 or Kerberos authentication to upload virtual machine images, launch virtual machines, and log into them.

Replace all self–signed and SimpleCA certs with IGTF–accredited certs.

Investigate integration of cloud X509 authentication with FermiGrid authentication services such as VOMS, GUMS

Investigate pluggable authentication mechanisms, make sure they work and decide which one to use.

Investigate inter–process communication and file transfer among the daemons of the cloud control software, and make sure it is done by secure protocols.

The Fermi National Accelerator Laboratory is operated by Fermi Research Alliance, LLC under Contract No. DE–AC02–07CH11359 with the United States Department of Energy.