

Xen Roll: Users Guide



Version 5.1 Edition



Xen Roll: Users Guide :

Version 5.1 Edition

Published Nov 2008

Copyright © 2008 University of California

This document is subject to the Rocks License (see Appendix A: Rocks Copyright).

Table of Contents

Preface	v
1. Overview	1
2. Installing	3
2.1. On a New Server	3
3. Using the Xen Roll	4
3.1. Installing a VM Server	4
3.2. Installing VM Containers	4
3.3. Installing Virtual Clusters.....	5
3.4. Physical Frontend with Virtual Compute Nodes.....	13
4. Command Reference	16
4.1. add	16
4.2. create	20
4.3. dump.....	20
4.4. list	21
4.5. move	24
4.6. pause.....	25
4.7. remove.....	25
4.8. report	27
4.9. restore	28
4.10. resume	29
4.11. save	29
4.12. set	30
4.13. start	33
4.14. stop	34
4.15. sync.....	35
A. Rocks Copyright	36
B. Third Party Copyrights and Licenses	38
B.1. xen.....	38

List of Tables

1-1. Summary..... 1
1-2. Roll Compatibility 1

Preface

The Xen Roll installs and configures the VMs on Rocks Clusters.

A physical frontend can configure VMs on client nodes (*VM container* appliances). A VM container is a physical machine that houses and runs VMs.

The Xen Roll also supports building virtual clusters. The frontend can be installed as a *VM server* appliance and the client nodes can be installed as VM containers. Then a virtual frontend can be installed on the VM server while virtual compute nodes can be installed on the VM containers. All network traffic is encapsulated within a unique VLAN, that is, each virtual cluster has its own VLAN.

Please visit the xen site¹ to learn more about their release and the individual software components.

Notes

1. <http://xen.org>

Chapter 1. Overview

Table 1-1. Summary

Name	xen
Version	5.1
Maintained By	Rocks Group
Architecture	i386, x86_64
Compatible with Rocks™	5.1

Table 1-2. Roll Compatibility

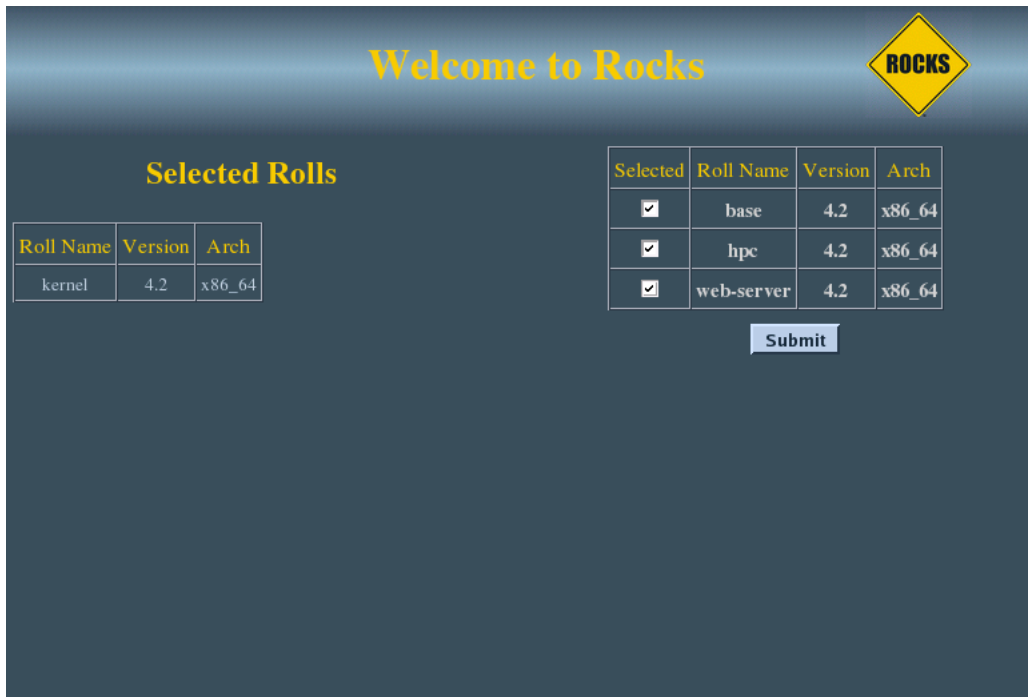
Roll	Requires ^a	Optional ^b	Conflicts
alpha		X	
area51		X	
base	X		
bio		X	
condor		X	
ganglia		X	
grid		X	
hpc		X	
java		X	
kernel	X		
os (disk 1)	X		
os (disk 2)	X		
os (disk 3)		X	
os (disk 4)		X	
os (disk 5)		X	
os (disk 6)		X	
os (disk 7)		X	
pbs		X	
service-pack		X	
sge		X	
viz		X	
web-server		X	
xen		X	

Roll	Requires ^a	Optional ^b	Conflicts
<p>Notes:</p> <ul style="list-style-type: none">a. You may also substitute your own OS CDs for the Rocks™ OS Roll CDs. In this case you must use all the CDs from your distribution and not use any of the Rocks™ OS Roll CDs.b. Only Rolls that have been verified as compatible with this Roll are listed. Other Rolls will likely work, but have not been tested by the maintainer of this Roll.			

Chapter 2. Installing

2.1. On a New Server

The xen Roll should be installed during the initial installation of your server (or cluster). This procedure is documented in section 1.2 of the Rocks™ usersguide. You should select the xen Roll from the list of available rolls when you see a screen that is similar to the one below.



Chapter 3. Using the Xen Roll

3.1. Installing a VM Server

A VM Server is machine that can house virtual frontend appliances. It is required to build a VM Server if you wish to build virtual clusters.

Building a VM Server is just like building a traditional frontend, except that you *must supply the Xen Roll during the frontend installation*. Follow the procedure *Install and Configure Your Frontend*¹ and be sure to supply the Xen Roll.

After you build the VM Server, you'll need to install VM Containers (see the next section).

3.2. Installing VM Containers

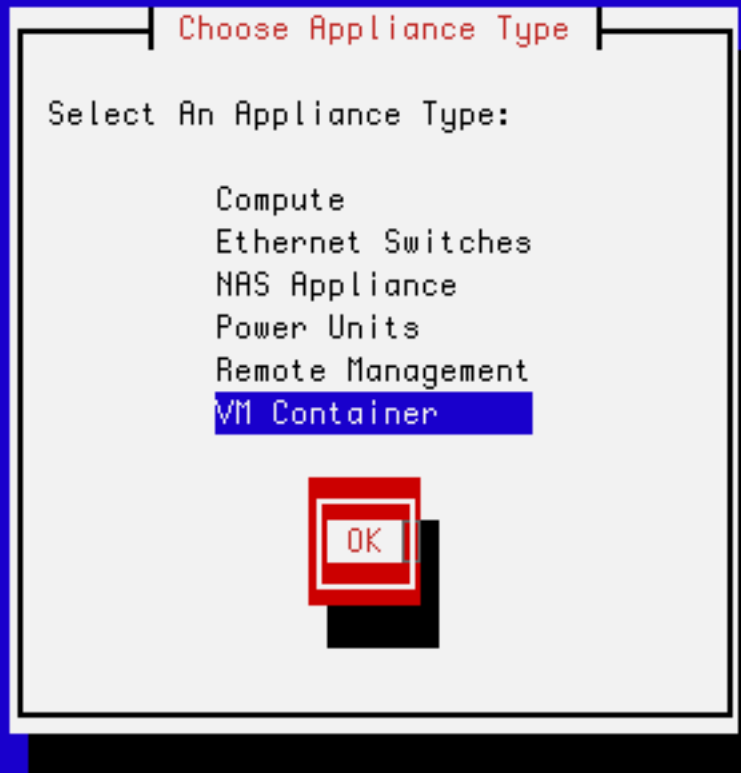
You will need to install a physical machine that will act as the "container" for your VMs. This method is very similar to the method for installing compute nodes.

On the frontend, execute:

```
# insert-ethers
```

You will see a screen that looks like:

```
Insert Ethernet Addresses -- version 5.0
Opened kickstart access to 10.0.0.0/255.0.0.0 network
```



Select the 'VM Container' appliance, then hit 'OK'.

Now PXE boot the physical machine that will be your VM container. Just like a compute node, the VM container will be recognized by insert-ethers and installed. The default name of the node will be `vm-container-X-Y`.

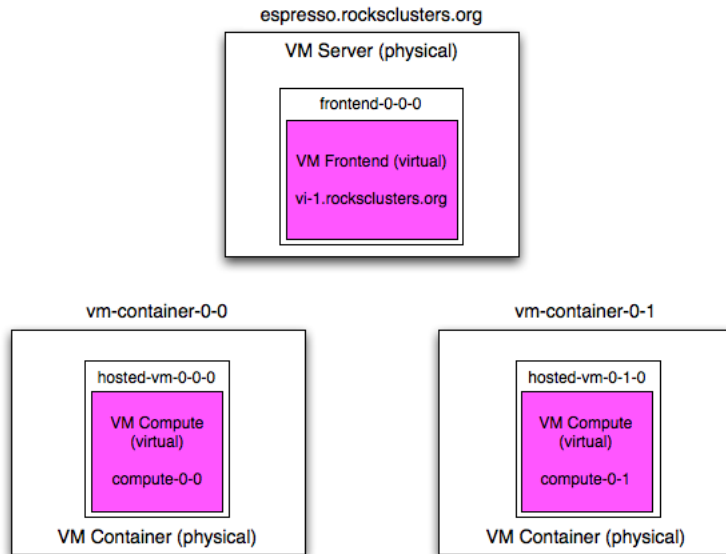
You can install as many VM containers as you like.

3.3. Installing Virtual Clusters

3.3.1. Installing a VM Frontend

After you install a VM Server and at least one VM Container, you are ready to install a virtual cluster.

We'll use the following illustration as a guide to help keep track of the names of the physical machines and the virtual machines.



In the above picture, "espresso.rocksclusters.org" is a physical machine. Also, "vm-container-0-0" and "vm-container-0-1" are physical machines that were kickstarted by "espresso". The machine "frontend-0-0-0" is a virtual machine that is hosted by "espresso". The machines "hosted-vm-0-0-0" and "hosted-vm-0-1-0" are VMs that are associated with "frontend-0-0-0" (they are all in the same VLAN). The names "frontend-0-0-0", "hosted-vm-0-0-0" and "hosted-vm-0-1-0" are names used by physical machine to control the VMs (e.g., to start and stop them).

The names in the virtual cluster look like the names in a traditional cluster -- the frontend is named "vi-1.rocksclusters.org" and its compute nodes are named "compute-0-0" and "compute-0-1". If you login to "vi-1.rocksclusters.org", you would be hard pressed to tell the difference between this virtual cluster and a traditional physical cluster.



You must select your own FQDN and IP address for your virtual frontend. The FQDN "vi-1.rocksclusters.org" and the IP address "137.110.119.118" are managed by UCSD and should not be used by you.

They are only used here to show you a concrete example.

First, we'll add a virtual cluster to the VM Server's database. In this example, we'll add a frontend with the FQDN of "vi-1.rocksclusters.org", IP of "137.110.119.118" and we'll associate 2 compute nodes with it:

```
# rocks add cluster fqdn="vi-1.rocksclusters.org" ip="137.110.119.118" num-computes=2
```

The above command will take some time and then output something similar to:

```
created frontend VM named: frontend-0-0-0
created compute VM named: hosted-vm-0-0-0
created compute VM named: hosted-vm-0-1-0
```

The command adds entries to the database for the above nodes and establishes a VLAN that will be used for the private network (eth0 inside the VM).

Info about all the defined clusters on the VM Server (including the physical cluster) can be obtained with the command: `rocks list cluster`:

```
# rocks list cluster
FRONTEND          CLIENT NODES      TYPE
espresso.rocksclusters.org: ----- physical
:                 vm-container-0-0 physical
:                 vm-container-0-1 physical
vi-1.rocksclusters.org: ----- VM
:                 hosted-vm-0-0-0 VM
:                 hosted-vm-0-1-0 VM
```

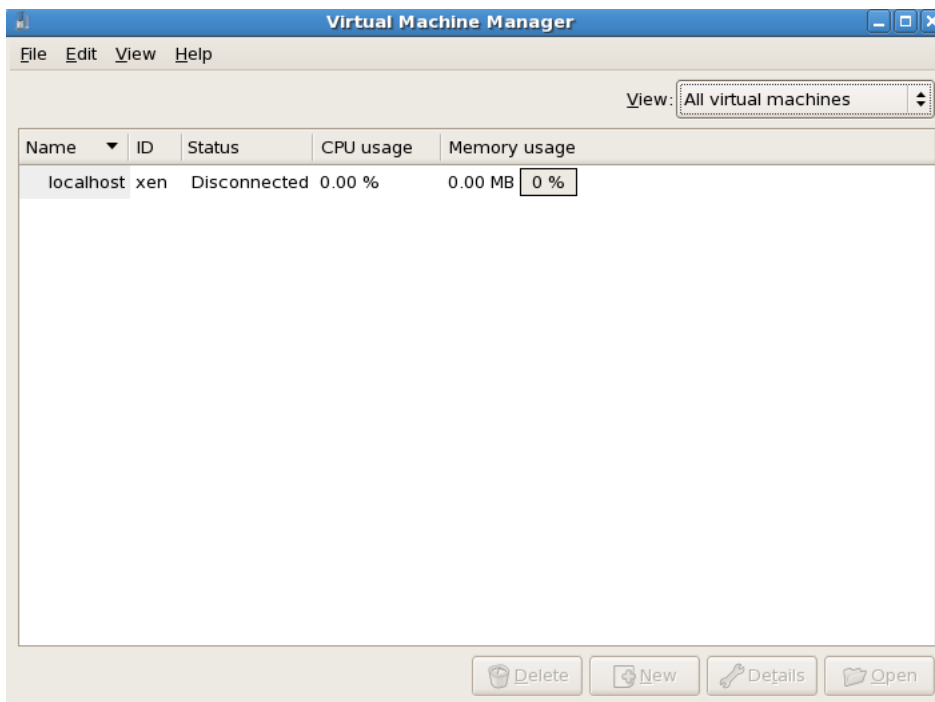
Now we need to install the VM frontend. This is done by "starting" the VM:

```
# rocks start host vm vi-1.rocksclusters.org install=yes
```

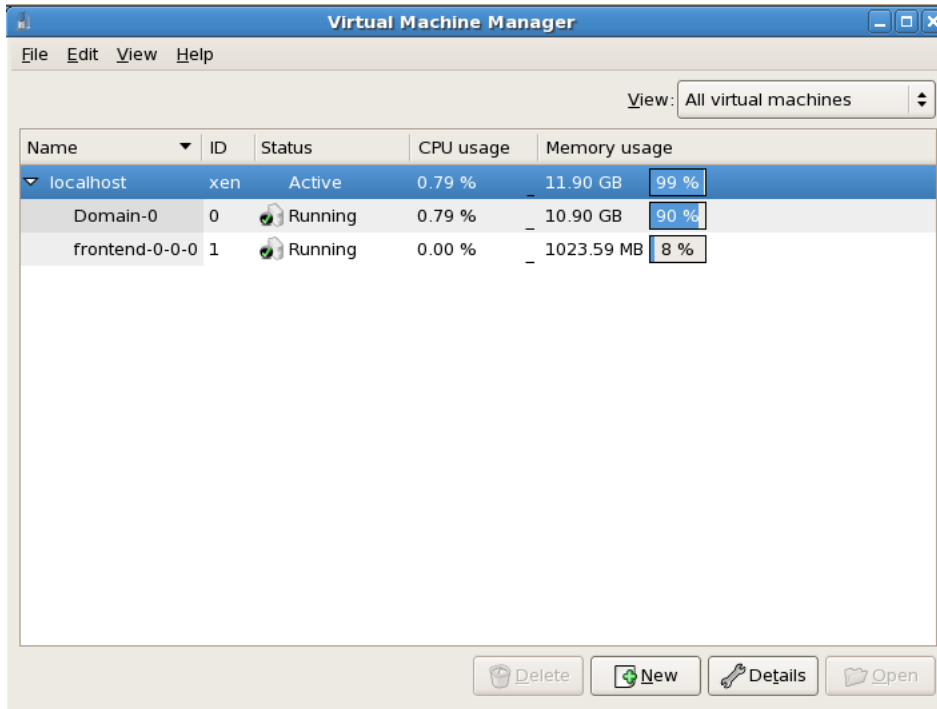
To interact with the VM frontend's console, you need to start "virt-manager"

```
# virt-manager
```

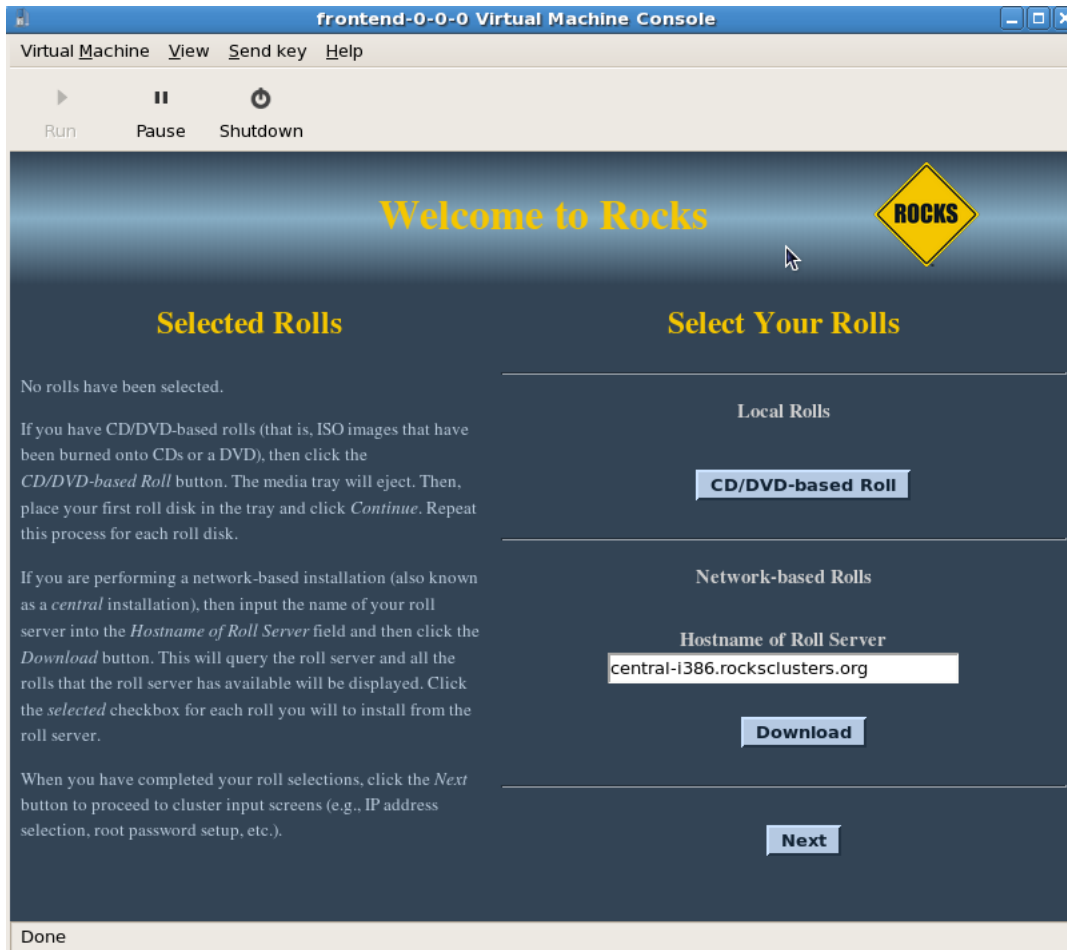
This will display a screen similar to:



Double click on the "localhost" entry and then you'll see:



To bring the up the console for the VM frontend, double click on "frontend-0-0-0". Now you should see the familiar frontend installation screen:



In the "Hostname of Roll Server" field, insert the FQDN of your VM Server (the name of the physical machine that is hosting the VM frontend). Then click "Download".

From here, you want to follow the standard procedure for bringing up a frontend² starting at Step 8.

After the VM frontend installs, it will reboot. After it reboots, login and then we'll begin installing VM compute nodes.

3.3.2. Installing VM Compute Nodes

Login to the VM frontend (the virtual machine named "vi-1.rocksclusters.org" in the example picture at the top of this page), and execute:

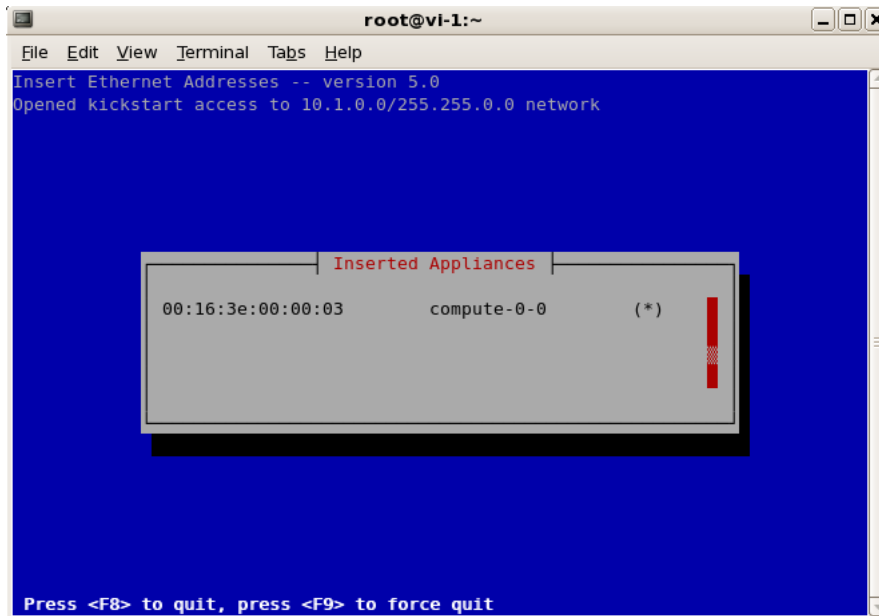
```
# insert-ethers
```

Select "Compute" as the appliance type.

Back on the VM Server, we'll need to start the first VM that hosts the VM compute node. In this example, the name is "hosted-vm-0-0-0".

```
# rocks start host vm hosted-vm-0-0-0 install=yes
```

Back in the VM frontend, you should see insert-ethers discover the VM compute node:



Repeat for each VM compute node.

You now have a virtual cluster up and running within your physical cluster.

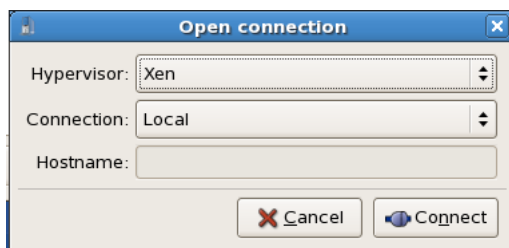
3.3.3. Connecting to a VM Compute Node's Console

You can connect to a virtual compute node's console in a manner similar to connecting to the virtual frontend's console. Again, bring up "virt-manager":

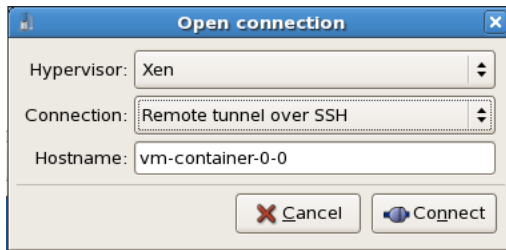
```
# virt-manager
```

We'll connect to the console for the virtual compute node "compute-0-0". In the example configuration described at the top of this page, The VM "compute-0-0" is hosted on the physical machine named "vm-container-0-0" so we'll need to tell "virt-manager" to open a connection to "vm-container-0-0".

Inside "virt-manager", click on "File" then "Open connection...". This brings up a window that looks like:

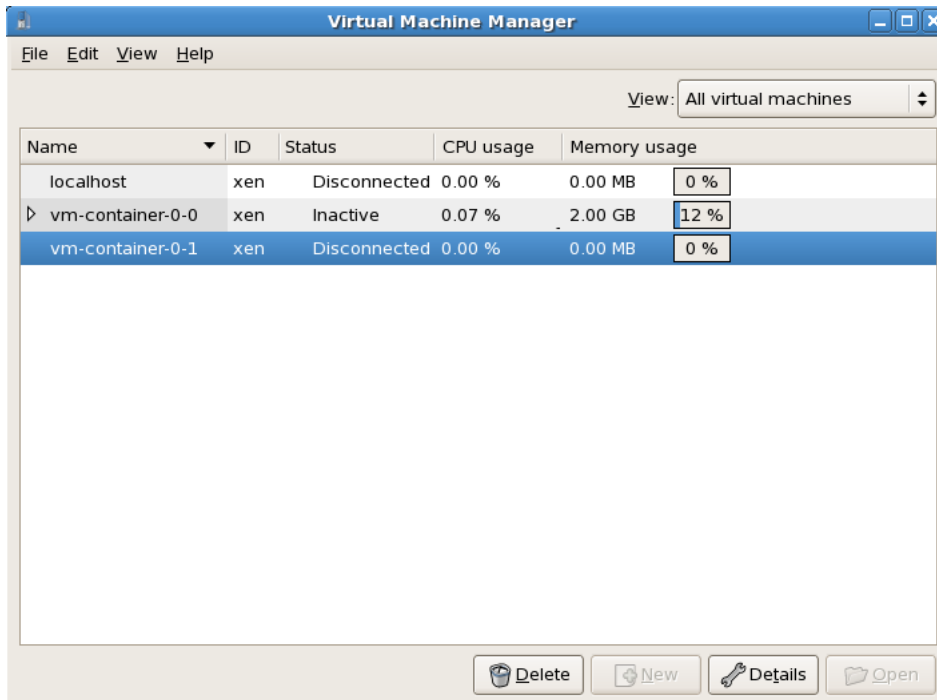


Now change the "Connection:" field to "Remote tunnel over SSH" and enter "vm-container-0-0" for the "Hostname:" field:

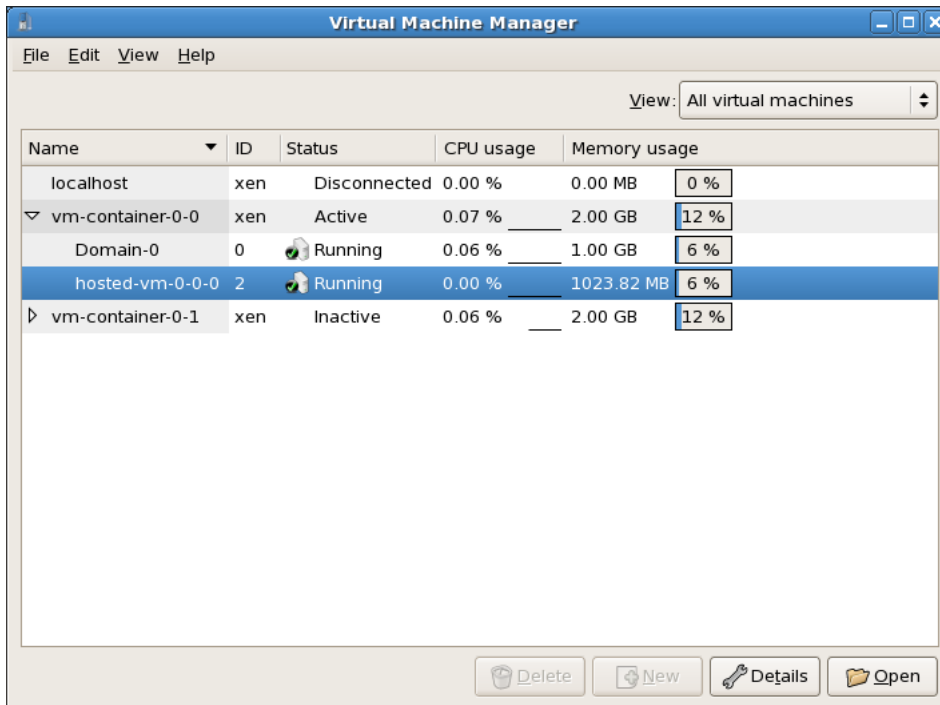


Then click "Connect".

In the "virt-manager" window, you should see something similar to:

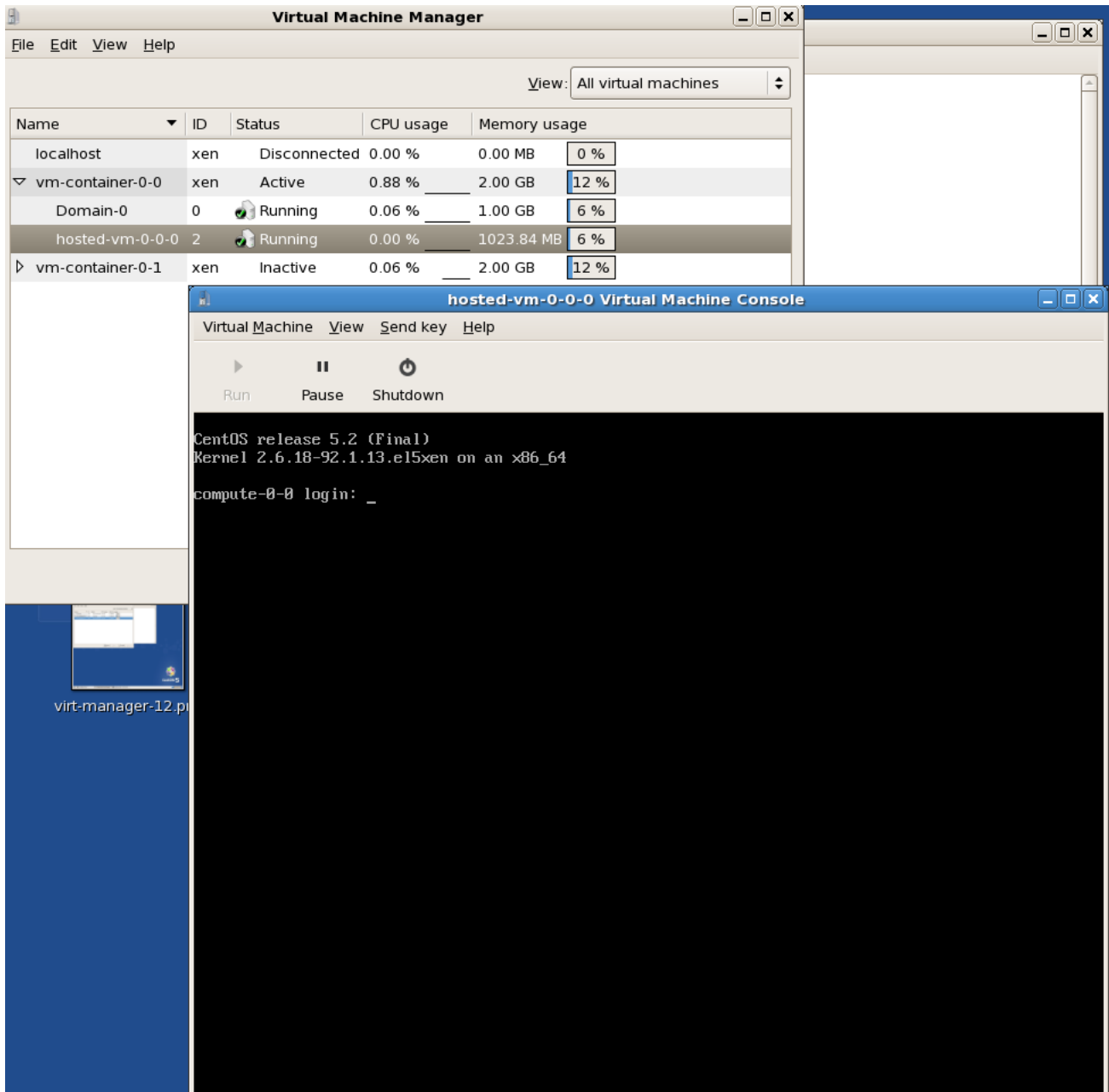


Double click on "vm-container-0-0" and then you'll see:



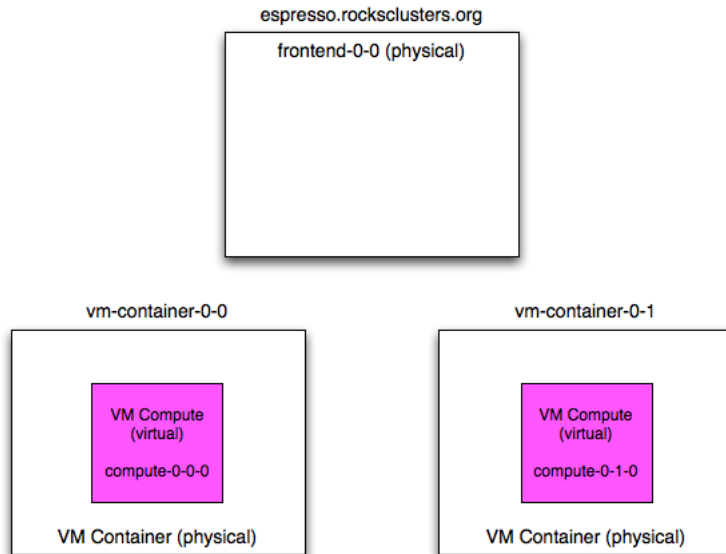
Now to connect to the compute node's console, double click on "hosted-vm-0-0-0". Recall that from the perspective of the physical frontend (the VM Server), "hosted-vm-0-0-0" is the name for the VM "compute-0-0" (again, see the figure at the top of this page).

You should now see the console for compute-0-0:



3.4. Physical Frontend with Virtual Compute Nodes

In this scenario, the frontend is a physical machine (not a VM) and the compute nodes are virtual machines.



In the above picture, "frontend-0-0" is a physical machine (with the public name of "espresso.rocksclusters.org"). The physical machine "frontend-0-0" controls two VM compute nodes named "compute-0-0-0" and "compute-0-1-0". This means that "compute-0-0-0" and "compute-0-1-0" are configured by "frontend-0-0". This is opposed to the "virtual cluster scenario" (Installing Virtual Clusters), where the virtual frontend ("frontend-0-0-0") configured the VM compute nodes, and the physical machine that housed "frontend-0-0-0" only started and stopped the virtual compute nodes.

3.4.1. Adding, Installing and Booting VMs with a Physical Frontend

In the common case, you will execute three Rocks commands over the lifetime of your VMs: add (to add VM info to the database), create (to install a newly added VM) and start (to boot an installed VM).

To add a VM to the system, you need to associate a VM with a physical machine (i.e., a VM container) and you need to assign an appliance type to the VM. Here's an example:

```
# rocks add host vm vm-container-0-0 membership="Compute"
```

The above command will output a message similar to:

```
added VM compute-0-0-0 on physical node vm-container-0-0
```

This tells us that, in the database, the compute VM named "compute-0-0-0" has been assigned to the physical machine "vm-container-0-0".

The next step is to install the VM.

VMs are installed with the `start` command with the `install=yes` flag. Here's how to install the VM that was added above:

```
# rocks start host vm compute-0-0-0 install=yes
```

The above command will start the standard Rocks installation process for the VM named "compute-0-0-0". After the installation process initializes the network inside the VM, you can monitor the installation just like a physical

machine installation by executing:

```
# rocks-console compute-0-0-0
```

After the installation completes, the VM will reboot. After the VM boots, you can interact with the VM just like any other physical machine.

Notes

1. </roll-documentation/base/5.1/install-frontend.html>
2. </roll-documentation/base/5.1/install-frontend.html>

Chapter 4. Command Reference

4.1. add

4.1.1. add cluster

```
rocks add cluster {fqdn} {ip} {num-computes} [container-hosts=string] [cpus-per-compute=string] [disk-per-compute=string] [fqdn=string] [ip=string] [mem-per-compute=string] [num-computes=string] [vlan=string]
```

Add a VM-based cluster to an existing physical cluster.

arguments

fqdn

The FQDN for the virtual frontend.

ip

The IP address for the virtual frontend.

num-computes

The number of compute nodes VMs to associate with the frontend.

parameters

[container-hosts=*string*]

A list of VM container hosts that will be used to hold the VM compute nodes. This must be a space-separated list (e.g., container-hosts="vm-container-0-0 vm-container-0-1"). The default is to allocate the compute nodes in a round robin fashion across all the VM containers.

[cpus-per-compute=*string*]

The number of CPUs to allocate to each VM compute node. The default is 1.

[disk-per-compute=*string*]

The size of the disk (in gigabytes) to allocate to each VM compute node. The default is 36.

[fqdn=*string*]

Can be used in place of the fqdn argument.

[ip=*string*]

Can be used in place of the ip argument.

[mem-per-compute=*string*]

The amount of memory (in megabytes) to allocate to each VM compute node. The default is 1024.

[num-computes=*string*]

Can be used in place of the num-computes argument.

[vlan=*string*]

The VLAN ID to assign to this cluster. All network communication between the nodes of the virtual cluster will be encapsulated within this VLAN. The default is the next free VLAN ID.

examples

```
# rocks add cluster vm.cluster.org 1.2.3.4 2
```

Create one frontend VM and assign it the name 'vm.cluster.org' with the IP address '1.2.3.4' and create 2 compute node VMs.

4.1.2. add host vm

```
rocks add host vm {host...} {membership} [cpus=string] [disk=string] [disk-size=string] [ip=string] [mac=string] [mem=string] [membership=string] [name=string] [num-macs=string] [slice=string] [subnet=string] [vlan=string]
```

Add a VM specification to the database.

arguments

host

One or more physical host names.

membership

The membership to assign to the VM.

parameters

[cpus=*string*]

The number of CPUs to assign to this VM. The default is: 1.

[disk=*string*]

A disk specification for this VM. The default is:

file:/<largest-partition-on-physical-node>/xen/disks/<vm-name>.hda,hda,w

[disksize=*string*]

The amount of disk space in gigabytes to assign to the disk specification. The default is: 36.

`[ip=string]`

The IP address to assign to the VM. If no IP address is provided, then one will be automatically assigned.

`[mac=string]`

A MAC address to assign to this VM. If no MAC address is specified, the next free MAC address will be selected.

`[mem=string]`

The amount of memory in megabytes to assign to this VM. The default is: 1024.

`[membership=string]`

Can be used in place of the membership argument.

`[name=string]`

The name to assign to the VM (e.g., 'compute-0-0-0').

`[num-macs=string]`

The number of MAC addresses to automatically assign to this VM. The default is 1.

`[slice=string]`

The 'slice' id on the physical node. Each VM on a physical node has a unique slice number. The default is the next available free slice number.

`[subnet=string]`

The subnet to associate to this VM. The default is: private.

`[vlan=string]`

The vlan ID to set for each interface. If you supply multiple MACs (e.g., 'num-macs' > 1), you can specify multiple vlan IDs by a comma separated list (e.g., `vlan="2,3,4"`). To not specify a vlanid for a MAC, use the keyword 'none'. For example, if you want to specify a vlan ID for interface 1 and 3, but not interface 2, type: `vlan="2,none,4"`. The default is to not assign a vlan ID.

examples

```
# rocks add host vm
```

Create a default VM.

```
# rocks add host vm mem=4096
```

Create a VM and allocate 4 GB of memory to it.

4.1.3. add host vm bootprofile

```
rocks add host vm bootprofile [host...] [args=string] [kernel=string] [profile=string] [ramdisk=string]
```

Add a boot profile for virtual a host.

arguments

[host]

List of hosts to add profile definitions. If no hosts are listed, then the global definition for 'profile=name' is added.

parameters

[args=*string*]

The second line for a pxelinux definition (e.g., append ks initrd=initrd.img ramdisk_size=150000 lang=devfs=nomount pxe kssendmac selinux=0)

[kernel=*string*]

URL for the kernel associated with this profile (e.g., 'file:///boot/kickstart/xen/vmlinuz')

[profile=*string*]

Label name for the boot profile. You can see the profile label names by executing: 'rocks list host vm bootprofile [host(s)]'.

[ramdisk=*string*]

URL for the kernel associated with this profile (e.g., 'file:///boot/kickstart/xen/initrd-xen.iso.gz')

examples

```
# rocks add host vm bootprofile profile=install kernel="file:///boot/kickstart/xen/vmlinuz"
```

Add the global 'install' boot profile

```
# rocks add host vm bootprofile compute-0-0-0 profile=install
kernel="http://10.1.1.1/xenkernels/experimental-vmlinuz"
```

Add the 'install' profile for compute-0-0-0

related commands

list host vm boot

list host vm bootprofile

remove host vm bootprofile


```
set host vm boot
```

```
set host vm bootprofile
```

4.2. create

4.2.1. create host vm

```
rocks create host vm {host...}
```

Create a VM slice on a physical node. This command will configure a VM and install it. This can be used for the initial setup of a VM or to reconfigure an existing VM.

arguments

host

A list of one or more VM host names.

examples

```
# rocks create host vm compute-0-0-0
```

Create VM host compute-0-0-0.

4.3. dump

4.3.1. dump host vm

```
rocks dump host vm [host...]
```

Dump host VM information as Rocks commands.

arguments

[host]

Zero, one or more host names. If no host names are supplied, information for all hosts will be listed.

examples

```
$ rocks dump host vm compute-0-0-0
```

Dump VM info for compute-0-0-0.

```
$ rocks dump host vm
```

Dump VM info for all configured virtual machines.

related commands

add host vm

4.4. list

4.4.1. list cluster

```
rocks list cluster [cluster...] [status=bool]
```

Lists a cluster, that is, for each frontend, all nodes that are associated with that frontend are listed.

arguments

[cluster]

Zero, one or more frontend names. If no frontend names are supplied, information for all clusters will be listed.

parameters

[status=*bool*]

If true, then for each VM-based cluster node, output the VM's status (e.g., 'active', 'paused', etc.).

examples

```
$ rocks list cluster frontend-0-0
```

List the cluster associated with the frontend named 'frontend-0-0'.

```
$ rocks list cluster
```

List all clusters.

4.4.2. list host vm

rocks list host vm [host...] [showdisks=*bool*] [status=*bool*]

Lists the VM configuration for hosts.

arguments

[host]

Zero, one or more host names. If no host names are supplied, information for all hosts will be listed.

parameters

[showdisks=*bool*]

If true, then output VM disk configuration. The default is 'false'.

[status=*bool*]

If true, then output each VM's status (e.g., 'active', 'paused', etc.).

examples

```
$ rocks list host vm compute-0-0
```

List the VM configuration for compute-0-0.

```
$ rocks list host vm compute-0-0 compute-0-1
```

List the VM configuration for compute-0-0 and compute-0-1.

4.4.3. list host vm boot

rocks list host vm boot [host...]

Lists the names of the install and run bootprofiles for [host(s)]. The a named profile describes the combination of kernel, ramdisk args that would be used if that profile where referenced for by the The RunProfile references the combination when a VM is booted for running (usually this is None). The InstallProfile references the combination when the nodes is booted in install mode. When the RunProfile is None, the VM node will boot from the kernel defined in its image.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

```
$ rocks list host vm boot compute-0-0-0
```

List the active run and install profiles defined for compute-0-0-0.

```
$ rocks list host vm boot
```

List the active run and install profiles defined all known VM hosts.

related commands

add host vm bootprofile

list host vm bootprofile

remove host vm bootprofile

set host vm boot

set host vm bootprofile

4.4.4. list host vm bootprofile

```
rocks list host vm bootprofile [host...] [global=bool] [profile=string]
```

Lists the current VM Profiles available for hosts. For each host supplied on the command line print the hostname and defined profiles available to that host The Profile describes the combination of kernel, ramdisk args that would be used if that profile were referenced for by the Host as either a RunProfile or an InstallProfile Usually, the RunProfile is empty and the VM node will boot from the kernel defined in its image.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

parameters

[global=*bool*]

If true, ignore the list of hosts and only return information about global bootprofiles. Default is 'n'

`[profile=string]`

List only the profile named 'profile'

examples

`$ rocks list host vm bootprofile compute-0-0-0`

List the profiles available to compute-0-0-0.

`$ rocks list host vm bootprofile`

List the profiles available for all known VM hosts.

`$ rocks list host vm bootprofile global=yes`

List only the globally-defined profiles without generating a list of hosts.

related commands

`add host vm bootprofile`

`list host vm boot`

`remove host vm bootprofile`

`set host vm boot`

`set host vm bootprofile`

4.5. move

4.5.1. move host vm

`rocks move host vm {host} {physhost} {file}`

Move a VM from its current physical node to another.

arguments

host

The name of the VM host to move.

physhost

The name of the physical host in which to move the VM.

file

The name of the file that stores the running VM's state.

examples

```
# rocks move host vm compute-0-0-0 vm-container-1-0
```

Move VM host compute-0-0-0 to physical host vm-container-1-0.

4.6. pause

4.6.1. pause host vm

```
rocks pause host vm {host...}
```

Pauses a VM slice on a physical node.

arguments

host

A list of one or more VM host names.

examples

```
# rocks pause host vm compute-0-0-0
```

Pause VM host compute-0-0-0.

4.7. remove

4.7.1. remove cluster

rocks remove cluster [cluster...]

Remove a virtual cluster.

arguments

[cluster]

One or more virtual frontend names.

examples

```
# rocks rmeove cluster frontend-0-0-0
```

Remove the cluster associated with the frontend named 'frontend-0-0'.

4.7.2. remove host vm

rocks remove host vm {host...}

Remove the configuration info in the database for the supplied hosts.

arguments

host

A list of one or more VM host names.

examples

```
# rocks remove host vm compute-0-0-0
```

Remove the configuration info in the database for compute-0-0-0.

4.7.3. remove host vm bootprofile

rocks remove host vm bootprofile {host...} [profile=*string*]

Remove a boot/install profile from a list of hosts.

arguments

host

List of hosts to remove profiles. If no hosts are listed, then the global definition that matches the 'profile=name' is removed. The global profile=install cannot be removed

parameters

[profile=*string*]

The label name for the profile to remove. You can see the profiles available for a host by executing: 'rocks list host vm profile <hostname>'.
'

examples

```
# rocks remove host vm bootprofile compute-0-0-0 profile=boot
```

Remove the 'boot' profile for compute-0-0-0.

4.8. report

4.8.1. report host vm

```
rocks report host vm {host} [install=bool]
```

Outputs the VM configuration file for a slice on a physical node.

arguments

host

One VM host name (e.g., compute-0-0-0).

parameters

[install=*bool*]

If install='y' is set, then the VM will be first boot from its install bootprofile. Default is 'n' VMs use different mechanisms to control booting as compared to PXE-booted hosts. However, If the pxeaction for a VM host is defined explicitly as "install*" for this VM, then this flag will be internally set to 'y'.

examples

```
$ rocks report host vm compute-0-0-0
```

Create the VM configuration file for host compute-0-0-0


```
$ rocks report host vm compute-0-0-0 install=y
```

Create the VM configuration file for host compute-0-0-0, and tell it to run its installprofile when it boots.

related commands

```
add host vm bootprofile
```

```
list host vm bootprofile
```

```
remove host vm bootprofile
```

```
set host vm boot
```

```
set host vm bootprofile
```

4.8.2. report host xen bridge

rocks report host xen bridge

Generates the Xen networking bridge configuration script for a host.

examples

```
$ rocks report host xen bridge
```

4.9. restore

4.9.1. restore host vm

```
rocks restore host vm {host...} {file}
```

Restore a VM on a physical node. This command restores a previously saved VM.

arguments

host

A list of one or more VM host names.

file

The file name the saved VM state is stored in. If you don't supply this parameter, then the default file name is: `/<largest-partition-on-physical-host>/xen/disks/<vm-name%gt;.saved`. For example, on a physical node with the default partitioning, the file that contains the state for VM `compute-0-0-0` is: `/state/partition1/xen/disks/compute-0-0-0.saved`

examples

```
# rocks restore host vm compute-0-0-0
```

Restore VM host `compute-0-0-0`.

4.10. resume

4.10.1. resume host vm

```
rocks resume host vm {host...}
```

Resume a paused VM slice on a physical node.

arguments

host

A list of one or more VM host names.

examples

```
# rocks resume host vm compute-0-0-0
```

Resume paused VM host `compute-0-0-0`.

4.11. save

4.11.1. save host vm

rocks save host vm {host...} {file}

Save a VM on a physical node. This command saves a currently running VM, then halts the VM. This saved state can be used to restart the VM with the command 'rocks restore host vm'.

arguments

host

A list of one or more VM host names.

file

The file name the saved VM state will be stored in. If you don't supply this parameter, then the default file name will be: /<largest-partition-on-physical-host>/xen/disks/<vm-name>.saved. For example, on a physical node with the default partitioning, the saved file for VM compute-0-0-0 will be named: /state/partition1/xen/disks/compute-0-0-0.saved

examples

```
# rocks save host vm compute-0-0-0
```

Save VM host compute-0-0-0.

4.12. set

4.12.1. set host vm

rocks set host vm

{host} [disk=*string*] [disksize=*string*] [mem=*string*] [physnode=*string*] [slice=*string*] [virt-type=*string*]

Change the VM configuration for a specific VM.

arguments

host

One or more VM host names.

parameters`[disk=string]`

A VM disk specification. More than one disk can be supplied. Each disk specification must be separated by a space.

`[disksize=string]`

The size of the VM disk.

`[mem=string]`

The amount of memory in megabytes to assign to this VM.

`[physnode=string]`

The physical machine this VM should run on.

`[slice=string]`

The slice ID for this VM.

`[virt-type=string]`

Set the virtualization type for this VM. This can be 'para' or 'hardware'.

examples

```
# rocks set host vm compute-0-0-0 mem=4096
```

Change the memory allocation for VM compute-0-0-0 to 4 GB.

4.12.2. set host vm boot

```
rocks set host vm boot {host...} [installprofile=string] [runprofile=string]
```

Set the install/run bootprofile for a vm host. This bootprofile defines what configuration is used to boot a vm in its running and/or installing configuration

arguments

host

One or more host names.

parameters`[installprofile=string]`

The label name for the bootprofile to use when installing a node. For a list of available bootprofiles, execute: 'rocks list host vm bootprofile <hostname>'. It is an error to set the install profile to None.

`[runprofile=string]`

The label name for the bootprofile to use when a node is running normally. Usually this is empty, and the kernel defined inside of the VM will be used for booting. For a list of bootprofiles, execute: `'rocks list host vm bootprofile <hostname>'`. Set this string to 'None' to clear the run profile

examples

```
# rocks set host vm boot compute-0-0-0 installprofile=install
```

Set the 'install' bootprofile for compute-0-0-0, when installing it.

```
# rocks set host vm boot compute-0-0-0 runprofile=None
```

clear the 'run' runprofile for compute-0-0-0, and use the kernel defined in the VM image,

4.12.3. set host vm bootprofile

```
rocks set host vm bootprofile [host...] [args=string] [kernel=string] [profile=string] [ramdisk=string]
```

Add a boot profile for virtual a host.

arguments

`[host]`

List of hosts to add profile definitions. If no hosts are listed, then the global definition for 'profile=name' is added.

parameters

`[args=string]`

The second line for a pxelinux definition (e.g., `append ks initrd=initrd.img ramdisk_size=150000 lang=devfs=nomount pxe kssendmac selinux=0`)

`[kernel=string]`

URL for the kernel associated with this profile (e.g., 'file:///boot/kickstart/xen/vmlinuz')

`[profile=string]`

Label name for the boot profile. You can see the profile label names by executing: `'rocks list host vm bootprofile [host(s)']`.

`[ramdisk=string]`

URL for the kernel associated with this profile (e.g., 'file:///boot/kickstart/xen/initrd-xen.iso.gz')

examples

```
# rocks set host vm bootprofile profile=install kernel="file:///boot/kickstart/xen/vmlinuz-experimental"
```

Change the kernel to vmlinuz-experimental for the global install profile

related commands

add host vm bootprofile

list host vm boot

list host vm bootprofile

remove host vm bootprofile

set host vm boot

4.13. start

4.13.1. start host vm

```
rocks start host vm {host...} [install=bool]
```

Boots a VM slice on a physical node.

arguments

host

A list of one or more VM host names.

parameters

[install=*bool*]

If install='y' is set, then the VM will be first boot from its install bootprofile. Default is 'n' VMs use different mechanisms to control booting as compared to PXE-booted hosts. However, If the pxeaction for a VM host is defined explicitly as "install*" for this VM, then this flag will be internally set to 'y'.

examples

```
# rocks start host vm compute-0-0-0
```

Start VM host compute-0-0-0.

```
# rocks start host vm install=y compute-0-0-0
```

Start VM host compute-0-0-0 in installation mode.

related commands

add host vm bootprofile

list host vm bootprofile

remove host vm bootprofile

set host vm boot

set host vm bootprofile

4.14. stop

4.14.1. stop host vm

```
rocks stop host vm {host...}
```

Destroy a VM slice on a physical node.

arguments

host

A list of one or more VM host names.

examples

```
# rocks stop host vm compute-0-0-0
```

Stop VM host compute-0-0-0. This is equivalent to a 'hard power off', (i.e., pulling the power cord from a node).

4.15. sync

4.15.1. sync host network xen

```
rocks sync host network xen
```

Reconfigure and restart the network for the named hosts.

examples

```
# rocks sync host network compute-0-0
```

Reconfigure and restart the network on compute-0-0.

Appendix A. Rocks Copyright

Rocks(r)
www.rocksclusters.org
version 5.1 (VI)

Copyright (c) 2000 - 2008 The Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice unmodified and in its entirety, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising and press materials, printed or electronic, mentioning features or use of this software must display the following acknowledgement:

"This product includes software developed by the Rocks(r)
Cluster Group at the San Diego Supercomputer Center at the
University of California, San Diego and its contributors."

4. Except as permitted for the purposes of acknowledgment in paragraph 3, neither the name or logo of this software nor the names of its authors may be used to endorse or promote products derived from this software without specific prior written permission. The name of the software includes the following terms, and any derivatives thereof: "Rocks", "Rocks Clusters", and "Avalanche Installer". For licensing of the associated name, interested parties should contact Technology Transfer & Intellectual Property Services, University of California, San Diego, 9500 Gilman Drive, Mail Code 0910, La Jolla, CA 92093-0910, Ph: (858) 534-5815, FAX: (858) 534-7345, E-MAIL:invent@ucsd.edu

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix B. Third Party Copyrights and Licenses

This section enumerates the licenses from all the third party software components of this Roll. A "best effort" attempt has been made to insure the complete and current licenses are listed. In the case of errors or omissions please contact the maintainer of this Roll. For more information on the licenses of any components please consult with the original author(s) or see the Rocks™ CVS repository¹.

B.1. xen

GNU General Public License

Most files in this repository are licensed under the terms of the GNU General Public License (GPL), a copy of which is attached at the end of this notice. Note that the only valid version of the GPL as far as the files in this repository are concerned is this particular version of the license (i.e., **only* v2*, not v2.2 or v3.x or whatever), unless explicitly otherwise stated.

Licensing Exceptions (the relaxed BSD-style license)

For the convenience of users and those who are porting OSes to run as Xen guests, certain files in this repository are not subject to the GPL when distributed separately or included in software packages outside this repository. Instead we specify a much more relaxed BSD-style license. Affected files include the Xen interface headers (xen/include/public/COPYING), and various drivers, support functions and header files within the Linux sparse source trees. In all such cases, license terms are stated at the top of the file or in a COPYING file in the same directory. Note that any file that is modified and then distributed within a Linux kernel is still subject to the GNU GPL.

-- Keir Fraser (on behalf of the Xen team)

=====

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains

a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent

access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR

REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Notes

1. <http://cvs.rocksclusters.org>