开发手册

系统全局相关

产品版本：ZStack 2.5.1

文档版本：V2.5.1

ZStack

版权声明

商标说明

ZStack商标和其他云轴商标均为上海云轴信息科技有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受上海云轴公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，上海云轴公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 引言

产品版本

目前与本文档相对应的产品版本为：ZStack 2.5.1

读者对象

本文档详述了ZStack 2.5.1 RESTful API的使用规范，并提供所有API的详细定义。本文档主要适用于以下读者：

- 架构设计师
- 开发工程师
- 测试工程师
- 项目实施人员
- 对ZStack有兴趣研究的相关人员

版本更新

2.5.1

2018/07/10主要更新：

1. Shared Block主存储功能增强
   - 一个集群支持挂载多个Shared Block主存储
   - 跨Shared Block主存储的整机迁移
2. 资源编排支持普通账户/企业管理账号体系使用
3. 修复已知问题，提升系统稳定性

2.5.0

2018/07/05主要更新：

1. 资源编排
2. 整机克隆
3. vCenter接管功能增强
   - vCenter监控报警
   - 多vCenter区分
   - 独立CPU授权

4. 操作日志/审计信息优化展示
5. 性能Top5页面展示优化
6. 其它相关功能和优化
   - 新增多个操作场景进度条
   - 操作助手和帮助文档
   - 优化界面交互
   - 优化部分业务逻辑

2.4.0

2018/06/11主要更新：

1. 企业管理模块：项目管理、工单审批、独立区域管理
2. 支持ARM服务器
3. 应用中心
4. 资源监控增强
   - 详情页资源监控
   - 资源实时监控
5. 新增主存储类型：Shared Block共享块存储
6. GPU功能增强
7. 模块许可证
8. 云主机导出增强
9. 计算规格的物理机分配策略新增非强制/强制模式
10. VPC路由器配置DNS
11. 其它相关功能和优化
    - 新增多个操作场景进度条
    - 操作助手和帮助文档
    - 优化界面交互
    - 优化部分业务逻辑

2.3.2

2018/05/11主要更新：

1. 云资源池：

- 云主机根云盘/数据云盘容量在线扩展
- 通过FTP和SFTP方式在线添加镜像模板

2. 硬件设施：
   - 分布式存储Ceph以存储池（Pool）粒度显示容量使用情况
   - 识别物理机CPU架构，识别主流Intel和AMD处理器
   - 集群按照物理机CPU架构定义属性，为云主机提供丰富的CPU多媒体指令集，以及提升热迁移兼容性
   - 指定集群云主机热/冷迁移网络
3. 网络服务：
   - 负载均衡监听协议支持HTTPS，需绑定证书使用
   - 强化监听器功能
4. VMware vCenter接管：
   - vCenter云主机迁移、克隆
   - vCenter物理机维护模式
5. 平台运维：TOP5性能分析，支持对应项搜索排序
6. 平台管理：
   - 强化定时任务功能
   - 在管理界面上修改控制台代理地址
7. 大屏监控：解决登录会话超时失效
8. 混合云：对接大河云联SD-WAN服务，提供混合云高速链路
9. 超融合解决方案：
   - 管理节点云主机管理员密码重置
   - 管理节点云主机跨网段创建/启动，跨网络异地部署
   - 管理节点云主机部署/迁移至非超融合节点，适应更广泛场景
10. 其它相关功能和优化：
    - 新增多个操作场景进度条
    - 操作助手和帮助文档
    - 优化界面交互
    - 优化部分业务逻辑

2.3.1

2018/04/03主要更新：

1. 网络拓扑
2. 新版菜单导航、新版首页
3. ZWatch：全新监控报警系统
4. ZStack定制版ISO新增：基于CentOS 7.4深度定制版本
5. 亲和组
6. 增强vCenter接管功能：接管vCenter云盘、基于vCenter云路由网络提供网络服务
7. 一个云主机加载多个ISO
8. 多种策略创建云主机
9. 一个二层网络可用于创建多个三层网络
10. 操作日志/审计全新改版
11. HTTPS安全访问UI管理界面
12. 内部访问业务流量的负载均衡
13. 优化自定义UI
14. 多个场景新增进度条、操作助手和帮助文档，优化UI交互
15. 优化部分业务逻辑

2.3.0

2018/02/08主要更新：

1. 专有网络VPC
2. 混合云灾备（混合云版支持）
3. 大屏监控
4. 用户自定义UI
5. ImageStore类型镜像服务器支持Ceph类型主存储
6. 支持vSwitch
7. 支持vCenter资源同步
8. ESXi云主机支持扁平网络
9. 云主机更换操作系统
10. 跨NFS存储数据迁移
11. 虚拟IP支持QoS

12.支持AD认证

13.云主机自定义MAC地址

14.强化浏览器上传镜像功能

15.新增云盘镜像资源

16.数据云盘扩容

17.数据云盘规格支持QoS

18.停止NeverStop状态的云主机

19.开放云路由公网IP，并支持同一虚拟IP多网络服务复用

20.支持USB设备透传，强化外接设备透传功能

21.增加VDI SPICE流量优化选项

22.支持修改已设置的存储网络

23.支持设置VXLAN对普通账户的配额

24.支持ImageStore类型镜像服务器间的数据同步

25.管理节点数据库自动备份到远程服务器

26.多个场景新增进度条、操作助手和帮助文档，优化UI交互

27.优化部分业务逻辑

2.2

2017/10/16主要更新：

1. 公有网络创建云主机

2. 自定义DHCP模式

3. 新增系统网络

4. 云主机根云盘扩容

5. 浏览器添加镜像（目前支持ImageStore类型镜像服务器）

6. 管理节点高可用：多网络配置

7. 跨Ceph存储数据迁移

8. 增强Ceph存储功能

9. 增强VDI功能

10.LDAP自定义过滤规则

11.增强裸机管理

12.单集群支持多类型主存储（目前支持本地存储+NFS/SMP类型）

13.更换License支持本地上传

14.共享存储指定存储网络，增强云主机高可用

15.多个场景新增进度条、操作助手和帮助文档，优化UI交互

16.优化部分业务逻辑

2.1

2017/08/14主要更新：

1. VDI

2. 裸机管理

3. GPU透传

4. 智能报警

5. 集群挂载多个主存储

6. 新版定时器

7. 静态路由

8. User Data导入

9. 云路由加载多个公有网络

10.增量升级

11.优化部分业务逻辑

2 系列索引

ZStack 2.5.1 开发手册系列索引如下：

- 《ZStack 2.5.1 开发手册 云资源池》
- 《ZStack 2.5.1 开发手册 硬件设施》
- 《ZStack 2.5.1 开发手册 网络资源》
- 《ZStack 2.5.1 开发手册 网络服务》
- 《ZStack 2.5.1 开发手册 vCenter》
- 《ZStack 2.5.1 开发手册 平台运维》
- 《ZStack 2.5.1 开发手册 平台管理》
- 《ZStack 2.5.1 开发手册 设置》
- 《ZStack 2.5.1 开发手册 系统全局相关》

3 API规范概述

ZStack 2.5.1提供原生RESTful支持。您可以通过REST定义的架构设计原则和约束条件，并使用支持HTTP的编程语言进行开发。

3.1 HTTP方法 (HTTP Verbs)

当前API支持如下操作资源的方法：

方法名	描述
GET	获取资源信息。 • 所有的查询API以及读API均使用该方法。
POST	创建一个资源。
PUT	修改一个资源。 • 所有对资源的修改操作，以及类RPC调用的操作，例如启动虚拟机，均使用该方法。
DELETE	删除一个资源。

3.2 传参方式

URL、Query String、HTTP body三种方式均可用于传参。每种方式可以单独使用，也可以混合使用，具体使用哪种传参方式由具体API决定。

URL传参

当对某具体资源进行操作时，资源的UUID通过编码到URL的方式进行传参。

例如启动一个UUID为 *f97143d60f1042c9badd9a1336d3c105*的虚拟机，URL格式为：

```
zstack/v1/vm-instances/f97143d60f1042c9badd9a1336d3c105/actions
```

这里UUID编码到URL路径当中。

Query String传参

所有使用HTTP GET方法的API均使用Query String传参。

例如查询所有状态为Running的虚拟机， URL格式为：

```
zstack/v1/vm-instances?condition=state=Running
```

HTTP Body传参

当使用POST方法创建一个资源，或PUT方法修改一个资源时，除通过URL传参的部分外，剩余参数均通过HTTP Body传参。

例如在指定物理主机上启动一个虚拟机：

```
PUT zstack/v1/vm-instances/f97143d60f1042c9badd9a1336d3c105/actions

{
  "startVmInstance": {
        "hostUuid": "8aef7e3a53b34eedaa05027a919156d9"
    }
}
```

这里虚拟机的UUID通过URL传参，参数**hostUuid**则通过HTTP Body传递。

3.3 HTTP Headers

当前API使用如下自定义HTTP Headers：

Authorization

除了少数API外（例如登录API），使用ZStack API前都需要一个会话(session)，在调用API时通过Authorization HTTP Header传递会话UUID。该Header的格式为：

```
Authorization: OAuth 会话UUID
```

举例：

```
Authorization: OAuth 34cbfddd470a47d8bdb0727cd2182618
```

注: OAuth和会话UUID之间用空格分隔。

X-Job-UUID

对于异步API，可以通过X-Job-UUID HTTP Header来指定该API Job的UUID，例如：

```
X-Job-UUID: d825b1a26f4e474b8c59306081920ff2
```

如果未指定该HTTP Header，ZStack会自动为API Job生成一个UUID。

注:

X-Job-UUID必须为一个v4版本的UUID（即随机UUID）字符串去掉连接符-。ZStack会验证X-Job-UUID格式的合法性，并对非法的字符串返回一个400 Bad Request的错误。

X-Web-Hook

对于异步API，可以通过X-Web-Hook HTTP Header指定一个回调URL用于接收API 返回。通过使用回调URL的方法，调用者可以避免使用轮询去查询一个异步API的执行结果。举例：

```
X-Web-Hook: http://localhost:5000/api-callback
```

X-Job-Success

当使用了X-Web-Hook回调的方式获取异步API结果时，ZStack推送给回调URL的HTTP Post请求中会包含X-Job-Success HTTP Header指明该异步API的执行结果是成功还是失败。例如：

```
X-Job-Success: true
```

当值为*true*时执行成功，为*false*时执行失败。

3.4 HTTP返回码 (HTTP Status Code)

ZStack使用如下返回码：

返回码	意义
200	API执行成功。
202	API请求已被ZStack接受，用户需要通过轮询或Web Hook的方式获取API结果。该返回码只在调用异步API时出现。
400	API请求未包含必要的参数或包含了非法的参数。具体信息可以从HTTP Response Body获得。
404	URL不存在，通常是指定了错误的API URL。如果访问的URL是异步API返回的轮询地址，表示该轮询地址已经过期。
405	API调用使用了错误的HTTP方法，例如在创建一个资源的时候用了GET方法而不是POST方法。
500	ZStack RESTful终端遭遇了一个内部错误。
503	API所执行的操作引发了一个错误，例如资源不足无法创建虚拟机。错误的具体信息可以从HTTP Response Body。

3.5 API种类

ZStack的API分为同步API和异步API两种：

同步API

所有使用GET方法的API都是同步API，调用方收到的HTTP Response中直接包含了API的结果。例如：

```
GET zstack/v1/zones/f3fa7671894a40f6a73f5bfc7d90c126

{
   "inventory": {
      "uuid": "f3fa7671894a40f6a73f5bfc7d90c126",
      "name": "zone1",
      "description": "test",
      "state": "Enabled",
      "type": "zstack",
      "createDate": "Jan 6, 2017 3:51:16 AM",
      "lastOpDate": "Jan 6, 2017 3:51:16 AM"
   }
}
```

异步API

除了登录相关的API外，所有不使用GET方法的API都为异步API。用户调用一个异步API成功后会收到202返回码以及 Body中包含的一个轮询地址（location字段），用户需要周期性的GET该轮询地址以获得API的执行结果。例如：

```
Status Code: 202

Body:

{
   "location": "http://localhost:8989/v1/api-jobs/967a26b7431c49c0b1d50d709ef1aef3"
}
```

通常情况下GET一个轮询地址可以得到四种返回：

1. 202返回码表示该API仍在处理中，用户需要继续轮询。
2. 200返回码表示API执行成功，Body中包含API结果。
3. 503返回码表示API执行失败，Body中包含错误码。
4. 404返回码，则表示轮询地址已经过期，产生这种结果的原因可能是用户访问了一个错误的轮询地址，或者太久没有访问该轮询地址（例如超过2天没有访问），该轮询地址已经被删除。

异步API也可以用Web Hook的方式获得结果，具体方法见后面章节。

3.6 API操作

跟所有的RESTful API类似，绝大多数ZStack API执行的是CRUD（Create, Read, Update, Delete）操作，以及类RPC操作。

创建资源

所有资源的创建都使用POST方法，参数通过HTTP Body传递，例如创建一个虚拟机：

```
POST zstack/v1/vm-instances

Authorization： OAuth 0c234e29a2ad4ff4b0d97d4f3b47c6cf

{
    "params": {
        "l3NetworkUuids": ["37a701c7fe4a40758da15593aedd8aff"],
        "defaultL3NetworkUuid": "37a701c7fe4a40758da15593aedd8aff",
        "dataDiskOfferingUuids": [],
        "name": "TestVm",
        "description": "Test",
        "systemTags": [],
        "instanceOfferingUuid": "dd53f94b58924510b0122e40799a4114",
        "type": "UserVm",
        "imageUuid": "cc7b56780879409f98c1f992b75a12b0"
    }
}
```

查询资源

资源的查询使用GET方法，查询条件通过Query String传参，例如查询集群*cluster1*中名字**不等于** *web-vm*的虚拟机：

```
GET zstack/v1/vm-instances?condition=name!=web-vm&condition=cluster.name=cluster1

Authorization： OAuth 0c234e29a2ad4ff4b0d97d4f3b47c6cf
```

如果已知资源的UUID，要直接获取该资源的信息，直接使用GET方法不加任何查询条件，例如：

```
GET zstack/v1/vm-instances/56f0fd314a2647ffb4f9565f6d05858e

Authorization： OAuth 0c234e29a2ad4ff4b0d97d4f3b47c6cf
```

返回UUID为*56f0fd314a2647ffb4f9565f6d05858e*虚拟机的信息。

删除资源

删除资源使用DELETE方法，被删除资源的UUID编码在URL中，例如：

```
DELETE zstack/v1/vm-instances/56f0fd314a2647ffb4f9565f6d05858e

Authorization： OAuth 0c234e29a2ad4ff4b0d97d4f3b47c6cf
```

删除UUID为*56f0fd314a2647ffb4f9565f6d05858e*的虚拟机。

修改资源与类RPC操作

但由于IaaS本身业务的性质，一部分操作更类似于RPC（远程调用）而非CRUD操作，例如启动虚拟机。根据RESTful API的一些最佳实践，ZStack将这些操作都归为资源的actions子资源，例如启动虚拟机、 停止虚拟机都是对虚拟机的actions子资源进行操作。举个例子：

启动虚拟机：

```
PUT zstack/v1/vm-instances/d46841bd4ebd47f8bf0bed85c3bdf0db/actions

{
   "startVmInstance": {}
}
```

停止虚拟机：

```
PUT zstack/v1/vm-instances/d46841bd4ebd47f8bf0bed85c3bdf0db/actions

{
   "stopVmInstance": {}
}
```

在上面的例子中，两个操作都访问的是相同的URL*/v1/vminstances/d46841bd4ebd47f8bf0bed85c3bdf0db/actions*， 具体的操作类型由包含在Body中的字段名表示，例如stopVmInstance，如果该API包含额外参数，则包含在操作字段名对应的map中。

资源操作的具体字段名和例子参考每个API的详细文档。

3.7 基本流程示例

在下例中，我们会创建一个Zone，以展示API使用的基本流程：

登录

使用API的第一步是登录以获取一个Session UUID，以供后续API调用使用。

```
PUT zstack/v1/accounts/login

body:

{
   "logInByAccount": {
      "password": "b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c
706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86",
      "accountName": "admin"
   }
}
```

这里的密码是用sha512哈希后的结果。

API返回如下：

```
status code: 200

body:

{
   "inventory": {
      "uuid": "00d038b699b74e76a01705918d48d939",
      "accountUuid": "36c27e8ff05c4780bf6d2fa65700f22e",
      "userUuid": "36c27e8ff05c4780bf6d2fa65700f22e",
      "expiredDate": "Jan 1, 2017 11:31:06 AM",
      "createDate": "Jan 1, 2017 9:31:06 AM"
   }
}
```

返回内容中包含账户UUID等其他字段，我们需要的session UUID包含在字段uuid中：*00d038b699b74e76a01705918d48d939*。

创建Zone

```
POST zstack/v1/zones

headers:

Authorization: OAuth 00d038b699b74e76a01705918d48d939

body:

{
   "params": {
      "name": "Zone1",
      "description": "Test"
   }
}
```

由于创建Zone操作是一个异步API，API返回不是直接的结果，而是一个轮询地址：

```
status code: 202

body:

{
   "location": "http://localhost:8989/v1/api-jobs/d0345d3ddcae485f8170572b15a2b581"
}
```

用户需要周期性的轮询API结果：

```
GET http://localhost:8989/v1/api-jobs/d0345d3ddcae485f8170572b15a2b581
```

```
Authorization: OAuth 00d038b699b74e76a01705918d48d939
```

如果API还未执行完成，上述GET操作得到的仍然是202返回码和上述轮询地址。当操作完成时，得到的结果如下：

```
status code: 200

body:

{
    "inventory": {
        "uuid": "f52fe55b64094ceb99b3893a238c4931",
        "name": "Zone1",
        "description": "Test",
        "state": "Enabled",
        "type": "zstack",
        "createDate": "Jan 1, 2017 9:31:07 AM",
        "lastOpDate": "Jan 1, 2017 9:31:07 AM"
    }
}
```

查询Zone

要获取创建的Zone的信息，可以用GET查询：

```
GET zstack/v1/zones/f52fe55b64094ceb99b3893a238c4931

Authorization: OAuth 00d038b699b74e76a01705918d48d939
```

返回：

```
status code: 200

body:

{
    "inventory": {
        "uuid": "f52fe55b64094ceb99b3893a238c4931",
        "name": "Zone1",
        "description": "Test",
        "state": "Enabled",
        "type": "zstack",
        "createDate": "Jan 1, 2017 9:31:07 AM",
        "lastOpDate": "Jan 1, 2017 9:31:07 AM"
    }
```

```
}
```

登出

当所有API调用完毕，我们需要对已登录的session进行登出操作：

```
DELETE zstack/v1/accounts/sessions/00d038b699b74e76a01705918d48d939
```

返回

```
status code: 200
```

3.8 Webhook

对于异步API使用轮询的方式查询操作结果是一种低效的方式，为此ZStack提供Webhook的方式主动推送异步API结果给调用者。

要使用Webhook功能，调用者只需在HTTP Headers中指定X-Job-UUID和X-Web-Hook即可。以上面创建Zone为例，使用Webhook的API版本为：

```
POST zstack/v1/zones

headers:

Authorization: OAuth 00d038b699b74e76a01705918d48d939
X-Job-UUID: d0345d3ddcae485f8170572b15a2b581
X-Web-Hook: http://127.0.0.1:8989/rest-webhook

body:

{
    "params": {
        "name": "Zone1",
        "description": "Test"
    }
}
```

API返回仍然是202返回码和一个轮询地址，但调用者无需再轮询。API执行成功后，结果会被推送到

http://127.0.0.1:8989/rest-webhook

```
POST http://127.0.0.1:8989/rest-webhook

headers:

X-Job-Success: true
X-Job-UUID: d0345d3ddcae485f8170572b15a2b581

body:

{
    "inventory": {
        "uuid": "f52fe55b64094ceb99b3893a238c4931",
```

```
        "name": "Zone1",
        "description": "Test",
        "state": "Enabled",
        "type": "zstack",
        "createDate": "Jan 1, 2017 9:31:07 AM",
        "lastOpDate": "Jan 1, 2017 9:31:07 AM"
    }
}
```

推送的结果之中，X-Job-Success指明了API执行成功与否，X-Job-UUID包含值跟API调用时的X-Job-UUID相同，调用者可以对应结果和API。

3.9 查询API

用户可以用GET方法对一个资源进行查询，并且可以像MySQL一样指定多个查询条件、排序方式、选择字段、以及进行跨表查询等等。

支持超过400万个单项查询条件，以及400万阶乘的组合查询条件。

单表查询

例如：

```
GET zstack/v1/vm-instances?q=name=vm1
```

查询名字为*vm1*的虚拟机。

例如：

```
GET zstack/v1/vm-instances?q=name=vm1&q=state=Running
```

查询名字为*vm1*并且状态为*Running*的虚拟机。

这两个例子都是对虚拟机资源本身查询，反应到数据库层面还属于单表查询。

跨表查询

我们可以通过.进行跨表查询。

例如：

```
GET zstack/v1/vm-instances?q=vmNics.ip=192.168.10.100
```

查询IP地址为*192.168.10.100*的虚拟机，这里对虚拟机和网卡两张表进行了跨表查询。

例如：

```
GET zstack/v1/vm-instances?q=host.managementIp=10.10.20.3
```

查询IP为*10.10.20.3*上运行的所有虚拟机。这里对虚拟机和物理机两张表进行了跨表查询。

所有资源的查询API都支持下列参数

名字	类型	位置	描述	可选值	起始版本
q (可选)	List	query	见查询条件。省略该字段将返回所有记录，返回记录数的上限受限于limit字段		0.6
limit (可选)	Integer	query	最多返回的记录数，类似MySQL的limit，默认值1000		0.6
start (可选)	Integer	query	起始查询记录位置，类似MySQL的offset。跟limit配合使用可以实现分页		0.6
count (可选)	Boolean	query	计数查询，相当于MySQL中的count()函数。当设置成true时，API只返回的是满足查询条件的记录数		0.6
groupBy (可选)	String	query	以字段分组，相当于MySQL中的group by关键字。例如groupBy=type		1.9
replyWithCount (可选)	Boolean	query	见分页查询		0.6
sort (可选)	String	query	以字段排序，等同于MySQL中的sort by关键字。必须跟+或者-配合使用，+表示升序，-表示降序，后面跟排序字段名，例如： • sort=+key，根据key进行升序排序 • sort=-key，根据 key进行降序排序	`+`字段名，`-`字段名	0.6
fields (可选)	List	query	指定返回的字段，等同于MySQL中的select字段功能。例如fields=name,uuid，则只返回满足条件记录的name和uuid字段		0.6

查询条件

的查询条件类似于MySQL数据库。

例如：

```
uuid=bfa67f956afb430890aa49db14b85153
totalCapacity>2000
```

```
vmInstanceUuid not null
```

注:

- **字段名、查询操作符、匹配值三者之间不能有任何空格。**
- 例如*uuid = 25506342d1384c07b7342373a57475b9*就是一个错误的查询条件，必须写为*uuid=25506342d1384c07b7342373a57475b9*。

多个查询条件之间是与关系。总共支持10个查询操作符：

1. **=**: 等于，例如：

```
vmInstanceUuid=c4981689088b40f98d2ade2548c323da
```

2. **!=**: 不等于，例如：

```
vmInstanceUuid!=c4981689088b40f98d2ade2548c323da
```

3. **>**: 大于
4. **<**: 小于
5. **>=**: 大于等于
6. **<=**: 小于等于
7. **?=**: **in**操作符，测试字段值是否在一个集合。集合中的值以,分隔。例如测试*uuid*是否属于某个集合：

```
uuid?=25506342d1384c07b7342373a57475b9,bc58d68090ac42358c0cb0fe72e3287f
```

8. **!?=**: **not int**操作符，测试字段值是否**不属于**一个集合。集合中的值以,分隔，例如测试*name*是否不等于VM1和VM2：

```
name!?=VM1,VM2
```

9. **~=**: 字符串模糊匹配，相当于MySQL中的**like**操作。使用**%**匹配一个或多个字符，使用_匹配一个字符。 例如查询一个名字是以*IntelCore*开头的：

```
name~=IntelCore%
```

10. 或者查询一个名字是以*IntelCore*开头，以*7*结尾，中间模糊匹配一个字符：

```
name~=IntelCore_7
```

这样名字是*IntelCoreI7*，*IntelCoreM7*的记录都会匹配上。

11. **!~=**: 模糊匹配非操作。查询一个字段不能模糊匹配到某个字符串，匹配条件与**~=**相同。

12.**is null**: 字段为null：

```
name=null
```

13.**not null**: 字段不为null：

```
name!=null
```

分页查询

start、**limit**、**replyWithCount**三个字段可以配合使用实现分页查询。

- **start**指定起始查询位置。
- **limit**指定查询返回的最大记录数。
- **replyWithCount**被设置成true后，查询返回中会包含满足查询条件的记录总数，跟**start**值比较就可以得知还需几次分页。

例如：

总共有1000记录满足查询条件，使用如下组合：

```
start=0 limit=100 replyWithCount=true
```

则API返回将包含头100条记录，以及total字段等于1000，表示总共满足条件的记录为1000。

获取资源可查询字段

由于支持的查询条件数非常巨大，我们无法在文档中枚举所有的查询条件。

用户可以使用命令行工具*zstack-cli*的自动补全功能来查看一个资源可查询的字段以及可跨表查询的字段。

- 以查询虚拟机为例，在*zstack-cli*里输入QueryVmInstance并按Tab键补全，可以看到提示页面：

```
- >>>QueryVmInstance
[Query Conditions:]
allVolumes.          cluster.             host.                image.               instanceOffering.
 rootVolume.
vmNics.              zone.

__systemTag__=       __userTag__=         allocatorStrategy=   clusterUuid=
 cpuNum=             cpuSpeed=
createDate=          defaultL3NetworkUuid=  description=        groupBy=             hostUuid
=            hypervisorType=
imageUuid=           instanceOfferingUuid=  lastHostUuid=       lastOpDate=
 memorySize=         name=
platform=            rootVolumeUuid=      state=               type=
uuid=                zoneUuid=

[Parameters:]
```

```
count=              fields=             limit=              replyWithCount=     sortBy=
 sortDirection=
start=              timeout=
```

- 这里中间行：

```
__systemTag__=      __userTag__=        allocatorStrategy=  clusterUuid=
 cpuNum=            cpuSpeed=
createDate=         defaultL3NetworkUuid=  description=     groupBy=            hostUuid
=           hypervisorType=
imageUuid=          instanceOfferingUuid=  lastHostUuid=     lastOpDate=
 memorySize=        name=
platform=           rootVolumeUuid=     state=              type=
uuid=               zoneUuid=
```

除**__systemTag__**和**__userTag__**两个特殊查询条件外，其余均为虚拟机表的原生字段，用户可以在API的查询条件里面指定它们，并且可以在**fields**参数中指定这些字段来过滤其它不希望API返回的字段。

例如：

```
GET zstack/v1/vm-instances?q=cpuNum>5
```

返回CPU数量多于5的虚拟机。

```
GET zstack/v1/vm-instances?q=hypervisorType=KVM&fields=uuid&fields=name
```

返回虚拟化类型为KVM的虚拟机，由于在**fields**指定了uuid和name两个字段，API返回中只会包含虚拟机的name和uuid。

注:

只有资源的原生字段可以被**fields**选取，**__userTag__**、**__systemTag__**以及下面讲到的跨表字段均不能出现在**fields**参数中。

- 提示的第一行：

```
allVolumes.         cluster.            host.               image.              instanceOffering.
 rootVolume.
vmNics.             zone.
```

指明了虚拟机资源可以跟哪些资源做跨表查询，例如：**allVolumes**代表云盘，**cluster**代表集群，**vmNics**代表网卡等。

如需查看这些资源的具体字段，只需输入资源名加.号，并按Tab键补全。

例如：

```
- >>>QueryVmInstance vmNics.
```

```
[Query Conditions:]
vmNics.eip.                vmNics.l3Network.          vmNics.loadBalancerListener.  vmNics
.portForwarding.       vmNics.securityGroup.
vmNics.vmInstance.

vmNics.__systemTag__=        vmNics.__userTag__=          vmNics.createDate=
 vmNics.deviceId=           vmNics.gateway=
vmNics.ip=                 vmNics.l3NetworkUuid=        vmNics.lastOpDate=          vmNics
.mac=                vmNics.metaData=
vmNics.netmask=            vmNics.uuid=               vmNics.vmInstanceUuid=
```

这里我们输入了资源**vmNics**并用**.**号表示我们要做一个跨表查询，Tab键为我们补全了**vmNics**资源的原生字段以及可跨表查询的其它资源。

- 例如**vmNics.ip**表示网卡的原生字段**ip**：

  ```
  GET zstack/v1/vm-instances?q=vmNics.ip=192.168.0.100
  ```

  进行了一个跨表查询，条件是网卡表的**ip**字段，返回的结果是**ip**为*192.168.0.100*的虚拟机。

- 网卡资源同样可以跟其它资源进行跨表查询，例如**vmNics.eip**。

  将网卡表和EIP表进行跨表：

  ```
  GET zstack/v1/vm-instances?q=vmNics.eip.ip=192.168.0.100
  ```

  进行了跨3表查询，返回的是EIP为*192.168.0.100*的虚拟机。

- 通过资源间连续跨表，一个资源几乎跟系统中多个有逻辑关系的资源进行跨表，例如：

```
- >>>QueryVmInstance zone.cluster.l2Network.l3Network.
[Query Conditions:]
zone.cluster.l2Network.l3Network.ipRanges.        zone.cluster.l2Network.l3Network.l2Netwo
rk.       zone.cluster.l2Network.l3Network.networkServices.
zone.cluster.l2Network.l3Network.serviceProvider.  zone.cluster.l2Network.l3Network.vmNic.
          zone.cluster.l2Network.l3Network.zone.


zone.cluster.l2Network.l3Network.__systemTag__=    zone.cluster.l2Network.l3Network
.__userTag__=      zone.cluster.l2Network.l3Network.createDate=
zone.cluster.l2Network.l3Network.description=      zone.cluster.l2Network.l3Network
.dnsDomain=        zone.cluster.l2Network.l3Network.l2NetworkUuid=
zone.cluster.l2Network.l3Network.lastOpDate=       zone.cluster.l2Network.l3Network.name=
           zone.cluster.l2Network.l3Network.state=
zone.cluster.l2Network.l3Network.system=           zone.cluster.l2Network.l3Network.type=
       zone.cluster.l2Network.l3Network.uuid=
zone.cluster.l2Network.l3Network.zoneUuid=
```

分别跟**zone**, **cluster**, **l2Network**, **l3Network**多个资源进行跨表。

注:

- 由于一个资源的逻辑关系存在环路，因此会存在环路路径。例如：以云主机为查询主体可以跟网卡进行跨表查询（例如：`QueryVmInstance vmNics.`），同时以网卡为主

体也可以跟云主机进行跨表查询（例如：QueryVmNic vmInstance.），这样就会存在环路路径。

- 使用中应该避免环路跨表查询。例如 QueryVmInstance vmNics.vmInstance.name =vm1通过跨表查询了name=vm1的云主机，它的实际效果跟QueryVmInstance name =vm1完全等同。这里的跨表是无意义的，只会生产复杂的SQL语句导致低效的数据库查询。

- **__systemTag__**和**__userTag__**是两个特殊的查询条件，允许用户通过tag查询资源。

  例如：

  ```
  QueryVmInstance __systemTag__=staticIp:10.10.1.20
  ```

  查询具有*staticIp:10.10.1.20*这个tag的虚拟机。

4 系统全局相关

4.1 管理节点相关接口

4.1.1 查询管理节点(QueryManagementNode)

API请求

URLs

```
GET zstack/v1/management-nodes
GET zstack/v1/management-nodes/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth f5db0d0f90104a79b02c8f1c34dfc13b" \
-X GET http://localhost:8080/zstack/v1/management-nodes?q=uuid=2c0761c3720f4d2c93435f1b8edb1297
```

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 5ffa91907f3c4d0392d7daa73b1a0627" \
-X GET http://localhost:8080/zstack/v1/management-nodes/bc27558979364d7c9b34a7317965e0be
```

可查询字段

运行*zstack-cli*命令行工具，输入QueryManagementNode并按Tab键查看所有可查询字段以及可跨表查询的资源名

API返回

返回示例

```
{
  "inventories": [
    {
      "uuid": "188b2c6376c34d3687c5a7a19685ffa6",
      "hostName": "127.0.0.1",
      "joinDate": "May 11, 2017 1:22:44 PM",
      "heartBeat": "May 11, 2017 1:22:44 PM"
    }
  ]
```

```
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventories	List	详情参考*inventories*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventories

名字	类型	描述	起始版本
uuid	String	资源的UUID，唯一标示该资源	0.6
hostName	String	宿主机名称	0.6
joinDate	Timestamp	加入时间	0.6
heartBeat	Timestamp	心跳时间	0.6

SDK示例

Java SDK

```
ueryManagementNodeAction action = new QueryManagementNodeAction();
action.conditions = asList("uuid=7c2075262bb44872b6ceaef5ee94b089");
```

```
action.sessionId = "293267e661d8400ca3808ff5f445a6e8";
QueryManagementNodeAction.Result res = action.call();
```

Python SDK

```
QueryManagementNodeAction action = QueryManagementNodeAction()
action.conditions = ["uuid=3bcbd030e10746eeba9f8ea938f2647a"]
action.sessionId = "21526cb97ac541b6b27192b4c671ec08"
QueryManagementNodeAction.Result res = action.call()
```

4.1.2 获取当前版本(GetVersion)

API请求

URLs

```
PUT zstack/v1/management-nodes/actions
```

Body

```
{
  "getVersion": {},
  "systemTags": [],
  "userTags": []
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-X PUT -d '{"getVersion":{}}' http://localhost:8080/zstack/v1/management-nodes/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
systemTags (可选)	List	body	系统标签		0.6
userTags (可选)	List	body	用户标签		0.6

API返回

返回示例

```
{
  "version": "1.9.x"
```

```
}
```

名字	类型	描述	起始版本
version	String	管理节点当前版本	0.6
success	boolean	成功标志	0.6
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

SDK示例

Java SDK

```
GetVersionAction action = new GetVersionAction();
GetVersionAction.Result res = action.call();
```

Python SDK

```
GetVersionAction action = GetVersionAction()
```

```
GetVersionAction.Result res = action.call()
```

4.1.3 获取当前时间(GetCurrentTime)

API请求

URLs

```
PUT zstack/v1/management-nodes/actions
```

Body

```
{
  "getCurrentTime": {},
  "systemTags": [],
  "userTags": []
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-X PUT -d '{"getCurrentTime":{}}' http://localhost:8080/zstack/v1/management-nodes/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
systemTags (可选)	List	body	系统标签		0.6
userTags (可选)	List	body	用户标签		0.6

API返回

返回示例

```
{
  "currentTime": {
    "MillionSeconds": 1.494480176967E12,
    "Seconds": 1.494480176E9
  }
}
```

名字	类型	描述	起始版本
currentTime	Map	管理节点当前时间	0.6

名字	类型	描述	起始版本
success	boolean	成功标志	0.6
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

SDK示例

Java SDK

```
GetCurrentTimeAction action = new GetCurrentTimeAction();
GetCurrentTimeAction.Result res = action.call();
```

Python SDK

```
GetCurrentTimeAction action = GetCurrentTimeAction()
```

```
GetCurrentTimeAction.Result res = action.call()
```

4.1.4 检查管理节点是否能正常开始工作(IsReadyToGo)

API请求

URLs

```
GET zstack/v1/management-nodes/ready
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 61f9cb0129d749ebbd3d6052bfc759b2" \
-X GET http://localhost:8080/zstack/v1/management-nodes/ready?
```

参数列表

名字	类型	位置	描述	可选值	起始版本
managementNodeId (可选)	String	query			0.6
systemTags (可选)	List	query	系统标签		0.6
userTags (可选)	List	query	用户标签		0.6

API返回

返回示例：

```
{
  "managementNodeId": "28ff5884cbf83d48b2e50203a5f2e636"
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

SDK示例

Java SDK

```
IsReadyToGoAction action = new IsReadyToGoAction();
action.sessionId = "4feaf1772217415a876735801e468fac";
IsReadyToGoAction.Result res = action.call();
```

Python SDK

```
IsReadyToGoAction action = IsReadyToGoAction()
action.sessionId = "432b9a5c589c4c09b33b79d8272528be"
```

```
IsReadyToGoAction.Result res = action.call()
```

4.2 标签相关接口

4.2.1 创建系统标签(CreateSystemTag)

API请求

URLs

```
POST zstack/v1/system-tags
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "params": {
    "resourceType": "HostVO",
    "resourceUuid": "c31d63c6d40d489e87fa7e1ee5707d47",
    "tag": "reservedMemory::1G"
  },
  "systemTags": [],
  "userTags": []
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X POST -d '{"params":{"resourceType":"HostVO","resourceUuid":"47609e353c6f331f93ac1
14d06379f8d","tag":"reservedMemory::1G"}}' \
http://localhost:8080/zstack/v1/system-tags
```

参数列表

名字	类型	位置	描述	可选值	起始版本
resourceType	String	body(包含在**params**结构中)	当创建一个标签时, 用户必须制定标签所关联的资源类型		0.6

名字	类型	位置	描述	可选值	起始版本
resourceUuid	String	body(包含在**params**结构中)	用户指定的资源UUID，若指定，系统不会为该资源随机分配UUID		0.6
tag	String	body(包含在**params**结构中)	标签字符串		0.6
systemTags (可选)	List	body	系统标签		0.6
userTags (可选)	List	body	用户标签		0.6

API返回

返回示例

```
{
  "inventory": {
    "inherent": false,
    "uuid": "55d86810cb564ecc934fa79d498b9dc3",
    "resourceType": "HostVO",
    "tag": "reservedMemory::1G",
    "type": "System",
    "createDate": "Apr 24, 2017 7:10:55 PM",
    "lastOpDate": "Apr 24, 2017 7:10:55 PM"
  }
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventory	SystemTagInventory	详情参考*inventory*	0.6

#error

名字	类型	描述	起始版本
ode	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventory

名字	类型	描述	起始版本
inherent	Boolean	内部系统标签	0.6
uuid	String	资源的UUID，唯一标示该资源	0.6
resourceUuid	String	用户指定的资源UUID，若指定，系统不会为该资源随机分配UUID	0.6
resourceType	String	当创建一个标签时, 用户必须制定标签所关联的资源类型(resource type)	0.6
tag	String	标签字符串	0.6
type	String	保留域, 请不要使用它	0.6
createDate	Timestamp	创建时间	0.6
lastOpDate	Timestamp	最后一次修改时间	0.6

SDK示例

Java SDK

```
CreateSystemTagAction action = new CreateSystemTagAction();
action.resourceType = "HostVO";
action.resourceUuid = "c63dfecc5c3f4f24bdab3eda92036eef";
```

```
action.tag = "reservedMemory::1G";
action.sessionId = "494209d421304d1cb9b8aaab7cac3a45";
CreateSystemTagAction.Result res = action.call();
```

Python SDK

```
CreateSystemTagAction action = CreateSystemTagAction()
action.resourceType = "HostVO"
action.resourceUuid = "d8f0bf84c4bc41c99ff7129a369515c1"
action.tag = "reservedMemory::1G"
action.sessionId = "f84ee562742b4aaab791842d71f5d472"
CreateSystemTagAction.Result res = action.call()
```

4.2.2 查询系统标签(QuerySystemTag)

API请求

URLs

```
GET zstack/v1/system-tags
GET zstack/v1/system-tags/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 67bf58bea6854e568556b0e9f4bc84f6" \
-X GET http://localhost:8080/zstack/v1/system-tags?q=inherent=true&q=resourceType=HostVO
```

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth c798e7031aaa430386636f9ff8d6bb17" \
-X GET http://localhost:8080/zstack/v1/system-tags/7c8162efed2840c08a43e0afdcebf01b
```

可查询字段

运行*zstack-cli*命令行工具，输入QuerySystemTag并按Tab键查看所有可查询字段以及可跨表查询的资源名。

API返回

返回示例

```
{
  "inventories": [
    {
      "inherent": false,
      "uuid": "0bba7cb618314646aea4e70b9be3bfea",
      "resourceType": "HostVO",
      "tag": "reservedMemory::1G",
      "type": "System",
```

```
      "createDate": "May 11, 2017 1:22:40 PM",
      "lastOpDate": "May 11, 2017 1:22:40 PM"
    }
  ]
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventories	List	详情参考*inventories*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventories

名字	类型	描述	起始版本
inherent	Boolean	内部系统标签	0.6
uuid	String	资源的UUID，唯一标示该资源	0.6
resourceUuid	String	用户指定的资源UUID，若指定，系统不会为该资源随机分配UUID	0.6
resourceType	String	当创建一个标签时, 用户必须制定标签所关联	0.6

名字	类型	描述	起始版本
		的资源类型(resource type)	
tag	String	标签字符串	0.6
type	String	保留域, 请不要使用它	0.6
createDate	Timestamp	创建时间	0.6
lastOpDate	Timestamp	最后一次修改时间	0.6

SDK示例

Java SDK

```
QuerySystemTagAction action = new QuerySystemTagAction();
action.conditions = asList("inherent=true","resourceType=HostVO");
action.sessionId = "05c74fc02e574e678a407f0ccef8ae5f";
QuerySystemTagAction.Result res = action.call();
```

Python SDK

```
QuerySystemTagAction action = QuerySystemTagAction()
action.conditions = ["inherent=true","resourceType=HostVO"]
action.sessionId = "a1d2f95577514d0a918ce1502dc27e33"
QuerySystemTagAction.Result res = action.call()
```

4.2.3 更新系统标签（UpdateSystemTag）

API请求

URLs

```
PUT zstack/v1/system-tags/{uuid}/actions
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "updateSystemTag": {
    "tag": "for-large-DB"
  },
  "systemTags": [],
  "userTags": []
```

```
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X PUT -d '{"updateSystemTag":{"tag":"for-large-DB"}}' \
http://localhost:8080/zstack/v1/system-tags/2cb4ac52794535c3ae61f9612c1579e7/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuid	String	url	资源的UUID，唯一标示该资源		0.6
tag	String	body(包含在**updateSystemTag**结构中)	标签字符串		0.6
systemTags (可选)	List	body	系统标签		0.6
userTags (可选)	List	body	用户标签		0.6

API返回

返回示例

```
{
  "inventory": {
    "inherent": false,
    "uuid": "c17377dfd0794521a4fe02b6eb16cc5a",
    "resourceType": "HostVO",
    "tag": "reservedMemory::1G",
    "type": "System",
    "createDate": "May 11, 2017 1:22:30 PM",
    "lastOpDate": "May 11, 2017 1:22:30 PM"
  }
```

```
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventory	SystemTagInventory	详情参考*inventory*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventory

名字	类型	描述	起始版本
inherent	Boolean	内部系统标签	0.6
uuid	String	资源的UUID，唯一标示该资源	0.6
resourceUuid	String	用户指定的资源UUID，若指定，系统不会为该资源随机分配UUID	0.6
resourceType	String	当创建一个标签时, 用户必须制定标签所关联的资源类型(resource type)	0.6

名字	类型	描述	起始版本
tag	String	标签字符串	0.6
type	String	保留域, 请不要使用它	0.6
createDate	Timestamp	创建时间	0.6
lastOpDate	Timestamp	最后一次修改时间	0.6

SDK示例

Java SDK

```
UpdateSystemTagAction action = new UpdateSystemTagAction();
action.uuid = "1ba32bc1af0446b29bd969029a6052a5";
action.tag = "for-large-DB";
action.sessionId = "367f6162107a47d58c471c9164139fab";
UpdateSystemTagAction.Result res = action.call();
```

Python SDK

```
UpdateSystemTagAction action = UpdateSystemTagAction()
action.uuid = "be98b2f4e342485486cfa612eeabcaa5"
action.tag = "for-large-DB"
action.sessionId = "6b7af956a9a9477eba02753417380070"
UpdateSystemTagAction.Result res = action.call()
```

4.2.4 创建用户标签(CreateUserTag)

API请求

URLs

```
POST zstack/v1/user-tags
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "params": {
    "resourceType": "DiskOfferingVO",
    "resourceUuid": "beff527f6ccb45c8b215c59434b2fa5c",
    "tag": "for-large-DB"
  },
  "systemTags": [],
  "userTags": []
```

```
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X POST -d '{"params":{"resourceType":"DiskOfferingVO","resourceUuid":"0a93a8d099a933
9bb21b6e14a4ca3eea","tag":"for-large-DB"}}' \
http://localhost:8080/zstack/v1/user-tags
```

参数列表

名字	类型	位置	描述	可选值	起始版本
resourceType	String	body(包含在params结构中)	当创建一个标签时, 用户必须制定标签所关联的资源类型		0.6
resourceUuid	String	body(包含在params结构中)	用户指定的资源UUID，若指定，系统不会为该资源随机分配UUID		0.6
tag	String	body(包含在params结构中)	标签字符串		0.6
systemTags (可选)	List	body	系统标签		0.6
userTags (可选)	List	body	用户标签		0.6

API返回

返回示例

```
{
  "inventory": {
    "uuid": "b97c64cbf0524f809c05bf9eaf262b79",
    "resourceType": "DiskOfferingVO",
    "tag": "for-large-DB",
    "type": "User",
    "createDate": "May 11, 2017 1:22:33 PM",
    "lastOpDate": "May 11, 2017 1:22:33 PM"
```

```
  }
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventory	UserTagInventory	详情参考*inventory*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventory

名字	类型	描述	起始版本
uuid	String	资源的UUID，唯一标示该资源	0.6
resourceUuid	String	用户指定的资源UUID，若指定，系统不会为该资源随机分配UUID	0.6
resourceType	String	当创建一个标签时, 用户必须制定标签所关联的资源类型(resource type)	0.6
tag	String	标签字符串	0.6

名字	类型	描述	起始版本
type	String	保留域, 请不要使用它	0.6
createDate	Timestamp	创建时间	0.6
lastOpDate	Timestamp	最后一次修改时间	0.6

SDK示例

Java SDK

```
CreateUserTagAction action = new CreateUserTagAction();
action.resourceType = "DiskOfferingVO";
action.resourceUuid = "824246fa21d8405fbd48ea1cc7dbf049";
action.tag = "for-large-DB";
action.sessionId = "5d9cde1f7e6a43f586ec78dedf743201";
CreateUserTagAction.Result res = action.call();
```

Python SDK

```
CreateUserTagAction action = CreateUserTagAction()
action.resourceType = "DiskOfferingVO"
action.resourceUuid = "11f5e230f43d49e69815c35fbd39bad5"
action.tag = "for-large-DB"
action.sessionId = "6301b756253c4103adbf8c891d858e5f"
CreateUserTagAction.Result res = action.call()
```

4.2.5 查询用户标签(QueryUserTag)

API请求

URLs

```
GET zstack/v1/user-tags
GET zstack/v1/user-tags/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X GET http://localhost:8080/zstack/v1/user-tags?q=resourceType=DiskOfferingVO&q=tag=for-large-DB
```

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
```

```
-X GET http://localhost:8080/zstack/v1/user-tags/e2a3aa3463dd382fb9e434304167f331
```

可查询字段

运行*zstack-cli*命令行工具，输入QueryUserTag并按Tab键查看所有可查询字段以及可跨表查询的资源名。

API返回

返回示例

```
{
  "inventories": [
    {
      "uuid": "ae4f2dd05a513e1e8d350d448c2071a9",
      "resourceType": "DiskOfferingVO",
      "tag": "for-large-DB",
      "type": "User",
      "createDate": "Nov 14, 2017 10:20:57 PM",
      "lastOpDate": "Nov 14, 2017 10:20:57 PM"
    }
  ]
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventories	List	详情参考*inventories*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6

名字	类型	描述	起始版本
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventories

名字	类型	描述	起始版本
uuid	String	资源的UUID，唯一标示该资源	0.6
resourceUuid	String	用户指定的资源UUID，若指定，系统不会为该资源随机分配UUID	0.6
resourceType	String	当创建一个标签时, 用户必须制定标签所关联的资源类型(resource type)	0.6
tag	String	标签字符串	0.6
type	String	保留域, 请不要使用它	0.6
createDate	Timestamp	创建时间	0.6
lastOpDate	Timestamp	最后一次修改时间	0.6

SDK示例

Java SDK

```
QueryUserTagAction action = new QueryUserTagAction();
action.conditions = asList("resourceType=DiskOfferingVO","tag=for-large-DB");
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
QueryUserTagAction.Result res = action.call();
```

Python SDK

```
QueryUserTagAction action = QueryUserTagAction()
action.conditions = ["resourceType=DiskOfferingVO","tag=for-large-DB"]
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
```

```
QueryUserTagAction.Result res = action.call()
```

4.2.6 删除标签(DeleteTag)

API请求

URLs

```
DELETE zstack/v1/tags/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 7a2a51da1758430382624e00144e70c5" \
-X DELETE http://localhost:8080/zstack/v1/tags/1d33d909355247508cdbf9a0c0037dbe
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuid	String	url	资源的UUID，唯一标示该资源		0.6
deleteMode (可选)	String	body	当设置成Permissive时, 如果删除过程中发生错误或者删除不被允许ZStack会停止删除操作; 在这种情况下, 包含失败原因的错误代码会被返回.当设置成Enforcing, ZStack会忽略所有错误和权限而直接删除资源; 在这种情况下, 删除操作总是会成功.		0.6

名字	类型	位置	描述	可选值	起始版本
systemTags (可选)	List	body	系统标签		0.6
userTags (可选)	List	body	用户标签		0.6

API返回

该API成功时返回一个空的JSON结构**{}**，出错时返回的JSON结构包含一个error字段，例如：

```
{
    "error": {
        "code": "SYS.1001",
        "description": "A message or a operation timeout",
        "details": "Create VM on KVM timeout after 300s"
    }
}
```

SDK示例

Java SDK

```
DeleteTagAction action = new DeleteTagAction();
action.uuid = "8262e178c23140f6a4de6f0d9e6a0913";
action.deleteMode = "Permissive";
action.sessionId = "94c2bff91f834a96b691a36e47888967";
DeleteTagAction.Result res = action.call();
```

Python SDK

```
DeleteTagAction action = DeleteTagAction()
action.uuid = "86a3f5e633dd40358cab072951058ec9"
action.deleteMode = "Permissive"
action.sessionId = "87c504af0b5a484d87a5ad4ddac57baa"
```

```
DeleteTagAction.Result res = action.call()
```

4.3 进度条相关接口

4.3.1 获取任务进度(GetTaskProgress)

API请求

URLs

```
GET zstack/v1/task-progresses/{apiId}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 8679db834cfc4b3b93fcbf8521581eff" \
-X GET http://localhost:8080/zstack/v1/task-progresses/7b016e3d2b4647e9abe3183728f77e57
?all=false
```

参数列表

名字	类型	位置	描述	可选值	起始版本
apiId (可选)	String	url	任务对应的API ID		1.11
all (可选)	boolean	query	指定获取所有进度信息		1.11
systemTags (可选)	List	query	系统标签		1.11
userTags (可选)	List	query	用户标签		1.11

API返回

返回示例

```
{
  "inventories": [
    {
      "taskUuid": "931102503f64436ea649939ff3957406",
      "taskName": "org.zstack.header.vm.APICreateVmInstanceMsg",
      "type": "Task",
      "content": "Choose backup storage for downloading the image",
      "time": 1.510669257141E12
```

```
    }
  ]
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventories	List	详情参考*inventories*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventories

名字	类型	描述	起始版本
taskUuid	String		0.6
taskName	String		0.6
parentUuid	String		0.6
type	String		0.6
content	String		0.6
opaque	LinkedHashMap		0.6
time	Long		0.6
totalSteps	Integer		0.6

名字	类型	描述	起始版本
currentStep	Integer		0.6
subTasks	List	详情参考subTasks	0.6

SDK示例

Java SDK

```
GetTaskProgressAction action = new GetTaskProgressAction();
action.apiId = "00c035b586634edeb3bfb21f17e02b40";
action.all = false;
action.sessionId = "dea874a205c04a838e46ff6f48294add";
GetTaskProgressAction.Result res = action.call();
```

Python SDK

```
GetTaskProgressAction action = GetTaskProgressAction()
action.apiId = "82f5efb211394297b1d6a4f8d5e5618e"
action.all = false
action.sessionId = "2100b95ab4064c81acc5db473b2a1ef6"
GetTaskProgressAction.Result res = action.call()
```

4.4 通知相关接口

4.4.1 删除通知(DeleteNotifications)

API请求

URLs

```
DELETE zstack/v1/notifications
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 5d6f74cc325a45128d28d695f334ad78" \
-X DELETE http://localhost:8080/zstack/v1/notifications
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuids	List	body			0.6

名字	类型	位置	描述	可选值	起始版本
systemTags (可选)	List	body			0.6
userTags (可选)	List	body			0.6

API返回

该API成功时返回一个空的JSON结构**{}**，出错时返回的JSON结构包含一个error字段，例如：

```
{
    "error": {
        "code": "SYS.1001",
        "description": "A message or a operation timeout",
        "details": "Create VM on KVM timeout after 300s"
    }
}
```

SDK示例

Java SDK

```
DeleteNotificationsAction action = new DeleteNotificationsAction();
action.uuids = asList("611e490aab314728b4f3c85c55f2eeca","fbe67eb0ec6c4624b113d
9666ca84610");
action.sessionId = "7d6826cd212540ee8cd52f57f41b7972";
DeleteNotificationsAction.Result res = action.call();
```

Python SDK

```
DeleteNotificationsAction action = DeleteNotificationsAction()
action.uuids = [3fe69c39933b48229d93f4f3c1eb5622, 758e0f2a8bfa4f86bce6a84ba39d954b]
action.sessionId = "472ea19a14fc482cafcaa6c1c272c9e4"
DeleteNotificationsAction.Result res = action.call()
```

4.4.2 查询通知(QueryNotification)

API请求

URLs

```
GET zstack/v1/notifications
```

```
GET zstack/v1/notifications/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 618e74352186444ea957eac5f9fc0bd4" \
-X GET http://localhost:8080/zstack/v1/notifications
```

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 3cf7403d6ee4474d90bae269f1cb4264" \
-X GET http://localhost:8080/zstack/v1/notifications/f340911c46604b27a92706dd141e688d
```

可查询字段

运行*zstack-cli*命令行工具，输入QueryNotification并按Tab键查看所有可查询字段以及可跨表查询的资源名。

API返回

返回示例：

```
{
  "inventories": [
    {
      "name": "system",
      "content": "the host[uuid:%s] becomes Disconnected, change the VM[uuid:%s]\u0027 state
 to Unknown",
      "arguments": "[\"5da0cef1d7714a059028d786dba8cca7\",\"25c7f03430dc467090c12
1a82b4afd7a\"]",
      "resourceType": "VmInstanceVO",
      "type": "Info",
      "time": 1.510669257141E12
    }
  ]
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventories	List	详情参考*inventories*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventories

名字	类型	描述	起始版本
uuid	String	资源的UUID，唯一标示该资源	0.6
name	String	资源名称	0.6
content	String		0.6
arguments	String		0.6
sender	String		0.6
status	String		0.6
resourceUuid	String		0.6
resourceType	String		0.6
type	String		0.6
time	Long		0.6
opaque	Object		0.6
createDate	Timestamp	创建时间	0.6
lastOpDate	Timestamp	最后一次修改时间	0.6

SDK示例

Java SDK

```
QueryNotificationAction action = new QueryNotificationAction();
```

```
action.conditions = asList();
action.sessionId = "4d7e5e3ef007455cbdfe32c55ca76b08";
QueryNotificationAction.Result res = action.call();
```

Python SDK

```
QueryNotificationAction action = QueryNotificationAction()
action.conditions = []
action.sessionId = "7b39f7acff3f4aefbecc7f07bfa4bd04"
QueryNotificationAction.Result res = action.call()
```

4.4.3 更新通知状态(UpdateNotificationsStatus)

API请求

URLs

```
PUT zstack/v1/notifications/actions
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "updateNotificationsStatus": {
    "uuids": [
      "43edf16acb684fbda048a9a4c9b88817",
      "e47ee64a55944d7ca099cbce03dc3a5d"
    ],
    "status": "Read"
  },
  "systemTags": [],
  "userTags": []
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X PUT -d '{"updateNotificationsStatus":{"uuids":["19cd73470a153e9f9254b46c5c996747
","4a2a9b427e3431b08cf44a7cd226f7a2"],"status":"Read"}}' \
http://localhost:8080/zstack/v1/notifications/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuids	List	body(包含在**updateNotificationsStatus**结构中)			0.6
status	String	body(包含在**updateNotificationsStatus**结构中)		• Unread • Read	0.6
systemTags (可选)	List	body			0.6
userTags (可选)	List	body			0.6

API返回

该API成功时返回一个空的JSON结构**{}**，出错时返回的JSON结构包含一个error字段，例如：

```
{
    "error": {
        "code": "SYS.1001",
        "description": "A message or a operation timeout",
        "details": "Create VM on KVM timeout after 300s"
    }
}
```

SDK示例

Java SDK

```
UpdateNotificationsStatusAction action = new UpdateNotificationsStatusAction();
action.uuids = asList("c6df11e283fd4cd18e799d6622871de0","f20ada22f564439f82818
7a60ea4f6a8");
action.status = "Read";
action.sessionId = "ac609aa4d6b14ae998d10bf6ec640d6c";
UpdateNotificationsStatusAction.Result res = action.call();
```

Python SDK

```
UpdateNotificationsStatusAction action = UpdateNotificationsStatusAction()
action.uuids = [90c0befea30743fc92d9a29339a717e6, e1787c41513544f1920e758e9f84c32c]
action.status = "Read"
action.sessionId = "5b1bfa06f56044eba3231c026486887f"
```

```
UpdateNotificationsStatusAction.Result res = action.call()
```

4.5 查询可用资源相关接口

4.5.1 获取cpu和内存容量(GetCpuMemoryCapacity)

API请求

URLs

```
GET zstack/v1/hosts/capacities/cpu-memory
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 36db618d277b4ce6bd9ea7ea858f1a93" \
-X GET http://localhost:8080/zstack/v1/hosts/capacities/cpu-memory?zoneUuids=c642bf6
d4d6b4e8c860d822cc5124199&zoneUuids=f948c5fe5f774ae98816a86ecdf27c5d&cluster
Uuids=b7ba13b36bb04f78a621657c90b2c055&clusterUuids=65997a1a65e147cfa4479a650f66
2898&hostUuids=1382152e1ecc4ae88aaddfc13a021df3&hostUuids=338f3e9fc89a4d799aa01
fea706b633c&all=true
```

参数列表

名字	类型	位置	描述	可选值	起始版本
zoneUuids (可选)	List	query	区域的UUID		0.6
clusterUuids (可选)	List	query	集群的UUID。用于挂载网络、存储等		0.6
hostUuids (可选)	List	query	物理主机的UUID。用于添加、删除host等		0.6
all (可选)	boolean	query	• all默认为false，此时区域UUID，集群UUID		0.6

名字	类型	位置	描述	可选值	起始版本
			,物理主机UUID必须有一个不为空，或者全部都填写 • all设置为true时，区域UUID、集群UUID、物理主机UUID均可不填		
systemTags (可选)	List	query	系统标签		0.6
userTags (可选)	List	query	用户标签		0.6

API返回

返回示例

```
{
  "totalCpu": 4.0,
  "availableCpu": 2.0,
  "totalMemory": 8.0,
  "availableMemory": 4.0
}
```

名字	类型	描述	起始版本
totalCpu	long	cpu总数	0.6
availableCpu	long	可用cpu数量	0.6
totalMemory	long	内存总量	0.6
availableMemory	long	可用内存	0.6
success	boolean	成功	0.6
error	ErrorCode	错误码，若不为null，则表示操作失	0.6

名字	类型	描述	起始版本
		败, 操作成功时该字段为null。 详情参考*error*	

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

SDK示例

Java SDK

```
GetCpuMemoryCapacityAction action = new GetCpuMemoryCapacityAction();
action.zoneUuids = asList("79383afd88f54a0b994515d84dbfa447","907c8ca71ee440e3bb0327f19d60754a");
action.clusterUuids = asList("9f5431123b834d95ae642c3cb923ba32","09cdcc7e28f340ccace7c427a32e2c05");
action.hostUuids = asList("2926839176934652ac3c6a77e4ce1484","c1b46284d0dd4d2a8a7637f0d4792d31");
action.all = true;
action.sessionId = "f3aedb623b3c47a7bb215ea3f4be90d7";
GetCpuMemoryCapacityAction.Result res = action.call();
```

Python SDK

```
GetCpuMemoryCapacityAction action = GetCpuMemoryCapacityAction()
action.zoneUuids = [28cd4ea55a72429d815205b0738b87b3, 2571613b0ba642b7a1c99ddc17aee8e3]
action.clusterUuids = [3e9e56c96fa545fea0b759cb601aad1c, 98ec43d31af446a9bf290b117af95e52]
action.hostUuids = [b5cf96f88a1943f89417d6bea0715b6c, b358d13ac15049a581e47c76183c1358]
action.all = true
action.sessionId = "b0b7e398d33d4d7893255547d0859731"
```

```
GetCpuMemoryCapacityAction.Result res = action.call()
```

4.6 垃圾回收相关接口

4.6.1 触发垃圾回收任务(TriggerGCJob)

API请求

URLs

```
PUT zstack/v1/gc-jobs/{uuid}/actions
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "triggerGCJob": {},
  "systemTags": [],
  "userTags": []
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X PUT -d '{"triggerGCJob":{}}' \
http://localhost:8080/zstack/v1/gc-jobs/1450275537b53cfeba9e6ac79ebadb16/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuid	String	url	资源的UUID，唯一标示该资源		0.6
systemTags (可选)	List	body			0.6
userTags (可选)	List	body			0.6

API返回

该API成功时返回一个空的JSON结构**{}**，出错时返回的JSON结构包含一个error字段，例如：

```
{
   "error": {
      "code": "SYS.1001",
      "description": "A message or a operation timeout",
      "details": "Create VM on KVM timeout after 300s"
   }
}
```

SDK示例

Java SDK

```
TriggerGCJobAction action = new TriggerGCJobAction();
action.uuid = "ff4eb99f62364bcab4cbcebe4dea7de7";
action.sessionId = "7a51dbf52d5d41b0a877491143b6937a";
TriggerGCJobAction.Result res = action.call();
```

Python SDK

```
TriggerGCJobAction action = TriggerGCJobAction()
action.uuid = "a1d1ab14128e4f7398924591915ac65c"
action.sessionId = "6dcc50cb3dc5401b8ce03540a225f623"
TriggerGCJobAction.Result res = action.call()
```

4.6.2 删除垃圾回收任务(DeleteGCJob)

API请求

URLs

```
DELETE zstack/v1/gc-jobs/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth ea06449412a24699955ea802430000f3" \
-X DELETE http://localhost:8080/zstack/v1/gc-jobs/3a5d968dd8d743c280888a4973ea71d5
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuid	String	url	资源的UUID，唯一标示该资源		0.6
systemTags (可选)	List	body			0.6
userTags (可选)	List	body			0.6

API返回

该API成功时返回一个空的JSON结构**{}**，出错时返回的JSON结构包含一个error字段，例如：

```
{
    "error": {
        "code": "SYS.1001",
        "description": "A message or a operation timeout",
        "details": "Create VM on KVM timeout after 300s"
    }
}
```

SDK示例

Java SDK

```
DeleteGCJobAction action = new DeleteGCJobAction();
action.uuid = "6bac0ebceac44a198609b0019c40a8eb";
action.sessionId = "78e46c7bfe0249fe9613223642222085";
DeleteGCJobAction.Result res = action.call();
```

Python SDK

```
DeleteGCJobAction action = DeleteGCJobAction()
action.uuid = "42db77385ebf4768bf105011ec8e6aaf"
action.sessionId = "331167c4de264088862128e5043c5a93"
DeleteGCJobAction.Result res = action.call()
```

4.6.3 查询垃圾回收任务(QueryGCJob)

API请求

URLs

```
GET zstack/v1/gc-jobs
```

```
GET zstack/v1/gc-jobs/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 8a590138fd064e5b8747e9fa75dd6c63" \
-X GET http://localhost:8080/zstack/v1/gc-jobs?q=name=gc&q=state=Enabled
```

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth 68ab95870cb4460fb9c82360f9ed64b2" \
-X GET http://localhost:8080/zstack/v1/gc-jobs/b41ea820e0bd47d5a028ddb3eb10bb98
```

可查询字段

运行*zstack-cli*命令行工具，输入QueryGCJob并按Tab键查看所有可查询字段以及可跨表查询的资源名。

API返回

返回示例

```
{
  "inventories": [
    {
      "uuid": "c7606f2ed20b4212aa3d1ce4e00acc1c",
      "name": "TestGC",
      "type": "TimeBased",
      "createDate": "Jun 7, 2017 9:21:16 PM",
      "lastOpDate": "Jun 7, 2017 9:21:16 PM"
    }
  ]
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventories	List	详情参考*inventories*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventories

名字	类型	描述	起始版本
uuid	String	资源的UUID，唯一标示该资源	0.6
name	String	资源名称	0.6
runnerClass	String		0.6
context	String		0.6
status	String		0.6
managementNodeUuid	String		0.6
type	String		0.6
createDate	Timestamp	创建时间	0.6
lastOpDate	Timestamp	最后一次修改时间	0.6

SDK示例

Java SDK

```
QueryGCJobAction action = new QueryGCJobAction();
action.conditions = asList("name=gc","state=Enabled");
action.sessionId = "3b403b588a96471cbee74a19d4a846aa";
```

```
QueryGCJobAction.Result res = action.call();
```

Python SDK

```
QueryGCJobAction action = QueryGCJobAction()
action.conditions = ["name=gc","state=Enabled"]
action.sessionId = "76af9b84c87f4dcf9aa8f9042a982a6b"
QueryGCJobAction.Result res = action.call()
```

4.7 许可证相关接口

4.7.1 获取许可证信息(GetLicenseInfo)

API请求

URLs

```
GET zstack/v1/licenses
```

Curl示例

```
curl -H "Content-Type: application/json" \
-X GET http://localhost:8080/zstack/v1/licenses?
```

参数列表

名字	类型	位置	描述	可选值	起始版本
systemTags (可选)	List	query	系统标签		0.6
userTags (可选)	List	query	用户标签		0.6

API返回

返回示例

```
{
  "inventory": {
    "licenseType": "Free",
    "licenseRequest": "example request",
    "issuedDate": "2017-01-19 14:31:06",
    "user": "example",
    "hostNum": 10.0,
    "expired": true
  }
```

```
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventory	LicenseInventory	详情参考*inventory*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventory

名字	类型	描述	起始版本
licenseType	String		0.6
licenseRequest	String		0.6
expiredDate	String		0.6
issuedDate	String		0.6
user	String		0.6
hostNum	Integer		0.6
cpuNum	Integer		0.6
availableHostNum	Integer		0.6
availableCpuNum	Integer		0.6

名字	类型	描述	起始版本
expired	boolean		0.6

SDK示例

Java SDK

```
GetLicenseInfoAction action = new GetLicenseInfoAction();
GetLicenseInfoAction.Result res = action.call();
```

Python SDK

```
GetLicenseInfoAction action = GetLicenseInfoAction()
GetLicenseInfoAction.Result res = action.call()
```

4.7.2 获取许可证容量(GetLicenseCapabilities)

API请求

URLs

```
GET zstack/v1/licenses/capabilities
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X GET http://localhost:8080/zstack/v1/licenses/capabilities
```

参数列表

名字	类型	位置	描述	可选值	起始版本
systemTags (可选)	List	query	系统标签		0.6
userTags (可选)	List	query	用户标签		0.6

API返回

该API成功时返回一个空的JSON结构{}，出错时返回的JSON结构包含一个error字段，例如：

```
{
```

```
    "error": {
        "code": "SYS.1001",
        "description": "A message or a operation timeout",
        "details": "Create VM on KVM timeout after 300s"
    }
}
```

SDK示例

Java SDK

```
GetLicenseCapabilitiesAction action = new GetLicenseCapabilitiesAction();
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
GetLicenseCapabilitiesAction.Result res = action.call();
```

Python SDK

```
GetLicenseCapabilitiesAction action = GetLicenseCapabilitiesAction()
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
GetLicenseCapabilitiesAction.Result res = action.call()
```

4.7.3 获取附加功能许可证信息(GetLicenseAddOns)

API请求

URLs

```
GET zstack/v1/licenses/addons
```

Curl示例

```
curl -H "Content-Type: application/json" \
-X GET http://localhost:8080/zstack/v1/licenses/addons
```

参数列表

名字	类型	位置	描述	可选值	起始版本
systemTags (可选)	List	query			2.4
userTags (可选)	List	query			0.6

API返回

返回示例

```
{
  "addons": []
```

```
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
addons	List	详情参考*addons*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#addons

名字	类型	描述	起始版本
uuid	String	资源的UUID，唯一标示该资源	0.6
name	String	资源名称	0.6
licenseType	String		0.6
expiredDate	String		0.6
issuedDate	String		0.6
modules	List		0.6
expired	boolean		0.6

SDK示例

Java SDK

```
GetLicenseAddOnsAction action = new GetLicenseAddOnsAction();
GetLicenseAddOnsAction.Result res = action.call();
```

Python SDK

```
GetLicenseAddOnsAction action = GetLicenseAddOnsAction()
GetLicenseAddOnsAction.Result res = action.call()
```

4.7.4 删除许可证文件(DeleteLicense)

API请求

URLs

```
DELETE zstack/v1/licenses/mn/{managementNodeUuid}/actions
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X DELETE http://localhost:8080/zstack/v1/licenses/mn/901e7d6871d14ff1b5751e9646e34056
/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuid (可选)	String	body	资源的UUID，唯一标示该资源		0.6
managementNodeUuid	String	url	管理节点的UUID		0.6
systemTags (可选)	List	body			0.6
userTags (可选)	List	body			0.6

API返回

该API成功时返回一个空的JSON结构{}，出错时返回的JSON结构包含一个error字段，例如：

```
{
   "error": {
      "code": "SYS.1001",
      "description": "A message or a operation timeout",
      "details": "Create VM on KVM timeout after 300s"
   }
}
```

SDK示例

Java SDK

```
DeleteLicenseAction action = new DeleteLicenseAction();
action.managementNodeUuid = "1814c18f7942494ba5919e6c5c2af55a";
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
DeleteLicenseAction.Result res = action.call();
```

Python SDK

```
DeleteLicenseAction action = DeleteLicenseAction()
action.managementNodeUuid = "48e101f104474087927ae3d07d376205"
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
DeleteLicenseAction.Result res = action.call()
```

4.7.5 重新加载许可证(ReloadLicense)

API请求

URLs

```
PUT zstack/v1/licenses/actions
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "reloadLicense": {
    "managementNodeUuids": [
      "839034e1da8e37a08b67e3a8279a1650",
      "7869acf9b8bd34079f14a93751e60810"
    ]
  },
  "systemTags": [],
  "userTags": []
```

```
}
```

注: 上述示例中systemTags、userTags字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X PUT -d '{"reloadLicense":{"managementNodeUuids":["839034e1da8e37a08b67e3a8279a1650","7869acf9b8bd34079f14a93751e60810"]}}' \
http://localhost:8080/zstack/v1/licenses/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
managementNodeUuids (可选)	List	body(包含在**reloadLicense**结构中)	管理节点UUID		0.6
systemTags (可选)	List	body	系统标签		0.6
userTags (可选)	List	body	用户标签		0.6

API返回

返回示例

```
{
  "inventory": {
    "licenseRequest": "eyJwcml2YXRlS2V5IjoiLS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVkt
LS0tLVxuTUlJRWhnSUJBQUtCL0RjMGtxWGl2UmRPTGRWeC8xK1MzMUtaejNQd1d
kek9ob3NvY29rU2wwdnBxYzc2cHF1V1xubDNwQ0FWZ01UaG1OUlo1Z05xaVM2MW
dvcmY4WkY3eHBGUWd1cms0WjZxVEl3aDc3emo2NEhPMHBmYnpndzFmL1xuMm9Ga
3l1N25GQU5wamxMUEpxalYva0xwL3RVeGpqY0w0ZG9TcDY1TGM3a2lrOXY1aTBp
eXlYZEs5dWJsTnFNNVxuV2VNNWdFLzdzeG5IZS9TbFY1Nk1UeWtDeUdpbGVMMmw
vUFlmQ2JuYUN4ZG9oOERzKzJ2RnpJaFZ3RWwyYkppZ1xuQkptMlU3VzNKZzltWGZEb2dDT
HMyMTE4TXdDUGFQVWVMRTArWlBBNUduL2ZxZUZad2hDMElMcnM3ZTZyZThxK1xu
VWpBN2RXeVoxVm50Y0xiWGUxbW94Vm1DSWx4VWd3SURBUUFCQW9IOE1JMXRXTFk
xQmhVOUhGZVplVTRzV1hKclxuVVJXWGd5cWtqQ2t6cXJpT0ZqYllvUEF6dS8vREU4U1BlQ
WZDZkppU0hUdVRkdWxZVVAzZUlkakkxbTA1RVN3MVxuSXYyS01BbGRLWERmTUt3
dXpLck1LbDBua1BLMkUwMDVhYW9VSVBncU0weDdQTXlRUmZ1VW81NWs3S2JucVN
yMVxuRE1iUmJZY3FzYklrdlBoNGhOK3V0bHNvb3dFM0FZQ1FxNjkrL3NkVzhRd0V6dTRiO
G9oYzFCWlJXUC9MdHhrUVxuemdxSVo4Qkw1R1BoLy9RN2xEeGZEa1d4Nkpia0o5
blEyWVJKbnBSZGhYTk1mZndValpRRG9wcmhKbi9lektSdFxueWZDc1d5UUttSjBvMlJYVz
BEczdycS9SeXJVelc5UElseHlqSnBBSEhMeUI0UndRUUhqbGVZczVBbjV6a0xSU1xuNGlh
ZFRGZFVZdi9UV0xvcVRtQ25ZNURmUWlnUTBqajNNRE0xRERmWUtKa2szazBHSkh
6cjQ2WlZiaWRqMEdhU1xuaXNqc3l3RERtN2wxM1BSL1ZFRVo0bDk5L2F2OFY0bGs5ZkZVY
mp5MG5sWEZZcjAzZ2NLZDBCYnlPbk84dU5MM1xuOUpWeG9GVUNqaG1HSm1WQk1t
TXFXMjN1L013Y1l5b2pnbjhDZm5wS2RTWHd2ZCtrSzUycEwyWC9kQlBYcnhFU1xuT2hyYX
pVN1lPamdKM2VuWVpVRG5oOEhFOGtzRTVVSk9RZDZ4enU4N2xmU1FkTEZnT29Dd
```

zFlSVVZNTl3TFdxTVxuRE9qaXdXdDlQdjBGSnFtOVBKaHpkTmZ3dVJhSU9QbDI3WTZ5ejV
RQkRrcnIvOTdtODB6d21KYWxhNjkwV2VPbFxuSU9jcE85ekVqUk1qL1FKK09DUGF4Y3poR
DFjUHNZbnFWMlJxQTdWUmxOOTBlRkN1SUFYUDhKMDBzeGQ5WlgvTFxuYmJLa1hs
Q0pjeHFES2tzNXpXNW05ZGQvNWpCejdaS1AxT2NDeTN4RnArcGxrMUlSSTZQdXR
lSVlXM2kxK3RSVVxuUEhCTUZWUHZGRFVMZ0o3T3JLWm9ybFhkNTNZSTloWEw1bl
J1aWU2TlovMUVlVzNMUmhzU1ZHL0hBbjVrV2NOR1xuRGdhUmhiTnczZ3VTSkkwSlYxMUdv
ZkZpWmhoQjBmZjhVa1J6QjJzWFZxYzFYN3IveXBXaTBuc0ZGbDNjUE1ZeFxuK1hSNHRhRn
NSekJNSkRVWCtjS2JBeS80ejFLdk5RakNNajdtM08zam9DTEtyTFFGaEdWVWVMNDU4Z3Vh
eFQ4SVxuUmg4aXQ5d0JVQkcyMVNuSHdhK2R1YnlGVkhYV0ZNZzdZN1VDZm1TQWl
yd0V0UFVkVUJId1dwY2Q4aFlqdHU1QlxuZWkzajQ5R1lmV1JYand0N3lLNnJheVpldE9Ke
kJLNGZ5Ynp1a0ZjbkF4TzRtTmxGZnI5cWprc0tmTXBWMGVyU1xuTHYwVjhnSDRRYVZpQ0h
qcDYxR1ZaeHcyaUp1OUxsZmNsK2x0Wjh0dURaNzA5MjRFWkpUY1FNN0RjQVcwY3ZKZlxud
XFCNFRGYWNzbFNBTFFcdTAwM2RcdTAwM2Rcbi0tLS0tRU5EIFJTQSBQUklWQVRF
IEtFWS0tLS0tXG4iLCJsaWNlbnNlUmVxdWVzdCI6ImV5SjBhSFZ0WW5CeWFXNTBJam9pWl
hsS2FtTklWblJpTWxKc1lrTkpOa2xyYkhWa1IxWnpTMFpKY0VsRlRuWmpiVlZ2VmtVd2NF
bEhhek5NVkZFelQxUkJaMUV4UWxaSlJVRm5UWGswTWsxRlpFbGxhVWx6U1cxT2QyUllUV2
xQYVVsNVNXbDNhVnB0UlRaWmFrazJXbTFKTmxsNlZUWk9la0UyVFVSQmFVOXBTb
XhrUjJoc1kybEpjMGx0YUhaak0xSjFXVmN4YkVscWlybE5WRUYwVFVNd2VFNTZW
WFJOYWxFMVNXbDNhV0pYVm5SaFZ6VnlXV2xKTmtscVRUUlBSRVUxVFhwWmFVeER
TbnBsV0U0eFpGZHNhMGxxYjJsTlZWbDZUV3BHUjFKVVRYUk5ha0Y0VVhrd01GRjZRa05NV
lVWNVVrVlZkRkZxWjNsTmFsRjVUa1ZaTVZKRVkzcEphWGRwWkcxV2VXTXliSFppYVVrMlN
XcEJkVTFUU2praUxDSndkV0pyWlhraU9pSXRMUzB0TFVKRlIwbE9JRkJWUWt4SlF5QkxSV
mt0TFMwdExWeHVUVWxKUWtoRVFVNUNaMnR4YUd0cFJ6bDNNRUpCVVVWR1FVRlBR
MEZSYTBGTlNVbENRa0ZMUWk5RVl6QnJjVmhwZGxKa1QweGtWbmd2TVN0VE16Rkx
XbHh1ZWpOUWQxZGtlazlvYjNOdlkyOXJVMnd3ZG5CeFl6YzJjSEYxVjJ3emNFTkJWbWROV
kdodFRsSmFOV2RPY1dsVE5qRm5iM0ptT0ZwR04zaHdSbEZuZFZ4dWNtczBXalp4VkVsM2F
EYzNlbW8yTkVoUE1IQm1ZbnBuZHpGbUx6SnZSbXQ1ZFRkdVJrRk9jR3BzVEZCS2
NXcFdMMnRNY0M5MFZYaHFhbU5NTkdSdlUxeHVjRFkxVEdNM2EybHJPWFkxYVRCc
GVYbFlaRXM1ZFdKc1RuRk5OVmRsVFRWblJTODNjM2h1U0dVdlUyeFdOVFpOVkhsclEzbEh
hV3hsVERKc0wxQlpabHh1UTJKdVlVTjRaRzlvT0VSekt6SjJSbnBKYUZaM1JXd3lZa3BwW
jBKS2JUSlZOMWN6U21jNWJWaG1SRzluUTB4ek1qRXhPRTEzUTFCaFVGVmxURVV3
SzF4dVdsQkJOVWR1TDJXA12SAWlRoeEsxVnFRVGRrVjNsYU1WWnVkR05NWWxobE
1XMXZlRlp0UTBsc2VGVm5kMGxFUVZGQlFseHVMUzB0TFMxRlRrUWdVRlZDVEVsR
ElFdEZXUzB0TFMwdFhHNGlmUVx1MDAzZFx1MDAzZCJ9",
    "expiredDate": "2017-02-05 19:44:21",
    "issuedDate": " 2017-01-06 19:44:21",
    "user": " example@mevoco.com",
    "hostNum": 10.0,
    "expired": true
  }
}

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventory	LicenseInventory	详情参考*inventory*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventory

名字	类型	描述	起始版本
licenseType	String		0.6
licenseRequest	String		0.6
expiredDate	String		0.6
issuedDate	String		0.6
user	String		0.6
hostNum	Integer		0.6
cpuNum	Integer		0.6
availableHostNum	Integer		0.6
availableCpuNum	Integer		0.6
expired	boolean		0.6

SDK示例

Java SDK

```
ReloadLicenseAction action = new ReloadLicenseAction();
action.managementNodeUuids = asList("839034e1da8e37a08b67e3a8279a1650","7869acf
9b8bd34079f14a93751e60810");
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
```

```
ReloadLicenseAction.Result res = action.call();
```

Python SDK

```
ReloadLicenseAction action = ReloadLicenseAction()
action.managementNodeUuids = [839034e1da8e37a08b67e3a8279a1650, 7869acf9b8bd34
079f14a93751e60810]
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
ReloadLicenseAction.Result res = action.call()
```

4.7.6 更新许可证信息(UpdateLicense)

API请求

URLs

```
PUT zstack/v1/licenses/mn/{managementNodeUuid}/actions
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "updateLicense": {
    "license": "this is license string"
  },
  "systemTags": [],
  "userTags": []
}
```

注: 上述示例中systemTags、userTags字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X PUT -d '{"updateLicense":{"license":"this is license string"}}' \
http://localhost:8080/zstack/v1/licenses/mn/f03b0d69643038f9812edbf4b259cac1/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
managem entNodeUuid	String	url	管理节点uuid		0.6

名字	类型	位置	描述	可选值	起始版本
license	String	body(包含在**updateLicense**结构中)	进行过base64 encode的license内容		0.6
systemTags (可选)	List	body			0.6
userTags (可选)	List	body			0.6

API返回

返回示例


```
{
  "inventory": {
    "licenseRequest": "eyJwcml2YXRlS2V5IjoiLS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVkt
LS0tLVxuTUlJRWhnSUJBQUtCL0RjMGtxWGl2UmRPTGRWeC8xK1MzMUtaejNQd1d
kek9ob3NvY29rU2wwdnBxYzc2cHF1V1xubDNwQ0FWZ01UaG1OUlo1Z05xaVM2MW
dvcmY4WkY3eHBGUWd1cms0WjZxVEl3aDc3emo2NEhPMHBmYnpndzFmL1xuMm9Ga
3l1N25GQU5wamxMUEpxalYva0xwL3RVeGpqY0w0ZG9TcDY1TGM3a2lrOXY1aTBp
eXlYZEs5dWJsTnFNNVxuV2VNNWdFLzdzeG5IZS9TbFY1Nk1UeWtDeUdpbGVMMmw
vUFlmQ2JuYUN4ZG9oOERzKzJ2RnpJaFZ3RWwyYkppZ1xuQkptMlU3VzNKZzltWGZEb2dDT
HMyMTE4TXdDUGFQVWVMRTArWlBBNUduL2ZxZUZad2hDMElMcnM3ZTZyZThxK1xu
VWpBN2RXeVoxVm50Y0xiWGUxbW94Vm1DSWx4VWd3SURBUUFCQW9IOE1JMXRXTFk
xQmhVOUhGZVplVTRzV1hKclxuVVJXWGd5cWtqQ2t6cXJpT0ZqYllvUEF6dS8vREU4U1BlQ
WZDZkppU0hUdVRkdWxZVVAzZUlkakkxbTA1RVN3MVxuSXYyS01BbGRLWERmTUt3
dXpLck1LbDBua1BLMkUwMDVhYW9VSVBncU0weDdQTXlRUmZ1VW81NWs3S2JucVN
yMVxuRE1iUmJZY3FzYklrdlBoNGhOK3V0bHNvb3dFM0FZQ1FxNjkrL3NkVzhRd0V6dTRiO
G9oYzFCWlJXUC9MdHhrUVxuemdxSVo4Qkw1R1BoLy9RN2xEeGZEa1d4Nkpia0o5
blEyWVJKbnBSZGhYTk1mZndValpRRG9wcmhKbi9lektSdFxueWZDc1d5UUttSjBvMlJYVz
BEczdycS9SeXJVelc5UElseHlqSnBBSEhMeUI0UndRUUhqbGVZczVBbjV6a0xSU1xuNGlh
ZFRGZFVZdi9UV0xvcVRtQ25ZNURmUWlnUTBqajNNRE0xRERmWUtKa2szazBHSkh
6cjQ2WlZiaWRqMEdhU1xuaXNqc3l3RERtN2wxM1BSL1ZFRVo0bDk5L2F2OFY0bGs5ZkZVY
mp5MG5sWEZZcjAzZ2NLZDBCYnlPbk84dU5MM1xuOUpWeG9GVUNqaG1HSm1WQk1t
TXFXMjN1L013Y1l5b2pnbjhDZm5wS2RTWHd2ZCtrSzUycEwyWC9kQlBYcnhFU1xuT2hyYX
pVN1lPamdKM2VuWVpVRG5oOEhFOGtzRTVVSk9RZDZ4enU4N2xmU1FkTEZnT29Dd
zFlSVVZNTl3TFdxTVxuRE9qaXdXdDlQdjBGSnFtOVBKaHpkTmZ3dVJhSU9QbDI3WTZ5ejV
RQkRrcnIvOTdtODB6d21KYWxhNjkwV2VPbFxuSU9jcE85ekVqUk1qL1FKK09DUGF4Y3poR
DFjUHNZbnFWMlJxQTdWUmxOOTBlRkN1SUFYUDhKMDBzeGQ5WlgvTFxuYmJLa1hs
Q0pjeHFES2tzNXpXNW05ZGQvNWpCejdaS1AxT2NDeTN4RnArcGxrMUlSSTZQdXR
lSVlXM2kxK3RSVVxuUEhCTUZWUHZGRFVMZ0o3T3JLWm9ybFhkNTNZSTloWEw1bl
J1aWU2TlovMUVlVzNMUmhzU1ZHL0hBbjVrV2NOR1xuRGdhUmhiTnczZ3VTSkkwSlYxMUdv
ZkZpWmhoQjBmZjhVa1J6QjJzWFZxYzFYN3IveXBXaTBuc0ZGbDNjUE1ZeFxuK1hSNHRhRn
NSekJNSkRVWCtjS2JBeS80ejFLdk5RakNNajdtM08zam9DTEtyTFFGaEdWVWVMNDU4Z3Vh
eFQ4SVxuUmg4aXQ5d0JVQkcyMVNuSHdhK2R1YnlGVkhYV0ZNZzdZN1VDZm1TQWl
yd0V0UFVkVUJId1dwY2Q4aFlqdHU1QlxuZWkzajQ5R1lmV1JYand0N3lLNnJheVpldE9Ke
kJLNGZ5Ynp1a0ZjbkF4TzRtTmxGZnI5cWprc0tmTXBWMGVyU1xuTHYwVjhnSDRRYVZpQ0h
qcDYxR1ZaeHcyaUp1OUxsZmNsK2x0Wjh0dURaNzA5MjRFWkpUY1FNN0RjQVcwY3ZKZlxud
XFCNFRGYWNzbFNBTFFcdTAwM2RcdTAwM2Rcbi0tLS0tRU5EIFJTQSBQUklWQVRF
IEtFWS0tLS0tXG4iLCJsaWNlbnNlUmVxdWVzdCI6ImV5SjBhSFZ0WW5CeWFXNTBJam9pWl
hsS2FtTklWblJpTWxKc1lrTkpOa2xyYkhWa1IxWnpTMFpKY0VsRlRuWmpiVlZ2VmtVd2NF
bEhhek5NVkZFelQxUkJaMUV4UWxaSlJVRm5UWGswTWsxRlpFbGxhVWx6U1cxT2QyUllUV2
xQYVVsNVNXbDNhVnB0UlRaWmFrazJXbTFKTmxsNlZUWk9la0UyVFVSQmFVOXBTb
XhrUjJoc1kybEpjMGx0YUhaak0xSjFXVmN4YkVscWIybE5WRUYwVFVNd2VFNTZW
```

WFJOYWxFMVNXbDNhV0pYVm5SaFZ6VnlXV2xKTmtscVRUUlBSRVUxVFhwWmFVeER
TbnBsV0U0eFpGZHNhMGxxYjJsTlZWbDZUV3BHUjFKVVRYUk5ha0Y0VVhrd01GRjZRa05NV
lVWNVVrVlZkRkZxWjNsTmFsRjVUa1ZaTVZKRVkzcEphWGRwWkcxV2VXTXliSFppYVVrMlN
XcEJkVTFUU2praUxDSndkV0pyWlhraU9pSXRMUzB0TFVKRlIwbE9JRkJWUWt4SlF5QkxSV
mt0TFMwdExWeHVUVWxKUWtoRVFVNUNaMnR4YUd0cFJ6bDNNRUpCVVVWR1FVRlBR
MEZSYTBGTlNVbENRa0ZMUWk5RVl6QnJjVmhwZGxKa1QweGtWbmd2TVN0VE16Rkx
XbHh1ZWpOUWQxZGtlazlvYjNOdlkyOXJVMnd3ZG5CeFl6YzJjSEYxVjJ3emNFTkJWbWROV
kdodFRsSmFOV2RPY1dsVE5qRm5iM0ptT0ZwR04zaHdSbEZuZFZ4dWNtczBXalp4VkVsM2F
EYzNlbW8yTkVoUE1lQm1ZbnBuZHpGbUx6SnZSbXQ1ZFRkdVJrRk9jR3BzVEZCS2
NXcFdMMnRNY0M5MFZYaHFhbU5NTkdSdlUxeHVjRFkxVEdNM2EybHJPWFkxYVRCc
GVYbFlaRXM1ZFdKc1RuRk5OVmRsVFRWblJTODNjM2h1U0dVdlUyeFdOVFpOVkhsclEzbEh
hV3hsVERKc0wxQlpabHh1UTJKdVlVTjRaRzlvT0VSekt6SjJSbnBKYUZaM1JXd3lZa3BwW
jBKS2JUSlZOMWN6U21jNWJWaG1SRzluUTB4ek1qRXhPRTEzUTFCaFVGVmxURVV3
SzF4dVdsQkJOVWR1TDJXA12SAWlRoeEsxVnFRVGRrVjNsYU1WWnVkR05NWWxobE
1XMXZlRlp0UTBsc2VGVm5kMGxFUVZGQlFseHVMUzB0TFMxRlRrUWdVRlZDVEVsR
ElFdEZXUzB0TFMwdFhHNGlmUVx1MDAzZFx1MDAzZCJ9",
    "expiredDate": "2017-02-05 19:44:21",
    "issuedDate": " 2017-01-06 19:44:21",
    "user": " example@mevoco.com",
    "hostNum": 10.0,
    "expired": true
  }
}

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	0.6
inventory	LicenseInventory	详情参考*inventory*	0.6

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	0.6
description	String	错误的概要描述	0.6
details	String	错误的详细信息	0.6
elaboration	String	保留字段，默认为null	0.6
opaque	LinkedHashMap	保留字段，默认为null	0.6
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	0.6

#inventory

名字	类型	描述	起始版本
licenseType	String		0.6
licenseRequest	String		0.6
expiredDate	String		0.6
issuedDate	String		0.6
user	String		0.6
hostNum	Integer		0.6
cpuNum	Integer		0.6
availableHostNum	Integer		0.6
availableCpuNum	Integer		0.6
expired	boolean		0.6

SDK示例

Java SDK

```
UpdateLicenseAction action = new UpdateLicenseAction();
action.managementNodeUuid = "f03b0d69643038f9812edbf4b259cac1";
action.license = "this is license string";
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
UpdateLicenseAction.Result res = action.call();
```

Python SDK

```
UpdateLicenseAction action = UpdateLicenseAction()
action.managementNodeUuid = "f03b0d69643038f9812edbf4b259cac1"
action.license = "this is license string"
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
```

```
UpdateLicenseAction.Result res = action.call()
```

4.8 长时任务相关接口

4.8.1 提交长时任务(SubmitLongJob)

API请求

URLs

```
POST zstack/v1/longjobs
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "params": {
    "name": "migrate-volume",
    "description": "migrate volume to another Ceph primary storage",
    "jobName": "APIPrimaryStorageMigrateVolumeMsg",
    "jobData": "{\"volumeUuid\":\"45a53d3d93384433add8ead7616586cf\", \"dstPrimaryStor
ageUuid\":\"70a0618804864b3dabe8be9824c8028c\"}",
    "targetResourceUuid": "45a53d3d93384433add8ead7616586cf"
  },
  "systemTags": [],
  "userTags": []
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X POST -d '{"params":{"name":"migrate-volume","description":"migrate volume to another
 Ceph primary storage","jobName":"APIPrimaryStorageMigrateVolumeMsg","jobData
":"{\"volumeUuid\":\"45a53d3d93384433add8ead7616586cf\", \"dstPrimaryStorageUuid\":
\"70a0618804864b3dabe8be9824c8028c\"}","targetResourceUuid":"45a53d3d93384433add8e
ad7616586cf"}}' \
http://localhost:8080/zstack/v1/longjobs
```

参数列表

名字	类型	位置	描述	可选值	起始版本
name (可选)	String	body(包含在**params**结构中)	资源名称		2.3
description (可选)	String	body(包含在**params**结构中)	资源的详细描述		2.3
jobName	String	body(包含在**params**结构中)	任务名称		2.3
jobData	String	body(包含在**params**结构中)	任务数据		2.3
resourceUuid (可选)	String	body(包含在**params**结构中)	资源UUID		2.3
systemTags (可选)	List	body	系统标签		2.3
userTags (可选)	List	body	用户标签		2.3
targetResourceUuid (可选)	String	body(包含在**params**结构中)			2.3

API返回

返回示例

```
{
  "inventory": {
    "uuid": "23598186b9ef35aa89fbebf8f04d497a"
  }
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	2.3

名字	类型	描述	起始版本
inventory	LongJobInventory	详情参考*inventory*	2.3

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	2.3
description	String	错误的概要描述	2.3
details	String	错误的详细信息	2.3
elaboration	String	保留字段，默认为null	2.3
opaque	LinkedHashMap	保留字段，默认为null	2.3
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	2.3

#inventory

名字	类型	描述	起始版本
uuid	String	资源的UUID，唯一标示该资源	2.3
name	String	资源名称	2.3
description	String	资源的详细描述	2.3
apiId	String	用于关联TaskProgress的APIID	2.3
jobName	String	任务名称	2.3
jobData	String	任务数据	2.3
jobResult	String	任务结果	2.3
targetResourceUuid	String	目标资源UUID	2.3
managementNodeUuid	String	管理节点UUID	2.3
createDate	Timestamp	创建时间	2.3
lastOpDate	Timestamp	最后一次修改时间	2.3

名字	类型	描述	起始版本
state	LongJobState	详情参考*state*	2.3

#state

名字	类型	描述	起始版本
name	String	资源名称	2.3
ordinal	int		2.3

SDK示例

Java SDK

```
SubmitLongJobAction action = new SubmitLongJobAction();
action.name = "migrate-volume";
action.description = "migrate volume to another Ceph primary storage";
action.jobName = "APIPrimaryStorageMigrateVolumeMsg";
action.jobData = "{"volumeUuid":"45a53d3d93384433add8ead7616586cf", "dstPrimaryStor
ageUuid":"70a0618804864b3dabe8be9824c8028c"}";
action.targetResourceUuid = "45a53d3d93384433add8ead7616586cf";
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
SubmitLongJobAction.Result res = action.call();
```

Python SDK

```
SubmitLongJobAction action = SubmitLongJobAction()
action.name = "migrate-volume"
action.description = "migrate volume to another Ceph primary storage"
action.jobName = "APIPrimaryStorageMigrateVolumeMsg"
action.jobData = "{"volumeUuid":"45a53d3d93384433add8ead7616586cf", "dstPrimaryStor
ageUuid":"70a0618804864b3dabe8be9824c8028c"}"
action.targetResourceUuid = "45a53d3d93384433add8ead7616586cf"
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
```

```
SubmitLongJobAction.Result res = action.call()
```

4.8.2 删除长时任务(DeleteLongJob)

API请求

URLs

```
DELETE zstack/v1/longjobs/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X DELETE http://localhost:8080/zstack/v1/longjobs/a87ae356c6583cc7b312412337f96609?
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuid	String	url	资源的UUID，唯一标示该资源		2.3
systemTags (可选)	List	body	系统标签		2.3
userTags (可选)	List	body	用户标签		2.3

API返回

该API成功时返回一个空的JSON结构{}，出错时返回的JSON结构包含一个error字段，例如：

```
{
    "error": {
        "code": "SYS.1001",
        "description": "A message or a operation timeout",
        "details": "Create VM on KVM timeout after 300s"
    }
}
```

SDK示例

Java SDK

```
DeleteLongJobAction action = new DeleteLongJobAction();
```

```
action.uuid = "a87ae356c6583cc7b312412337f96609";
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
DeleteLongJobAction.Result res = action.call();
```

Python SDK

```
DeleteLongJobAction action = DeleteLongJobAction()
action.uuid = "a87ae356c6583cc7b312412337f96609"
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
DeleteLongJobAction.Result res = action.call()
```

4.8.3 查询长时任务(QueryLongJob)

API请求

URLs

```
GET zstack/v1/longjobs
GET zstack/v1/longjobs/{uuid}
```

Headers

```
Authorization: OAuth the-session-uuid
```

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X GET http://localhost:8080/zstack/v1/longjobs
```

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X GET http://localhost:8080/zstack/v1/longjobs/f031e4ea64673a5f96e022f4eb81fc59
```

可查询字段

运行*zstack-cli*命令行工具，输入QueryLongJob并按Tab键查看所有可查询字段以及可跨表查询的资源名。

API返回

返回示例

```
{
  "inventories": [
    {
      "uuid": "e8a3fcaa8c393684b78a6dc1b20db49f"
    }
  ]
```

```
}
```

名字	类型	描述	起始版本
error	ErrorCode	错误码，若不为null，则表示操作失败, 操作成功时该字段为null。 详情参考*error*	2.3
inventories	List	详情参考*inventories*	2.3

#error

名字	类型	描述	起始版本
code	String	错误码号，错误的全局唯一标识，例如SYS.1000, HOST.1001	2.3
description	String	错误的概要描述	2.3
details	String	错误的详细信息	2.3
elaboration	String	保留字段，默认为null	2.3
opaque	LinkedHashMap	保留字段，默认为null	2.3
cause	ErrorCode	根错误，引发当前错误的源错误，若无原错误，该字段为null	2.3

#inventories

名字	类型	描述	起始版本
uuid	String	资源的UUID，唯一标示该资源	2.3
name	String	资源名称	2.3
description	String	资源的详细描述	2.3
apiId	String	用于关联TaskProgress的APIID	2.3
jobName	String	任务名称	2.3
jobData	String	任务数据	2.3
jobResult	String	任务结果	2.3

名字	类型	描述	起始版本
targetResourceUuid	String	目标资源UUID	2.3
managementNode Uuid	String	管理节点UUID	2.3
createDate	Timestamp	创建时间	2.3
lastOpDate	Timestamp	最后一次修改时间	2.3
state	LongJobState	详情参考*state*	2.3

#state

名字	类型	描述	起始版本
name	String	资源名称	2.3
ordinal	int		2.3

SDK示例

Java SDK

```
QueryLongJobAction action = new QueryLongJobAction();
action.conditions = asList();
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
QueryLongJobAction.Result res = action.call();
```

Python SDK

```
QueryLongJobAction action = QueryLongJobAction()
action.conditions = []
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
```

```
QueryLongJobAction.Result res = action.call()
```

4.8.4 取消长时任务(CancelLongJob)

API请求

URLs

```
PUT zstack/v1/longjobs/{uuid}/actions
```

Headers

```
Authorization: OAuth the-session-uuid
```

Body

```
{
  "cancelLongJob": {},
  "systemTags": [],
  "userTags": []
}
```

注: 上述示例中**systemTags**、**userTags**字段可以省略。列出是为了表示body中可以包含这两个字段。

Curl示例

```
curl -H "Content-Type: application/json" \
-H "Authorization: OAuth b86c9016b4f24953a9edefb53ca0678c" \
-X PUT -d '{"cancelLongJob":{}}' \
http://localhost:8080/zstack/v1/longjobs/d1d78f09820e310e9f6305cee0dfffc5/actions
```

参数列表

名字	类型	位置	描述	可选值	起始版本
uuid	String	url	资源的UUID，唯一标示该资源		2.3
systemTags (可选)	List	body	系统标签		2.3
userTags (可选)	List	body	用户标签		2.3

API返回

该API成功时返回一个空的JSON结构{}，出错时返回的JSON结构包含一个error字段，例如：

```
{
   "error": {
      "code": "SYS.1001",
      "description": "A message or a operation timeout",
      "details": "Create VM on KVM timeout after 300s"
   }
}
```

SDK示例

Java SDK

```
CancelLongJobAction action = new CancelLongJobAction();
action.uuid = "d1d78f09820e310e9f6305cee0dfffc5";
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c";
CancelLongJobAction.Result res = action.call();
```

Python SDK

```
CancelLongJobAction action = CancelLongJobAction()
action.uuid = "d1d78f09820e310e9f6305cee0dfffc5"
action.sessionId = "b86c9016b4f24953a9edefb53ca0678c"
CancelLongJobAction.Result res = action.call()
```

术语表

区域（Zone）

ZStack中最大的一个资源定义，包括集群、二层网络、主存储等资源。

集群（Cluster）

一个集群是类似物理主机（Host）组成的逻辑组。在同一个集群中的物理主机必须安装相同的操作系统（虚拟机管理程序，Hypervisor），拥有相同的二层网络连接，可以访问相同的主存储。在实际的数据中心，一个集群通常对应一个机架（Rack）。

管理节点（Management Node）

安装系统的物理主机，提供UI管理、云平台部署功能。

计算节点（Compute Node）

也称之为物理主机（或物理机），为云主机实例提供计算、网络、存储等资源的物理主机。

主存储（Primary Storage）

用于存储云主机磁盘文件的存储服务器。支持本地存储、NFS、 Ceph、FusionStor、Shared Mount Point等类型。

镜像服务器（Backup Storage）

也称之为备份存储服务器，主要用于保存镜像模板文件。建议单独部署镜像服务器。

镜像仓库（Image Store）

镜像服务器的一种类型，可以为正在运行的云主机快速创建镜像，高效管理云主机镜像的版本变迁以及发布，实现快速上传、下载镜像，镜像快照，以及导出镜像的操作。

云主机（VM Instance）

运行在物理机上的虚拟机实例，具有独立的IP地址，可以访问公共网络，运行应用服务。

镜像（Image）

云主机或云盘使用的镜像模板文件，镜像模板包括系统云盘镜像和数据云盘镜像。

云盘（Volume）

云主机的数据盘，给云主机提供额外的存储空间，共享云盘可挂载到一个或多个云主机共同使用。

计算规格（Instance Offering）

启动云主机涉及到的CPU数量、内存、网络设置等规格定义。

云盘规格（Disk Offering）

创建云盘容量大小的规格定义。

二层网络（L2 Network）

二层网络对应于一个二层广播域，进行二层相关的隔离。一般用物理网络的设备名称标识。

三层网络（L3 Network）

云主机使用的网络配置，包括IP地址范围、网关、DNS等。

公有网络（Public Network）

由因特网信息中心分配的公有IP地址或者可以连接到外部互联网的IP地址。

私有网络（Private Network）

云主机连接和使用的内部网络。

L2NoVlanNetwork

物理主机的网络连接不采用Vlan设置。

L2VlanNetwork

物理主机节点的网络连接采用Vlan设置，Vlan需要在交换机端提前进行设置。

VXLAN网络池（VXLAN Network Pool）

VXLAN网络中的 Underlay 网络，一个 VXLAN 网络池可以创建多个 VXLAN Overlay 网络（即 VXLAN 网络），这些 Overlay 网络运行在同一组 Underlay 网络设施上。

VXLAN网络（VXLAN）

使用 VXLAN 协议封装的二层网络，单个 VXLAN 网络需从属于一个大的 VXLAN 网络池，不同 VXLAN 网络间相互二层隔离。

云路由（vRouter）

云路由通过定制的Linux云主机来实现的多种网络服务。

安全组（Security Group）

针对云主机进行第三层网络的防火墙控制，对IP地址、网络包类型或网络包流向等可以设置不同的安全规则。

弹性IP（EIP）

公有网络接入到私有网络的IP地址。

快照（Snapshot）

某一个时间点上某一个磁盘的数据备份。包括自动快照和手动快照两种类型。