

Foreman - Bug #13507

Creating VMs with hammer requires too many parameters

02/01/2016 01:01 PM - Guido Günther

Status: Resolved	
Priority: Normal	
Assignee:	
Category: Host creation	
Target version:	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request:	
Description	
<p>Hi</p> <p>while hammer is awesome for command line usage one has to be a bit verbose to get a VM going even if one is using host groups and compute profiles:</p> <p>Let's say I have a hostgroup with a compute profile selected. That compute profile has two compute resources:</p> <p>If have a host group (deriving from another host group that has the compute profile association). This has a single compute profile and two compute resources so all should be set (except for maybe which of the two compute resources to use but that's not what happens):</p> <pre>hammer host create --name=test.example.com --hostgroup-id=1 --location-id 1 Could not create the host: Error: At least one interface must be set as primary</pre> <p>Both compute profiles in the hostgroup only have a single interface but lets help a little:</p> <pre>hammer host create --name=test.example.com --interface=primary=true,provision=true --hostgroup-id=1 --location-id 1 Could not create the host: Mac can't be blank Partition Table can't be blank unless a custom partition has been defined Medium can't be blank</pre> <p>Both compute profiles in the host group have an image specified in the default vm attributes but let's be explicit: and add a compute resource:</p> <pre>hammer host create --name=test.example.com --interface=primary=true,provision=true --hostgroup-id=1 --location-id 1 --compute-resource-id=1 Could not create the host: Partition Table can't be blank unless a custom partition has been defined Medium can't be blank</pre> <p>O.k. then add the compute-profile as well:</p> <pre>hammer host create --name=test.example.com --interface=primary=true,provision=true --hostgroup-id=1 --location-id 1 --compute-resource-id=1 --compute-profile-id=1 Could not create the host: Partition Table can't be blank unless a custom partition has been defined Medium can't be blank</pre> <p>still not derived from there so let's be explicit:</p> <pre>hammer host create --name=test.example.com --interface=primary=true,provision=true --hostgroup-id=1 --location-id 1 --compute-resource-id=1 --compute-profile-id=1 --provision-method=image Could not create the host:</pre>	

Ip can't be blank

So we're at the network step now (no idea it would pick the *right* image though), let's help again:

```
hammer host create --name=test.example.com --interface=primary=true,provision=true --hostgroup-id=1 --location-id 1 --compute-resource-id=1 --compute-profile-id=1 --provision-method=image --subnet-id=1
could not create the host:
Failed to create a compute MyComRes (VMware) instance test.example.com: cluster is required for this operation
```

The cluster is stored in the compute profile but again:

```
hammer host create --name=test.example.com --interface=primary=true,provision=true --hostgroup-id=1 --location-id 1 --compute-resource-id=1 --compute-profile-id=1 --provision-method=image --subnet-id=1 --compute-attributes="cluster=foo"
Could not create the host:
Failed to create a compute MyComRes (VMware) instance test.example.com: failed to create vm: InvalidDatastorePath: Ungültiger Datenspeicherpfad '[datastore1]'.
```

There is no datastore1, I could give this as well but I give up here since maintaining this list of details twice (in the hammer script and in the database) is too error prone (and giving the datastore gives more errors due to the disk not being fully specified).

Interestingly, the GUI gets it mostly right (new host->select location, select host group, select deploy on (which is always bare metal by default): it then gets the right compute profile and assigns the correct interface as primary/provision by itself; Head to "Operating System" Tab, Select provision method (defaults to network), select image (not taken from compute profile). So in hammer cli terms this would be:

```
hammer host create --name=test.example.com --location-id=1 --hostgroup-id=1 --compute-resource-id=1 --provision-method=image --image=myimage
```

and that would be o.k. to put into a script.

and if it would also pick the right image from the defined compute profile:

```
hammer host create --name=test.example.com --location-id=1 --hostgroup-id=1 --compute-resource-id=1
```

which would be totally awesome. Tested with hammer-cli-foreman 0.4 and 0.5. I'm happy to help fix this if somebody points me into the right direction.

Related issues:

Related to Foreman - Bug #6342: Compute profile should be used when empty vol...	Closed	06/23/2014
Related to Foreman - Bug #13281: vsphere: Fails to use DistributedVirtualPort...	New	01/19/2016
Related to Foreman - Bug #13984: When using image-based provisioning via hamm...	Closed	03/01/2016

History

#1 - 02/02/2016 06:54 AM - Dominic Cleal

- Category set to Host creation

It appears there are two distinct, but similar issues. One is that the inheritance of attributes from the host group is failing, the second is inheritance of the compute profile attributes.

The compute profile attributes is likely to be caused by [#6342](#), which is unresolved but has a PR open. Hammer sends a near-empty set of compute attributes which stops the profile attributes from being used.

The host group attributes issue I can't reproduce - is this on Foreman 1.10.1?

#2 - 02/03/2016 12:56 PM - Guido Günther

Dominic Cleal wrote:

It appears there are two distinct, but similar issues. One is that the inheritance of attributes from the host group is failing, the second is inheritance of the compute profile attributes.

The compute profile attributes is likely to be caused by [#6342](#), which is unresolved but has a PR open. Hammer sends a near-empty set of compute attributes which stops the profile attributes from being used.

The host group attributes issue I can't reproduce - is this on Foreman 1.10.1?

Updating from 1.10.0 to 1.10.1 changed the error slightly to "invalid MAC". I switched to using JSON POSTs directly to get more control over what gets posted with (excuse the python syntax):

```
data = {
  "host": {
    "name": "testhost",
    "provision_method": 'image',
    'build': True,
    'enabled': True,
    'managed': True,
    'subnet_id': 1,
    "hostgroup_id": 1,
    "location_id": 1,
    'compute_profile_id': 1,
    'compute_resource_id': 1,
    'architecture_id': 1,
    'operatingsystem_id': 2,
    'image_id': 1,
  },
  "interfaces": {
    "0": {
      "primary": True,
      "provision": True,
    }
  }
}
```

I do get a VM but that doesn't start (and therefore doesn't get provisioned). This is due to:

```
"compute_attributes": {
  "start": '1',
}
```

lacking in the JSON (should this be added by default, it defaults to PowerOn in the GUI). If I add this

it fails like:

```
[app] [W] Failed to create a compute ComputeRes (VMware) instance testhost.example.com: undefined method `include?' for nil:NilClass
|
| NoMethodError: undefined method `include?' for nil:NilClass
| /usr/share/foreman/vendor/ruby/2.1.0/gems/fog-1.34.0/lib/fog/vsphere/requests/compute/get_virtual_machine.rb:34:in `get_vm_by_name'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/fog-1.34.0/lib/fog/vsphere/requests/compute/get_virtual_machine.rb:25:in `block in get_vm_ref'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/fog-1.34.0/lib/fog/vsphere/requests/compute/get_virtual_machine.rb:25:in `map'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/fog-1.34.0/lib/fog/vsphere/requests/compute/get_virtual_machine.rb:25:in `get_vm_ref'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/fog-1.34.0/lib/fog/vsphere/requests/compute/list_vm_interfaces.rb:32:in `list_vm_interfaces'
| /usr/share/foreman/app/models/compute_resources/foreman/model/vmware.rb:359:in `clone_vm'
| /usr/share/foreman/app/models/compute_resources/foreman/model/vmware.rb:324:in `create_vm'
| /usr/share/foreman/app/models/concerns/orchestration/compute.rb:76:in `setCompute'
| /usr/share/foreman/app/models/concerns/orchestration.rb:168:in `execute'
| /usr/share/foreman/app/models/concerns/orchestration.rb:113:in `block in process'
| /usr/share/foreman/app/models/concerns/orchestration.rb:105:in `each'
| /usr/share/foreman/app/models/concerns/orchestration.rb:105:in `process'
| /usr/share/foreman/app/models/concerns/orchestration.rb:35:in `on_save'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/activerecord-3.2.21/lib/active_support/callbacks.rb:671:in `_run__1680473319727549563_save__1586870868225049732_callbacks'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/activerecord-3.2.21/lib/active_support/callbacks.rb:405:in `__run__callback'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/activerecord-3.2.21/lib/active_support/callbacks.rb:385:in `_run__save_callbacks'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/activerecord-3.2.21/lib/active_support/callbacks.rb:81:in `run_callbacks'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/activerecord-3.2.21/lib/active_record/callbacks.rb:264:in `create_or_update'
| /usr/share/foreman/vendor/ruby/2.1.0/gems/activerecord-3.2.21/lib/active_record/persistence.rb:84:in `save'
```

```
| /usr/share/foreman/vendor/ruby/2.1.0/gems/activerecord-3.2.21/lib/active_record/validations.rb:50:in `save'
```

Don't get confused about the line numbers, I have to apply the patch from [#13281](#) to get Foreman to build cloned VMs at all. Without that patch it would fail a couple of lines later in `clone_vm` since the `raw_args` hash is mostly empty.

If I then add the patch from [#6342](#) I can at least create VMs with the above JSON that start and then get provisioned via SSH since this one fixes copying of `compute_attributes`.

Why inheritance of hostgroup attributes doesn't work I haven't figured out that.

So it would be great if the PR from [#6342](#) would make it into 1.10.2 to get image provisioned VMware going a bit better in 1.10.x.

#3 - 02/03/2016 12:56 PM - Guido Günther

- Related to Bug [#6342](#): Compute profile should be used when empty volumes and interfaces passed in `compute_attributes` added

#4 - 02/03/2016 12:57 PM - Guido Günther

- Related to Bug [#13281](#): vsphere: Fails to use `DistributedVirtualPortGroup` added

#5 - 03/01/2016 11:01 AM - Ivan Necas

I was able to reproduce the 'Ip can't be blank' issue when using the `provision-method=image`. I've filed separate issue for this <http://projects.theforeman.org/issues/13984>. Once resolved I think this issue should be closed as resolved as well.

#6 - 03/01/2016 11:02 AM - Ivan Necas

- Related to Bug [#13984](#): When using image-based provisioning via hammer, the subnet is not inherited from the hostgroup added

#7 - 06/25/2016 07:08 AM - Guido Günther

- Status changed from *New* to *Closed*

I think this can be closed since we have tracking bugs for the detailed issues (and most of them already fixed). Thanks!

#8 - 06/27/2016 03:08 AM - Dominic Cleal

- Status changed from *Closed* to *Resolved*