

## Smart Proxy - Feature #1224

### DNSMadeEasy/Foreman/SmartProxy integration

10/13/2011 12:07 AM - Brian Gupta

<b>Status:</b> New	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Category:</b> DNS	
<b>Target version:</b>	
<b>Difficulty:</b>	<b>Fixed in Releases:</b>
<b>Triaged:</b>	<b>Found in Releases:</b>
<b>Bugzilla link:</b>	<b>Red Hat JIRA:</b>
<b>Pull request:</b>	
<b>Description</b>	
Now that DNSmadeeasy has REST API, we should support it <a href="http://www.dnsmadeeasy.com/enterprise-dns/rest-api/">http://www.dnsmadeeasy.com/enterprise-dns/rest-api/</a>	
<b>Related issues:</b>	
Related to Foreman - Tracker #5409: DNS Proxy Improvements	<b>New</b>

#### History

##### #1 - 10/13/2011 12:16 AM - Brian Gupta

Here is some sample code <https://github.com/bgupta/dnsmadeeasy>

##### #2 - 07/09/2013 09:45 PM - Jeremy Kitchen

This seems like it's actually pretty easy, however I have a quick implementation question:

There exists a rubygem called 'dnsmadeeasy/api' which we could use. However, this is **not** packaged for debian, and most likely not packaged for other things as well, so would probably not be a good dependency to add.

The license for the rubygem is unknown, the gemspec file has no license information and there doesn't seem to be a github or anything for it, so I don't think we can just include it in vendor/ or something.

Since it's only 2 calls we really need to implement with dnsmadeeasy, should we just use net-http or rest-client or such to do the calls directly?

Once I get an answer on this I can have this implemented pretty quickly.

I'm also contacting the creator of the rubygem and asking about licensing information.

Thanks!

##### #3 - 07/09/2013 11:56 PM - Brian Gupta

Jeremy Kitchen wrote:

This seems like it's actually pretty easy, however I have a quick implementation question:

There exists a rubygem called 'dnsmadeeasy/api' which we could use. However, this is **not** packaged for debian, and most likely not packaged for other things as well, so would probably not be a good dependency to add.

The license for the rubygem is unknown, the gemspec file has no license information and there doesn't seem to be a github or anything for it, so I don't think we can just include it in vendor/ or something.

Since it's only 2 calls we really need to implement with dnsmadeeasy, should we just use net-http or rest-client or such to do the calls directly?

Once I get an answer on this I can have this implemented pretty quickly.

I'm also contacting the creator of the rubygem and asking about licensing information.

Thanks!

I'd suggest NOT using a full gem for this as the API is dirt simple. That said, feel free to steal any of the code/examples I wrote (all GPLv2+):

<https://github.com/bgupta/dnsmadeeasy>

This is likely the script that would be most useful example: <https://github.com/bgupta/dnsmadeeasy/blob/master/ec2-set-dme-dns.rb>

-Brian

#### #4 - 07/10/2013 12:37 AM - Jeremy Kitchen

ok, that's what I was figuring, just hack the http requests together and it should Just Work. I'll try to bang this out by the end of the week, I have a bunch of meetings tomorrow and jury duty thursday (and hopefully ONLY thursday)

thanks!

#### #5 - 07/15/2013 05:27 PM - Jeremy Kitchen

another design question now: with dnsmadeeasy you apparently need to RESTfully write directly to a domain's endpoint, based on ID:

POST /V2.0/dns/managed/<id>/records, for example

I could look this up every time, but in our case, we have several thousand domains managed by dnsmadeeasy and having to parse them out and figure out which domain is which every time ... oof. Also, since our foreman domain is actually a subdomain of the domain we have hosted with dnsmadeeasy, there's no 1:1 relationship so we'd have to go through each domain in the list and try to find the best match, which just doesn't sound fun.

So, I was thinking of having the proxy setup similar to dhcp where you specify a list of domains and their IDs in the proxy's settings.yml file and it'll use that to look up the domain's ID.

If Foreman gave the domain's parameters to the smart-proxy during the API call then I could just make this a parameter of the domain, so that's another route which could be taken.

Thoughts?

#### #6 - 07/15/2013 05:34 PM - Brian Gupta

Jeremy Kitchen wrote:

another design question now: with dnsmadeeasy you apparently need to RESTfully write directly to a domain's endpoint, based on ID:

POST /V2.0/dns/managed/<id>/records, for example

I could look this up every time, but in our case, we have several thousand domains managed by dnsmadeeasy and having to parse them out and figure out which domain is which every time ... oof. Also, since our foreman domain is actually a subdomain of the domain we have hosted with dnsmadeeasy, there's no 1:1 relationship so we'd have to go through each domain in the list and try to find the best match, which just doesn't sound fun.

So, I was thinking of having the proxy setup similar to dhcp where you specify a list of domains and their IDs in the proxy's settings.yml file and it'll use that to look up the domain's ID.

If Foreman gave the domain's parameters to the smart-proxy during the API call then I could just make this a parameter of the domain, so that's another route which could be taken.

Thoughts?

Yuck, hadn't really thought about that, but not loving the idea of configuring this in a text file. I'm starting to wonder if the proxy is the right place to do this. (IE: We are doing a lot more in Foreman directly than we did in the past, perhaps your idea to make this a parameter to the proxy call might be one way.)

I think taking this to the foreman-dev mailing list is the right path at this point.

-Brian

#### #7 - 07/15/2013 06:55 PM - Jeremy Kitchen

ok, will do that in a bit, thanks!

#### #8 - 04/23/2014 02:34 PM - Stephen Benjamin

- Related to Tracker #5409: DNS Proxy Improvements added

#### #9 - 07/10/2015 09:36 AM - Dominic Cleal

- Category set to DNS

The Smart Proxy's DNS module can now be extended by plugins, thanks to [#7008](#). This would be a great way to start developing this feature if anybody's interested.

Here are some useful links:

- [http://projects.theforeman.org/projects/foreman/wiki/How\\_to\\_Create\\_a\\_Smart-Proxy\\_Plugin](http://projects.theforeman.org/projects/foreman/wiki/How_to_Create_a_Smart-Proxy_Plugin) - developer information
- [https://github.com/theforeman/smart\\_proxy\\_dns\\_plugin\\_template](https://github.com/theforeman/smart_proxy_dns_plugin_template) - a near-working template to create a new DNS provider

- [https://github.com/theforeman/smart\\_proxy\\_dns\\_route53](https://github.com/theforeman/smart_proxy_dns_route53) - a working example adding AWS Route53 support