* *9:09:56 PM*
* **saggi**
* So you want us to just ask the bottom KeyStone?
* *9:10:00 PM*
* **saggi**
* instead of storing it in the DB?
* *9:10:05 PM*
* **gampel**
* can you please explain what is stored in KeyStone I am not sure i follow
* *9:10:08 PM*
* **joehuang**
* So we may only store region info in the cascade service, and ask keystone if required
* *9:10:27 PM*
* **saggi**
* joehuang: What region info?
* *9:11:04 PM*
* **joehuang**
* there is one table in KeyStone,called region, one OpenStack instance one region
* *9:11:42 PM*
* **joehuang**
* and each region will include service like Nova,cinder, neutron, glance, ceilometer, or other service
* *9:11:59 PM*
* **gampel**
* can we get from Keystone the urls to access a region
* *9:12:08 PM*
* **joehuang**
* each service has endpoint like pulic/internal/admin url
* *9:12:20 PM*
* **joehuang**
* hi gampel, you got my idea
* *9:12:39 PM*
* **joehuang**
* we simply store region, and cache the endponit url
* *9:13:02 PM*
* **joehuang**
* if the cached url can't be accessed( timeout), then refresh from keystone
* *9:13:07 PM*
* **gampel**
* So we could store the assess URL for the bottom stack in Ketstone
* *9:13:23 PM*
* **joehuang**
* KeyStone already stores that
* *9:13:47 PM*
* **saggi**
* joehuang: It's odd that Keystone already has that feature
* *9:13:56 PM*
* **saggi**
* joehuang: What is it being used for?
* *9:14:56 PM*
* **joehuang**
* when the user has be authenticated, the user can retrieve the endponit list, and select one region to access
* *9:15:56 PM*
* **joehuang**
* hello, zhipeng, welcome back. Hope your babies is sleeping ;)
* *9:16:31 PM*
* **zhipeng**
* joehuang thx they are sleeping like babies :P
* *9:16:32 PM*
* **saggi**
* The cascade service will need to be able to access the information for all regions.
* *9:16:41 PM*
* **saggi**
* zhipeng: :)
* *9:16:56 PM*
* **saggi**
* joehuang: Is it possible to query for multiple regions?
* *9:17:12 PM*
* **joehuang**
* sure, keystone 's ability
* *9:18:01 PM*
* **gampel**
* So you mean when an admin add a Cascaded site we will configure new region in keystone and store all the information
* *9:18:01 PM*
* **joehuang**
* for cascade service, we can provide a way which regions will be managed by the cascade service
* *9:18:24 PM*
* **saggi**
* joehuang: So for each region there are multiple tenants or is that orthogonal?
* *9:18:36 PM*
* **joehuang**
* yes, the admin first manage these region and endpoints in keystone
* *9:18:57 PM*
* **joehuang**
* each tenenat may cross several regions
* *9:19:12 PM*
* **joehuang**
* any region may support multi=tenancy
* *9:19:22 PM*
* **gampel**
* But we want to create One Cascading API to mange adding new sites
* *9:19:51 PM*
* **gampel**
* is it possible to create the region in Keystone from the cascading openstack service
* *9:20:16 PM*
* **joehuang**
* no need to do this.
* *9:20:26 PM*
* **joehuang**
* keystone is for this
* *9:20:53 PM*
* **joehuang**
* but cascade service need to know which regions the cascade service will cover
* *9:21:06 PM*
* **saggi**
* joehuang: And generate appropriate AZ information
* *9:21:07 PM*
* **joehuang**
* this configuration APi is needed
* *9:21:44 PM*
* **gampel**
* How do you configure regions in keystone ?
* *9:21:44 PM*
* **joehuang**
* AZ infomation should also be configured in the bottom OpenStack
* *9:21:53 PM*
* **joehuang**
* and retrieved to cascade service
* *9:23:47 PM*
* **saggi**
* joehuang: I wanted to talk about that. I was wondering why not just expose all the compute node and create a logical immutable AZ for them. Do you follow?
* *9:24:10 PM*
* **joehuang**
* #link<http://docs.openstack.org/cli-reference/content/keystoneclient_commands.html>
* *9:24:17 PM*
* **gampel**
* Let first action item this topic I suggest that we study this a bit and if make sense update the doc and change the db schema
* *9:24:34 PM*
* **saggi**
* gampel: +1
* *9:25:23 PM*
* **joehuang**
* to saggi, you mean nodes in bottom openstack or top layer openstack?
* *9:25:32 PM*
* **joehuang**
* agree
* *9:25:34 PM*
* **saggi**
* bottom layer
* *9:25:41 PM*
* **saggi**
* joehuang: ^^
* *9:25:48 PM*
* **saggi**
* to expose them as is in the top layer
* *9:26:09 PM*
* **saggi**
* With an automatic AZ to be able to lock scheduling to a specific site.
* *9:26:10 PM*
* **joehuang**
* yes, it's possible in new design
* *9:26:36 PM*
* **saggi**
* joehuang: great
* *9:26:42 PM*
* **joehuang**
* because all nodes in bottom openstack will be grouped into AZs
* *9:26:49 PM*
* **saggi**
* yes
* *9:26:52 PM*
* **gampel**
* and then we will create default AZ for site but allow creating iner AZ
* *9:26:58 PM*
* **joehuang**
* so we can get AZ info from all bottom openstack
* *9:27:02 PM*
* **zhipeng**
* would there be two layers of AZs??
* *9:27:21 PM*
* **saggi**
* zhipeng: not layers. Just multiple AZ. Like a ven diagram.
* *9:27:30 PM*
* **joehuang**
* there are should be one interal AZ in top layer
* *9:27:42 PM*
* **gampel**
* no same layer because you could have one VM in multiple AZs
* *9:28:08 PM*
* **zhipeng**
* got it
* *9:28:34 PM*
* **joehuang**
* how about one action to redesign the DB shcema?
* *9:28:56 PM*
* **saggi**
* joehuang: I don't follow
* *9:28:58 PM*
* **joehuang**
* according to info in keystone and bottom opensrack
* *9:29:14 PM*
* **saggi**
* joehuang: Oh, action item. I get it.
* *9:29:19 PM*
* **gampel**
* Yes lets look a it a bit I suggest that we study this a bit and if make sense update the doc and change the db schema
* *9:29:51 PM*
* **saggi**
* gampel, joehuang: It should contain the endpoint cache at the very least
* *9:30:02 PM*
* **joehuang**
* tables in the google doc store a lot dupplicate info, a bit hard to maintain if somthing changed in keystone or bottom openstack
* *9:30:04 PM*
* **gampel**
* thats true
* *9:30:08 PM*
* **joehuang**
* yes
* *9:30:16 PM*
* **joehuang**
* I mentioned cache.
* *9:30:48 PM*
* **zhipeng**
* #action study info on keystone and update the db schema design
* *9:31:18 PM*
* **joehuang**
* thanks. zhipeng. I remove what I typed :)
* *9:31:35 PM*
* **zhipeng**
* joehuang :P
* *9:31:50 PM*
* **joehuang**
* #topic BP
* *9:31:51 PM*
* openstack changed the topic to: BP (Meeting topic: tricircle)
* *9:32:09 PM*
* **joehuang**
* we have a very good start for the source code
* *9:32:45 PM*
* **joehuang**
* we need a BP to link the first patch, frame work, plugin
* *9:33:21 PM*
* **joehuang**
* so that peoples are easy to be involved in
* *9:33:59 PM*
* **zhipeng**
* do we need one BP to cover all that? Or BP per patch?
* *9:34:33 PM*
* **gampel**
* BP link please review <https://blueprints.launchpad.net/tricircle/+spec/new-design>
* *9:35:03 PM*
* **zhipeng**
* #link<https://blueprints.launchpad.net/tricircle/+spec/new-design>
* *9:35:55 PM*
* **joehuang**
* great
* *9:36:17 PM*
* **gampel**
* can we push it to master branch ?
* *9:36:20 PM*
* **joehuang**
* we can shares the bp in the mail-list, so it will be visible for all people.
* *9:36:41 PM*
* **joehuang**
* ok. please merge on master branch
* *9:37:22 PM*
* **joehuang**
* and remove all old code. Zhiyuan has already put a tag on the master.
* *9:37:29 PM*
* **saggi**
* joehuang: Do you want to merge after committing to "experiment" or do you want to close experiment and have me resend it based on "master"?
* *9:37:51 PM*
* **saggi**
* OK I'll add a rearrange the patches.
* *9:38:05 PM*
* **saggi**
* And clean up the code base.
* *9:38:11 PM*
* **joehuang**
* how about nova patch
* *9:38:26 PM*
* **joehuang**
* nova patch in Nova-API
* *9:38:59 PM*
* **joehuang**
* sorry that I was on business trip, not review so much
* *9:39:07 PM*
* **saggi**
* joehuang: It's in progress. It's a very complex API.
* *9:39:36 PM*
* **joehuang**
* for monday and tuesday
* *9:39:54 PM*
* **joehuang**
* to saggi, you mean boot-vm?
* *9:40:12 PM*
* **joehuang**
* this is the complex one indeed
* *9:40:23 PM*
* **gampel**
* we could collaborate on the other cascading service building blocks
* *9:40:31 PM*
* **joehuang**
* sure
* *9:40:40 PM*
* **saggi**
* joehuang: For now I'm trying to get a skeleton up so we could start collaborating.
* *9:40:59 PM*
* **joehuang**
* understand
* *9:41:39 PM*
* **joehuang**
* I talked with core member Lingxian of Mistral
* *9:41:56 PM*
* **gampel**
* Yes what does he think
* *9:42:07 PM*
* **joehuang**
* Mistral doesn't provide lib, but work as standalone serice
* *9:42:43 PM*
* **joehuang**
* that means can only call restful API or use its python client (the client to call restful API)
* *9:42:44 PM*
* **gampel**
* so using rest api
* *9:43:13 PM*
* **joehuang**
* it'll be great challenge
* *9:44:00 PM*
* **joehuang**
* add one more restful API calling hop
* *9:44:17 PM*
* **joehuang**
* some extra complexity will introduce
* *9:44:32 PM*
* **gampel**
* if we will do run time query caching an way , this is not a big problem
* *9:44:37 PM*
* **joehuang**
* especiall for status refresh. etc
* *9:45:16 PM*
* **joehuang**
* we need a BP and detail infor about run time query
* *9:45:47 PM*
* **saggi**
* joehuang: Did you change the topic? Also add an action item for that BP
* *9:45:55 PM*
* **joehuang**
* There is a lot of chllenge we found for status
* *9:46:12 PM*
* **saggi**
* joehuang: We could bypass minstral for status
* *9:46:21 PM*
* **saggi**
* since it doesn't require any transaction
* *9:47:14 PM*
* **joehuang**
* could you please share your ideas how to handle the status
* *9:47:31 PM*
* **gampel**
* we need to add to the design the way we handle run time query
* *9:47:33 PM*
* **joehuang**
* especially if a VM is in live-migration, or migrate volume
* *9:49:00 PM*
* **joehuang**
* if migration task is running in mistral
* *9:49:22 PM*
* **gampel**
* live migration is not a relevant use case in my opinion
* *9:49:51 PM*
* **saggi**
* We could define a caching policy per VM or VM group. That is good for high latency situations. When we detect a low latency situation we will start refreshing the cache in shorter intervals. Whatever the case the user always get the information for the cache.
* *9:50:02 PM*
* **joehuang**
* so I am wondering to see if the ideas work for status uery
* *9:50:21 PM*
* **saggi**
* Status query should bypass minstrall IMHO
* *9:50:46 PM*
* **saggi**
* I don't think they even have that use case.
* *9:50:54 PM*
* **saggi**
* Since it's not a real 'task'
* *9:51:12 PM*
* **joehuang**
* where to define the policy, in cascade service?
* *9:51:21 PM*
* **saggi**
* joehuang: Yes
* *9:51:41 PM*
* **joehuang**
* there are multi-worker multi-processes
* *9:52:01 PM*
* **joehuang**
* so which one will do the cacahe, cache in memory or in DB
* *9:52:42 PM*
* **saggi**
* joehuang: Depends on scale. We might need to use a caching service like redis.
* *9:53:33 PM*
* **saggi**
* joehuang: It's not information we care about persisting. But I suspect that for a huge amount of nodes it will get to big to keep in memory.
* *9:53:41 PM*
* **joehuang**
* so which cascade service will be responisble for which VM/VM group cacahed status refreshment
* *9:54:09 PM*
* **saggi**
* joehuang: Sharding, either by site or by VM\_ID % cascade\_service
* *9:54:17 PM*
* **gampel**
* I suggest that we will add an action item to add design for cache and task exec and then discuss if it make sense
* *9:54:53 PM*
* **joehuang**
* agree
* *9:54:58 PM*
* **zhiyuan**
* +1
* *9:55:07 PM*
* **zhipeng**
* +1
* *9:55:14 PM*
* **saggi**
* +1
* *9:55:16 PM*
* **joehuang**
* status is very important.
* *9:55:41 PM*
* **gampel**
* I think that we need to fix our CI to include pep8 testing
* *9:55:58 PM*
* **saggi**
* gampel: Or I can keep fixing your pep8 errors :)
* *9:55:59 PM*
* *9:56:09 PM*
* **gampel**
* :-D
* *9:56:41 PM*
* **joehuang**
* ci included pep8, is n't it? Zhiyuan
* *9:57:16 PM*
* **saggi**
* We need to add tox configuration
* *9:57:20 PM*
* **saggi**
* for it to run
* *9:57:21 PM*
* **zhiyuan**
* not setup yet. I am coding for DAL these days. But enabling pep8 is easy
* *9:58:03 PM*
* **zhipeng**
* what is DAL?
* *9:58:18 PM*
* **zhiyuan**
* just add a script to run flake8 in our third party CI
* **joehuang**
* #action design for cache and task exec and then discuss
* *9:59:10 PM*
* **saggi**
* zhipeng: Data Access Layer - And abstraction of all the data sources in the top layer.
* *9:59:27 PM*
* **saggi**
* zhipeng: The cascade service DB the nova DB the neutron DB etc.
* *9:59:32 PM*
* **zhipeng**
* saggi thx :)
* *10:00:47 PM*
* **joehuang**
* let's have a short summary
* *10:00:54 PM*
* **zhiyuan**
* I will take some time to enable pep8 tomorrow
* *10:01:18 PM*
* **saggi**
* zhiyuan: great
* *10:01:49 PM*
* **joehuang**
* #action pep8 enable in ci
* *10:02:16 PM*
* **joehuang**
* 3 action, is there any more, or something missed?
* *10:03:57 PM*
* **zhipeng**
* should be ok for now joehuang
* *10:04:04 PM*
* **zhipeng**
* let's wrap up for the day :)
* *10:04:16 PM*
* **joehuang**
* pls.
* *10:04:59 PM*
* **joehuang**
* I am not family to use the irc
* *10:05:07 PM*
* **zhipeng**
* just #endmeeting
* *10:05:10 PM*
* **gampel**
* thank you
* *10:05:16 PM*
* **zhipeng**
* but you should be the one typing it
* *10:05:18 PM*
* **joehuang**
* ok ,thank you all
* *10:05:26 PM*
* **gampel**
* bye
* *10:05:28 PM*
* **joehuang**
* #endmeeting
* *10:05:29 PM*
* openstack changed the topic to: zuul jobs are currently stuck while problems with gearman are debugged.
* *10:05:30 PM*
* **openstack**
* Meeting ended Wed Jul 22 14:06:03 2015 UTC. Information about MeetBot at<http://wiki.debian.org/MeetBot> . (v 0.1.4)
* *10:05:31 PM*
* **openstack**
* Minutes: <http://eavesdrop.openstack.org/meetings/tricircle/2015/tricircle.2015-07-22-13.05.html>
* *10:05:32 PM*
* **zhiyuan**
* bye
* *10:05:32 PM*
* **openstack**
* Minutes (text):<http://eavesdrop.openstack.org/meetings/tricircle/2015/tricircle.2015-07-22-13.05.txt>
* *10:05:33 PM*
* **joehuang**
* bye
* *10:05:34 PM*
* **saggi**
* bye
* *10:05:34 PM*
* **openstack**
* Log: <http://eavesdrop.openstack.org/meetings/tricircle/2015/tricircle.2015-07-22-13.05.log.html>