

Foreman - Feature #5752

We need an additional context Stage or Environment, alongside Location and Organization

05/16/2014 12:47 AM - Simon Mügge

Status: New	
Priority: Normal	
Assignee:	
Category: Organizations and Locations	
Target version:	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request:	
Description	
<p>Hi!</p> <p>What?</p> <p>I would like foreman to have a third, optional Context, namely "Stage", functionally equivalent to Organizations Locations. As in: "Pilot", "Live", "Pre-live", "CI", "Devel", "Sandbox".</p> <p>At first glance you might want to say "do something with a custom fact and be done with it", but at large scale this does not actually suffice.</p> <p>"Just add it as a location" does not work either: location_fact = 4xcountries/x2cities/x3datacenters is bad enough, but when you have that times 5 root trees (earth_live, earth_dev, earth_pre-live).. try it, look at it, and tell me that does not look completely wrong. We could just manage 5 separate foreman setups.....</p> <p>All of this is written from the perspective of someone who provides self-service infrastructure on a large scale, and has as direct "customers" A LOT of other knowledgeable admin (~150+) and dev (2-10+ times the number of admins) folk, who should all be able to use the same foreman, for their needs, without me or them going insane.</p> <p>Abbreviated & abstracted resulting features:</p> <ul style="list-style-type: none">- natively stage SmartProxy infrastructure and computational resources- dissect infrastructure (not only things that have facts) along a 3rd axis- enhanced rolemanagement through third logical layer of abstraction (reduces total number of required roles)- opens up to native support for CI things like katello, simply by having a native concept of "stages"? (not sure how off topic this is)- enables nested Lifecyclemanagement for the infrastructure that is bringing all "normal" Foreman users (admins) their Foreman-integrated infrastructure. Also, hurts brains. <p>Why:</p> <p>The Goal is to have "Infrastructure" not be a 2-, but a 3-dimensional thing, and our scale has been a very interesting teacher over the ~10 years I've seen it grow as to why that is actually true for everyone, just not (as) necessary. As an optional structural addition it would carry benefit for most, I believe.</p> <p>Scale:</p> <p>We are just in the process of converting an existing "Foreman as Puppetreport Frontend only, no organizations, no nothing", to a possibly no-holds-barred, all foreman setup.</p> <p>Right now there are 10000 hosts in foreman, and we will, over the course of about 2-4 years, get to ~25000 - which is just looking at actual hardware/vms we have in our infrastructure today and would make sense to assimilate currently.</p> <p>We currently have a set of vastly different, huge, sometimes redundant and sometimes in various stages of disrepair "toolchains", wherever the term might loosely apply, stemming from essentially taking 4 major IT companies worth of everything, putting them all in roughly the same building / datacenter / everything and saying "play nicely" - and then its 10 years later. Not saying that</p> <p>We (infrastructure team) have every debian for the last ~ 10 years installable, last ~7 years for RedHat/Centos, ~6 different CuriosityOS kind of things (roughly 70/30 Debian/Redhat/Stuff). Then there is windows stuff I do not yet have to care about, but who</p>	

knows.

We manage location based infrastructure-as-a-service everything internally (at least that's the goal), and a lot of it. ldap, puppetmaster, dhcp, tftp, dns (via custom API, no idea yet), \$os repositories, logserver and on and on, customizable by our "internal customers", which are on any day about 250+ admins and.. 4-10? times the number of devs.

Mind you, "none" of that is in foreman yet, we currently use an internally developed tool in production that we have used for the better part of 10 years, but the evaluation process has been a blast so far.

"Assimilation of infrastructure" is the best term that comes to mind, quite an enjoyable process with 1.5 and a fact-loaded foreman/puppet DBs, I have to hand it to you.

But on, to the point, and why "Stages"?

All of that bragging about our infrastructure had a point, and it does go on a little bit, just to show "why stages".

How our Locations, Organizations and, yes, Stages look:

-Locations:

Pattern: earth/(\$country/\$city/\$datacenter)/\$room/\$row/\$rack/\$slot

\$Data:country = "4", \$city = "3", \$datacenter="3"

Which are 36 actual host-containing locations and WHOLE LOT of nested structure.

Where the bracketed part is the only part we actually currently use as the nested structure in foreman, but more on that, later.

-Organizations

Pattern: corp/\$subsidiary/lvl_0/lvl_1/lvl_2/lvl_3/lvl_4/lvl_5/members

Data: \$subsidiary = "3", \$lvl_0 = "2", lvl_5 = "4", \$members = "7",

Here we are currently "stuck" at only taking lvl_5 and be done with it until LDAP groups are with us, and we will not likely go Live with this infrastructure overhaul without them. Which is fine, I saw that feature in "ready for testing" :)

-Stages:

Pattern: \$stage)

Data: \$stage= [pilot, live, prelive, qa, dev, sandbox]

"Haha! Puppetenvironments!" you say.

"Nope," I say, "of those we have.. 156, I believe I saw today. ALL necessary? Probably not. Are at least 60+ a fact of life? Yes, at this scale, a lot of insane things are.

Fun Fact: We have a host with an organization_fact of 'software , & other stuff'. That came to be into Foreman via A Database into THE LDATEST OF ALL LDAPS into a Configtool into a Rubyapi into a Customfact into 'location_fact' because there is a bug with that. Cool that it worked. But that ' , & ' will likely never ever be changed by a human for fear of the company burning down..."

1. Nested example, bit of a mind melting example to be able

2. to conveniently stage the infrastructures infrastructure.

#Pattern: \$type/(\$stage)

#Data: \$type = [customer-directed-services, infrastructure], \$stage= [pilot, live, prelive, qa, dev. sandbox]

2. Other possibility (not thought through) could be something like

#Data: \$type = [infrastructure, storage, databases, backend_services, middleware_services, frontend_services], \$stage= [pilot, live, prelive, qa, dev. sandbox]

We want be able to use the stage as essentially a copy of the Locations Context, we would use it to:

- assign roles to a user/usergroup

- designate smartproxys (see nested stages)

- designate computational resources

- hosts require a stage as an attribute when feature enabled (see location_fact, organization_fact)

- assign locations/organizations

- trends

What then?

This (for us) serves multiple real world functions:

- We could simply see (as in charts and stuff) the state of your whole infrastructure, dissected by stage.

- We could delegate the ability to create a task normally performed by an ADMIN (creating a new partitioning template) to DEVs in sandbox, devel, and qa, but only have a "real ADMIN" be able to move it over to prelive, live and pilot.

- Computational Resources. I want to our DEVs to be able to have an all out war at who can hog the most resources - with their own

liboovirt machine that generates a slice each to sandbox only, while dev and qa share another, where only ADMINS can create new hosts but devs can still (re) deploy them.

-- Again, possible to implement with current model through Roles afaiu, until ADMNs are actually about 4-6 different roles for different levels of admin-access to foreman based on Location and Organization, then theres 6 more DEV roles, a couple more for SECURITY, MANAGEMENT and 3RD_LEVEL and than all that times 5 Stages derived from a custom fact - and you still would not have that other functionality.

- We could natively Stage smartproxys via Foreman; see Computational resources, just different type of "customers".

- [nested] We could test different stages of "infrastructure" benaeath what ever is using it as its infrastructure (see above, makes you a bit insane)

-- How do you let other people continuously test new infrastructure? People hate that moment of "we're swithcing DNS servers. As in BIND to Powerdns". We had that. Could really have helped. Especially in conjunction with

- Foreman would be opened up to "other kinds of lifecylce manaegement" other than just that for a host, it would open the door for CI on an infrastructure level (think 10 stages, the 3 are for devs to play, 5 CI one another into the next stage and the last 2 stages require an actual manual action to finally propagate to live.

Even when implemented "only" as a 3rd way to limit access to all kinds of resources, for large scale environments there would be an easy benefit.

Simon

History

#1 - 05/16/2014 07:52 AM - Dominic Cleal

- *Category set to Organizations and Locations*