Hello,

Kudos for this awesome plugin, ran into one bug while testing it.

Running Foreman 1.7 (upgrade from 1.6), errors occurred in both 1.6 and 1.7 and using ruby193-rubygem-foreman_digitalocean-0.0.2-1.el6.noarch

When associating VMs, it will error with:

ActiveRecord::StatementInvalid
PGError: ERROR: operator does not exist: character varying = integer
LINE 1: ...sts"."compute_resource_id" = 1 AND "hosts"."uuid" IN (335119... ^ HINT: No operator matches the given name and argument type(s). You might need to add explicit type casts. :
SELECT COUNT FROM "hosts" WHERE "hosts"."type" IN ('Host::Managed') AND "hosts"."compute_resource_id" = 1 AND "hosts"."uuid" IN (3351196)
app/controllers/compute_resources_controller.rb:46:in `block in associate'
app/controllers/compute_resources_controller.rb:45:in `associate'
app/models/concerns/foreman/thread_session.rb:33:in `clear_thread'
lib/middleware/catch_json_parse_errors.rb:9:in `call'

Associated revisions
Revision fd875a19 - 12/10/2014 04:46 AM - Tommy McNeely
  fixes #8555 - uuid issues during create or associate

History
#1 - 12/03/2014 05:04 AM - Dominic Cleal
  If I'm reading that correctly, it looks like the DO "UUIDs" being retrieved are actually integers, not UUID strings?

#2 - 12/08/2014 12:55 PM - Tommy McNeely
  - Assignee set to Tommy McNeely
  - Difficulty set to easy

This also happens during host creation. The problem as noted seems to be that "uuid" is being returned as an int, and it needs to be a string. While I believe that is actually a problem in "core" code, we can work around it in the module.

#3 - 12/08/2014 10:46 PM - Tommy McNeely
  Note that I tried the following based on a suggestion in the #foreman-dev channel

[vagrant@devforeman digitalocean]$ git diff
diff --git a/app/models/concerns/fog_extensions/digitalocean/server.rb b/app/models/concerns/fog_extensions/digitalocean/server.rb
index 5050637..712e2bb 100644
However, now it fails immediately on host creation, rather than after the VM is created and ready. It gives the error:

```
creating VM with the following options: {"name"=>"testvm06.test.example.com", "flavor_id"=>"66", "region_id"=>"4", "image_id"=>"7053293"}
Unhandled DigitalOcean error: Fog::Errors::Error:Resaving an existing object may create a duplicate
/home/vagrant/.gem/ruby/1.9.1/gems/fog-1.25.0/lib/fog/digitalocean/models/compute/server.rb:118:in `save'
/home/vagrant/.gem/ruby/1.9.1/gems/fog-core-1.25.0/lib/core/collection.rb:50:in `create'
/home/vagrant/foreman/app/models/compute_resource.rb:147:in `create_vm'
/home/vagrant/foreman-digitalocean/app/models/foreman_digitalocean/digitalocean.rb:35:in `create_vm'
/home/vagrant/foreman/app/models/concerns/orchestration/compute.rb:74:in `setCompute'
```

#4 - 12/09/2014 09:04 AM - Dominic Cleal

Maybe instead of redefining 'identity' you could add a new method which we'll use for retrieving the UUID.

```
def identity_to_s
  identity.to_s
end
```

And then change DO's provided_attributes method (https://github.com/theforeman/foreman-digitalocean/blob/master/app/models/foreman_digitalocean/digitalocean.rb#L16) to:

```
super.merge({ :uuid => :identity_to_s, :ip => :public_ip_address })
```
That looks like it might work. I have tested it in my development instance, but not on "postgres" yet... However, it looks good form the logs:

```sql
SELECT * FROM "hosts" WHERE "hosts"."uuid" = '3426805' LIMIT 1
```

Note the single quotes around the uuid number...

Once I figure out a "sane" way to test it against postgres I will submit the PR.

~tommy

---

#6 - 12/10/2014 05:02 AM - Tommy McNeely

- Status changed from New to Closed
- % Done changed from 0 to 100

Applied in changeset foreman-digitalocean|fd875a19fb45f9df1277bca9d2de9bf71eed6636