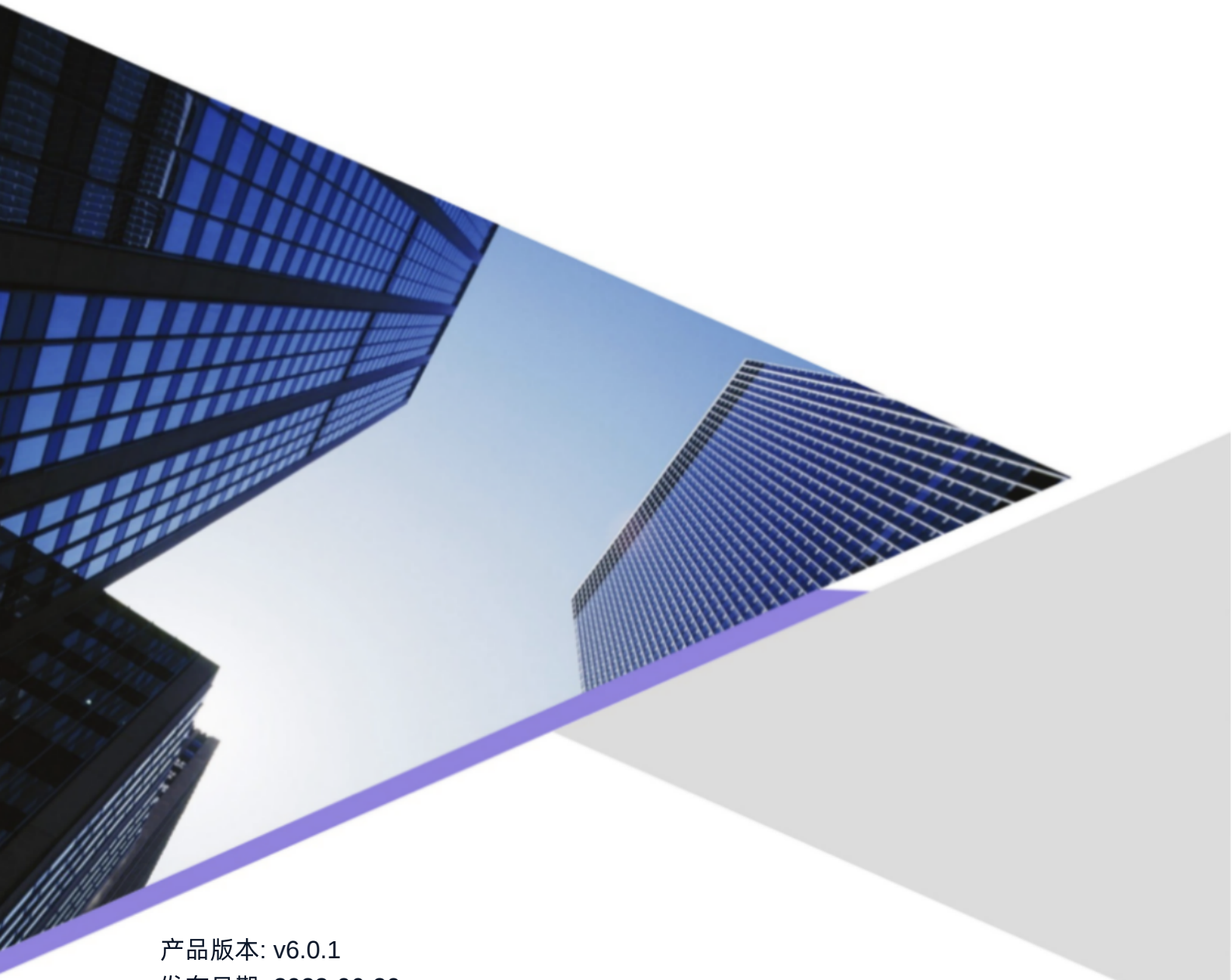
易捷行云
EasyStack

资源编排

API参考



产品版本: v6.0.1
发布日期: 2023-06-20

目录

1 API参考

1.1 API简介

欢迎使用API文档，如果您熟悉网络服务协议和一种以上编程语言，推荐您调用API管理您的资源和开发自己的应用程序。 本文档提供了API的描述、语法、参数说明及示例等内容。在调用API之前，请确保已经充分了解相关术语，详细信息请参见下表。

术语	说明
云主机	运行在云环境上的虚拟机，相当于数据中心的一台物理服务器。用户可以通过选择合适的CPU／内存／操作系统磁盘空间，网络，安全组等配置创建云主机。
云硬盘	为云主机提供块级存储设备，相当于一台物理机的硬盘。云硬盘是独立的资源，其生命周期独立于云主机，可以被挂载到任何云主机上，也可以从云主机卸载，然后挂载到其他云主机。
镜像	操作系统的安装模版，用户可以选择合适的操作系统镜像创建所需要的云主机。只有云管理员用户具有上传镜像操作权限，其他权限的用户只能使用和查看。但用户可以通过云主机快照创建新的镜像，并在启动云主机时选择”云主机快照”类型来使用新的镜像。
镜像	用户可以对云主机和云硬盘创建快照，保存当时状态下的云主机和云硬盘数据作为备份。用户可以基于这个快照创建新的云主机。云硬盘快照保存当时状态下的硬盘数据，并可以基于快照创建新的云硬盘。
物理节点	一个云环境中包含一组物理节点，每个物理节点对应一台物理服务器。物理节点可分为不同的角色，如控制节点、计算节点、存储节点和融合节点等。其中带计算角色的物理节点可以运行云主机。物理节点也可简称为“节点”。
安全组	一系列防火墙规则组成安全组，创建云主机时，用户可以选择合适的安全组来保障云主机的安全。安全组对主机上的所有网卡生效，新增网卡也将应用已有的安全组。
公网IP	独立的IP地址资源，用户可以将申请的公网IP绑定到自己的云主机上，之后便可从外部网络通过公网IP来访问云主机提供的服务。

术语	说明
SSH密钥对	基于密钥的安全验证登录方法，保证云主机安全。我们推荐使用密钥对登录云主机。
网络	网络与现实世界的交换机／路由器／服务器／连线组成的基础设施网络类似， 创建网络后，用户可以在网络内创建子网，创建云主机时选择网络，组建服务器集群。我们提供的基础网络包含共享网络和外部网络，创建在共享网络上的云主机处于同一个网络内，通过安全组保障云主机访问安全。外部网络主要用于公网IP地址的分配。 用户可以为项目创建内部网络，并在内部网络中创建子网。如同在物理网络上通过交换机将服务器连接到一起的局域网，服务器通过交换机连接到子网中。不同的内部网络之间是完全隔离的，因此不同的网络中可以配置相同的IP地址而不会产生冲突。同一个网络内可以创建多个子网，以适应业务的需求。
路由器	用户创建路由器，为不同的子网提供三层路由，从而让子网内的云主机与其他子网的云主机互联互通。也可以将用户创建的内部网络连接到外部网络，让内部网络的云主机访问Internet。路由器配置网关后，还可以为内网的云主机做端口转发，以节约公网IP地址资源。
负载均衡	用户创建负载均衡，能够将所收到的网络流量分配给若干个提供相同处理功能的虚拟机，并按照特定的算法保证每台虚拟机工作在最优的负载状态，从而达到更高效的使用计算资源的目的。这些虚拟机构成了一个集群，负载均衡会为集群设置一个对外提供服务的地址Virtual IP，外部用户通过Virtual IP实现对集群的访问。Virtual IP可以来自公网IP或者内网IP，分别提供对外和对内访问的负载均衡服务。
防火墙	防火墙提供网络间的访问控制功能，通过防火墙策略中的过滤规则对当前项目中的网络流量进行过滤。防火墙必须与一个防火墙策略相关联，防火墙策略是防火墙规则的集合，防火墙规则支持多种网络协议。
网络拓扑	展示用户当前所在项目的网络结构图。点击各个设备可以展示详细配置。
告警	用户对资源（云主机／云硬盘等）的监控数据设置告警条件，当监控数据达到阈值就会发送告警到通知列表中的邮件。
部门	部门是云平台中用户权限的一个划分层级，用户不能横跨多个部门。
项目	项目是定义资源所有权的基本单元，可理解为租户。所有资源（如云主机等）都要隶属于某个项目中。项目必须隶属于一个部门。项目名称在单个部门中的管理范围内是唯一的，但在整个云平台中可以不唯一。

术语	说明
用户	用户可以被云管理员、部门管理员创建。 用户通过登录后，可以操作云平台提供的各项资源，如云主机／云硬盘等。

1.2 调用方式

请求结构

API支持基于URI发起HTTP/HTTPS GET请求。请求参数需要包含在URI中。本文列举了GET请求中的结构解释，并以云主机的服务接入地址为例进行了说明。

结构示例

以下为一条未编码的URI请求示例： `http://cloud.com/v1/{project_id}/servers` 在本示例中：

- `http` 指定了请求通信协议
- `cloud.com` 指定了服务接入地址
- `/v1/{project_id}/servers` 为资源路径，也即API访问路径

通信协议

支持HTTP或HTTPS协议请求通信。为了获得更高的安全性，推荐您使用HTTPS协议发送请求。涉及敏感数据时，如用户密码和SSH密钥对，推荐使用HTTPS协议。

服务网址

调用本文档所列举的API时均需使用OpenStack身份服务进行身份验证。 他们还需要一个从“compute”类型的标识符提取出来的“service URI”。 这将是根URI，将添加下面的每个调用来构建一个完整的路径。例如，如果“service URI”是 `http://mycompute.pvt/compute/v2.1` ，那么“/servers”的完整API调用是 `http://mycompute.pvt/compute/v2.1/servers` 。 根据部署计算服务网址可能是http或https，自定义端口，自定义路径，并包含您的租户ID。 要知道您的部署网址的唯一方法是通过使用服务目录。计算URI不应该被硬编码在应用程序中，即使他们只希望在单一地点工作。应始终从身份令牌中发现。因此，对于本文件的其余部分，我们将使用短针，其中"GET /servers"的真正含义"GET your_compute_service_URI/servers"。

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

方法	说明

方法	说明
GET	从服务端读取指定资源的所有信息，包括数据内容和元数据（Metadata）信息，其中元数据在响应头（Response Header）中返回，数据内容在响应体（Response Body）中。
PUT	向指定的资源上传数据内容和元数据信息。如果资源已经存在，那么新上传的数据将覆盖之前的内容。
POST	向指定的资源上传数据内容。与PUT操作相比，POST的主要区别在于POST一般用来向原有的资源添加信息，而不是替换原有的内容：POST所指的资源一般是处理请求的服务，或是能够处理多块数据。
DELETE	请求服务器删除指定资源，如删除对象等。
HEAD	仅从服务端读取指定资源的元数据信息。

字符编码

请求及返回结果都使用UTF-8字符集编码。

公共参数

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

公共请求参数

名称	类型	是否必选	描述
Host	String	否（使用AK/SK认证时该字段必选）	请求的服务器信息，从服务API的URI中获取。值为hostname[:port]。端口缺省时使用默认的端口，https的默认端口为443。

名称	类型	是否必选	描述
Content-Type	String	是	消息体的类型（格式）。推荐用户使用默认值application/json，有其他取值时会在具体接口中专门说明。
Content-Length	String	否	请求body长度，单位为Byte。
X-Project-Id	String	否	project id，项目编号。
X-Auth-Token	String	否（使用Token认证时该字段必选）	用户Token。用户Token也就是调用获取用户Token接口的响应值，该接口是唯一不需要认证的接口。请求响应成功后在响应消息头（Headers）中包含的“X-Subject-Token”的值即为Token值。

公共返回参数

参数名称	参数类型	描述
RequestId	String	请求ID。无论调用接口成功与否，都会返回该参数。

签名机制

调用接口的认证方式为Token认证，通过Token认证通用请求。 Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。 Token可通过调用获取用户Token接口获取，调用本服务API需要project级别的Token，即调用获取用户Token接口时，请求body中 `auth.scope` 的取值需要选择 `project` ，如下所示：

```
{
    "auth": {
        "scope": {
            "project": {
                "domain": {
                    "name": "Default"
```

```
                },
                "name": "admin"
            }
        },
        "identity": {
            "password": {
                "user": {
                    "password": "devstacker",
                    "id": "858634b407e845f14b02bcf369225dcd0"
                }
            },
            "methods": ["password"]
        }
    }
}
```

获取Token后，再调用其他接口时，您需要在请求消息头中添加 `X-Auth-Token` ，其值即为 `Token` 。例如Token值为“ABCDEFJ....”，则调用接口时将 `X-Auth-Token: ABCDEFJ....` 加到请求消息头即可，如下所示:

```
POST https://iam.cn-north-1.mycloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

返回结果

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。 状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态。 为了便于查看和美观，API 文档返回示例均有换行和缩进等处理，实际返回结果无换行和缩进处理。

正确返回结果

接口调用成功后会返回接口返回参数和请求 ID，我们称这样的返回为正常返回。HTTP 状态码为 2xx。 以云主机的接口创建云主机 `(POST /v1/{project_id}/servers)` 为例，若调用成功，其可能的返回如下：

```
{
    "error": {
        "OS-DCF:diskConfig": "AUTO",
        "adminPass": "6NpUwoz2QDRN",
```

```
        "id": "f5dc173b-6804-445a-a6d8-c705dad5b5eb",
        "links": [
            {
                "href":
"http://openstack.example.com/v2/6f70656e737461636b20342065766572/servers/f5
dc173b-6804-445a-a6d8-c705dad5b5eb",
                "rel": "self"
            },
            {
                "href":
"http://openstack.example.com/6f70656e737461636b20342065766572/servers/f5dc1
73b-6804-445a-a6d8-c705dad5b5eb",
                "rel": "bookmark"
            }
        ],
        "security_groups": [
            {
                "name": "default"
            }
        ]
    }
}
```

错误返回结果

接口调用出错后，会返回错误码、错误信息和请求 ID，我们称这样的返回为异常返回。HTTP 状态码为 4xx 或者 5xx。

```
{
    "error": {
        "message": "The request you have made requires authentication.",
        "code": 401,
        "title": "Unauthorized"
    }
}
```

公共错误码

http状态码	Error Message	说明
300	multiple choices	被请求的资源存在多个可供选择的响应。
400	Bad Request	服务器未能处理请求。
401	Unauthorized	被请求的页面需要用户名和密码。
403	Forbidden	对被请求页面的访问被禁止。
404	Not Found	服务器无法找到被请求的页面。
405	Method Not Allowed	请求中指定的方法不被允许。
406	Not Acceptable	服务器生成的响应无法被客户端所接受。
407	Proxy Authentication Required	用户必须首先使用代理服务器进行验证，这样请求才会被处理。
408	Request Timeout	请求超出了服务器的等待时间。
409	Conflict	由于冲突，请求无法被完成。
500	Internal Server Error	请求未完成。服务异常。
501	Not Implemented	请求未完成。服务器不支持所请求的功能。
502	Bad Gateway	请求未完成。服务器从上游服务器收到一个无效的响应。
503	Service Unavailable	请求未完成。系统暂时异常。
504	Gateway Timeout	网关超时。

1.3 可视化编排

验证一个编排模板

功能介绍

验证一个编排模板。

前提条件

项目中剩余资源配置要满足资源需求。

URI

```
POST /v1/{tenant_id}/validate
```

参数	是否必选	描述
tenant_id	是	项目ID。
resource_validate	否	强制进行资源检查。当为True时，则增加资源的检查；当为False时，则不进行额外的资源检查。

请求消息

参数	参数类型	是否必选	描述
environment	object	否	编排模板的json格式环境变量。
environment_files	object	否	Files里面的有序的名字列表。
files	object	否	提供模板引用文件的内容。

参数	参数类型	是否必选	描述
ingore_errors	String	否	要忽略的以逗号为分隔的错误代码列表。
show_nested	boolean	否	设置为True当编排模板中包含嵌套的编排时。
template	object	否	执行操作的编排模板。
template_url	String	否	一个本地的编排模板地址的url。

请求示例

示例：验证编排模板

```
{
    "template_url":
"/PATH_TO_HEAT_TEMPLATES/WordPress_Single_Instance.template"
}
```

响应消息

参数	参数类型	描述
Description	String	编排模板中指定的描述信息。
ParameterGroups	array	编排模板的参数组列表。每个组包含一个参数名称列表。
Parameters	object	CFN格式的参数结构
Environment	object	编排模板的环境变量，当有多个时，这里会是一个合并的结果。

响应示例

```
{
    "Description": "A template that provides a single server instance.",
    "Parameters": {
        "server-size": {
            "default": "1GB Standard Instance",
            "description": "Server size",
            "type": "String",
            "constraints": [
                {
                    "allowed_values": [
                        "512MB Standard Instance",
                        "1GB Standard Instance",
                        "4GB Standard Instance",
                        "8GB Standard Instance"
                    ],
                    "description": "Must be a valid server size."
                }
            ]
        },
        "key_name": {
            "description": "Keypair name for SSH access to the server",
            "required": true,
            "type": "String"
        },
        "server_name": {
            "default": "My server",
            "description": "My server",
            "type": "String"
        }
    },
    "ParameterGroups": [
        {
            "label": "Parameter groups",
            "description": "My parameter groups",
            "parameters": [
                "param_name-1",
                "param_name-2"
            ]
        }
    ],
    "Environment": {
```

```
        "event_sinks": [],
        "parameter_defaults": {},
        "parameters": {},
        "resource_registry": {
            "resources": {}
        }
    }
}
```

正常响应代码

200

错误码

400，500

1.4 编排部署

创建编排部署

功能介绍

创建一个编排部署。

URI

`POST /v1/{tenant_id}/stacks`

参数	是否必选	描述
tenant_id	是	项目ID。

请求消息

参数	参数类型	是否必选	描述
disable_rollback	boolean	否	启用或禁用编排部署失败时删除所有编排资源。为True时创建会保留所有资源；为False时创建失败时清理所有资源。
environment	object	否	编排部署的json格式环境变量。
environment_files	object	否	Files里面的有序的名字列表。
files	object	否	提供编排引用文件的内容。
files_container	String	否	在swift内的容器的子编排和环境变量的名字。
parameters	object	否	为编排部署中的变量提供参数值。

参数	参数类型	是否必选	描述
stack_name	String	是	编排部署的名字。
tags	String	否	一个或者多个编排部署的标识。
template	object	否	执行操作的编排部署。
template_url	String	否	一个本地的编排部署地址的url
timeout_mins	integer	否	编排部署的超时时间。

请求示例

示例：创建编排部署

```
{
    "files": {},
    "disable_rollback": true,
    "parameters": {
        "flavor": "m1.heat"
    },
    "stack_name": "teststack",
    "template": {
        "heat_template_version": "2021-01-03",
        "description": "Simple template to test heat commands",
        "parameters": {
            "flavor": {
                "default": "m1.tiny",
                "type": "String"
            }
        },
        "resources": {
            "hello_world": {
                "type": "OS::Nova::Server",
                "properties": {
                    "key_name": "heat_key",
                    "flavor": {
                        "get_param": "flavor"
                    },
```

```
                    "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
                    "user_data": "#!/bin/bash -xv\necho \"hello world\" >
/root/hello-world.txt\n"
                }
            }
        }
    },
    "timeout_mins": 60
}
```

响应消息

参数	参数类型	描述
stack	object	编排部署的对象
id	String	编排部署的UUid
links	String	编排部署的链接

响应示例

```
{
    "stack": {
        "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
        "links": [
            {
                "href": "http://heat-
api.openstack.svc.cluster.local:8004:8004/v1/eb1c63a4f77141548385f113a28f0f5
2/stacks/teststack/3095aefc-09fb-4bc7-b1f0-f21a304e864c",
                "rel": "self"
            }
        ]
    }
}
```

正常响应代码

201

错误码

400，401，409

获取编排部署列表

功能介绍

获取编排部署列表。

URI

```
GET /v1/{tenant_id}/stacks
```

参数	是否必选	描述
tenant_id	是	项目ID。

请求消息

参数	参数类型	是否必选	描述
id	String	否	通过编排部署uuid过滤
status	String	否	通过状态来过滤编排部署列表。
name	String	否	通过名字来过滤编排部署列表。
action	String	否	通过action来过滤编排部署列表
tenant	String	否	通过项目来过滤编排部署列表。
usename	String	否	通过用户名来过滤编排部署列表。

参数	参数类型	是否必选	描述
owner_id	String	否	通过owner_id来过滤编排部署列表
limit	integer	否	每个分页的数量限制。
marker	String	否	分页看到的最后一个id。
sort_keys	String	否	编排部署列表按照某一个属性排序。
sort_dir	String	否	编排部署列表的排序方式。（asc或者desc）
show_deleted	boolean	否	资源编排列表包含已删除的编排部署。
show_nested	boolean	否	编排部署列表包含嵌套的编排部署。
tags	String	否	理出包含一个或者多个字符串tags（AND关系）的编排部署。
tags_any	String	否	列出包含一个或者多个字符串tags（OR关系）的编排部署。
not_tags	String	否	列出不包含一个或者多个字符串tags（AND关系）的编排部署。
not_tags_any	String	否	列出不包含一个或者多个字符串tags（OR关系）的编排部署。
global_tenant	boolean	否	设置为True时，返回所有项目的编排部署列表
with_count	boolean	否	设置为True时，返回带有计数的字段

请求示例

示例：验证编排模板 请求body可以为空。 `{}`

响应消息

参数	参数类型	描述
stacks	array	编排部署列表
creation_time	String	创建时间
deletion_time	String	删除时间
description	String	编排部署的描述
links	array	编排部署的链接地址列表。
parent	String	编排部署的父编排的uuid。
stack_name	String	编排部署的名称。
stack_owner	String	编排部署的所有者。
stack_status	String	编排部署的状态
stack_status_reason	String	编排部署当前状态的原因。
tags	String	编排部署的标签
updated_time	String	编排部署的更新时间
stack_user_project_id	String	编排部署用户的项目UUID

响应示例

```
{
    "stacks": [
        {
            "creation_time": "2014-06-03T20:59:46Z",
            "deletion_time": null,
            "description": "sample stack",
            "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
            "links": [
                {
                    "href":
"http://192.168.123.200:8004/v1/eb1c63a4f77141548385f113a28f0f52/stacks/simple_stack/3095aefc-09fb-4bc7-b1f0-f21a304e864c",
```

```
                    "rel": "self"
                }
            ],
            "parent": null,
            "stack_name": "simple_stack",
            "stack_owner": null,
            "stack_status": "CREATE_COMPLETE",
            "stack_status_reason": "Stack CREATE completed successfully",
            "stack_user_project_id": "71510cbd459a49ac989ca1055de7038b",
            "tags": null,
            "updated_time": null
        }
    ]
}
```

正常响应代码

200

错误码

400，401，500

获取编排部署

功能介绍

获取一个编排部署的详细信息。

URI

`GET /v1/{tenant_id}/{stack_name}/{stack_id}`

参数	是否必选	描述
tenant_id	是	项目ID。
stack_name	否	编排部署的名字。

参数	是否必选	描述
stack_id	是	编排部署的uuid。

请求消息

参数	参数类型	是否必选	描述
resolve_outputs	boolean	否	是否输出编排部署的片段。

请求示例

示例：创建编排部署 请求body为空。 `{}`

响应消息

参数	参数类型	描述
stack	object	编排部署对象。
capabilities	array	编排部署功能的列表。
creation_time	String	编排部署的创建时间。
deletion_time	String	编排部署的删除时间
description	String	编排部署的描述
disable_rollback	String	编排部署失败时资源是否回滚清理。
id	String	编排部署的uuid。
links	arrary	编排部署的链接列表。
notification_topics	array	编排部署的提醒通道。
outputs	array	编排部署的输出列表。

参数	参数类型	描述
parameters	object	编排部署的键值对参数。
stack_name	String	编排部署的名称。
stack_owner	String	编排部署的拥有者
stack_status	String	编排部署的状态
stack_status_reason	String	编排部署状态的原因
stack_user_project_id	String	编排部署用户的项目UUID
tags	String	编排部署的标签。
template_description	String	编排部署使用模板的描述。
timeout_mins	integer	编排部署的超时时间。
updated_time	String	编排部署å的更新时间。

响应示例

```
{
    "stack": {
        "capabilities": [],
        "creation_time": "2014-06-03T20:59:46Z",
        "deletion_time": null,
        "description": "sample stack",
        "disable_rollback": true,
        "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
        "links": [
            {
                "href":
"http://192.168.123.200:8004/v1/eb1c63a4f77141548385f113a28f0f52/stacks/simple_stack/3095aefc-09fb-4bc7-b1f0-f21a304e864c",
                "rel": "self"
            }
        ],
        "notification_topics": [],
```

```
        "outputs": [],
        "parameters": {
            "OS::project_id": "3ab5b02f-a01f-4f95-afa1-e254afc4a435",
            "OS::stack_id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
            "OS::stack_name": "simple_stack"
        },
        "parent": null,
        "stack_name": "simple_stack",
        "stack_owner": "simple_username",
        "stack_status": "CREATE_COMPLETE",
        "stack_status_reason": "Stack CREATE completed successfully",
        "stack_user_project_id": "65728b74-cfe7-4f17-9c15-11d4f686e591",
        "tags": null,
        "template_description": "sample stack",
        "timeout_mins": null,
        "updated_time": null
    }
}
```

正常响应代码

200

错误码

400，401，404，500

删除编排部署

功能介绍

删除一个编排部署。

URI

`DELETE /v1/{tenant_id}/stacks/{stack_name}/{stack_id}`

参数	是否必选	描述
tenant_id	是	项目ID。

请求消息

无

请求示例

示例：创建编排部署 请求body为空。 `{}`

响应消息

无

响应示例

此请求不会在响应正文中返回任何内容。

正常响应代码

204

错误码

400，401，404，500

1.5 发布记录

01 <2021-02-03>

第一次正式发布。

咨询热线：400-100-3070

北京易捷思达科技发展有限公司：
北京市海淀区西北旺东路10号院东区1号楼1层107-2号
南京易捷思达软件科技有限公司：
江苏省南京市雨花台区软件大道168号润和创智中心4栋109-110

邮箱：
contact@easystack.cn (业务咨询)
partners@easystack.cn(合作伙伴咨询)
marketing@easystack.cn (市场合作)