

Foreman - Bug #13779

Setting compute profile in EC2 provisioning over-rides image AMI with a default AMI

02/17/2016 04:17 PM - Matthew Ceroni

Status:	Duplicate	
Priority:	High	
Assignee:		
Category:	Compute resources - EC2	
Target version:		
Difficulty:		Fixed in Releases:
Triaged:		Found in Releases:
Bugzilla link:		Red Hat JIRA:
Pull request:		

Description

For EC2 provisioning, setting the compute profile on a new host causes a default AMI to be set that differs from the AMI set in the image associated with the compute resource.

Steps to reproduce:

- 1) add a new AWS EC2 compute resource
- 2) create an image, set an AMI (in this case an HVM AMI)
- 3) create t2.micro compute profile (t2.micro requires HVM AMI)
- 4) set the image on compute profile to the image you created in step 2
- 5) provision a host, setting compute profile and image (on Operating System Page)

AWS will report back message:

InvalidParameterCombination</Code><Message>Virtualization type 'hvm' is required for instances of type 't2.micro'. Ensure that you are using an AMI with virtualization type 'hvm'. For more information, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/virtualization_types.html

Which is confusing since the AMI set on the image is of type HVM. I setup a developer install of Foreman and enable compute resource API debug. The following was noted:

```
excon.request
  :body                => "Action=RunInstances&ImageId=ami-e0ec60d0&InstanceType=t2.micro&KeyName=foreman-1c5b55767-4b56-46a0-a8ec-955757c4f9df&MaxCount=1&MinCount=1&Placement.AvailabilityZone=us-west-2b&SecurityGroupId.1=sg-f52ffd92&SubnetId=subnet-420be726&Version=2014-10-01"
  :chunk_size          => 1048576
  :ciphers              => "HIGH:!SSLv2:!aNULL:!eNULL:!3DES"
  :connect_timeout     => 60
  :debug_request       => true
  :debug_response      => true
  :expects             => 200
  :headers             => {
    "Authorization" => "REDACTED"
    "Content-Type"  => "application/x-www-form-urlencoded"
    "Host"          => "ec2.us-west-2.amazonaws.com"
    "User-Agent"    => "fog-core/1.35.0"
    "x-amz-date"    => "20160217T205521Z"
  }
  :host                => "ec2.us-west-2.amazonaws.com"
  :hostname            => "ec2.us-west-2.amazonaws.com"
  :idempotent          => false
  :instrumentor        => Excon::StandardInstrumentor
  :instrumentor_name   => "excon"
  :method              => "POST"
  :middlewares         => [
    Excon::Middleware::ResponseParser
    Excon::Middleware::Expects
```

```

Excon::Middleware::Idempotent
Excon::Middleware::Instrumentor
Excon::Middleware::Mock
]
:mock => false
:nonblock => true
:omit_default_port => false
:path => "/"
:persistent => false
:port => 443
:query => nil
:read_timeout => 60
:response_block => #<Proc:0x0000000bba69a8@/root/foreman/vendor/ruby/gems/fog-xml-0.1.2/lib
/fog/xml/sax_parser_connection.rb:29 (lambda)>
:retries_remaining => 4
:retry_limit => 4
:scheme => "https"
:ssl_verify_peer => true
:tcp_nodelay => false
:thread_safe_sockets => true
:uri_parser => URI
:versions => "excon/0.45.4 (x86_64-linux) ruby/2.0.0"
:write_timeout => 60
excon.error
:error => #<Excon::Errors::BadRequest: Expected(200) <=> Actual(400 Bad Request)
excon.error.request
:body => "Action=RunInstances&ImageId=ami-e0ec60d0&InstanceType=t2.micro&KeyName=
foreman-1c5b55767-4b56-46a0-a8ec-955757c4f9df&MaxCount=1&MinCount=1&Placement.AvailabilityZone=us-
west-2b&SecurityGroupId.1=sg-f52ffd92&SubnetId=subnet-420be726&Version=2014-10-01"
:chunk_size => 1048576
:ciphers => "HIGH:!SSLv2:!aNULL:!eNULL:!3DES"
:connect_timeout => 60
:debug_request => true
:debug_response => true
:expects => 200
:headers => {
  "Authorization" => "REDACTED"
  "Content-Length" => 250
  "Content-Type" => "application/x-www-form-urlencoded"
  "Host" => "ec2.us-west-2.amazonaws.com"
  "User-Agent" => "fog-core/1.35.0"
  "x-amz-date" => "20160217T205521Z"
}
:host => "ec2.us-west-2.amazonaws.com"
:hostname => "ec2.us-west-2.amazonaws.com"
:idempotent => false
:instrumentor => Excon::StandardInstrumentor
:instrumentor_name => "excon"
:method => "POST"
:middlewares => [
  Excon::Middleware::ResponseParser
  Excon::Middleware::Expects
  Excon::Middleware::Idempotent
  Excon::Middleware::Instrumentor
  Excon::Middleware::Mock
]
:mock => false
:nonblock => true
:omit_default_port => false
:path => "/"
:persistent => false
:port => 443
:query => nil
:read_timeout => 60
:response_block => #<Proc:0x0000000bba69a8@/root/foreman/vendor/ruby/gems/fog-xml-0.1.2/lib
/fog/xml/sax_parser_connection.rb:29 (lambda)>
:retries_remaining => 4

```

```

:retry_limit      => 4
:scheme           => "https"
:ssl_verify_peer  => true
:tcp_nodelay      => false
:thread_safe_sockets => true
:uri_parser       => URI
:versions         => "excon/0.45.4 (x86_64-linux) ruby/2.0.0"
:write_timeout    => 60
excon.error.response
:body             => "<?xml version='1.0' encoding='UTF-8'>\n<Response><Errors><Error><Code>InvalidParameterCombination</Code><Message>Virtualization type 'hvm' is required for instances of type 't2.micro'. Ensure that you are using an AMI with virtualization type 'hvm'. For more information, see http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/virtualization_types.html</Message></Error></Errors><RequestID>3d90ec03-d920-4721-bea7-c17alb7e0934</RequestID></Response>"
:headers          => {
  "Date"           => "Wed, 17 Feb 2016 20:55:21 GMT"
  "Server"         => "AmazonEC2"
  "nnCoectio"     => "close"
}
:local_address    => "10.200.18.245"
:local_port       => 50057
:reason_phrase    => "Bad Request"
:remote_ip        => "54.240.251.0"
:status           => 400
:status_line      => "HTTP/1.1 400 Bad Request\r\n"
>

```

Looking at the body request

```
body => "Action=RunInstances&ImageId=ami-e0ec60d0&InstanceType=t2.micro&KeyName=foreman-1c5b55767-4b56-46a0-a8ec-955757c4f9df&MaxCount=1&MinCount=1&Placement.AvailabilityZone=us-west-2b&SecurityGroupId.1=sg-f52ffd92&SubnetId=subnet-420be726&Version=2014-10-01"
```

You can see that it set ImageId to ami-e0ec60d0, that is not the AMI set in the image under the Compute Resource.

If you leave the compute profile blank on the host provisioning page you get

```
:body => "Action=RunInstances&ImageId=ami-1255b321&InstanceType=t2.micro&KeyName=foreman-1c5b55767-4b56-46a0-a8ec-955757c4f9df&MaxCount=1&MinCount=1&Placement.AvailabilityZone=us-west-2b&SecurityGroupId.1=sg-f52ffd92&SubnetId=subnet-420be726&Version=2014-10-01"
```

Where ImageId ami-1255b321 is the AMI set in the image under the Compute Resource.

Maybe of note, vendor/ruby/gems/fog-aws-0.8.1/lib/fog/aws/models/compute/server.rb specifies this default AMI (ami-e0ec60d0) that is being sent.

Related issues:		
Is duplicate of Foreman - Bug #13980: VMWARE: since 1.10, NIC type and Networ...	Closed	03/01/2016

History

#1 - 02/18/2016 01:52 AM - Matthew Ceroni

Verified info in database is accurate:

```
foreman=> select * from compute_attributes;
 id | compute_profile_id | compute_resource_id | name | vm_attrs
-----+-----+-----+-----+-----
 | updated_at | created_at |
-----+-----+-----+-----+-----
 1 | 4 | 1 | t2.micro - Micro Instance | --- !ruby/hash:ActiveSupport::Has
hWithIndifferentAccess | 2016-02-18 00:29:44.802081
 | 2016-02-18 00:30:07.964041
 | : flavor_id: t2.micro
 | : image_id: ami-1255b321
```

```

: availability_zone: us-west-2b
: subnet_id: subnet-420be726
: security_group_ids:
: - ''
: - sg-f52ffd92
: managed_ip: private
:

```

AMI is set correct in above compute profile.

```

foreman=> select * from images;
 id | operatingsystem_id | compute_resource_id | architecture_id | uuid | username |
 name | updated_at | iam_role | user_data | password
-----+-----+-----+-----+-----+-----+-----
 1 | 1 | 1 | 1 | 1 | ami-1255b321 | ec2-user | CentOS Linux 6 x8
6_64 HVM EBS 20150928_01 | 2016-02-18 00:28:37.77874
5 | 2016-02-18 00:28:37.778745 | t
(1 row)

```

Same AMI is set in the image.

But when using the above profile and image when provisioning the AMI in the issue details is sent.

#2 - 02/18/2016 03:00 AM - Dominic Cleal

- Description updated
- Category set to Compute resources - EC2

The parameters sent in the POST when submitting the form from production.log might also show the image ID that's selected, which would rule out whether this is happening in the UI or somewhere later.

#3 - 02/22/2016 02:25 PM - Matthew Ceroni

Post parameters (on host create)

```

2016-02-22 11:19:40 [app] [I] Parameters: {"utf8"=>"", "authenticity_token"=>"XXXXXXXX", "host"=>{"name"=>"aws-test-04", "location_id"=>"1",
"hostgroup_id"=>"", "compute_resource_id"=>"1", "compute_profile_id"=>"4", "environment_id"=>"", "puppet_proxy_id"=>"", "managed"=>"true",
"progress_report_id"=>"[FILTERED]", "type"=>"Host::Managed", "interfaces_attributes"=>{"0"=>{"_destroy"=>"0", "type"=>"Nic::Managed", "mac"=>"",
"identifier"=>"", "name"=>"aws-test-04", "domain_id"=>"1", "subnet_id"=>"1", "ip"=>"10.140.0.55", "managed"=>"1", "primary"=>"1", "provision"=>"0",
"execution"=>"1", "virtual"=>"0", "tag"=>"", "attached_to"=>""}}, "compute_attributes"=>{"flavor_id"=>"t2.micro", "availability_zone"=>"us-west-2b",
"subnet_id"=>"subnet-420be726", "security_group_ids"=>["", "sg-f52ffd92"], "managed_ip"=>"private"}, "architecture_id"=>"1",
"operatingsystem_id"=>"1", "provision_method"=>"image", "build"=>"1", "ptable_id"=>"", "disk"=>"", "root_pass"=>"[FILTERED]",
"is_owned_by"=>"5-Users", "enabled"=>"1", "model_id"=>"", "comment"=>"", "overwrite"=>"false"}, "capabilities"=>"image", "provider"=>"EC2",
"bare_metal_capabilities"=>"build"}

```

Compute profile_id is being set to 4.

```

foreman=> select vm_attrs.name from compute_attributes where compute_profile_id = 4;
 vm_attrs | name
-----+-----
--- !ruby/hash:ActiveSupport::HashWithIndifferentAccess | t2.micro - Micro Instance
flavor_id: t2.micro
image_id: ami-1255b321
availability_zone: us-west-2b
subnet_id: subnet-420be726
security_group_ids:
- ''
- sg-f52ffd92

```

managed_ip: private

(1 row)

foreman=>

Image ID in vm_attrs is set to ami-1255b321 but that is not the ami sent in the request to AWS.

#4 - 02/23/2016 03:47 AM - Dominic Cleal

The image_id is missing from the list of parameters, which should be set to the OS image ID with that AMI set. The image ID should be set on the operating system tab when creating the host - do you have that dropdown showing?

The AMI from the compute profile isn't used directly - it should be loaded into the image dropdown menu and sent back to the server.

#5 - 02/23/2016 01:03 PM - Matthew Ceroni

Yes, I am setting the image on the OS tab.

The compute profile has the image set as well and ideally (there is another ticket open on this) setting the compute profile should auto populate the image on the OS tab. Currently it doesn't.

However, if I set the compute profile (which requires you to set an image) and even though the OS tab isn't pre-populated with the image setting from the compute profile, I set the image myself, that gets clobbered and a default AMI sent.

I specifically can not set the compute profile on the new host settings.

#6 - 02/23/2016 07:59 PM - Matthew Ceroni

Further troubleshooting, I actually missed the image_id in the post request (in production.log)

So to generate the error:

- 1) set compute profile on Host tab of new host
- 2) then set the OS details along with the image you want to use

The post request is

```
Parameters: {"utf8"=>"", "authenticity_token"=>"fIZhUpoVF1M5grxtz+/xPLrnMs3yyTHuPEk/ZB1r5jg=", "host"=>{"name"=>"test2", "hostgroup_id"=>"", "compute_resource_id"=>"1", "compute_profile_id"=>"5", "environment_id"=>"", "managed"=>"true", "progress_report_id"=>"[FILTERED]", "type"=>"Host::Managed", "interfaces_attributes"=>{"0"=>{"_destroy"=>"0", "type"=>"Nic::Managed", "mac"=>"", "identifier"=>"", "name"=>"test2", "domain_id"=>"1", "subnet_id"=>"1", "ip"=>"10.140.0.2", "managed"=>"1", "primary"=>"1", "provision"=>"1", "virtual"=>"0", "tag"=>"", "attached_to"=>""}}, "compute_attributes"=>{"flavor_id"=>"t2.micro", "availability_zone"=>"us-west-2b", "subnet_id"=>"subnet-420be726", "security_group_ids"=>["", "sg-f52ffd92"], "managed_ip"=>"private"}, "architecture_id"=>"1", "operatingsystem_id"=>"1", "provision_method"=>"image", "build"=>"1", "medium_id"=>"", "ptable_id"=>"", "disk"=>"", "root_pass"=>"[FILTERED]", "is_owned_by"=>"3-Users", "enabled"=>"1", "model_id"=>"", "comment"=>"", "overwrite"=>"false"}, "capabilities"=>"image", "provider"=>"EC2", "bare_metal_capabilities"=>"build"}
```

Notice that the compute_attributes does not contain an image_id. It contains all other parameters required for building in EC2 (subnet_id, availability zone, etc).

The above will fail with the error noted in the ticket details due to the fact that the AMI sent is not the one set in the image.

To successfully provision:

- 1) Compute Host tab but leave compute profile blank
- 2) complete the OS tab, filling out the image to use (thus the AMI to use)
- 3) go back to Host tab and set the compute profile as this will populate all the virtual machine settings (availability zone, security groups, etc)

The post request is as follows:

```
2016-02-23T19:51:45 [app] [I] Parameters: {"utf8"=>"", "authenticity_token"=>"fIZhUpoVF1M5grxtz+/xPLrnMs3yyTHuPEk/ZB1r5jg=", "host"=>{"name"=>"test2", "hostgroup_id"=>"", "compute_resource_id"=>"1", "compute_profile_id"=>"5", "environment_id"=>"", "managed"=>"true", "progress_report_id"=>"[FILTERED]", "type"=>"Host::Managed", "interfaces_attributes"=>{"0"=>{"_destroy"=>"0", "type"=>"Nic::Managed", "mac"=>"", "identifier"=>"", "name"=>"test2", "domain_id"=>"1", "subnet_id"=>"1", "ip"=>"10.140.0.2", "managed"=>"1", "primary"=>"1", "provision"=>"1", "virtual"=>"0", "tag"=>"", "attached_to"=>""}}, "compute_attributes"=>{"flavor_id"=>"t2.micro", "availability_zone"=>"us-west-2b", "subnet_id"=>"subnet-420be726", "security_group_ids"=>["", "sg-f52ffd92"], "managed_ip"=>"private", "image_id"=>"ami-1255b321"}, "architecture_id"=>"1", "operatingsystem_id"=>"1", "provision_method"=>"image", "build"=>"1", "medium_id"=>"", "ptable_id"=>"", "disk"=>"", "root_pass"=>"[FILTERED]", "is_owned_by"=>"3-Users", "enabled"=>"1", "model_id"=>"", "comment"=>"", "overwrite"=>"false"}, "capabilities"=>"image", "provider"=>"EC2", "bare_metal_capabilities"=>"build"}
```

Notice the compute_attributes, this time it contains image_id, which is correctly set to the AMI specified in the image. The above will complete successfully.

So there appears to be an order of operations issues here.

#7 - 02/24/2016 11:38 PM - Matthew Ceroni

I have been trying to debug the UI but haven't had much luck in figuring out where this isn't being set.

Did validate through Chrome Developer Tools that the form data submitted on save doesn't contain host[compute_attributes][image_id] when you first set compute profile vs leaving it blank and setting image on OS tab. It also appears that if you set profile the image selection on OS tab is greyed out. You can then go clear compute profile, go back to OS tab and select the image but the form parameter still isn't set.

#8 - 03/02/2016 09:46 AM - Dominic Cleal

I tentatively think this might be the same issue as [#13980](#), where the compute profile simply doesn't get applied when a compute resource is selected.

Perhaps you could try the same workaround as the reporter there, which is to set the compute profile on a host group, then on the New Host form, select the host group first before selecting the compute resource.

#9 - 03/11/2016 07:17 PM - Matthew Ceroni

This work around worked. Setting the compute profile in the host group and selecting it prior to setting the compute resource succeeds.

#10 - 03/14/2016 04:19 AM - Dominic Cleal

- Is duplicate of Bug #13980: VMWARE: since 1.10, NIC type and Network different from profile configuration added

#11 - 03/14/2016 04:19 AM - Dominic Cleal

Thanks for confirming and your investigation. I'm working on a fix for the other ticket, so I'll close this one in favour of that.

#12 - 03/14/2016 04:20 AM - Dominic Cleal

- Status changed from New to Duplicate

#13 - 03/14/2016 08:09 AM - Dominic Cleal

There's a relatively small patch now linked from the other ticket if you're interested in trying it.