OpenNebula - Backlog #1290

Allowing autostart setting for KVM deployed VMs with persistent images, to restart them upon node reboot

05/23/2012 08:21 PM - Olivier Berger

Status:	Pending	**Start date:**	05/23/2012
Priority:	High	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Drivers - VM	**Estimated time:**	0.00 hour
Target version:			

Description

As discussed in http://lists.opennebula.org/pipermail/users-opennebula.org/2012-May/008959.html I think it would be great to offer some support for automatic restarting of VMs (libvirt autostart setting) images that would have persistent images, in case a node is rebooted (or in case of power outages and other restarts).

One main change involved is the need to define the domains, instead of transient domains.

See the discussion of proposed changes

History

#1 - 05/24/2012 06:48 PM - jordan pittier

Sounds great but, isn't this KVM specific ? Although I am using KVM, I like the way Opennebula tries to be as much "cross hypervisor" as possible.

#2 - 05/24/2012 08:55 PM - Olivier Berger

jordan pittier wrote:

> *Sounds great but, isn't this KVM specific ? Although I am using KVM, I like the way Opennebula tries to be as much "cross hypervisor" as possible.*

I don't think this is specific to KVM (although I mentioned KVM in the title of this ticket), as it is a a libvirt option (see http://libvirt.org/sources/virshcmdref/html/sect-autostart.html).

Still it may happen that libvirt only supports this for KVM. I haven't tested with others. But I couldn't imagine other hypervisors wouldn't have such a feature :-/

#3 - 05/31/2012 02:14 PM - jordan pittier

You are correct.

The this is, Opennebula uses libvirt only to manage KVM hosts.

#4 - 05/09/2013 01:13 PM - Ruben S. Montero

- Category set to Drivers - VM

#5 - 05/09/2013 03:14 PM - Ruben S. Montero

- Tracker changed from Feature to Request

#6 - 05/09/2013 09:11 PM - Ruben S. Montero

- Tracker changed from Request to Backlog

#7 - 05/09/2013 09:57 PM - Ruben S. Montero

- Status changed from New to Pending

#8 - 10/02/2013 12:07 PM - Daniel Dehennin

+1

It could be great to have a checkbox option in template definition and/or at VM instance time.

Some tests may be performed before enabling the “autostart” as I'm not sure it will work for non-persistent disk.

Thanks.

#9 - 10/10/2013 08:59 AM - Ruben S. Montero

- Priority changed from Normal to High

#10 - 11/07/2013 11:17 AM - Daniel Dehennin

Daniel Dehennin wrote:

> *+1*
>
> *It could be great to have a checkbox option in template definition and/or at VM instance time.*

We could also define a name prefix to use when ONE will **onetemplate instanciate** automatically.

> *Some tests may be performed before enabling the “autostart” as I'm not sure it will work for non-persistent disk.*

I thought a little to this issue as we need it and I wonder if it could not be implemented with actual features of ONE instead of using libvirt feature (for KVM).

In fact, I'm quite sure we should not use libvirt to manage this even for KVM VMs, on my own system, service dependencies is problematic between libvirt and OpenvSwitch.

Instead we could use the **suspend**, **stop** and **resume** mechanisms, making it working with non-persistent storage if I understood correctly theses commands.

We must face different use-cases, depending of what happen and if we are in single node or multi nodes.

For single node

1. on planned shutdown/reboot after an upgrade for example (init 0/init 6)
    1. when the ONE node is shutdown, we could just run **onevm suspend** all the running VM and put them in an **AUTOBOOT** state.
    2. when the ONE node is booted, run **onevm resume** on each VMs in **AUTOBOOT** state
2. on ONE node crash like hardware failure
    1. when the ONE node is booted
        1. run **onevm boot** on each VMs in **UNKNOWN** state using an auto-start enabled template

2. search for auto-start enabled templates and for each one: if no **RUNNING** VMs use it, run **onetemplate instanciate**

For multi-nodes

1. on planned shutdown/reboot after an upgrade for example (init 0/init 6)
   - if **system** datastore is shared, just live migrate all VMs to other nodes
   - if **system** datastore is local to the node, run **onevm stop** and **onevm resume** to perform a “cold migration”

I'm not sure about the best thing to do on hardware failure, in fact, as I'm missing test machines for now I don't even know what ONE does in such situation.

Regards.

#11 - 10/03/2014 09:45 PM - Daniel Dehennin

The ideal would be to have a trusted synchronous communication from the node to the frontend to report a shutdown/reboot.

My idea it to use the monitoring system with an init script on the node like the **libvirt-guest** one:

1. started last, stopped first
2. push some kind of shutdown monitoring
3. then the frontend run a node hook and apply a policy like the one I describe in my previous comment (single/multi nodes, with/without shared storage)
4. the script must wait for some feedback of the frontend

This will not work with pull based monitoring.

Another option is to hit the frontend directly by RPC but this requires authentication/authorization of nodes on the frontend.

Is there a way for nodes to advertise the frontend of a shutdown/reboot?

#12 - 10/07/2014 02:28 PM - Ruben S. Montero

> *Is there a way for nodes to advertise the frontend of a shutdown/reboot?*

In a shutdown/reboot cycle:

1. Long enough cycle the host should transit the error -> on state so a hook can be easily triggered
2. For quick reboots oned could not even notice the reboot, in that case we can add a probe in the monitor system. However the VMs will be moved to POWER-OFF. We can add a hook (on vm power off) to power on them if the host was rebooted. We need to add a probe for that.

So I'd suggest

1. Add a probe with uptime information
2. Write a hook for VM on POWER-OFF, if the VM went to power off just after the host booted restart it.

I really like this approach as it is hypervisor independent.

#13 - 10/07/2014 02:57 PM - EOLE Team

Ruben S. Montero wrote:

> *So I'd suggest   1. Add a probe with uptime information*
> *2. Write a hook for VM on POWER-OFF, if the VM went to power off just after the host booted restart it.*

Could we make a difference between VMs in **POWER-OFF** because of the reboot and VMs in **POWER-OFF** because user want them powered off?

I'm not sure we can blindly boot VMs after a reboot.

> *I really like this approach as it is hypervisor independent.*

Yes, me too, even if I personally only use KVM ;-)

Regards.

#14 - 10/09/2014 09:01 AM - Ruben S. Montero

EOLE Team wrote:

> *Ruben S. Montero wrote:*
>
> > *So I'd suggest 1. Add a probe with uptime information*
> > *2. Write a hook for VM on POWER-OFF, if the VM went to power off just after the host booted restart it.*
>
> *Could we make a difference between VMs in **POWER-OFF** because of the reboot and VMs in **POWER-OFF** because user want them powered off?*

Yes I think we can use the REASON field of history. Simply add a new reason for automatic transitions (vs user requested). This together with the uptime of the host should be enough...

Chhers

#15 - 08/12/2015 02:17 PM - Olivier Berger

FWIW, a discussion about this issue : https://forum.opennebula.org/t/automatically-restart-vms-after-host-restart/454

Any progress to expect ?

#16 - 08/12/2015 02:19 PM - Olivier Berger

Olivier Berger wrote:

> *As discussed in http://lists.opennebula.org/pipermail/users-opennebula.org/2012-May/008959.html ...*

Btw, the list archive is gone, but one can still find it on https://www.mail-archive.com/users%40lists.opennebula.org/msg06649.html

Hth

#17 - 08/13/2015 09:31 AM - Olivier Berger

Would it be possible to at least have KVM VMs created as non-transient, i.e. using virsh define + virsh start instead of just virsh create so that it is possible to manually perform a virsh autostart if needed (virsh autostart won't work on transient VMs), like in https://gist.github.com/anonymous/2776202, but without line 34 ?

I'm not sure there would be any side effects, and that would be a first improvement for KVM, until a more generic solution is found ?

#18 - 08/14/2015 03:00 PM - Ruben S. Montero

Yes the main reason for holding this back is the side effects for all the operations. I agree the idea would be to define+start and when the VM is removed from the host it needs to be undefined. We need to review all the operations to check when we need to do the undefine, e.g. poweroff, migrations etc... now it is assumed that the VM is not defined.

#19 - 11/17/2015 01:35 PM - EOLE Team

Ruben S. Montero wrote:

> *Yes I think we can use the REASON field of history. Simply add a new reason for automatic transitions (vs user requested). This together with the uptime of the host should be enough...*

Could we open a new issue for this point ?

This could solve the “host crashed” case:

- we set a **REASON** for any operation (automatic or user requested)
- in case of a host crash, the VMs will be reported as **POWEROFF** without any **REASON**.

Then, we should add the possiblity to have a **HOST_HOOK** executed when a host enter the **ON** state, in which case we list all VMs on that host in **POWEROFF** state without any **REASON** and **resume** them.

Regards.