mode (+ns) by <u>card.freenode.net</u> (13:47:07)

mode (-o dizz1) by services. (13:47:07)

13:47:07 card.freenode.net: (notice) *** Notice -- TS for <u>#occi</u> changed from 1309438027 to 1256737190

mode (+ct-s) by services. (13:47:07)

mode (+o ChanServ) by services. (13:47:07)

ChanServ [ChanServ@services.] entered the room. (13:47:07)

ChanServ left the room. (13:47:07)

befreax [~*tmetsch@dslb-084-056-091-101.pools.arcor-ip.net*] entered the room. (14:42:23) befreax left the room (quit: Quit: Leaving.). (14:55:40)

macredcape [~markcarl@216-160-151-78.hlrn.qwest.net] entered the room. (14:58:21)

15:13:20 macredcape: Is there a call now?

ffeldhaus [~Adium@dynam142.itmc.tu-dortmund.de] entered the room. (15:14:09)

15:15:28 ffeldhaus: Hi! Who's already there?

15:21:23 dizz1: hey I"m here (soz wasn't watchin the room)

rnyren [*c2ed8e11@gateway/web/freenode/ip.194.237.142.17*] entered the room. (15:24:38) 15:24:53 rnyren: hi there

15:25:37 rnyren: sorry I missed the start of the meeting

15:27:33 dizz1: np - so did I 😇

15:27:50 rnyren: btw, do you have an update on the occi test tools?

15:27:58 rnyren: are they updated to the latest spec now?

15:28:06 rnyren: i.e. support for attribute properties etc

15:28:09 dizz1: none from me - thijs is the best man for that

15:28:29 dizz1: I did however make some minor changes to the ANTLR grammar (java specific)

15:28:56 rnyren: I have updated the occi-py library to support the latest stuff. Not released yet though, would like to do some more testing

15:29:13 rnyren: ok, nice. Did you remove the support for single quotes? 🙂

15:29:40 rnyren: it is not compatible with the RFCs 😌

15:30:06 dizz1: naw - that's in there still

15:30:34 dizz1: but it's not a biggie to remove or for someone to do so via a git-fork

15:31:15 rnyren: no, very easy. Just the thought of people using the antlr grammar to see if their stuff is valid or not. Nice to be a bit strict then

15:31:39 dizz1: it's something I can sort out after SLA@SOI work dies down

15:31:50 rnyren: I can do a fork and send you a pull request 🥰

15:32:08 rnyren: anyhow, any progress on the client side?

15:32:15 dizz1: (fyi: https://github.com/dizz/occi-grammar/issues/4)

15:32:45 ffeldhaus: I have a couple of things I like to discuss which came up during implementing OCCI for OpenNebula

15:32:45 dizz1: I know there's some work on the jclouds client from SLA@SOI, but I've yet to try it out 15:33:07 rnyren: I've been working on a occi client abstraction in occi-py, don't know when I'll have time to complete it though

15:34:32 dizz1: is that up on any public place?

15:34:34 rnyren: funny thing there from the client perspective is that although <u>occi.core.id</u> is unique within the service provider namespace it is not sufficient to identify a resource instance over OCCI Http

15:34:41 rnyren: not yet

15:35:04 rnyren: dev-branch now only. Will commit when it is closer to being actually usable 15:35:18 dizz1: ya - <u>occi.core.id</u> can only be unique if presented as a fully qualified URI

15:35:20 dizz1: cool! 💛

15:35:42 dizz1: I don't suppose you have anytime for a 200-300 word post? 🕓

15:35:49 rnyren: nah, not entirely true. <u>occi.core.id</u> plus the Kind category should do it

15:36:30 rnyren: I have great plans for an entire blog series on all the cool stuff you can do with occi (with occi-py as the example of course) but time time

15:36:42 rnyren: :')

15:37:11 ffeldhaus: a blog series would be great

15:37:52 dizz1: you fancy taking start at it Florian? 15:38:09 ffeldhaus: why not. I have plenty of great news to share

15:38:19 dizz1: awesome! Ӵ 15:38:25 rnyren: yeah, cool!

15:38:31 dizz1: well don't be shy - shout out!

15:38:50 ffeldhaus: first, we ramping up the work on the ruby OCCI server

15:39:00 ffeldhaus: we use ANTLR now

15:39:07 rnyren: nice

15:39:13 dizz1: +1 (hope google doesn't sue me 🕓)

15:39:14 ffeldhaus: we have written a client for stress testing and verification

15:39:37 ffeldhaus: we are able to use NGINX and Apache now as web servers using Phusion Passenger

15:40:19 ffeldhaus: we are currently finnishing the work on making the OpenNebula backend compatible with the soon to be released OpenNebula 3.0

15:40:40 ffeldhaus: then the server is fully functional and supports actions, templates etc.

15:41:01 dizz1: that def sounds like a cool blog post in the making!

15:42:05 ffeldhaus: we will move from Ruby to Ruby on Rails to make handling the code easier. We are also now independent of OpenNebula and the OCCI Server can be deployed standalone, containing all the necessary libraries to talk to OpenNebula

15:42:41 ffeldhaus: for the cloud plugfest in 2 weeks we try to create a very simple webinterface to show all the nice features of OCCI

15:42:53 rnyren: how did you solve upload of vm images?

15:44:33 ffeldhaus: currently uploading of images is still done using the body of the POST request and thus breaking the text/plain rendering. We have discussed including the vm images as a "header" field together with the other OCCI headers in the body, but this is not really nice

15:44:59 ffeldhaus: it's probably best to talk about this with the people implementing OCCI clients 15:45:56 dizz1: I'd suggest that you use multipart for the upload of images

15:46:06 dizz1: one part can contain text/plain

15:46:12 dizz1: the other an octet stream

15:46:50 dizz1: I've been using multipart in the SLA@SOI OCCI implementation and it works out well 15:46:59 dizz1: (I'm not uploading images however)

15:47:06 ffeldhaus: good idea, I will try it out tomorrow

15:47:53 ffeldhaus: during implementation of OS templates and Resource templates I encountered a few problems or better I had a few ideas

15:48:28 rnyren: another slightly easier way is to use the headers for ordinary attributes and use content-type octet-stream for the whole body where you put the image data

15:49:19 rnyren: since application/octet-stream still is undefined in an occi context you don't break anything, you're just not compliant

15:49:22 fieldbaus: that's what I did until now but this would

15:49:22 ffeldhaus: that's what I did until now, but this would break the compatibility with clients which don't support text/occi

15:49:43 rnyren: but multipart is probably the best solution going forward

15:49:56 dizz1: rnyren: sounds like something for v1.2

15:50:02 rnyren: mmm, but clients MUST support text/occi

15:50:05 ffeldhaus: Shouldn't be too hard to get multipart working with ruby

15:50:17 rnyren: @dizz: indeed

15:50:46 rnyren: so what cool stuff did you figure out with templates?

15:50:49 ffeldhaus: regarding the templates: currently the spec says, that these templates are only implementation specific and can't be changed or viewed by the clients. I would like to modify this and introduce a new location /templates within the OCCI locations

15:51:38 ffeldhaus: there the users or some users with special rights can create templates which are directly attached to a new mixin

15:51:55 dizz1: the setting of a location is one that is managed by the provider

15:52:03 dizz1: just like location for compute etc

15:52:18 dizz1: permissions are then an aspect outside of the OCCI remit

15:52:45 rnyren: you could (and still be compliant) create a new Kind which mapps into /templates

and magically creates new templates on the fly

15:53:26 ffeldhaus: that could be a good solution

15:53:31 rnyren: the resource instances of the "template-Kind" would then correspond to the dynamic templates

15:54:23 rnyren: as long as you follow the current attributes/categories rules for resource instances you're still compliant. At least from the point of view that an occi client will be able to create templates without any modifications

15:55:11 rnyren: I mean, you could have a example.com.templates.body attribute which is some format defining the template

15:55:57 ffeldhaus: this would give the clients much greater control over templates. As far as I would use an OCCI cloud, I would try to offer the users as much templates as possible but still be able to create and managem templates through OCCI

15:56:52 rnyren: but hang on. templates are just mixins. So you can already create them...

15:57:10 rnyren: sorry for the confusion, been working on too much other stuff of late

15:57:11 ffeldhaus: yeah, but you can't view or change the attributes of a mixin

15:57:25 ffeldhaus: sorry, the values of the attributes

15:57:46 dizz1: good point on the aspect of values assigned to mixins

15:57:53 rnyren: no, the mixin do not have any attributes in itself. It just have the attribute names 15:58:08 ffeldhaus: yes. and this is the problem

15:58:11 dizz1: but for a resource template they will be required

15:58:19 ffeldhaus: no

15:58:30 ffeldhaus: a resource template can just be a mixin with the name "small"

15:58:37 ffeldhaus: everything else is implementation specific

15:58:53 ffeldhaus: and I would like to connect the mixin with existing OCCI entities

15:59:03 dizz1: yes I know that but in the case that I have, I need to offer to clients of what a small means

15:59:28 dizz1: therefore I expose the values of the fixed attributes via the 'attributes' attribute value 15:59:41 rnyren: I don't remember, for user-defined mixins, can you specify the attribute names list then?

15:59:54 rnyren: i.e. POST /my_mixin ...

15:59:59 rnyren: doh, no

16:00:12 ffeldhaus: yes, but I don't think it's a good solution to change the attributes list to also contain the values!

16:00:12 rnyren: POST /-/ Category: ...

16:01:01 rnyren: I have been thinking abouth extending the attributes list in the Category type to include a "default value" in addition to the name

16:01:16 rnyren: that should work with the current definition of kind/mixin etc

16:01:20 dizz1: rnyren: yep that's what I have in effect

16:01:41 dizz1: lemme grab an example rendering I currently have...

16:01:57 rnyren: yes, you are likely to end up implementing an Attribute class which is tied to the Kind/Mixin

16:02:08 ffeldhaus: I would propose the following: we could use the related attribute of a mixin to connect it with existing entities (e.g. compute, storage and network) under /templates

16:02:48 dizz1: Category: small; scheme='<u>http://sla-at-soi.eu/occi/infrastructure/res_template#</u>'; class='mixin'; title='Small Compute Resource Template'; rel='<u>http://schemas.ogf.org/occi/</u>

<u>infrastructure#resource_tpl</u>'; location=/template/small/; attributes='small.occi.compute.cores {immutable int}=1 small.occi.compute.speed{immutable float}=1.0 small.occi.compute.memory {immutable float}=1.0'

16:03:07 dizz1: that's an example of a small template and from which you can tell what you get for that 16:03:18 rnyren: ah, that's a way of rendering it all as well

16:03:42 dizz1: it adds to the ABNF but the additions can be optional

16:03:49 ffeldhaus: but that's not really nice

16:03:49 rnyren: so essentially you have an AttributeDef class with attribs {type, mutable, required,

default_value} right?

16:03:55 dizz1: ya

16:04:12 rnyren: yep, that's the way I do it too

16:04:13 ffeldhaus: mixins extend the attributes of an entity

16:04:13 dizz1: (I've AttributeType and AttributeValue)

16:04:57 ffeldhaus: you end up having entities which can have occi.compute.speed as well as small.occi.compute.speed{immutable float}=1.0

16:05:01 rnyren: I like your HTTP rendering syntax dizz

16:05:18 rnyren: nice to have the type there as well, I have that internally but it is not in the spec so...

16:05:53 dizz1: ffeldhaus: the small.* names is a hack (I've an issue that needs fixing)

16:06:11 dizz1: ffeldhaus: the idea is that the fixed attributes are only attributes that a provider supports

16:06:24 dizz1: and that includes all the specified occi.* ones

16:06:45 dizz1: of course the attributes need to be related to the resource type that the template will instantiate

16:07:12 dizz1: so internally I track what template can be applied to what resource type

16:09:51 ffeldhaus: hm. I'm not convinced I will just finnish my implemenation and try to clarify what the problems are I have with the way you're doing it. Some of them are related to the way OpenNebula is seeing templates and how I can map them between OCCI and OpenNebula 16:10:33 dizz1: sure thing - anyways I'm sure there'll be more opinions and views once we get this going for v1.2 via the mailing list and here

16:10:41 rnyren: sounds like a plan although the small.xxx thingy looks a bit, well not perfect Ӱ

16:10:53 dizz1: it's not Ӱ

16:11:22 ffeldhaus: one more thing

16:11:30 ffeldhaus: what are you doing with backups and snapshots?

16:11:53 dizz1: how do you mean?

16:12:01 rnyren: how you access then through occi?

16:12:03 dizz1: as in supplying additional parameters or?

16:12:05 ffeldhaus: yes

16:12:07 dizz1: ah

16:12:16 dizz1: that's not specified

16:12:20 ffeldhaus: after you did a snapshot, how are you supposed to find it?

16:12:29 dizz1: that's a good point

16:12:34 rnyren: _very_ good question 💛

16:12:42 ffeldhaus: is it a good idea to use something like /snapshots/123 for this?

16:13:08 ffeldhaus: and 123 being a new Storage entity linked to the original one?

16:13:28 rnyren: I guess what you could do is define a new Kind which is a storage-snapshot-Kind which is located at /snapshot (if you like)

16:13:40 rnyren: yes, you add all the Links automatically

16:14:30 dizz1: it'd be cool if we could express sub-resources of a resource

16:14:32 rnyren: Category: snapshot; scheme=http://schemas.example.com/occi#; location=/

snapshot; attributes=occi.storage.size

16:14:54 rnyren: yap, very cool.

16:15:16 dizz1: that would require us expressing some sort of wild card in the location specification

16:15:48 rnyren: ah, yes

16:15:55 dizz1: e.g. /storage/%ID1/snapshot/%ID2

16:16:08 rnyren: Category: snapshot; location=/storage/.../snapshot/...

16:16:09 dizz1: oh

16:16:10 rnyren: exactly

16:16:20 dizz1: actually, we can use a link, right?

16:16:26 rnyren: yes

16:16:51 rnyren: it can be done with links, no problem. But it can be a bit bothersome to manange all the links

16:16:54 ffeldhaus: only the implementation needs to know, that the linked resource is a snapshot and not a normal storage entity

16:17:19 dizz1: you can have the type information in the link afaik

16:17:52 rnyren: Think of implementing a cloud based blog service using occi:

16:18:02 rnyren: you'd have lots of blogs with posts with comments

16:18:23 rnyren: to add a comment you'd like to do POST /myblog/post1/comments/

16:18:43 rnyren: but with OCCI you'd have to go through the process of creating all the links to tie the flat namespace together

16:19:20 rnvren; can be done but a bit tricky from the client perspective

16:20:10 rnyren: in that case sub-resources would be super-cool 💛

16:20:31 dizz1: def

16:20:46 dizz1: i think it'd be worth a few chats here on that - see what could be done

16:21:09 rnyren: yeah, that'd be lots of fun 😇

16:21:32 rnyren: however, got a run now. I can join a bit tomorrow if you'd like to chat some more 16:21:37 rnyren: have fun!

16:22:27 ffeldhaus: see you tomorrow

16:22:57 dizz1: thanks ralf! Ӵ

16:23:28 ffeldhaus: I also need to leave now, my students are waiting... I'll drop in tomorrow and would be glad to further discuss todays topics

16:24:26 dizz1: cool! thanks for dropping by Florian!

16:24:42 dizz1: btw: do you know anyone in the FP7 project 'bonfire'?

rnyren left the room (quit: Quit: Page closed). (16:25:48)

16:27:26 ffeldhaus: I don't think I know anyone

16:28:06 ffeldhaus: I will get to know people from the contrail project next week, but I can't remember a contact from bonfire

16:28:09 ffeldhaus: what are they doing?

16:28:44 dizz1: they're running a large testbed (main focus of the project) and I found references to OCCI being used there

16:28:50 dizz1: must keep digging 💛

16:28:58 dizz1: ah yes - contrail, I'll be on that call too

16:29:38 dizz1: I think it was their coordinator (yves jegon) that was at the DMTF meeting

16:30:25 ffeldhaus: I think it's very important that we have a usable testbed with an actual Cloud

Manager behind

16:30:34 ffeldhaus: we try to make OpenNebula ready for it

16:30:57 dizz1: yup - absolutely

16:31:06 dizz1: open nebula is very popular in FP projects

16:31:10 ffeldhaus: in the last Cloud Plugfest call, there was the urge to move forward with a demo for **OCCI 1.1**

16:32:06 ffeldhaus: ah. one thing I forgot: we will add basic CDMI support for OpenNebula next week

 $\stackrel{\textcircled{}_{\scriptstyle \bigcirc}}{\underset{\scriptstyle \bigcirc}{\scriptstyle \bigcirc}}$ It's work in an early stage, but it enables us to use CDMI together with OCCI on top of OpenNebula

16:32:28 dizz1: nice - macredcape will be glad to know of another CDMI implementation 😇



16:34:21 ffeldhaus: who's behind macredcape? Garry Mazzaferro? 16:34:44 dizz1: naw - macredcape is... drum roll...

16:34:47 macredcape: HI Florian - This is Mark Carlson

16:34:54 ffeldhaus: ah

16:35:44 ffeldhaus: good to know

16:36:02 ffeldhaus: ok, I really need to go, I hope the students are still there...

16:36:07 ffeldhaus: see you tomorrow

16:36:24 dizz1: thanks Florian!