Foreman - Bug #4408

cannot provision VM using API or UI

02/20/2014 09:53 PM - Greg Petras

Status:	Resolved		
Priority:	Normal		
Assignee:			
Category:	API		
Target version:			
Difficulty:		**Fixed in Releases:**	
Triaged:		**Found in Releases:**	
Bugzilla link:		**Red Hat JIRA:**	
Pull request:			

Description

After an upgrade to Foreman 1.4, I cannot provision a VM using the API or the UI.

Previously, we would POST something like this to /api/hosts:

```
{
"subnet_id" : null,
"compute_attributes" : {
"interfaces_attributes" : {
"0" : {
"network" : "<network>",
"_delete" : ""
}
},
"cluster" : "<cluster>",
"volumes_attributes" : {
"1" : {
"thin" : "0",
"datastore" : "<datastore>",
"name" : "Hard disk 1",
"size_gb" : 30
}
},
"power_action" : "start",
"memory_mb" : 2048,
"path" : "<path>",
"cpus" : 1,
"start" : "1"
},
"hostgroup_id" : "81",
"environment_id" : 40,
"provision_method" : "build",
"ptable_id" : 11,
"operatingsystem_id" : null,
"domain_id" : null,
"overwrite" : "false",
"enabled" : "1",
"name" : "<fqdn>",
"build" : "1",
"architecture_id" : null,
"medium_id" : null,
"managed" : "true",
"compute_resource_id" : 2,
"puppet_proxy_id" : null
}
```

Now I am trying this:

```
{
"name" : "<fqdn>",
"build" : "1",
"hostgroup_id" : "81",
"managed" : "true",
"compute_profile_id" : 1,
"environment_id" : 40,
"provision_method" : "build",
"ptable_id" : 11,
"compute_resource_id" : 2,
"overwrite" : "false",
"enabled" : "1"
}
```

The error I get back from Foreman is:

```
{
"errors" : {
"mac" : [
"has already been taken",
"is invalid",
"can't be blank"
]
},
"full_messages" : [
"MAC address has already been taken",
"MAC address is invalid",
"MAC address can't be blank"
],
"id" : null
}
```

I don't see any output in the Foreman proxy logs. This is what I see in the Foreman log:

```
Started POST "/api/hosts" for 10.123.0.31 at 2014-02-20 21:44:17 +0000
Processing by Api::V1::HostsController#create as JSON
Parameters: {"host"=>{"name"=>"<fqdn>", "build"=>"1", "hostgroup_id"=>"81", "managed"=>"true", "compute_profile_id"=>1,
"environment_id"=>40, "provision_method"=>"build", "ptable_id"=>11, "compute_resource_id"=>2, "overwrite"=>"false",
"enabled"=>"1"}, "apiv"=>"v1"}
Authorized user greg.petras(Greg Petras)
Unprocessable entity Host::Managed (id: new):
MAC address has already been taken
MAC address is invalid
MAC address can't be blank

Rendered api/v1/errors/unprocessable_entity.json.rabl (2.7ms)
Completed 422 Unprocessable Entity in 2402ms (Views: 3.8ms | ActiveRecord: 32.7ms)
```

I've also seen the same behavior when trying to POST to /api/v2/hosts.

Related issues:

Related to Foreman - Feature #4250: Expose compute profiles via API	**Closed**	**02/04/2014**
Related to Foreman - Feature #4581: Implement available_networks API for VMware	**Closed**	**03/06/2014**

History

#1 - 02/21/2014 03:41 PM - Greg Petras

This issue is the same under 1.4.1.

#2 - 03/04/2014 09:00 PM - Andrew Cooper

Greg Petras wrote:

> This issue is the same under 1.4.1.

We are also affected by this issue. This did work successfully in 1.3.x, but is no longer working in 1.4.0 or 1.4.1. We would be more than happy to test any code changes if that would help speed up the lead time for a fix!

-Andrew

#3 - 03/05/2014 05:33 PM - Dominic Cleal

- Related to Feature #4250: Expose compute profiles via API added

#4 - 03/05/2014 05:36 PM - Dominic Cleal

Using compute_profile_id won't work at the moment from what I can tell, since the attributes get filled in from the compute profile **after** validation. The validation though (MAC addresses) is failing because it needs to know during validation that the attributes are present.

It's not clear from either the description or Andrew's response whether you're only trying to use compute_profile_id or if you're using the original API call unsuccessfully too. If so, please supply the corresponding logs (the ones supplied show compute_profile_id only).

#5 - 03/05/2014 10:14 PM - Greg Petras

- File production.log added

Dominic -

Thanks for the response. I tried to use the original API call against 1.4.0, and it fails. Here's the curl command:

```
# curl -H "Content-Type: application/json" -u "$FOREMAN_USERNAME:$FOREMAN_PASSWORD" -d @host.json http://forem
an.example.com/api/hosts
```

Here's an example of what we currently POST to our Foreman 1.3.0 system to create a host:

```
{
  "host": {
    "subnet_id": null,
    "name": "<fqdn>",
    "build": "1",
    "compute_attributes": {
      "interfaces_attributes": {
        "0": {
          "network": "<network>",
          "_delete": ""
        }
      },
      "cluster": "<cluster>",
      "volumes_attributes": {
        "1": {
          "thin": "0",
          "datastore": ">datastore>",
          "name": "Hard disk 1",
          "size_gb": 20
        }
      },
      "power_action": "start",
      "memory_mb": 2048,
      "path": "<path>",
      "cpus": 1,
      "start": "1"
    },
    "hostgroup_id": "81",
    "medium_id": null,
    "managed": "true",
    "environment_id": 40,
    "provision_method": "build",
    "ptable_id": 11,
    "compute_resource_id": 2,
    "overwrite": "false",
    "enabled": "1"
  }
```

After upgrading to Foreman 1.4.x (tried both 1.4.0 and 1.4.1), we get the following error when we POST the same content:

```
{
  "host": {
    "id": null,
    "errors": {
      "base": [
        "Failed to create a compute <compute_resource> (VMWare) instance <fqdn>: Unknown Network ID: <network>
\n "
      ]
    },
    "full_messages": [
      "Failed to create a compute <compute_resource> (VMWare) instance <fqdn>: Unknown Network ID: <network>\n
 " 
    ]
  }
}
```

I then updated my code to include the network interface type, and still get the same error when doing an HTTP POST to either /api/hosts or /api/v2/hosts. I have attached the log.

I then created a compute profile, and added its compute_profile_id to the API call and removed the compute_attributes. That POST looks like this:

```
{
  "host": {
    "subnet_id": null,
    "name": "<fqdn>",
    "build": "1",
    "compute_profile_id": 2,
    "hostgroup_id": "81",
    "medium_id": null,
    "managed": "true",
    "environment_id": 40,
    "provision_method": "build",
    "ptable_id": 11,
    "compute_resource_id": 2,
    "overwrite": "false",
    "enabled": "1" 
  }
}
```

The response was:

```
{
  "errors": {
    "mac": [
      "has already been taken",
      "is invalid",
      "can't be blank" 
    ]
  },
  "full_messages": [
    "MAC address has already been taken",
    "MAC address is invalid",
    "MAC address can't be blank" 
  ],
  "id": null
}
```

Let me know if anything else will help.

Thanks for your help!

Greg

#6 - 03/06/2014 10:11 AM - Dominic Cleal

- File Screenshot from 2014-03-06 10_07_31.png added

I think there are three issues here:

1. "Unknown Network ID" on VMware: I'd probably double check the network IDs being used here, it's possible that something below Foreman (e.g. Fog or rbvmomi) changed these. The only way to check at the moment is by inspecting the HTML source on the new host form - see attached screenshot, the network IDs to use in the API would be "network-17" etc.

2. production.log shows errors like "Name can't be blank". This looks like a formatting issue with the uploaded JSON, as the parameters on the request show the entire JSON is being interpreted as a parameter name. Perhaps check in this instance that the content-type was correctly set.

3. compute_profile_id as noted above in c4 isn't processed in the correct order over the API. The UI happens to work because selecting a compute profile causes the form to refresh, so rather than relying on compute_profile_id being submitted in the form, it's actually submitting compute_attributes in full.

#7 - 03/06/2014 05:54 PM - Greg Petras

Hey Dominic -

#1 is definitely an issue. I was able to provision a host using the corresponding "network-<ID>" for the VM network. Here's an example:

```
{
  "host": {
    "subnet_id": null,
    "name": "<fqdn>",
    "build": "1",
    "compute_attributes": {
      "interfaces_attributes": {
        "0": {
          "network": "network-426",
          "type": "VirtualE1000",
          "_delete": ""
        }
      },
      "cluster": "QAC1-ESX3001C",
      "volumes_attributes": {
        "1": {
          "thin": "0",
          "datastore": "qac1_nfs_1",
          "name": "Hard disk 1",
          "size_gb": 20
        }
      },
      "power_action": "start",
      "memory_mb": 2048,
      "path": "<path>",
      "cpus": 1,
      "start": "1"
    },
    "hostgroup_id": "81",
    "medium_id": null,
    "managed": "true",
    "environment_id": 40,
    "provision_method": "build",
    "ptable_id": 11,
    "compute_resource_id": 2,
    "overwrite": "false",
    "enabled": "1"
  }
}
```

We used to be able to provision VM's using the friendly VM network name. This is definitely a change from previous behavior. Can this be fixed? Or, can these networks be made available through the API?

#2 I inadvertently attached a log without the Content-Type header set.

3. Sounds good. Is this something that can potentially be fixed in the API as well?

I think the biggest issue at this point is being able to get a list of network id's and their respective names from the API (#1).

Thanks again for your help.

Greg

#8 - 03/06/2014 07:11 PM - Greg Petras

Sorry for the formatting issues. Reposting for clarity.

Hey Dominic -

1. This is definitely an issue. I was able to provision a host using the corresponding "network-<ID>" for the VM network. Here's an example:

```
{
  "host": {
    "subnet_id": null,
    "name": "<fqdn>",
    "build": "1",
    "compute_attributes": {
      "interfaces_attributes": {
        "0": {
          "network": "network-426",
          "type": "VirtualE1000",
          "_delete": ""
        }
      },
      "cluster": "<cluster>",
      "volumes_attributes": {
        "1": {
          "thin": "0",
          "datastore": "<datastore>",
          "name": "Hard disk 1",
          "size_gb": 20
        }
      },
      "power_action": "start",
      "memory_mb": 2048,
      "path": "<path>",
      "cpus": 1,
      "start": "1"
    },
    "hostgroup_id": "81",
    "medium_id": null,
    "managed": "true",
    "environment_id": 40,
    "provision_method": "build",
    "ptable_id": 11,
    "compute_resource_id": 2,
    "overwrite": "false",
    "enabled": "1"
  }
}
```

We used to be able to provision VM's using the friendly VM network name. This is definitely a change from previous behavior. Can this be fixed? Or, can these networks be made available through the API?

2. I inadvertently attached a log without the Content-Type header set.

3. Sounds good. Is this something that can potentially be fixed in the API as well?

I think the biggest issue at this point is being able to get a list of network id's and their respective names from the API (#1).

Thanks again for your help.

Greg

#9 - 03/06/2014 08:27 PM - Dominic Cleal

- Related to Feature #4581: Implement available_networks API for VMware added

#10 - 03/06/2014 08:32 PM - Dominic Cleal

- Status changed from New to Feedback

1) I'm glad that worked. I've filed #4581 to track implementing an available_networks API backend for VMware. If you want to have a go at it, look at the code from #4222 - the generic part's already done, it just needs available_* methods implementing on the VMware compute resource.

3) Let's track this via #4250 - we need to definitely get compute_profile_id working, and probably a basic API around profiles themselves so you can see what's available.

If you're otherwise up and running then, I'll close this ticket and we'll track the more specific issues instead.

#11 - 03/06/2014 08:37 PM - Dominic Cleal

Oh, I forgot to say: I'm not sure how the network IDs can be changed, we haven't done anything ourselves to do this. I suspect it's either rbvmomi (the VMware Ruby bindings) or on vSphere itself, as I can't see any relevant changes in network code in Fog.

#12 - 03/07/2014 06:51 AM - Greg Petras

Thanks Dominic.

1. I've submitted a PR for the available_* methods for VMware compute resources (#4581). This is working on our 1.4 test system.

3. I agree. Definitely will help to make an API around compute_profiles as compute profiles are somewhat hidden right now without it.

I appreciate your help. You can close this issue out.

Thanks,

Greg

#13 - 03/07/2014 08:42 AM - Dominic Cleal

- Status changed from Feedback to Resolved

Fantastic, thanks for the PR. I'll review it soon.

Files

production.log	25.9 KB	03/05/2014	Greg Petras
Screenshot from 2014-03-06 10_07_31.png	124 KB	03/06/2014	Dominic Cleal