OpenNebula - Bug #96
oned does not discern correctly the virsh running states.
03/30/2009 01:49 PM - Marlon Nerling

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
<th>Start date:</th>
<th>03/30/2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>High</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>Tino Vázquez</td>
<td>% Done:</td>
<td>0%</td>
</tr>
<tr>
<td>Category:</td>
<td>Drivers - Auth</td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td>Release 1.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution:</td>
<td>duplicate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Affected Versions:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

When a virtual machine goes down, by own or by a kvm bug the opennebula does not recognize correctly the returned state of virsh. /usr/lib/one/one_vmm_kvm.rb waits somehow the state to be shutdown, but virsh dominfo returns "shut off". I have tried to complement the 'case' fall on /usr/lib/one/one_vmm_kvm.rb by "shut off" and by "shut", but stills onevm gets lcm_state running and state active, although the machine(s) are down and virsh dominfo gets "shut off". I have no ruby knowhow to spot where is my fault, so there I need some of your help. The (try)patch is already posted as attachment of the issue #91: http://trac.opennebula.org/attachments/14/one_vmm_kvm.rb.patch I'm attaching my last (try)-patch of one_vmm_kvm.rb, which does not work, too.

Related issues:
Duplicates Bug # 91: onevm does not define the guest, deletes after shutdown

Associated revisions
Revision bdb76d43 - 05/12/2016 11:16 AM - melehin

Update kvm.rb (#96)

The LANG environment variable does not work as expected with virsh -c qemu:///system nodeinfo. But LC_ALL=C works fine.
Without LC_ALL=C monitoring fails with a “Error monitoring Host hostname (0): ./kvm.rb:49: undefined method ‘*’ for nil::NilClass (NoMethodError)”.

Server configuration:
OS: CentOS 6.2
OpenNebula: 4.14.2
Default language: ru_RU.UTF-8

Bug example:
LANG=C virsh -c qemu:///system nodeinfo
Модель процессора: x86_64
CPU: 16
Частота процессора: 1600 MHz
Сокеты: 1
Ядер на сокет: 4
Потоков на ядро: 2
Ячейки NUMA: 2
Объём памяти: 74237932 KiB

Correct behavior:
LC_ALL=C virsh -c qemu:///system nodeinfo
CPU model: x86_64
CPU: 16
CPU frequency: 1600 MHz

05/10/2021
Hi

Thanks for the feedback. There is no problem with the ruby, it is probably the libvirt to OpenNebula state mapping. The driver may notify (as a result of a monitor action, as the case you mention) the following states:

- `a` (active) The machine is still active
- `p` (paused) The VM was suspended (e.g. manually bypassing OpenNebula)
- `e` (error) There monitoring reported an error with the VM
- `d` (deleted) The VM was shut down (e.g. by its own or bypassing OpenNebula)

So probably if you discover that the VM you are monitoring is in shut down probably we need to report 'd', something like.

```
when "shut down"
  state = "d"
```

Could you verify that this works in with your problem

THANKS FOR YOUR FEEDBACK!

---

#2 - 03/30/2009 02:45 PM - Marlon Nerling

Hey Ruben

I don't think that when a VM is down it should be reported as deleted! or that it bypassed open nebula ! I strongly oppose it actually.

Open nebula should (and I by myself am astonished one have none 'start' parameter/option since virsh gives this possibility!)

Try to see my (l)users .. they shutdown their machines 'bypassing' the 'Programmed' Interface (UI). It is the nature-of-the-game.

They 'should' have saved the state of the machine throught the UI (I ported opennebula to a php User-interface ), but they 'can' shutdown the VM, too.

All I need, when I program my UI, is to know the REAL STATE of the machine, I don't need it to be deleted, moved (or touched, please don't touch my machines :) ) (Actually opennebula does sometimes too much, by my mean). All I need is to trust opennebula it one knows the state of the VMs, (if not just-in-time, at best in the minute).

And then I (I) WILL handle it, I (I) WILL give a option to start the VM (And I DON'T WILL (Yes I could, if I only would) ask virsh(libvirt) remotely, since I trusted open nebula by every step until this simple function).

I never thought it was ruby's issue, the issue is that I actually don't know the ruby language well enough to patch it by myself.

And so I go on, I WILL also help you, by mean of testing and debugging opennebula and I thank you very much for the attention, really.

---

#3 - 03/30/2009 02:54 PM - Ruben S. Montero

Hi,
Yes we feel the same about this one. We are working in this direction, please check #75 does it make sense?

Cheers

#4 - 03/30/2009 03:10 PM - Marlon Nerling

Si, the issue #75 is part of the 'Whole Problem', please see http://dev.opennebula.org/issues/91 ,too. This would resolve the issue #75 !
Open nebula sees, i.E perfectly if a machine is resumed by virsh (although it will very dangerous if it assumes it can to resume the VM, since we don't know who the save image lies ). This is a really nice feature.
Now: it should be the same (and really many simplest) to do it when the machine ist down. All we need is a new onevm option names 'start' in /usr/lib/one/ruby/one.rb and a new action_start (calling virsh start remotely). And o'course we MUST to get the RIGHT vm state from virsh!

Do you see what I see? I see it as a bug, so I hope we can to backport it into one1.2, if you see it as a 'new' feature, then we can flame each other ;)
(OLE!)

Hasta la vitoria!

Marlon

#5 - 03/30/2009 03:58 PM - Marlon Nerling

Little UPDATE:
by onevm shutdown it tries to delete the (and unfortunately do it) VM machine! It it not a spected behaviour (My mean).

#6 - 04/05/2009 11:42 PM - Ruben S. Montero

- Target version changed from Release 1.2 to Release 1.2.1

#7 - 04/10/2009 04:12 PM - Ruben S. Montero

I think I am getting a little confused. Let me try to summarized:

- It can be useful to define a domain and the boot it using start. In this way we can always get the state of a VM because it is defined within the libvirt system.
- A user can always bypass the management system (e.g. by logging in the remote server and play with the hypervisor virsh, xen or whatever...). OpenNebula must be able to recover from this (at least mark this VM as in an error state). Here I do not see how define/start can help us, because the user can always issue a virsh undefine command and the situation would be the same.
- In my opinion we can not leave all the VMs defined in the livirt system. We could have some scalability issues here, and we can not relay in it (What would happen if libvirtd crashes?)
- Shutdown, you suggest that shutdown should stop the VM but leave it defined in the remote server, don't you?. If we assume that the user can always undefine a domain I do not see the benefits here. BTW the VM
  is not deleted in the DB it is marked as DONE and it is not shown in onevm list, but it is there for accounting purposes.

Finally, I'd love to hear more about yout UI. Can it be downloaded, tested?

Cheers!

Ruben

#8 - 04/10/2009 05:30 PM - Marlon Nerling

05/10/2021
redmine@opennebula.org wrote:

Issue #96 has been updated by Ruben S. Montero.

I think I am getting a little confused. Let me try to summarized:

- It can be useful to [define] a domain and the boot it using [start]. In this way we can always get the state of a VM because it is defined within the libvirt system.

I think so.

- A user can always bypass the management system (e.g. by logging in the remote server and play with the hypervisor virsh, xen or whatever...). OpenNebula must be able to recover from this (at least mark this VM as in an error state). Here I do not see how [define/start] can help us, because the user can always issue a [virsh undefine] command and the situation would be the same.

The user should NOT be able to access virsh on the hosts! But the shutdown button is in every Window-Management! So what help us here is that when the user shuts the VM down, 'virsh dominfo' returns the very status of the machine (if defined), elsewhere it give a 'domain not defined' error. Do you see it?

- In my opinion we can not leave all the VMs defined in the libvirt system. We could have some scalability issues here, and we can not relay in it (What would happen if libvirtd crashes?)

I think the same, my patch does not handle the deletion of the VMs!.. it does not undefines the machine on delete. I also think we should work it on.

- Shutdown, you suggest that shutdown should stop the VM but leave it defined in the remote server, don't you?. If we assume that the user can always [undefine] a domain I do not see the benefits here. BTW the VM is not deleted in the DB it is marked as DONE and it is not shown in onevm list, but it is there for accounting purposes.

As I said above, the user should NOT have the possibility to undefine a VM.

By my mean we should only reuse the functionality of the VMMs (in this case libvirt). Libvirt handles very well the shutdown of the machines (Well, at most of the time, because without acpi it actually cannot shutdown a VM! Maybe we could use 'virsh destroy'!).

But really important is to use the libvirt status of the machines. If it said 'shut off', then all we need to do - to reuse this VM - is 'virsh start' - and the VM is full functional.

I think the problem should now be clear, about solution now?? .

Finally, I'd love to hear more about your UI. Can it be downloaded, tested?

The UI is actually very simple, the most interesting to you would be the PHP-classes who abstracts onevm (and as long as one buggy is, libvirt too).

I must ask my Employers for you to see it, but I don't see any problem, we advocate and support strongly Open Sourcing. Please give me time to the next Week (14.04.09) and then I will have an Answer.

Cheers!

05/10/2021
Buena Semana Santa.
Best Regards
Marlon

#9 - 04/10/2009 06:50 PM - Ruben S. Montero

The user should NOT be able to access virsh on the hosts! But the shutdown button is in every Window-Management! So what help us here is that when the user shuts the VM down, ‘virsh dominfo’ returns the very status of the machine (if defined), elsewhere it give a ‘domain not defined ’ error. Do you see it?

OK now I got it. The problem is that the user can shutdown the VM from the inside. I thought the problem were users issuing virsh shutdown

I think the same, my patch does not handle the deletion of the VMs!.. it does not undefines the machine on delete. I also think we should work it on.

OK let me look closer at this. I want to test a couple of things...

But really important is to use the libvirt status of the machines. If it said 'shut off', then all we need to do - to reuse this VM - is ‘virsh start’ - and the VM is full functional.

I think the problem should now be clear, about solution now?? .

Yes, but I am not convinced of using the start command from libvirt. As OpenNebula only implements create/shutdown. I think the behavior is coherent. This is: 1. The user logs in the VM.
2. The user shutdown the VM (from the inside)
3. OpenNebula no longer finds the VM (the user has actually shutdown it)
4. OpenNebula marks the VM as DONE (or ERROR as suggested in #75)
5. If the user wants the VM again she can use onevm create (or if the VM was left in ERROR, we can implement a re-submit command for onevm).

Note that we have a pool of servers, so if the user wants to start the VM again we should let the scheduler allocate the VM (just imagine that there is no capacity left in the server where the VM was running). So we may end up starting the VM in a server where the VM has not been defined previously.

So if the user shutdowns the VM, and we put that VM in a ERROR state, then the user can delete the VM resubmit it with the same description file. Would not that solve your issue?

Buena Semana Santa.
Best Regards
Marlon

Gracias!!!
Ruben

#10 - 04/16/2009 03:00 PM - Marlon Nerling

Ola Rubens.
It would work only if the user could resubmit the machine whit the same onevm id!
It would be actually nice if by shutdown oned holds the machine back to the master and, when by ‘re-submit’ it submits the vm says from
/var/lib/one/<onevm id>/deployement.1 to some other machine.
This would resolve the question, more or less!
But I think that:
1. VMs should not be marked as failed! I shutdown my own machine every night to save some energy! ;)
2. oned must to handle right when the machine is shutdown.

See my landscape:
I have a pool of hosts, there my guests.
If I use my ‘virsh define’ patch, wenn the machine goes down, virsh stays knowing all about it (one could it too! ).
Now I have a VM, who exists, is consistent, the disks are placed, the Host may have enought resources (since one known that the machine is there!! ).
I cannot start it without a workaround through "ssh virsh", but actually I have no chance to know it is down ( only when I (my users) try to connect to the
machine ), because oned does not accept the fact that the machine is down (Go away.. You are a Phantom, you don't exist! ) and answer saying it is
ACTIVE and RUNNING!!
If I dont use 'virsh define', next time oned asks about the guest - after a shutdown (think about the shutdown of the host!!) - virsh on the host answers
like : 'I dont know nothing about the VM are you asking for, go away', oned deletes the machine and marks it as failed.
So or so I have an Impasse .. I can not use a machine after shutdown, although it is would work (maybe with mission critical Documents stored on it!!
aaaa ;) ).

Yes .. I see solutions coming of our interloquio.

#11 - 04/16/2009 04:09 PM - Ruben S. Montero

Marlon Nerling wrote:

    Ola Rubens.
    It would work only if the user could resubmit the machine whit the same onevm id!
    It would be actually nice if by shutdown oned holds the machine back to the master and, when by 're-submit' it submits the vm says from
    /var/lib/one/<onevm id>/deployement.1 to some other machine.

that is exactly stop/resume!!!. This is if you do onevm stop, and then onevm resume you get the above behavior (including deployment.1)

    This would resolve the question, more or less!
    But I think that:
    1. VMs should not be marked as failed! I shutdown my own machine every night to save some energy! ;)

OK then stop it ;)

    2. oned must to handle right when the machine is shutdown.

oned should handle ok the stopped VMs

    See my landscape:
    I have a pool of hosts, there my guests.
    If I use my 'virsh define' patch, wenn the machine goes down, virsh stays knowing all about it (one could it too! ).
    Now I have a VM, who exists, is consistent, the disks are placed, the Host may have enought resources (since one known that the machine is

05/10/2021
there!!). I cannot start it without a workaround through "ssh virsh", but actually I have no chance to know it is down (only when I (my users) try to connect to the machine!), because oned does not accept the fact that the machine is down (Go away.. You are a Phantom, you don't exist!) and answer saying it is ACTIVE and RUNNING!!

If I dont use 'virsh define', next time oned asks about the guest - after a shutdown (think about the shutdown of the host!!) - virsh on the host answers like: 'I dont know nothing about the VM are you asking for, go away', oned deletes the machine and marks it as failed.

So or so I have an Impasse .. I can not use a machine after shutdown, although it is would work (maybe with mission critical Documents stored on it!! aaaaa ;)).

Yes .. I see solutions coming of our interloquio.

So if you want to shutdown a VM to later use it you can use stop/resume it (and the state of the VM is also preserved).

However, if the VM is supose to be running and is not in the hypervisor list, then we have a problem because that could be caused by:

- a crash in the libvirt daemon
- someone (maybe acidentally) shutdown the VM by interacting with the hypervisor
- someone (the VM owner) shutdown the VM from the inside (e.g. init 6)

In either case I believe that the VM passed away, and if you want it again then you have to resubmit it. I agree with you that this should be handle better, and not just silently removing the VM form the list. And there the virsh define can help us to better deal with it.

Did we arrive to somewhere?

#12 - 04/17/2009 10:03 AM - Marlon Nerling

that is exactly stop/resume!!!. This is if you do onevm stop, and then onevm resume you get the above behavior (including deployment.1)

Yes! This ist exactly stop/resume, but I cannot resume a deleted VM! You see .. now if the VM is shut down, one deletes it!

OK then stop it ;)

I will stop it! I have a option for the user to do it .. with onevm stop and onevm start. But the user is dumb (This is the Tao of Unix)!

oned should handle ok the stopped VMs

I think it should, too. By my mean it does not. It should not delete the machine by shutdown, this is clear to me.

So if you want to shutdown a VM to later use it you can use stop/resume it (and the state of the VM is also preserved).

Yes, a very nice behaviour!

However, if the VM is supose to be running and is not in the hypervisor list, then we have a problem because that could be caused by:

- a crash in the libvirt daemon
- someone (maybe accidentally) shutdown the VM by interacting with the hypervisor
- someone (the VM owner) shutdown the VM from the inside (e.g. init 6)
If you 'define' the machine after creating it, the hypervisor will know about it forever, after reboot of the host, too.

In either case I believe that the VM passed away, and if you want it again then you have to resubmit it. I agree with you that this should be handle better, >and not just silently removing the VM form the list. And there the virsh define can help us to better deal with it.

Yes, I can to live with it, but without delete!

Did we arrive to somewhere?

I think so ..

1. virsh define by create. virsh undefine by delete. = We have one mean.
2. hold back the VM after shut down. = We have one mean.
3. Delete the VM lease. = We have different means.
4. onevm resume should handle stoped machines, too. = This is a good idea.
5. Stop the damn VMs instead of shutdown. = We have one mean. (Will you say it to my [L]users, please ;)

I see no more than 3 Man-Hours of programing and testing.

Hasta mÃ¡s
Marlon

#13 - 04/30/2009 04:39 PM - Ruben S. Montero
- Status changed from New to Closed
- Resolution set to duplicate

Files
one_vmm_kvm.rb.patch 568 Bytes 03/30/2009 Marlon Nerling