



**Neopath Integrated Systems**

**Desenvolvimento em ambiente *Cloud***

**Beatriz de Andrade Campos**  
**[beatriz.campos@neopath.com.br](mailto:beatriz.campos@neopath.com.br)**

1. Vantagens no desenvolvimento para *Clouds*.
2. Vantagens no uso do Openstack.
3. Utilização da biblioteca Shade:
  - Criação de instâncias.
  - Criação de *storage* de objetos.
  - *Deploy* de aplicações.

- **Escalabilidade:** habilidade do sistema de lidar com o aumento do gerenciamento e manipulação de dados.
- **Elasticidade:** habilidade do sistema de se adaptar ao aumento e a diminuição de demanda.
- **Disponibilidade:** habilidade do sistema de estar disponível em qualquer circunstância.

- ***Open source.***
  - Documentação e comunidade.



- **Modularidade.**



- **Modularidade.**
  - Adequação a necessidades.



**CINDER**

*an OpenStack Community Project*



**NEUTRON**

*an OpenStack Community Project*



**NOVA**

*an OpenStack Community Project*



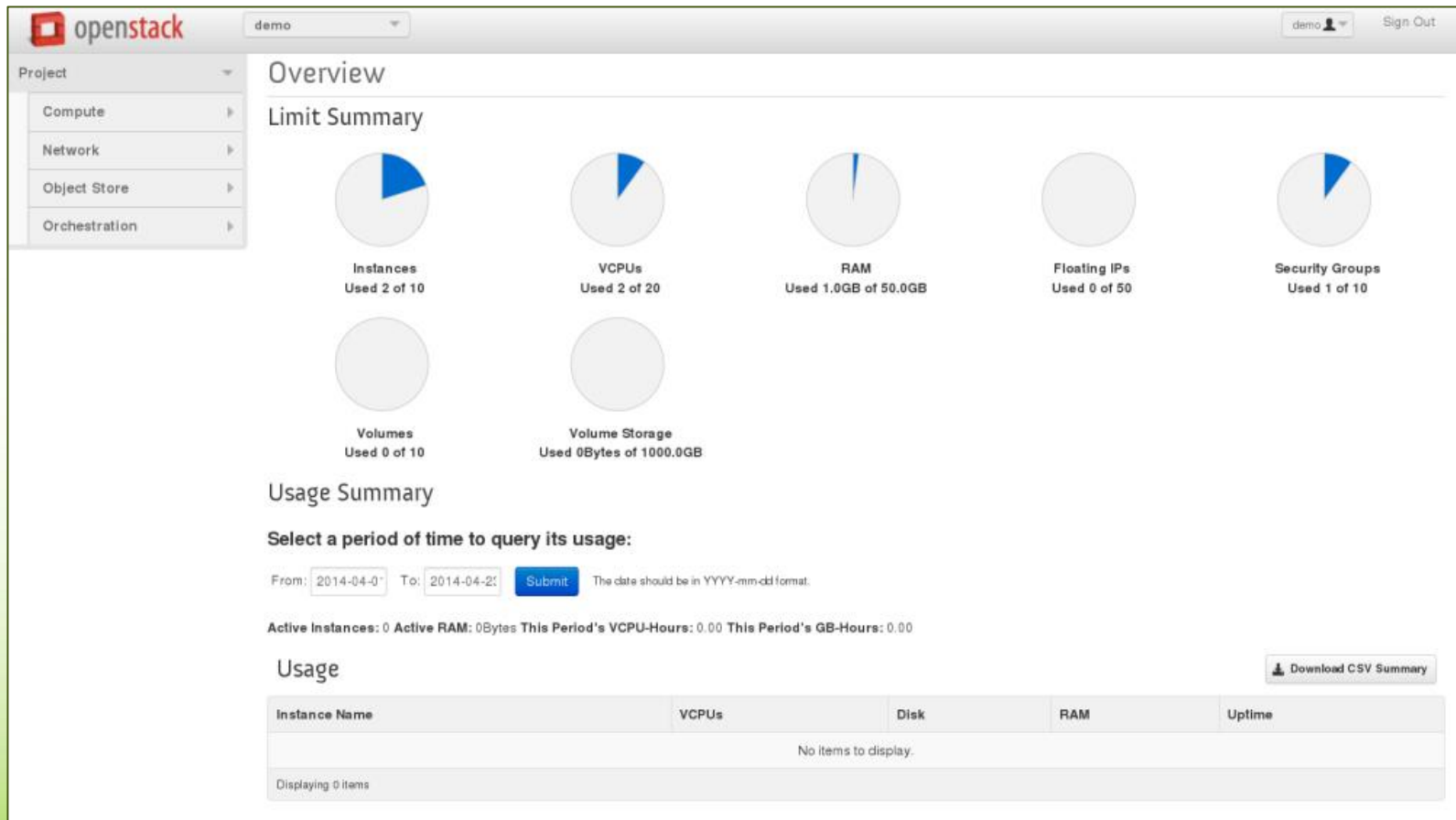
**SWIFT**

*an OpenStack Community Project*

Outros mais...

# Como realizar o *deploy* ?

## Horizon



The screenshot displays the OpenStack Horizon dashboard interface. At the top, the OpenStack logo and a 'demo' dropdown menu are visible. A sidebar on the left lists project categories: Project, Compute, Network, Object Store, and Orchestration. The main content area is titled 'Overview' and 'Limit Summary'. It features seven circular progress indicators representing resource usage: Instances (2 of 10), VCPUs (2 of 20), RAM (1.0GB of 50.0GB), Floating IPs (0 of 50), Security Groups (1 of 10), Volumes (0 of 10), and Volume Storage (0 Bytes of 1000.0GB). Below this is a 'Usage Summary' section with a date range selector (From: 2014-04-01, To: 2014-04-21) and a 'Submit' button. The summary text indicates: 'Active Instances: 0 Active RAM: 0Bytes This Period's VCPU-Hours: 0.00 This Period's GB-Hours: 0.00'. A 'Download CSV Summary' button is located to the right. At the bottom, a table with columns 'Instance Name', 'VCPUs', 'Disk', 'RAM', and 'Uptime' is shown, containing the message 'No items to display.' and 'Displaying 0 items'.

Instance Name	VCPUs	Disk	RAM	Uptime
No items to display.				

Displaying 0 items



# Como realizar o *deploy* ?

## CLI Openstack

```
| owner      | f4ca451c7812473f8c9def826968d3c6 |
| protected  | False                               |
| size       | 255459328                           |
| status     | active                              |
| updated_at | 2013-12-20T21:54:09                 |
+-----+
root@openstack:~# glance image-list
+-----+-----+-----+-----+-----+-----+
| ID          | Name          | Disk Format | Container Format | Size      | Status |
+-----+-----+-----+-----+-----+-----+
| bde870c4-0d4f-429b-aac0-2ec444c92587 | fedora-image  | qcow2       | bare             | 237371392 | active |
| 57622627-e318-4cb9-a6ec-0022ce2c2a96 | ubuntu       | qcow2       | bare             | 254935040 | active |
| 581c02bf-0be0-4e23-a712-2f160c37eba7 | ubuntu-image  | qcow2       | bare             | 254607360 | active |
| e1ee80c3-f29a-4d10-ba68-d2160ebdc958 | ubuntu_motd  | qcow2       | bare             |           | queued  |
| 7d636918-d543-4afa-b2cd-1b88cff89ab8 | ubuntu_motd  | qcow2       | bare             | 255459328 | active |
+-----+-----+-----+-----+-----+-----+
root@openstack:~# glance image-delete e1ee80c3-f29a-4d10-ba68-d2160ebdc958
root@openstack:~# glance image-list
+-----+-----+-----+-----+-----+-----+
| ID          | Name          | Disk Format | Container Format | Size      | Status |
+-----+-----+-----+-----+-----+-----+
| bde870c4-0d4f-429b-aac0-2ec444c92587 | fedora-image  | qcow2       | bare             | 237371392 | active |
| 57622627-e318-4cb9-a6ec-0022ce2c2a96 | ubuntu       | qcow2       | bare             | 254935040 | active |
| 581c02bf-0be0-4e23-a712-2f160c37eba7 | ubuntu-image  | qcow2       | bare             | 254607360 | active |
| 7d636918-d543-4afa-b2cd-1b88cff89ab8 | ubuntu_motd  | qcow2       | bare             | 255459328 | active |
+-----+-----+-----+-----+-----+-----+
root@openstack:~# nova boot ubuntu_motd --flavor 2 --image ubuntu_motd --key-name adminKey --poll
usage: nova [--version] [--debug] [--os-cache] [--timings]
           [--timeout <seconds>] [--os-username <auth-user-name>]
           [--os-password <auth-password>]
           [--os-tenant-name <auth-tenant-name>]
           [--os-tenant-id <auth-tenant-id>] [--os-auth-url <auth-url>]
           [--os-region-name <region-name>] [--os-auth-system <auth-system>]
           [--service-type <service-type>] [--service-name <service-name>]
           [--volume-service-name <volume-service-name>]
           [--endpoint-type <endpoint-type>]
           [--os-compute-api-version <compute-api-ver>]
           [--os-cacert <ca-certificate>] [--insecure]
           [--bypass-url <bypass-url>]
           <subcommand> ...
```

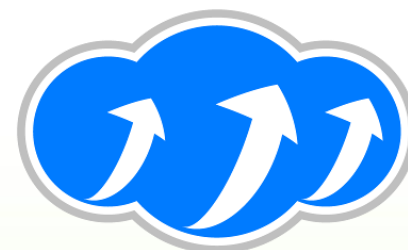


## SDKs

- Python Shade
- Python Libcloud
- Java jClouds
- PHP Opencloud
- ...



shade 1.24.0



libcloud

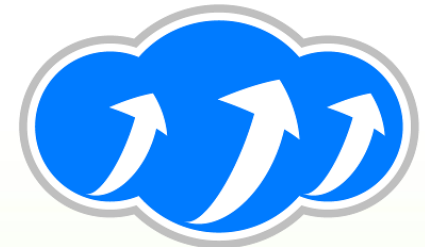
php-opencloud

## SDKs

- **Python Shade**
- Python Libcloud
- Java jClouds
- PHP Opencloud
- ...



shade 1.24.0



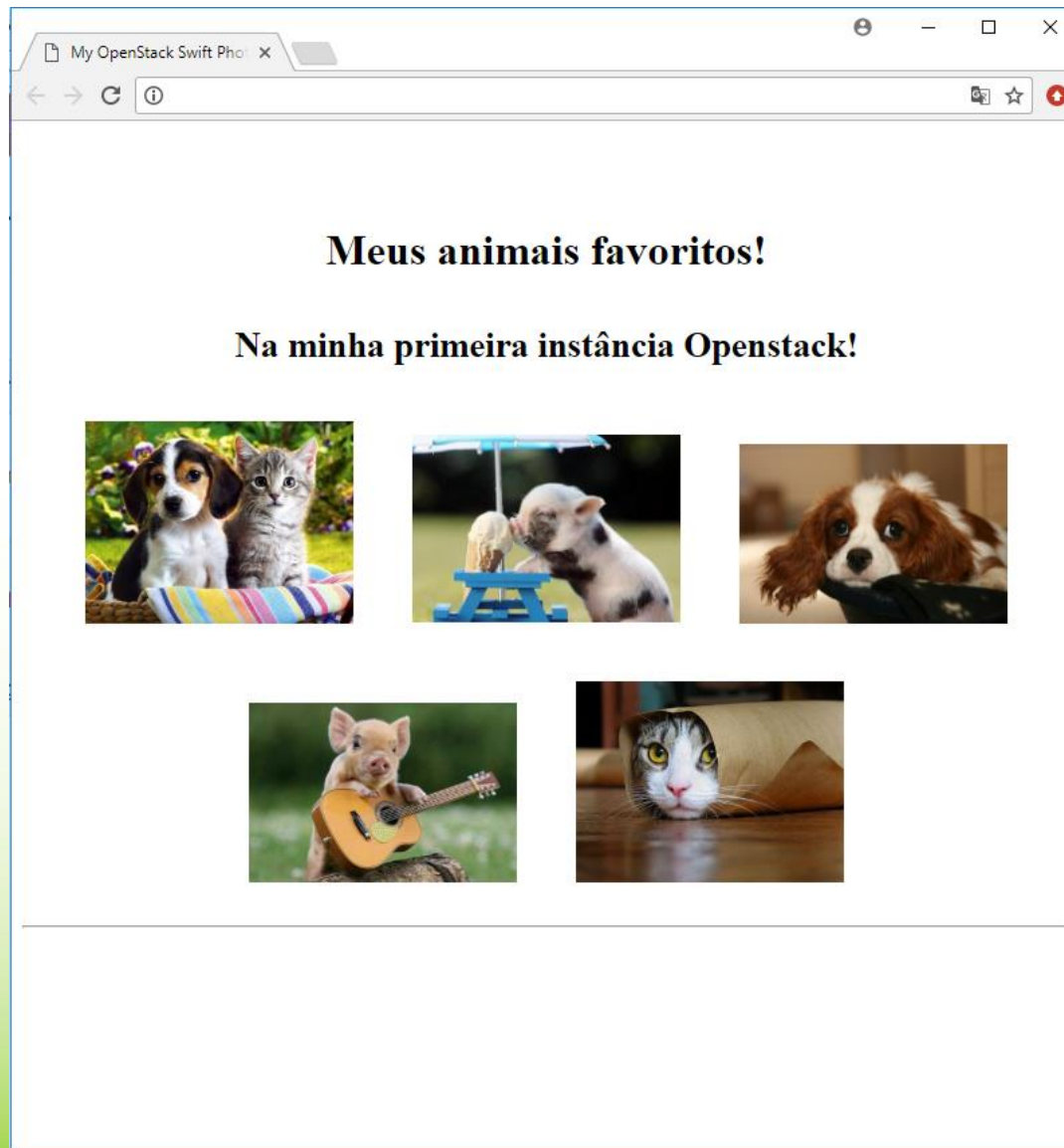
libcloud



- SDK desenvolvida pela Openstack Foundation para interagir com a nuvem Openstack sem a necessidade de se referir diretamente aos módulos da nuvem.
- Facilita o *deployment* de instâncias e aplicações no Openstack.
- Interação direta do desenvolvedor com a linguagem Python.
- **Métodos da biblioteca:**

<https://docs.openstack.org/shade/latest/user/usage.html>

# Python Shade – O que vamos aprender?



**Criação de instâncias, com  
*deploy* de uma aplicação que  
utiliza *object storage*, que  
utilizam balanceamento de carga.**

- ✓ Autenticação com o servidor Openstack
- ✓ Python e Shade instalados.
- ✓ Acesso a internet a partir da instância, neste exemplo.
- ✓ Imagem de um SO (de sua preferência).
- ✓ Aplicação para realizar o *upload*.



```
# apt-get install python
```

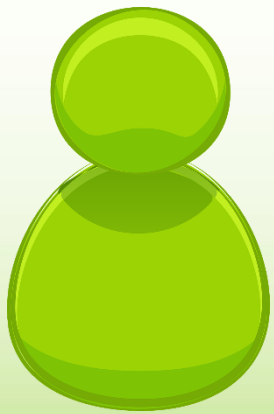
```
# apt-get install python-pip python-dev  
build-essential
```

```
# pip install shade
```

## Windows?

<https://github.com/naturalis/openstack-docs/wiki/Howto:-Installing-and-configuring-the-OpenStack-commandline-tools-on-Windows>

# Como acessar e interagir com o Openstack ?

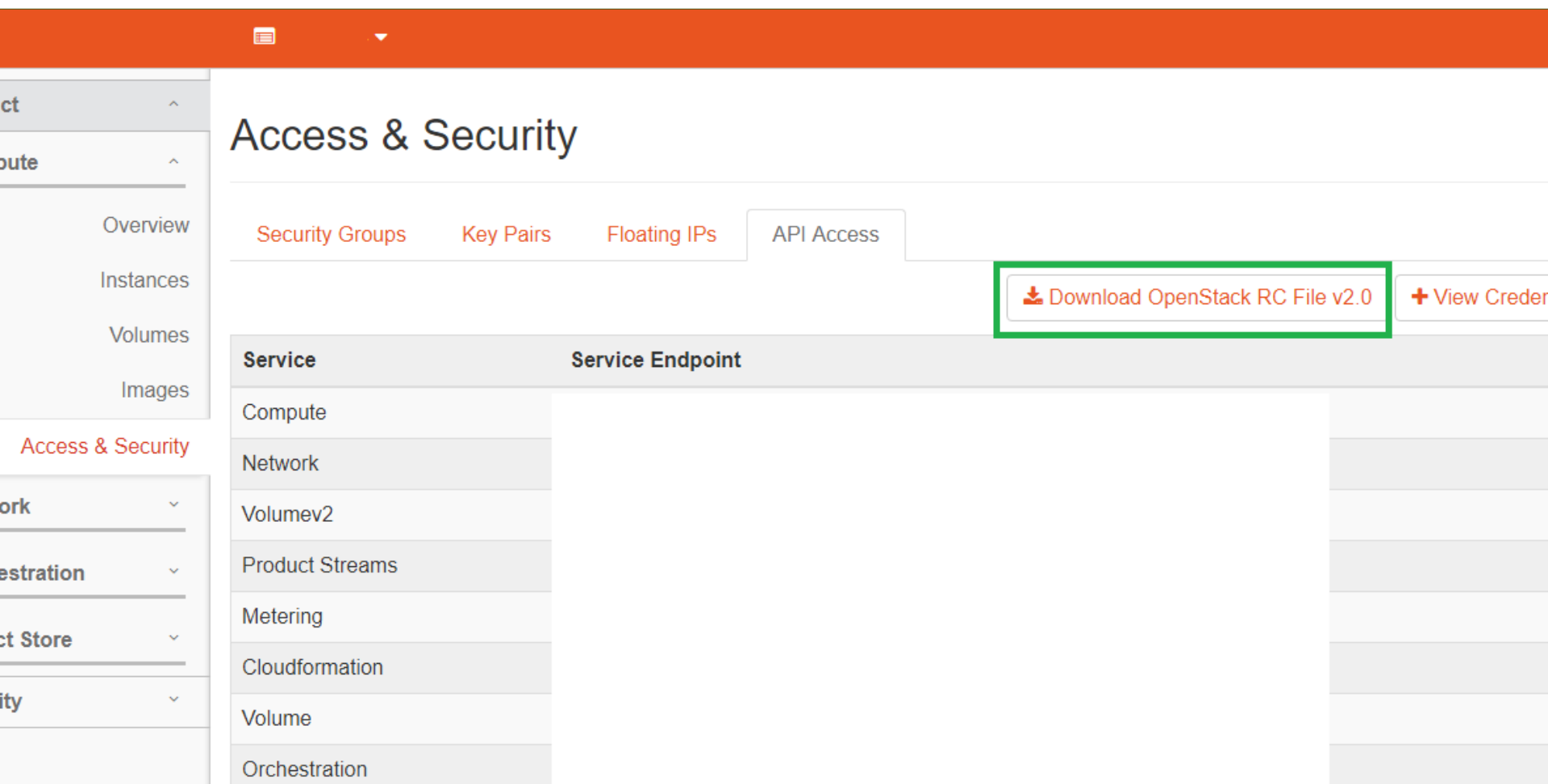


openstack®

## Autenticação com o *clouds.yaml*

```
ubuntu@Bia:~$ cat clouds.yaml
clouds:
  devstack:
    region_name: regioao-openstack
    auth:
      username: 'meu-usuario'
      password: 'minha-senha'
      project_name: 'meu-projeto'
      auth_url: 'http://minha-url/identity'
      project_domain_id: 'default'
```

## Autenticação via comando "source openrc.sh"



The screenshot shows the OpenStack Horizon dashboard. The left sidebar contains navigation links: Overview, Instances, Volumes, Images, Access & Security (highlighted), Network, Registration, Object Store, and Identity. The main content area is titled 'Access & Security' and has four tabs: Security Groups, Key Pairs, Floating IPs, and API Access (selected). In the API Access tab, there is a button labeled 'Download OpenStack RC File v2.0' which is highlighted with a green rectangle. To its right is a button labeled '+ View Credentials'. Below these buttons is a table with two columns: 'Service' and 'Service Endpoint'.

Service	Service Endpoint
Compute	
Network	
Volumev2	
Product Streams	
Metering	
Cloudformation	
Volume	
Orchestration	

- Verificando a conectividade com o Openstack.
  - **python -m os\_client\_config.config**





- Verificando a conectividade com o Openstack.
  - **python -m os\_client\_config.config**

```
ubuntu@Bia:~$ python -m os_client_config.config
('envvars', ' ', {'auth_type': u'password', u'compute_api_version': u'2', u'orchestration_api_version': u'2', u'api_version': u'1.0', u'metering_api_version': u'2', u'network_api_version': u'2', u'message': u'', u'image_format': u'qcow2', u's': [], u'image_api_version': u'2', 'verify': True, u'dns api version': u'2', u'object_store_api_version': u'1', u'sta': u'container_infra_api_version': u'1', 'region_name': 'us-east-1', 'api_timeout': None, u'baremetal_api_version': u'1', 'username': 'beatriz.neopath', 'project id': '3a98ad500134611ba17c067a026e2a5', 'password': 'minha-senha', 'project na': 'neopath', 'auth_url': 'https://keystone.us-east-1.amazonaws.com', u'image_api_use_tasks': False, u'floating_ip_source': u'neutron', u'cacert': None, u'application_catalog_api_version': u'1', u'key_manager_api_version': u'v1', u'workflow_api_version': u'1', u'volume_api_version': u'2', u'cert': None, u'secgroup_source': u'neutron', u'container_api_version': u'1', u'interface': 'publicURL', u'disable_vendor_agent': {}})
```

## #Primeira interacao

### # importando a biblioteca

```
import shade
import munch
```

### # conectando com o openstack

```
cloud = shade.openstack_cloud(cloud='devstack')
```

### # Imprimindo uma imagem existente no openstack

```
print("Imprimindo images existentes no Openstack \n")
images = cloud.list_images()
for image in images:
    print(image.name + ' ' + image.id + '\n')
```

### # Imprimindo flavors existentes

```
print("Imprimindo flavors com 1GB de RAM, 1 CPU e +1GB de disco existentes  
no Openstack /n")
flavors = cloud.list_flavors()
for flavor in flavors:
    if (flavor.ram == 1024) and (flavor.disk > 1) and (flavor.vcpus == 1):
        print(flavor.name + ' ` ' + flavor.id + '\n')
```

## Vídeo: OpenStack - Listagem de images e flavors com Shade

<https://www.youtube.com/watch?v=sGlqsE2Md9Q&t=6s>

## # Importando a biblioteca

```
import shade
```

## # Habilitando o modo debug (opcional)

```
shade.simple_logging(debug=True)
```

## # Conectando com o openstack

```
cloud = shade.openstack_cloud(cloud='devstack')
```

## # *Upload* da imagem

```
image = cloud.create_image('ubuntu-trusty', filename='ubuntu-trusty.qcow2', wait=True)
```

```
print "O id da imagem é: " + cloud.get_image("ubuntu-trusty").id + " \n"  
print "Imagem criada com sucesso!"
```

### Vídeo: OpenStack - Upload de imagens com Shade

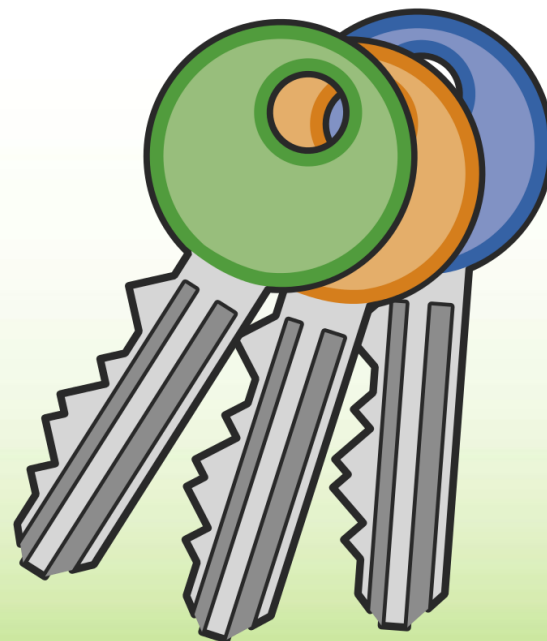
[https://www.youtube.com/watch?v=L8KHh7sD\\_Do](https://www.youtube.com/watch?v=L8KHh7sD_Do)

- Queremos acessar as instâncias remotamente!
  - Configurar acesso remoto.
  - Liberar o acesso às portas desejadas.



```
# apt install ssh
```

```
$ ssh-keygen
```



**./ssh/id\_rsa.pub**

(chave pública, a ir para o servidor)

**./ssh/id\_rsa**

(chave privada, a ficar na máquina local)

```
import shade
print "Iniciando script \n"
cloud = shade.openstack_cloud(cloud='devstack')
```

**# Inserindo a chave no Openstack para acesso remoto a instancia que será criada**

```
print('\n Inserindo chave para acesso remoto ')
print('\n Verificando se a chave já existe... ')
```

```
keypair_name = 'minha-chave'
pub_key = open("./minha-chave.pub", 'r').read().strip()
```

```
if cloud.search_keypairs(keypair_name):
    print('\n Uma chave com esse nome já existe, ela será usada.')
else:
    print('\n Adicionando a chave.')
    cloud.create_keypair(keypair_name, pub_key)
```

```
for keypair in cloud.list_keypairs():
    print("\n Nome da chave: " + keypair.name + "\n Chave: " +
        keypair.public_key)
print "\n Chave adicionada com sucesso"
```

```
import shade
print "Iniciando script \n"
cloud = shade.openstack_cloud(cloud='devstack')
print("\n Criando grupo de acesso")

# Criando grupo para liberar o acesso a instancia
print('\n Verificando se o grupo já existe...')

sec_group_name = 'meu-grupo'

if cloud.search_security_groups(sec_group_name):
    print('\n Um grupo com esse nome já existe, ele será usado.')
else:
    print('\n Adicionando grupo')
    cloud.create_security_group(sec_group_name, 'Grupo de acesso a rede da minha
    primeira instancia com Shade')
    cloud.create_security_group_rule(sec_group_name, 80, 80, 'TCP', remote_ip_prefix=IP)
    cloud.create_security_group_rule(sec_group_name, 22, 22, 'TCP', remote_ip_prefix=IP)

print ("\n grupo criado com sucesso " +
    cloud.get_security_group(sec_group_name).name)
```

```
import shade
print "Iniciando script \n"
cloud = shade.openstack_cloud(cloud='devstack')

# ... outros parametros nos slides anteriores.

# Criando ultimos parametros da instancia
ex_userdata = '''#!/usr/bin/env bash
echo "nameserver 8.8.8.8" >> /etc/resolv.conf; sudo apt-get update; sudo
    apt-get -y install apache2;
echo "<html><head><title>Minha primeira instancia no
    Openstack!</title></head><body><h1>Apenas uma
    instancia</h1><p>Parabens!</p></body></html>" >
    /var/www/html/index.html'''

print "\n Criando instancia"
instance_name = 'minha-instancia'
rede = "interno"
cloud.create_server(instance_name, image=image_id, flavor=flavor_name,
    wait=True, auto_ip=True, key_name=keypair_name,
    security_groups=[sec_group_name], network=rede,
    userdata=ex_userdata)
```

O *storage* criado pode ser utilizado para guardar dados da aplicação.

- Fotos;
- Arquivos;
- Etc.



```
import shade
print ("Iniciando script")
cloud = shade.openstack_cloud(cloud='devstack')
```

## **# Criando o object storage para armazenar os dados**

```
container_name = 'meu-album'
container = cloud.create_container(container_name, public=True)
print 'Object Storage criado \n'
```

## **# Criando uma estrutura (dicionario) para relacionar o endereco das imgs # que serao enviadas com os nomes que elas terao no storage.**

```
pets = {'cat1': 'fotos/cat1.jpg', 'cat-dog1': 'fotos/cat-dog1.jpg', 'dog1':  
        'fotos/dog1.jpg'}
print "Lista de imgs criada \n"
```

## **# Upload das imagens para o storage**

```
print "Criando objetos no storage \n"
```

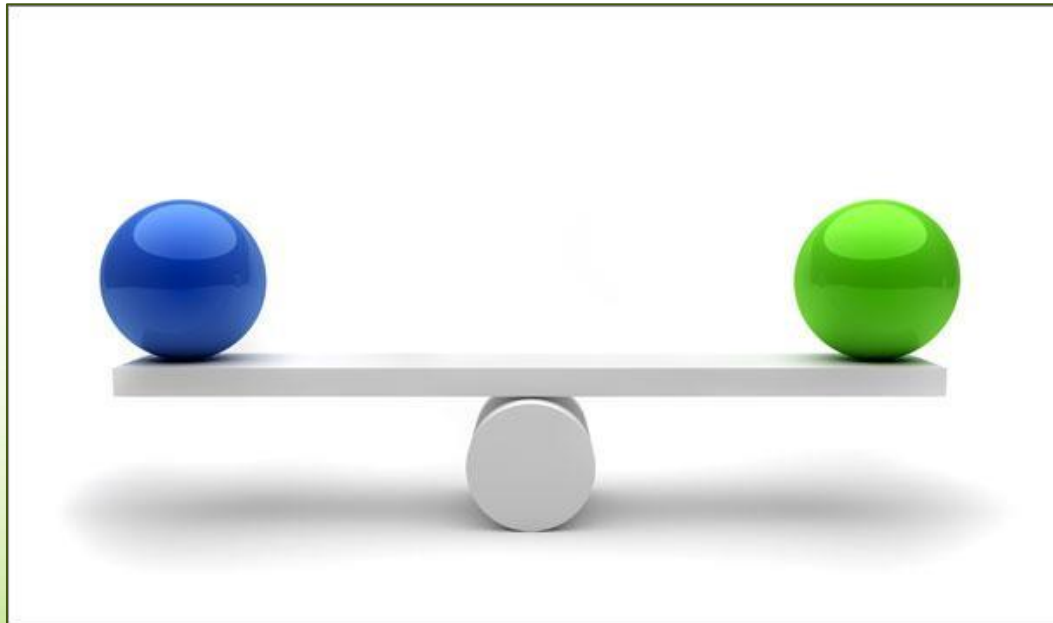
```
for object_name, file_path in pets.items():
    cloud.create_object(container=container_name, name=object_name,  
                        filename=file_path)
```

```
print "Objetos adicionados com sucesso! \n"
```

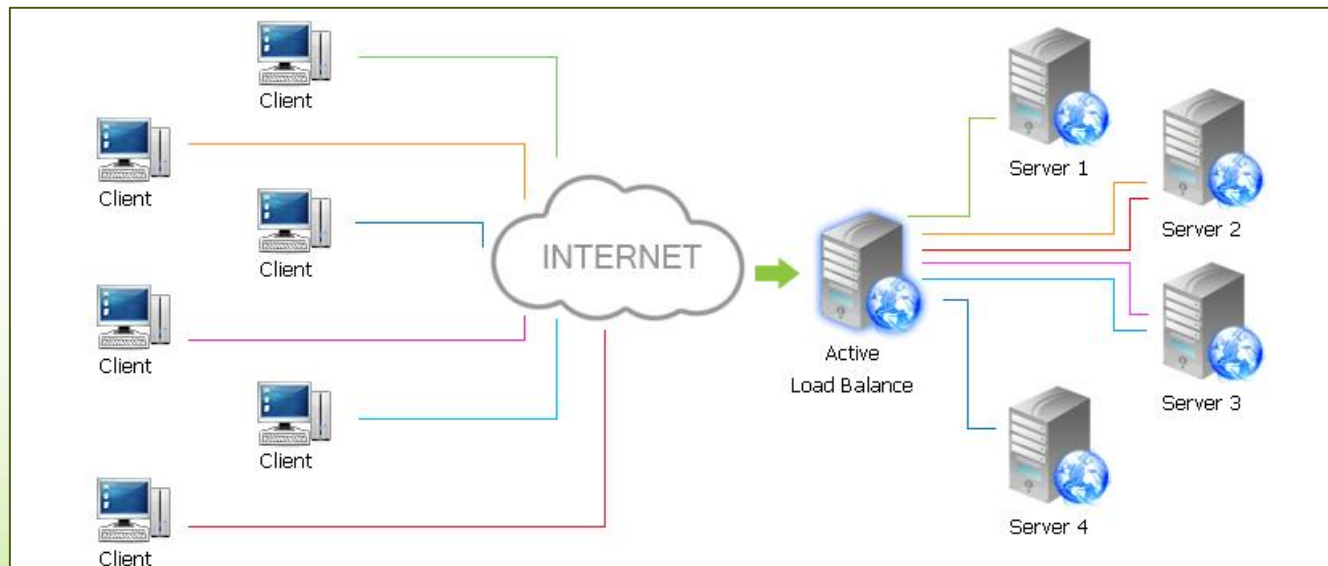
### **Vídeo: OpenStack - Criação de storage de objetos com Shade**

<https://www.youtube.com/watch?v=1dU6kaKbr9g>

# Como utilizar o balanceamento?



## Com multiplas instâncias!



## Como o Shade auxilia?



## Como o Shade auxilia?

...

```
cloud.create_server(instance_name,      image=image_id,  
flavor=flavor_name,      wait=True,      auto_ip=True,  
key_name=keypair_name,  
security_groups=[sec_group_name],      network=rede,  
userdata=ex_userdata, min_count=2, max_count=5)
```

## Vídeo: OpenStack - Criação de instâncias e deploy de aplicações com Shade

<https://www.youtube.com/watch?v=JfncOUHeqBg>

- Não é necessário que a aplicação seja desenvolvida em Python para utilizar o Shade.
- A aplicação que se deseja colocar no ar pode estar em qualquer lugar.
- Link do Github para *download* dos códigos completos: **<https://github.com/Neopath-Integrated-Systems/OpenStack-Meetup>**





obrigada !!



1. **Métodos da biblioteca:**  
<https://docs.openstack.org/shade/latest/user/usage.html>
2. **Utilização do Swift:**  
<https://github.com/MBonell/openstack-sdks-challenges/tree/master/shade/swift/photo-album>
3. **Utilização do Shade:**  
[https://developer.openstack.org/firstapp-shade/getting\\_started.html](https://developer.openstack.org/firstapp-shade/getting_started.html)
4. **Balanceamento de carga (CLI):**  
<https://docs.openstack.org/octavia/pike/user/guides/basic-cookbook.html>



**NEOPATH<sup>®</sup>**

**Excelência em Comunicação e Colaboração**

<http://www.neopath.com.br>