* *8:59:58 PM*

**joehuang**

#info I found saggi has comment on my comment. The first one is the VM status. API interception and status cache is a good idea. How long for the cache time.

* *9:00:48 PM*

**saggi**

joehuang: Depends on what is happening. gampel suggested we should also invalidate cache on certain commands.

* *9:00:58 PM*

**saggi**

like when VM creation ends invalidate the cache

* *9:01:14 PM*

**joehuang**

#info And one question for the cache is , if multiple VM be queried at the same time, can we fresh the status cache one by one or with one query api to cascaded Nova

* *9:01:48 PM*

**saggi**

IMHO cache refresh should always be done in the background.

* *9:02:04 PM*

**saggi**

Even if we invoked a refresh we should return stale data until we have an update or time out

* *9:02:24 PM*

**saggi**

in the case of timeout we should update status to something the makes it clear to the use that the status is unknown

* *9:02:29 PM*

**joehuang**

so there is one periodic polling task at the background to sync status

* *9:02:39 PM*

**gampel**

we will pass runtime queries to the cascading service as saggi said and maybe we could have few refresh options on trigger periodic etc

* *9:02:40 PM*

**saggi**

joehuang: yes

* *9:02:59 PM*

**saggi**

joehuang: Should be configurable through the cascade API

* *9:03:22 PM*

**saggi**

Since it's site related

* *9:03:37 PM*

**joehuang**

Could be

* *9:04:05 PM*

**irenab**

saggi: what is the data model managed by CascadingService?

* *9:04:25 PM*

**saggi**

I don't understand the question :)

* *9:04:37 PM*

**saggi**

what entities does it manage?

* *9:04:51 PM*

**gampel**

we are in th process of defining the DB for the cascaded layer

* *9:05:09 PM*

**irenab**

Do you mean cascading?

* *9:05:14 PM*

**gampel**

it should have the entities mapping

* *9:05:20 PM*

**saggi**

in general it should contain mapping data (global UUID to site UUIDs) and site related information.

* *9:05:32 PM*

**gampel**

yes cascaded sorry the TOP layer

* *9:05:53 PM*

**irenab**

ok. When ready, please add it tho the doc

* *9:06:08 PM*

**joehuang**

the VM status polling should be carefully if some long lasting task is executing, for example, volume migration

* *9:06:26 PM*

**saggi**

that is handled by the Task Exec

* *9:06:36 PM*

**irenab**

and what enitites will get UUID mappings

* *9:06:43 PM*

**saggi**

it will poll in appropriate interval while a specific task is running.

* *9:07:00 PM*

**saggi**

It's also true for starting a VM or creating a network.

* *9:07:25 PM*

**gampel**

entities that we can not set their GUID on the bottom (cascaded) layer

* *9:07:44 PM*

**zhipeng**

shall we just rephrase to top and bottom layer ?

* *9:07:51 PM*

**saggi**

+1

* *9:07:52 PM*

**irenab**

yes :-)

* *9:07:55 PM*

**zhipeng**

it would be more easier to remember

* *9:07:56 PM*

**joehuang**

If we keep state machine in the cascading layer, be sure the polling task will not harm the state machine in the task

* *9:08:01 PM*

**gampel**

+1

* *9:08:13 PM*

**joehuang**

+1

* *9:08:27 PM*

**zhipeng**

#info rephrase cascading/cascaded to Top/Bottom

* *9:09:04 PM*

**irenab**

gampel: I guess it will be most of the entities, since majority generate uuid du ring resource creation

* *9:09:30 PM*

**gampel**

yes unfortunately

* *9:09:59 PM*

**joehuang**

vm, volume, backup, snapshot, even flavor, network,subnet,port, router...

* *9:10:11 PM*

**saggi**

yep

* *9:10:54 PM*

**gampel**

Joe: you raised a requirement that is not supported in this design to be able to control the bottom from horizon on local site

* *9:11:55 PM*

**irenab**

gampel: why to restrict it?

* *9:11:58 PM*

**saggi**

joehuang: Could you elaborate on the use cases.

* *9:12:14 PM*

**joehuang**

yes. In OPNFV multisite project, the local APP manager need to be able to provision app even other sites failed

* *9:12:43 PM*

**gampel**

But will he need to create resources ?

* *9:13:07 PM*

**joehuang**

yes, create new VM.volume at least

* *9:13:25 PM*

**gampel**

Not network ?

* *9:13:33 PM*

**saggi**

joehuang: But what if he is missing the networks. Current configuration is reactive.

* *9:13:38 PM*

**joehuang**

These two scenario can work independently, but not together

* *9:13:55 PM*

**saggi**

What two scenarios?

* *9:13:57 PM*

**joehuang**

Network too if needed

* *9:14:28 PM*

**gampel**

network is a shared resource and this make it more difficult to sync to all sites

* *9:14:56 PM*

**joehuang**

In the past, we often provide one scenario: multisite with one global API service which is provided by the top layer

* *9:15:50 PM*

**joehuang**

This is what has been reflected in the design doc, all resource provision request comes from the top layer

* *9:16:07 PM*

**saggi**

joehuang: Sow what comes from the bottom layer

* *9:16:13 PM*

**saggi**

sow=so

* *9:16:43 PM*

**joehuang**

On the other hand, another scenario, don't want the single top API layer

* *9:17:39 PM*

**joehuang**

but still need the centralized service to provide cross neutron networking, image replication, security group replication, from one site to another site

* *9:17:53 PM*

**saggi**

joehuang: Without the top layer all the algorithms need to be built as distributed.

* *9:18:01 PM*

**saggi**

hmmm

* *9:18:03 PM*

**gampel**

the problem will be to sync the change made in one site horizontally to all other sites

* *9:18:50 PM*

**gampel**

the current design was built with one TOP API in mind

* *9:19:12 PM*

**joehuang**

to gampel: correct, the Nova/cinder/neutron API can still be called in each site seperatly

* *9:19:25 PM*

**saggi**

joehuang: What about just making the Top layer as distributed. You could just install it on every site.

* *9:19:46 PM*

**joehuang**

only if some cross OpenStack function, will issue api calling to the centralized service

* *9:20:01 PM*

**gampel**

It's only possib

* *9:20:23 PM*

**joehuang**

so I suggest to make the newly introduced service to work in these two scenario

* *9:20:26 PM*

**gampel**

it is only reasonable if we introduce some constraints on which resource creation is allowed

* *9:20:54 PM*

**saggi**

joehuang: If we make the Top layer able to work on multiple sites. Using distributed database.

* *9:21:14 PM*

**gampel**

As far as we understand, getting it accepted by the openstack community is topmost priority

* *9:21:17 PM*

**gampel**

is this correct?

* *9:21:34 PM*

**joehuang**

yes

* *9:21:45 PM*

**gampel**

in that case, we will not be able to comply with what you're saying at the moment

* *9:21:57 PM*

**gampel**

you need to understand that NFV and Openstack use cases are very different

* *9:22:02 PM*

**gampel**

and in many cases collide

* *9:22:22 PM*

**gampel**

the OPNFV community does not understand this and that is why they face a lot of problems when coming to openstack

* *9:22:40 PM*

**gampel**

introducing changes into the underlying openstack will be rejected

* *9:22:48 PM*

**gampel**

we need to take a step back here

* *9:23:04 PM*

**gampel**

clearly define the requirements and then think clearly what can be done

* *9:23:10 PM*

**gampel**

we do not design on irc

* *9:23:12 PM*

**joehuang**

for sure. no change to the underlyingOpenStck

* *9:23:34 PM*

**gampel**

you cannot keep the underlying openstack unchanged and yet sync changes back to the top

* *9:24:23 PM*

**irenab**

maybe more advanced case, can be get bottom resources and allow to match them top one

* *9:24:40 PM*

**saggi**

We don't want bottom being aware of TOP

* *9:24:46 PM*

**gampel**

I suggest we just focus on the different use cases now, list the requirements and then see how to resolve the conflicts between openstack cloud and NFV

* *9:24:51 PM*

**joehuang**

the second scenario does not introduce change to the bottom openstack

* *9:25:09 PM*

**irenab**

saggi: agree, thas why top will retrive the bottom, and user will match

* *9:25:35 PM*

**joehuang**

absolutely "We don't want bottom being aware of TOP"

* *9:26:08 PM*

**irenab**

I think it makes sense to add functional requirements section in the doc. It starts directly with design principles.

* *9:26:33 PM*

**irenab**

So maybe few user stories will clarify the scope

* *9:26:38 PM*

**gampel**

that will be very hard to sync by pool mode , i think that we should limit the resource creation

* *9:26:46 PM*

**gampel**

irenab: i agree

* *9:27:02 PM*

**joehuang**

make sense to describe the requirement in the doc

* *9:27:18 PM*

**zhipeng**

I agree that certain constraints should be placed

* *9:27:24 PM*

**joehuang**

what is "limit the resource creation"

* *9:27:27 PM*

**gampel**

OK i can add you both as author

* *9:27:37 PM*

**saggi**

I also wouldn't like the user to directly access the bottom if it's being managed.

* *9:27:58 PM*

**zhipeng**

not all scenarios of creation should necessarily be needed in Tricircle

* *9:28:01 PM*

**zhipeng**

joehuang

* *9:28:26 PM*

**joehuang**

Ok, I will also describe the second scenario in more detail

* *9:28:26 PM*

**gampel**

if we do not allow net create on the bottom layer just start stop create vm

* *9:28:52 PM*

**saggi**

if it's managed we want to have full control or it will be hard to make sense of a host's state on error flows

* *9:29:49 PM*

**gampel**

maybe we could split the work

* *9:30:02 PM*

**gampel**

on the design

* *9:30:43 PM*

**joehuang**

we can limit the content in the first stage, and make it work ASAP

* *9:30:44 PM*

**gampel**

we need use cases , requirements , and architect design

* *9:30:52 PM*

**saggi**

I would like to have the use cases written up

* *9:30:57 PM*

**saggi**

gampel: +1

* *9:31:05 PM*

**zhipeng**

+1 good idea

* *9:31:22 PM*

**joehuang**

+1

* *9:31:59 PM*

**gampel**

saggi and myself can go deeper on the POC and design building blocks

* *9:32:35 PM*

**zhipeng**

I would participate in the use case and requirements

* *9:33:11 PM*

**zhiyuan**

me, too. use case and requirements

* *9:33:51 PM*

**zhipeng**

we could use your observations as well irenab

* *9:33:51 PM*

**gampel**

joehuang: i think that you are the most knowledge on the use cases

* *9:34:01 PM*

**joehuang**

ok, I'll also working on use case/ requirements/design part. The doc is already a good base for design

* *9:34:32 PM*

**gampel**

I will add you so you could edit the document

* *9:35:03 PM*

**joehuang**

Can we co-work on the google doc just like etherpad?

* *9:35:08 PM*

**gampel**

yes

* *9:35:34 PM*

**zhipeng**

google doc is etherpad on steroid

* *9:35:43 PM*

**gampel**

another topic is the l2-gw

* *9:36:22 PM*

**gampel**

we started looking at the API for the networking-l2gw and it need to be modified a bit

* *9:36:30 PM*

**joehuang**

Good. I think the major challenge is that we need to support the second scenario or not , or step by step

* *9:36:53 PM*

**joehuang**

L2GW currently is for Neutron inside to outside

* *9:37:10 PM*

**joehuang**

but not for cross -neutron

* *9:37:27 PM*

**gampel**

it currently focused on another use case connect to HW

* *9:37:37 PM*

**zhipeng**

#agreed work split on design doc drafting, gampel and saggi on designing blocks, zhiyuan and zhipeng on use case/req, and joe on overall enhancement

* *9:37:40 PM*

**joehuang**

Agree

* *9:38:13 PM*

**saggi**

+1

* *9:38:24 PM*

**gampel**

agree we need to split the req to openstack and Opnfv

* *9:38:26 PM*

**zhiyuan**

+1

* *9:38:27 PM*

**gampel**

+1

* *9:38:49 PM*

**zhipeng**

#agreed seperate req in opnfv from openstack

* *9:38:58 PM*

**joehuang**

We need enhancement on the L2GW api, but L2GW api itself is not merged into the trunk yet

* *9:39:17 PM*

**zhipeng**

that was supposed to be networking-tricircle right?

* *9:39:48 PM*

**joehuang**

+1

* *9:40:32 PM*

**zhipeng**

gampel joehuang, Kyle just post new admin rules for networking-\* repos, dun know if we would be affected

* *9:40:49 PM*

**zhipeng**

should be nothing but tagging

* *9:41:14 PM*

**zhipeng**

we should check it out to make sure tho

* *9:41:42 PM*

**gampel**

we want to use networking-l2gw but i am not sure if they will agree to support site to site tunnel creation we will try to start talking with them

* *9:42:23 PM*

**joehuang**

that's important to have site 2 site suport on L2GW

* *9:42:49 PM*

**joehuang**

To zhipeng, what's the new rule

* *9:43:32 PM*

**zhipeng**

joehuang not rule per se, check it out in the mailing list

* *9:43:52 PM*

**joehuang**

ok'

* *9:46:18 PM*

**joehuang**

so let's continue work on the doc and have discussion in M-L?

* *9:47:13 PM*

**saggi**

sure

* *9:47:16 PM*

**gampel**

OK i it will be great to discuss offline your idea for local site control

* *9:48:02 PM*

**zhipeng**

we should always use ml as frequent as possible

* *9:48:10 PM*

**joehuang**

yes. maybe I did not describe it very clear. in fact, the centralized service should have no function overlapping with each site.

* *9:48:25 PM*

**joehuang**

agree