

OpenStack API快速入门

目录

OpenStack API快速入门	1
OpenStack API介绍	1
获取证书	1
向API发送请求	3
安装python-novaclient	3
列出所有的镜像	4
启动实例	5

OpenStack API快速入门

OpenStack系统包含几个核心项目，他们是分开安装的，但能共同工作，取决于你的云的需求：OpenStack计算服务，OpenStack对象存储，OpenStack身份服务，和OpenStack镜像存储。借助TryStack中OpenStack安装，OpenStack计算服务，OpenStack标识服务，和OpenStack镜像存储几个项目共同工作在一起。

OpenStack API介绍

本页包含使用计算API与OpenStack云之间交互的基础知识，这是在通过身份服务API完成认证之后。你可以启动镜像，赋予实例元数据，构建一个云，这些全部可以采用API完成。在docs.openstack.org/api里发布有一份API参考，包含所有有效的命令，[OpenStack Compute API 1.1 规范](#) 和 [Identity Service 2.0 规范](#)。

获取证书

证书是个组合信息，包括你的用户名、密码，以及云运行在哪个租户（或者说，项目）之下。当你与云直接通过API服务端口交互的时候，你只需要生成一个附加令牌。你的云管理员会提供你的用户名，密码，以及你的租户标识符，你可以根据这些信息生授权令牌。你还可以从仪表盘URL那里得到租户标识符，例如，<https://trystack.org/dash/296/images/>里面表明了租户标识符为296。

通常来说，这些令牌24小时之内是有效的，当令牌过期后，你会得到401（未授权）错误，那时你可以通过程序请求另外一张令牌。一般的工作流程如下所示：

1. 开始API请求之前，首先要从云管理员提供的服务端口那里请求授权令牌，服务端口通常是<http://hostname:port/v2.0/tokens>。发送你的用户名，密码，和你所属的组或者帐户（也就是auth-speak中的”租户“）

```
curl -k -X 'POST' -v https://nova-api.trystack.org:5443/v2.0/
tokens -d '{"auth":{"passwordCredentials":{"username": "joecool",
"password": "coolword"}, "tenantId": "5"}}' -H 'Content-type: application/
json'
```

2. 服务器返回响应消息，其中含有一个24小时有效的令牌。可以使用这个令牌发送API请求，请求头中包含X-Auth-Token字段。

```
curl -k -D - -H "X-Auth-Token: 7d2f63fd-4dcc-4752-8e9b-1d08f989cc00" -X 'GET' -v https://nova-api.trystack.org:9774/v1.1/296/extensions -H 'Content-type: application/json'
```

3. 将这个令牌置于信息头x-auth-token中，反复发送API请求，直到任一种情况发生：1)任务完成，或者2)你得到401（未授权）响应
4. 当你得到401的响应时，再次请求令牌，直到脚本任务完成。

对一个使用身份服务的典型的OpenStack部署来说，你可以通过cURL的这个命令来请求令牌：

```
$ curl -X 'POST' -v https://nova-api.trystack.org:5443/v2.0/tokens -d '{"auth":{"passwordCredentials":{"username": "joecool", "password": "coolword"}, "tenantId": "5"}}' -H 'Content-type: application/json'
```

作为回复，你会得到一个200 OK的响应，同时以“id”的形式，带回一个令牌：“cd427a33-bb4a-4079-a6d7-0ae148bdeda9”和一个有效期：从现在开始的24小时。响应消息看起来如下所示：

```
{
  "access": {
    "serviceCatalog": [
      {
        "endpoints": [
          {
            "adminURL": "https://nova-api.trystack.org:9774/v1.1/1",
            "internalURL": "https://nova-api.trystack.org:9774/v1.1/1",
            "publicURL": "https://nova-api.trystack.org:9774/v1.1/1",
            "region": "RegionOne"
          }
        ],
        "name": "nova",
        "type": "compute"
      },
      {
        "endpoints": [
          {
            "adminURL": "https://GLANCE_API_IS_NOT_DISCLOSED/v1.1/1",
            "internalURL": "https://GLANCE_API_IS_NOT_DISCLOSED/v1.1/1",
            "publicURL": "https://GLANCE_API_IS_NOT_DISCLOSED/v1.1/1",
            "region": "RegionOne"
          }
        ],
        "name": "glance",
        "type": "image"
      },
      {
        "endpoints": [
          {
```

```

        "adminURL": "https://nova-api.trystack.org:5443/v2.0",
        "internalURL": "https://keystone.trystack.org:5000/v2.
0",
        "publicURL": "https://keystone.trystack.org:5000/v2.
0",
        "region": "RegionOne"
    }
},
{
    "name": "keystone",
    "type": "identity"
}
],
"token": {
    "expires": "2012-02-15T19:32:21",
    "id": "5df9d45d-d198-4222-9b4c-7a280aa35666",
    "tenant": {
        "id": "1",
        "name": "admin"
    }
},
"user": {
    "id": "14",
    "name": "annegentle",
    "roles": [
        {
            "id": "2",
            "name": "Member",
            "tenantId": "1"
        }
    ]
}
}
}

```

向API发送请求

要通过API向OpenStack发送请求，你有很多选择。开发人员和测试人员可能喜欢使用cURL，一个命令行工具，下载地址为<http://curl.haxx.se/>。使用cURL，你可以通过命令行发送HTTP请求和接收响应。

如果你希望采用一个更图形化的界面，Firefox的REST客户端也可以用于测试和尝试命令。参见<https://addons.mozilla.org/en-US/firefox/addon/restclient/>。你也可以下载和安装rest-client，一个Java应用，来测试RESTful web服务，下载地址为<http://code.google.com/p/rest-client/>。

如果你使用cURL或者REST客户端，你需要生成一个令牌，如上面所示。

安装python-novaclient

你可以通过使用客户端，比如python-novaclient或者python-compute客户端，做更多的复杂工作。Python-novaclient实现了Compute 1.1的API。openstack-compute客户端工作在Rackspace Cloud Servers公有云之上，实现的是OpenStack Compute 1.0 API。使用Python-novaclient工具时，你只需要提供用户名和密码。

安装python-novaclient后，就会生成一个nova shell命令行，它可以通过命令行形式调用Compute API。你安装客户端，提供你的用户名和密码，为方便起见，可以把他们设为环境变量，之后，你就可以在命令行窗口向你的云发送命令了。

安装 python-novaclient，如下所示，获取稳定的nova版本

```
$ pip install -e git+https://github.com/openstack/python-novaclient.git#egg=python-novaclient
```

现在你已经完成python-novaclient的安装，确认安是否装成功，请输入：

```
$ nova help

usage: nova [--debug] [--os_username OS_USERNAME] [--os_password OS_PASSWORD]
           [--os_tenant_name OS_TENANT_NAME] [--os_auth_url OS_AUTH_URL]
           [--os_region_name OS_REGION_NAME] [--service_type SERVICE_TYPE]
           [--service_name SERVICE_NAME] [--endpoint_type ENDPOINT_TYPE]
           [--version VERSION]
           <subcommand> ...
```

作为返回，你将得到nova命令行客户端中的所有的命令和参数的列表。通过将所需参数设为环境变量，你可以在命令行中快速使用这些命令。你可以在nova命令中添加--os_username，或者设定为环境变量。当使用TryStack时，你将得到一个端点（endpoint），这个端点就是nova客户端认定的NOVA_URL。在TryStack中，登录后你可以从https://trystack.org/dash/api_info/看到这些环境变量。

```
export NOVA_USERNAME=facebook12345679
export NOVA_API_KEY=coolword
export NOVA_PROJECT_ID=facebook12345679
export NOVA_URL=https://nova-api.trystack.org:5443/v2.0
export NOVA_VERSION=1.1
```

列出所有的镜像

在你开始搭建你的云之前，你应该知道有哪些镜像是可用的。这可以通过询问镜像服务”有哪几种可用的配置类型“来实现。镜像服务类似于云计算中的iTunes - 在使用你喜爱的镜像创建新的实例之前，你可以查看镜像的播放类表。要得到镜像列表、他们的名字、状态和ID，使用如下命令：

```
$ nova image-list
```

ID	Name	Status	Server
12	natty-server-cloudimg-amd64-kernel	ACTIVE	
13	natty-server-cloudimg-amd64	ACTIVE	
14	oneiric-server-cloudimg-amd64-kernel	ACTIVE	
15	oneiric-server-cloudimg-amd64	ACTIVE	

下一步，你需要知道他们中每一个的相对大小。

```
$ nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs	RXTX_Factor
1	m1.tiny	512	0	N/A	0	1	
2	m1.small	2048	20	N/A	0	1	
3	m1.medium	4096	40	N/A	0	2	
4	m1.large	8192	80	N/A	0	4	
5	m1.xlarge	16384	160	N/A	0	8	

你还可以缩小列表范围，通过用grep来只查找Ubuntu 11.04的镜像，命令如下所示：

```
$ nova image-list | grep 'natty'
```

12	natty-server-cloudimg-amd64-kernel	ACTIVE	
13	natty-server-cloudimg-amd64	ACTIVE	

使用这些对你来说有用的信息，你可以选择镜像和类型的组合来创建你的虚拟服务器，启动实例。

启动实例

要启动服务器，你首先选择你想要的镜像以及相配的大小，找到相应的镜像ID和类型ID，然后根据这些ID们创建命令。从我们之前得到的信息来看，我们知道那个Ubuntu Natty镜像的ID是13，如果你想启动一个小的实例，2GB内存和20GB硬盘，你应该选择m1.small类型，当在TryStack中使用1.1版本的API时，它的ID是2。将这些参数代入boot命令，你就可以创建一个新的虚拟服务器了。



注意

当使用支持Compute API 1.1版本的服务端点时，你能够通过一个ID来启动实例。而当使用支持Compute API 2版本的服务端点时，你必须使用UUID来启动实例。

```
$ nova boot --flavor=2 --image=13 testserver
```

Property	Value
adminPass	*****
created	2011-09-01T21:40:41Z
flavor	m1.small
hostId	
id	1805
image	natty
metadata	{}
name	testserver
progress	0
status	BUILD
updated	2011-09-01T21:40:41Z

```
| uuid | ce044452-f22e-4ea4-a3ec-d1cde80cf996 |
+-----+
```

现在，你可以在你的云中查看这台服务器了，使用nova list命令：

```
$ nova list
+-----+-----+-----+-----+
| ID | Name | Status | Networks |
+-----+-----+-----+-----+
| 1805 | testserver | ACTIVE | private=10.4.96.81 |
+-----+-----+-----+-----+
```

你能够看到三种状态：ACTIVE，BUILDING，和UNKNOWN。状态BUILDING是个瞬时状态，很有可能你根本看不到它。如果你看到UNKNOWN，再次运行nova list，直到这个状态消失。

要查看某个特定服务器的所有信息，使用nova show以及你刚才从nova list命令中看到的服务器ID。

```
$ nova show 1805
+-----+-----+
| Property | Value |
+-----+-----+
| created | 2011-09-01T21:40:41Z |
| flavor | m1.small |
| hostId | 58a7430169aa42cde5ce2456b0cb5bb5ac1ab0703bab6420e8a49e6e |
| id | 1805 |
| image | natty |
| metadata | {} |
| name | testserver |
| private network | 10.4.96.81 |
| progress | 100 |
| status | ACTIVE |
| updated | 2011-09-01T21:40:46Z |
| uuid | ce044452-f22e-4ea4-a3ec-d1cde80cf996 |
+-----+-----+
```

现在你可以再次启动那个镜像了，不过启动时需要给它增加一些信息，这样你就能在日益增长的弹性云中更容易的识别它了。使用--meta选项，带一个key=value键值对，这里你可以填入键和值。例如，你可以增加一个描述，以及服务器的创建者。

```
$ nova boot testserver --meta description='Use for testing purposes' --meta
creator=joecool
```