this document shows how to use openstack python sdk to create and delete VMs

import modules

```
>>> import keystoneclient.v2_0.client as ksclient
>>> import novaclient.client as nvclient
>>> import glanceclient.v2.client as glclient
```

authenticate keystone

```
>>> keystone = ksclient.Client(auth_url="http://192.168.1.77:5000/v2.0",username = "admin",password = "stack",tenant_name="demo
>>> print(keystone.auth_token)
116b8475df6c4d31855d694a88d3ed01
>>> keystone = ksclient.Client(auth_url="http://192.168.1.77:5000/v2.0",username = "demo",password = "stack",tenant_name="demo"
>>> print(keystone.auth_token)
7fe5ed6e522647b99be5dd61572b2805
```

note that different users has different authenticate tokens

then start glance client to get the inventory of all available images

```
>>> glance_endpoint = keystone.service_catalog.url_for(service_type="image")
>>> glance=glclient.Client(glance_endpoint,token=keystone.auth_token)
>>> imgs=glance.images.list()
```

now list the image names

```
>>> for i in imgs:
...     print(i.name)
...
cirros-0.3.2-x86_64-uec
cirros-0.3.2-x86_64-uec-ramdisk
cirros-0.3.2-x86_64-uec-kernel
```

set up nova client

```
>>> nova_client = nvclient.Client(auth_url="http://192.168.1.77:5000/v2.0",username="demo",api_key="stack",project_id="demo",version='2')
>>> print(nova_client.servers.list())
[]
```

by now, the user “demo” has not run any VM yet, so the list of VMs is empty

before creating instance, make proper configurations

```
>>> image = nova_client.images.find(name="cirros-0.3.2-x86_64-uec")
>>> print(image)
<Image: cirros-0.3.2-x86_64-uec>
>>> flavor=nova_client.flavors.find(name="m1.tiny")
>>> print(flavor)
<Flavor: m1.tiny>
>>> net=nova_client.networks.find(label="private")
>>> print(net)
<Network: private>
>>> print(net.id)
ad47f55e-cf3c-47fc-897a-05e7657b9cbf
>>> nics=[{'net-id':net.id}]
```

choose image,flavor and network ID

then create instance

```
>>> instance = nova_client.servers.create(name="ins_01",image=image,flavor=flavor,nics=nics)
>>> print(nova_client.servers.list())
[<Server: ins_01>]
```

now we have a new instance in the list
we can create another one with same configuration except a different name

```
>>> instance = nova_client.servers.create(name="ins_02",image=image,flavor=flavor,nics=nics)
>>> print(nova_client.servers.list())
[<Server: ins_02>, <Server: ins_01>]
```

now we have two instances

to delete one, we have to find the instance by its name

```
>>> for vm in nova_client.servers.list():
...     if vm.name == "ins01":
...             break
...
>>> vm.name
u'ins_01'
```

then delete from nova client

```
>>> nova_client.servers.delete(vm)
>>> print(nova_client.servers.list())
[<Server: ins_02>]
```

as ins_01 has been deleted, there is only ins_02 in the list

from the webUI, we can also see this instance